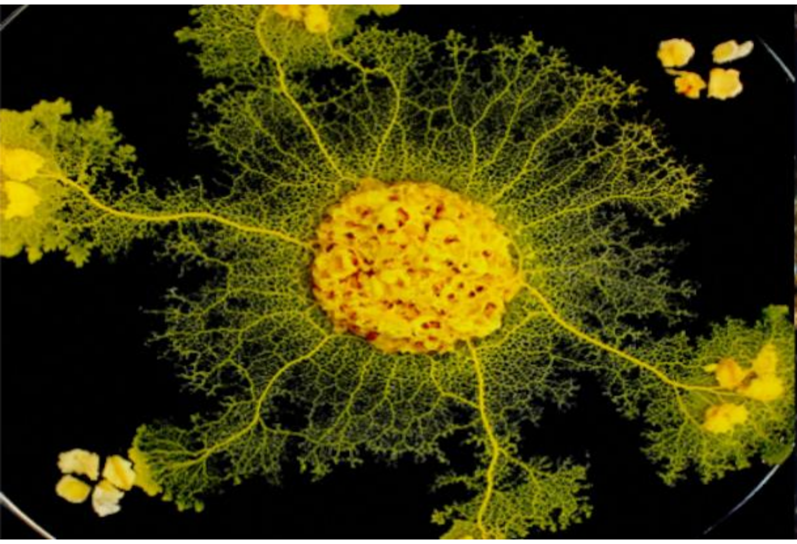
Multi-agent Models and the Design Process

M.Sc. Thesis E21 – Simon Muff Laporte



DTU Civil Engineering
Department of Civil Engineering

“Everyone spoke the same language, but they could not understand each other”

- Narrator, Metropolis (1927)

“Researchers took one of their most computationally advanced strains of DNA and grafted it onto Lactobacillus Delbrueckii, commonly used to ferment yoghurt.
Initial tests appeared to be failures.
However, one of the researchers sneaked some of the bacillus out of the lab to use for her homemade yoghurt.
The night of June 27th, it became sentient.”

- Narrator, When the Yoghurt took over. Love, Death + Robots (2019)

Acknowledgements

Many thanks to my supervisor Tim McGinley for his great engagement and guiding this thesis beyond just experimentation. Our discussions are what made this thesis.

I would also like to thank Nikolai Aasprong for his discussions and collaboration during our agile building design project in 2020.

A special thanks to Amanda Rasmussen for letting me work with her in Nottingham, trusting me to explore biology during the depths of the pandemic in 2020 and welcoming me in her lab with open arms and staying invested in my person even during lockdowns.

Abstract

Projects in AEC are becoming increasingly complex with design constraints involving several interdependent disciplines. Untangling the consequences of a decision takes collaboration between different types of knowledge, a process that is notorious for being time consuming and difficult.

This thesis investigates solutions to this problem by applying a novel **multi-agent algorithm utilizing reaction-diffusion mechanism** based on the Physarum Polycephalum algorithm by Jeff Jones. The focus was to develop a model to coordinate and solve multi-disciplinary design problems and to outline its place within the **design process.**

Throughout the thesis, an iterative design science research methodology (**DSRM**) was applied, consisting of three experiment iterations. Each experiment was evaluated qualitatively on the **solution objections**: Expressiveness of the algorithm, knowledge added by the algorithm, collaboration of disciplines and interaction options for a designer. Discrepancies between experiment results and solution objectives defined problems to be solved through the adaptation of concepts from biological development through a biomimetic process.

As a result of the research process, the thesis proposes a framework for developing a unified **collaboration algorithm**.

Following experiments were conducted:
Purpose: Proof-of-concept and algorithm development
Case study: Concrete and steel location in reinforced concrete cross section
Finding: Further expressiveness of algorithm was needed

Purpose: Engineering problem solving
Case study: Simplified bridge design in steel and concrete under uniform loading.
Finding: Algorithm returned bridge typology, but without regard for materiality.

Purpose: Multi-disciplinary architectural integration.
Case study: Multi-disciplinary architectural proposal of building.
Finding: A framework for a collaborative algorithm was proposed. The algorithm was able to integrate disciplines with defined boundary conditions using hierarchical relationships between disciplines.

Lastly, a guide for continuing the research was proposed.

Table of contents

Thesis Structure

This thesis is split into the following sections:

- **Introduction and motivation**
  This section introduces the problem motivating the thesis.

- **Problem statement**
  This section describes the overarching problem being investigated and defines the solution objectives for the experiments.

- **Background**
  This section investigates the state of the art of multi-agent design and biodesign and discusses how this thesis relates to the field. It also gives a brief introduction to the biological principles used as inspiration for the model.

- **Methodology**
  This section describes the research framework used throughout the thesis to arrive at results. A brief overview of the tools used in the thesis is also included in this section.

- **Experiment introduction**
  This section describes how the computational model works on a functional level and gives an overview of the experiments.

- **Experiments 1-3**
  The thesis is split into 3 experiments. Each segment will have an introduction, experiment results and a discussion of the experiment.

- **Discussion & Conclusion**
  This section discusses the overall performance of the experiments and relate them to the motivation. The thesis is then concluded by summarizing the findings

- **Further work**
  Topics for further research are proposed based on the motivation presented in the introduction

1 Introduction

Somewhere during the introduction for every civil engineering university degree, you are put in groups to compete to build spaghetti towers or paper bridges. While a great technical challenge, you quickly realize that the real challenge is not to design the best spaghetti tower but rather to share ideas and find good compromises among varying and strong opinions. This lesson only gets more important as you progress through school and jobs. Developers have different opinions than engineers, bosses have different opinions than employees, designers have different opinions than other designers. You start to wonder how anything ever gets built.

Sometimes you hear stories from your friends in biology of how civil engineering is so simplistic and engineers should try a real challenge; the complexity of life is orders of magnitude that of a building, yet designers still spend most of the project hours coordinating, while life just grows without any visible conflict. Maybe your friend is right, there is an obvious gap between how human builders and the builders of nature deal with complexity, and just shrinking that gap slightly would have huge benefits for the industry. You wonder what your old spaghetti tower would have looked like if it was cells rather than humans that had to coordinate.

This thesis is a result of my work during the autumn semester 2021, looking towards nature to solve the problems of multi-disciplinary design collaboration. It mainly investigates the application in construction of a multi-agent algorithm based on work around the slime mould Physarum Polycephalum by Jeff Jones.

The preliminary ideas behind the thesis were developed during the special courses Agile Building Design and BioDesign: Exploring opportunities in biology and engineering at DTU. Both courses had Tim Pat McGinley as supervisor.

The courses resulted in two entry points for the thesis. Firstly, the problem of multi-disciplinary collaboration in AEC-industry, which suggested a problem-focused entry. Secondly, collaboration with biological scientists as well as the course Agile Building Design led to early investigations of a promising model for slime mould looking for application, which suggested an opposite solution-first based approach. These two ideas were investigated together throughout the thesis.

2 Motivation

This section outlines the problem-based motivation for the thesis by outlining the issues with complex collaboration in AEC. Then a vision for what it would mean to solve the issue is presented along with avenues of how to get there. The solution-based motivation of the multi-agent algorithm is presented in the background section.

2.1 The problem of multi-disciplinary collaboration

Construction projects vary from the very simple and routine to the extremely complex. It is very easy to use intuition when building with low complexity. Most people with a sense for craftsmanship and the right tools can build a small tree house without too many complications. This stands in stark contrast to large scale complex buildings like the infamous Niels Bohr building, that as of writing is 2,5 bil. DKK over budget and 5 years delayed.

The two examples are subject to the same physical laws and the forces acting on the buildings are equally difficult to predict truthfully. So as a designer, what is the largest difference between building a wooden shag and the Niels Bohr building? I would argue that it is complexity, derived from the scale of the building, the number of interdependent disciplines involved, the number of stakeholders and the size of the monetary and societal risk.

