A Computer Aided Innovation Tool for Generating Solutions for Mechanical Engineering Functions

Master Thesis Mechanical Engineering

D. Linschoten

s1108999
DPM-1606

UNIVERSITEIT TWENTE.

Department of Design, Production & Management
University of Twente

June 2019

A Computer Aided Innovation Tool for Generating Solutions for Mechanical Engineering Functions

D. Linschoten

Abstract

Many computer aided innovation (CAI) tools exist but all either rely on a creative input of the designer or make use of some sort of database of working principles. This research proposes a CAI-tool that generates artifacts, without introducing any working principles. The artifacts generated are solutions to a mechanical engineering problem.
A framework for the new CAI-tool is developed that disengages the problem description from the actual artifact generation and optimization. For the problem description, the functional basis for engineering design by Hirtz et al. [1] is used to describe a mechanical engineering problem as a mechanical engineering function which converts a certain input flow to an output flow. This function determines the objective function for the computational optimization. Artifacts are designed to consist of basic elements that have behavior and characteristics but no functionality of their own. A computational optimization algorithm is used to create and optimize new versions of the artifacts. Better artifacts are selected based on their fitness determined by the objective function. Because no working principle is introduced to the CAI-tool, it is hypothesized that these working principles will be generated by the tool itself.

A proof of concept was made for the proposed CAI-tool based on Evosoro [2]. This software uses compositional pattern-producing networks (CPPN) that produce three-dimensional multi-material artifacts made up of voxels which are optimized using neuro-evolution of augmenting topologies (NEAT). These voxels are the basic elements of which the artifacts consist, in this proof of concept. A different population selection method, passing multiple objects and new fitness calculations are among the newly introduced or changed features to the software.
As a proof of concept, a simple mechanical problem was tested: solutions for the function 'rotational transmission' were tried to be generated. Analysis of solutions was done using a modified version of the contact and channel model to identify the difference in working principles. Multiple times, in single runs without a change of parameters, more than one working principle was identified among the generated solutions. This proves that no single working principle was predetermined and proves the ability of the proposed CAI-tool to fulfill a mechanical engineering function without prior knowledge of working principles.

Acknowledgements

I would like to thank the following people: My parents for their support throughout my study, their limitless interest in my research and proofreading this report. My girlfriend for supporting me, helping me in structuring my days and proofreading this report as well. My friends, for their interest in, enthusiasm for and discussions about my research. Wouter den Breeijen and the Multi Scale Mechanics group for the usage of their cluster and helping me debug when I really couldn't solve it. Sam Kriegman for supporting me in changing Evosoro. Eduardo Lalla, who has helped me on the topic of genetic algorithms and making sure the report is written on a scientific level. And finally: Tom Vaneker, for his supervision, insights and the freedom he has given by allowing me to pursue my own research interests that were too broad to begin with but nonetheless, by his guidance, resulted in this report.

Contents

Definitions

Age in the context of the genetic algorithm: the age of an individual is increased every generation and starts with age 0 at its (random) generation.

Artifact Physical object or (digital) representation thereof.

CPPN compositional pattern producing network, see Section 2.3.1.

Channel and Support Structures (CSS) Used in the C&CM method: forms the internal structure of a part, by connecting two *inner* ports of WSs, can store energy.

Computer aided innovation (CAI) All-encompassing term for all software that plays any role in the innovative part of new product design.

Conceiving concepts In this thesis it is useful to have a distinction between actions that are done by computers or by humans. Using *conceiving* implies that concepts are produced by an engineer or designer.

Conceptual level The function solution is determined but some parameters are not necessarily exactly established for a final product.

Connector Used in the C&CM method: integrates the system into its environment. A connector describes anything that links to a WS at the system boundary [3].

Contact and Channel Method (C&CM) is used to describe design problems and artifacts in a "systematized but free and dynamic way of modelling", in order to provide an abstract representation of the artifact [4]. Used in this research to describe the way an artifact fulfills its function.

Design method (DM) Structured design processes.

Discursive thinking is a conscious process that can be communicated, influenced and where facts and relationships are analyzed, varied, combined, checked all in a conscious way [5].

Discursive thinking A conscious process that can be communicated and influenced, where facts and relationships are analyzed, varied, combined, checked all in a conscious way [5].

EMS (energy, material, signals) model Model to describe a mechanical engineering function using an input and output flow of energy, material or signal on which the function acts. See Appendix A.3 for a list of functions and flows.

Flow Used either in the EMS model or the C&CM to describe an energy, material or signal. In the EMS model, a function acts on the input flow to some output flow. In the C&CM, a flow is conveyed through WSs or CSS. See Appendix A.3 with Table A.3 for a list of flows and Table A.4 and A.5 for the energy flows in detail.

Function
Mechanical engineering ∼ A description of an operation to be performed by a device or artifact [6]. In this research, the EMS model is used to describe a function.
Mathematical ∼ the relation that associates each input element to an output element.

Generating concepts In this thesis it is useful to have a distinction between actions that are done by computers or by humans. Using *generating* implies that concepts are produced by a computer.

Genotype In genetic algorithms: the lowest-level of encoding used in the algorithm [7]

Individual Single possible solution within a population of a genetic algorithm optimization.

Innovation From Hüsig [8]: "An iterative, interactive, context-specific, multi-activity, uncertain, path-dependent process and the result of a new combination of ends and means from a certain perspective. From this perspective, someone must perceive a difference concerning the qualitative newness of an object compared to a prior status in a given context. This new combination must be realized and introduced into a specific context which is the point of reference of the prior status."

Intuitive thinking takes place unconsciously with insights happening suddenly by association or some trigger [5].

Inventing To design or create something that has not existed before within the reference framework: the creation of function solutions with different working principles from existing function solutions.

Mechanical engineering problem A problem which can be solved by mechanical engineering functions.
Basic ∼ A problem which can be solved by a mechanical engineering function.

NEAT NeuroEvolution of Augmenting Topologies, see Section 2.3.1.

Non-predefined solution The solution to the mechanical engineering problem is not defined to have one particular working principle. Also no expert knowledge of existing solutions is used to create a new solution.

Phenotype In genetic algorithms: any higher-level expression of the genotype can be defined as the phenotype.

Physical properties and functionalities Properties and functionalities that are governed by the laws of physics, e.g. elasticity of a material, movement under influence of a force.

Population A set of individuals within a genetic algorithm optimization that are in one generation.

Solution
Function ∼ artifact that fulfills the goal of that mechanical engineering function.
Optimization ∼ output of an optimization that has a positive result for the objective function.

TRIZ Acronym for the theory of inventive problem solving. This is a Russian problem solving method that was started by analysis of existing patents and has led to the "science of creativity that relies on the study of the patterns of problems and solutions, not on the spontaneous and intuitive creativity of individuals or groups" [9].

Working Surface Pairs (WSP) Used in the C&CM method: the connection between the *outer* ports of two WS. This is where two objects interact with one or multiple flows

Working principle The interactions of an object with its surroundings that lead to its functionality. Differences between working principles are deducted by using the C&CM in this research.

Working surface (WS) Used in the C&CM method: interacts with its environment via input or output flows

1 Introduction

The use of computers in product design and production is widespread and many products would not have existed without computer technology. The computer is used to increase efficiency in design teams, to model concepts, perform simulations, to control machines and processes of production and in many more facets of the entire product design process.

However, there are opportunities to make the next step in the unexplored field of digitization in the creative part of concept generation. Now, this is still done by engineers depending on the knowledge, creativity, and ingenuity they possess at that particular moment.
Of course, there is more than one way to move forward from this situation. One is to improve the functioning of the engineer to contrive concepts by using various methods to get the best out of an engineer. Another possibility would be to include computers in the process of concept generation. Software could be developed that relies on expert knowledge: existing design knowledge [10] or input of a user.
However, the path that this research will investigate is the generation of function solutions on a conceptual level without any foreknowledge implemented in the software.

To start with, the foundation and the basics of several elements of this research will be introduced to lead to the main goal and research question.

1.1 Foundation

The basics of the elements that lead to an adequate research question for this thesis comprise a broad spectrum. The basics lie in the use of computers in innovation, computational optimization methods and algorithms, functional modeling, inventing and design methods. These are covered in the three following elements:

- Design methods, i.e. structured design processes. Using knowledge about principles of inventing and methods to analyze functionality provides the higher contextual level that enables to clearly categorize and analyze mechanical engineering problems and possible function solutions unambiguously.
- Computer aided innovation (CAI) is the all-encompassing term for all software that plays any role in the innovative part of new product design. To determine the placement of this research and relevant areas, an overview of CAI will be given.
- Computational optimization is used to find a global optimum within a solution space with the support of a computer. Many computational optimization methods exist for different optimization problems. It is important to determine which one suits the needs of a particular optimization problem.

Incremental innovation is easy for humans to do by combining existing knowledge and creating the next logical step: an improved version of a product with the same working principle. Radical innovation that is very different from the status quo on a functional level is harder to achieve. Most patents only show incremental innovation [11] but the Kano model shows that consumer satisfaction can be improved with a more innovative product [12]. One way of reaching the *blue ocean* (i.e. the level of innovation where no comparable products exist) is by radical innovations [13].

Design methods often lead to incremental innovation because the creative part is not supported in a systematic way or because they are supported by brainstorming type of techniques. These are mostly based on creative imagination or combined group knowledge.

Some design methods have a computerized version which makes those CAI-tools. For the creative part, often the input of a human designer is used. To be able to reach that *blue ocean* it would be beneficial to also reach outside the grasp of human imagination or knowledge. For this, a design method needs to be developed that works without this limitation. One apparent solution is to consider a very wide range of solutions and select the ones that work well or one that works best, which makes this method very suitable to carry out as a computerized tool if it were possible, because human evaluation of a very wide range of solutions is impractical.

Exploring a wide range of solutions systematically calls for the quantification of the innovation goal. Computational optimization can then be used to find candidates for an innovative design solution: optimums within the solution space.

1.1.1 Invention, design methods, and functional analysis

Humans have always made inventions, artifacts that had never been made before which functioned by a different working principle than anything else ever made before. Early inventions, like the rocket, were accidentally discovered [14] or evolved empirically like fins on an arrow [15]. But with the start of the scientific revolution, knowledge of the laws of physics led to inventions by people who understood scientific principles [16] such as Maxwell, who mentioned the working principle of a dynamo [17] in his investigation into electromagnetic phenomena. In more recent years various methods to structure the process of invention and new product design (NPD) have emerged.

Design methods

A new design method will be created in this research to generate solutions without using any existing design methods that rely on human creativity or databases. Computational optimization is the basis for the proposed method which itself can, of course, be seen as a design tool for some incremental innovation methods. Nonetheless, other design methods will be used to understand and analyze designs and design processes.
Design methods (DM) can be characterized along four underlying principles [18]: hierarchical decomposition, systematic variation, satisficing, and discursiveness. The first separates a design problem in distinct and independent parts that are easier to find concepts for. The second relies on searching and recombining solutions for design sub-problems. The third, satisficing, allows settling for a non-optimal solution and alternatives. And the fourth, discursiveness, divides a design problem into separate steps and transitions that are described beforehand so conscious decisions can be made and iterations can be done within them. Several of these four principles are often seen in a single method or tool.

For example, a well-known design tool is the use of a morphological chart, which combines decomposition and variation. The design problem is deconstructed into sub-problems and for every one of them, many solutions (variations) are contrived.
Another example which has a more complex method of solving problems is TRIZ [9] (the acronym for the theory of inventive problem solving). This is a Russian method that was started by analysis of existing patents and has led to the "science of creativity that relies on the study of the patterns of problems and solutions, not on the spontaneous and intuitive creativity of individuals or groups". The aspect of such a technique that relates to this research is the analytic way in which creativity is approached.
TRIZ mainly uses abstract and generalized descriptions of problems and solutions that encompass many physical occurrences (e.g. the inventive principle "partial or excessive actions") and often in relation to other existing artifacts or solutions. However, this research has the need for descriptive methods that provide unambiguous ways to segregate these physical occurrences and analyze the working principle of function solutions without relying on their possible similarities to existing artifacts or principles. Those methods perform a functional analysis. There is not one standard to perform a functional analysis (FA) but methods to describe and analyze engineering design, abstract definitions encompassing all possible design solutions do exist. Most work by combining a *flow*, describing an *energy* and a *function* that operates on the flow. Exploring these methods and deciding which method is best for use in this research, is done in the literature study in Section 2.2.2.
Besides describing artifacts on a functional level, an analysis of the degree of newness or inventiveness is required. This will help in interpreting differences between eventually generated solutions and explore possibilities in a systematic fashion. TRIZ could provide such a method as it specifies five levels of invention varying from parametric changes to the discovery of a new phenomenon. This will also be dealt with in Sections 2.2.1 and 2.2.2.

1.1.2 Computer aided innovation

While computer-aided design (CAD) and computer aided manufacturing (CAM) usage is widespread in new product development (NPD), computer-aided innovation (CAI) is not.
The definition of CAI is one with fuzzy boundaries but distinct areas within it exist. These will be explained later on in this section. The unclear definition of CAI leads directly from the unclear terminology and categorization of innovation [8]. It is also important to keep in mind that innovativeness is relative and dependent on the context, as becomes clear in the following definition used by Hüsig [8] for innovation:

"Innovation is an iterative, interactive, context-specific, multi-activity, uncertain, path-dependent process and the result of a new combination of ends and means from a certain perspective. From

this perspective, someone must perceive a difference concerning the qualitative newness of an object compared to a prior status in a given context. This new combination must be realized and introduced into a specific context which is the point of reference of the prior status."
This definition only takes into account processes that involve designers and inherently excludes innovation processes solely executed by computers.The key element in this definition lies in the reference to the specific context of the prior status of the innovated element. This statement is very relevant to this research as the specific context will be defined in relation to the limited "knowledge" of the simulation. This will be explained later, in Section 2.2.1.

Categorizing CAI

According to Kohn and Hüsig [19] applications in CAI can be divided into three main categories when based on application field and four when looking at potential benefits.
The three application fields are strategy, idea and patent management. Within the field of idea management, two elements are placed that are related closely to this research, i.e. idea generation and analysis. Looking at categories of potential benefits, it seems that this research has links with the categories of creativity and competence enhancing.
Examples of software in these categories are computerized versions of design methods for idea generation like TRIZ (Goldfire) and bio-mimicry (AskNature). Competence enhancing methods to improve designs include generative design or structural optimization software that uses various algorithms to improve an existing design.

Usually, the competence and creativity that is mentioned within CAI is of the designers themselves and not seen as something that can be emulated by software, but rather something that is enhanced by software. Even though the proposed tool in this research is not consistent with those properties of CAI, it is still seen as a CAI-tool since it has the ability to provide aid in innovation.
A more extended overview of specific CAI categories can be found in Appendix A.1.
Still, this research is placed in those categories since one could argue that any improvement, also a complete takeover of a process, leads to a better result presented by a designer and thus to the enhancement of the designers' abilities.

1.1.3 Computational optimization

As mentioned before, computational optimization is used to find a global optimum or multiple optimums with the support of a computer. This optimum is determined by means of the evaluation of one or multiple objective function(s) and changing input parameters and iterating this so a better result is achieved. An objective function describes the desirable properties that need to be maximized or minimized within an optimization. Besides an objective function, boundary conditions that limit the values of certain parameters can be introduced.
Optimization is used for many engineering problems. With the increase of computing power and better algorithms, designs influenced by simulations and optimizations differ more and more from previous human-made designs. A couple of examples are shown below.

Using sequential quadratic programming (SQP) optimization algorithm, the x and y position of the upper right-hand corner and thickness of four beams (six parameters in total) of a truss structure are optimized so the structure is as light as possible without breaking when weight is applied. The same problem was solved using a topology optimization script [20] and shows a much more "organic" structure which could be 3D-printed but also more difficult to produce conventionally.

Topological optimization has the ability to reach any shape within the design space as every pixel can be given a weight from zero to one and works by concentrating and moving mass to areas that provide a better result at every iteration. To get results that can be used, filters can be added like the middle result in Figure 1.1 as opposed to the right output which has small elements.

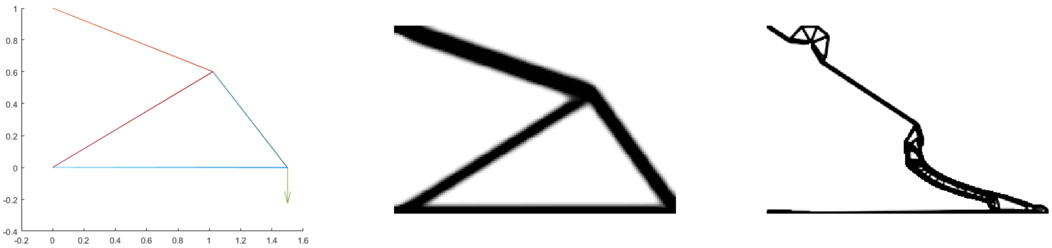
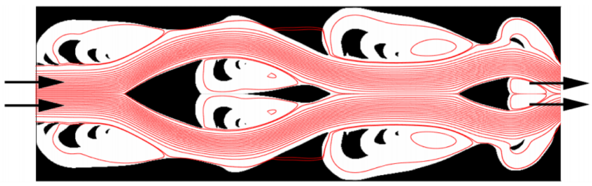
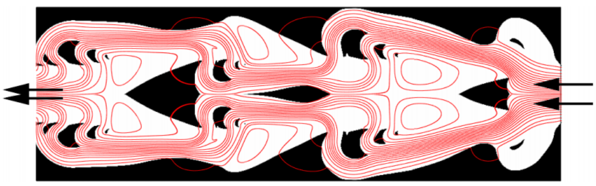


Figure 1.1. Optimization of a truss with 6 parameters (left) and using topological optimization (middle, right) [21].

An important lesson in optimization can already be learned from this: every optimization needs the right algorithm to lead to a good and workable result.
Another example shows topological optimization applied in a different field: fluid engineering. Lin et al. [22] apply topological optimization in fluid engineering to optimize a fluid diode. This structure maximizes the flow resistance to one end while it minimizes the flow resistance in the other direction. The result can be seen in Figure 1.2. In combination with a piezo-element, this can be used as a pump with the benefit that it has no moving parts.



(a) Forward flow



(b) Reverse flow

Figure 1.2. Topological optimization applied in fluid engineering for a fluid diode [22].

The next example shows the application of topologically optimized structures in 3D (see Figure 1.3) and a shape that would have been impossible to design manually.

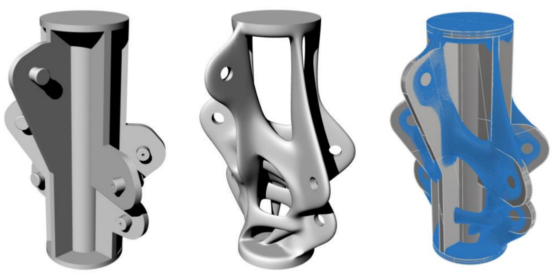


Figure 1.3. An example of topological optimization as published by Galjaard, Hofman, and Ren[23].

This way of optimizing structures is already implemented in some CAD programs like Autodesk and SolidWorks and is used for various applications. Often this is called *generative design* which can include other optimization techniques as well. Using the generative design module of Autodesk a clamp was optimized which can be seen in Figure 1.4, achieving a weight reduction of 60% while other properties still fulfill user demands [24].

Figure 1.4. A 3D-printed clamp, topologically optimized in Autodesk [24].

Multi-objective optimization is used to find an optimum for a multitude of objectives. For example, in the design of an airplane, both the bending moment and the drag of a wing needs to be minimized. Trying to minimize one, will increase the other. A combination needs to be found where both values are satisfactory and one value cannot be increased without diminishing the other. Such a point is called a pareto efficient solution. An example of these points for the wing design can be seen in in Figure 1.5. A genetic optimization algorithm was used in this optimization.

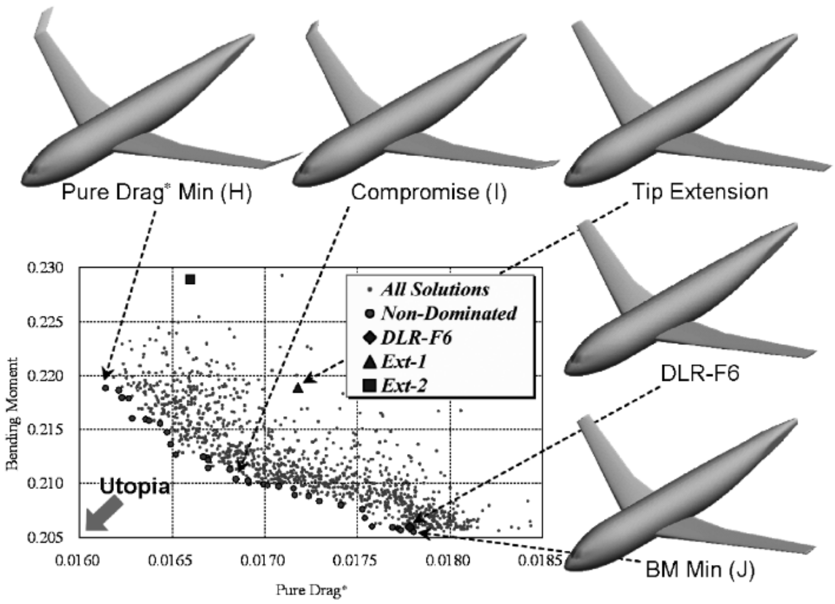


Figure 1.5. Pareto front for an optimization of the design of winglets [25].

Aforementioned examples could not have been designed by an engineer to this extent without the help of computational optimization. However, they still rely on an engineer to provide input on how to achieve the functionality that is intended by defining the working principle. Imagine the level of freedom that is displayed in the examples, not within the solution space of one object with a prescribed working principle but on a higher conceptual and functional level. This idea

is an essential part of this research.
When multiple working principles are included in one optimization, it implies that sharp transitions within the solution space exist. When a solution space is highly non-linear it becomes very hard with previously mentioned optimization techniques to obtain a result that is not just a local optimum. Changing parameters slightly at that point does not lead to better results and so it becomes stuck in that location within the solution space without ever reaching a different point with a potentially better result. Restarting with different parameters and starting searching from that point is one way to temporarily solve this problem. A method that is generally suitable for highly non-linear problems is a genetic optimization algorithm [26].

Genetic optimization algorithms

Genetic algorithms (GA) are optimization algorithms based on creating, evaluating, recombining and altering genomes that describe individual instances. In Algorithm 1, pseudo-code for a genetic algorithm is presented [27] which is described below.

```
Generate population (Pop) with n individuals;
Evaluate fitness of each individual in Pop;
while (stopping criterion not satisfied) do
    while (Pop_new < n) do
        select random individuals Pop(P_a, P_b);
        Pop_new ← crossover(P_a, P_b);
    end
    for (i=1 to |Pop_new|) do
        if random > prob_m then
            Pop ← P_i;
        else
            Pop ← mutate(P_i);
        end
    end
    Evaluate fitness of each new individual in Pop sort Pop according to fitness values;
    for (i=n+1 to |Pop|) do
        delete Pop(P_i);
    end
end
```

Algorithm 1: Pseudo-code for a GA. With *Pop*: population, n: number of individuals, Pop_{new}: new population, P_i: individual i in population *Pop*

A starting population of many individuals (possible solutions) is generated. These individuals are tested, compared with data, evaluated using simulations or evaluated in any other way. Every successful evaluation gives a fitness value as output for each individual. When all individuals are evaluated, the next step is to compare all the individuals and sort them according to their fitness. It is also possible to combine more than one parameter as a fitness function or to add a second sorting for a different variable in case multiple individuals reach the same fitness.
Now the individuals are selected which are to populate the next generation through breeding and/or mutation. Two individuals are bred by crossing over their genomes and merging those for one new individual. Mutation is done by randomly altering properties of an individual.
The selecting of individuals for these procedures could be done by simply choosing the best according to fitness, or by including certain numbers of lower scoring individuals. This leads to a more diverse population as it allows individuals to evolve through several generations before they reach their optimum. Individuals that function well are not necessarily the ones that functioned

well, or at all, in preceding generations.
The new generation is populated by the individuals from the previous generation(s), newly generated individuals, the mutated individuals, and bred individuals. When the new population is formed the evaluation step is performed again. These steps are repeated until the desired result or specified number of generations is reached.

One of the advantages of GA compared to single solution optimization algorithms is that it finds multiple solutions, works well with optimization problems with parameters and a highly non-linear objective function. Examples include: shape optimization of a vertical-axis wind turbine using a numerical simulation [28], designing of electrical circuits [29] , determining the optimal fabrication direction for layered manufacturing to minimize post-machining [30] and the generation of multiple solutions for the inverse kinematics of a serial robotic manipulator [31].

Simulation

A simulation environment is necessary to compute the fitness of each individual by performing an evaluation (for GA, or evaluating the objective function every iteration for other computational optimizations). A simulation will mimic the functioning of an object (i.e. an individual) in real-life. The properties of the simulation algorithm will influence the solution space in this case. The level of detail, possible material properties, which laws of physics and external influences can be simulated, all determine what the solution space will look like. The particular simulation environment for this research will be discussed in Section 3.3.2.

1.2 Relating inventing, design methods and CAI

Components exist that can be placed in more than one of the previously mentioned elements (Inventing, DM and FA, CAI and computational optimization).
It is important to notice the difference between most related CAI-tools and computational optimization used as a CAI-tool. Computational optimization is used in many fields and is able to solve specific (design) problems. It requires exact *objective function, design variables and constraints* and, in the case of multiple design goals, is based on the trade-off of the different goals as prioritized by the designer. However, most CAI-tools that are related to idea generation and analysis only aid in these processes by combining existing knowledge and not creating unique solutions without reference.

For the rest of this section, functional analysis (FA) will be disregarded since this is not used as a tool to create artifacts on its own, just like CAI, because it is merely a descriptive term for underlying methods and the elements of invention and design methods (DM) will be separated because they do not completely overlap.
Of course, there are also design methods or computational optimizations that do not lead to inventions and inventions that are done without any structured design method or computational optimization.
A graphic interpretation of these relations can be seen in Figure 1.6. A: Many design methods exist that do not need any computational optimization but can still lead to inventions, such as brainstorming. B: Optimization methods that can lead to inventions while relying on predefined design methods or databases. Examples of this are automatic shape and topology variations [32], generative design which includes interaction with its user [33] or product design using a colony of virtual ants [34]. Theoretically, computational optimization could result in inventions (C) but as of now, no examples or proof of concepts thereof exist. Methods are only placed in C when designing is done in a methodical way and thus only possible with some sort of database with design knowledge. D: Where this research is placed, inventions are reached through

computational optimization without predefined design methods or databases. Within itself, this could again be considered as a design method and section D would cease to exist by combining it into C. However, to keep this partition, inventions done through methodical designing and exploring the entire solution space with an algorithm are strictly separated.

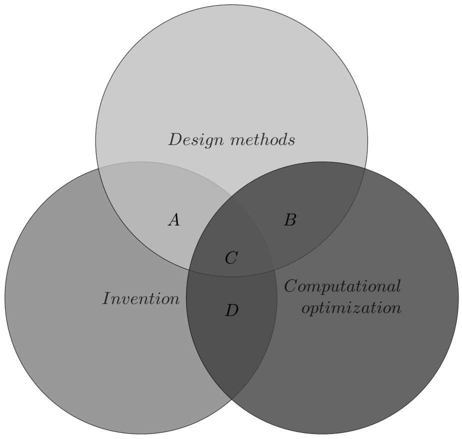


Figure 1.6. Venn-diagram which shows the relation between areas of interest for this research.

The possibilities of using computational optimization to create inventions (D) will be explored in this research. It is avoided to apply any existing design methods, reuse existing solutions (e.g. databases) or creativity (C). This would result in limiting the availability of reachable solutions within the solution space. This is because it would only allow a discrete amount of options to be explored in some fixed method of exploration and not providing total freedom of exploration of the solution space. It must be noted that the essence of the principle in itself of (systematically) considering a large variety of solutions without initial rejection, can be seen in design methods (e.g. TRIZ, brainstorming) and also shows an overlap of these design methods with how genetic optimization works.
An impressive step taken towards invention using computational optimization was done by Cheney et al. [2] who presented Evosoro. Using voxels that resemble bone, expanding or contracting muscles and soft tissue, genetic optimization is used to generate objects that fulfill the objective function. More about their work will be discussed in Section 2.3.1. In Figure 1.7, the results were obtained by defining *moving forward* as the objective function. The diversity of results shows that it not only has an application in the field of computational biology or soft robots but potentially also in a wider array of fields.

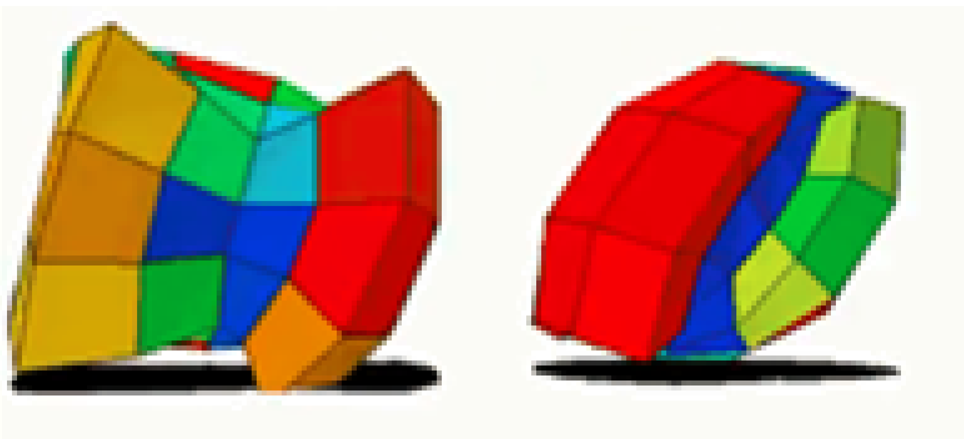


Figure 1.7. Two soft robots that move forward in different ways (walking and rolling) generated by Kriegman et al.[35]

To be more specific: these optimizations, done by Cheney et al.[2], all result in a single object that has no effect on other objects but itself. Using the definition of a mechanical engineering problem that is used in this research, the object generated has to impose a function on another object or entity. This was no goal in itself for Cheney et al. but adding this to the goal of this research will widen the spectrum of possible applications from soft robots to many mechanical engineering problems.

1.3 Research goal

The research goal will be:

Proving that it is possible to use computational optimization to generate artifacts that are function solutions for basic mechanical engineering problems on a conceptual level without introducing the working principle as input for said optimization.

1.4 Research question

The main research question will be:

A computer aided innovation tool: how can artifacts that fulfill a mechanical engineering function be generated using computational optimization without prior knowledge of working principles?

To answer this question several sub-questions are formulated in order to split the problem into multiple parts.

- **What should the structure, components and main working method of such a CAI-tool be?**
  - **Which components are needed besides the computational optimization algorithm?**
  - **How can an engineering problem be converted to an input for such a CAI-tool?**
  - **How can be detected to what extent a mechanical engineering function is fulfilled by a solution?**
  - **How can be ensured that no knowledge of working principles is introduced?**
- **How can a proof of concept be made to test such a CAI-tool based on a genetic algorithm?**
  - **Which genetic algorithm based software is suitable to be adapted for a proof of concept for such a CAI-tool?**
  - **What adaptions should be made to that software?**
  - **How can (generated) artifacts be distinguished as function solutions?**
  - **What demands need to be met for the proof of concept to be successful?**

1.5 Plan of action

Each of the sub-questions will need a different approach to arrive at a satisfactory answer.

- **What should the structure and components of such a CAI-tool be?** This requires knowledge on design methods, computational optimization and an overview of existing CAI-tools.
  - **Which components are needed besides the computational optimization algorithm?** It needs to be clear which components are necessary to adapt the problem to be suitable as input for some computational optimization algorithm.
  - **How can an engineering problem be converted to an input for such a CAI-tool?** The answer to this question might be found in fundamental research on design and engineering. Even though different views may be found in literature, abstract functions should fit this particular topic. The abstract functions need to be reduce to their most basic level at which they function.
  - **How can be detected to what extent a mechanical engineering function is fulfilled by a solution?** A quantification of functionality is necessary to be able to answer this question.
  - **How can be ensured that no knowledge of working principles is introduced?** For this, knowledge of working principles needs to be established and a method needs to be determined that is able to distinguish different working principles. Some method must be found that presents the problem to a CAI-tool without the implication of any working principle, just as the tool itself should not have any working principles included.
- **How can a proof of concept be made to test such a CAI-tool based on a genetic algorithm?** Taking into account the answers to the previous sub-questions, it is clear that a match needs to be found between the prerequisites that are implied by those answers and the possible adjustable properties of a genetic algorithm. Also clear demands for a successful proof of concept need to be set.
  - **Which genetic algorithm based software is suitable to be adapted for a proof of concept for such a CAI-tool?** To answer this question, knowledge about genetic algorithms and implications of choices in object representation, evaluation, and selection methods needs to be gathered. Software that can be adapted to serve as a proof of concept without many changes is preferable.
  - **What adaptions should be made to that software?** Demands of the CAI-tool and its proof of concept need to be compared to the chosen genetic algorithm software.
  - **How can (generated) artifacts be distinguished as function solutions?** The quantification method for functionality must be implemented in the proof of concept for this.
  - **What demands need to be met for the proof of concept to be successful?** An example problem must be found which, when solved, proves the working of the CAI-tool for all comparable problems.

2 Literature review

The objective of this literature review is to acquire an overview of related work, determining the state of the art and prove the novelty of this research. This will be done in the areas of research that were introduced before.
An overview of concept generation methods within existing design methods and CAI-tools will determine the state of the art and highlight gaps in current research.
Functional analysis and description methods of working principles will provide a background from a new product design point of view as well as determining demands for proof of concept of the CAI-tool presented in this research.
Finally, a deeper look into GA, neural networks and topological representation will provide the knowledge to create a proof of concept and execute runs.

2.1 Design methods and CAI in conceptual design

In this section, design methods will be classified and CAI-tools for each class will be searched for. The usage of this classification will identify opportunities for a new CAI-tool using creativity.

2.1.1 Design methods

The new product design (NPD) process can be divided into six sub-processes [36] which can be seen in Figure 2.1. The sub-process that is relevant for this research is the one where function solutions are generated: concept development.

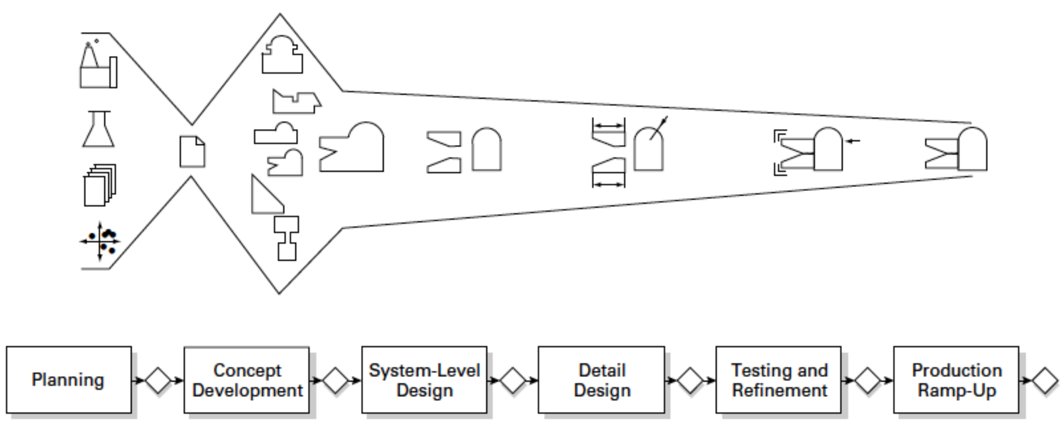


Figure 2.1. Timeline for a generic new product development process [36]

The phase in a DM where concepts are generated is also known as the ideation phase. Whereas

DMs can be classified into four principles, just two principles classify the methods used in the ideation phase itself [5]: intuitive thinking or discursive thinking.
Intuitive thinking takes place unconsciously with insights happening suddenly by association or some trigger. It is impossible to determine the time frame in which a solution is produced but ways exist in which triggers can be induced such as sketching. Discursive thinking, on the contrary, is a conscious process that can be communicated, influenced and, where facts and relationships are analyzed, varied, combined, checked all in a conscious way.
Examples of intuitive methods include brainstorming, method 6-3-5 and the gallery method. Some examples of discursive methods are TRIZ, SIT (systematic inventive thinking), and the use of design catalogs. These methods are shown at the top in Table 2.1 on page 25.

2.1.2 CAI-tools in concept generation and invention software

Generating concepts for mechanical engineering problems requires knowledge about the problem: specifications, requirements, available production processes, but also general knowledge on relevant physical phenomena and so on. These are the input for every DM, and it is important to establish how this can be achieved for a CAI-tool. A look at an example of a computational intelligence-based process model for conceptual design proposed by Huang, Bo, and Chen [37] (presented in Figure 2.2) shows that their method of mapping requirement to function domain is the use of human-machine interaction. After that, a combination of systems adds up to a DM where the concept generation itself is done by the usage of a genetic algorithm to make combinations from a morphology matrix.

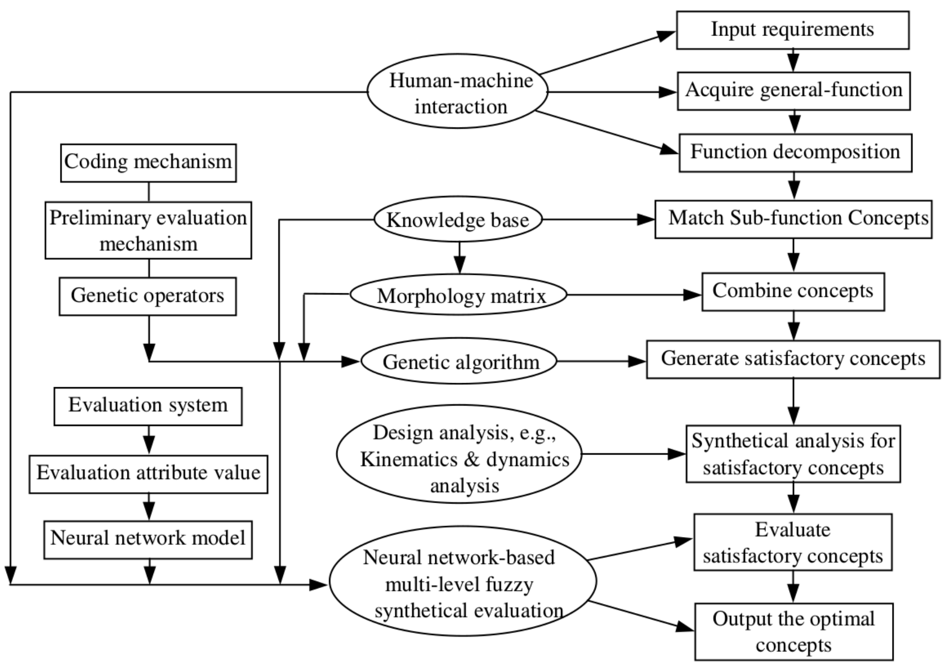


Figure 2.2. Process model for conceptual design based on computational intelligence [37].

This CAI-tool uses a morphology matrix, which is a known non-CAI DM. Just like for many of the other aforementioned DMs in the previous section, some sort of computerized version or CAI-tool exists that uses a comparable method. Every CAI-tool based on intuitive methods still relies on the user providing the solution itself and merely facilitates performing the DM on a computer. Looking at potential benefits [19], these CAI-tools have the potential of enhancing efficiency, effectiveness and/or competence. Creativity enhancing potential is only found in CAI-tools based on discursive methods because only with those CAI-tools solutions for concepts are generated by the CAI-tool itself.

Examples of these CAI-tools, which are also referred to in Table 2.1 on page 25, include:

- **Systematical study of physical processes**: Žavbi, Fain, and Rihtaršič [38] use knowledge twisting, the manipulation of physics, structure and design, to achieve functions. By chaining physical laws and complementary basic schemata, elementary concepts are generated. Another CAI-tool, SOPHY [39], uses a database of physical phenomena as well to create chains of these phenomena but also includes the possibility for the user to couple structural elements to these chains.

- **Design catalogues** were used very early to create CAI-tools [40] but also a more recent work by Brockmöller [41] describes a CAI-tool that creates detailed concepts for a specific manufacturing technology. Han and Lee [42] reuse previous design concepts in conceptual synthesis of mechanisms by representing them in a graph structure and synthesizing chains of these to create solutions. Kurtoglu et al. [43] make a database of dissected artifacts based on their functions which could be applied in different types of CAI-tools that benefit from the link between functional description and corresponding components.

- **Biomimicry** is the basis of the AskNature [44] database. It does not generate concepts but its website is a well known search-engine for biomimetric design solutions. Similar work is done by Chakrabarti et al. [45] by creating a functional representation for biomimetric design, allowing users to search a database for solutions. Goel et al. [46] developed *DANE*, short for Design by Analogy to Nature Engine. This tool helps users find relevant biological systems and aiding them in understanding their functioning. Integration of biologically inspired design with established function-based design methods is published by Nagel and Stone [47].

- **TRIZ** is used in various CAI-tools such as open CAI 2.0 [48] which allows users to collaborate on solutions but not generating solutions on its own, Goldfire [49] is a software package using unspecified artificial intelligence to process data from previous projects within a company to create a database which now includes 8000 physical, chemical and geometrical effects. Innovation Workbench [50] also provides a digitized version of the TRIZ method but software output is limited to generalized analogous solutions.

- **SIT** based software is mentioned by Yoo [51] and a framework for a CAI-tool named DSIT explorer (design-oriented systematic inventive thinking) is also published [52]

Other noteworthy CAI-tools that are not based on structured DMs do exist as well. In 1986, Dyer, Flowers, and Hodges [53] published about the EDISON project that intended to model processes of invention, analogy and naive mechanical device representation. No topological representations were used but merely linguistic representations of topology, connections, motion, and constraints thereof. The underlying reason for EDISON however, is to get insight in naive physical reasoning and invention of people. More recently, an algorithm that generates function solutions based on linguistic representations for functional elements was made [54], but evaluation of solutions is done by matching output and input of solutions to the functional requirement. Besides that, the linguistic representation method requires a database-like structure.
"Creative" synthesis of mechanisms which is based on matrix representation [55] [56] is possible by using knowledge that describes only mechanical systems. Although this is not necessarily a database of solutions or a systematical study of physical processes, it can only function with foreknowledge and for a specific range of problems in mechanisms. Another similar method, for functional synthesis of solutions in mechanical conceptual design, uses an abstract representation of design problems and functional elements. Subsequently rules for combination and transformation of input, output, and orientation are used to create solutions [57].

Usually, no evaluation of concepts is done in CAI using equations of physical laws but merely by using schematic representations of components. The amount of possibilities in the case of detailed shape representation and simulation of physical laws is enormous and could lead to problems in terms of computational power because of the risk of a combinatorial explosion. No proof of this exists however [58], and using smart shape representation [59] (see Section 2.3.1) or search methods such as GA [28] (see Section 2.3) can result in feasible CAI-tools.

Lipson [60] uses GA to find solutions for kinematic mechanisms that need to follow a straight line. Even though this has a narrow application and the foreknowledge of the method itself could be seen as an elementary design catalog, their conclusion is that is fair to compare the algorithms' performance with early engineers that had knowledge about basic linkage mechanisms.

To determine the potential of CAI-tools, the next section will compare these to previously mentioned DMs. This will help in clarifying unexplored areas of research.

2.1.3 Comparing DMs and CAI

To compare DMs and CAI-tools, a classification for DMs by Kulkarni et al. [61] is used. They identified the set of components that are used within DMs for concept generation that help designers overcome or *tackle* specific metal *blocks*, which can be seen in the Appendix in Table A.1. For various DMs as well as the DM belonging to the CAI-tool proposed by this research, the usage of these tackles is indicated in Table 2.1. For CAI-tools, the *solution source* is indicated: the user could be the source of ideas and the tool either taking another role to aid in this process or using a database or some other form of foreknowledge about constructing a solution. It is also indicated whether CAI-tools in that category are capable of generating an artifact as a solution.

		Intuitive						Discursive					
Block	**Tackle**	Brainstorming	Method 635	Gallery method	C-sketch	Synectics	Morphological analysis	Systmatic study of physical processes	Design catalogues	Biomimicry	TRIZ	SIT	This research
Premature judgment	Suspend judgment	•	-	•	•	•	-	•	•	-	•	•	■
Emphasis on quality	Emphasis on quantity	•	-	•	-	-	-	-	-	-	-	-	■
	Emphasis on variety	-	-	-	-	-	-	•	•	-	•	•	■
Lack of motivation: satisfied with present solution	Generate alternatives anyway for its own sake	-	-	•	-	-	-	•	•	-	-	-	■
Having a tight grip on problem specifications	Change the frame of reference (view data in a different way)	-	-	-	-	-	-	•	-	-	•	•	-
	Use of analogies and metaphors	(•)	-	-	-	•	-	-	•	•	•	•	-
Rigid problem representation (textual, mathematical)	Flexible problem representation (abstract, pictorial)	-	-	-	•	-	-	•	•	-	•	-	-
Design fixation (attached to one idea)	provocative stimuli	(•)	•	•	-	-	-	-	-	-	•	•	■
	Random connections	-	-	•	-	-	•	-	-	-	-	-	■
	Incubation	-	-	-	-	-	-	-	-	-	-	-	-
Imposing fictitious constraints	break rules	-	-	-	-	-	-	-	-	-	-	•	-
	Work on higher problem	-	-	-	-	•	-	•	-	•	•	-	-
	Solution source												
	Computer generates solution	-	-	-	-	-	-	■	(■)	-		-	■
	User generates solution	■	■	■	■	■	■/-	-	■	■	■	■	-
	Database	-	-	-	-	-	■/-	■	■	■	■	-	-
	CAI-tools	[62]	[63]	[64]	[65]	[66]	[67] [68] [69] [70]	[38][39]	[41]	[44] [45] [46] [47]	[48] [49] [50]	[51] [52]	

Table 2.1. Overview of DM indicating the *blocks* that are explicitly *tackled* by the various DM (•), based on Shah, Kulkarni, and Vargas-Hernandez [71] and Kulkarni et al. [61]. References to literature mentioning corresponding CAI-tools are shown at the bottom for each DM with indication of the solution source (■).

Comparing these components to their counterpart within CAI shows the potential of CAI in overcoming or avoiding these blocks in general but it can be seen as well that for intuitive methods, the user still is the sole provider for a solution. For discursive methods, apart from CAI-tools where the user provides a solution, it can be seen that a database is used or some foreknowledge about solutions is necessary to generate a solution. It may be that some specific method is used to search this knowledge (e.g. a GA) but the usage of foreknowledge or a database in itself is inextricably linked to discursive CAI-tools.

The proposal in this research to work without a database nor user provided solutions is not used in other research. The method itself seems to coincide with both intuitive and discursive methods as opposed to other CAI-tools, in which a solution is generated by the program itself which are all within the discursive category. This leads to the fundamental question about creativity: to what extent software is capable of being creative.

2.1.4 Creativity

Creativity is defined by the Oxford dictionary as "The use of imagination or original ideas to create something". One important aspect is omitted here however: not only does an idea need to be new, but for engineering it also needs to be useful to be creative [72]. Generally, it is seen

that computers cannot be "creative" but with the emergence of evolutionary algorithms (of which genetic algorithms are a part) this statement is started to be disputed [73].
On first sight, creative thinking appears to be non-continuous but in reality, this is caused by the intermediate steps in the continuous process that are hidden within the inexplicit mind [74]. While these intermediate steps are now impossible to define, still more can be said about processes in creativity that can be executed by computers. These processes are defined by Gero and Maher [75] as combination, transformation/mutation, analogy, emergence and first principles. Graphical examples of these processes can be seen in Figure 2.3, and all of these have been applied in CAI-tools [76].

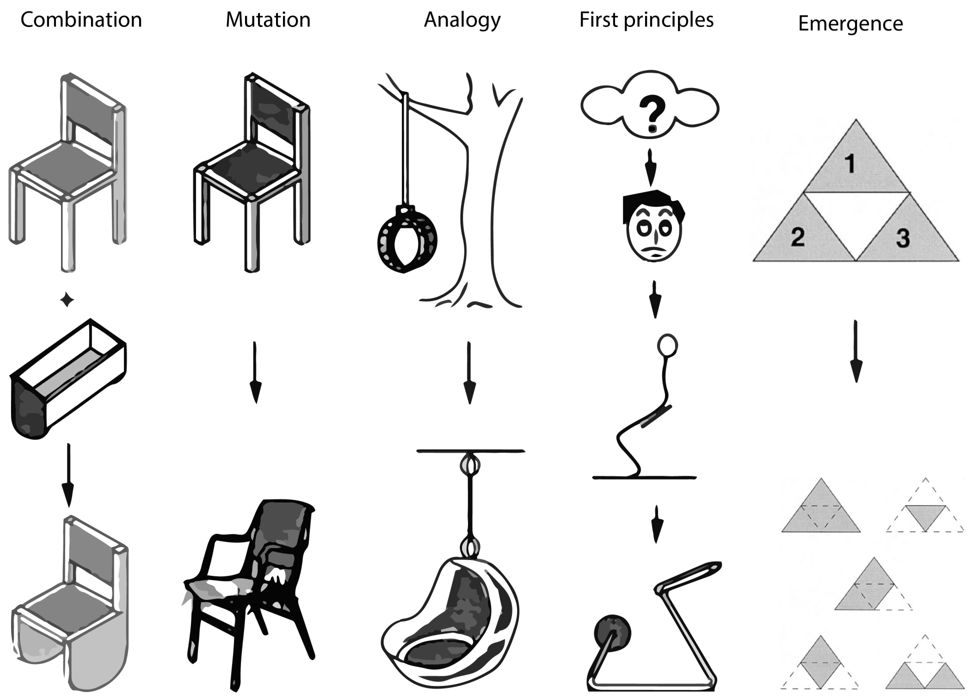


Figure 2.3. Processes in creativity [75] [76].

Whereas the first four processes are based on existing knowledge or a database, the use of first principles does not incorporate any shallow knowledge but deep knowledge: only propositions or assumptions that cannot be deduced from other propositions or assumptions are first principles (e.g. physical laws). However, compiled knowledge can be created from first principles [77]. The difference in these is that shallow knowledge is not based on underlying laws of behavior. This can only result in solving problems that are within the knowledge-base that was programmed, but deep knowledge has the ability to solve problems that are different from pre-defined solutions [78] as opposed to methods such as TRIZ which only lead to incremental innovation [79].

Using computerized evolutionary processes in creatively generating new things shows great potential [80]. Genetic algorithms can be used to extract useful design information by looking at different trade-off options when there are multiple conflicting objectives [81]. But even human-competitive results are produced in various situations: by genetic programming [82], when finding topology is part of the problem, when a highly non-linear solution space has to be evaluated, in the case that an approximate solution is sufficient and where there is a clear objective but it is hard for humans to create a solution.

New artifacts created with (computational) creativity can be put in two distinguished groups, P-creativity (psychological) and H-creativity (historical) [83]. Artifacts that are novel for the individual, or for the computer that produced it, are P-creative, and H-creative artifacts are

historically novel. The solution space where any of these artifacts exist is described by Maher and Fisher [84] as "not bounded before the process for producing the potentially creative artifact begins and can include an initially fixed state space representation, personal knowledge, historical knowledge, or the knowledge available to a network of humans and computers". This means that this solution space should not be bounded to exclude solutions or explicitly only including some solutions before starting the process.
For example Hiller and Lipson [85] see the limitation of topological optimization (mentioned in Section 1.1.3), or to be more specific: iterative homogenization techniques in optimization [86] to meet what they call "high-level functional goals" and explore the possibility of using evolutionary algorithms.

A framework for creative evolution is presented by Bentley and Corne [87] which contains five elements: an evolutionary algorithm, a genetic representation, an embryogeny using components, a phenotype representation and fitness function(s) and/or user input. Renner and Ekart [88] argue that the origin of creativity for computers could lay in the non-traditional application of evolutionary techniques and narrow down the earlier presented solution space for creative artifacts even more: no possible solutions are presented to the system, only tools for constructing the solution. This supports the foundation for this research in using evolutionary techniques in creating non-predefined solutions. Relating that to existing CAI-tools: it is seen that they all rely on user input (for intuitive methods) or some sort of foreknowledge or database (for discursive methods).
This section can be concluded by stating that using evolutionary techniques to search a solution space without foreknowledge is s kind of creativity which is not yet used in a CAI-tool and can be further explored in this research. This is based on the observation that, when dividing DMs into discursive and intuitive methods and linking these with corresponding CAI-tools, CAI-tools linked to intuitive methods do not generate solutions yet but rely on the user for that task. Creativity in computers can be seen by using evolutionary techniques and this could be applied to make new CAI-tools.

2.2 Analysis of novelty, working principles, and mechanical functions

Solutions that are created by CAI-tools need to be analyzed and suitable problems need to be set up to be able to evaluate the CAI-tool in this research. The newness or novelty of solutions, relative to some specific frame of reference should be capable of being analyzed, which in turn demands descriptions of working principles and functions. The final goal is to be able to validate the proof of concept of the CAI-tool in this research.

2.2.1 Measuring novelty and variety for CAI-tools

Analyzing new products, in general, is often done in relation to previous work. Because for instance TRIZ is based on analysis of patents it has a clear method to categorize innovations which will be taken as example. Five innovation levels are distinguished [89].

1. Only quantitative change of parameters. This is possible to do using most optimization algorithms.

2. Qualitative changes and improvements of components or configuration based on the existing function. This is possible in some cases, e.g. using topological optimization.

3. Extending a known function-principle combination to a new market, e.g. function *to displace* is achieved on basis of thermal expansion and can be used for exact positioning of a table in a microscope and a magnetic head in a tape recorder [90]. This is not yet possible using any CAI-tool without using a database.

4. Radically new function-principle combination. For instance: the first application of electromagnetic waves in a radio transmitter.

5. New principle. E.g. the discovery of X-rays.

These levels are usually only used for rating within the context of historical novelty such as patents and are not applicable to measure the effectiveness of CAI-tools. This is because for a new artifact to be an innovation, it needs to be introduced into the specific context which was the point of reference to the prior status [8]. Still, even without quantifying abilities, these levels could be used to describe a general level of innovativeness that can be reached by a CAI-tool by changing the frame of reference. This means that the definition of *new* will need to be different in this research because the previously stated innovation levels relate to already existing patents and solutions. The fact that the proposed CAI-tool will work without any database of existing solutions implies any functioning solution to be new. However, certain laws of physics will be an input in order to be able to simulate artifacts, which immediately rules out the possibility of reaching the innovation level 5 described above.

A method to measure the ability of CAI-tools to expand design space and their effectiveness in design space exploration using two criteria is proposed by Shah, Smith, and Vargas-Hernandez [91]. Quantity, variety, novelty, and quality metrics are used for this, as can be seen in the overview in Figure 2.4 Only a small portion of creative characteristics is quantifiable when ordering ideas in a tree structure.

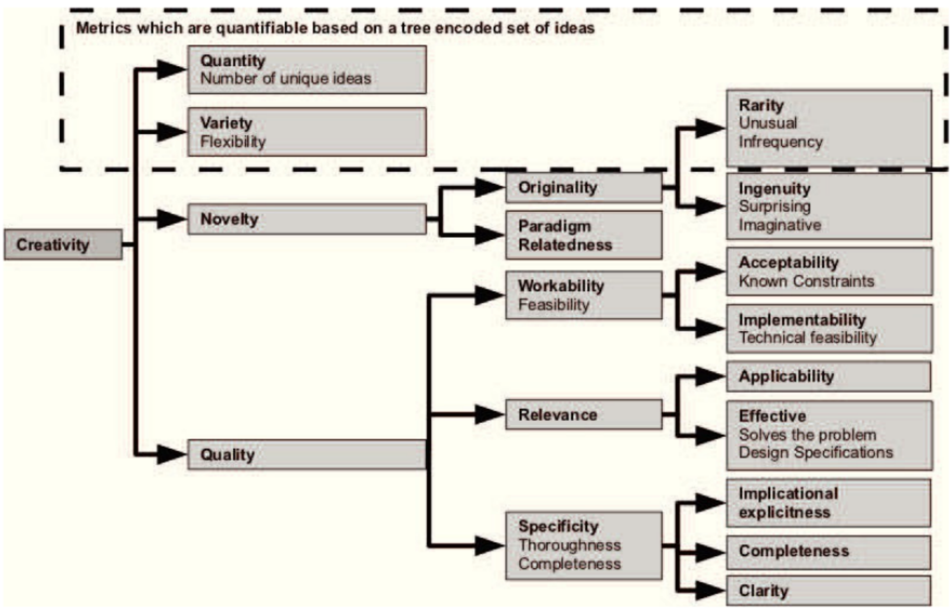


Figure 2.4. Overview of ideation effectiveness metrics [92]

Another rating, created to measure variety between a number of concepts in relation to each other instead of historically existing solutions, is proposed by Shah, Kulkarni, and Vargas-Hernandez [71], from highest variety to lowest:

- Ideas use different physical principle.
- Same physical principle, different working principle.
- Same working principle, different embodiment.
- Same embodiment, different detail.

As an example, the concept for a motor could be solved with the *physical principle electromagnetism* and use “coils for attracting and repelling permanent magnets” as *working principle* while the description of the location of the coils and permanent magnets is done at the *embodiment level* and finally the details in the technical drawing, manufacturing, and assembly are described at the *detail level* [92].

2.2.2 Functional analysis

A mechanical engineering function describes the teleology of the object, the intent [93]: what operation should a device or artifact do or what effect should it have on its environment [94] [6]. Especially some discursive DMs benefit from a clear abstract description of functions and their design solutions. The National Institute of Standards and Technology (NIST) has developed a formal functional representation based on previous research [1]. Three levels of flows and functions specification are defined; from a class to a tertiary level.
At the class level the EMS (energy, material, signals) model is used in representing flows, based on Pahl et al. [5], just as their classification for functions is used. Full tables for these flows and functions can be found in Appendix A.3, in Table A.3 and Table A.2, respectively. In Section 3.1, more about the EMS model will be discussed as well.

As an example: gears transmit a rotational velocity from one axle to the other. This would be placed in the class *channel* (moving a flow from one location to another), on the second level in *transfer* (shifting or conveying flow from one place to another) and as a third level *transmit* (moving energy from one place to another).

These descriptions, however, do not provide any insight into how the function is fulfilled. Cheng and Ma [95] propose to define a function as the combination of design intent, EMS flow and behaviour. They also introduce functional feature modeling, integrating function, and structure. These, however, are intended to integrate functional design with procedural feature operations in CAD and are not applicable to analyzing existing artifacts. Therefore it is necessary to find methods to describe working principles and behavior of artifacts from another perspective.

2.2.3 Behaviour and working principles

Behaviour is defined as “the way in which a machine (or natural phenomenon) works or functions” by the Oxford dictionary. Describing behavior will bring more detail to the way a function is fulfilled by an artifact. This level of detail is represented better by defining behavior as sequential changes of states of a physical structure or element of a system, over time, in reaction to its surroundings. The underlying causes of behaviors are changes in the different modes of a physical system or entity called *states* [96]. For example, the behavior of a fan depends on the torque generated by the vacuum motor, *rotation* itself is a physical phenomenon that depends on the state transition of torque.
Having a more direct look at the physical meaning of behavior in relation to working principles gives the opportunity to have a more analytical description of functionality.

Integration of function and form is done in the *Contact and Channel Method* (C&CM) which considers the function dependent on the form [97] and is used to describe design problems and artifacts in a “systematized but free and dynamic way of modelling”, in order to provide an abstract representation of the artifact[4]. In this case, *form* is describing geometry as well as other properties that are relevant in fulfilling its function. Describing geometry is done using these elements:

- **Working Surface:** interacting with its environment via input or output flows. It can interface with a WS on other parts of the system with its *outer* port and connects to an adjacent CSS within the same part via an *inner* port.
- **Working Surface Pairs (WSP):** the connection between the *outer* ports of two WS. This is where two objects interact with one or multiple flows.
- **Channel and Support Structures (CSS):** forms the internal structure of a part, by connecting two *inner* ports of WSs and can store energy (e.g. kinetic or potential energy).
- **Connector:** integrates the system into its environment. A connector describes everything that links to a WS at the system boundary [3].
- **Module:** a subdivision which contains more modules or parts.
- **Part:** performs functions by interacting with other system elements through WSs and/or CSSs. This is the lowest subdivision that is possible in the C&CM approach.

A graphical representation of these can be seen in Figure 2.5.

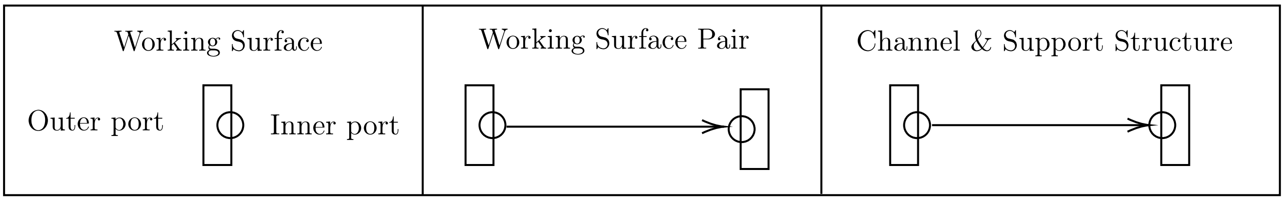


Figure 2.5. Graphical representations of elements used in the C&CM approach. [97]

The flows described in Section 2.2.2 (e.g. mechanical, thermal, electrical) are used to describe the particular interaction of subsystems or parts by associating them with each WSP [98]. It must be noted that some special cases such as gravity and magnetic fields, which would need *working volumes*, are excluded in this method for now but have been in development [99]. Three rules hypothesized by Matthiesen and Ruckpaul [100] are used to ensure consistent application:

1. Every technical system fulfills its function by interaction with adjacent systems. This is only the case when two WSs are in contact with each other so a WSP exists.
2. Functions are represented by at least two WSPs, a CSS which connects those two and at least two connectors to the environment.
3. Every (sub)system can be described by using WSP, CSS, and connectors, on different levels of abstraction and detail.

By using this description, a particular solution for a function can be analyzed in a concrete way. As an example, the analysis of the functioning of a ballpoint pen on two different levels of detail is shown in Figure 2.6. The functioning of the pen can only be fulfilled if $WSP_{0.1}$ (between pen and paper), $WSP_{0.2}$ (between hand and pen) and $CSS_{0.1/0.2}$ (the body of the pen) exist [4].

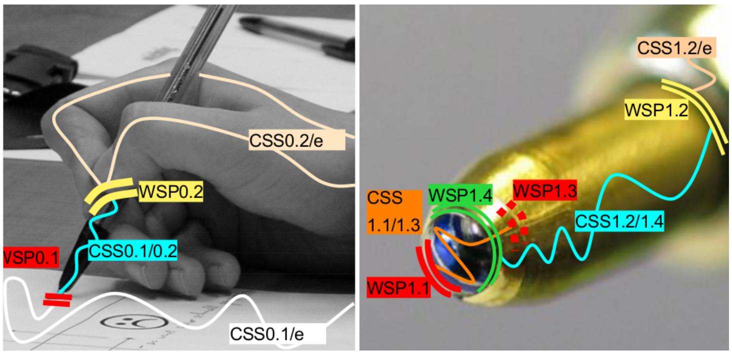


Figure 2.6. C&CM description of a ballpoint pen (right image is more detailed) [4].

In addition to interfaces (similar to WSPs and CSSs), Aifaoui, Deneux, and Soenen [101] also use constraints in space or mobility to describe their "technological model". This even allows for a further specific description of an artifact. Relating this to the example in Figure 2.6: WSP1.4, describing the connection (within the pen) between the ball and its holder, can be extended by a description of the constraints holding the ball in place.

Other models of describing artifacts on an even more analytical level include: parameter, field, space, shape and topology description of artifacts [102]. An example of another descriptive method, tailored for mechanical structures can be found in Appendix A.4.
However, these descriptions are not based on a generalized functional description for working principles so a direct comparison of two solutions using such a description is not possible and thus will not be used. For example: assessing the difference in working principle of a ballpoint pen to a fountain pen is not possible by solely comparing widths, diameters, shapes, positions or other parameters used in defining the designs.

In this section, a method to rate variety between concepts was presented. Also, in addition to functional analysis, the analysis of working principles of artifacts have been discussed as a way of analyzing and subsequently distinguishing solutions.

2.3 Genetic algorithms for soft robots and neural networks

In Section 1.1.3 and 2.1.2, already some examples of GA usage in optimization, design and CAI-tools were included. Instead of using GA to search within combinations of database elements or higher level function representation, this section introduces pattern producing networks as a means of shape representation for GA. This is used, among other things, to generate soft robots. This method will also be used for the proof of concept of the CAI-tool proposed in this research.

2.3.1 Shape representation for GA

The way in which individuals are represented is important in evolutionary computation, especially when the solution space could be large, since this influences which solutions are possible, the way that individuals are changed and the memory size occupied by an individual. Representation, in this case, is the mapping of the genotype to the phenotype. The genotype is the lowest-level of encoding used in the algorithm, the phenotype does not have such a clear definition: any higher-level expression of the genotype can be defined as the phenotype [7]. The simplest genotype is bit-string representation: after discretization of the shape, every discrete element is represented in the genotype. Kane and Schoenauer [103] conclude however, that using algorithmic geometry in GA instead of bitstring representation leads to promising results.

Some representations were already treated in Section 2.1.2 but none of these used a representation without foreknowledge about the problem.

Other similarities can be found in research by Chen and Xie [54] where optimization is used to generate concepts. Or the work by Chen et al. [68], which has a design catalogue model to represent mechanisms. However, the solution space is then also made up of a database of partial solutions and linguistic representations of mechanical objects. This restricts the solution space drastically. Any foreknowledge or other restrictions that are used to generate results will bring the optimization back a step towards existing solutions and diminish potentially creative solutions as discussed in Section 2.1.4. An essential element to prevent this is to also allow shapes to be generated that are not known as solutions to that mechanical engineering problem.
Using basic shapes, that fulfill no function on their own, to represent solutions avoids the introduction of foreknowledge in a CAI-tool. Using super shapes, for instance, is done by Preen and Bull [104] who use a genetic algorithm known as the Gielis superformula to optimize topological representations of vertical-axis wind turbines. This is still specified only for the wind turbine problem, a more general approach needs to be found. Gero [105] presented an early computational model of evolution-based creative design mentioning shape grammars that express artifacts, but no concrete implementation is presented. More recent work, with an implementation, is done by Ebner [106]: the proposed method includes the usage of scene graphs, which are also applied in computer graphics. Scene graphs consist of various nodes (for e.g. shape, transformation, and property) to store an object or entire scene. No knowledge about the functioning of the object is introduced by scene graphs. However, more complex shapes can be represented without defining every detail (points, vertices, faces etcetera) of the shape. An example of a scene graph and the object it represents can be seen in Figure 2.7.

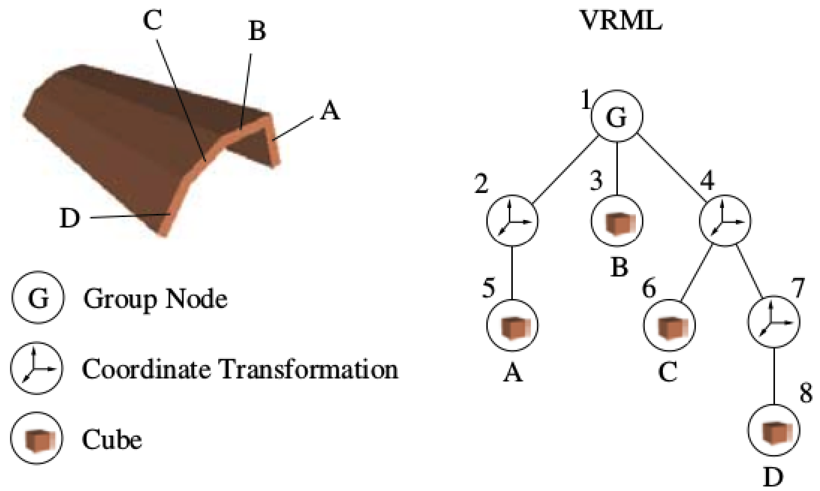


Figure 2.7. One object represented by a VRML scene graph [106].

Even though the scene graph approach has a limited set of basic shapes, it still relies on only this set to represent shapes. This leads to a very specific niche of shapes that are created when this method is used in optimization. Using pattern producing networks, however, will allow the creation of entire shapes without relying on any basic shapes. These will be introduced further on in this section.

Artificial neural networks

Before reviewing CPPNs (compositional pattern producing networks) and NEAT (neuroevolution of augmenting topologies), a basic understanding of NNs (neural networks) or ANNs (artificial neural networks) is needed. An ANN is a network of artificial "neurons" that takes one or multiple real-value or binary input signal and transforms this to an output between fixed minimum and maximum limits, usually zero and one [107]. Neurons are connected by "edges" to form a connection between the main input and output nodes via hidden layers of these neurons. Every connection can have a weighting factor that scales the signal. In Figure 2.8, an example of an ANN with two input nodes, a single hidden layer, and one output node is shown.

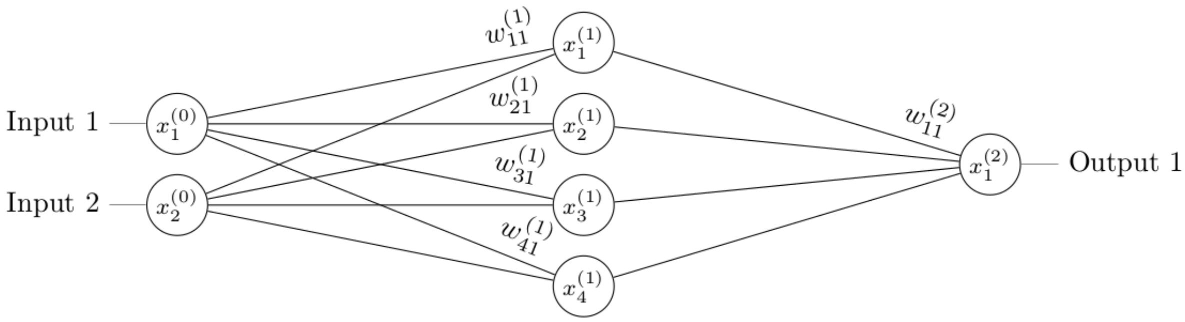


Figure 2.8. An ANN consisting of two input nodes, a hidden layer of four nodes and a single output node [108]. (Weights are indicated with a w.)

ANNs have many applications in processing complex data. To achieve a desirable result, a sample data-set is used to train an ANN: weightings are optimized so the output presents desired results and it can be applied to real data [107]. Examples include analyzing radiographic images to determine patient treatment [109] and speech recognition [110].

NEAT

Instead, NEAT (NeuroEvolution of Augmenting Topologies) optimizes networks by using a GA [111]. Not only weighting parameters are optimized but the topology of the network is changed as well. A small network is started with, which is then complexified by adding more nodes and connections to it. Figure 2.9 depicts the encoding for the genotype that is used in NEAT: node and connection genes with corresponding properties are used as a basis. An "innovation number" is added to each gene to show the moment of its creation, so corresponding genes of two future individuals can be easily matched for mating when creating a new individual. The creation of offspring is done by combining the genotypes of the two parent individuals, this can be seen in Figure 2.10. Genes from both parents are used in the new individual when they occur exactly the same in both parents or in just one. In the case that two different genes are encountered at the same location, the gene of the individual with the highest fitness is adapted.

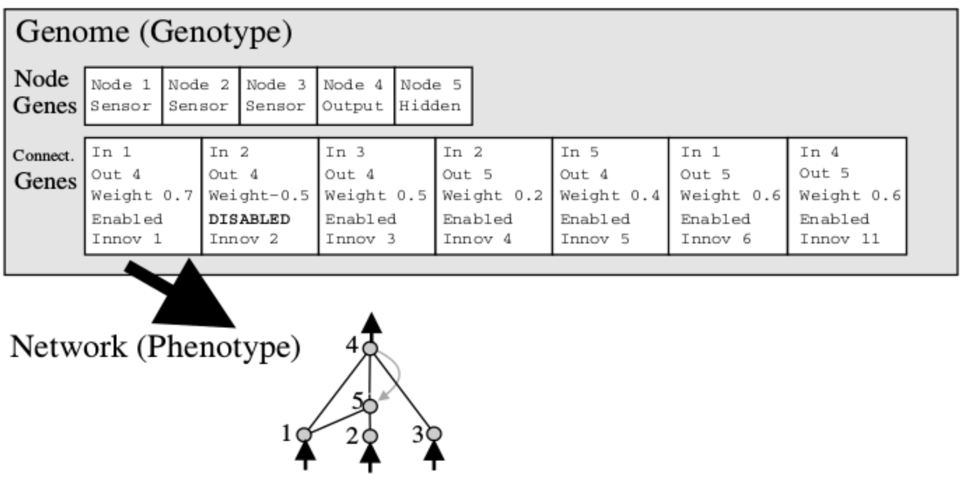


Figure 2.9. The genotype that is used in NEAT [111].

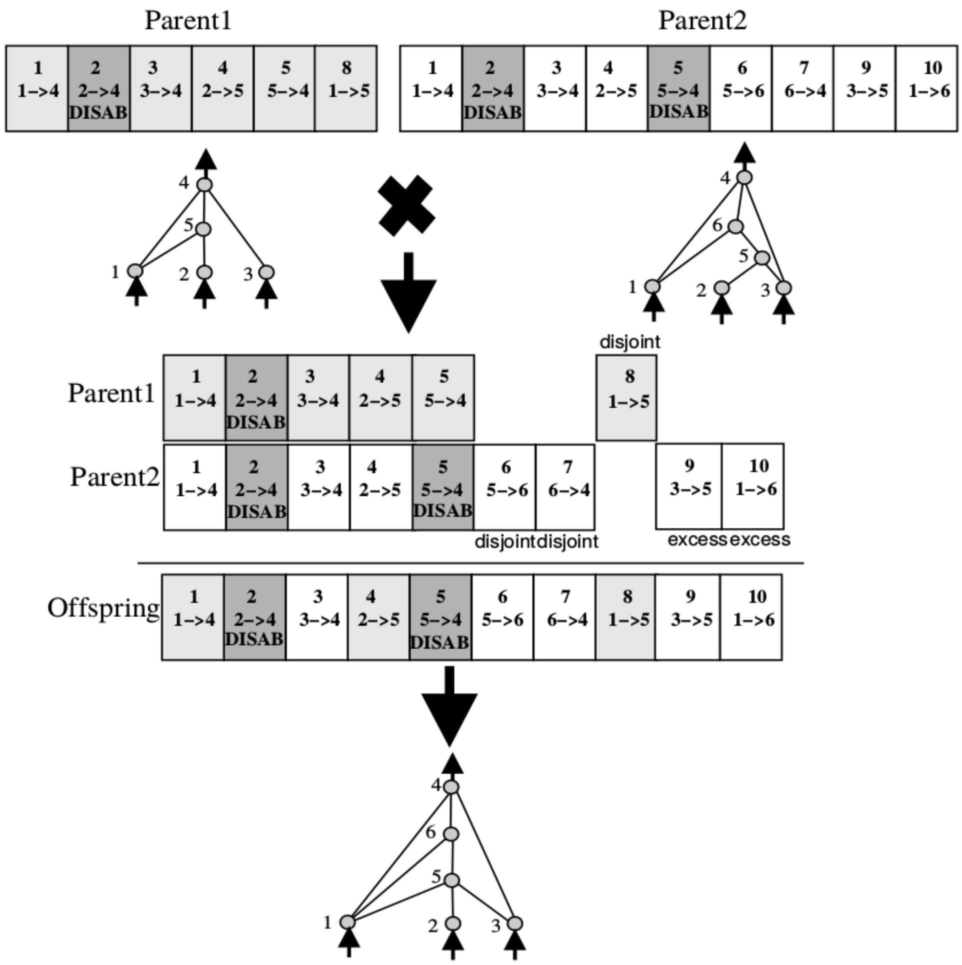


Figure 2.10. Creation of new offspring from two individuals in NEAT [111].

Figure 2.11 shows the two possibilities for mutation within NEAT. Either a connection with a random weight is added or a new node is added with input weight 1 and an output weight of the old connection.

Through these specific elements, NEAT is able to optimize and complexify solutions at the same time. This allows for complex solutions to emerge. The next step is to apply NEAT in the creation of artifacts.

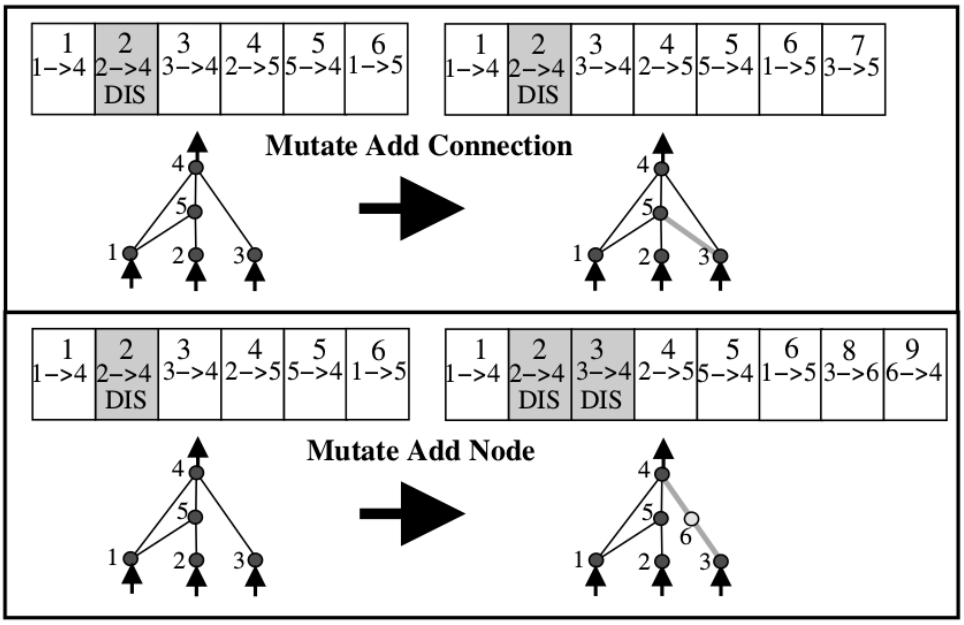


Figure 2.11. Two options for mutation of the genotype for NEAT [111].

Representing shapes in GA can be done using direct encoding where every gene represents a single entity in a shape or, in the case of NEATs, where the output nodes of a NN represent a single entity in a shape. This, however, results in a large genotype which is harder to compute. Hiller and Lipson [112] mention some of the problems of direct encoding or representing every voxel in the genotype as is done by [113] for example that it results in many adjustable parameters and the great likelihood of destroying solutions at mutation or crossover.

CPPN-NEAT

A more adequate shape representation method for this research than the ones mentioned in Section 2.3.1, is the use of compositional pattern producing networks (CPPNs). These can be used to generate structures in 2D or 3D using networks that are very similar to ANNs [114]. As opposed to ANNs, where only sigmoid or only Gaussian functions are used, the functions in the nodes can include many types of functions in one CPPN, e.g. linear, sigmoid and also Gaussian functions. A graphical representation can be seen in figure 2.12.

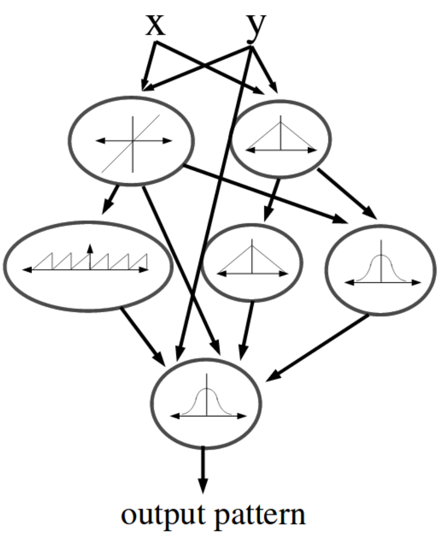


Figure 2.12. Example of a CPPN for a 2-dimensional output image [114].

Besides this difference in usage of functions, CPPNs are different from ANNs because the full range of possible inputs is swept to create the output structure in 2D or 3D instead of the mapping of a specific data-set from input to output.
It is possible to use NEAT to evolve CPPNs: to evolve complex topologies that are described by a neural network that evolves using GA. These CPPNs are used in different applications such as Picbreeder [115]. Users assign fitness to individuals by selecting images that are the most appealing. This fitness is used in evolving the networks to the point that they generate the desired output. An example of images generated by Picbreeder can be seen in figure 2.13.