To solve the problems stemming from complexity, they first need to be defined:
Problems from complexity can be defined as mapping of interdependent functions of different subsystems which can have unpredictable consequences. This results in issues that come in both a **social** and a **technical** form (Collopy et al., 2020).

Technical issues are intrinsic within multi-disciplinary design. Different designers have different goals, and each design decision will favour one designer over the others, even just with a miniscule preference. Hazelrigg states, that it is game-theoretically impossible for a group of rational designers to achieve a rational design. Every design must therefore be a compromise if only the goals of each individual designer is considered (Hazelrigg, 1997). This issue can be solved by having each discipline being defined in terms of a holistic goal. Defining a perfect holistic goal before a design process begins is however extremely challenging. Mapping the holistic goal to low level decisions is even more difficult.

Social issues are the coordination of tasks and knowledge flows between designers. Knowing the perfect distribution of tasks and having each designer understand both the intent and limits of other designers seem like a utopian dream. Lee describes AEC collaboration as a group of organizations with fragmented knowledge, collaborating for a short amount of time, this creates challenges for clear communication and transfer of knowledge (Lee & Jeong, 2012).

Obviously, designers are pragmatic and can intuitively solve these problems, but there is a limit to human cognition, which starts to be apparent when the projects get complex.

Let me set forward two extremes where designers do not act pragmatically. In the first extreme, designers have perfect social communication, but only try to fulfil their own technical goals, without consideration for the common holistic goal. No decisions will be able to be taken as every single decision favour one designer over the other, as even the slightest conflict in interest result in decision block. In the other extreme, where designers are unable to communicate, but care about a perfect holistic goal. The designers do not know the influence of each decision on other disciplines as the systems are interdependent, it will therefore be impossible to predict whether a decision will improve the design or make it worse.

A multi-disciplinary design must address both challenges. Building the wooden shag, the holistic goal of “owning a shag” is easy to maintain for a single designer/construction worker, while knowledge transfer between disciplines for a single person is as simple as thinking and drawing. A building like the Niels Bohr building has hundreds if not thousands of actors, with individual goals and knowledge, not all sharing the holistic goal o making it more prone to miscommunication, mistakes, and conflict.

This all creates risk that any rational developer tries to avoid, making them favour less complex projects over more complex projects, and by extension less innovative architecture.

2.2 Vision

This section presents a short scenario for what a perfect solution to the collaboration problem could look like. This vision then guides the design objectives that guide the thesis.

A solution to the collaboration problem will give the designer an intuitive understanding of the consequences of interdisciplinary interaction early in the design process. This needs to happen across levels of detail, as many problems does not arise before finer levels of detail are reached. Ultimately, the design should be limited by specifications, natural laws, and imagination rather than collaboration, scope, and time.

It is natural to imagine this process in terms of an interactive algorithm, where the designer takes a decision, and instantly sees the consequences in budget, effectivity, aesthetics, and risk. Allowing to iteratively design a complex building without considerations for time, risk, and collaboration. This brings the final design closer to the designer allowing for a better connection between design intention/specifications and the final design.

The design decisions should be able to be added in a parallel fashion, with each design input being untangled from one another allowing for the *hot swapping* of ideas.

The produced designs should be sufficiently informed by engineering analysis, and the necessary information should be readily available in a nontechnical language, so that the craftsmen always understand the engineering context and function of the individual element they are working on.

A designer is used loosely in this context as ideally a non-technical designer, who should be able to bypass the classic design cycle and derive actual design solutions from an intuitive human/computer interface.

The model should be general and adaptable, allowing every design case to be solved within the same framework, without having to set up a specific model each time.

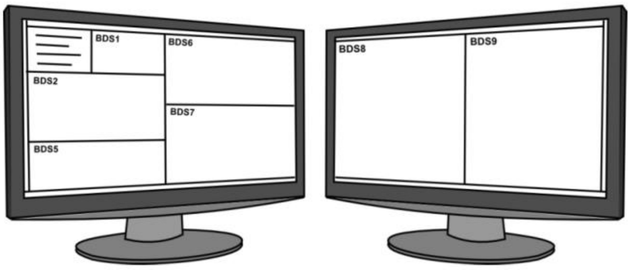


Figure 1. Design interaction across levels of detail with instant design feedback [1]

2.2.1 Translation into solution objectives

These are lofty goals, but they can be translated into continuous parameters, that can break down progress into smaller steps. A better solution scores higher on the parameters. This provides a range of solution objectives to use throughout the thesis and for future research.

- A model needs to be **expressive**, to provide solutions appropriate to the given design case, and not just be constrained to trivial solutions biased by a given model.

- A model needs to be able to work in several levels of detail. The designer should be able to make changes at any level of detail to see the impact on the final design, informed by engineering analysis. This could be as a nested model or parallel model or a single very advanced model. Thereby adding **knowledge** to the design process.

- A model needs to acknowledge the interdependence of disciplines, striving for a holistic goal rather than solving each issue independently. Thereby dealing the social and technical **collaboration** issues defined in the previous section. This requires human designers to outsource some of the negotiation process.

- Human/computer interfaces need to be simplified, such that intuitive human decisions have a meaningful impact on the final design. (Tim McGinley et al., 2015), allowing a designer to **interact** with the input and output of a potential model

It is important to note that while the solution objectives are as distinct from each other as possible, they can still affect one another. For example, it is not possible to evaluate the knowledge added by model if the model isn't expressive enough for the knowledge to show.

2.3 How do we get there?

The utopian idea of a system that can fully solve the design objhectives is obviously out of reach for our current understanding, and will require a drastic shift in perspectives and tools. A few different domains look promising for dealing with this type of complexity:

2.3.1 Computation

Computers and algorithms are the obvious starting point, as they deal with complexity in a very different way than humans, being able to cover the blind spots of human intuition and knowledge. Computer aided design (CAD) has long been an integrated part of documentation/description within AEC and is even being used for generation of design in rare cases. I would argue that this development is not finished, there are more domains within engineering design other than generation and description that the computer would be able to deal with. It seems clear that the future human designer will be increasingly augmented by machines. However, relying on computers also comes with disadvantages. Computers are by nature strictly rules based, which makes it difficult to deal with holistic design problems. This also means that for complex systems, computer programs are increasingly becoming “black boxes”, as humans are not directly able to fathom the complexity of what is going on inside. Both issues could be the starting point of design research.

2.3.2 Biology

While biology can be considered “rules based” as it is governed by natural laws, these rules are continuous rather than discrete as seen in computers. This allows for an almost infinite amount of complexity contained within a single water droplet expressed in terms of the fluid dynamic behaviour down to the smallest of scales. All life can be seen as continuous (analog) computers containing a specific program. The design of these computers and the emergent behaviour not from transistors, but from cells, is still being studied by biologists.

Nature is still able to design beyond the capabilities of humans, a power we should try to harness where appropriate. Biomimetics is the method used to adapt biological concepts to an engineering context. It seeks to abstract concepts found in nature, and reapply them to solve engineering problems, harnessing some of the ingenious solutions found in nature. However, divorcing a concept from its original context often results in misappropriation of the concept, which is a challenge biomimetic solutions must consider.

2.3.3 Computation & biology

The continuous and discrete nature of life and machine are often seen as each other's opposites. While this is true, computers are still the only practical tool humans have to simulate a biological system for a design process. Real biological systems could potentially also be adapted to design, which at the current state seems impractical.

3 Problem statement

This thesis investigates the following:
Can biologically inspired multi-agent models assist designers coordinate during the design process?

To investigate, the statement has been broken down to the following solution objectives, that are indications of a good solution to the problem statement as suggested in section 2 - Motivation:

Expressiveness - *Can a model express the needed geometry?*
A model should be able to provide solutions beyond the trivial. This means that the model needs to be expressive enough to return a broad range of solutions, depending on the inputs. A good model can provide geometry detached from biases within the model and return solutions specific to the problem.