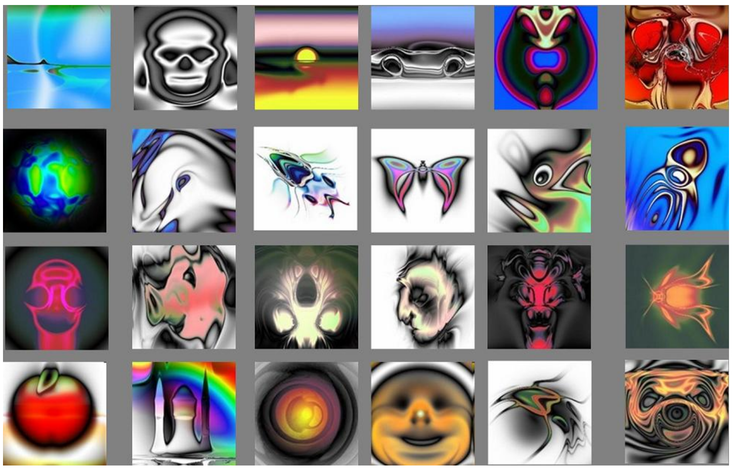


Figure 2.13. Using CPPN-NEAT, Picbreeder allows users to select images and evolve them to recognizable shapes [115].

Using CPPN-NEAT, not only 2D pictures can be evolved, but 3D shapes as well. One application of CPPN-NEAT in evolving 3D shapes is the creation of soft robots: soft (virtual) creatures that fulfill a simple task. Soft robots are made of low-stiffness materials that resemble biological tissue, therefore being able to adapt shape and locomotion to the situation: the environment, obstacles or tasks [116].

Groundbreaking work on virtual creatures was done by Sims [117] who showed that it is possible to evolve topologies and neural systems that represent rigid creatures in a virtual environment.

Based on this, as already mentioned in the introduction, Cheney et al. [2] created soft creatures with complex shapes made up from voxels, rather than creatures made up from a combination of rigid shapes. Movement is induced by the varying temperature of surroundings which results in thermal expansion or contraction of different materials. The CPPN-NEAT algorithm is used to optimize these soft robots. Each individual is represented by a CPPN, this is converted to a voxel structure by evaluating the CPPN on each discrete possible voxel location (the total size is determined by the user) to determine presence and material properties of voxels. This voxel structure is now evaluated in the multi-material voxel simulation software Voxelyze [118] to determine fitness parameters that are fed back into the CPPN-NEAT algorithm.

This method resulted in virtual creatures capable of different tasks such as walking or growing towards a light that are possible to 3D-print. It turns out that soft robots are ideal objects to evolve using genetic algorithms because their topology is easy to represent and simulate. More recent work was done on the influence of the environmental transition from land to water on evolving soft robots [119]

Using the hypothesis for a technical system from Section 2.2.3, it can be seen that soft robots, as they are generated by Cheney et al., do not satisfy the rule that the function is fulfilled by interacting with adjacent systems: just one connector can be defined for the soft robots that are produced, the input of a changing temperature. This is closely related to the difference to the definition of a mechanical engineering function (see Section 2.2.2) that an artifact performing a mechanical engineering function performs an action on its environment. This is not done by the soft robots either.
Another difference with this research is that the goal or objective function for these robots is already fulfilled in the first generation of the simulation in a lot of individuals. This in contrast with artifacts that need to fulfill more complex tasks that have a hard threshold in their functional fulfillment such as a continuous transmission. These two differences lead to the conclusion that there is a sufficient reason to explore further possibilities based on the research of Cheney et al.

2.3.2 Promoting diversity

One of the problems in optimization is convergence: ultimately, the population of a GA will converge to a single solution [120] but it is necessary to ensure that no premature convergence to some local optimum happens. This is avoided by promoting a diverse population, which is also preferable in concept generation: to find a variety of solutions.

Niching methods

One group of methods to preserve diversity are so-called niching methods. By keeping individuals far away from each other in the solution space, diversity is preserved. Various methods of niching exist such as crowding, sharing, and sequential niching.[121].

- Crowding prevents the existence of a highly similar individual. A simple method replaces old individuals of a random subset of the population whose size is defined as the crowding factor (CF) that have the highest similarity in genotype to the new individual [122]. Restricted tournament selection is another crowding type method in which two individuals are selected for recombination after which a random sample of CF individuals is taken where the just created individual competes with the closest individual. This procedure is repeated $N/2$ times (where N is the number of individuals) [123].

- Fitness sharing lowers the reward for individuals that are very similar. It influences the search space for crowded areas in terms of either phenotype or genotype. It seems that using phenotype comparison leads to more diverse results and costs fewer computation-time [124].

- Sequential niching is done by lowering the reward in areas where a local optimum is found so the search expands into other areas of the solution space [125][88].
- Population-based simulated annealing is based on simulated annealing which is taken from the process of slowly cooling metals: a starting solution is calculated and a virtual temperature determines the chance of moving to a newly generated solution. High temperatures enable moving easily while a low temperature disables large jumps. Population-based simulated annealing [126] uses multiple solutions at the same time to minimize the influence of the starting position or jumps to unfavorable solutions.

Novelty search

Instead of only using a fitness function that is related to the performance of an individual, promoting novelty as well as incorporating local competition allows more difficult morphologies to evolve [73]. Novelty search can lead to solutions with a high fitness value that otherwise would have gone undiscovered because their ancestors had low fitness values. This can be achieved by looking for novel solutions instead of high fitness. To quantify novelty, difference in properties (e.g. behavior) or in individuals are described [127] [128].

This section can be concluded by stating that using a GA has potential in creating technical systems with different multiple working principles for a mechanical engineering function when a suitable representation is chosen. By evolving a neural network that is capable of expressing a 3D-shape, a low dependency between representation and function is observed. CPPN-NEAT is suitable for this purpose and has been used in the generation of soft robots which show relevant results but do not comply with the demands for a technical system.

2.4 Conclusion

When CAI-tools are linked to corresponding DMs it becomes clear that no intuitive CAI-tools exist that do not rely on user-generated solutions. CAI-tools that do generate solutions, use introduced knowledge about working principles such as a database or a function specific representation. Using evolutionary techniques, however, creativity can be observed in computers. Measuring the output produced by a creative CAI-tool is important to validate the proposed method. In addition to functional analysis, methods to describe working principles and variety between concepts were presented. To avoid the introduction of knowledge on working principles into the CAI-tool, CPPN-NEAT can be used. A GA is used to evolve NNs that produce 3D shapes. Even though the work of Cheney et al. [2] is in the direction of the result that is sought after in this research, the approach and perspective are different. While their focus lies on soft robots and has foundations in the biological field, the approach in this research is from an engineering design viewpoint that results in mechanical engineering functions, performing actions on its environment possibly with multiple parts involved. Another difference is the threshold in functionality which is seen in most mechanical engineering solutions. This leads to different requirements for the outcomes that were generated by CPPNs evolved earlier with the NEAT algorithm.

3 Method

A method for the CAI-tool and its proof of concept (PoC) and also a method for validation will be introduced which is based on quantifying the variety of different concepts based on their working principles.

3.1 Quantifying variety in working principles

Shah, Kulkarni, and Vargas-Hernandez' method [71] to rate variety between concepts is extended using C&CM. Their scoring is follows:

Score	Description of idea
10	Ideas use different physical principle.
7	Same physical principle, different working principle.
4	Same working principle, different embodiment.
1	Same embodiment, different detail.

Just the classification of this rating is used to make the distinction between different artifacts and their variety.

The physical principle used by a solution to fulfill a mechanical engineering function can be determined unambiguously (using Table 3.1 or A.3). However, distinguishing different working principles without a method leaves room for personal interpretation. Therefore it is proposed to use the C&CM to identify differences in working principle between two solutions. Across WSs, energy transfers are described in terms of flows as done by Albers and Schyr [129], of which examples can be seen in Table 3.1. In addition to these descriptions of energy transfers, a graph is made to show parts, connectors, WSPs, CSSs and connections between those. *Flow* is used as a collective term for any energy, material or signal related term that crosses WSs such as an effort or momentum. Not to be confused with some specific flows (e.g. linear velocity) as mentioned in Table 3.1. A flow as is meant in the C&CM model, unless stated otherwise, can represent any expression of the domain it represents (effort, flow, momentum, etc.). The C&CM does not describe time-dependent properties well, therefore a dashed line is introduced to indicate non-continuous flows (see Figure 3.2). Connections that have a negative impact on the total flow are indicated with an arrow that points from the input WS to the output WS of another CSS.

To identify differences in working principle, the energy transfers and the associated graph are compared. A different working principle is indicated by one or more of the following differences: the number of parts, graph structure, graph elements, or difference in corresponding WSs or CSSs in domain, displacement, effort or flow aspect. When two C&CM graphs show a different number of replicates of a WSP, this is regarded as a difference in embodiment between the two artifacts the graphs describe. Remaining differences, in the case that physical principle, working

2nd class	3rd class	Power conjugate complements Effort analogy	Unit	Flow analogy	Unit	Displacement	Unit	Momentum/ Carrier	Unit
Electrical		Electromotive force	V	Current	A	Charge	C	Flux linkage	λ
Hydraulic		Pressure	Pa	Volumetric flow	m^3/s	Volume	m^3	Fluid momentum	Γ
Magnetic		Magnetomotive force	A	Magnetic flux rate	V/s	Magnetic flux	V	-	-
Mechanical	Rotational	Torque	$N \cdot m$	Angular velocity	rad/s	Rotation	rad	Angular momentum	η
Mechanical	Translational	Force	N	Linear velocity	m/s	Displacement	m	Linear momentum	p
Thermal		Temperature difference	K	Entropy flow rate	S/s	Entropy	S	-	-

Table 3.1. Some of the flow domains with associated effort, flow, displacement and carrier quantities as used in the EMS as well as the C&CM method [1] [129].

principle, and embodiment are the same, are defined as details. These details could still lead to a difference in the performance of an artifact.
For clarification, an example will be analyzed. Take the mechanical engineering problem which should solve the function *transmitting* for the physical principle *rotational, mechanical, energy* from location A to B, in opposite direction. Solutions for this problem include for instance a gear transmission, but also a crossed belt transmission. These are shown in Figure 3.1.

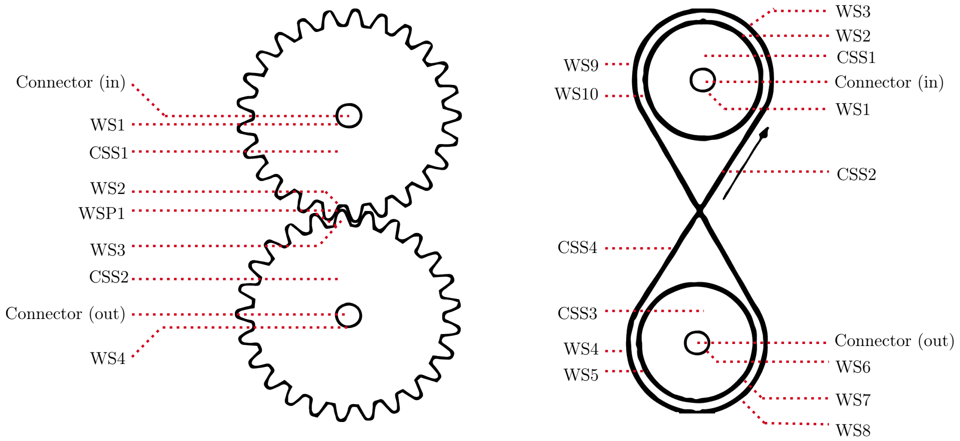


Figure 3.1. Two solutions for the function *transmitting* for the physical principle *rotational, mechanical, energy* with different working principles: gear transmission (left) and a crossed belt transmission (right).

Figures 3.3 and 3.4 with complementary Tables 3.2 and 3.3 show the working principles as analyzed using the C&CM for both solutions. Figure 3.2 provides a legend of the used symbols.

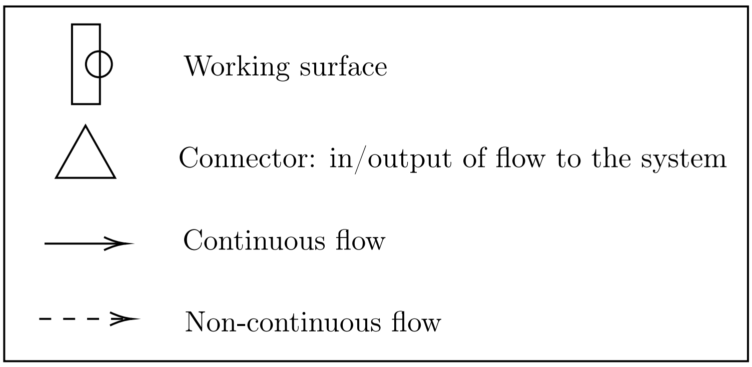


Figure 3.2. Legend for the C&CM graphs in this research.

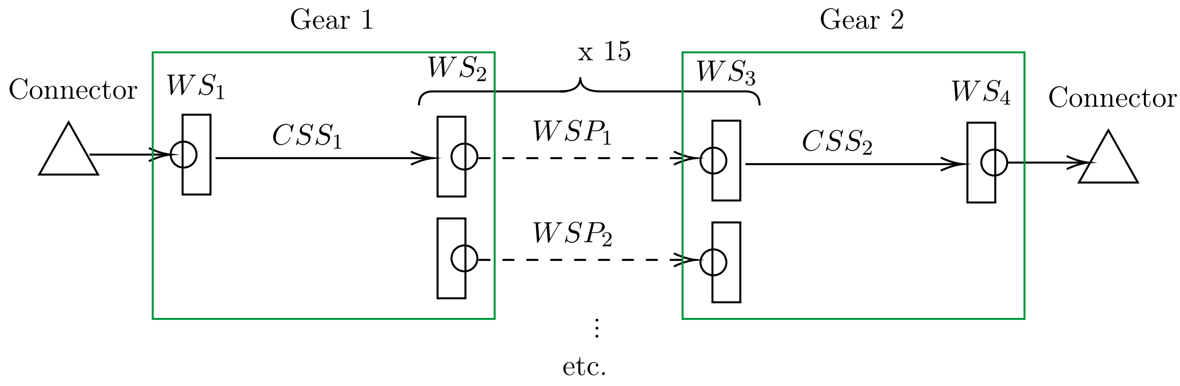


Figure 3.3. C&CM graph of the gears in Figure 3.1.

Working Surface	**Flow domain**	**Displacement**	**Effort**	**Carrier**
$WS_{1,4}$	Rotational 1D	Angle	Torque	Angular momentum
$WS_{2,3}$	Translational 1D	Position	Normal force	Linear momentum

Table 3.2. Description of WSs for the gear solution seen in Figure 3.3.

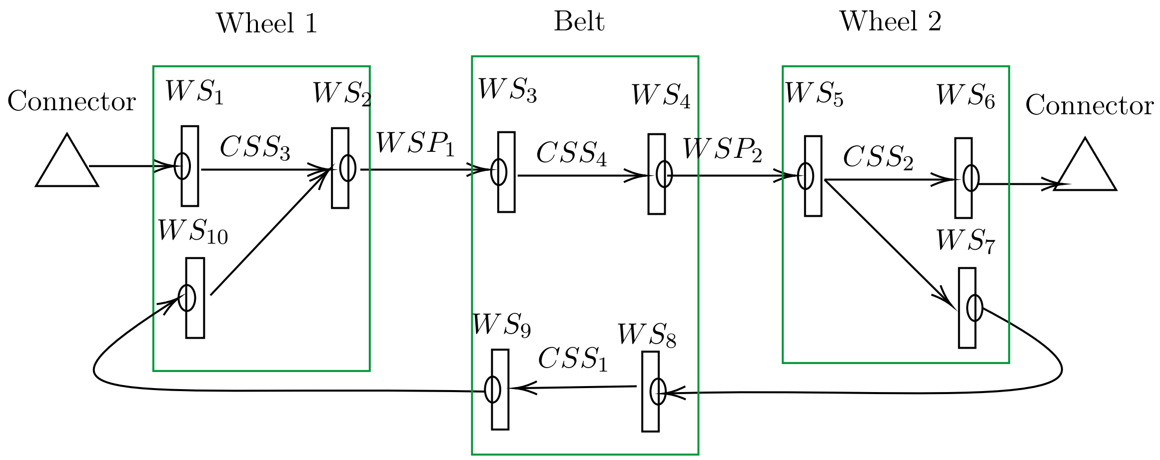


Figure 3.4. C&CM graph of the crossed belt transmission in Figure 3.1.

Working Surface	**Flow domain**	**Displacement**	**Effort**	**Carrier**
$WS_{1,6}$	Rotational 1D	Angle	Torque	Angular momentum
$WS_{2-5,7-10}$	Translational 1D	Position	Friction force	Linear momentum

Table 3.3. Description of WSs for the crossed belt solution seen in Figure 3.4

From the combination of the scheme and table, it can be seen that these two solutions have a different graph: the number of parts and working surfaces connect different flows. This leads to the conclusion that there is a difference in working principle.

The presented method will be used to manually identify the variety of concepts, generated by the CAI-tool as validation of the proof of concept.

3.2 CAI-tool

The proposed CAI-tool will be able to generate multiple solutions for a mechanical engineering problem without foreknowledge of working principles. An overview of this method for a CAI-tool is presented which will be validated by the proof of concept in Section 3.3.

The CAI-tool is based on the optimization of 3D-artifacts of which the fitness is determined by an external simulation. User input related to functionality is restricted to these elements: the function, the input flow, the output flow (all according to Hirtz et al. [1]) and for both input and output flow, specific physical properties that are necessary to define the flow for the specific problem (e.g. defined as effort or flow, location, orientation). See Figure 3.5 for a schematic view. The addition of physical properties to flows, used in the functional basis by Hirtz et al. [1], is necessary to describe the problem specific properties such as the location where a translation is applied, the input area through which some fluid enters or a direction on an applied force. These user input elements are not directly used as input for the optimization but placed in the simulation environment so that the creation of solutions by the optimization algorithm is not influenced by knowledge about the problem. The relation between input and output of the mechanical engineering problem is described by the function. The objective function describes how well a solution is able to convert input to output flow according to the prescribed function. In the simulation, the in- and output flow are measured by performing a simulation on each solution. This measurement is provided to the optimization algorithm that records the fitness for that particular individual.
The simulation can calculate the objective function by simulating the performance of each individual. This is done by using only deep knowledge such as physical laws (see Section 2.1.4): no knowledge on working principles is provided in any way, only laws of physics related to flows in the mechanical engineering problem and available materials along with a shape representation.

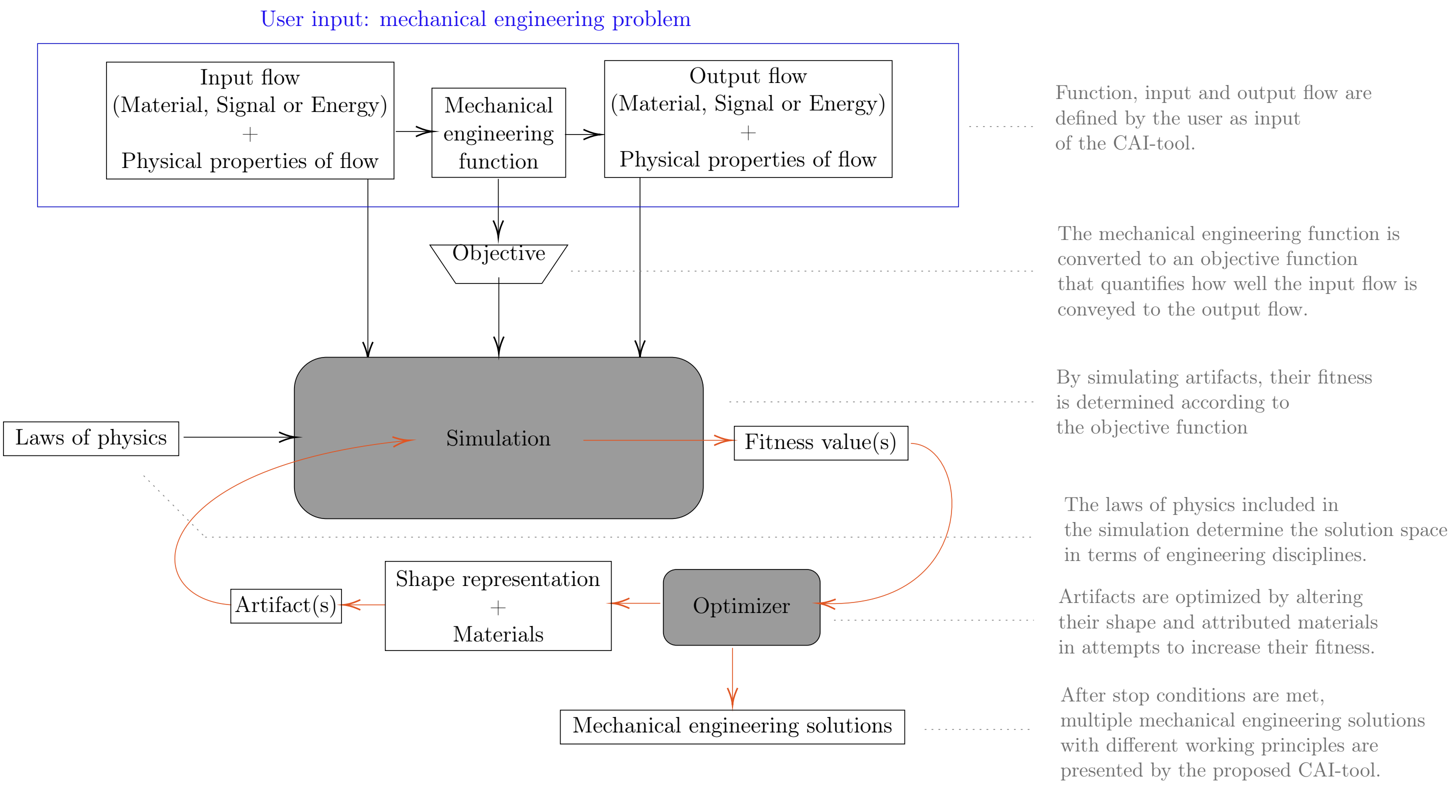


Figure 3.5. Schematic view of the proposed CAI-tool. The blue box indicates the user input, based on the functional basis by Hirtz et al. [1], other elements are part of the CAI-tool where grey elements represent software.

By detaching user input from the optimization no working principles or foreknowledge about simulation outcome are introduced. Instead, user input on *flows* and *physical properties* thereof are included in the simulation. The objective function links the user input *function* to the optimization by giving feedback on fulfillment of the function in the form of a fitness value.

For clarification, the mechanical engineering problem *transmitting mechanical rotational energy* is entered in the scheme for the CAI-tool (see Figure 3.6). For example, the two solutions seen in Figure 3.1 are possible solutions to this problem.

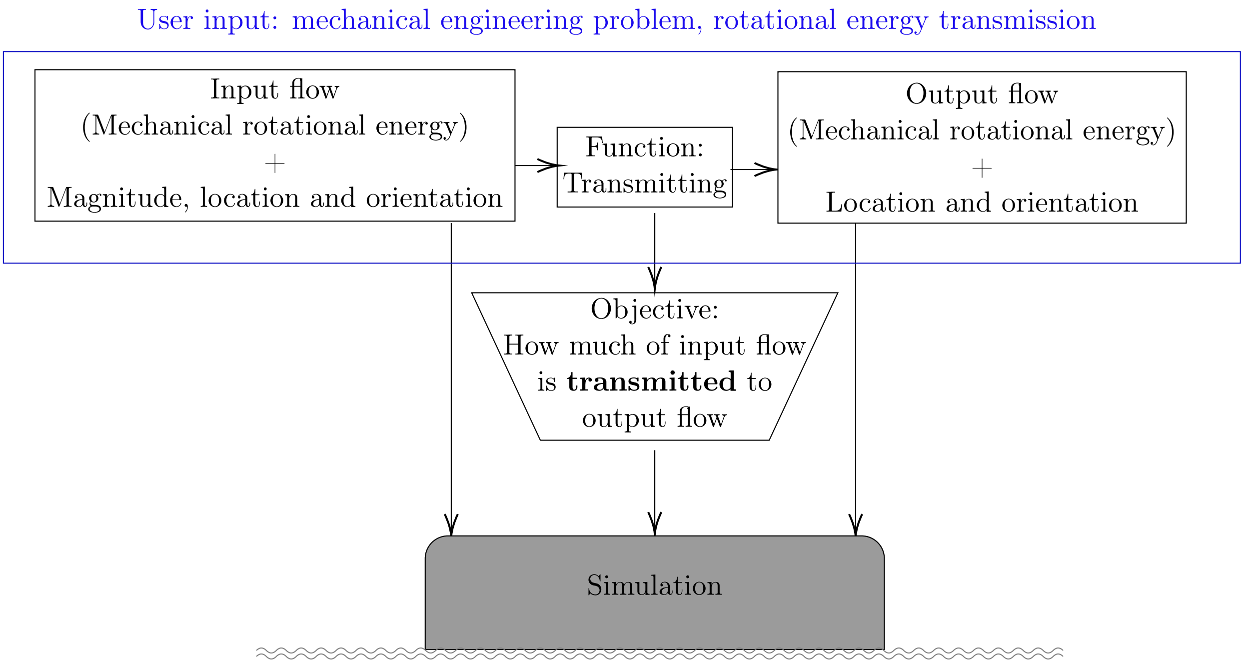


Figure 3.6. Example of the mechanical engineering problem *transmitting mechanical rotational energy* in the schematic view of the proposed CAI-tool.

To prove that this method for a CAI-tool leads to multiple working principles, it is proposed to develop a software implementation of this CAI-tool as a proof of concept.

3.3 Software development for a proof of concept

A proof of concept will be made based on Evosoro, software made by Cheney et al. [2]. Instead of generating soft robots, the software will be adapted to be the implementation of the aforementioned method for a CAI-tool. This will be used to solve a mechanical engineering problem of which the output will be manually analyzed using the previously described method presented in Section 3.1.

3.3.1 Demands for a successful proof of concept

To determine whether the PoC is successful and the desired goal is reached, the following demands are determined that have to be met:

- The mechanical engineering problem will need to be described by one function and an input- and output flow, as is prescribed by the CAI-tool (Figure 3.5).
- Any basic element used to build artifacts in the proof of concept should, in itself, not fulfill a function that is sought in the mechanical engineering problem that is to be solved.

- The proof of concept will be successful when at least two solutions for the entered mechanical engineering problem, that have different working principles, are produced in one run. When just one working principle is found, this could just be because that one is by far the best performing working principle and the software works well. However, it cannot be eliminated in that situation that it is because user input or software parameters already implicated this working principle for this problem resulted in a predetermined solution. Therefore an outcome with multiple working principles is required for this proof.

3.3.2 Evosoro

The version of Evosoro that was used by Cheney et al. [2] is used as a basis in this proof of concept. Soft robots consisting of voxels of four different materials (hard, soft, contracting, and expanding) are optimized using an objective function that refers to the displacement of the robot, in combination with an objective of low age and a low number of voxels. The age of an individual starts with zero at its initial generation and is increased for every generation it survives. Figure 3.7 shows some of the soft robots that are generated by this version of Evosoro.

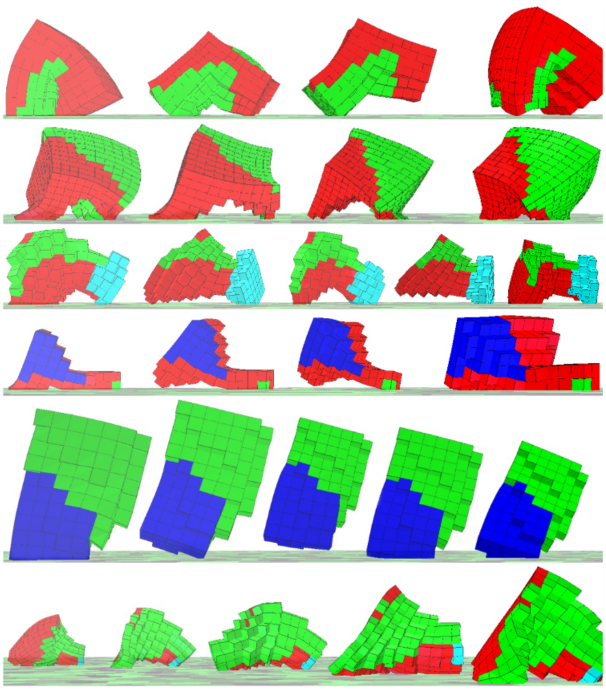


Figure 3.7. Time series of various walking soft robots that were generated using Evosoro [2]. Red: expanding then contracting voxels, green: contracting then expanding voxels, light blue: low stiffness voxels, dark blue: high stiffness voxels.

This version is used as a basis to make the software for the proof of concept for this research, by adding or changing the elements of Evosoro, as shown in Table 3.4.

	Evosoro	Change in parameter values	Addition to/change in code
Boundary conditions	Changing temperature	Allowed DOFs, applied moments	-
Selection methods	Pareto, pareto tournament	-	Pareto diversifying ancestry, pareto reset, simulated annealing
Simulation sensor	Voxel movement	-	Voxel rotation
Materials	Expanding, contracting, soft, hard	Soft, hard, rotation measurement label	-
Valid individuals	Minimum percentage expanding or contracting voxels, minimum number of voxels	Minimum number of voxels	-
Environmental temperature	Changing with time	Constant	-
Simulation settings	-	*see experiments*	-
Objective functions	-	*see experiments*	*see experiments*
Population properties	-	*see experiments*	-
Number of objects in one individual	Reduced to one	-	Multiple objects allowed

Table 3.4. Changes and additions to Evosoro code

Other parts of Evosoro will remain unchanged for this proof of concept. For clarity, an overview and relevant elements of the program will be summarized here, including the aforementioned additions and changes.

Overview of the code

For every run, Evosoro will execute the following steps to generate solutions. References are made to the elements of the proposed CAI-tool (see Figure 3.5 on page 42) in *italics*:

1. At the start of the optimization, an initial population is generated through the random generation of a single CPPN per individual. This is done in the *optimizer* seen in Figure 3.5, where the CPPN is the *shape representation*. *Materials* are determined by the CPPN as well.

2. Every individual, converted to voxels, (now an *artifact*) is simulated in Voxelyze, the *simulation*. Doing this, the objective functions are evaluated to determine the *fitness values*, which are then saved in an output file. When individuals have already been previously simulated, cached values are retrieved. Crashing simulations are restarted but given up on after a predetermined amount of time.

3. The population is sorted according to fitness (in the *optimizer*).

4. A selection is made (in the *optimizer*) to populate the next generation. Also, statistics of the old population are saved at this moment. If a new optimum is found, a copy of this individual is saved in a separate folder.

5. Age is increased by one for every individual and the population is doubled by adding a mutated copy of each individual.

6. A number of new randomly generated individuals are added to the population with age zero. Now, the steps 2-6 above are repeated until stop conditions are reached.

The individuals that perform best (presented as *mechanical engineering solutions*) are visually inspected during, or after the run using VoxCAD. VoxCAD is the graphical user interface (GUI) used with Voxelyze, see 3.3.2.

```
Generate population (Pop) with n random individuals;
while (stopping criterion not satisfied) do
    evaluate fitness of each new individual in Pop;
    sort Pop according to fitness values;
    while (|Pop|< n) do
        add P_i to Pop_new using some selection method;
    end
    replace Pop with Pop_new;
    increase age of every P_i;
    duplicate population Pop to Pop_clone;
    for (i in Pop_clone) do
        mutate(P_i);
    end
    add m random individuals to Pop;
    add Pop_clone to Pop;
end
```

Algorithm 2: Pseudo-code for a Evosoro. With *Pop*: population, n: number of individuals, m: number of random individuals, Pop_{new}: new population, P_i: individual i in population *Pop*.

Representation of individuals

An individual has a genotype: the CPPN, which represents the artifact made from voxels as described in Section 2.3.1. The input nodes for each CPPN are the x, y and z location as well as the distance to the center d. The output nodes define the existence of voxels and the type of material they are made of. This is done by enabling dependency between output nodes, e.g. the node that determines the material type is dependent on the node that determines voxel existence. The function in a node can be either one of these functions: $(+/-)|x|$, $(+/-)\sqrt{x}$, $(+/-)\sqrt{|x|}$, $Sin(x)$. An individual is initialized with each input attached to each output after which a mutation is performed.
Filters can be applied or changes can be made to the phenotypes themselves as well. In the original Evosoro script, only the largest continuous object is kept. This function is disabled in the software so that multiple objects can exist. Voxels can be placed in set positions for each individual as well. All filters or changes after genotype expression are not incorporated or converted back into the genotype however and have to be applied to every individual.

Simulation environment

Voxelyze [130] is used to simulate the behaviour of artifacts to determine the fitness and other properties of each individual. Voxelyze is able to model statics, dynamics and non-linear deformation of heterogeneous (soft)bodies in a computing time-efficient way. A VXA-file (VoxCAD Analysis) is used as input for Voxelyze, after which a txt-file is produced as output with relevant data such as the obtained fitness value. Material parameters that can be adjusted in Voxelyze are the following:

- Name
- Color
- Stiffness
- Poisson's ratio, set at 0.35 [2]
- Density
- Coefficient of thermal expansion, set at 0 so no thermal expansion can occur.

- Static friction coefficient, set at 1 [2]
- Dynamic friction coefficient, set at 0.5 [2]

Other relevant parameters that can be adjusted in Voxelyze are:

- Material stiffness model (linear, with failure or from a data set. Linear is used.)
- Initiation time after which measurements start.
- Stopping condition for a simulation (e.g. number of time steps, equilibrium or simulated time)
- Voxel edge size. This is set at 10 mm [2].
- Collision horizon. This determines the distance between voxels where collisions are detected, which is set at 2 [2].
- Existence of a floor, this prohibits voxels to move beyond $z < 0$.
- Bond damping, set at 1 [2].
- Collision damping, which is set at 0.8 [2].
- Ground damping, always set at 0.1 [2].

Boundary conditions (force, moment or displacement) can be added as well on specific locations or volumes. The size of the time step for the simulation is determined automatically by the highest occurring eigenfrequency in the object(s).
Measuring displacement of a voxel, or an average of all voxels of the same material, is done by recording the position at the initiation time and at the end of the simulation. Measuring relative rotation (to the environment) is done in the same manner but absolute, or total, rotation is measured by adding the number of total rotations during the simulation. These are recorded separately during the simulation by monitoring the angle for a change from 0 to $2/\pi$ or vice versa, at which moment an additional rotation is recorded.

Selection methods

Selecting individuals for populating the next generation is done by using the following techniques:

- Pareto selection: this is determined for each individual by how many other individuals they dominate (perform equal or better for all objectives and better for at least one objective). Similar scoring individuals are grouped in pareto levels. Within a pareto level, the individuals are sorted according to the objective functions, starting with the one with lowest importance and ending with the objective that determines fitness. First, it is tried to fit a complete level in the new population, starting with the highest level. When limited places are available, individuals with higher fitness have a larger chance of entering the new population.

- Pareto tournament selection: two random individuals are picked and the dominating individual enters the new population.

Newly added selection methods:

- Population-based simulated annealing, as described in Section 2.3.2: based on regular simulated annealing, a virtual temperature is decreased from the start of the first generation, to be at its lowest point at the final generation. The temperature (see Equation 3.2) influences the chance better individuals have to make it to the next generation, when the temperature is low, individuals with a high fitness have a high chance to survive. Individuals are copied to the new generation randomly using a beta-distribution (see Equation 3.1) until the population size is reached.

$$n_{individual} = B(\alpha, \beta) \cdot Pop_{buffer-size} \tag{3.1}$$

Equation beta-distribution B. $\alpha = 1$ see Equation 3.2 for β.

$$\beta = T_{annealing} = \frac{a \cdot Gen}{Gen_{max}} + b \tag{3.2}$$

Equation for annealing temperature. Gen: generation number, Gen_{max}: final generation number, a: total decrease of annealing temperature, and b: onset temperature, equal beta-distribution for $b = 1$.

- Diversifying ancestry: a newly introduced method, individuals are selected starting with the highest fitness but their similarity to individuals in the new population is checked. If an individual has an overlap of more than a certain amount of ancestors with more than a certain amount of individuals in the new population, it is rejected. This selection method promotes diversity.

- Pareto selection with reset: introduced in this research as well: at the moment that fitness improvement stagnates (i.e. does not improve more than a set value) after a set number of generation (see Equation 3.3), a reset is performed. The reset consists of the removal of the individual with the highest fitness along with individuals that have an overlap of ancestry to a certain degree with that individual. The deleted individuals are replaced by newly generated individuals. This technique has a similar goal as the sequential niche technique as described in Section 2.3.2: getting out of a local optimum.

$$\sum_{gen-h}^{gen} \frac{f_{gen} - f_{gen-1}}{f_{gen-1}} < a \tag{3.3}$$

Equation for reset condition a. f: fitness of best individual gen: current generation, h: amount of generations to wait for reset.

Mutation

Mutation in NEAT is described in Section 2.3.1. One of the following actions is performed on the CPPN as a mutation:

- Random node additions(10) and removals(0).
- Random link additions(10) and removals(5).
- Random functions to change (100).
- Random weight changes (100).

The value in brackets indicates the number of times this action is performed in the mutation directly after the creation of a new individual. After each mutation, the phenotype is evaluated. When the phenotype has not changed or an individual is produced that does not meet demands to be valid, another mutation attempt is made. The maximum mutation attempts for one individual is set at 1500. These demands for individuals can only be made about properties that are derived from the phenotype without simulation, e.g. "allow no empty individuals" or "the occurrence of material x needs to be above threshold y".

Problem parameters

Parameters for general settings for the problem that determine the properties of the individuals, population and the simulation:

- Individual size in voxels (x, y, z)
- Population size
- Random individuals to insert for each generation
- Maximum number of generations.
- The amount of time waiting for a simulation result before restarting it.

- The maximum time for all the simulations in one generation to be finished, average per individual.
- Maximum running time for the program.
- The seed that is used to generate numbers (using the same seed will result in exactly the same run)

3.3.3 Problem selection

The function “Transmission of rotational mechanical energy acting on vertical line A, to parallel positioned line B” will be taken as the mechanical engineering problem used in the proof of concept. The problem is selected so that it meets the following demands: no trivial solution can be reached in starting conditions of the software and at least two working principles can be simulated by the used software (Voxelyze).
Converting this problem to an objective function will be done by using the method for conversions of mechanical engineering problems to the input of a CAI-tool (see Section 3.2). This will be executed in Section 4.1.

3.3.4 Method for analyzing results

To determine whether the proof of concept has met the requirements set in Section 3.3.1, solutions in the 'best so far' that perform the function well (with a fitness above a certain threshold) are imported in VoxCAD and analyzed manually using the variety rating combined with the C&CM as described in Section 3.1.
For voxel surfaces it is determined whether it can be seen as a WS and which WSP it is part of. CSSs are described similarly but by groups of voxels that take part in connecting the same flows. The voxels are colored to indicate to which connect and channel structure they belong. This means that even voxels that are attached to one another can be analyzed as different parts.

The quality of a solution is determined by its fitness value. A distinction can be made between non-working solutions resulting in a very low fitness value ($fitness \approx 0$) working solutions, with a fitness value that reaches a set value after some time ($fitness = x$) and finally continuously working solutions, of which the fitness value keeps increasing over time ($fitness = x \cdot t$).

3.3.5 Determining experimental runs

The experiments, i.e. each run of Evosoro, performed as part of the proof of concept are designed heuristically. First, optimization parameter values are determined empirically, after which more experiments are performed using these values.

Determining optimization parameter values

Comparative runs are done to determine values for parameters that influence simulation and overall program computing time, solution variety (visual inspection) and solution quality (see the method in the previous section). After determining parameters that influence program computing time, these parameters are chosen so that faster runs can be done to explore the influence of other parameters on the variety and quality of solutions.

Performing runs

Runs to determine optimization parameters that result in working solutions are explored further. When more than one working principle is seen upon visual inspection of working solutions or high fitness values are seen, the working principle is verified by using the method for determining solution variety. The optimization parameters, as well as the optimization parameters that result in runs with a higher than average number of working solutions, are used to perform more runs. In the runs with initially promising results, parameter values that result in more computer time-consuming runs are used or more identical runs are performed using a different seed for random values.

This iterative process is continued until satisfactory results are obtained: at least two solutions within one run must solve the problem, where each makes use of a different working principle.

4 Proof of concept

The proof of concept is split into three parts: the conversion of the mechanical engineering problem to the set-up needed in Evosoro using the method for CAI-tools presented in Section 3.2, the different runs that were performed and, for each run, determining variety of the solutions with the highest fitness, using the method from Section 3.1. The code as used in this research can be found on https://github.com/DanielLinsc/evosoro.

The simulations in Voxelyze (C++) were performed in parallel on a cluster with four nodes, each consisting of a Dell PowerEdge R610 with two 6-core 2.66 GHz Intel Xeon X5650 processors. The optimization script (Python) itself, that generates the files which are then simulated in Voxelyze, is run on another, additional, node with the same processors but this script does not make use of multithreading.

4.1 Converting the problem to input values

The function *Transmission* from A to B of rotational mechanical energy from line A to rotational mechanical energy at parallel positioned line B is taken as the mechanical engineering problem. It is converted to the input elements of the CAI-tool:

- The orientation of input flow A is parallel to the orientation of output flow B but they do not coincide.
- Input domain is rotational mechanical energy. This is provided by an effort: imposing a moment as a boundary condition on A.
- The function is to channel-transfer transmit the flow from A to B.
- Output domain is also rotational mechanical energy, this is measured as the flow: average angular velocity at B.

Measuring how well the function is performed is done by measuring how much flow at A (rotational mechanical energy) is transmitted to B. Please note that remaining energy can stay in the system in the form of kinetic or potential energy and the rest is dissipated through friction or damping. The following parameters do not result in a change of problem definition but do influence experiments. Values are varied between experiments.

- Location A is positioned around $(0.5, 0.33)$, always indicated with **orange** voxels.
- Location B is positioned around $(0.5, 0.66)$, indicated by **yellow** voxels.
- The moment imposed at A has different values for the experiments.
- The shape of A and B are not the same for each experiment.
- The density and elastic modulus of the soft **light blue** and hard **dark blue** voxels is varied.

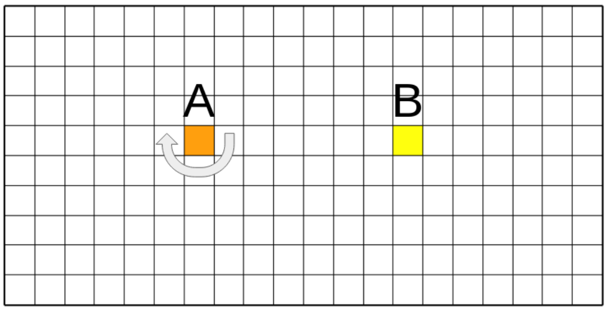


Figure 4.1. Example of simulation environment, top view. A and B are marked by voxels on which a moment is imposed and of which the rotation is measured, respectively.

4.2 Performed runs

The runs that produced relevant results are presented in this section, as well as some undesirable results. More simple runs that require less computation time are shown first, after which parameters are changed that result in more time-consuming runs. As the objective, the rotation of A and B in the similar or opposite direction is used. The main objective is always for B to turn counterclockwise but the direction of the moment is switched in some runs. Additionally, two minimizing objectives are added for the age of the individual as well as the number of voxels of an individual. These were used in the original Evosoro as well. Minimizing age promotes novel solutions over old ones and minimizing voxels avoids trivial large structures over smaller ones [2]. Pareto selection (see Section 3.3.2) is used in most runs and includes these other objectives to be able to sort the population in pareto levels.
As described in Section 3.3.5, parameters and settings for the runs are determined empirically. Before elaborating on each run, an overview is given of the important discriminating elements for each run:

Run A: a small run is performed, resulting in a single working principle.
Run B: large deformation is detected in simulation that is resolved in the next run.
Run C: the objective function is changed from measuring angle to total rotation.
Run D: a larger individual size is used and the moment at A is applied in the opposite direction to the desired rotation at B. This run results in one continuous working principle.
Run E: a second, softer, material is added. The moment is reversed, no continuous working principles are found.
Run F: simulated annealing is tried as a selection method.
Run G: same as run F, more random individuals inserted.
Run H: ancestor reset is tested as a selection method, the moment is applied in opposite direction.
Run I: pareto reset selection method is used and three continuously working solutions with different working principles are found.
Run J: individual size as well as population size are reduced and the run is performed ten times with a different seed. Just one working principle for a continuously working solution is found.
Run K: gravity is introduced laterally. One working principle for a continuously working solution is found.
Run L: two input nodes that refer to the distance of A and B are used in the CPPN. Two continuously working principles are found with a different working principle.
Run M: same as run L but a larger individual size is used.

The exact parameter values and settings used in each run are found in the corresponding section, but an overview can also be seen in Appendix A.5 in Table A.6 and A.7.

4.2.1 Run A

The first run is performed with just one material, a small population size (10) and a small individual size (10,5,1). All parameter values can be seen in Table 4.1. A and B are each applied to a set of three voxels which are present in each individual, as is shown in Figure 4.2. The objective function that determines fitness in this experiment measures rotation of B voxels relative to their environment, at the end of each simulation. This run was allowed to run for 5 minutes.

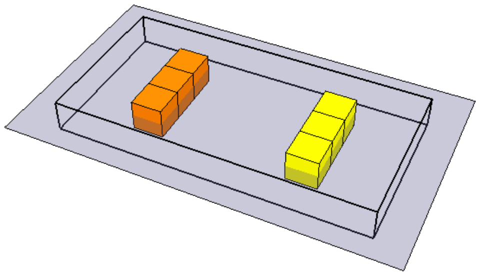


Figure 4.2. A and B are marked by voxels on which a moment is imposed and of which the rotation is measured, respectively. The box indicates the maximum size of an individual.

Experiment	**A**
Size (x,y,z)	(10,5,1)
Population size	10
Random individuals	10
Number of generations	26
Initiation time	0.1 s
Simulation time	2 s
Hard material $E(MPa), \rho(\frac{kg}{m^3})$	$500, 1 \cdot 10^6$
Soft material $E(MPa), \rho(\frac{kg}{m^3})$	-
A material $E(MPa), \rho(\frac{kg}{m^3})$	$500, 1 \cdot 10^6$
B material $E(MPa), \rho(\frac{kg}{m^3})$	$5, 1 \cdot 10^6$
Location of A	(0.72,0.45,0)
Location of B	(0.32,0.45,0)
Moment at A	-100 Nm
Selection method	Pareto
Computing time (h:m)	0:5

Table 4.1. Parameter values used in run A.

After five minutes this run was stopped. At this point 26 generations had been evaluated. Individuals that showed a significant increase in fitness were evaluated manually. Not only the objective function (angle of B) but also rotation and rotation velocity are included in Table 4.2. Neither rotation nor rotation velocity was used as feedback for this run. Measurement of rotation was implemented later but calculated for this run for the sake of completeness.

Individual	A1	A2	A3	A4
Generation	0	1	4	26
Fitness (Angle)	0	2.94	3.41	5.2
Rotation	0	2.94	3.41	17.75
Rotation velocity	0	-6.52	45.24	288.5
Age	0	1	0	17
No. voxels	50	13	27	11

Table 4.2. Four solutions generated in run A.

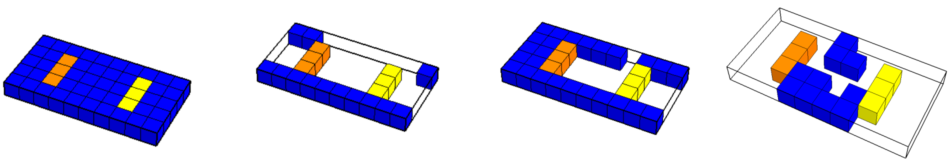


Figure 4.3. Phenotype output for individuals A1 to A4 (left to right).

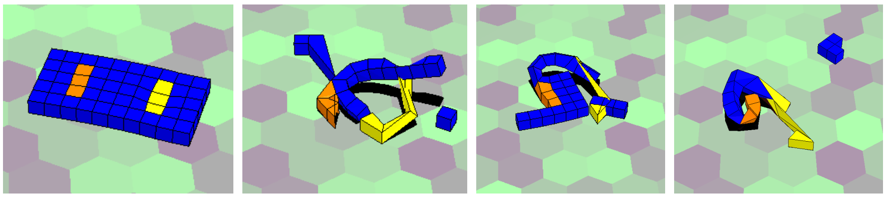


Figure 4.4. Simulations for individuals A1 to A4 (left to right).

Analysis run A

Unrealistic behaviour of voxels in the simulation is seen: large deformations and voxels moving through each other where collisions should have occurred. This will be solved in later runs. It seems that the number of voxels is not correct but this can be blamed on the fact that this measurement takes place before the introduction of the voxels indicating location A and B. This introduction overwrites voxels belonging to the evolved individual that might be present in the same spot.
Still, C&CM is used to analyze the two solutions with the highest fitness (highest angle): A3 and A4. The voxels are colored to indicate to which connect and support structure they belong. Figure 4.5 shows two situations that define the working principle for solution A3.

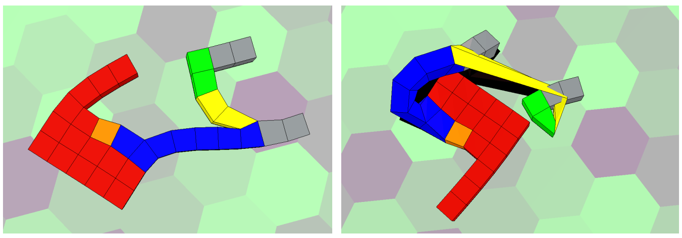


Figure 4.5. Coloring used in the analysis of A3. Left: coupling of rotation can be seen. Right: the red part makes contact with the green part, which has a negative impact on the function (i.e. flow in the opposite direction).

The analysis is shown in Figure 4.6 with specifications for the flows that cross the WSs in Table 4.3. Angular momentum and torque is converted to linear momentum, angular momentum, force and torque, while a non-continuous negative effect can be seen as collisions take place between WS_8 and WS_9.

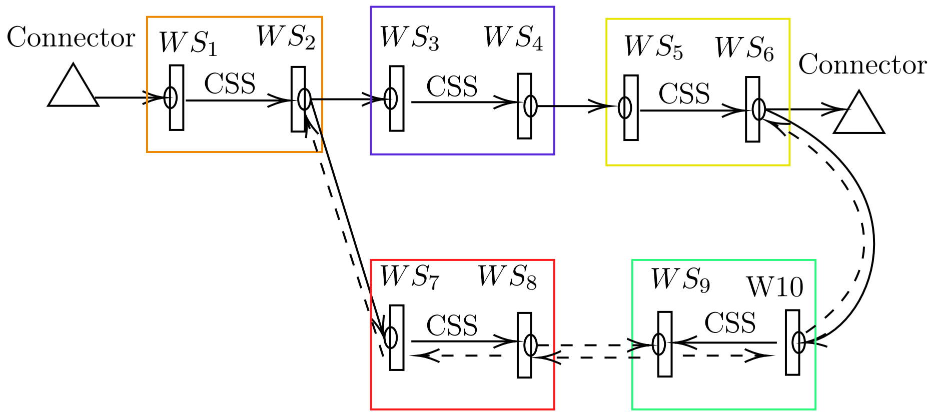


Figure 4.6. Graph analysis using the C&CM for solution A3. Colors represent the voxels seen in Figure 4.5

WS	Flow domain	Displacement	Effort	Carrier
$WS_{1,6}$	Rotational 1D	Angle	Torque	Angular momentum
$WS_{2-5,7-10}$	Mechanical 3D	Position, Angle	Internal force, Torque	Linear & angular momentum (3D)

Table 4.3. Description of WS flows in Figure 4.6.

The analysis for solution A4 is shown in Figure 4.8. Specifications for the flows that cross the WSs and are stored in the CSSs can be seen in Table 4.4 and 4.5. Torque is converted to linear and angular velocity which is again stored as potential energy. The potential energy is stored and released in CSS_d, as can be seen in Figure 4.7. CSS_a also stores a high amount of potential energy, but since no work is performed in relation to the storage, this is not shown in the graph. Both CSS_b and CSS_e store kinetic energy.

This C&CM analysis leads to the conclusion that many differences exist between the working principles of solution A3 and A4.

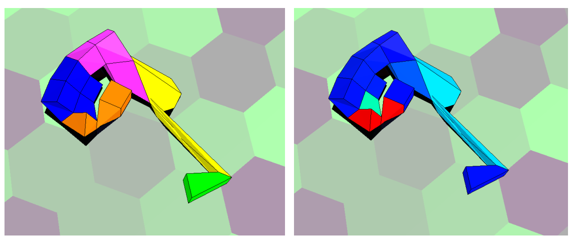


Figure 4.7. Solution A4 with colors used in the analysis (left). Potential energy shown (right): blue-red equals low-high values.

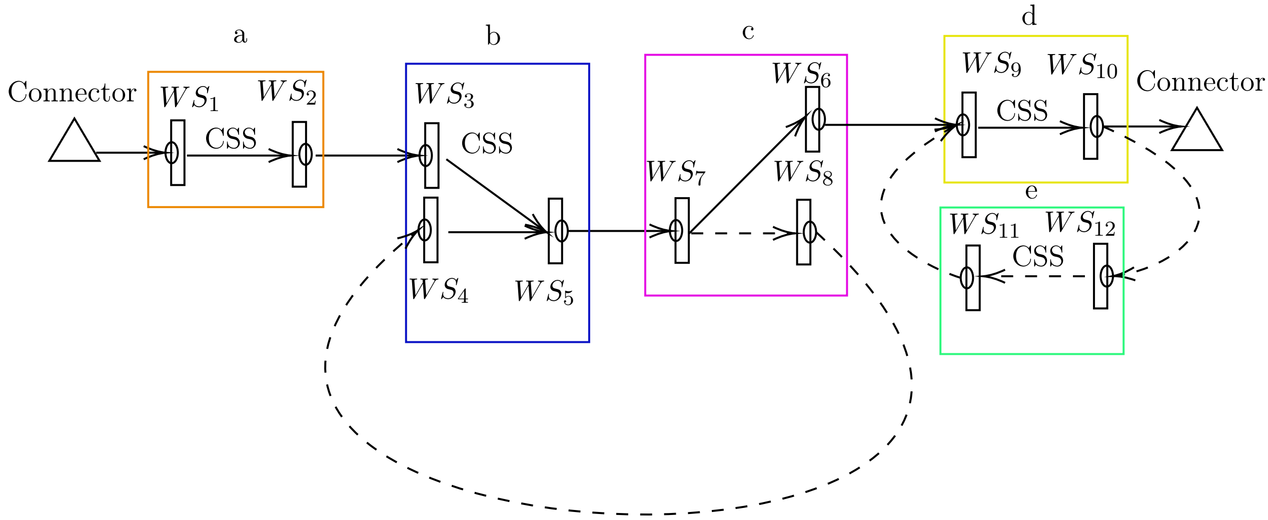


Figure 4.8. Graph analysis using the C&CM for solution A4. Colors represent the voxels seen in Figure 4.7.

WS	**Flow domain**	**Displacement**	**Effort**	**Carrier**
$WS_{1-3,6,9-12}$	Rotational 1D	Angle	Torque	Angular momentum
$WS_{4,5,7,8}$	Mechanical 3D	Position, Angle	Internal force, Torque	Linear & angular momentum (3D)

Table 4.4. Description of WS flows in Figure 4.8.

CSS	**Flow domain**	**Energy storage**
$CSS_{b,c}$	Mechanical 3D	Kinetic energy
CSS_c	Mechanical 3D	Potential energy

Table 4.5. Description of CSSs in Figure 4.8.

4.2.2 Run B

Even though just a single working principle was seen, run A showed an increase of fitness over the course of the generations. This run is repeated but with a larger population and for more generations.

Experiment	**B**
Size (x,y,z)	(10,5,1)
Population size	200
Random individuals	2
Number of generations	1000
Initiation time	0.1 s
Simulation time	2 s
Hard material $E(MPa), \rho(\frac{kg}{m^3})$	$500, 1 \cdot 10^6$
Soft material $E(MPa), \rho(\frac{kg}{m^3})$	-
A material $E(MPa), \rho(\frac{kg}{m^3})$	$500, 1 \cdot 10^6$
B material $E(MPa), \rho(\frac{kg}{m^3})$	$5, 1 \cdot 10^6$
Location of A	(0.72,0.45,0)
Location of B	(0.32,0.45,0)
Moment at A	-100 Nm
Selection method	Pareto
Computing time (h:m)	4:24

Table 4.6. Parameter values used in run B.

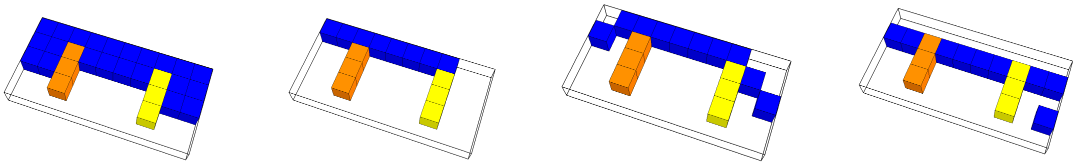


Figure 4.9. Phenotype output for individuals B1 to B4 (left to right).

Individual	**B1**	**B2**	**B3**	**B4**
Generation	0	3	433	679
Fitness (Angle)	0.09	9.94	11.96	11.99
Rotation	0.09	35.08	-3.11	-151.4
Rotation velocity	0	0	0	-131.0
Age	0	3	276	220
No. voxels	18	8	10	11

Table 4.7. Four solutions generated in run B.

Analysis run B

Even without a C&CM analysis, visual inspection of solutions B3 and B4 shows that even though a high fitness was allocated to these individuals, rotation at output flow location (yellow voxels) is clockwise. Even though a positive counter-clockwise rotation is used to determine fitness, the opposite seems to be achieved while the fitness value does not reveal this.
A closer look shows that at the moment the stop condition (simulation time = 2s) is reached, a nearly complete rotation is reached. Further inspection of Voxelyze code reveals that instead of $2 \cdot \pi$ as maximum absolute rotation, rotation is measured from $-2 \cdot \pi$ to $2 \cdot \pi$ which is equal to two complete rotations. This explains why fitness values approach this value as generations progress and after individual B4, no fitter individual is found: the objective function measures the angle relative to the origin which can not exceed two rotations: $4 \cdot \pi$.
Just as in run A, unrealistic behaviour of voxels in the simulation is seen: large deformations

and voxels moving through each other. Both of these problems are addressed in later runs.

4.2.3 Run C

It is hypothesized that by replacing the previously used angle measurement by a measurement of rotation as objective function, the problem of a maximum measurable angle, encountered in run B, is avoided.
Other values for the combination of material parameters and applied moment are determined empirically. This is done so that very large deformations do not occur in the simulations. A lower moment is applied at A and all materials are adjusted to have the same (lower) density and (higher) elasticity. This combination causes the voxels to move faster which allows the simulation time to be diminished.
Instead of three voxels, only one central voxel is used on location A and B to apply the moment and measure rotation, respectively. Figure 4.1 shows a top view of the shape of these voxels for every following run. This leaves more room for evolvable voxels.

Experiment	**C**
Size (x,y,z)	(10,5,1)
Population size	50
Random individuals	2
Number of generations	140
Initiation time	0.01 s
Simulation time	0.2 s
Hard material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
Soft material $E(MPa), \rho(\frac{kg}{m^3})$	-
A material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
B material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
Location of A	(0.72,0.45,0)
Location of B	(0.32,0.45,0)
Moment at A	-5 Nm
Selection method	Pareto
Computing time (h:m)	0:20

Table 4.8. Parameter values used in run C

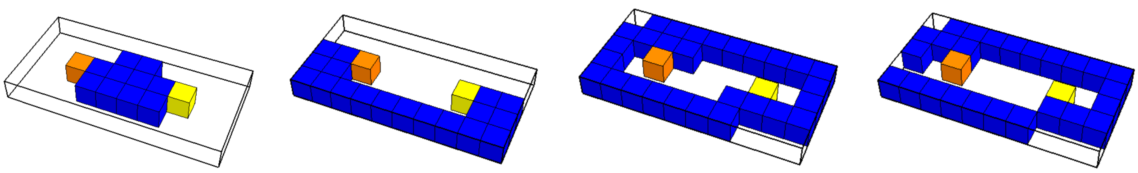


Figure 4.10. Phenotype output for individuals C1 to C4 (left to right).

Individual	**C1**	**C2**	**C3**	**C4**
Generation	4	5	85	97
Fitness (Rotation)	0.22	1.37	2.50	3.00
Rotation velocity	0	0	0	288
Age	4	5	2	14
No. voxels	12	18	28	25

Table 4.9. Four solutions generated in run C.

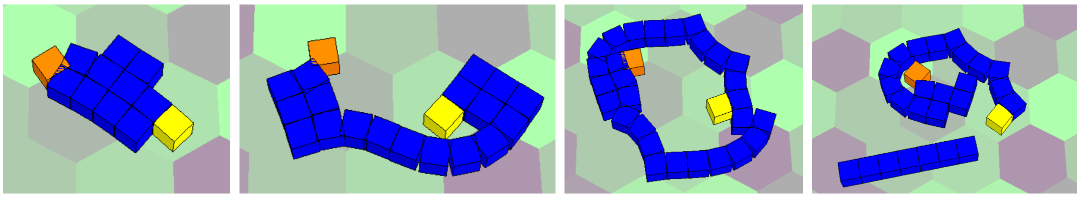


Figure 4.11. Simulations for individuals C1 to C4 (left to right).

Analysis run C

No continuous working solution was found. All working principles of individuals are the same, see the C&CM analysis in Figure 4.12 and Table 4.10.
In solution C4 some voxels appear that are not connected to any moving part. Re-running solution C4 without these seemingly non-functional voxels results in a fitness value (rotation) of 3.03 which is very close to the original value of 3.00. This indicates that even though these voxels form a part that does not increase functionality, its existence is compatible with the complete solution C4, quite similar to vestigial features seen in nature.

WS	**Flow domain**	**Displacement**	**Effort**	**Carrier**
$WS_{1,6}$	Rotational 1D	Angle	Torque	Angular momentum
WS_{3-5}	Mechanical 3D	Angle, position	Internal Force, Torque	Linear & angular momentum (3D)

Table 4.10. Description of WS flows in Figure 4.12.

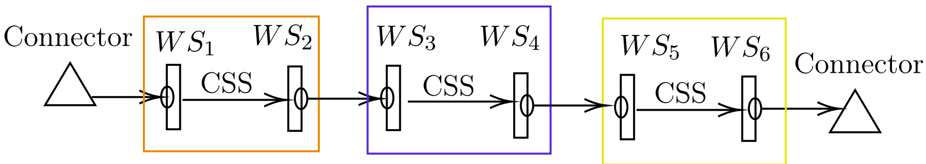


Figure 4.12. Graph analysis using the C&CM for solution C1-4. Colors represent the voxels seen in Figure 4.11.

4.2.4 Run D

To allow more intricate solutions to form, the allowable individual size is increased in this run. The direction of the moment is also reversed and locations of A and B are adapted to the location of the centre of the voxels. This will be done in the following runs as well.
The error in rotation conversion discovered in run B is corrected so that $2 \cdot \pi$ resembles a complete rotation instead of $4 \cdot \pi$.

Experiment	D
Size (x,y,z)	(20,11,1)
Population size	250
Random individuals	20
Number of generations	3000
Initiation time	0.01 s
Simulation time	1 s
Hard material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^3$
Soft material $E(MPa), \rho(\frac{kg}{m^3})$	-
A material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
B material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
Location of A	(0.66,0.47,0)
Location of B	(0.31,0.47,0)
Moment at A	0.5 Nm ↺ ↻
Selection method	Pareto
Computing time (h:m)	224:03

Table 4.11. Parameter values used in run D

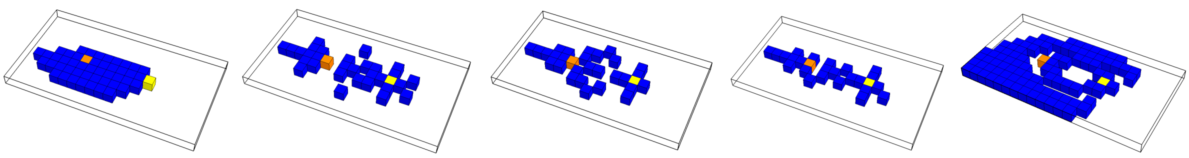


Figure 4.13. Phenotype output for individuals D1 to D5 (left to right).

Individual	D1	D2	D3	D4	D5
Generation	0	828	964	2789	1970
Fitness (Rotation)	0.01	4.79	6.08	6.77	2.65
Rotation velocity	0	15.67	0	45.82	24.89
Age	0	309	445	2270	1142
No. voxels	47	31	29	24	94

Table 4.12. Five solutions generated in run D.

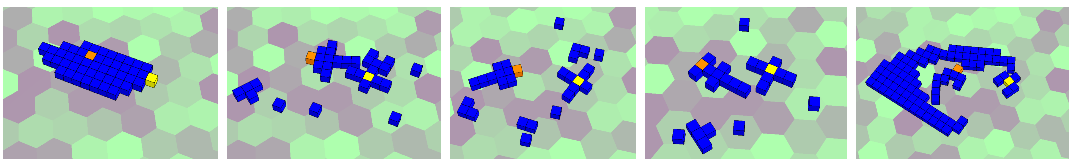


Figure 4.14. Simulations for individuals D1 to D5 (left to right).

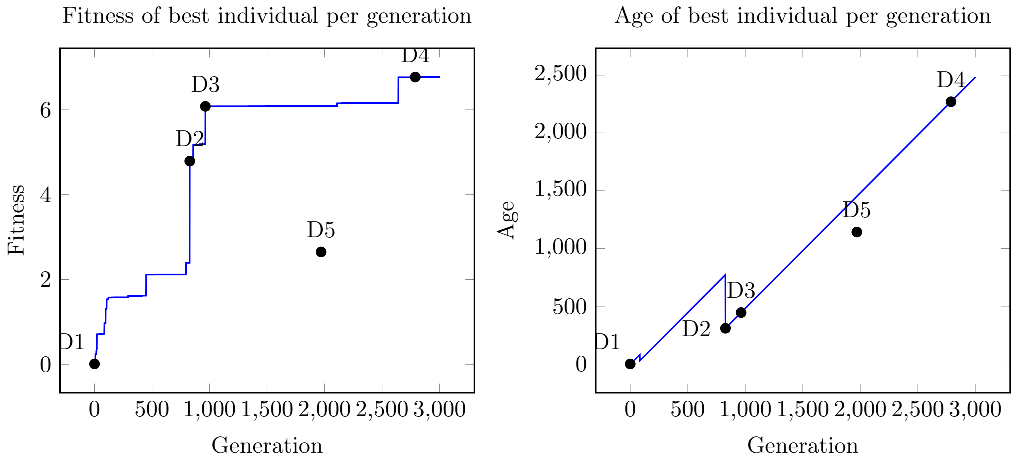


Figure 4.15. Fitness and age for the best individual over the course of generations for run D.

Analysis run D

Multiple working principles are seen in the solutions. Solution D1 to D4 are selected because they represented a large increase in fitness. Solution D5 however, was selected randomly from the working solutions (with $fitness > 0$) and shows a different working principle from the other solutions.
First solution D1 is analyzed, as seen in Table 4.13 and Figure 4.16.

WS	**Flow domain**	**Displacement**	**Effort**	**Carrier**
$WS_{1,6}$	Rotational 1D	Angle	Torque	Angular momentum
WS_{3-5}	Mechanical 3D	Angle, position	Internal Force, Torque	Linear & angular momentum (3D)

Table 4.13. Description of WS flows in Figure 4.16.

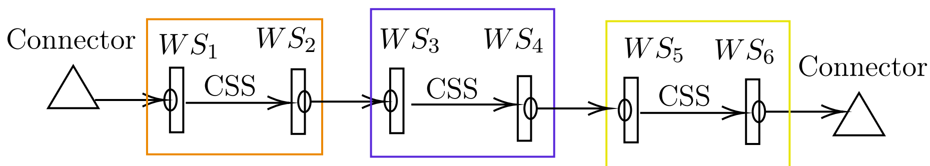


Figure 4.16. Graph analysis using the C&CM for solution D1. Colors represent the voxels seen in Figure 4.14.

Solution D2 is analyzed (see Table 4.14, Figure 4.17 and 4.18) and shows a different working principle than solution D1. WS_4 and WS_5 are duplicated because they are connected at different moments and different locations but still convey the same flow (mechanical 3D) and use the same CSS. Grey voxels are non-functional and not taken into account in the C&CM analysis.

WS	**Flow domain**	**Displacement**	**Effort**	**Carrier**
$WS_{1,14}$	Rotational 1D	Angle	Torque	Angular momentum
$WS_{4-8,4B,5B}$	Mechanical 3D	Angle, position	Force, Torque	Linear & angular momentum (3D)
$WS_{2,3,9-13}$	Mechanical 3D	Angle, position	Internal Force, Torque	Linear & angular momentum (3D)

Table 4.14. Description of WS flows in Figure 4.18 for solution D2.

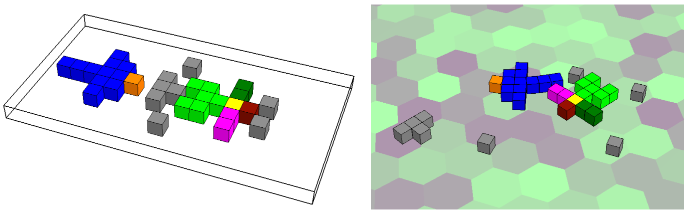


Figure 4.17. Solution D2, voxel coloring as it is used in Figure 4.18 to indicate parts. Grey voxels are not functional.

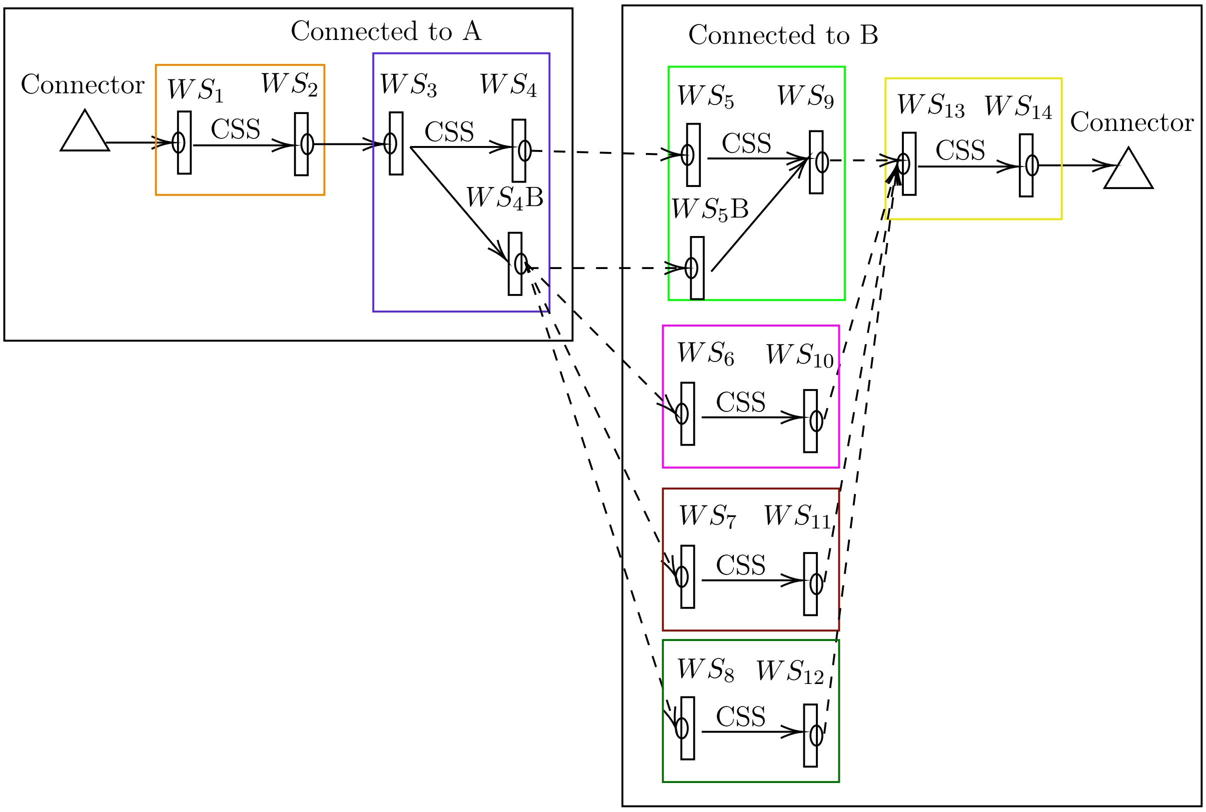


Figure 4.18. Graph analysis using the C&CM for solution D2. Colors represent the voxels as seen in Figure 4.17. Black boxes separate the parts that are connected to flow input A and flow output B.

Solution D3 and D4 are analyzed as well (see Table 4.15, Figure 4.19, 4.20 and 4.21) which results in a highly similar graph resembling solution D2 (Figure 4.18) with only a duplication of WS_4 and WS_5 left out. This duplication makes solution D2 a different embodiment of the same working principle as solutions D3 and D4, that have the same working principle and embodiment with only different details.

WS	**Flow domain**	**Displacement**	**Effort**	**Carrier**
$WS_{1,14}$	Rotational 1D	Angle	Torque	Angular momentum
WS_{4-8}	Mechanical 3D	Angle, position	Force	Linear momentum (3D)
$WS_{2,3,9-13}$	Mechanical 3D	Angle, position	Internal Force, Torque	Linear & angular momentum (3D)

Table 4.15. Description of WS flows in Figure 4.21 for solutions D3 and D4.

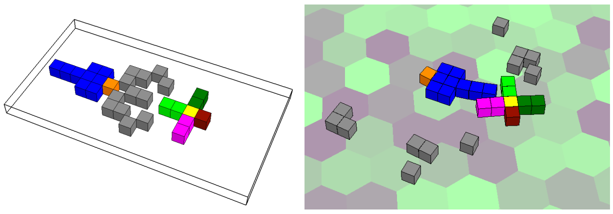


Figure 4.19. Solution D3, voxel coloring as it is used in Figure 4.21 to indicate parts. Grey voxels are not functional.

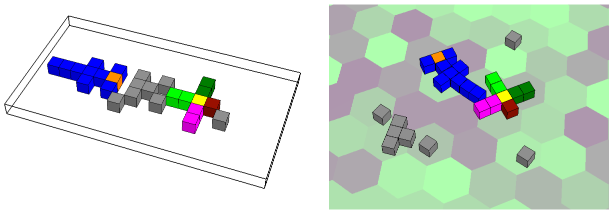


Figure 4.20. Solution D4, voxel coloring as it is used in Figure 4.21 to indicate parts. Grey voxels are not functional.

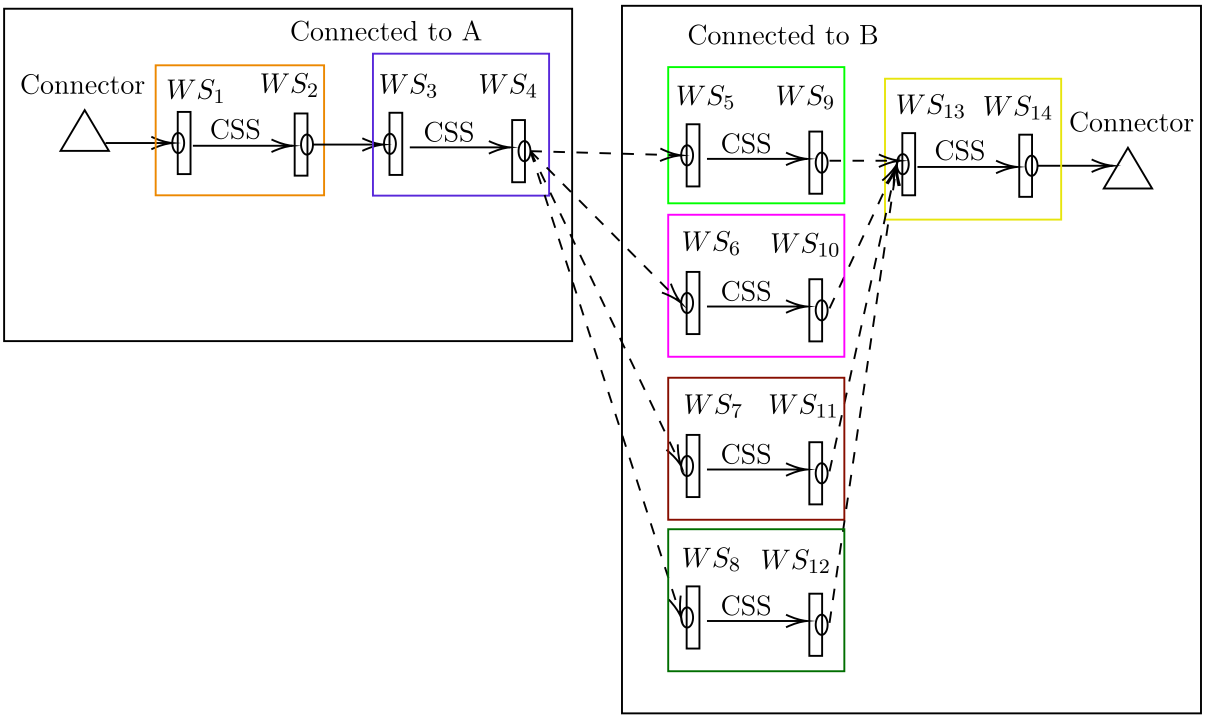


Figure 4.21. Graph analysis using the C&CM for solution D3 and D4. Colors represent the voxels as seen in Figure 4.19 and 4.20. Black boxes separate the parts that are connected to flow input A and flow output B.

Solution D5 is analyzed as well (see Table 4.16, Figure 4.22 and 4.23) which shows that it is a different working principle from the other solutions: an extra part is in place and other connections are present in the graph. Note the connection of WS_6 to WS_{10} that indicates a counter productive flow, this is caused by the collision of the red and blue parts while moving in opposite direction.

WS	Flow domain	Displacement	Effort	Carrier
$WS_{1,14}$	Rotational 1D	Angle	Torque	Angular momentum
WS_{4-10}	Mechanical 3D	Angle, position	Force	Linear momentum (3D)
$WS_{2,3,10B-13}$	Mechanical 3D	Angle, position	Internal Force, Torque	Linear & angular momentum (3D)

Table 4.16. Description of WS flows in Figure 4.23 for solution D5.

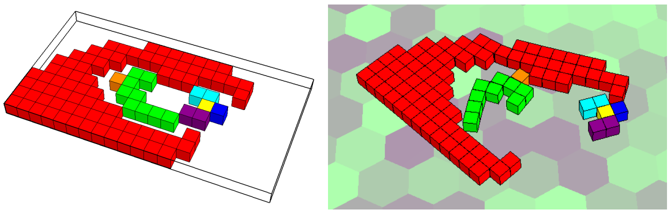


Figure 4.22. Solution D5, voxel coloring as it is used in Figure 4.23 to indicate parts.

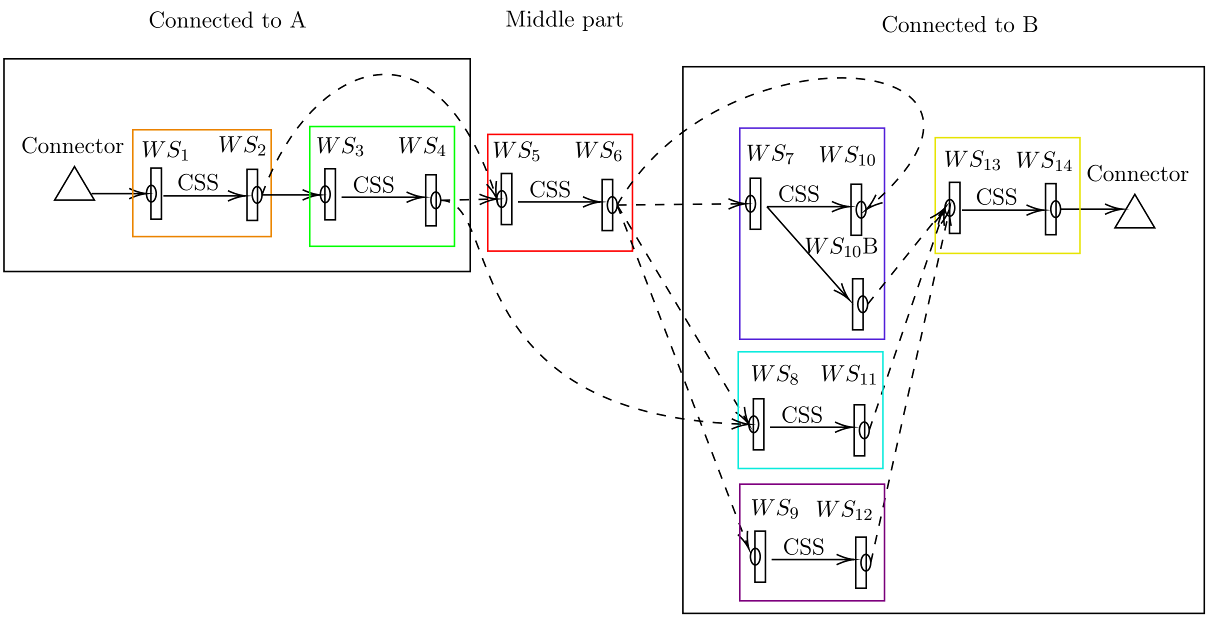


Figure 4.23. Graph analysis using the C&CM for solution D5. Colors represent the voxels as seen in Figure 4.22. Black boxes separate the parts that are connected to flow input A and flow output B.