Knowledge - *Can a model provide additional information not contained in the input?*
A good model should be able to add knowledge to a design problem such that each engineering challenge is being addressed or such that a designer has a better understanding of a decision than before.

Collaboration - *Can competing design objectives find a common solution?*
A good model should be able to allow agents to negotiate the space, to create solutions where the design choices of one discipline is not detrimental to the other, but the best compromise is found for a holistic goal.

Interaction - *How should a designer interact with a model?*
What is the input/output interface between the designer and the model? A good solution should provide an understandable human/machine interface that can be interacted with and does not lose the design intent during translation.

3.1 Scope

The thesis is focused on applying the multi-agent framework and figuring out what concepts it takes to incorporate it into the design process. This means that in-depth technical analysis and design is not included. Each individual experiment comes with its own individual scope and is solely focused of solving the presented case study and evaluate the results in terms of the solution objectives.

The development process for the model took many iterations, and many unsuccessful features are omitted from the thesis.

4 Background

This section gives an introduction to the theory informing the thesis. It presents biology within the design process, the multi-agent algorithm used for the thesis and the underlying biological concepts.

4.1 Biology and design

Construction is an infamously conservative field. Thousands of years of traditions and experience is hard to challenge with unproven technologies. Many construction projects are also a large expense for the stakeholder's incentivising adversity to risk. This all increases the inertia to change. However, change is needed more than ever; global demographic changes, environmental challenges and changing price of labour puts the whole sector under pressure.

4.1.1 Biomimetics

One of the idea-generation engines that can produce solutions to some of these problems is biomimetics (Many similar terms with almost similar meaning exist such as: biodesign, biomimetics, bionics and bioengineering (Pawlyn, 2016)).

Biomimetics is simply defined by Pawlyn as: "Design inspired by the way functional challenges are solved in nature" (Pawlyn, 2016) Which distinguishes biomimicry designs from designs that simply try to mimic the way nature looks.

Biomimetics have been formalized into the research process as either a top-down approach (technology pull), where a problem found in technology is the driving motivation or a bottom-up approach (biology push), where interesting properties found in nature drive the research (Speck & Speck, 2008). The top-down approach is used in this thesis for idea generation. It consists of identifying a **problem**, then searching for equivalent **concepts** in biology. **Abstracting** those concepts and finally **adapting** them to solve the original problem.

A full biomimetic process from initial research by biologists to a developed product often take many years and require designer and biologist to share a language for collaboration.

Mechanical design has a long tradition of using biomimetics in the process. One of the famous examples is Velcro that was inspired by how burr stuck to the fur of the inventor's dog.

Construction on the other hand has a harder time adapting biological concepts, without losing the function: Sagrada De Familia is one of the most impressive buildings ever constructed and is almost universally praised for its organic geometry, but there is also a reason why it has taken almost 139 years to build as of writing. Architecture tends to focus on the aesthetic component of biology rather than the functional.

My report from the special course "BioDesign: Exploring opportunities in biology and engineering" Fall 2020 explores both physical and design process applications of biomimicry and lists a range of current examples and research opportunities.

4.1.2 Beyond the physical – A morphogenetic approach

The most straightforward application of biomimetics is to take morphological features from biology and adapt them into physical engineering concepts. However, nature has much more to offer than just geometry and materials. Strategies and processes in nature can also teach us valuable lessons, the biomimetic method allows for non-physical adaptation, this could be processes such as different C2C systems or the evolutionary algorithm. However central is still detachment and abstraction of the biological concepts, and the focus on developing a narrow technology.

In the context of the parametric design process, many applications of the biomimetic method often end up being less parametric due to being focused on narrow novelty rather than a meta-design approach to benefit the field of design, resulting in the paradox of parametric models being more difficult to change due to the strict underlying logic as described by (Tim McGinley, 2022)

McGinley argues instead for a potential method of *morphogenetic prototyping*, that aims to extract the *genotype* of a design artifact through the metaphor of developmental stages, mapping features of the *phenotype* to an analogy of biological developmental features.
Ultimately, this broadens the application of a specific genotypic model, as the designer can alter steps in the development process to tailor the design for a specific context and use. Having a range of predefined genotypes (the architectural analogy would be building typologies) would then redefine the role of the designer to controller of the developmental process that generates design.

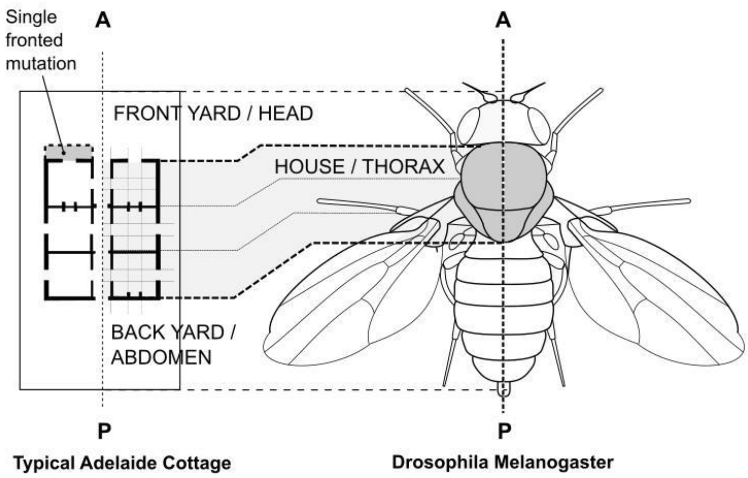


Figure 2. Body segment mapped to building typology. [2]

The developmental process relies on the logic of the body axes to define the segmentation and body plan. The segment is then controlled by sub-processes and sub-segmentation of the body plan to become the elements (organs) of the building.

While being an idealistic method yet to be implemented, morphogenetic prototyping still has provided concepts that has been appropriated by this thesis, in its criticism of the state-of-the-art and its views in the role of the designer:

This thesis also deals with a meta-approach of having a model that can be applied to broad range of design topic, proposing more general method to use in the design industry, rather than producing novel geometry for a single design problem. This process also happens through a process where the specific feature of a phenotype is articulated through morphogenesis-like process. Additionally, this thesis also examines the potential role of the designer within parametric design.

4.1.3 Design in nature

The thesis uses a top-down biomimetic approach to solve problems that arise during the development of the project. These concepts are appropriated from the processes during embryonic development.

The human design process has no direct translation in the natural world; however, embryonic development translates a genetic ideal blueprint to a real design. I will briefly describe the central concept in its natural context as a primer for the reader for understanding the concepts adapted.

Embryonic development is the process of how organisms develop their form and function through the differentiation and growth of cells and through morphogenesis, the process responsible for expression. This is obviously a very complex topic but can be explained very simplified as:

Cells are the fundamental units of life. The basic structure of cells is similar for all animals, from snails to humans. Cells produce proteins to take many different functions. Proteins can be building blocks, send signals, embed energy, regulate other cells and so on. Proteins are also used to decide the identity of a cell, whether a cell is a nerve cell or a skin cell.

A cell can potentially produce many different proteins but need to decide which ones to produce. This decision is coded in the genome of the cell. Here regulatory genes control lower-level genes that decide which proteins to produce. Some of the highest-level genes are the homeotic genes that decide the fundamental layout of the organism, such as “head goes here” and “tail goes here”, and then starts the process of making these elements without knowing how these elements might turn out further down the genetic chain.

This happens through the power of gradients and diffusion. The embryological development for most animals starts with the bicoid, a cell containing a gradient of bicoid protein concentration, effectively acting as a coordinate system within the cell. Each of the highest-level controlling genes has a different level of affinity for the bicoid proteins, resulting in different genes being expressed at different concentrations of the bicoid protein. These genes in turn produce their own respective signalling proteins, furthering the segmentation of the initial embryo, until a rough body plan has been established, with different homeotic genes controlling the development of each segment (Casares, 2000). The development of the embryo then becomes complex interactions of local protein concentrations and a hierarchy of expressive and controlling genes, at which point morphogenesis takes place to express the elements of the organism.

The difference between humans and single cell amoeba is physically speaking fully contained within the genome. The difference between the two is a result of billions of years of evolution.

4.2 Agents and emergence

Emergence is the property of a system that is only visible when observing the system as a whole, and not its individual units. These units can be individually acting agents, each able to make independent decisions affecting the system. This can be seen in the stock market, the design process (the process only emerges as a result of the individual contributions), city development, the Internet, and many other fields. Emergence can also be found without agents. Rocks subjected to natural forces over many years will over time create patterns, unremarkable if not viewed as a whole, but stunning with an overview, despite being inanimate. The perhaps most impressive example of emergence is the development of intelligent life made up of billions of tiny agents, the cells.

*Figure 3. Examples of emergent patterns (**Left)** Snow flakes [3] **(Middle)** Bank run of the 1932 financial crash [4] **(Right)** The slime mould Physarum Polycephalum [5]*

Modern computing allows the creation of complex multi-agent simulations, with several applications within AEC already. For example, can each human be considered an agent when evacuating a burning building, each person takes independent decisions to save their own life resulting in an emergent behaviour if everybody is considered together as a group. This emergent behaviour can be studied and used to verify existing design or guide future design decisions. The algorithms are possible because most people react in a predictable manner during panic situations (Almeida et al., 2013). These models are also quite robust as outlier behaviour by individuals do not affect the emergence of the group as a whole, due to large numbers.

City planners also make extensive use of multi-agent systems, when traffic intensity is being evaluated, again with humans as agents, each with individual needs, jobs and families (Doniec et al., 2008). (Not too different from many popular city-planning computer games such as SimCity and Cities: Skylines)

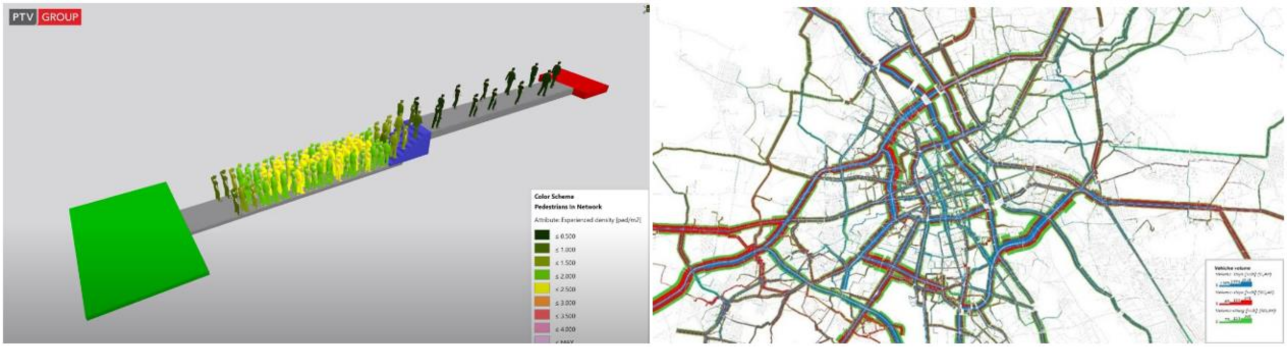


*Figure 4. Examples of commercial multi-agent software used in AEC **(Left)** PTV Viswalk, software for evacuation simulations [6]. **(Right)** PTV Visum, traffic simulation software [7].*

Common for both use-cases are the dynamic systems involving many variables that would be impossible to evaluate analytically. Game theory studies the behaviours that occur, when self-optimizing agents interact which could be a potential future overlap of research.

4.2.1 The slime mould Physarum Polycephalum

In 2010 the world was stunned as Japanese scientists were able to show a culture of the slime mould Physarum Polycephalum that was able to solve a shortest path problem through a maze, and later emulate the rail network of Tokyo, by connecting nodes of food representing real-life stations in the same way that humans have designed the routes. (Tero et al., 2010)

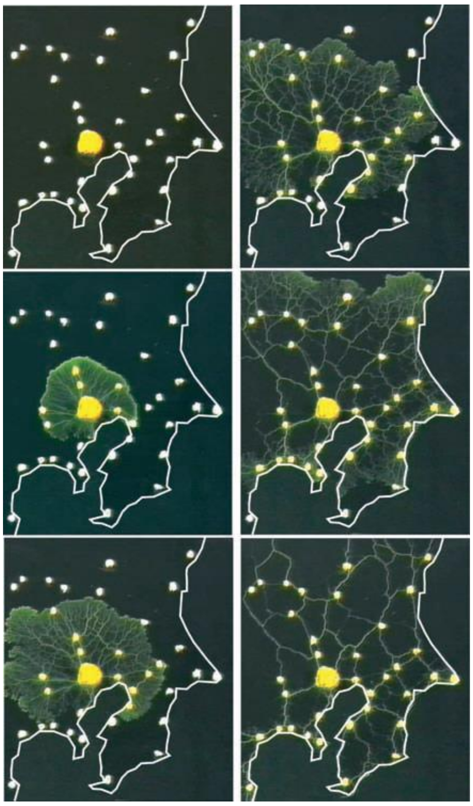


Figure 5. Nutrition located at Tokyo rail stations; the Physarum culture forms edges topologically equivalent to the existing rail network [8]

Physarum Polycephalum is a slime mould meaning a collection of single-celled, self-sufficient organisms that together form a multicellular superstructure. The slime mould forages for nutrition, that then are transported through the culture by the formation of tubes. The network of tubes is dynamic and can adapt to the environment. These tube networks are often laid out considering the relative size and location of nutrition sources, forming a minimal network, using the least amount of cells to transport the nutrition around the network. (Jones, 2015)

These tubes are randomly initialized to forager for nutrition in the environment. However, once a tube connects to a nutrition source (chemo-attractant), an optimization of the network occurs. Tubes that are not connected to nutrition sources disappear, while tubes connected to nutrition grow. Over time shorter tubes persist over longer tubes. This results in the mass of the colony being shifted towards the nutrition sources. The opposite effect occurs if the slime mould encounters chemo-repellents. As a result, the colony creates a shortest path between nutrition sources, while foraging for new nutrition sources (Jones, 2015). These were the properties that (Tero et al., 2010) exploited to generate the Tokyo rail system.

The Unconventional Computing Lab at University of the West of England under Andrew Adamatzky, work among other things with biological and chemical computers. The lab has worked with Physarum for many years exploring the computational properties of the slime both *in vitro* and *in silico,* with the goal of being able to produce a robust biological computer (Schumann & Pancerz, 2016). One of the findings is Jeff Jones' simple algorithmic approximation of the behaviour of the slime mould (Jones, 2015)

4.2.2 Jones' Algorithm - A computational model for biological emergence

Describing any biological phenomena algorithmically is always going to be an approximation. This model is focused on the morphological adaptation of the slime mould to chemo-attractants/repellents, with each cell being represented as an agent. The emergent pattern will exhibit a similar foraging and adapting behaviour.