Even though a large number of generations is evaluated, no new working principle is found after D2, just a different embodiment (D2, D3 and D4) and different details within the same working principle (D3 and D4). Figure 4.15 shows that the increase of fitness gets lower over time. Considering the fact that D5 has a different working principle but has no high fitness, it is hypothesized that the current approach leads to local optima. The increasing age is an indication as well that no new solutions are evolved into new best solutions.
This possibly means that solutions with new working principles but a lower-than-best fitness are discarded from the population without allowing them to evolve to a better fitness. In later runs, other selection methods will be evaluated to get out of these local optima.

4.2.5 Run E

An extra material is added in this run, that has a lower stiffness than the other, previously used one, see Table 4.17. This should allow further bending of voxels to be able to make other movements that were not possible in previous runs. The solution size is increased twofold but to accommodate for extra simulation time, the population size is reduced. The direction of the moment is reversed again, similar to run A, B and C.
The results for this run can be seen in Figure 4.24 where progressing fitness and age of the best individual in the population is shown and Figure 4.25 and 4.26 where a selection of individuals are seen.

Experiment	**E**
Size (x,y,z)	(20,11,2)
Population size	30
Random individuals	6
Number of generations	4500
Initiation time	0.01 s
Simulation time	1 s
Hard material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^3$
Soft material $E(MPa), \rho(\frac{kg}{m^3})$	$5, 1 \cdot 10^3$
A material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
B material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
Location of A	(0.66,0.47,0)
Location of B	(0.31,0.47,0)
Moment at A	-0.5 Nm ↻ ↻
Selection method	Pareto
Computing time (h:m)	378:19

Table 4.17. Parameter values used in run E.

Individual	**E1**	**E2**	**E3**	**E4**	**E5**
Generation	191	834	2810	3737	3955
Fitness (Rotation)	1.161	1.364	1.737	1.652	2.39
Age	58	771	2156	1267	55
No. voxels	114	54	65	67	67

Table 4.18. Five solutions generated in run E.

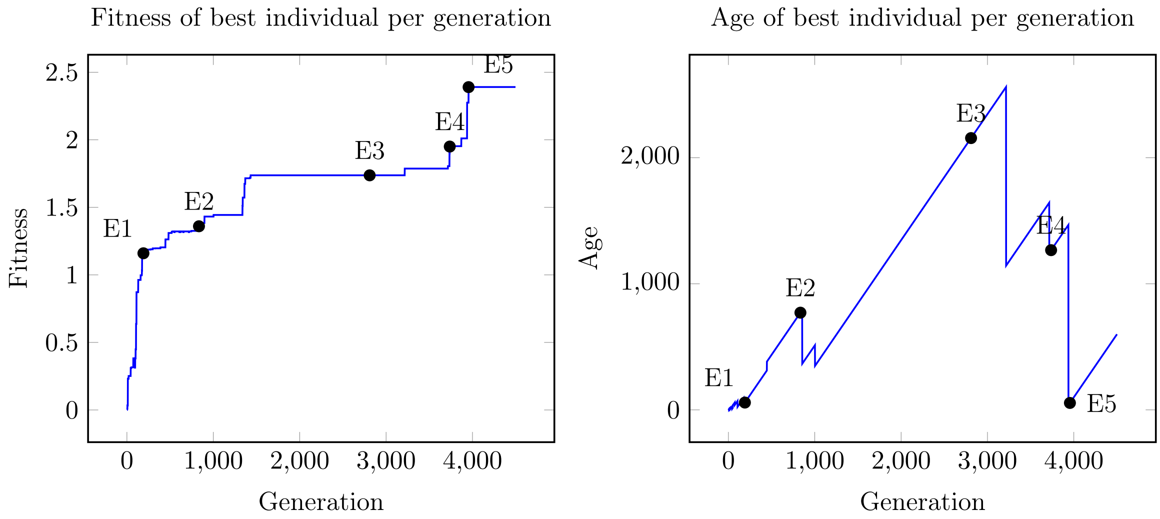


Figure 4.24. Fitness and age of the best individual over the course of generations for run E.

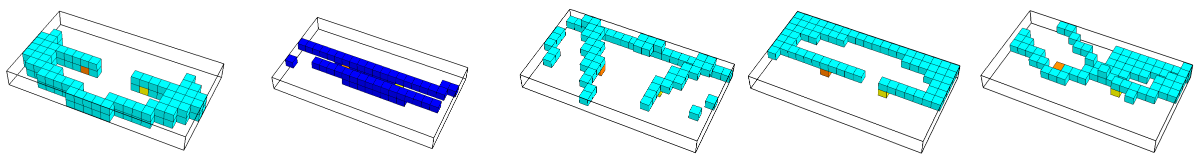


Figure 4.25. Individuals E1 to E5 (left to right).

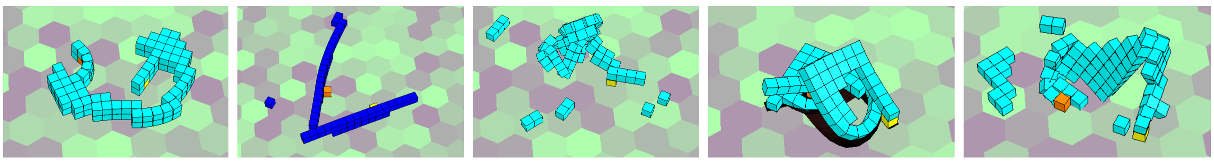


Figure 4.26. Simulations for individuals E1 to E5 (left to right).

Analysis run E

No continuous working principle is found after 5000 generations. However, three different working principles can be distinguished as can be seen below: E1, E2 and E3, E4, E5 each have a different working principle.
The first one is a direct connection from input to output.

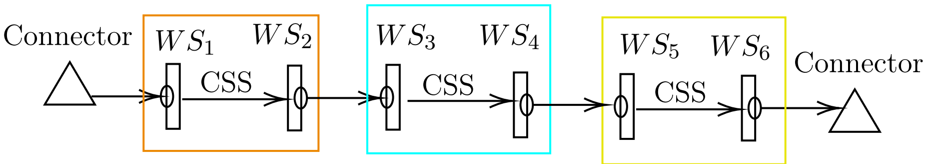


Figure 4.27. Graph analysis using the C&CM for solution E1. Colors represent the voxels as seen in Figure 4.26 .

WS	Flow domain	Displacement	Effort	Carrier
$WS_{1,6}$	Rotational 1D	Angle	Torque	Angular momentum
WS_{3-5}	Mechanical 3D	Angle, position	Internal force, Torque	Linear & angular momentum (3D)

Table 4.19. Description of WS flows in Figure 4.27.

Solution E2 has a part that pushes against a, separate, second part that connects to the output.

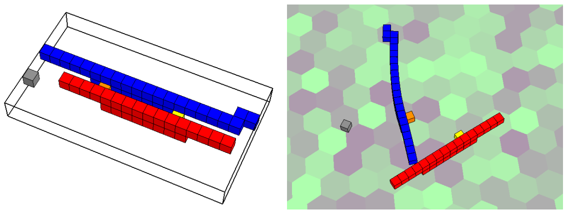


Figure 4.28. Solution E2, voxel coloring as it is used in Figure 4.29 to indicate parts.

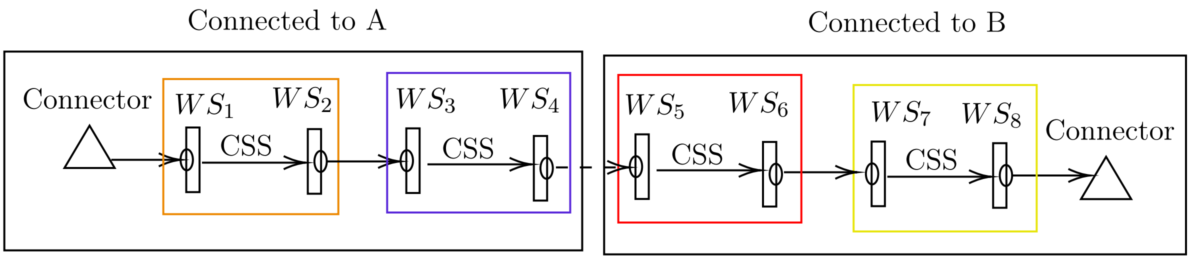


Figure 4.29. Graph analysis using the C&CM for solution E2. Colors represent the voxels as seen in Figure 4.28. Black boxes separate the parts that are connected to flow input A and flow output B.

WS	Flow domain	Displacement	Effort	Carrier
$WS_{1,8}$	Rotational 1D	Angle	Torque	Angular momentum
$WS_{2,3,6,7}$	Mechanical 3D	Angle, position	Force	Linear momentum (3D)
$WS_{4,5}$	Mechanical 3D	Angle, position	Internal Force, Torque	Linear & angular momentum (3D)

Table 4.20. Description of WS flows in Figure 4.29 for solution E2.

Solution E3, E4 and E5 all have one part, just as solution E1, but two additional CSSs are identified that help the artifact to *wind* itself. This analysis for E3 through E5 can be seen in Figure 4.33 and Table 4.21.

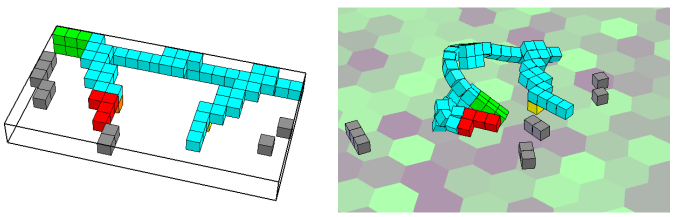


Figure 4.30. Solution E3, voxel coloring as it is used in Figure 4.33 to indicate parts.

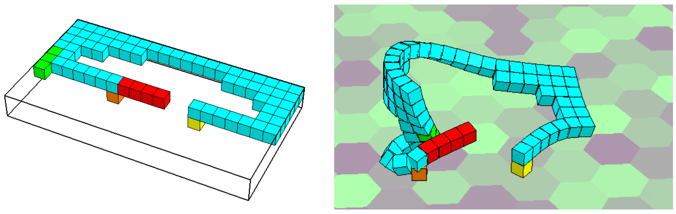


Figure 4.31. Solution E4, voxel coloring as it is used in Figure 4.33 to indicate parts.

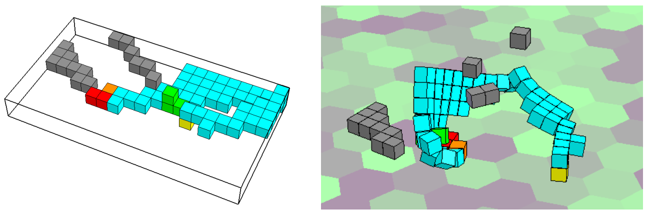


Figure 4.32. Solution E5, voxel coloring as it is used in Figure 4.33 to indicate parts.

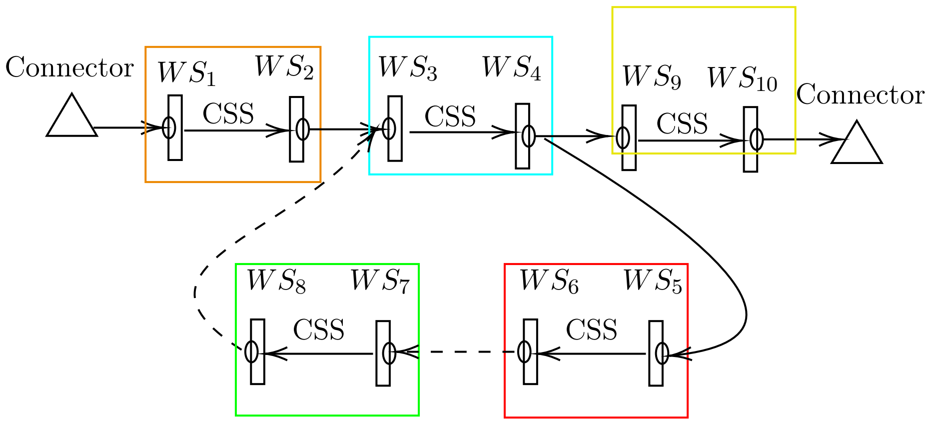


Figure 4.33. Graph analysis using the C&CM for solution E3 through E5. Colors represent the voxels as seen in Figure 4.30 for E3, 4.31 for E4 and 4.32 for E5.

WS	**Flow domain**	**Displacement**	**Effort**	**Carrier**
$WS_{1,10}$	Rotational 1D	Angle	Torque	Angular momentum
$WS_{2-5,8,9}$	Mechanical 3D	Angle, position	Force	Linear momentum (3D)
$WS_{6,7}$	Mechanical 3D	Angle, position	Internal Force, Torque	Linear & angular momentum (3D)

Table 4.21. Description of WS flows in Figure 4.33 for solution E3 through E5.

As seen in Figure 4.24, large (> 1500 generations) periods are seen where no or little (< 1%) increase in fitness is seen, just like in run D. The age of the fittest individual does drop in higher generations which indicates that the best individual at that point has evolved from another individual. This can also be seen upon visual inspection of the solutions (see Figure 4.25): differences in the embodiment of the solutions are present. Other selection methods will be tried to artificially promote this diversity.

A difference in artifact behavior is seen between simulation for similar solutions on the server and desktop. The discrepancy between those indicated a flaw which is caused by the correction of rotation conversion done in run C, which was not implemented on the desktop. To ensure correct simulations in Voxelyze, the correction is undone for later runs: two complete rotations are again defined as $4 \cdot \pi$.

4.2.6 Run F

This run will explore another selection method, one that is adapted from simulated annealing. The algorithm used was introduced in Section 3.3.2. See also Equation 3.2 and 3.1. For this run $b = 6$ and the maximum generation in the simulated annealing calculation is taken at 1000 (note that this does not correspond to the actual maximum generation). The selection probability of the first and last generation can be seen in Figure 4.34.
To speed up the run, next to the maximum time of $1s$, an additional stopping condition is added: when voxel movement per time step is lower than $5 \cdot 10^{-8} mm$, the simulation is stopped as well. Other parameters for this run can be found in Table 4.22.
Five individuals with a high fitness are selected and shown in Figure 4.35 and 4.36 with corresponding properties seen in Table 4.23.

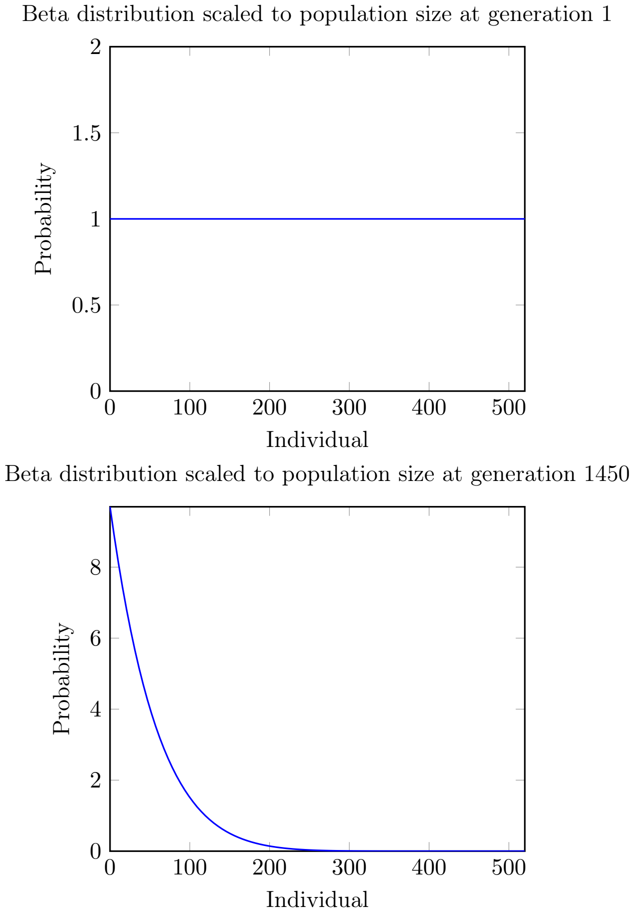


Figure 4.34. Relative probability of an individual to be selected for the next generation at generation 0 (left) and 1450 (right).

Experiment	F
Size (x,y,z)	(20,11,1)
Population size	250
Random individuals	20
Number of generations	1450
Initiation time	0.01 s
Simulation time	1s or no motion
Hard material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^3$
Soft material $E(MPa), \rho(\frac{kg}{m^3})$	-
A material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
B material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
Location of A	(0.66,0.5,0)
Location of B	(0.33,0.5,0)
Moment at A	0.5 Nm ↺ ↻
Selection method	Simulated Annealing
Computing time F_2(h:m)	27:28

Table 4.22. Parameter values used in run F, changes compared to run D highlighted.

Individual	F1	F2	F3	F4	F5
Generation	46	1327	1373	1376	1440
Fitness (Rotation)	1.975	3.812	5.898	7.577	7.445
Age	10	275	321	324	388
No. voxels	40	85	92	108	90

Table 4.23. Five solutions generated in run F_1.

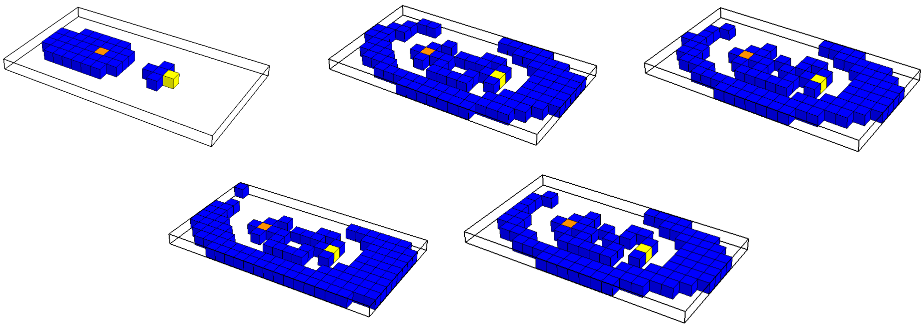


Figure 4.35. Solutions F1 to F5 (left to right).

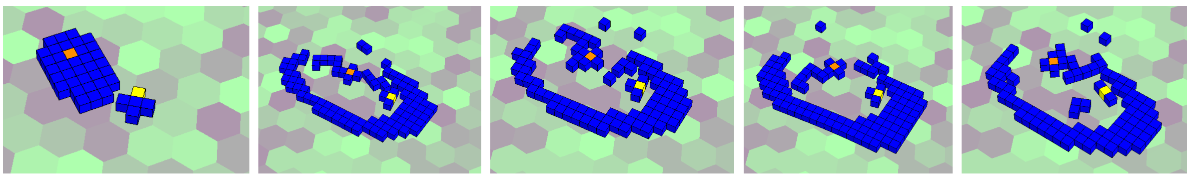


Figure 4.36. Simulations of solutions $F_1$1 to $F_1$5 (left to right).

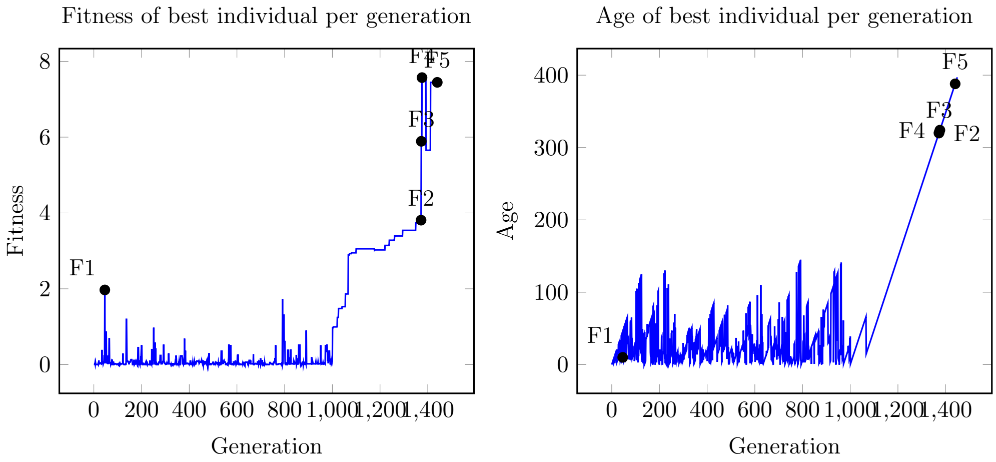


Figure 4.37. Increasing fitness over the course of generations for run F_1.

Analysis run F

No continuous working principle was found in this run. Unlike solution F1, solutions F2 to 5 all have the same working principle as solution D5 (see Figure 4.23 and Table 4.16). The influence of the simulated annealing selection method can be seen in Figure 4.37, a sudden increase in fitness is seen after generation 1000. This can be explained by an error in the calculation of annealing temperature, instead of a float, an integer was used so the annealing temperature was kept constant during generations 0 to 1000. This run is repeated without this error. This run is included for illustrative purposes since a high fitness is reached.

Analysis rerun F_2

Older individuals dominate the higher generations in rerun F_2 as can be seen in Figure 4.38. This possibly indicates that a local optimum was entered in an early generation. In run F_2, the best individual is not better than the one in run F_1. The sudden increase in fitness in that (erroneous) run happens, after 1000 generations of completely random selection, when suddenly a strict selection for higher fitness is performed. This leads to the conclusion that this method of simulated annealing is highly dependent on the state of the population at the moment the annealing temperature increases. In the next run, some parameters are adjusted in an attempt to get a more gradual increase of fitness.

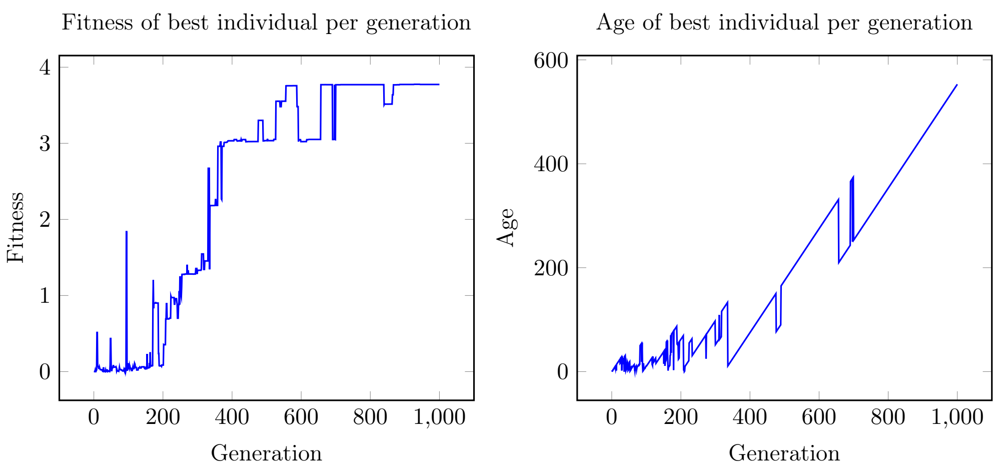


Figure 4.38. Fitness and age of the best individual over the course of generations for run F_2.

4.2.7 Run G

Continuing with the same simulated annealing selection method, run G is performed with a larger influx of random individuals in each generation, see Table 4.24. It is hypothesized that this will result in less reductions to a low fitness of the best individual, because more individuals are deleted from the population and the randomly added individuals usually have a low fitness, so the chance of higher fitness individuals remaining increases. This should result in a more gradual increase of fitness.
In this experiment, the moment is applied in the same direction as the objective rotation, the soft material is added again but, erroneously, with a higher density. The simulation floor is disabled as well to allow solutions to move in more directions. The data about the individuals with a jump in fitness are seen in Table 4.25, their phenotype in Figure 4.40 and the fitness over the course of generations in this run can be seen in Figure 4.39.

Experiment	**G**
Size (x,y,z)	(20,11,2)
Population size	200
Random individuals	200
Number of generations	1000
Initiation time	0.01 s
Simulation time	1s or no motion
Hard material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^3$
Soft material $E(MPa), \rho(\frac{kg}{m^3})$	$5, 1 \cdot 10^6$
A material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
B material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
Location of A	(0.66,0.5,0)
Location of B	(0.33,0.5,0)
Moment at A	-0.5 Nm
Selection method	Simulated Annealing
Computing time (h:m)	60:19

Table 4.24. Parameter values used in run G, changes compared to run F highlighted.

Individual	**G1**	**G2**	**G3**	**G4**	**G5**
Generation	366	580	620	850	1000
Fitness (Rotation)	0.767	1.177	0.286	2.273	1.443
Age	13	20	12	142	292
No. voxels	245	36	166	24	10

Table 4.25. Five solutions generated in run G.

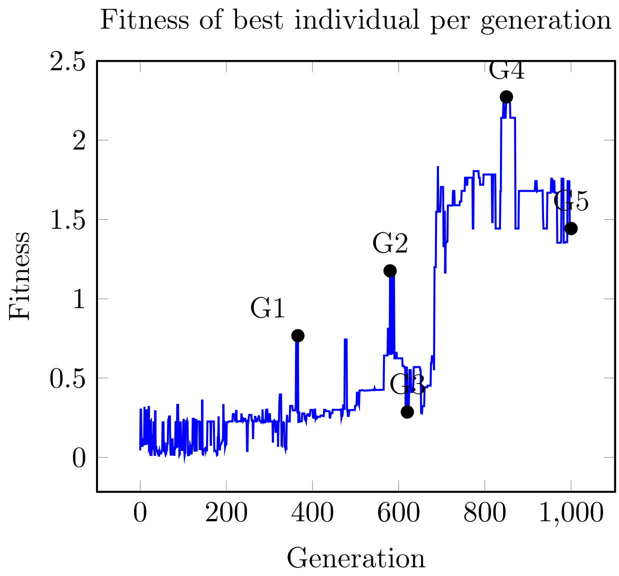


Figure 4.39. Increasing fitness over the course of generations for run G.

Analysis run G

More higher performing individuals are kept in this run which resulted in a more gradual increase of fitness, as hypothesized. Often, still the best individual is removed from the population which can be seen from the significant and often lowering of the fitness over the course of generations. No continuous working principles were found and G2 through G5 all lead to the working principle as analyzed for solution C1 through C4 (see Table 4.10 and Figure 4.12). The high density of the soft material has an influence on the fitness of individuals that consist of soft material. Since the torque is kept the same as in previous runs, a lower rotational velocity can be achieved with the same shape. However, since this run also included the same hard material as previous runs and this could have led to a higher rotational velocity, when used in individuals, because of the lower density, the run is incorporated. It can still be concluded that the usage of this particular implementation of simulated annealing did not result in more different working principles and that individuals with a high fitness are deleted from the population without resulting in a new higher fitness.

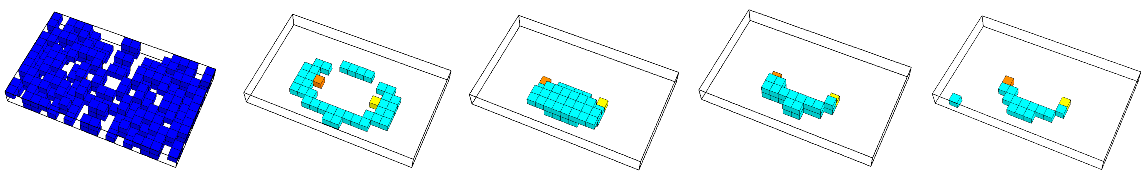


Figure 4.40. Solutions G1 to G5 (left to right).

To conclude the two runs with simulated annealing: it can be said that the usage of this particular implementation of simulated annealing does not lead to more diverse results but closes in on one local optimum at a point with a high annealing temperature is reached since individuals with a high fitness are sometimes deleted from the population. This is not a desirable result so a different selection method is tested in the next run.

4.2.8 Run H

Run H is performed with the same settings as run D and F except a different selection method has been applied: ancestor comparison selection, other settings are shown in Table 4.26. This

was introduced in Section 3.3.2. The ancestry of every individual that is introduced into the new population after the introduction of the best individual, is compared to the ancestry of all individuals. In the case that an individual has more than 50% identical ancestors to more than 10% of the population, that individual is rejected.
It is hypothesized that this selection method does not discard as many individuals as the simulated annealing selection method (runs F and G).
This run was cut short since too many individuals were deleted from the population in generation 190 to continue the run. It was chosen to continue with the next run with another selection method instead of rerunning this run because it was hypothesized that allowing multiple individuals with overlapping ancestry to co-exist is more beneficial to explore the solution space, than to have many individuals with a unique ancestry. The individuals that showed a significant increase in fitness or at a stagnation point (H3) (see Figure 4.41) are shown in Figure 4.42 with their data in Table 4.27.

Experiment	**H**
Size (x,y,z)	(20,11,1)
Population size	250
Random individuals	20
Number of generations	190
Initiation time	0.05 s
Simulation time	1s or no motion
Hard material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^3$
Soft material $E(MPa), \rho(\frac{kg}{m^3})$	-
A material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
B material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
Location of A	(0.66,0.5,0)
Location of B	(0.33,0.5,0)
Moment at A	0.5 Nm ↻ ↺
Selection method	Ancestor comparison
Computing time (h:m)	4:47

Table 4.26. Parameter values used in run H, changes compared to run F highlighted.

Individual	**H1**	**H2**	**H3**	**H4**
Generation	6	15	93	169
Fitness (Rotation)	0.0069	0.5151	1.385	1.669
Age	6	8	68	162
No. voxels	18	18	82	13

Table 4.27. Four solutions generated in run H.

Analysis run H

No continuous or new working principles were found in this run, solution H1 has the same working principle as solution D1 (see Table 4.13 and Figure 4.16) and solution H2 through H4 have the same working principle as solution D3 and D4 (see Table 4.15 and Figure 4.21), but a different embodiment because of a difference in repetition of WSPs.

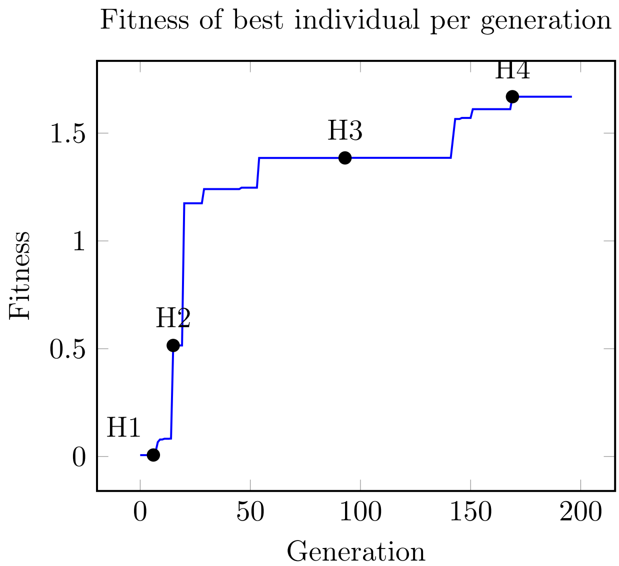


Figure 4.41. Increasing fitness over the course of generations for run H.

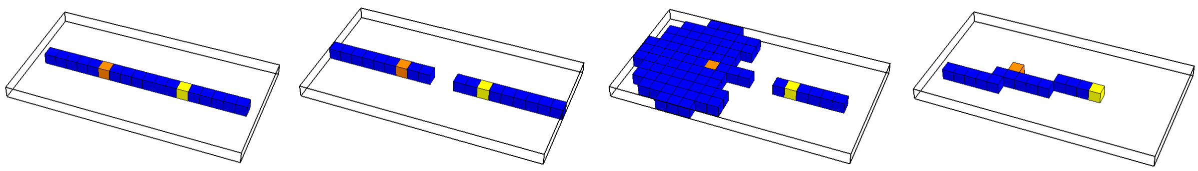


Figure 4.42. Solutions H1 to H4 (left to right).

To analyze the implications of the three selection methods so far, the composition of a single generation is analyzed. Figure 4.43 shows three histograms of the age distribution of individuals in generation 190 for run D, F and H. Generation 190 was chosen since this is the last generation of run H.

Run F shows many young ($0 < age < 38$) individuals which is explained by the low virtual temperature at that generation (190 of 1000). Young individuals tend to have a lower fitness but, due to their random creation and thus their independence from older individuals, provide a possibility to get out of a local optimum. However, it was already concluded that this particular implementation of simulated annealing does not provide the best selection method.

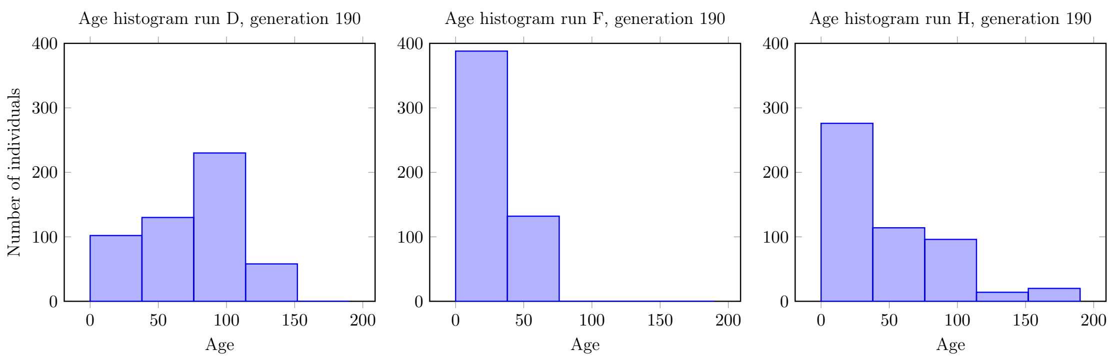


Figure 4.43. Histograms of the age distribution of every individual in the population at generation 190 for pareto selection (run D, left), annealing selection (run F, middle) and ancestor deletion selection (run H, right).

The histogram of the last run, run H, has less young individuals than run F, but more than run D. The hypothesis is that this will enhance variety among solutions which will result in a better chance of avoiding local optima as long as the correct individuals get the opportunity to evolve to a high fitness. It is not possible to determine these individuals beforehand but it is hypothesized that, unlike in the ancestor comparison selection method, niches of similar individuals should exist in the population to allow the exploration of a local optimum instead of just one individual. This will be tested in run I with a combined pareto and ancestor deletion selection method. Please note that no comparison with run G is done because the number of randomly added individuals per generation is different.

4.2.9 Run I

In previous runs, no selection method was found that sufficiently identifies a local optimum and abandons it after exploring. To solve this, a new selection method is applied in this run which is a combination of a pareto selection and the ancestor comparison method (used in run H). Pareto selection is used as long as there is an increase in fitness. A reset will be performed at the moment that the sum of the relative change in fitness over the last 300 generations is less than 0.5% (see Equation 4.1). This reset deletes the best individual and any close relatives: individuals that have more than 70% of their ancestors in common.

$$\sum_{gen-300}^{gen} \frac{fitness_{gen} - fitness_{gen-1}}{fitness_{gen-1}} < 0.005 \tag{4.1}$$

Parameters for run I (see Table 4.28) are taken so that artifacts have a larger space (similar to run E) and many generations are allowed to be able to analyze the influence of the introduced resets. The population size is lowered to conserve computing time. Fitness and age of the best individual in each generation is shown in Figure 4.44, data of the individuals with the highest fitness are shown in Table 4.29.

Experiment	**I**
Size (x,y,z)	(20,11,2)
Population size	100
Random individuals	20
Number of generations	9999
Initiation time	0.01 s
Simulation time	1 s
Hard material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^3$
Soft material $E(MPa), \rho(\frac{kg}{m^3})$	$5, 1 \cdot 10^3$
A material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
B material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
Location of A	(0.66,0.47,0)
Location of B	(0.31,0.47,0)
Moment at A	0.5 Nm $\circlearrowleft$ $\circlearrowright$
Selection method	Pareto reset
Computing time (h:m)	232:47

Table 4.28. Parameter values used in run I

Individual	I1	I2	I3	I4	I5
Generation	433	2310	4676	7623	9995
Fitness (Rotation)	105.97	14.86	10.038	70.565	8.4907
Age	287	894	1124	1158	535
No. voxels	59	48	48	17	77

Table 4.29. Five solutions generated in run I.

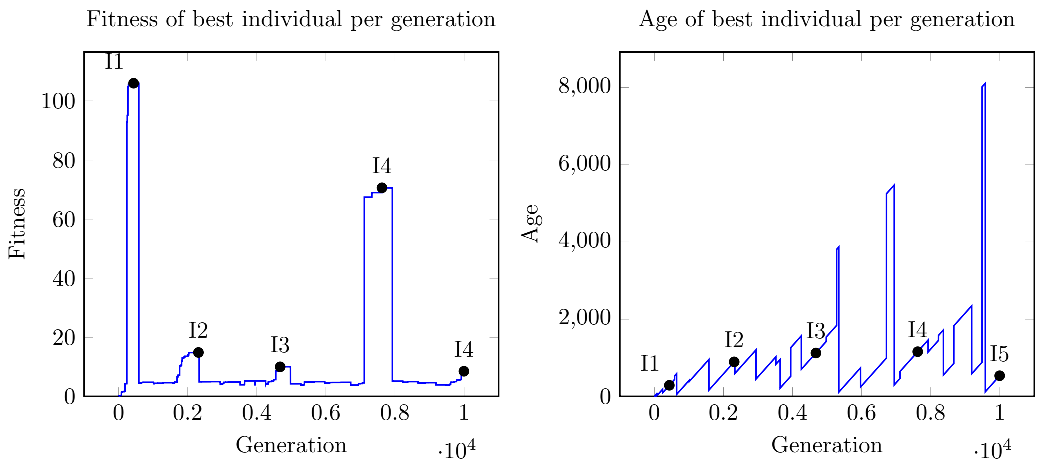


Figure 4.44. Fitness and age of the best individual over the course of generations for run I.

Analysis run I

This run resulted in multiple solutions with a very high fitness. These can be seen in Figure 4.45 and 4.46.

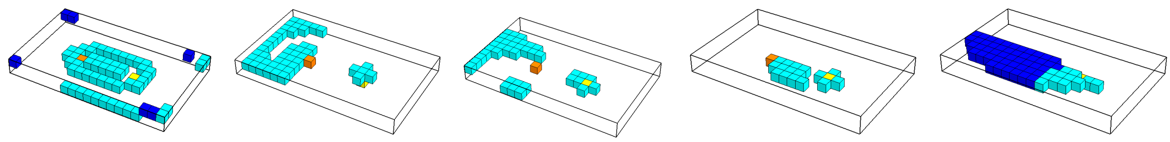


Figure 4.45. Solutions I1 to I5 (left to right).

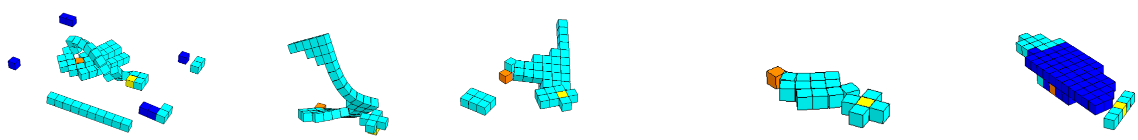


Figure 4.46. Simulations of solutions I1 to I5 (left to right).

First the influence of the new pareto reset selection method will be analyzed. As can be seen in Figure 4.49, the resets (indicated with dotted lines) bring down the fitness and age of the best individual after 300 generations of no increase in fitness. This seems to have a positive effect on the variety of the solutions, as can be seen in Figure 4.45 and 4.46. The individuals that were removed from the population in generation 584 are shown in Figure 4.47. This reset was performed because a stagnation in increase of fitness was measured since generation 284, 300 generations earlier. The best performing individual (I1) was removed, along with all individuals that have more than 70% similar ancestors. Various similar artifacts can be seen as individuals compared to individual I1 but some do not share visible similarities.

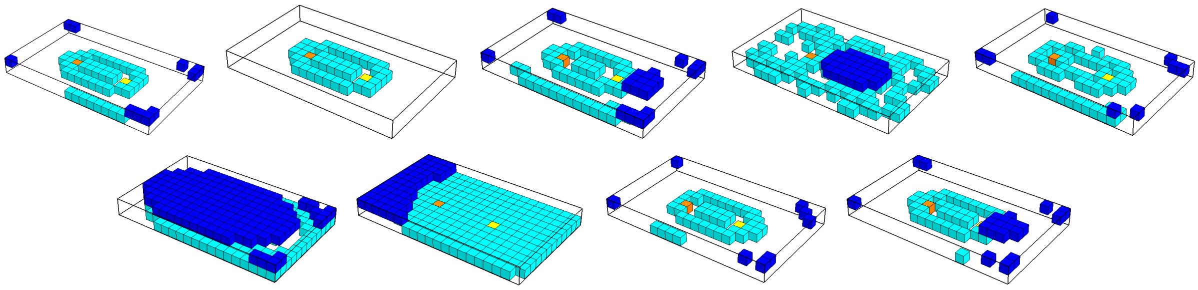


Figure 4.47. Individuals that were removed in the reset after generation 584 with more then 70% overlap in ancestors with individual I1. Fitness (top left to bottom right) 105.5, 2.532, 1.034, 0.3530, 0.04426, 0.005832, -0.00074, -0.4743, -0.7155

Since the CPPN determines the shape of the artifacts, it is not possible to determine beforehand which percentage of overlap of ancestors results in visually comparable artifacts. It is confirmed that the next best individual does have a different shape, as is shown in Figure 4.48. For now, these reset parameters are kept.