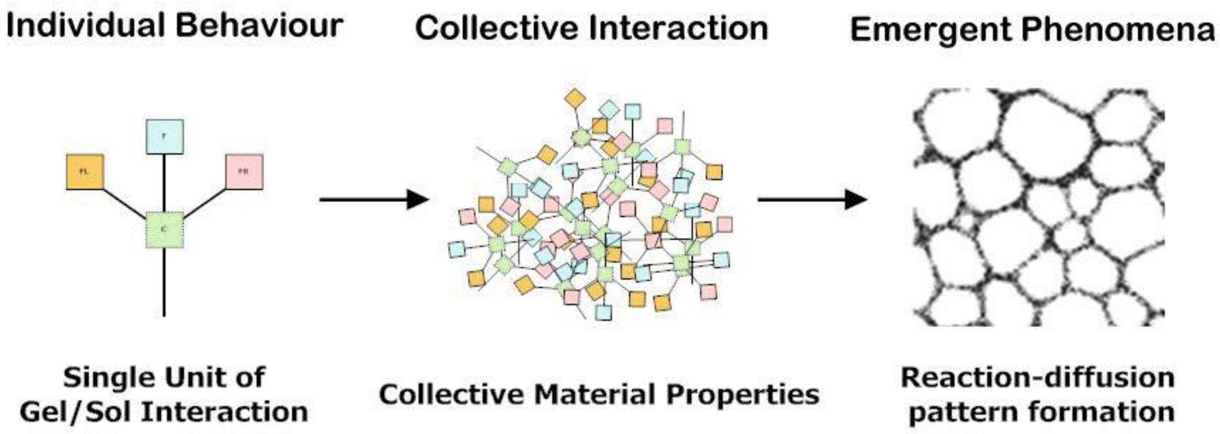


Figure 6. Emergence from simple rules [9]

These agents exist within a position matrix, having both a floating point and a discreet position. The agents interact with a chemical matrix. Here chemicals are diffused, allowing information to disperse. Concentrations of chemicals are embedded on the matrix as numerical values. The environment is included as an environment matrix which contains chemo attractant/repellent sources, to represent nutrition and adverse conditions (such as direct light or high temperatures), this layer adds or removes chemicals in the chemical matrix.

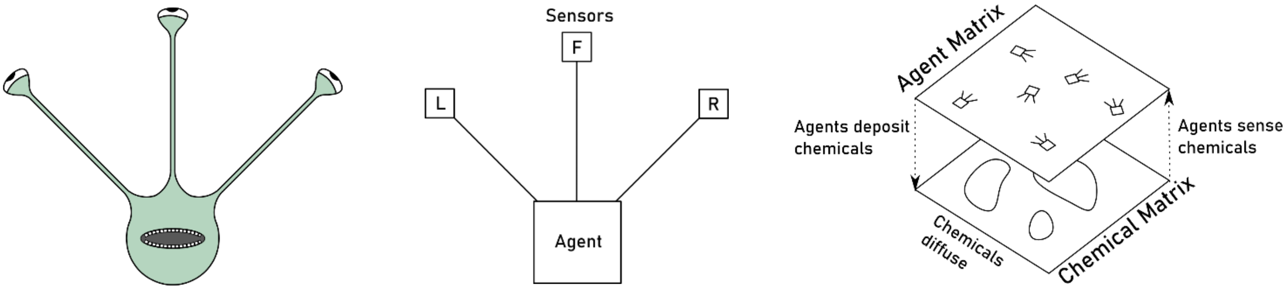


*Figure 7. Physarum model overview. **(Left)** An agent **(Middle)** Computationally idealized agent. **(Right)** Agent and chemical matrices.*

Agents go through three actions for each timestep. First the agent senses the chemical matrix with three sensors and will adjust its direction towards the sensor with the highest chemoattractant value. Then the agent will attempt to move forward, if the destination is empty, it moves successfully. If the destination is occupied, the agent chooses a random new direction. Lastly, if the agent was able to move successfully it deposits chemical attractors into the chemical matrix. The model can also be set up to include conditions for cellular division and death by overpopulation.

This model was able to solve a range of similar problems to the physical mould, such as convex hulls, concave hulls, Voronoi diagrams, shortest path problems, and logic gates among others. The simplicity of interaction with the model is one of the advantages, one simply has to provide the locations of nodes in the form of a drawing rather than programming the whole model.
The model also comes with downside: the chemical diffusion relies on a 3x3 kernel filter applied at every timestep, which is computationally expensive, when the algorithm is scaled up.

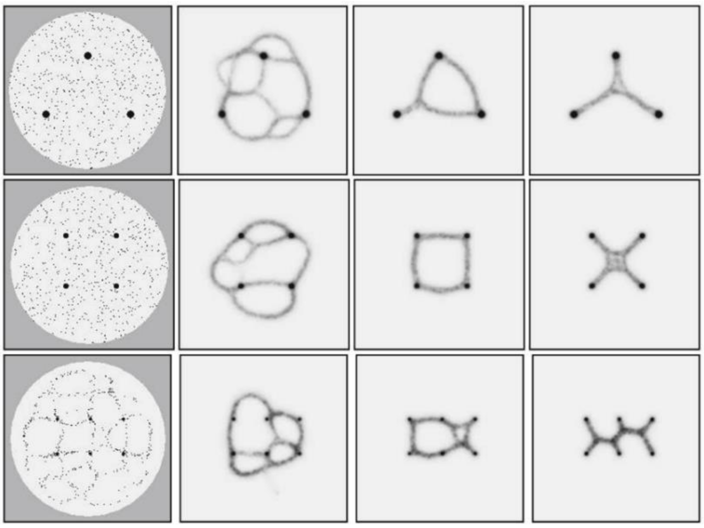


Figure 8.Steiner minimum spanning trees from Physarum algorithm [10]

4.2.3 Applications of multi-agent systems in design

Emergence is being increasingly explored within engineering design, most often as a result of non-agent particle simulations such as mesh relaxation and funicular shells. Multi-agent models have been used for spatial layouts within architecture or as optimization engines within engineering.

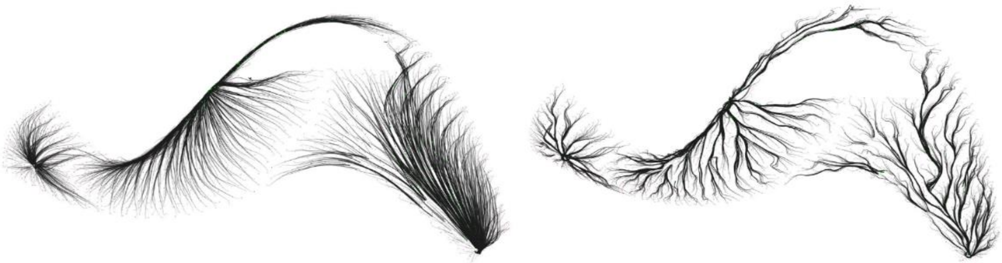


Figure 9. Agents reacting to Mexico City contamination data, here both with and without self-sensing behaviour [11]

Within architectural research, a range of different model types are used. These are pheromones, flocking, physics, least mean square fit, spectral layout, heuristics, metaheuristics, neural nets and Markov chains (Veloso et al., 2019). These can be combined with different representation forms and design objectives.

Selected articles about architectural multi-agent design are included in Table 1:

Article	Model features	Use case
(Pantazis & Gerber, 2018)	Non-physical agents, Designer interaction, Differentiated agents, Genetic Optimization	Facade generation
(Klemmt et al., 2019)	Environment forces Agent division	Generative sculpture, Mesh generation
(Gerber & López, 2014)	Agent - environment interaction, Neighbour sensing	Mexico City - Infrastructure Network - Soil salinity as chemo repellent
(Meyboom & Reeves, 2013)	Pheromone based ant colony, Agent - environment interaction, Chemo-attractants, Environment diffusion	Multi-level building, Space generation from building programme

(Chalmers, 2008)	Agent - environment interaction Neighbour sensing, Chemo-attractants	Cell aggregates within site boundary

Table 1. Multi-agent research in architecture

Common for all is that the agents have an internal decision process based on outside factors.

Jeff Jones' Physarum Polycephalum model falls under the category of a pheromones-based model, with the chemo-attractor being equivalent of pheromones. The Physarum model has recently attracted attention from artists and architects with many different approaches being tried, but is yet to be appreciated by engineers (Adamatzky, 2019).

4.2.4 Discussion of current research

It is still early in the development of multi-agent models for design (the evaluation-based models in traffic and evacuation have already matured). Whether the algorithms will have a place in the design process, and how they will assist the designer is still unknown.

Parameter	Expressiveness	Knowledge	Collaboration	Interaction
Score				
Comment	The research puts forward many different types of expressions, but none are able to deal with varying demands, and is solely controlled by the bias of the algorithm.	Most research focuses on a solely aesthetic appropriation without regard for function.	No noteworthy interactions have been investigated.	Papers have used site boundary conditions as an input interface, but no robust output interface has been presented.

Table 2. Evaluation of solution objectives for current research.

Common for all the algorithms is the focus on the narrow adaptation of biological concepts, and not on as a driver for the design process itself, focusing solely on the phenotype and not on the genotype to frame in a morphogenetic context. This thesis sets out to investigate the use of multi-agent algorithms in the context of design process itself, rather than the use for a narrow design, to understand how the designers of the future can be assisted to handle complexity in design and collaboration.

To do this, the challenges of the current research put forward need to be addressed.
The current challenges can be simplified in terms of the solution objectives as seen in Table 2

5 Methodology

5.1 Research framework

The thesis resides within the overarching framework of design research methodology (DSRM) (vom Brocke et al., 2020). the framework suggests an artifact driven iterative process with predefined steps. Three iteration cycles were conducted, time constrained to about one month each.

1. **Motivation & problem -** The motivation for the thesis is presented. The problem description led to the solution objectives through the use of a vision for a perfect solution.
2. **Solution objectives -** Solution objectives are presented, in the context of this thesis, they were used to evaluate the usefulness of a model. Solution objectives are expression, knowledge, collaboration, and interaction.
3. **Design and development -** A design model was developed to fulfil the solution objectives. The development of the model was aided by a top-down biomimetic approach to generate solutions to problems defined in the evaluation of the previous iteration.
4. **Demonstration -** A case study was developed to test the features of the developed design model. Each case study was based on the design process within the construction sector. First case study was designed as a rough proof-of-concept, second case study was designed to investigated engineering problem solving, while the third was designed to investigate architectural integration.
5. **Evaluation -** The results of the experiment were evaluated based on the solution objectives. The discrepancies were used to inform the next iteration of the model.
6. **Communication -** Research results were published with this thesis. Code is published on GitHub under MIT license.

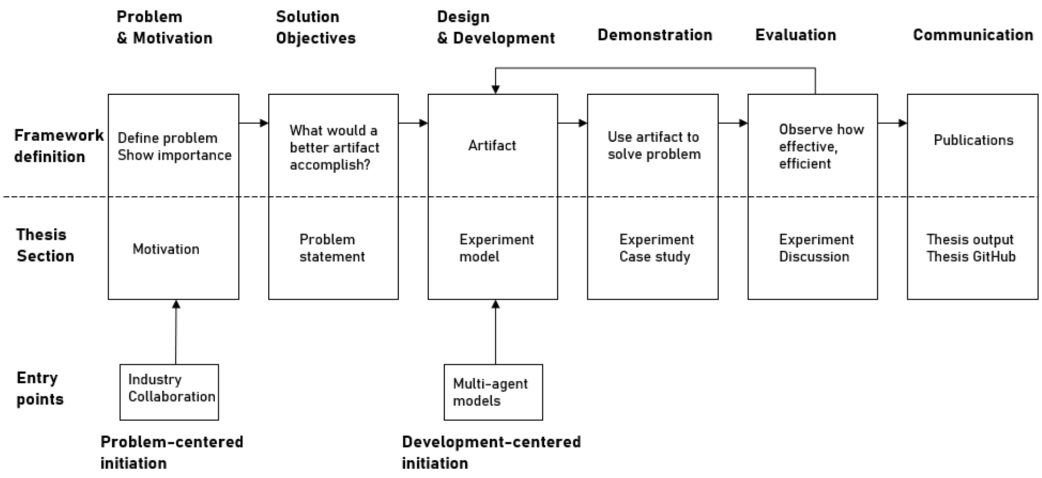


Figure 10. Research framework of the thesis. Adapted from (vom Brocke et al., 2020).

5.2 Tools

The thesis was developed within the Grasshopper3D environment within the CAD programme Rhino3D. The computational parts of the thesis were developed as plugins for Grasshopper3D. This was done in C#, due to the native C# API for Grasshopper3D and the computational speed advantages of a statically typed programming language such as C# in comparison with a dynamically typed language such as Python. Computational speed was important for simulation heavy aspects of the thesis.

The compiled plugins could interface with geometry and further technical analysis within the Grasshopper3D environment. Each experiment has a unique Grasshopper3D plugin, containing of a few components. Note that due to GUID conflict, only one experiment plugin can be installed at a time.

The analysis was supported by the native component library and following Grasshopper3D plugins:

- Karamba3d for finite-elements analysis
- Ladybird for ray-trace analysis
- Squid for generation of image files

All code and developed grasshopper plugins can be found on the thesis GitHub:

https://github.com/SimonMLaporte/Multi-Agent-Thesis

6 Experiment introduction

This section presents the algorithm used in the experiments. These experiments were derived iteratively but will be presented together for an overview.

Every experiment was based on Jones' Physarum as presented in section 4.2.2, but with a few important alterations. The algorithm was extended to allow several "types" of agents. Instead of a universal chemo-attractant, the simulation contained several chemical matrices with separate chemicals. Each chemical had a corresponding cell type that could represent different materials, disciplines or building elements. This was to allow for several optimizations to happen in parallel within the same space and for the agents to pursue different goals.

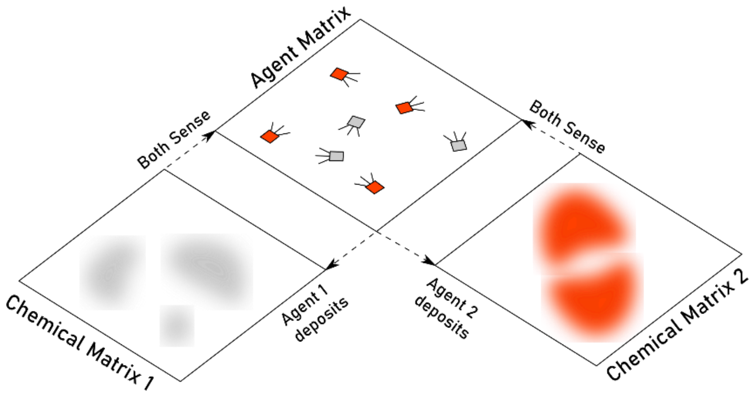


Figure 11. Model with several chemical layers.

Additionally, the agents were also supplied with a range of settings. These settings influenced the way the agent interacted with the environment. This created a hierarchy between the types of agents and allowed for negotiations of space. The settings were unique to each type of agent and was what differentiated them.

The algorithm ran at a predetermined number of timesteps or was stopped at convergence. The agent positions at the end of the simulation could then be further analysed for the needs of each experiment. Values of the chemical matrix were also available to the designer.

Initially in the experiments, these settings were derived using a genetic algorithm, however in the later parts of the thesis, it was deemed more useful to predetermine the settings, based on the needs of a designer. This decision was taken as the search space of *useful settings* was too narrow in comparison with the full search space, creating a computational bottleneck. Additionally, too many settings did not create a fitness gradient towards *useful settings*.

The goal of the experiments was to connect the computational biological model with the goal of collaboration of designers through two intermediate experiments, that elaborated the ideas.