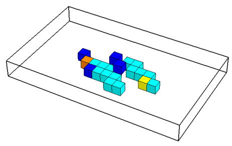


Figure 4.48. The next best individual (with fitness 4.630) after the reset in generation 584.

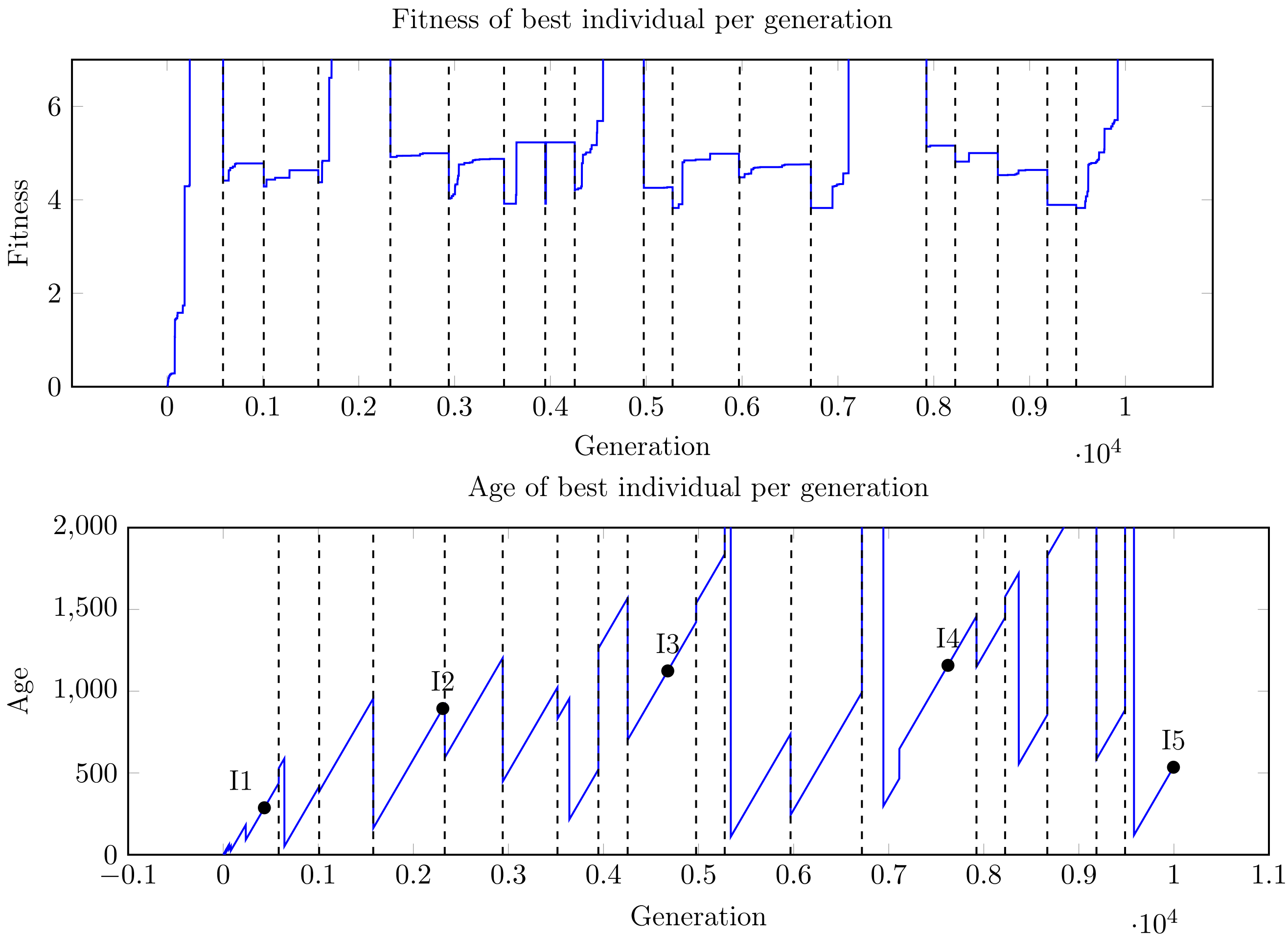


Figure 4.49. Age and fitness of best individual over the course of generations for run I, dotted lines indicate a reset. The fitness and age scale are capped to show the details in lower values.

All the individuals with a high fitness are now analyzed. Solution I1 has a working principle that is not seen before, as analyzed in Figure 4.51 and Table 4.30. It works continuously but closer inspection shows that even though this solution is feasible in the simulation, this solution could not work in real life. The blue part passes line A and B on both sides of the voxel (yellow and orange) which prohibits any physical connection of input and output flow to an external system. Only some kind of magnetic mounting might make this feasible.

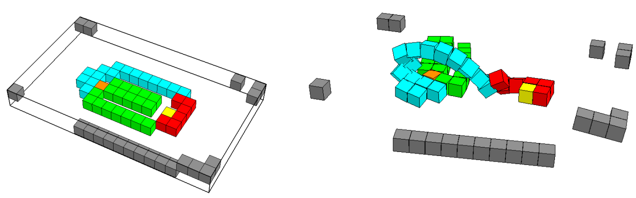


Figure 4.50. Solution I1, voxel coloring as it is used in Figure 4.51 to indicate parts.

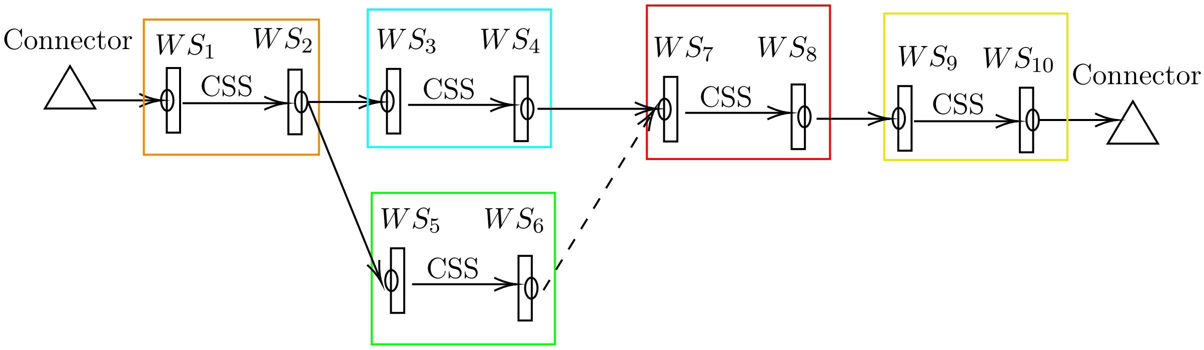


Figure 4.51. Graph analysis using the C&CM for solution I1. Colors represent the voxels as seen in Figure 4.50 for I1

WS	Flow domain	Displacement	Effort	Carrier
$WS_{1,10}$	Rotational 1D	Angle	Torque	Angular momentum
$WS_{2-5,8,9}$	Mechanical 3D	angle, position	Force	Linear momentum (3D)
$WS_{6,7}$	Mechanical 3D	Angle, position	Internal Force, Torque	Linear & angular momentum (3D)

Table 4.30. Description of WS flows in Figure 4.51 for solution I1.

Solution I2, I3 and I4 (Figure 4.52, 4.53 and 4.54, respectively) have the same working principle and also work continuously, as seen in the analysis in Figure 4.55 and Table 4.31. Only a difference in embodiment is seen because there is a different number of repetitions of a number of WSs.

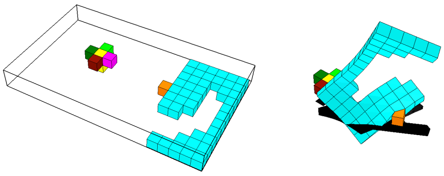


Figure 4.52. Solution I2, voxel coloring as it is used in Figure 4.55 to indicate parts.

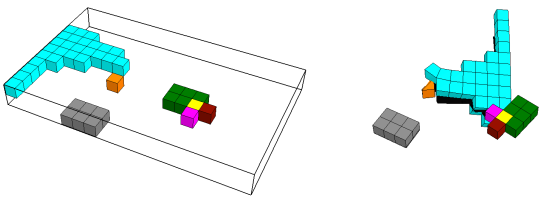


Figure 4.53. Solution I3, voxel coloring as it is used in Figure 4.55 to indicate parts.

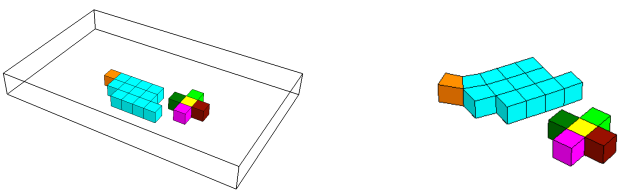


Figure 4.54. Solution I4, voxel coloring as it is used in Figure 4.55 to indicate parts.

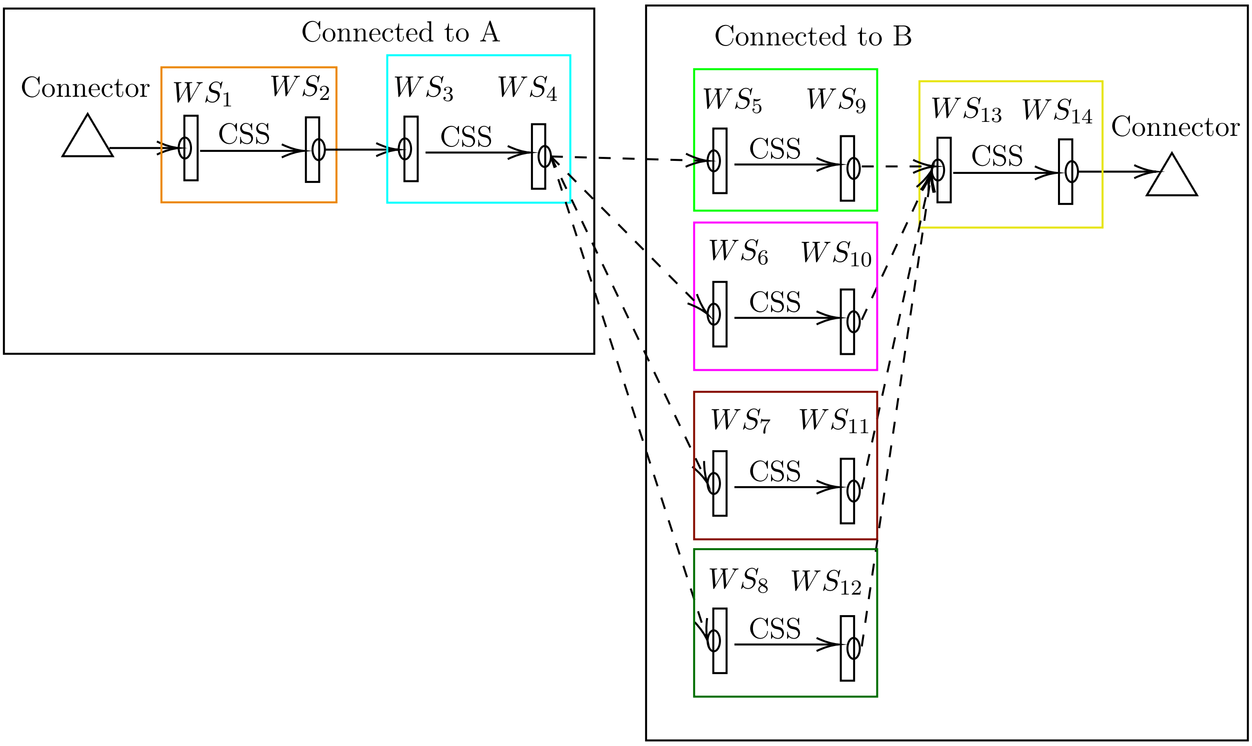


Figure 4.55. Graph analysis using the C&CM for solution I2,3 and 4. Colors represent the voxels as seen in Figure 4.52 for I2, Figure 4.53 for I3 and Figure 4.54 for I4.

WS	**Flow domain**	**Displacement**	**Effort**	**Carrier**
$WS_{1,10}$	Rotational 1D	Angle	Torque	Angular momentum
$WS_{2-5,8,9}$	Mechanical 3D	Angle, position	Force	Linear momentum (3D)
$WS_{6,7}$	Mechanical 3D	Angle, position	Internal Force, Torque	Linear & angular momentum (3D)

Table 4.31. Description of WS flows in Figure 4.55 for solution I2, 3 and 4. WS_5 and WS_9 are not present for solution I3. Black boxes separate the parts that are connected to flow input A and flow output B.

The working principle of solution I5 (see Figure 4.56) seems to be similar to that of the previous solutions, but the difference in stiffness causes one end (dark blue) to make direct contact with the Connected to B, and the other end (light blue) to slightly bend and convey its momentum using a friction force. This analysis can be seen in Figure 4.57 and Table 4.32.

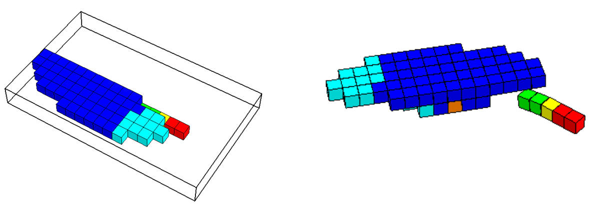


Figure 4.56. Solution I5, voxel coloring as it is used in Figure 4.57 to indicate parts.

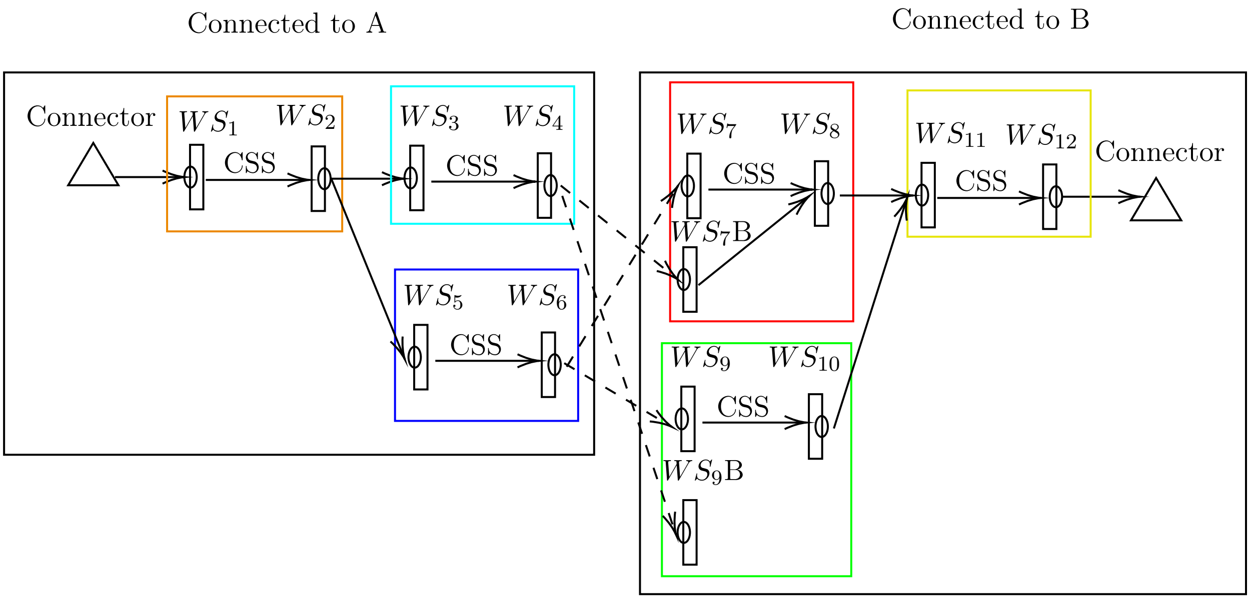


Figure 4.57. Graph analysis using the C&CM for solution I5. Colors represent the voxels as seen in Figure 4.56 for I5. Black boxes separate the parts that are connected to flow input A and flow output B.

WS	**Flow domain**	**Displacement**	**Effort**	**Carrier**
$WS_{1,12}$	Rotational 1D	Angle	Torque	Angular momentum
$WS_{6,7,9}$	Mechanical 3D	Angle, position	Force	Linear momentum (3D)
$WS_{4,7B,9B}$	Translational 1D	Position	Friction force	Linear momentum
$WS_{2,3,5,8,10,11}$	Mechanical 3D	Angle, position	Internal Force, Torque	Linear & angular momentum (3D)

Table 4.32. Description of WS flows in Figure 4.57 for solution I5.

It can be said that this selection method, using these parameters, does lead to solutions that meet the verification demands set in Section 3.3.1: Three distinct working principles were found that work continuously.

4.2.10 Run J

For better insight in current PoC for the CAI-tool, run I is repeated with parameters that make the simulation faster to accommodate ten simultaneous runs (see Table 4.33) with a different seed for any random numbers generated in the run.
The individual size as well as population size is reduced for a faster simulation. In earlier runs, collision of voxels is inaccurate: collision is detected before a contact can be seen in the simulation GUI. Therefore, a different collision horizon is used (see Section 3.3.2), 1 instead of 2. The floor is enabled again and the introduced axles at A (orange) and B (yellow) are lengthened with one voxel on each side, to prevent results such as solution I1 the shape of these voxels can be seen in Figure 4.58. The individuals with the highest fitness are shown in Figure 4.59 and 4.60, their data can be seen in Table 4.33.

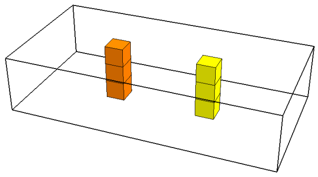


Figure 4.58. Voxels that are introduced at A (orange) and B (yellow).

Experiment	**J0-J9**
Size (x,y,z)	(14,7,1+2)
Population size	19
Random individuals	1
Number of generations	9999
Initiation time	0.1 s
Simulation time	1 s
Hard material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^3$
Soft material $E(MPa), \rho(\frac{kg}{m^3})$	$5, 1 \cdot 10^3$
A material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
B material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
Location of A	(0.66,0.47,0)
Location of B	(0.31,0.47,0)
Moment at A	0.5 Nm ↺ ↻
Selection method	Pareto reset
Computing time (h:m)	108:47-131:29

Table 4.33. Parameter values used in run J

Individual	**J0-1**	**J2-1**	**J4-1**	**J5-1**	**J5-2**	**J7-1**	**J9-1**
Generation	1648	5224	6492	1735	1741	8034	5080
Fitness (Rotation)	55.91	70.83	84.82	7.512	13.53	63.72	55.86
Age	1309	1000	876	1634	1640	932	650
No. voxels	18	16	23	18	18	13	16

Table 4.34. Five solutions generated in run J.

Analysis runs J

In this run, not many continuously working solutions were found as can be seen from the low number of significant increases in fitness in Figure 4.61. Although a different embodiment is seen, all have the same working principle, which is the same as solution D2 (see Figure 4.18 and Table 4.14). In some solutions (J0-1, J5-1 and J5-2), it appears that friction is used in transferring the flow between some WSs but a closer inspection shows that this is not the case and only the direct force causes this, similar to the other solutions. The reduced collision horizon does not lead to a more accurate simulation visually but simulation time is increased. The collision horizon is reverted to the original value of 2 for further experiments.
It is hypothesized that the small individual size results in less continuously working solutions and the small population size makes it harder to explore the solution space to find these solutions.

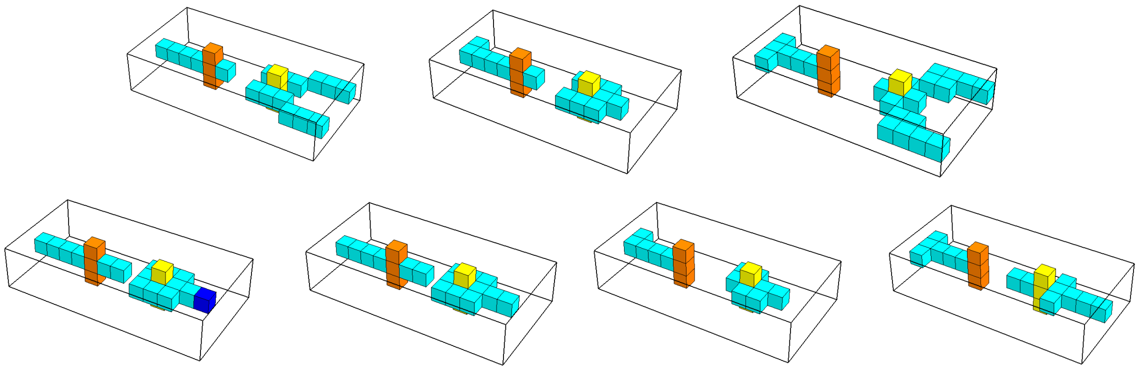


Figure 4.59. Solutions J0-1 to J9-1 (top left to bottom right).

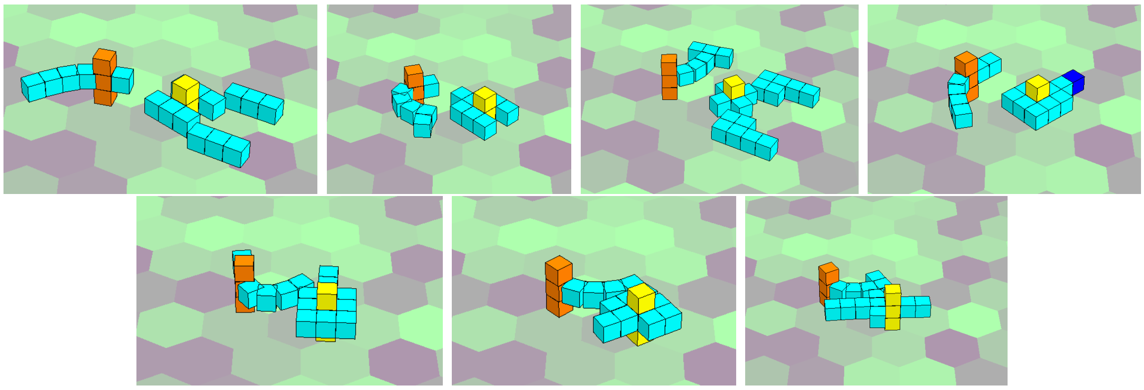


Figure 4.60. Simulations for solutions J0-1 to J9-1 (top left to bottom right).

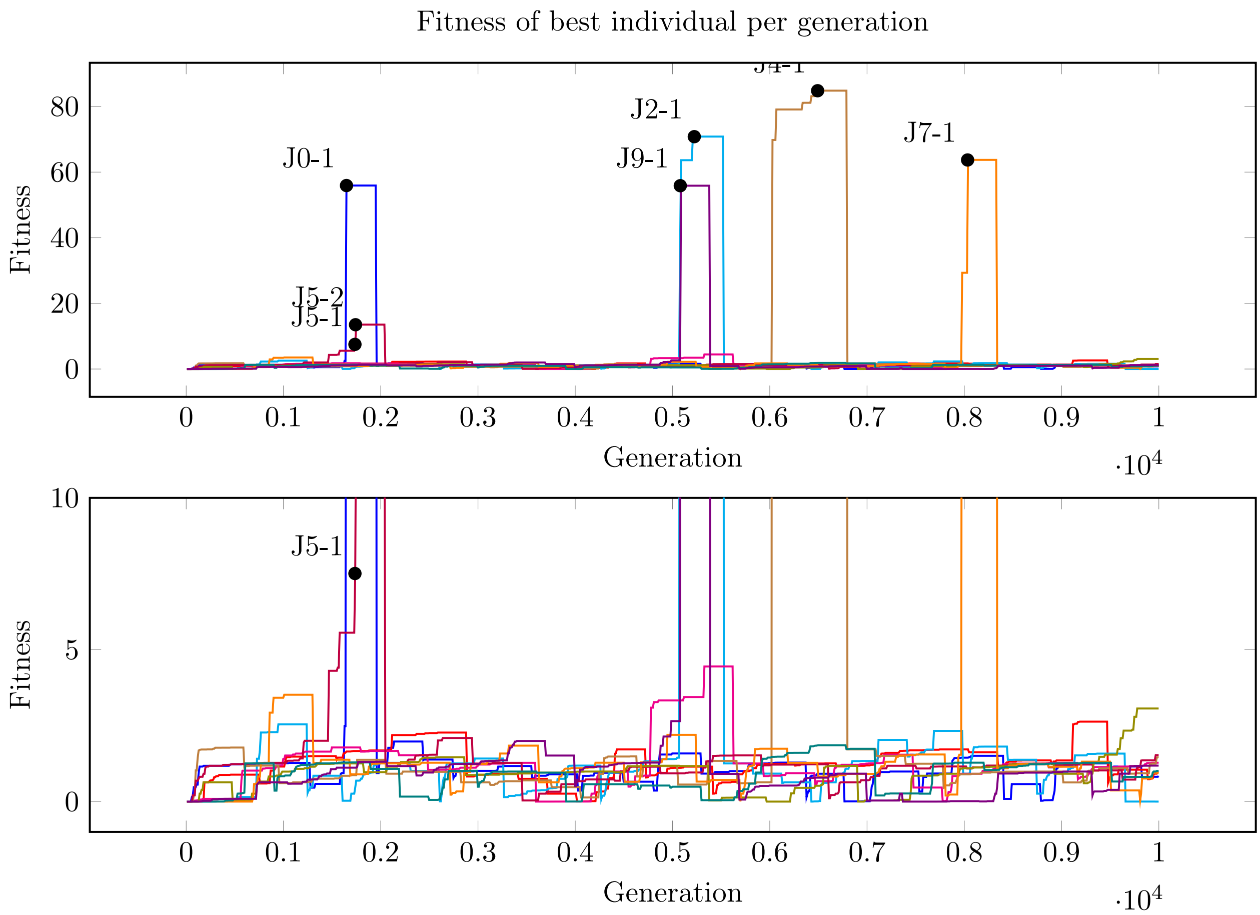


Figure 4.61. Increasing fitness over the course of generations for run J0, run J1, run J2, run J3, run J4, run J5, run J6, run J7, run J8 and run J9. The bottom graph is zoomed in to provide a more detailed view of lower fitness.

4.2.11 Run K

In this run, gravity was added to the environment, perpendicular to the rotation axes. The settings of run I were used except for a reversed moment and no extended axles, also the number of generations is lower. See Table 4.35 for these settings.

Experiment	**K**
Size (x,y,z)	(14,7,1)
Population size	100
Random individuals	20
Number of generations	1500
Initiation time	0.1 s
Simulation time	1 s
Hard material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^3$
Soft material $E(MPa), \rho(\frac{kg}{m^3})$	$5, 1 \cdot 10^3$
A material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
B material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$
Location of A	(0.66,0.47,0)
Location of B	(0.31,0.47,0)
Moment at A	0.5 Nm ↺ ↻
Selection method	Pareto reset
Computing time (h:m)	118:35

Table 4.35. Parameter values used in run K

Analysis runs K

Solutions in early generations make use of gravity to rotate, but this does not lead to any continuous working principle (K1). Continuous working solutions (K2-4) show the same working principle as I1. Solutions can be seen in Figure 4.62 and 4.63, with their values seen in Table 4.36.
This run was repeated but with a floor, in that run no continuous working solutions were found after two runs and this path was abandoned. It should be pointed out that no continuously working solutions could be designed manually either by the author for this situation.

Individual	**K1**	**K2**	**K3**	**K4**
Generation	498	511	1236	1420
Fitness (Rotation)	3.572	50.84	272.2	327.7
Age	93	37	762	946
No. voxels	42	101	71	48

Table 4.36. Four solutions generated in run K.

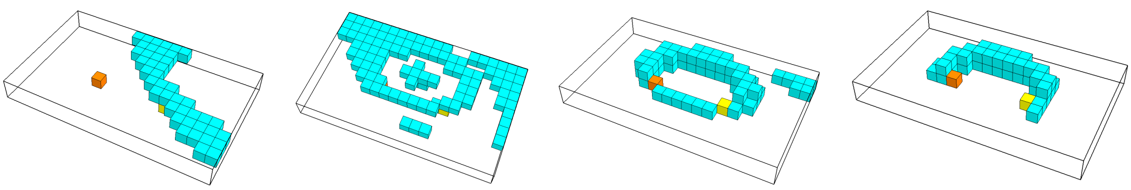


Figure 4.62. Solutions K1 to K4 (left to right).

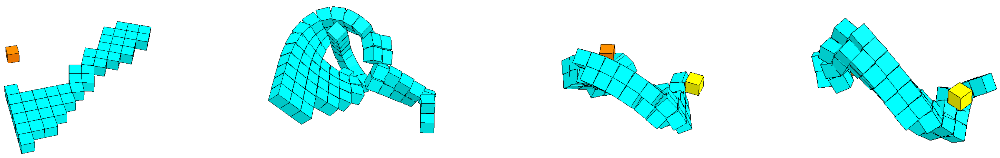


Figure 4.63. Simulations of solutions K1 to K4 (left to right).

4.2.12 Runs L and M

The inputs for the CPPN in these runs were changed. Instead of x, y, z and d, distance to centre and b, a bias, the distance to centre is replaced by two inputs that are defined by their distance to flow input location A (d_a) and flow output location B (d_b). This does provide information about the mechanical engineering problem to the optimization itself, but still no working principle is introduced. It is hypothesized that this changed CPPN structure leads to faster results since the flow locations are now known, but that ultimately a lower number of different working principles will be found since the CPPN is more inclined to certain shapes. Run L has a height of 3 voxels and run M a height of 2 voxels. Both runs were performed twice with different seeds. Other parameters are shown in Table 4.37. Fitness and age of the best individual over the course of generations for these runs can be seen in Figure 4.64, data of the individuals with the highest fitness is seen in Table 4.38 and their phenotype in Figure 4.65 and 4.66.

Experiment	**L1,L2**	**M1,M2**
Size (x,y,z)	(20,10,3+2)	(20,10,2+2)
Population size	50	50
Random individuals	5	5
Number of generations	11000	15000
Initiation time	0.01 s	0.01 s
Simulation time	3 s	3 s
Hard material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^3$	$50, 1 \cdot 10^3$
Soft material $E(MPa), \rho(\frac{kg}{m^3})$	$5, 1 \cdot 10^3$	$5, 1 \cdot 10^3$
A material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$
B material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$
Location of A	(0.66,0.5,0)	(0.66,0.5,0)
Location of B	(0.33,0.5,0)	(0.33,0.5,0)
Moment at A	0.5 Nm $\circlearrowright$ $\circlearrowleft$	0.5 Nm $\circlearrowright$ $\circlearrowleft$
Selection method	Pareto reset	Pareto reset
Computing time (h:m)	367:44-376:48	336:34-367:09

Table 4.37. Parameter values used in run L and M

Individual	**L1-1**	**L1-2**	**M1-1**	**M1-2**
Generation	3550	10134	5869	6225
Fitness (Rotation)	36.15	13.58	14.77	13.53
Age	270	678	644	643
No. voxels	54	112	130	93

Table 4.38. Four solutions generated in run L and M.

Analysis runs L and M

Not many continuously working solutions were found in either runs. Only run M1 resulted in two different continuously working solutions.

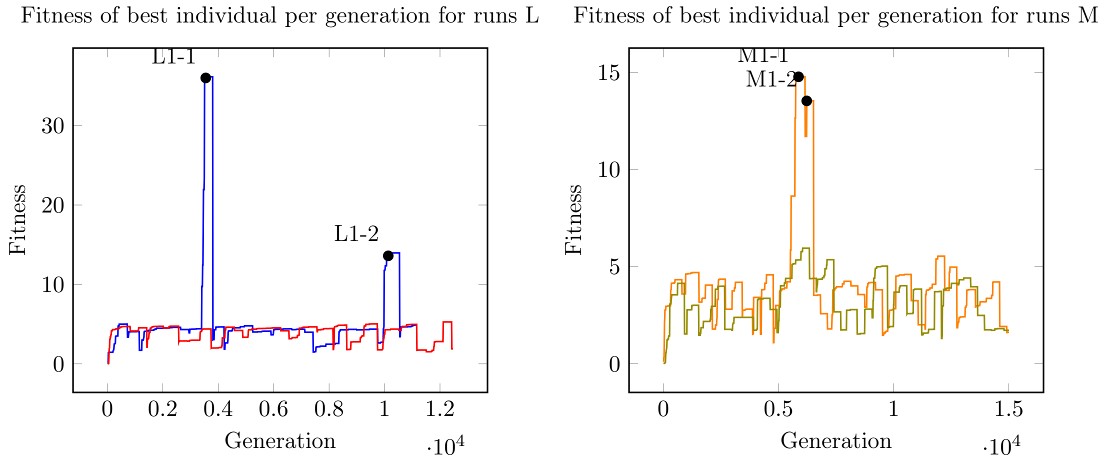


Figure 4.64. Increasing fitness over the course of generations for run L1, run L2 (left) and run M1 and run M2 (right)

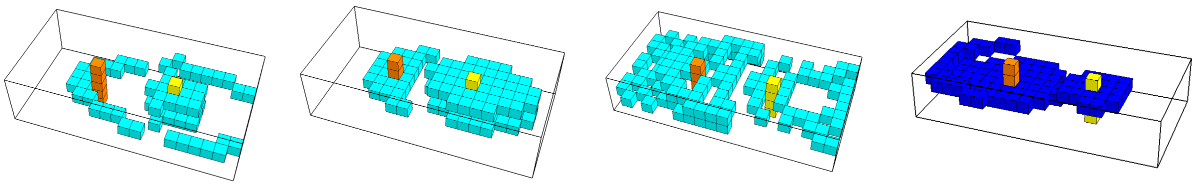


Figure 4.65. Solutions L1-1, L1-2, M1-1 and M1-2 (left to right).

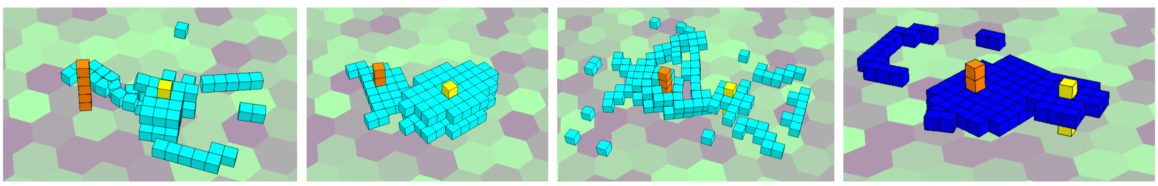


Figure 4.66. Simulations of solutions L1-1, L1-2, M1-1 and M1-2 (left to right).

Solution L1-1 and L1-2 seem to make use of a similar working principle. However, L1-1 has an extra CSS that stores energy as can be seen in Figure 4.68 and Table 4.39 and 4.40, coloration indicated in Figure 4.67. Solution L1-2 is analyzed in Figure 4.70 with Tables 4.41 and 4.42 using the coloration as shown in Figure 4.69. According to the definition, the solutions have a different working principle.
Both solutions make use of some sort of clamping force which is stored by CSS_d and CSS_e in solution L1-1 and in solution L1-2 by CSS_c and CSS_d. This causes a force to be exerted on the Connected to A in both situations which slows it down, but it is beneficial for the working principle as a whole since the caused friction makes the Connected to B rotate.

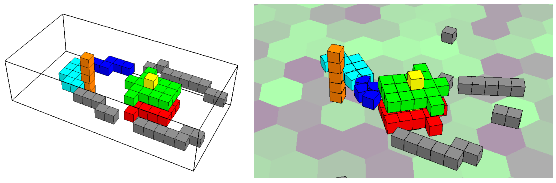


Figure 4.67. Coloration as used in Figure 4.68 for the analysis of solution L1-1.

WS	**Flow domain**	**Displacement**	**Effort**	**Carrier**
$WS_{1,12}$	Rotational 1D	Angle	Torque	Angular momentum
$WS_{6,7,9}$	Mechanical 3D	Angle, position	Force	Linear momentum (3D)
$WS_{2-5,8,10,11}$	Mechanical 3D	Angle, position	Internal Force, Torque	Linear & angular momentum (3D)

Table 4.39. Description of WS flows in Figure 4.68 for solution L1-1.

CSS	**Flow domain**	**Energy storage**
$CSS_{c,d,e}$	Mechanical 3D	Potential energy

Table 4.40. Description of CSSs in Figure 4.68 for solution L1-1.

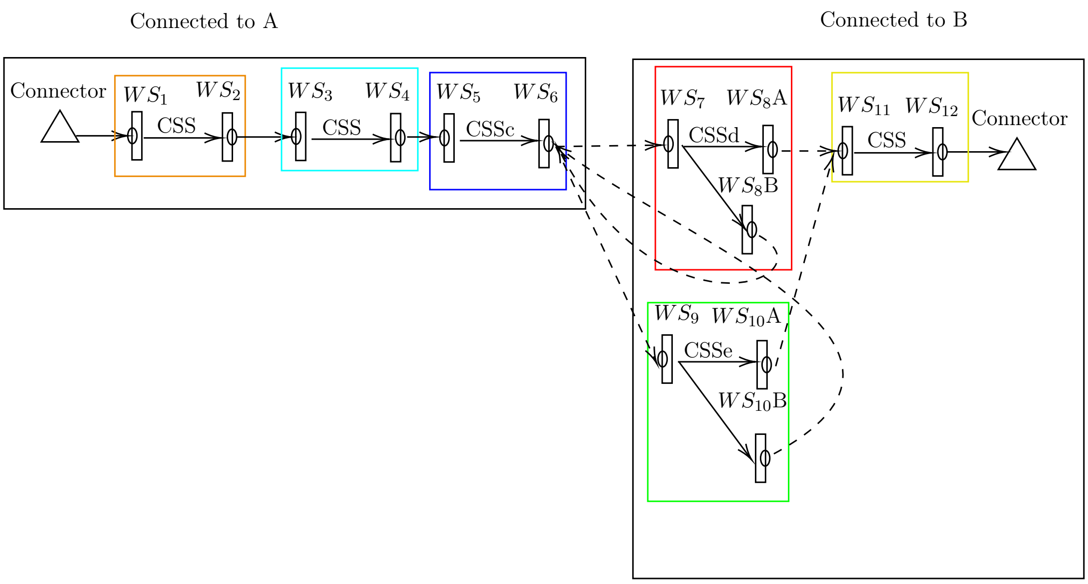


Figure 4.68. Graph analysis using the C&CM for solution L1-1. Colors represent the voxels as seen in Figure 4.67.

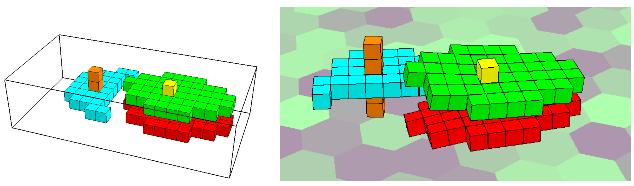


Figure 4.69. Coloration as used in Figure 4.70 for the analysis of solution L1-2.

WS	Flow domain	Displacement	Effort	Carrier
$WS_{1,10}$	Rotational 1D	Angle	Torque	Angular momentum
WS_{4-6}	Mechanical 3D	Angle, position	Force	Linear momentum (3D)
$WS_{2,3,7-9}$	Mechanical 3D	Angle, position	Internal Force, Torque	Linear & angular momentum (3D)

Table 4.41. Description of WS flows in Figure 4.70 for solution L1-2.

CSS	Flow domain	Energy storage
$CSS_{c,d}$	Mechanical 3D	Potential energy

Table 4.42. Description of CSSs in Figure 4.70 for solution L1-2.