Three experiments were conducted in total, each with a different case study of increasing complexity.
Each case study features different types of agents, with different goals. Experiment 1 designed a reinforced concrete cross-section for bending by having an agent represent reinforcement steel and another agent representing concrete. Experiment 2 designed a bridge in steel and concrete under uniform loading. Experiment 3 elaborated a design proposal by finding locations for rooms,

structural components, and ventilation ducts, with each agent representing architectural goals, structural goals, or ventilation goals. The findings of each experiment informed the model for the next experiment

	Experiment 1	Experiment 2	Experiment 3
Timeframe	September	October	November/December
Goal	Proof of concept	Engineering problem solving	Architectural integration
Case	RC cross section	Bridge	New DTU building 127
Features	Multi-agent algorithm Gene logic Genetic algorithm	Environmental interface Genetic algorithm Functional genes	Gradient maps Graphs Meta model
Result			

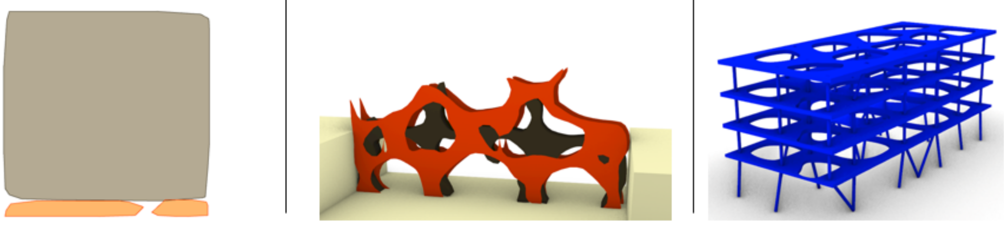


Figure 12. Overview of experiments

7 Experiment 1

The experiment was an initial proof-of-concept to develop the algorithm and to work out how the data should flow through the programme before more ambitious case studies were conducted. It investigated a reinforced concrete section with a single agent type representing one of two materials: reinforcement steel and concrete.

This experiment developed an initial version of the algorithm. The corresponding Grasshopper3D plug-in on the thesis GitHub is “Experiment1.gha”, which should be used with the Grasshopper3D file “ScenarioConcrete.gh” and the file “Empty.bmp” for simulation environment.

7.1 Algorithm

The algorithm has a single agent type with a range of settings. These settings are contained within a reconfigurable “genome”. The genome represents the decision-making process for each agent.

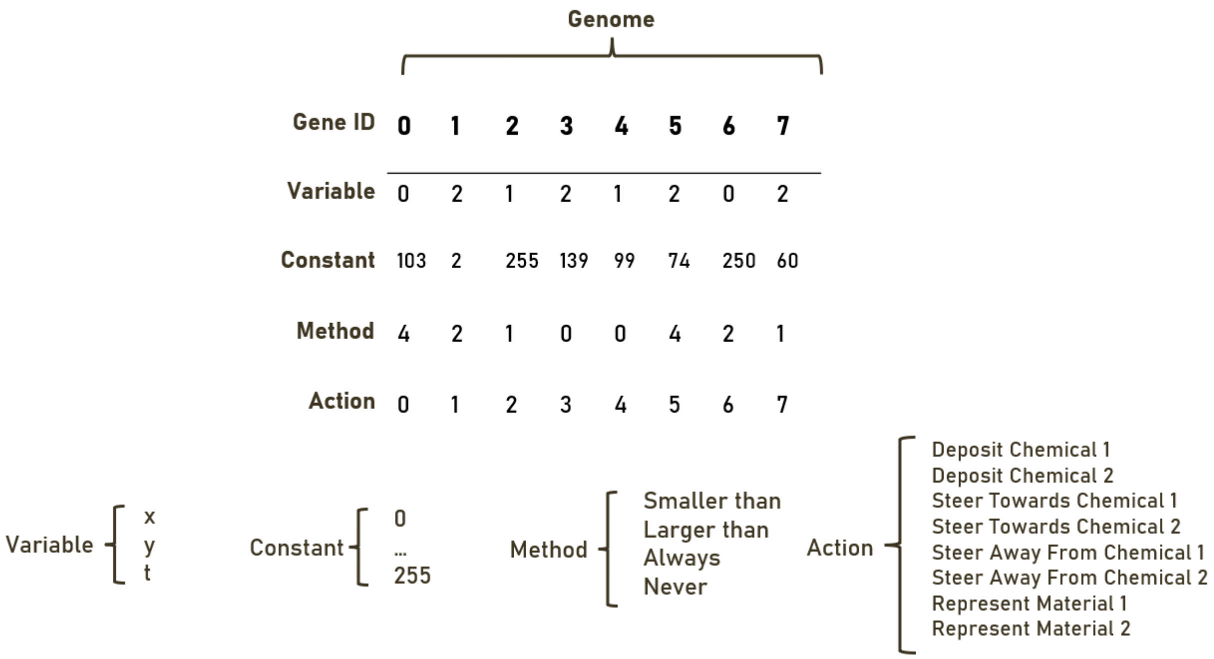


Figure 13. Example of a genome setup for genetic algorithm.

Each gene within the genome represents an action the agent could take. At each timestep of the simulation, it is decided whether or not the agent will take the given action.

For example, *geneID 2* in Figure 13 states that the agent will take *Action 2* (Steer Towards Chemical 1) if *Method 1* (larger than) is true for *Variable 1* (y location) and the *constant value 255*. The methods *Always* and *Never* bypass comparison.

Notice that opposite actions can be true at the same time, such as “*Steer Away From Chemical 1*” and “*Steer Towards Chemical 1*”, which would simply cancel out.

Constant values, variables and comparison methods are the genes that are under the control of the genetic algorithm.

The multi-agent simulation was run in a 200x200 empty environment consisting of 1500 agents containing the same genome, and the emergent pattern were then evaluated with a fitness function. The fitness function guided a genetic algorithm, that mate and mutate each genome, to produce the next generation of the genomes, more likely to produce a result with high fitness. A rank-based selection method was used. Genes were combined using single point crossover.

Hyperparameters were: Population size of 50 genomes per generation, mutation rate of 0,01 and elitism of 2. These were found through experimentation and a trade-off of between computation and complexity.

The location of each point after 150 timesteps were evaluated. A convex hull was drawn around the points of each material to convert the point cloud into a surface to analyse.

7.2 Case study

To evaluate the algorithm, I choose the moment resistance of a concrete cross section as the fitness. The cross-section was evaluated assuming full shear connection between parts, and asymmetric effects ignored. Furthermore, steel was always considered yielding. These approximations were very rough and were a guide for the experiment, rather than actual design recommendations.
The fitness was then calculated as moment resistance per weighted area:

$$Fitness = \frac{M_{Rd}}{WA_s + A_c}$$

With W being a constant factor between the price of a unit area of concrete and a unit area of steel, to incentivise a clever material distribution. Here steel is 32 times as expensive as concrete. M_{Rd} is the design moment resistance in kNm, A_s is steel area in m^2 and $A_c$is concrete area in m^2.
A custom numerical algorithm was developed to estimate the moment resistance for an arbitrary non-rectangular cross-section.

This scenario was chosen, as there is a clear geometrical solution. A good solution has a small amount of steel at the bottom of the section to take tension, and a larger amount of concrete at the top of the section to take compression, as steel is more expensive to utilize for compression.

To be able to evaluate the genome, each agent, represented as a point in space, was grouped based on proximity to each other and material, and a convex hull was drawn around the point group. This was considered the phenotype to the genotype contained by the genome.

The found solution was compared with the fitness of a simple rectangular reinforced concrete cross-section of 1000x800mm and A_s = 2000mm^2 with a fitness of 0,859.