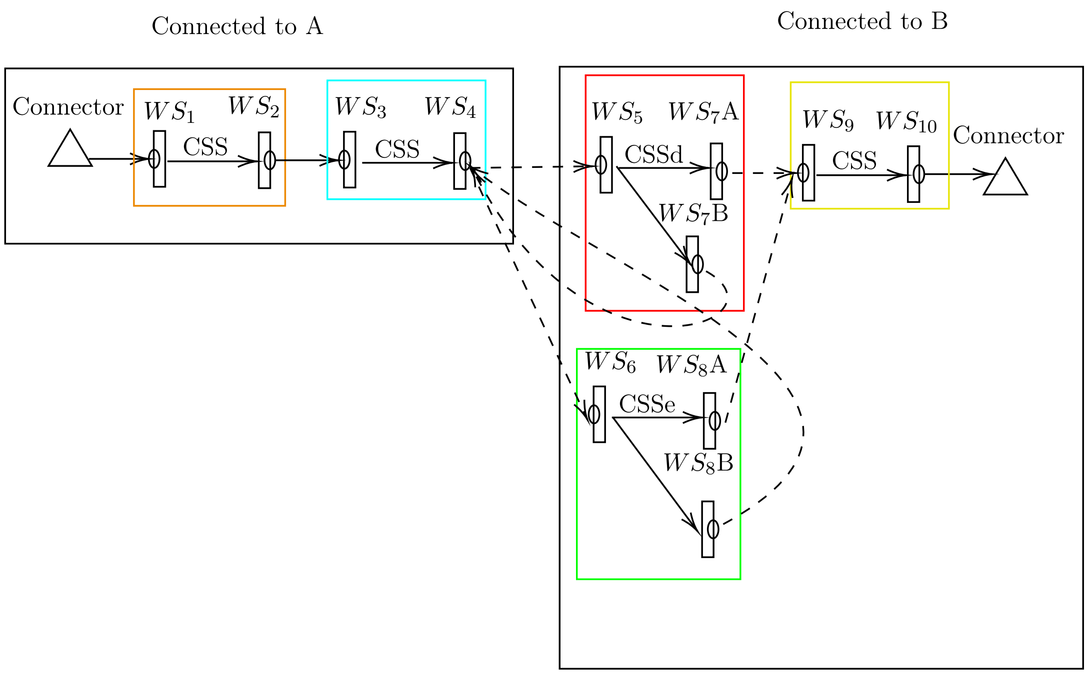


Figure 4.70. Graph analysis using the C&CM for solution L1-2. Colors represent the voxels as seen in Figure 4.69. Black boxes separate the parts that are connected to flow input A and flow output B.

Solutions M1-1 and M1-2 show a different working principle on first sight but a closer inspection learns that the difference is quite small. The analysis for both is shown in Figure 4.73 using coloring indicated in Figures 4.71 and 4.72, Tables belonging to these analyses are 4.43 and 4.44. The only difference is the storage of some energy in the Connected to A (CSSb) in solution M2-1. Taking into account this difference, according to the definition a different working principle is seen for both solutions.

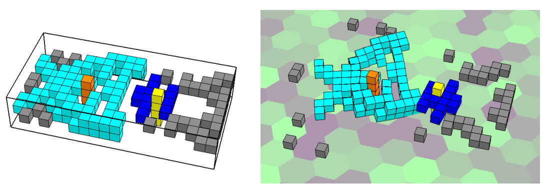


Figure 4.71. Coloration as used in Figure 4.73 for the analysis of solution M1-1.

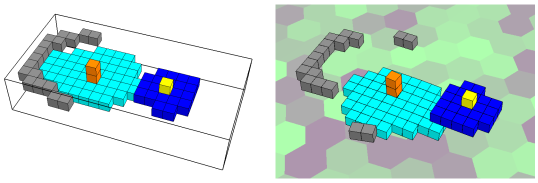


Figure 4.72. Coloration as used in Figure 4.73 for the analysis of solution M1-2.

WS	Flow domain	Displacement	Effort	Carrier
$WS_{1,10}$	Rotational 1D	Angle	Torque	Angular momentum
WS_{4-6}	Mechanical 3D	Angle, position	Force	Linear momentum (3D)
$WS_{2,3,7-9}$	Mechanical 3D	Angle, position	Internal Force, Torque	Linear & angular momentum (3D)

Table 4.43. Description of WS flows in Figure 4.73 for solution M1-1 and M1-2.

CSS	Flow domain	Energy storage
CSS_b	Mechanical 3D	Potential energy

Table 4.44. Description of CSSs in Figure 4.73 for solution M1-1 (with CSS_b).

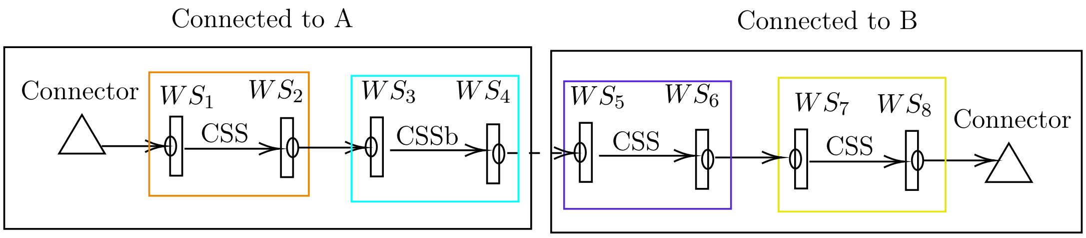


Figure 4.73. Graph analysis using the C&CM for solution M1-1 and M1-2. Colors represent the voxels as seen in Figure 4.71 and 4.72. Note that CSS_b only stores energy in solution M1-1 and not in M1-2.

The influence of the difference in this run to other runs, network input of the distance to A and B instead of distance to the center, is not clear. The shapes of the solutions are slightly more concentrated around A and B, but when random individuals from the population are taken, no clear concentration close to A and B can be seen (see Figure 4.74). No conclusion can be drawn from just this run.

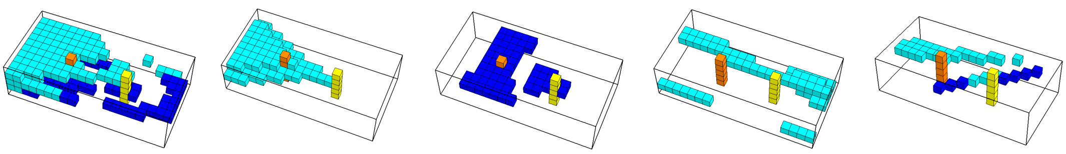


Figure 4.74. Some randomly selected individuals from run L1 that are non-continuous solutions.

4.2.13 Miscellaneous runs

More runs were performed that had results on which no conclusions could be drawn. For completeness, these are mentioned below.

- Changing network mutation probability. The probability of mutating parts of the network might have some influence on the outcome but not enough runs were performed to draw any conclusions.
- Valid phenotype was changed to not only check whether an individual has at least one voxel, but also if it is not completely filled with voxel. This is done because completely full individuals have a close to zero fitness. This did not have the desired result: many individuals are completely full when initialized, but when these are considered invalid, more computation-time is needed to generate new individuals. Besides this, fitness of later mutations of these individuals could be high, which cannot be determined at initialization.
- An extra CPPN was added to evolve stiffness of the artifacts as well as the shape. Erroneous behaviour was seen in the simulation in Voxelyze and as of now, no proper integration has been achieved.

4.3 Evaluation and conclusions of proof of concept

The software in the PoC, based on Evosoro, has been tested and runs to determine the right parameter settings and algorithms were performed empirically. Various parameters and elements were varied to test their influence on the ability of the software, an overview of these can be seen in Appendix A.5 in Table A.6 and A.7.
Various introduced shapes for A (applied torque) and B (measured rotation) were evaluated. It is unnecessary to use more than one voxel to portray the input and output flow for this particular problem. As objectives for the simulated artifact, angle or total rotation, rotation speed and number of voxels were evaluated. In the optimization, individual age was also used as an objective, next to these. Using total rotation to measure fitness and age and number of voxel elements as secondary objectives leads to the best results. Age and the number of voxels was already used in Evosoro [2]. Applying a large force or moment relative to material stiffness results in incorrect simulations because this results in a divergence or a failure to detect collisions. No threshold for values for the occurrence of these phenomena has been determined.
Different selection methods that were added to the software were tested for their desired effect: the introduced pareto-reset selection method did indeed result in other individuals evolving to a new optimum. The particular simulated annealing and ancestor selection implementations used in this PoC did not result in individuals with a higher fitness or more different working principles. A higher population size as well as a greater number of generations leads to a wider variety of solutions. A large individual size is important to guarantee the existence of a multitude of solutions within the solution space with different working principles.

The demands set earlier for the PoC will be evaluated in Section 5.1 to determine whether the PoC was successful.

5 Verification and validation

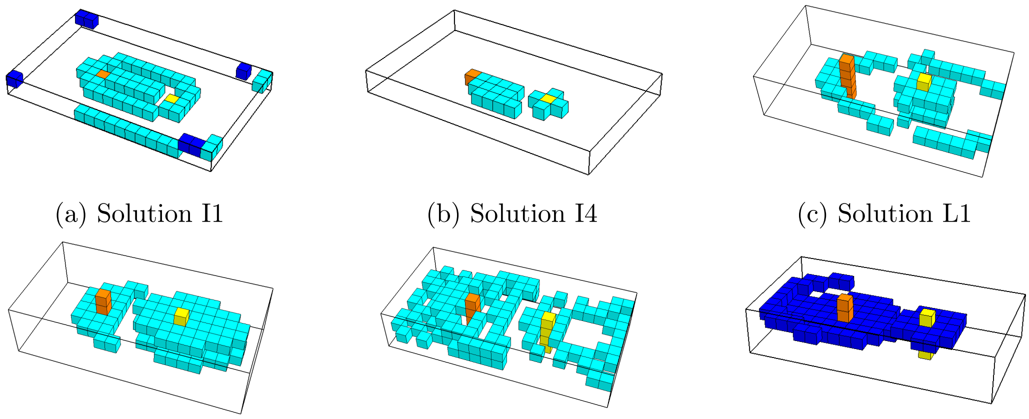
Verification of software is done to check whether it meets the demands determined on beforehand, validation of software is done to check that the software fulfills the goal, i.e. is it useful and to check if the demands were set correctly.
The proof of concept (PoC) will be verified by comparing the results of the runs and analyses with the goals set for the proof of concept in Section 3.3.1. To validate the proof of concept and the CAI-tool presented in this research, the versatility and usability are discussed.

5.1 Verification

This proof of concept has been evaluated as successful according to the demands set in Section 3.3.1:

- **The mechanical engineering problem will need to be described by one function and an input- and output flow, as is prescribed by the CAI-tool (Figure 3.5)**.
This has been done as the description of the problem shows: a function (transmitting) and input- and output flow (mechanical rotational energy) were defined and used as input for the software.
- **Any basic element used to build artifacts in the proof of concept should, in itself, not fulfill a function that is sought in the mechanical engineering problem that is to be solved.**
It can be said that indeed no basic elements, i.e. voxels, fulfill any function of their own in the created runs with the related mechanical engineering problem, as it is impossible for a single voxel to transfer the rotational mechanical energy from A to B.
- **The proof of concept will be successful when at least two solutions for the entered mechanical engineering problem, with different working principles, are produced in one run.** Because more than one working principle was found for the used mechanical engineering problem within one run without changing any parameters, this demand has also been met. This implies that no working principle was predetermined in any way in the software.

Run I, run L and run M resulted in two continuously working solutions with different working principles, these can be seen again in Figure 5.1. It must be noted that solution I1 is physically impossible. This was caused by the simulation environment but, if that environment could be made in real life, the solution has the ability to work. Next to those results, run D resulted in multiple continuous working solutions with a different embodiment and at least one other non-continuous working solution with a different working principle. Run J and run M resulted in continuously working solutions with different embodiment but with the same working principle.



(a) Solution I1 (b) Solution I4 (c) Solution L1

(d) Solution L2 (e) Solution M1-1 (f) Solution M1-2

Figure 5.1. Continuously working solutions with different working principles.

The demands were determined to rule out the introduction of working principles, even if implicitly. The working principles that were found in the runs were not introduced in any way since they are represented by their underlying CPPN that is in no way related to any working principle. This is confirmed by the fact that multiple working principles are found per run.

By fulfilling the demands set beforehand it can be said that the working of the software as a PoC is verified because all demands were met.

5.2 Validation

The demands set in Section 3.3.1 for a CAI-tool were met by the conducted proof of concept as discussed in the previous section. This is merely a sample of the possible problems that could have been tested for the proof of concept. Now two questions remain: is the PoC validated and is the CAI-tool itself validated?

5.2.1 Validation of the proof of concept

The goal of the proof of concept was to prove that the proposed CAI-tool has the envisaged abilities. In the opinion of the author, non standard mechanisms were created which did solve the specific problem of a transmission as defined in the PoC, and several distinct solution “concepts” of working principles could be identified. Since the CAI-tool is the same for all the functions and flows (see Table A.2 and A.3 in Appendix A.3), theoretically all mechanical engineering problems described in that way can be entered in the CAI-tool. It must be noted that when incorporating other flows or functions, these must first be implemented in the software. Moreover, based on the PoC nothing can be said about the viability of solving these other problems with the software or the method for the CAI-tool. In conclusion, it can be said that the PoC only has been partially successful in confirming all the abilities of the CAI-tool since the full application range has not been tested yet.

5.2.2 Validation of the CAI-tool

The goal of the CAI-tool itself is to generate artifacts that are function solutions for mechanical engineering problems on a conceptual level. The demand, that this needs to happen without introducing the working principle, is made so the solutions generated by the tool are not dependent on user or historical knowledge. However, solutions, that are recognizable by the author as historical solutions, were found and can be seen in Figure 5.2. For instance the transmission with teeth-uncompleted gears works in a similar way as solution D2. Solution K4 could be seen as a flexible coupling.

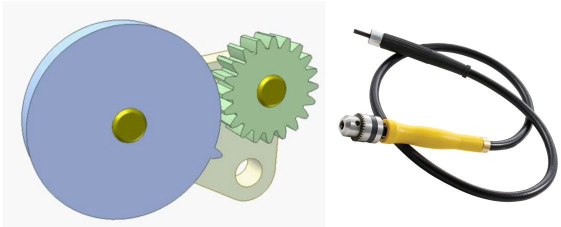


Figure 5.2. Transmission with teeth-uncompleted gears [131] and flexible coupling [132].

Nonetheless, it can be said that some of the solutions found in the PoC (e.g. I1) are novel and can not be found in literature on mechanisms [133]. Both novel solutions and ones that can be recognized as historical would be useful since these correspond to solutions that are achieved using discursive and intuitive, respectively, CAI-tools (see Table 2.1).
However, the level of detail of the solutions achieved in the PoC was not yet high enough to be applicable in mechanical engineering problems in practice. But, if the CAI-tool would be extended from the PoC on out, its ability to generate a multitude of working principles for one mechanical engineering problem would validate the CAI-tool as a novel one that blurs the line between intuitive and discursive CAI-tools as described in Section 2.1.2.

6 Discussion, conclusion, and recommendations

In this final chapter, the results of the complete research are interpreted, put into perspective of existing work and strengths, weaknesses and relevance are discussed. Finally, a conclusion is drawn and recommendations are made for further work.

6.1 Discussion

In this research, a CAI-tool was developed and a proof of concept was made. By using a form of computational optimization, artifacts were generated that were function solutions for a basic mechanical engineering problem. No working principle was introduced. Therefore, the main goal of this research has been met. However, some methods and findings in this research leave room for discussion which will be reviewed in this section.

6.1.1 CAI-tool

To use computational optimization to generate artifacts that are solutions to mechanical engineering problems a framework for a CAI-tool is developed (for clarification see Figure 3.5 on page 42).

A key element of this tool is the disengagement of knowledge on working principles and the mechanical engineering problem from the generation and optimization of artifacts. The basic elements that make up these artifacts have behavior and characteristics but should have no functionality of their own. Also no implications of certain working principles must be made in the definition of the problem that is used in the optimization (e.g. number of objects or their connection).
The intention of this disengagement is to allow *creative* solutions to be generated. The assumption that the reverse (to generate creative solutions with foreknowledge in the CAI-tool) is not possible, might have ruled out otherwise viable options for this CAI-tool but this has not been considered in this research.

The functional basis, using the EMS model, by Hirtz et al. [1] is implemented to describe the mechanical engineering problems according to their input and output flow and the function between those. This *flow* is also used in the C&CM method to describe the flows across working surfaces to analyze solutions in the proof of concept. This use of the word *flow* is quite confusing: a flow in the mechanical domain would mean a linear velocity, while in the EMS model a flow can be an energy, material or signal. In the C&CM method, the same terminology is used, since in that method flows move from one working surface to another, which is something a momentum or effort (e.g. force) could do but not a linear velocity.
While all this does not improve the clarity of analysis and mechanical engineering problem descriptions in this research, it is expected by the author that a change of definitions would have

had no effect on the outcome of this research.
It must be noted that even though the usage of flow in defining the mechanical engineering problem has been tested in the proof of concept for an energy, the case that this flow is a material or signal has not been verified in the PoC, which would be necessary for the verification of the complete model.

The model for the CAI-tool, as it is now, just incorporates one mechanical engineering function. While many properties can be included in the introduced problem definition, no extension for the implementation of other design requirements was made. For example, the combination of multiple functions or environmental constraints have not been incorporated even though these could have an influence on the solutions of a particular problem.

Many other CAI-tools exist but not one using these particular elements, structure and method. A CAI-tool that has some similarities to this research was mentioned in Section 2.1.2: Huang, Bo, and Chen [37] presented a process model for conceptual design based on computational intelligence. Key differences are the inclusion of a knowledge base, evaluation of solutions using predetermined values for partial solutions and strong human-machine interaction.
The possibility of using evolutionary algorithms in CAI-tools to hybridize (i.e. combining) existing solutions was mentioned by Leon [134]. However, again this makes use of existing solutions. An example is the synthesis of mechanisms using genetic algorithms by Zhang, Wang, and Dai [135], but they still used a predefined set of which the solutions consist.
The work of Cheney et al. [2], that is used in the PoC, shows similarities to the proposed CAI-tool but, as already mentioned in Section 1.2 and 2.4 does not create a technical system since no interaction occurs with adjacent systems according to the definition presented in Section 2.2.3.

The use of this definition is a point of discussion: different definitions of technical systems exist as well, for instance, TRIZ requires “a tool, engine, transmission and control” in a minimal technical system [136], which would see the solutions of the CAI-tool as a *tool* with a *transmission* within a technical system instead of a technical system itself. The *engine* providing energy then refers to the input flow, which is not part of the generated artifact.
Compared to the soft robots generated by Cheney et al. [2] a difference can be seen in the *control*: the element which governs at least one of the other systems. The function that the soft robots fulfill is converting-transferring the flow *thermal energy* to *mechanical translational energy*. While the soft robots rely on an oscillating temperature to control the *tool* to deliver their function, for the solutions that are generated in the PoC in this research, no clear *control* element can be indicated but the function is fulfilled without outside regulation.

Another point of interest is the time-complexity of the CAI-tool. The large solution space is the direct result of the desire to work without any input of working principles or foreknowledge about solutions. In the PoC, a relatively simple simulation was used as evaluation of artifacts and a limited set of materials and physical phenomena. Runs of the PoC were already costly considering the relative simplicity of the problem and thus small solution size, this would be increased should more materials or physical elements be included. No statements were made considering the implications thereof but knowledge on time-complexity is an essential part of further research.

6.1.2 Proof of concept

To prove that the framework for the CAI-tool results in software that is able to meet the set demands, a proof of concept was made, based on Evosoro [2]. Various elements were changed or added to the software including different population selection methods, allowing multiple objects and new fitness calculations.

With this software, a mechanical engineering problem was solved: transferring the input flow rotational mechanical energy to output flow rotational mechanical energy. Various runs were performed to adjust parameters and algorithms empirically. Multiple runs, wherein nothing was changed manually during the run, resulted in multiple solutions with a different working principle, thereby complying with the demands set for the PoC.

The choice of using CPPNs and choice of implemented mathematical functions to represent artifacts has a strong influence on their shape. While this is beneficial to the time-complexity of the software, some shapes appear more often than others (concentric shapes such as seen in solution I1, Figure 4.50). This prevalence of certain shapes could be seen as an unwanted effect: a possible inclination to these shapes in spite of the fact that an explicit limitation of the solution space is unwanted for the CAI-tool. The possible complexifying of the CPPNs at every mutation which is a result of the use of NEAT can cause a dithered artifact at higher ages (see Figure 4.40, solution G1, for instance). This results not only in many unfunctional voxels but also in an uniform complexity in many of the individuals.
The use of voxels in the PoC to build artifacts restricts solutions to these shapes that are not practical to implement in real-life. That makes the current software in the PoC more suitable as an inspiration tool for engineers than that it does provide directly applicable solutions.

Additionally, the difference between a continuous working principle and non-continuous working principle in this PoC often results in a discontinuity in the fitness-generation graph as a single voxel can determine that difference. In the work of Cheney et al. [2], most of the robots already work continuously from very early generations and single voxel differences have a smaller influence on functionality. These differences result in a more gradual increase in the fitness over generations.
No clear confirmation can be given of the hypothesis on why the introduced pareto selection with reset after a stagnating improvement seems to work better than the pareto selection method as used for the soft robots: not enough comparative runs have been performed for a thorough analysis. It can be said that these selection methods lead to rejuvenation of the population which leads to the discovery of unexplored niches in the solution space.

The implementation of the flow-function-flow description to convert the desired mechanical engineering function to an objective function and input and output flows in the simulation for the example problem leaves room for discussion as well, as also mentioned in Section 6.1.1.
The flow that is imported as well as exported, 'rotational mechanical energy', in the example problem of the PoC, is represented by a continuously applied moment at the input, an effort source, and is measured as rotation at the output, which can be seen as the average speed since simulation times are equal within a run. The input flow is not defined as an angular velocity which would be the correct definition of *flow* while the output flow is. Defining an angular velocity at the input could lead to an infinite torque when rotation is blocked. These unclarities with regard to the conversion of the mechanical engineering problem to measurements for the objective function and imported energies should be solved to improve the CAI-tool.
Additionally, instead of transferring mechanical energy, the problem which is solved in the proof of concept could also be seen as turning (function) of a solid object (flow). This might even fit in with the used objective function where the number of turns determines fitness, not the amount of energy that was conveyed from A to B. This ambiguity is also not beneficial for the clarity of the CAI-tool.

The C&CM is used to analyze the differences in working principles generated in the PoC. It could be argued that, since this method can be used at different levels of detail and the user determines the desired level of detail, it does not provide the desired unambiguousness. Especially for complex shapes, for instance solution M1-1 (see Section 4.2.12), the distinction between different WSs and CSSs is hard to make. Even though the author feels that the distinction between different working principles using the C&CM has largely corresponded with intuitive assessment, a different analysis method or changes to the C&CM could be beneficial to a more substantiated proof.

The friction model in Voxelyze is inconsistent, which became apparent when friction coefficients were changed, this could have effected certain outcomes where friction was used to describe the working principle.

The number of runs to determine parameters and properties of selection methods was restricted due to time-complexity of the software. The fact that the demands for the PoC were met is worth no less because of this but, due to the restricted number of runs, it cannot be said that optimal values for parameters were determined. The same applies to the choice for the pareto reset selection method: empirically better results are achieved but this might not hold when a larger set of runs is performed with a larger variety in parameter values.

Reviewing the discussion of both the CAI-tool and the PoC, it can be said that the demands for the proof of concept were met, the proposed CAI-tool has a feasible structure and that the CAI-tool is new and unique. However, further research needs to be done on the viability and possible application of the CAI-tool as well as the particular software implementation as it is executed in the PoC.

6.2 Conclusion

The goal of this research was to prove that it is possible to use computational optimization to generate artifacts that are function solutions for basic mechanical engineering problems on a conceptual level without introducing the working principle as input for said optimization. This goal has been met by answering the following research questions:

- **What should be the structure and components of such a CAI-tool?**
  - **Which components are needed besides the computational optimization algorithm?**
    As was discussed in Section 3.2, it is necessary to have a method which allows the user input to be converted from a description of an engineering problem into input for the optimization. Along with that, some shape representation needs to be used that is able to express basic elements that have behavior and characteristics but no functionality on their own. An evaluation of artifacts needs to be done using a simulation that is able to simulate any laws of physics that are relevant to that mechanical engineering problem.
  - **How can an engineering problem be converted to an input for such a CAI-tool?**
    The problem is described as a function using the functional basis by Hirtz et al. [1] (as in Section 2.2.2) which is imposed upon an input flow to create an output flow. These flows (energy, material or signal) and their properties for this specific problem are inserted in the simulation and the function is converted to an objective function that describes the relation between in- and output in this situation. See also Section 3.1 and 3.2
  - **How to detect to what extent a mechanical engineering function is fulfilled by a solution?**
    The objective function returns a fitness value so a comparison between solutions can be made. A distinction can be made from this value, as described in Section 3.3.4, whether a solution does not work, resulting in a very low fitness value ($fitness \approx 0$), or a working solution, with a fitness value that reaches a set value after some time ($fitness = x$) and finally a continuously working solution, of which the fitness value keeps increasing over time when the simulation is continued ($fitness = x \cdot t$).

  - **How to ensure no knowledge of working principles is introduced?**
    Besides the usage of basic elements without functionality, the problem is disengaged from the actual artifact generation and optimization by the usage of the flow-function-flow description, which was presented in Section 3.2. Also a shape representation is used to describe the shapes that are made up from the basic elements as this ensures an extra uncoupling between the problem description and generated artifacts, another advantage is that it compresses the size of artifact description.

- **How can a proof of concept be made to test such a CAI-tool based on a genetic algorithm?**

  - **Which genetic algorithm based software is suitable to be adapted for a proof of concept for such a CAI-tool?** The CPPN-NEAT based Evosoro [2] is suitable to be adapted as a proof of concept for the CAI-tool because it has all the necessary elements as described above: a shape representation (CPPN) that represents artifacts made from basic elements (voxels) and a powerful generative encoding to optimize these (CPPN-NEAT) using a fast simulation (Voxelyze). Evosoro was discussed in Section 3.3.2 and other genetic algorithms in 2.3.
  - **What adaptions should be made to that software?** To be able to solve mechanical engineering problems as described for the CAI-tool, multiple objects need to be allowed, other boundary conditions and fitness measurement are necessary and, to avoid getting stuck in local optima, because of the highly non-linear solution space, a different selection method needs to be introduced. All adaptions can be seen in Table 3.4.
  - **How can (generated) artifacts be distinguished as function solutions?** Besides the fitness value to establish continuous working principles, an adapted version of the C&CM (see Section 2.2.3 and 3.1) is used to distinguish working principles from one another by analyzing the flows across working surfaces and channel and connect structures using a graph structure.
  - **What demands need to be met for the proof of concept to be successful?**
    To be sure that no working principle is introduced, more than one working principle, fulfilling the mechanical engineering function, needs to be found in one run of the software in the proof of concept. These requirements can be found in Section 3.3.1.

This research has demonstrated that it is possible to let software generate working principles for mechanical engineering functions, without introducing any working principles first. The ability to do this could be useful in concept generation as a CAI-tool or, after another software implementation has been realized, to directly generate applicable solutions. This tool has the potential to uncover the path to easier radical innovation because it is not affected by foreknowledge. More work needs to be done to improve the software made in the proof of concept so that different functions can be solved and more extensive experiments are necessary to determine parameter values and other elements that are in need of improvement.

6.3 Recommendations

The CAI-tool that is presented here lends itself to be explored further in different areas. The following recommendations are made to direct future research on the CAI-tool itself:

- The conversion of the problem description using the functional basis leaves, to say the least, room for discussion. Instead of transferring mechanical energy, the problem in the proof of concept could also be seen as turning (function) of a solid object (flow). This might even fit in with the used objective function where the number of turns determines fitness, not the amount of energy that was conveyed from A to B.

- Add a way of describing the environment of the mechanical engineering problem in the CAI-tool such as external forces or constraints so that a more detailed problem can be solved for a particular situation.
- Evaluate what other functional demands could be included as opposed to just one mechanical engineering function as an objective.
- Perform a user study to determine user demands and discover possible applications.

The following recommendations are made about pursuing further research on the specific software implementation as was done in the PoC:

- Extend the software to solve other mechanical engineering functions to verify the other parts of the full spectrum of functions and flows, as used in the CAI-tool, such as material and signal flows.
- Make the C&CM more explicit or use a different method so distinction between working principles can be made less user-dependent.
- Implementing the C&CM to automatically analyze and evaluate solutions during the simulation could enable distinguishing niches in the solution space and their relevant working principles. This could be combined with an automatic calculation of the variety rating that was presented in Section 3.1, leading to a variety sum of the complete population at that point.
- Analyze ancestry of successful individuals to compare them to unsuccessful individuals and use that difference to detect promising local optima if possible.
- Vary constraints, environments, the way of defining flows or apply more complicated requirements. For instance: applying a rotational velocity instead of a moment as the boundary condition, as a source of flow instead of effort. Or a requirement can be set on the rotation transmission ratio.
- Continue with the implementation of an extra CPPN to evolve stiffness as well as shape. A small difference in stiffness can have a large influence on the performance of a working principle since for instance artifacts can bend to accommodate further movement.
- Improve the friction implementation in Voxelyze, since no consistent results were achieved.
- Test if not allowing completely filled individuals influences the optimization in a positive way by analyzing the genealogy of individuals that have high fitness.
- Apply crossover when generating the new population. This is already possible using CPPN-NEAT, just not in Evosoro. This will allow individuals with high fitness to combine beneficial properties.
- When pursuing simulated annealing, better results from earlier generations should not be ignored but kept in the population. This way, the best individuals are not discarded.
- Determine the influence of the ancestor overlap and stagnation threshold for the reset in the pareto reset selection method. These were determined empirically and not tested thoroughly.
- Instead of voxels, use smooth basic elements or a representation that allows this to have directly applicable solutions.

- A less time costly simulation could lead to better results since more individuals can be evaluated.
- Try different software to implement the CAI-tool. The usage of the CPPNs as artifact representation instead of direct encoding diminished the time complexity but could have omitted results. Otherwise, other activation functions in the network could enlarge the possibilities. Or a totally different representation such as the VRML scene graph (see Figure 2.7) might result in a faster search, a wider range of possible working principles or other positive effects on the solutions that are generated.

Future possibilities

Even though it is proven that concepts can be created using genetic algorithms for the specific case of the presented proof of concept, the relevance of the posed method remains a point of discussion. With the emergence of other artificial intelligence systems, the author thinks it is not a question of *if* but *how* computers will be able to take over (parts) of conceptual designing. It could well be that machine learning combined with existing patents and designs could be implemented faster and/or more easily to help in the conceptual phase.
Next to new results for stand-alone engineering problems, the proposed CAI-tool could provide a crossover between solutions for more complex problems that would otherwise be impossible to be created. Using for instance multi-material 3D-printers that are capable of printing a gradient of stiffness, more complex shapes could be created than can be designed by a human. The time-complexity and computing power limit the complexity of problems that could be solved. However, with ever-increasing computer capacity, the author thinks eventually integrated or combined results will be possible.

A Appendix

A.1 CAI Categories

A.1.1 Categories based on application field

This section is based on Kohn and Hüsig[19].

- **Strategy management**, which aids designing projects on a higher strategy level. Scenario-, portfolio-, project management and business intelligence are within this category.
- **Idea Management**, helps from the start of idea generation, collection of these ideas and finally evaluating them.
  - **Idea generation** includes tools that rely on techniques such as TRIZ, brainstorming and mind mapping tools.
  - **Idea collection** helps in gathering ideas by different designers and bringing them together to increase efficiency of the designing process.
  - **Idea classification and portfolio** usually helps in clustering and subsequently visualizing the ideas.
  - **Analysis of ideas** One example is the use of TRIZ function trimming on an already existing design [137].
- **Patent management** Not only can patents protect inventions, they can also help in creating new inventions. TRIZ for example, relies partially on existing solutions.
  - **Invention report** Allows for a more efficient process because new inventions are reported in a CAI tool.
  - **Patent search** Many online search engines exist that perform searches through databases of historic patents.
  - **Patent analysis** To digitize or understand the patents they have to be analyzed. Recent research has been able to categorize and interpret patents using linguistic processing [138].
  - **Patent administration** This is less relevant for a designer because CAI tools in this category help after a design is made, for instance registering patents and financial aspects.
  - **Patent evaluation** For the patents that are owned or in use by a company it is necessary for them to be evaluated for significance.
  - **Patent portfolio management** Evaluated patents can be managed by CAI.

When multiple categories are included in one CAI product it is called a *holistic solution*, which leads to results that could not have been obtained if just one of the separate categories was used.

A.1.2 Categories based on potential benefits

- **Efficiency enhancing** Routine tasks and sharing of information have great potential to be integrated in CAI. It relieves workforce when repeating tasks that do not require a deeper insight in the situation are outsourced to CAI. The streamlining and digitizing of information sharing leads to more consistent, easier accessible and updatable data.

- **Effectiveness enhancing** The innovativeness of a project is usually set in the first phase with no easy change of concept in later phases, therefore a large part of the outcome of a project (which is already defined in these early stages) could benefit if CAI was implemented in this early stage. Visualisation and fast simulation of concepts helps in the evaluation of more concepts than usual but also more complex tools that are capable of analyzing and evaluating customer or technological information will improve the effectiveness of NPD projects.

- **Competence enhancing** Helps a larger part of a firm become familiar with CAI methods. Users transcend their level of competence by using CAI and having access to up-to-date data. Also a better connection to the end user is possible, for instance Sosa and Gero [139] investigated a social simulation to relate the context of the market to the process of innovation.

- **Creativity enhancing** There is only one application category that has the potential to enhance creativity: idea generation. Kohn and Hüsig [19] argue that while later stages in NPD are focussed on quality, the early ideation stage is characterized by quantity. This is reflected by the works cited and other related work, which tend to focus on the support of designer creativity such as "electronic meeting" or structured approach of brainstorming techniques such as TRIZ. And while it is true that generally a large pool of concepts is preferred, the quality of the best concept determines the maximum level of quality the whole product will eventually be able to reach. This makes any improvement of creative capabilities valuable to NPD.

A.2 Blocks and tackles

BLOCKS	TACKLES	EVIDENCE
• premature judgment	• suspend judgement	(Candy and Edmonds, 1996), (Christensen et al, 1957)*, (Donovan, 1985), (Ekvall and Parnes, 1984), (Grossman and Wiseman, 1993), (Hyman, 1961)*, (Kolodner and Wills, 1996), (Osborn, 1979), (Schwab, 1991), (Stein, 1975)*, (Tovey, 1986)
• emphasis on quality	• emphasis on quantity	(Basadur et al, 1982), (Basadur and Thompson, 1986), (Bugliarello, 1969), (Candy and Edmonds, 1996), (Carson and Carson, 1993)*, (Cohen et al, 1960), (Cross and Cross, 1996), (Donovan, 1985), (Grossman and Wiseman, 1993)*, (Hist, 1992), (Kumar et al, 1991), (Osborn, 1979), (Parnes, 1987), (Parnes, 1961), (Parnes and Meadow, 1959), (Purcell, and Gero, 1996), (Rickards and Freedman, 1978), (Rowatt et al, 1997)*, (Roy, 1993), (Simonton, 1987), (Tovey, 1986)
	• emphasis on variety	(Roy, 1993), (Tovey, 1986)
• lack of motivation: satisfied with present solution	• generate alternatives anyway for its own sake	
• having a tight grip on problem specifications	• change the frame of reference (view data in a different way)	(Akin and Akin, 1996), (Barron, 1988), (Candy and Edmonds, 1996), (Cross and Cross, 1996), (Kolodner and Wills, 1996), (Lipshitz and Waingortin, 1995), (Rawlinson, 1981), (Savage and Miles, 1998)*, (Watzlawick et al, 1974)
	• use of analogies and metaphors	(Candy, 1996), (Candy and Edmonds, 1996), (Cross, 1996), (Ekvall and Parnes, 1984), (Gordon, 1961), (Grossman and Wiseman, 1993), (Isaksen and Parnes, 1985), (Parnes, 1987), (Perkins, 1981), (Polya, 1945), (Poze, 1983), (Roy, 1993), (Shaw, 1986), (Sternberg, 1977), (Visser, 1996), (Visser, 1995), (Visser., 1992)
• rigid problem representation (textual, mathematical)	• flexible problem representation (abstract, pictorial)	(Cross and Cross, 1996), (Lawson, 1994), (Roy, 1993), (Tovey, 1986)
• design fixation (attached to one idea)	• provocative stimuli	(DeBono, 1984), (Osborn, 1979)
	• random connections	(Grossman and Wiseman, 1993), (Parnes, 1987), (Zwicky, 1969)
	• incubation	(Cross and Cross, 1996), (Kumar et al, 1991), (Lawson, 1994), (Parnes, 1987)
• imposing fictitious constraints	• break rules	(Candy and Edmonds, 1996), (Kolodner and Wills, 1996)
	• work on higher problem	(Candy and Edmonds, 1996), (Cross and Cross, 1996), (Kolodner and Wills, 1996)

Table A.1. Components used in DM to overcome blocks [71].

A.3 A functional basis by Hirtz et al.

Class (Primary)	*Secondary*	*Tertiary*	*Correspondents*
Branch	Separate		Isolate, sever, disjoin
		Divide	Detach, *isolate*, release, sort, split, disconnect, subtract
		Extract	Refine, filter, purify, percolate, strain, *clear*
		Remove	Cut, drill, lathe, polish, sand
	Distribute		Diffuse, dispel, disperse, dissipate, diverge, scatter
Channel	Import		Form entrance, *allow*, input, *capture*
	Export		Dispose, eject, *emit*, empty, *remove*, destroy, eliminate
	Transfer		Carry, deliver
		Transport	Advance, lift, move
		Transmit	Conduct, convey
	Guide		Direct, shift, steer, straighten, switch
		Translate	Move, relocate
		Rotate	Spin, turn
		Allow DOF	*Constrain*, unfasten, unlock
Connect	Couple		Associate, connect
		Join	Assemble, fasten
		Link	Attach
	Mix		Add, blend, coalesce, combine, pack
Control Magnitude	Actuate		Enable, initiate, start, turn-on
	Regulate		Control, equalize, limit, maintain
		Increase	*Allow*, open
		Decrease	Close, delay, interrupt
	Change		Adjust, modulate, *clear*, demodulate, invert, normalize, rectify, reset, scale, vary, modify
		Increment	Amplify, enhance, magnify, multiply
		Decrement	Attenuate, dampen, reduce
		Shape	Compact, compress, crush, pierce, deform, form
		Condition	Prepare, adapt, treat
	Stop		End, halt, pause, interrupt, restrain
		Prevent	Disable, turn-off
		Inhibit	Shield, insulate, protect, resist
Convert	Convert		Condense, create, decode, differentiate, digitize, encode, evaporate, generate, integrate, liquefy, *process*, solidify, transform
Provision	Store		Accumulate
		Contain	*Capture*, enclose
		Collect	Absorb, consume, fill, reserve
	Supply		Provide, replenish, retrieve
Signal	Sense		Feel, determine
		Detect	Discern, perceive, recognize
		Measure	Identify, *locate*
	Indicate		Announce, show, denote, record, register
		Track	Mark, time
		Display	*Emit*, expose, select
	Process		Compare, calculate, check
Support	Stabilize		Steady
	Secure		*Constrain*, hold, place, fix
	Position		Align, *locate*, orient
Overall increasing degree of specification →			

Table A.2. Functions used in the functional basis by Hirtz et al.[1]

Class (Primary)	*Secondary*	*Tertiary*	*Correspondents*
Material	Human		Hand, foot, head
	Gas		Homogeneous
	Liquid		Incompressible, compressible, homogeneous,
	Solid	Object	Rigid-body, elastic-body, widget
		Particulate	
		Composite	
	Plasma		
	Mixture	Gas-gas	
		Liquid-liquid	
		Solid-solid	Aggregate
		Solid-Liquid	
		Liquid-Gas	
		Solid-Gas	
		Solid-Liquid-Gas	
		Colloidal	Aerosol
Signal	Status	Auditory	Tone, word
		Olfactory	
		Tactile	Temperature, pressure, roughness
		Taste	
		Visual	Position, displacement
	Control	Analog	Oscillatory
		Discrete	Binary
Energy	Human		
	Acoustic		
	Biological		
	Chemical		
	Electrical		
	Electromagnetic	Optical	
		Solar	
	Hydraulic		
	Magnetic		
	Mechanical	Rotational	
		Translational	
	Pneumatic		
	Radioactive/Nuclear		
	Thermal		
	Overall increasing degree of specification →		

Table A.3. Flows used in the functional basis by Hirtz et al.[1]

Domain Type	**Potential**	**Flow**	**Carrier**
Electrical	Voltage	Electric current	Charge
Translational 1D	Position	Force	Linear momentum
Rotational 1D	Angle	Torque	Angular momentum
Mechanical 3D	Position and Rotation angle (3D)	Force and Torque (3D)	Linear and angular momentum (3D)
Magnetic	Magnetic potential	Magnetic flux rate	Magnetic flux
Hydraulic flow	Pressure	Volume flow rate	Volume
Heat	Temperature	Heat flow rate	Heat
Chemical	Chemical potential	Particle flow rate	Particles

Table A.4. Energy carriers with associated potential and flow quantities [129].

Class (Primary)	Secondary	Tertiary	Power conjugate complements	
			Effort analogy	Flow analogy
Energy			Effort	Flow
	Human		Force	Velocity
	Acoustic		Pressure	Particle velocity
	Biological		Pressure	Volumetric flow
	Chemical		Affinity	Reaction rate
	Electrical		Electromotive force	Current
	Electromagnetic		Effort	Flow
		Optical	Intensity	Velocity
		Solar	Intensity	Velocity
	Hydraulic		Pressure	Volumetric flow
	Magnetic		Magnetomotive force	Magnetic flux rate
	Mechanical		Effort	Flow
		Rotational	Torque	Angular velocity
		Translational	Force	Linear velocity
	Pneumatic		Pressure	Mass flow
	Radioactive/Nuclear		Intensity	Decay rate
	Thermal		Temperature	Heat flow

Table A.5. Power conjugate complements for the energy class of flows [1].

Types of energy, physical effects and manifestations with examples [5], as seen in Table A.4 and A.5:

- Mechanical: gravitation, inertia, centripetal force, friction.
- Hydraulic: hydrostatic, hydrodynamic.
- Pneumatic: aerostatic, aerodynamic.
- Electrical: electrostatic, inductive, capacitative, piezo-electric, transformation, rectification.
- Magnetic: ferromagnetic, electromagnetic.
- Optical: reflection, refraction, diffraction, interference, polarisation
- Thermal: expansion, heat transfer, heat conduction, heat storage.
- Chemical: combustion, oxidation, reduction, dissolution, electrolysis, exothermic and endothermic reactions.
- Nuclear: radiation, isotopes.
- Biological: fermentation, putrefaction, composition.

A.4 Mechanical systems

Since the proof of concept in this research uses a mechanical system, specific description methods for working principles based on kinematic features could be relevant as well. A kinematic feature is described as a generic entity that can be used to study the motion of an artifact.

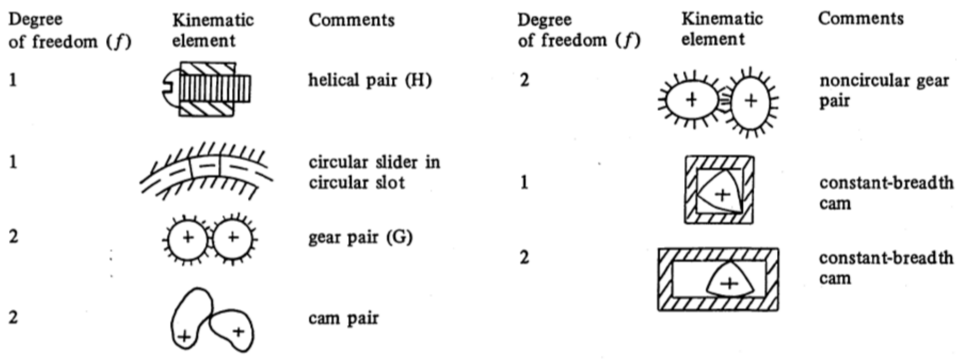


Figure A.1. Kinematic elements with coupled rotational and translational freedoms [140]

Several elements are used in describing a kinematic feature: [140]:

- **Joints**, such as pin joints, a sliding pair or a ball joint.
- **Links**: the rigid connections between joints.
- **Support**: the connection to the world.
- **DOF (degree of freedom)** of every joint and the total space.

This allows analyzing mechanisms as their kinematic structure. Figure A.1 shows kinematic elements that have coupled freedoms, analysed by their DOF.
Chakrabarti and Bligh [57] discourage usage of this approach. In these cases an application beyond mechanisms might be possible: it is not generalizable and integrable with designs in other domains. It also does not support decomposing problems from general to specific as it has one set (joints, links and supports) of elements that it works with on all levels. But most important: it does not provide individual functional reasons why each element is necessary in a solution concept. Their alternative is to follow the power flow path from input to output point, where the motion transmission elements are the "primary structures". These transform some of the characteristics of the variables between input and output. A bicycle drive is shown as example in Figure A.2 where for instance *structure A* transforms the input *force I* at *position p1* to an output torque at *position p2*.
This method is too specific to be used to analyze solutions in the PoC and therefore is not included any of the chapters.

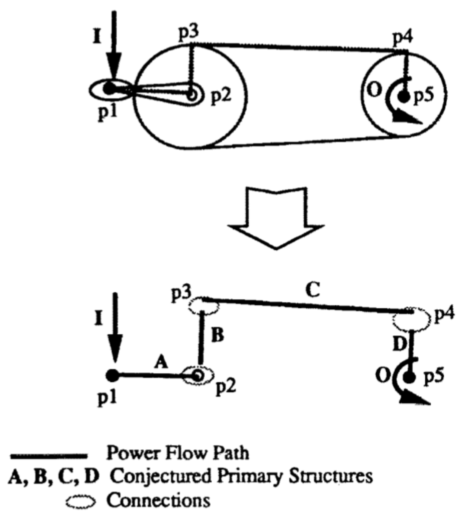


Figure A.2. Design knowledge extracted using power flow paths [57]

A.5 Overview run parameters

Experiment	A	B	C	D	E	F
Size (x,y,z)	(10,5,1)	(10,5,1)	(10,5,1)	(20,11,1)	(20,10,2)	(20,10,1)
Pop. size	10	200	50	250	30	250
Random inds.	10	2	2	20	6	20
No. of gens.	26	1000	140	3000	4500	1450
Init. time	0.1 s	0.1 s	0.01 s	0.01 s	0.1 s	0.01 s
Sim. time	2 s	2 s	0.2 s	1 s	1 s	1 s or no motion
Hard material $E(MPa), \rho(\frac{kg}{m^3})$	$500, 1 \cdot 10^6$	$500, 1 \cdot 10^6$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^3$	$50, 1 \cdot 10^3$
Soft material $E(MPa), \rho(\frac{kg}{m^3})$	-	-	-	-	$5, 1 \cdot 10^3$	-
A material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^6$	$500, 1 \cdot 10^6$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$
B material $E(MPa), \rho(\frac{kg}{m^3})$	$5, 1 \cdot 10^6$	$5, 1 \cdot 10^6$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$
Location of A	(0.72,0.45,0)	(0.72,0.45,0)	(0.72,0.45,0)	(0.66,0.47,0)	(0.66,0.47,0)	(0.66,0.5,0)
Location of B	(0.32,0.45,0)	(0.32,0.45,0)	(0.32,0.45,0)	(0.31,0.47,0)	(0.31,0.47,0)	(0.33,0.5,0)
Moment at A	-100 Nm $\circlearrowright$ $\circlearrowleft$	-100 Nm $\circlearrowright$ $\circlearrowleft$	-5 Nm $\circlearrowright$ $\circlearrowleft$	0.5 Nm $\circlearrowright$ $\circlearrowleft$	-0.5 Nm $\circlearrowright$ $\circlearrowleft$	0.5 Nm $\circlearrowright$ $\circlearrowleft$
Selection method	Pareto	Pareto	Pareto	Pareto	Pareto	Simulated annealing

Table A.6. Overview of parameters used in runs A-F.

Experiment	**G**	**H**	**I**	**J0-J9**	**K**	**L**	**M**
Notes	-	-	-	-	perpendicular gravity	-	-
Size (x,y,z)	(20,11,2)	(20,11,1)	(20,11,2)	(14,7,3)	(14,7,1)	(20,10,3+2)	(20,10,2+2)
Pop. size	200	250	100	19	100	50	50
Random inds.	200	20	20	1	20	5	5
No. of gens.	1000	190	9999	9999	1500	11000	15000
Init. time	0.1 s	0.05 s	0.1 s	0.1 s	0.1 s	0.1 s	0.1 s
Sim. time	1 s or no motion	1 s or no motion	1 s or no motion	1 s or no motion	1 s or no motion	3 1 s or no motion	3 s or no motion
Hard material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^3$	$50, 1 \cdot 10^3$	$50, 1 \cdot 10^3$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^3$	$50, 1 \cdot 10^3$	$50, 1 \cdot 10^3$
Soft material $E(MPa), \rho(\frac{kg}{m^3})$	$5, 1 \cdot 10^3$	-	$5, 1 \cdot 10^3$	$5, 1 \cdot 10^5$	$5, 1 \cdot 10^3$	$5, 1 \cdot 10^3$	$5, 1 \cdot 10^3$
A material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$
B material $E(MPa), \rho(\frac{kg}{m^3})$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$	$50, 1 \cdot 10^5$
Location of A	(0.66,0.5,0)	(0.66,0.5,0)	(0.66,0.47,0)	(0.66,0.47,0)	(0.66,0.47,0)	(0.66,0.5,0)	(0.66,0.5,0)
Location of B	(0.33,0.5,0)	(0.33,0.5,0)	(0.33,0.47,0)	(0.33,0.47,0)	(0.33,0.47,0)	(0.33,0.5,0)	(0.33,0.5,0)
Moment at A	-0.5 Nm ↻ ↻	0.5 Nm ↺ ↺	0.5 Nm ↺ ↺	0.5 Nm ↺ ↺	-0.5 Nm ↻ ↻	0.5 Nm ↺ ↺	0.5 Nm ↺ ↺
Selection method	Simulated annealing	Ancestor comparison	Pareto reset	Pareto reset	Pareto reset	Pareto reset	Pareto reset

Table A.7. Overview of parameters used in runs G-M.

Bibliography

[1] Julie Hirtz et al. "A functional basis for engineering design: Reconciling and evolving previous efforts". In: *Research in Engineering Design* 13.2 (Mar. 2002), pp. 65–82. DOI: 10.1007/s00163-001-0008-3.

[2] Nick Cheney et al. "Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding". In: *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2011).* ACM Press, 2011.

[3] Thomas Alink. "Bedeutung, Darstellung und Formulierung von Funktion für das Lösen von Gestaltungsproblemen mit dem C&C-Ansatz". In: *Forschungsberichte. IPEK ; 48* (2010), p. 239.

[4] A Albers, T Deigendesch, and T Alink. "Support of Design Engineering Activity - the Contact and Channel Model (C&CM) in the Context of Problem Solving and the Role of Modelling". In: *DS 48: Proceedings DESIGN 2008, the 10th International Design Conference, Dubrovnik, Croatia.* 2008, pp. 97–102.

[5] G. Pahl et al., eds. *Engineering design: a systematic approach.* 3rd ed. London: Springer, 2007.

[6] Robert B. Stone and Kristin L. Wood. "Development of a Functional Basis for Design". In: *Journal of Mechanical Design* 122.4 (2000), p. 359. DOI: 10.1115/1.1289637.

[7] Wolfgang Banzhaf. "Genotype-phenotype-mapping and neutral variation—a case study in genetic programming." In: *International Conference on Parallel Problem Solving from Nature.* 1994, pp. 322–332.

[8] Stefan Hüsig. "A Typology for Radical Innovation Projects Based on an Innovativeness Framework". In: *International Journal of Innovation and Technology Management* 11.04 (Aug. 2014), p. 1450023. DOI: 10.1142/S0219877014500230.

[9] S.D. Savransky. *Engineering of Creativity: Introduction to TRIZ Methodology of Inventive Problem Solving.* 2000.

[10] Byungun Yoon and Christopher L. Magee. "Exploring technology opportunities by visualizing patent information based on generative topographic mapping and link prediction". In: *Technological Forecasting and Social Change* 132 (July 2018), pp. 105–117. DOI: 10.1016/j.techfore.2018.01.019.

[11] Boris Zlotin and Alla Zusman. "Levels of Invention". In: *Encyclopedia of Creativity, Invention, Innovation and Entrepreneurship.* Ed. by Elias G. Carayannis. New York, NY: Springer New York, 2013, pp. 1199–1205. DOI: 10.1007/978-1-4614-3858-8_41.

[12] Kay Chuan Tan, Min Xie, and Xiao-Xiang Shen. "Development of Innovative Products Using Kano's Model and Quality Function Deployment". In: *International Journal of Innovation Management* 03.03 (Sept. 1999), pp. 271–286. DOI: 10.1142/S1363919699000141.

[13] Cam Caldwell and Verl Anderson. "Blue ocean strategies and disruptive innovation: Creative insights for competitive advantage". In: *Competitive Advantage: Strategies, Management and Performance.* Jan. 2017, pp. 97–108.

[14] Frank H. Winter, Michael J. Neufeld, and Kerrie Dougherty. "Was the rocket invented or accidentally discovered? Some new observations on its origins". In: *Acta Astronautica* 77 (Aug. 2012), pp. 131–137. DOI: 10.1016/j.actaastro.2012.03.014.

[15] Mohamed Gad-el-Hak. "Nine Decades of Fluid Mechanics". In: *Journal of Fluids Engineering* 138.10 (Aug. 2016), p. 100802. DOI: 10.1115/1.4033961.

[16] Ronald Kline. "Science and Engineering Theory in the Invention and Development of the Induction Motor, 1880-1900". In: *Technology and Culture* 28.2 (Apr. 1987), p. 283. DOI: 10.2307/3105568.

[17] Maxwell James Clerk. "On the theory of the maintenance of electric currents by mechanical work without the use of permanent magnets". In: *Proceedings of the Royal Society of London.* Vol. 15. Jan. 1867, pp. 397–402. DOI: 10.1098/rspl.1866.0089.

[18] Roger Th. A. J. Leenders, Jo M. L. van Engelen, and Jan Kratzer. "Systematic Design Methods and the Creative Performance of New Product Teams: Do They Contradict or Complement Each Other?" In: *Journal of Product Innovation Management* 24.2 (Mar. 2007), pp. 166–179. DOI: 10.1111/j.1540-5885.2007.00241.x.

[19] Stefan Kohn and Stefan Hüsig. "Development of an empirical-based categorisation scheme for CAI software". In: *International Journal of Computer Applications in Technology* Vol.30.No.1/2 (2007), p. 33 –46.

[20] Erik Andreassen et al. "Efficient topology optimization in MATLAB using 88 lines of code". In: *Structural and Multidisciplinary Optimization* 43.1 (Jan. 2011), pp. 1–16. DOI: 10.1007/s00158-010-0594-7.

[21] Daniël Linschoten. *Assignments Computational Structural Optimization.* Nov. 2017.

[22] Sen Lin et al. "Topology Optimization of Fixed-Geometry Fluid Diodes". In: *Journal of Mechanical Design* 137.8 (Aug. 2015), p. 081402. DOI: 10.1115/1.4030297.

[23] Salomé Galjaard, Sander Hofman, and Shibo Ren. "New Opportunities to Optimize Structural Designs in Metal by Using Additive Manufacturing". In: *Advances in Architectural Geometry 2014*. Ed. by Philippe Block et al. Cham: Springer International Publishing, 2015, pp. 79–93. DOI: 10.1007/978-3-319-11418-7_6.

[24] *The Power of Generative Design is coming to Autodesk Netfabb 2018 | In the Fold.* June 2018.

[25] W. Yamazaki, K. Matsushima, and K. Nakahashi. "Aerodynamic Design Optimization Using the Drag-Decomposition Method". In: *AIAA Journal* 46.5 (May 2008), pp. 1096–1106. DOI: 10.2514/1.30342.

[26] Takao Yokota, Mitsuo Gen, and Yin-Xiu Li. "Genetic algorithm for non-linear mixed integer programming problems and its applications". In: *Computers & Industrial Engineering* 30.4 (Sept. 1996), pp. 905–917. DOI: 10.1016/0360-8352(96)00041-1.

[27] Eduardo Lalla-Ruiz et al. "Biased random key genetic algorithm for the Tactical Berth Allocation Problem". In: *Applied Soft Computing* 22 (Sept. 2014), pp. 60–76. DOI: 10.1016/j.asoc.2014.04.035.

[28] Ivo Marinić-Kragić, Damir Vučina, and Zoran Milas. "Concept of flexible vertical-axis wind turbine with numerical simulation and shape optimization". In: *Energy* 167 (Jan. 2019), pp. 841–852. DOI: 10.1016/j.energy.2018.11.026.

[29] John R. Koza et al. "Genetic Programming: Biologically Inspired Computation That Exhibits Creativity in Producing Human-Competitive Results". In: *Creative Evolutionary Systems*. Ed. by Peter J Bentley and David W. Corne. The Morgan Kaufmann Series in Artificial Intelligence. San Francisco: Morgan Kaufmann, Jan. 2002, pp. 275–298. DOI: 10.1016/B978-155860673-9/50048-3.

[30] Daekeon Ahn, Hochan Kim, and Seokhee Lee. "Fabrication direction optimization to minimize post-machining in layered manufacturing". In: *International Journal of Machine Tools and Manufacture* 47.3-4 (Mar. 2007), pp. 593–606. DOI: 10.1016/j.ijmachtools.2006.05.004.

[31] Saleh Tabandeh, William W. Melek, and Christopher M. Clark. "An adaptive niching genetic algorithm approach for generating multiple solutions of serial manipulator inverse kinematics with applications to modular robots". In: *Robotica* 28.04 (July 2010), pp. 493–507. DOI: 10.1017/S0263574709005803.

[32] Noel Leon Rovira et al. "Automatic shape and topology variations in 3D CAD environments for genetic optimisation". In: *International Journal of Computer Applications in Technology* 30.1/2 (2007), p. 59. DOI: 10.1504/IJCAT.2007.015697.

[33] Kristina Shea, Robert Aish, and Marina Gourtovaia. "Towards integrated performance-driven generative design tools". In: *Automation in Construction* 14.2 (Mar. 2005), pp. 253–264. DOI: 10.1016/j.autcon.2004.07.002.

[34] M. David Albritton and Patrick R. McMullen. "Optimal product design using a colony of virtual ants". In: *European Journal of Operational Research* 176.1 (Jan. 2007), pp. 498–520. DOI: 10.1016/j.ejor.2005.06.042.

[35] Sam Kriegman et al. "A Minimal Developmental Model Can Increase Evolvability in Soft Robots". In: *GECCO '17 Proceedings of the Genetic and Evolutionary Computation Conference*. arXiv: 1706.07296. 2017, pp. 131–138. DOI: 10.1145/3071178.3071296.

[36] Karl T. Ulrich. *Product design and development*. New York : McGraw-Hill/Irwin, 2012.

[37] Hong-Zhong Huang, Ruifeng Bo, and Wei Chen. "An integrated computational intelligence approach to product concept generation and evaluation". In: *Mechanism and Machine Theory* 41.5 (May 2006), pp. 567–583. DOI: 10.1016/j.mechmachtheory.2005.07.006.

[38] Roman Žavbi, Nuša Fain, and Janez Rihtaršič. "Evaluation of a method and a computer tool for generating concept designs". In: *Journal of Engineering Design* 24.4 (Apr. 2013), pp. 257–271. DOI: 10.1080/09544828.2012.721539.

[39] J Rihtarši and R Žavbi. "SOPHY – Tool for Structural Synthesis of Conceptual Technical Systems". In: *DS 60: Proceedings of DESIGN 2010, the 11th International Design Conference, Dubrovnik, Croatia*. 2010, p. 8.

[40] Allen Corlies Ward. "A Theory of Quantitative Inference for Artifact Sets Applied to a Mechanical Design Compiler". In: (1988), p. 144.

[41] T Brockmöller. "Design Catalogue in a CAE Environment for the Illustration of Tailored Forming". In: *Engineering for a Changing World* Vol. 59 (2017), p. 13.

[42] Young-Hyun Han and Kunwoo Lee. "A case-based framework for reuse of previous design concepts in conceptual synthesis of mechanisms". In: *Computers in Industry* 57.4 (May 2006), pp. 305–318. DOI: 10.1016/j.compind.2005.09.005.

[43] Tolga Kurtoglu et al. "Deriving a Component Basis For Computational Functional Synthesis". In: *DS 35: Proceedings ICED 05, the 15th International Conference on Engineering Design, Melbourne, Australia*. 2005, p. 278–279.

[44] Jon-Michael Deldin and Megan Schuknecht. “The AskNature Database: Enabling Solutions in Biomimetic Design”. In: *Biologically Inspired Design*. Ed. by Ashok K Goel, Daniel A McAdams, and Robert B. Stone. London: Springer London, 2014, pp. 17–27. DOI: 10.1007/978-1-4471-5248-4_2.

[45] Amaresh Chakrabarti et al. “A functional representation for aiding biomimetic and artificial inspiration of new ideas”. In: *AI EDAM* 19.02 (May 2005). DOI: 10.1017/S0890060405050109.

[46] Ashok K. Goel et al. “Cognitive, collaborative, conceptual and creative — Four characteristics of the next generation of knowledge-based CAD systems: A study in biologically inspired design”. In: *Computer-Aided Design* 44.10 (Oct. 2012), pp. 879–900. DOI: 10.1016/j.cad.2011.03.010.

[47] Jacquelyn K.S. Nagel and Robert B. Stone. “A computational approach to biologically inspired design”. In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 26.02 (May 2012), pp. 161–176. DOI: 10.1017/S0890060412000054.

[48] René Lopez Flores et al. “Collaboration Framework for TRIZ-Based Open Computer-Aided Innovation”. In: *TRIZ – The Theory of Inventive Problem Solving*. Ed. by Denis Cavallucci. Cham: Springer International Publishing, 2017, pp. 211–236. DOI: 10.1007/978-3-319-56593-4_9.

[49] *Goldfire Cognitive Search | IHS Markit, https://ihsmarkit.com/products/enterprise-knowledge.html, accessed: 21-2-2019.*

[50] *Innovation WorkBench, http://www.whereinnovationbegins.net/software/, accessed 21-2-2019.*

[51] Seung-Hyun Yoo. “Sustainable Design by Systematic Innovation Tools (TRIZ, CAI, SI, and Biomimetics)”. In: *Handbook of Sustainable Engineering*. Ed. by Joanne Kauffman and Kun-Mo Lee. Dordrecht: Springer Netherlands, 2013, pp. 451–470. DOI: 10.1007/978-1-4020-8939-8_97.

[52] Yu-Hui Wang, Ching-Hung Lee, and Amy J.C. Trappey. “Modularized design-oriented systematic inventive thinking approach supporting collaborative service innovations”. In: *Advanced Engineering Informatics* 33 (Aug. 2017), pp. 300–313. DOI: 10.1016/j.aei.2016.11.006.

[53] Michael G Dyer, Margot Flowers, and Jack Hodges. “EDISON: An engineering design invention system operating naively”. In: *Artificial Intelligence in Engineering* 1.1 (1986), p. 36–44.

[54] Bin Chen and You-Bai Xie. “A computer-assisted automatic conceptual design system for the distributed multi-disciplinary resource environment”. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*. Vol. 231. Mar. 2017, pp. 1094–1112. DOI: 10.1177/0954406216638886.

[55] D.A. Hoeltzel and W.-H. Chieng. “Knowledge-based approaches for the creative synthesis of mechanisms”. In: *Computer-Aided Design* 22.1 (Jan. 1990), pp. 57–67. DOI: 10.1016/0010-4485(90)90030-G.

[56] D. A. Hoeltzel and Wei-Hua Chieng. “Pattern Matching Synthesis as an Automated Approach to Mechanism Design”. In: *Journal of Mechanical Design* 112.2 (1990), p. 190. DOI: 10.1115/1.2912592.

[57] Amaresh Chakrabarti and Thomas P. Bligh. “An approach to functional synthesis of solutions in mechanical conceptual design. Part I: Introduction and knowledge representation”. In: *Research in Engineering Design* 6.3 (Sept. 1994), pp. 127–141. DOI: 10.1007/BF01607275.

[58] Roman Žavbi and Janez Rihtaršič. “Synthesis of elementary product concepts based on knowledge twisting”. In: *Research in Engineering Design* 21.2 (Apr. 2010), pp. 69–85. DOI: 10.1007/s00163-009-0076-3.

[59] Luca Zimmermann, Tian Chen, and Kristina Shea. “Generative Shape Design Using 3D Spatial Grammars, Simulation and Optimization”. In: *Design Computing and Cognition ’16*. Ed. by John. S Gero. Cham: Springer International Publishing, 2017, pp. 279–297. DOI: 10.1007/978-3-319-44989-0_16.

[60] Hod Lipson. “How to Draw a Straight Line Using a GP: Benchmarking Evolutionary Design Against 19th Century Kinematic Synthesis”. In: *Late Breaking Papers at the 2004 Genetic and Evolutionary Computation Conference* (2004), p. 12.

[61] Santosh Kulkarni et al. “Evaluation of Collaborative Sketching (C-Sketch) as an Idea Generation Technique for Engineering Design”. In: *Journal of Creative Behavior* (2000).

[62] Milam Aiken et al. “Electronic brainstorming in small and large groups”. In: *Information & Management* 27.3 (Sept. 1994), pp. 141–149. DOI: 10.1016/0378-7206(94)90042-6.

[63] Yan Li et al. “Design creativity in product innovation”. In: *The International Journal of Advanced Manufacturing Technology* 33.3-4 (July 2007), pp. 213–222. DOI: 10.1007/s00170-006-0457-y.

[64] Ali Imre Aydeniz, Serdar Tumkor, and Ismail Fidan. “Teamwork And Gallery Method In Engineering Design Projects”. In: *Annual ASEE Conference* (2004), p. 8.

[65] Pedro Company et al. “Computer-aided sketching as a tool to promote innovation in the new product development process”. In: *Computers in Industry* 60.8 (Oct. 2009), pp. 592–603. DOI: 10.1016/j.compind.2009.05.018.

[66] H. James Harrington and Frank Voehl. *The Innovation Tools Handbook, Volume 3: Creative Tools, Methods, and Techniques that Every Innovator Must Know*. CRC Press, Sept. 2016.

[67] Johan Ölvander, Björn Lundén, and Hampus Gavel. “A computerized optimization framework for the morphological matrix applied to aircraft conceptual design”. In: *Computer-Aided Design* 41.3 (Mar. 2009), pp. 187–196. DOI: 10.1016/j.cad.2008.06.005.

[68] Yong Chen et al. “Automated Conceptual Design of Mechanisms Using Improved Morphological Matrix”. In: *Journal of Mechanical Design* 128.3 (2006), p. 516. DOI: 10.1115/1.2180807.

[69] Cheng-Hung Lo, Kevin C. Tseng, and Chih-Hsing Chu. “One-Step QFD based 3D morphological charts for concept generation of product variant design”. In: *Expert Systems with Applications* 37.11 (Nov. 2010), pp. 7351–7363. DOI: 10.1016/j.eswa.2010.04.028.

[70] Cari R Bryant Arnold, Robert B Stone, and Daniel A McAdams. “MEMIC: An Interactive Morphological Matrix Tool for Automated Concept Generation”. In: *IIE Annual Conference and Expo 2008* (2008), p. 6.

[71] Jami J. Shah, Santosh V. Kulkarni, and Noe Vargas-Hernandez. “Evaluation of Idea Generation Methods for Conceptual Design: Effectiveness Metrics and Design of Experiments”. In: *Journal of Mechanical Design* 122.4 (2000), p. 377. DOI: 10.1115/1.1315592.

[72] Margaret A. Boden. “What Is Creativity?” In: *Dimensions of Creativity*. 1994.

[73] Joel Lehman et al. “The Surprising Creativity of Digital Evolution: A Collection of Anecdotes from the Evolutionary Computation and Artificial Life Research Communities”. In: (Mar. 2018).

[74] Toshiharu Taura et al. “Constructive simulation of creative concept generation process in design: a research method for difficult-to-observe design-thinking processes”. In: *Journal of Engineering Design* 23.4 (Apr. 2012), pp. 297–321. DOI: 10.1080/09544828.2011.637191.

[75] John S. Gero and Mary Lou Maher. *Modeling Creativity and Knowledge-based Creative Design*. Psychology Press, 1993.

[76] John S Gero. "Computational Models of Innovative and Creative Design Processes". In: *Technological Forecasting and Social Change* 64.2-3 (June 2000), pp. 183–196. DOI: 10.1016/S0040-1625(99)00105-5.

[77] S. G. Tzafestas and H. B. Verbruggen. *Artificial Intelligence in Industrial Decision Making, Control and Automation*. Springer Science & Business Media, Dec. 2012.

[78] Steven John Winter. "Methodology development of an engineering design expert system utilizing a modular knowledge-base inference process". In: *Retrospective Theses and Dissertation* (1998), p. 163.

[79] Debasis Mitra. "Computational Creativity: Three Generations of Research and Beyond". In: *AAAI Spring Symposium: Creative Intelligent Systems* (2008), p. 7.

[80] Hod Lipson. "Curious and Creative Machines". In: *50 Years of Artificial Intelligence. Lecture Notes in Computer Science, vol 4850*. Lungarella M., Iida F., Bongard J., Pfeifer R., 2007.

[81] Kalyanmoy Deb. "Unveiling innovative design principles by means of multiple conflicting objectives". In: *Engineering Optimization* 35.5 (Oct. 2003), pp. 445–470. DOI: 10.1080/0305215031000151256.

[82] John R. Koza. "Human-competitive results produced by genetic programming". In: *Genetic Programming and Evolvable Machines* 11.3-4 (Sept. 2010), pp. 251–284. DOI: 10.1007/s10710-010-9112-3.

[83] Margaret A. Boden. *The Creative Mind, Myths and Mechanisms.* 2nd edition. 2003.

[84] Mary Lou Maher and Douglas H Fisher. "Using AI to Evaluate Creative Designs". In: *DS 73-1 Proceedings of the 2nd International Conference on Design Creativity*. Vol. 1. 2012, p. 10.

[85] Jonathan Hiller and Hod Lipson. "Automatic Design and Manufacture of Soft Robots". In: *IEEE Transactions on Robotics* 28.2 (Apr. 2012), pp. 457–466. DOI: 10.1109/TRO.2011.2172702.

[86] Martin Philip Bendsøe and Noboru Kikuchi. "Generating optimal topologies in structural design using a homogenization method". In: *Computer Methods in Applied Mechanics and Engineering* 71.2 (Nov. 1988), pp. 197–224. DOI: 10.1016/0045-7825(88)90086-2.

[87] Peter J. Bentley and David W. Corne. "An introduction to Creative Evolutionary Systems". In: *Creative Evolutionary Systems*. Elsevier, 2002, pp. 1–75. DOI: 10.1016/B978-155860673-9/50035-5.

[88] Gabor Renner and Aniko Ekart. "Genetic algorithms in computer aided design". In: *Computer-Aided Design* 35 (2003), pp. 709–726.

[89] Altshuller. *Creativity As an Exact Science*. CRC Press, Jan. 1984.

[90] *The Triz Journal, https://triz-journal.com/, accessed: 25-2-2019.*

[91] Jami J. Shah, Steve M. Smith, and Noe Vargas-Hernandez. "Metrics for measuring ideation effectiveness". In: *Design Studies* 24.2 (Mar. 2003), pp. 111–134. DOI: 10.1016/S0142-694X(02)00034-0.

[92] Paul-Armand Verhaegen et al. "A Variety Metric Accounting for Unbalanced Idea Space Distributions". In: *Procedia Engineering* 131 (2015), pp. 175–183. DOI: 10.1016/j.proeng.2015.12.368.

[93] John S. Gero and Udo Kannengiesser. "A Function-Behaviour-Structure ontology of processes". In: *In J. S. Gero (ed), Design Computing and Cognition'06*. Springer, 2006, pp. 407–422.

[94] Amaresh Chakrabarti et al. “A case for multiple views of function in design based on a common definition”. In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 27.03 (Aug. 2013), pp. 271–279. DOI: 10.1017/S0890060413000279.

[95] Zhengrong Cheng and Yongsheng Ma. “Explicit function-based design modelling methodology with features”. In: *Journal of Engineering Design* 28.3 (Mar. 2017), pp. 205–231. DOI: 10.1080/09544828.2017.1291920.

[96] Tufail Habib and Hitoshi Komoto. “Comparative analysis of design concepts of mechatronics systems with a CAD tool for system architecting”. In: *Mechatronics* 24.7 (Oct. 2014), pp. 788–804. DOI: 10.1016/j.mechatronics.2014.03.003.

[97] Albert Albers et al. “System Architecture Modeling in a Software Tool Based on the Contact and Channel Approach (C&C-A)”. In: *Journal of Mechanical Design* 133.10 (2011), p. 101006. DOI: 10.1115/1.4004971.

[98] John Clarkson. “Engineering Design Centre, University of Cambridge”. In: *Design process improvement*. Ed. by John Clarkson and Claudia Eckert. London: Springer London, 2005, pp. 470–473. DOI: 10.1007/978-1-84628-061-0_22.

[99] A Albers, N Burkardt, and M Ohmer. “How C&CM can help the Designer to find the Right Principles”. In: *Proceedings of CIRP Design Seminar 2005. New trends in engineering design*. 2005, pp. 232–241.

[100] S Matthiesen and A Ruckpaul. “New insights on the Contact & Channel Approach - Modelling of Systems with several logical states.” In: (2012), p. 10.

[101] Nizar Aifaoui, Dominique Deneux, and René Soenen. “Feature-based Interoperability between Design and Analysis Processes”. In: *Journal of Intelligent Manufacturing* 17.1 (Feb. 2006), pp. 13–27. DOI: 10.1007/s10845-005-5510-4.

[102] J.M. Jauregui-Becker, H. Tragter, and F.J.A.M. van Houten. “Structure and models of artifactual routine design problems for computational synthesis”. In: *CIRP Journal of Manufacturing Science and Technology* 1.3 (Jan. 2009), pp. 120–125. DOI: 10.1016/j.cirpj.2008.10.002.

[103] Couro Kane and Marc Schoenauer. “Genetic operators for two-dimentional shape optimization”. In: *Artificial Evolution*. Ed. by Gerhard Goos et al. Vol. 1063. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 355–369. DOI: 10.1007/3-540-61108-8_50.

[104] Richard J. Preen and Larry Bull. “Towards the evolution of vertical-axis wind turbines using supershapes”. In: *Evolutionary Intelligence* 7.3 (Nov. 2014), pp. 155–167. DOI: 10.1007/s12065-014-0116-4.

[105] John S. Gero. “Creativity, emergence and evolution in design”. In: *Knowledge-Based Systems* 9.7 (Nov. 1996), pp. 435–448. DOI: 10.1016/S0950-7051(96)01054-4.

[106] Marc Ebner. “Evolutionary Design of Objects Using Scene Graphs”. In: *Genetic Programming*. Ed. by G. Goos et al. Vol. 2610. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 47–58. DOI: 10.1007/3-540-36599-0_5.

[107] Riccardo Leardi. *Nature-inspired Methods in Chemometrics: Genetic Algorithms and Artificial Neural Networks*. Elsevier, Dec. 2003.

[108] J C Adcock et al. “Advances in quantum machine learning”. In: (2015), p. 37.

[109] Mauro Annarumma et al. “Automated Triaging of Adult Chest Radiographs with Deep Artificial Neural Networks”. In: *Radiology* 291.1 (Apr. 2019), pp. 196–202. DOI: 10.1148/radiol.2018180921.

[110] Alexander B. Graves. “System and method for speech recognition using deep recurrent neural networks”. US10199038B1. Feb. 2019.

[111] Kenneth O. Stanley and Risto Miikkulainen. “Evolving Neural Networks through Augmenting Topologies”. In: *Evolutionary Computation* 10.2 (June 2002), pp. 99–127. DOI: 10.1162/106365602320169811.

[112] Jonathan D. Hiller and Hod Lipson. “Multi material topological optimization of structures and mechanisms”. In: *Proceedings of the 11th Annual conference on Genetic and evolutionary computation - GECCO ’09*. Montreal, Qubec, Canada: ACM Press, 2009, p. 1521. DOI: 10.1145/1569901.1570105.

[113] Mark J. Jakiela et al. “Continuum structural topology design with genetic algorithms”. In: *Computer Methods in Applied Mechanics and Engineering* 186.2-4 (June 2000), pp. 339–356. DOI: 10.1016/S0045-7825(99)00390-4.

[114] Kenneth O. Stanley. “Compositional pattern producing networks: A novel abstraction of development”. In: *Genetic Programming and Evolvable Machines* 8.2 (June 2007), pp. 131–162. DOI: 10.1007/s10710-007-9028-8.

[115] Jimmy Secretan and Nicholas Beato. “Picbreeder: evolving pictures collaboratively online”. In: *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI ’08*. Florence, Italy: ACM Press, 2008, p. 1759. DOI: 10.1145/1357054.1357328.

[116] Carmel Majidi. “Soft Robotics: A Perspective—Current Trends and Prospects for the Future”. In: *Soft Robotics* 1.1 (Mar. 2014), pp. 5–11. DOI: 10.1089/soro.2013.0001.

[117] Karl Sims. “Evolving virtual creatures”. In: *SIGGRAPH ’94 Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM Press, 1994, pp. 15–22. DOI: 10.1145/192161.192167.

[118] Jonathan Hiller and Hod Lipson. “Dynamic Simulation of Soft Multimaterial 3D-Printed Objects”. In: *Soft Robotics* 1.1 (Mar. 2014), pp. 88–101. DOI: 10.1089/soro.2013.0010.

[119] Francesco Corucci et al. “Evolving Soft Locomotion in Aquatic and Terrestrial Environments: Effects of Material Properties and Environmental Transitions”. In: *Soft Robotics* 5.4 (Aug. 2018), pp. 475–495. DOI: 10.1089/soro.2017.0055.

[120] David Goldberg. *Genetic algorithms in search, optimization & machine learning*. 1989.

[121] Samir W Mahfoud. “Niching Methods for Genetic Algorithms”. PhD thesis. University of Illinois at Urbana-Champaign, 1995.

[122] Kenneth Alan De Jong. “An Analysis of the Behavior of a Class of Genetic Adaptive Systems.” PhD Thesis. Ann Arbor, MI, USA: University of Michigan, 1975.

[123] Georges Harik. “Finding Multimodal Solutions Using Restricted Tournament Selection”. In: (1995), p. 12.

[124] Fernández Natalia, Alfonso Hugo, and Gallard Raúl. “Crowding under Diverse Distance Criteria for Niche Formation in Multimodal Optimization”. In: (1989), p. 12.

[125] David Beasley. “A Sequential Niche Technique for Multirnodal Function Optimization”. In: 1.2 (1993), p. 25.

[126] A. Askarzadeh et al. “A population-based simulated annealing algorithm for global optimization”. In: *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Oct. 2016, pp. 004626–004633. DOI: 10.1109/SMC.2016.7844961.

[127] Joel Lehman and Kenneth O. Stanley. “Evolving a Diversity of Creatures through Novelty Searchand Local Competition”. In: *Proceedings of the Genetic and Evolutionary Computation Conferenc*. 2011.

[128] Jean-Baptiste Mouret and Jeff Clune. “An Algorithm to Create Phenotype-Fitness Maps”. In: (2012), p. 3.

[129] A. Albers and C. Schyr. “Model Based Validation of Automotive Powertrains”. In: *ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy* (2005), p. 2939.

[130] Jonathan Hiller and Hod Lipson. “Dynamic Simulation of Soft Heterogeneous Objects”. In: *arXiv:1212.2845 [physics]* (Dec. 2012). arXiv: 1212.2845.

[131] Nguyen Duc Thang. *1700 Animated Mechanical Mechanisms.* 2014.

[132] *Guinevere Flexible Shaft.*

[133] Henry T. Brown. *507 Mechanical Movements.* Google-Books-ID: CSH5UgzD8oIC. Courier Corporation, Aug. 2005.

[134] Noel Leon. “The future of computer-aided innovation”. In: *Computers in Industry* 60.8 (Oct. 2009), pp. 539–550. DOI: `10.1016/j.compind.2009.05.010`.

[135] Liping Zhang, Delun Wang, and Jian S. Dai. “Biological Modeling and Evolution Based Synthesis of Metamorphic Mechanisms”. In: *Journal of Mechanical Design* 130.7 (2008), p. 072303. DOI: `10.1115/1.2900719`.

[136] Gaetano Cascini. “Functional Modeling for TRIZ-based Evolutionary Analyses”. In: *DS 58-5: Proceedings of ICED 09, the 17th International Conference on Engineering Design.* Vol. 5. 2009.

[137] Leonid S. Chechurin et al. “Introducing Trimming and Function Ranking to SolidWorks Based on Function Analysis”. In: *Procedia Engineering* 131 (2015), pp. 184–193. DOI: `10.1016/j.proeng.2015.12.370`.

[138] Hongbin Kim, Junegak Joung, and Kwangsoo Kim. “Semi-automatic extraction of technological causality from patents”. In: *Computers & Industrial Engineering* 115 (Jan. 2018), pp. 532–542. DOI: `10.1016/j.cie.2017.12.004`.

[139] Ricardo Sosa and John S. Gero. “Computational Explorations of Compatibility and Innovation”. In: *Trends in Computer Aided Innovation.* Ed. by Noel León-Rovira. Vol. 250. Boston, MA: Springer US, 2007, pp. 13–22. DOI: `10.1007/978-0-387-75456-7_2`.

[140] F Freudenstein and E R Maki. “The Creation of Mechanisms According to Kinematic Structure and Function”. In: *Environment and Planning B: Planning and Design* 6.4 (Dec. 1979), pp. 375–391. DOI: `10.1068/b060375`.