7.3 Results

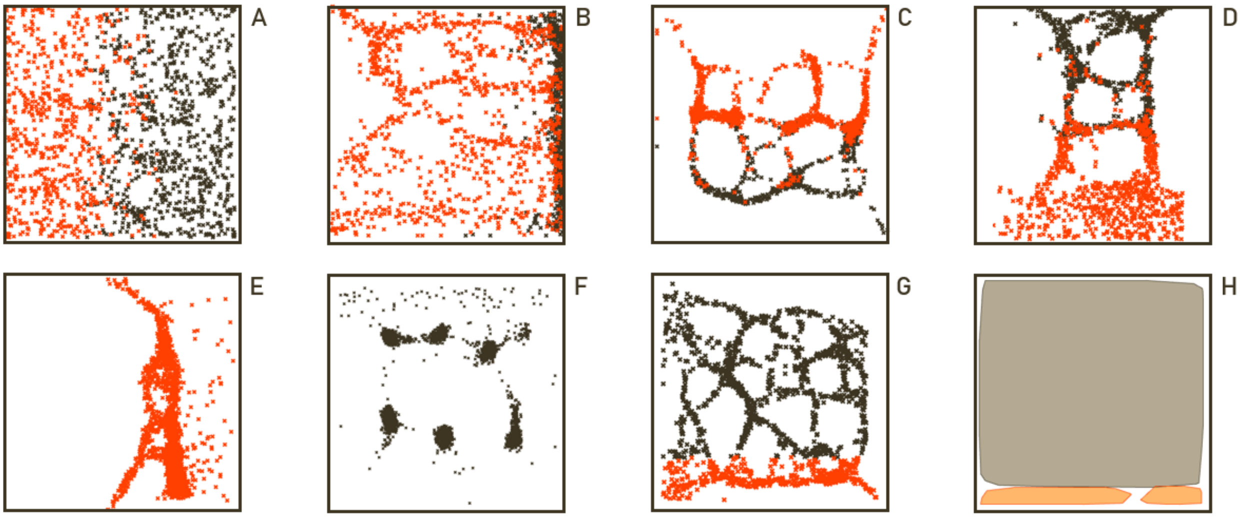


*Figure 14. Selected motifs from simulation results. (**H**) Represents a phenotype, drawn by convex hulls around groups of close points.*

The setup of genome and optimization was able to produce a range of different geometric motifs as seen in Figure 14. A-G These range from very chaotic, where no emergent patterns show, to the very organized.

The best phenotype is shown in H, which showed the expected result after only a few generations, resulting in a fast convergence with a fitness of 1,684, almost doubling the fitness of the benchmark cross-section, this is not a sign of clever design, but simply achieved by placing reinforcement lower than what would be possible in the real world as well as having a better balance between concrete and steel than the benchmark. Common for all successful genomes were that they all expressed two materials, differentiating on the y axis, which is what is required for moment resistance.

7.4 Discussion

The model showed some ability to solve a problem, as the fitness function was maximized resulting in a plausible phenotype, however the chemical diffusion interactions and emergent patterns didn't seem to have much effect.

1. There was no noticeable difference between the fitness measured after the first timestep and the 150th timestep, the fitness was almost solely decided by whether or not the genome had a vertical split between concrete and steel, removing the purpose of the multi-agent simulation. A better phenotype would see the distance between the centroid of the two materials increase. This could be achieved if concrete was only located at the top with a gap to the steel at the bottom. However, this behaviour was never observed in any of the genomes suggesting that steel and concrete will have to act with separate rulesets, allowing each material to fulfil their theoretical potential, acting as separate agents with separate goals.

2. The discretization methods of convex hulls did not allow for finer geometrical expression, and there was often a large gap between the agent locations to the phenotype. This can be solved by changing the way each agent is represented as an analysable mesh. This can be a Delaunay mesh or a bin-lattice.

3. Additionally, the model was very heavy computationally, with a uniform 3x3 median filter performed on all pixels in the chemical matrices at each timestep slowing down the algorithm such that each generation took more than 20 minutes. Convergence happened after relatively few generations, as it is known that a better solution exists, a larger population size combined with more computational power could potentially give better solutions. This can be achieved by scaling the computational power or optimizing the algorithm, using a histogram-based approach rather than a conventional median filter. ($O(n)$ vs $O(n^2 \log(n))$ (Bae & Yoo, 2018)), where n is the size of the window, here 3 resulting in potentially. 40% speed increase)

None of the solution objectives were dealt with during the initial experiment, prompting additional development. However, the computational framework was prepared for the rest for the rest of the experiments.

7.5 Summary

Parameter	Expressiveness	Knowledge	Collaboration	Interaction
Score				
Comment	Model provides a range of different motifs but is unable to differentiate the different materials.	The multi-agent algorithm does not add knowledge as the fitness was solely determined by the parameters within the genome.	No noteworthy interactions between agents were observed.	The convex hull interpretation disconnected the solutions of the model to the output seen by the designer.

Table 3. Evaluation of solution objectives for experiment 1.

7.6 Abstraction

To find solutions to the problems outlined by solution objectives, biomimetic method suggests looking at equivalent biological systems within embryonic development. The most urgent of these issues was expressiveness, as it hindered progress of the other solution objectives, as they rely on expression of an artifact to be evaluated.

Problem:
Model is unable to differentiate between materials and lacks expressiveness.

Biological analogies:
Embryonic development is about expression. Concepts can however still help satisfy other solution objectives, as expression is achieved as a result of collaboration. To further the development of the model the following concepts were identified:

Cell differentiation is one of the key concepts that are needed to create patterns and interaction. The cell differentiation is driven by **body segmentation**. Body segmentation first occur because of the 1d bicoid gradient, resulting in the anterior/posterior axis.

This cell differentiation is expressed by **differential growth**, allowing each segment to subdivide into organs of different sizes.

Higher-level proteins control the development of the lower-level proteins, providing a canvas for them to differentiate.

Abstraction:
Removing the concepts from their native context, left the following:

- Cells can be considered agents. Each agent type act differently within the environment depending on conditions, to form the features of a certain part of a design artifact.
- Segments control the elements of a design artifact, and the order of those based on axes, first derived in 1D. The direction of these axes depends on the artifact.
- Differential growth determines proportions and provides robust interfaces between elements, as the difference in growth speed is gradual, thus resolving disjointed geometry.
- High level/low level proteins create a hierarchy of interaction. With the high-level chemicals acting as boundary conditions for the lower levels.

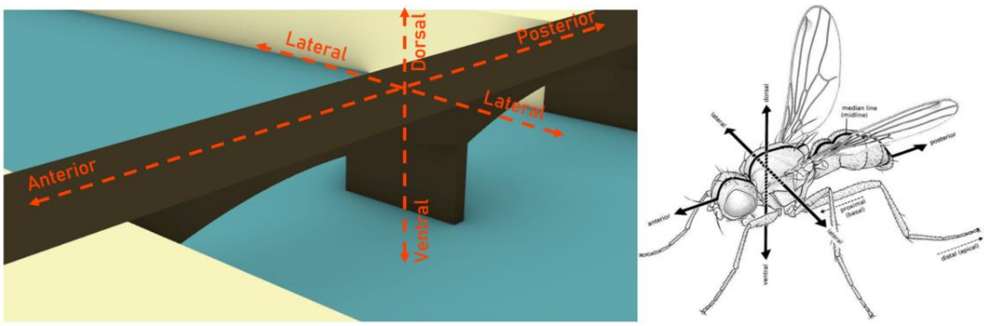


*Figure 15. **(Left)** Body axes of a bridge. **(Right)** Body axes of Drosophila [12]*

Application in model:

Each feature was tested out in the next iteration of the experiment and was applied as follows:

- Each agent type followed separate rulesets as compared to one rule set for all agents.
- Simulation environment was split into segments, each with an individual ruleset. These segments were represented by faces in a mesh.
- Segments were allowed to shrink and grow. This was done by having mesh faces grow by moving other vertices away from the growing mesh centre. The growth of a segment was controlled by decisions taken by the agents within the segment.
- A static map of chemicals constantly being deposited into the chemical matrix, create the boundary conditions for the simulation.

8 Experiment 2

The second experiment focused on increasing the expressiveness of the model and understanding the interplay between genes and simulation behaviour, as well as exploring the possibilities for the algorithm to solve engineering problems.

An important parameter of this experiment was the inclusion of a model environment, that could add or remove chemicals from the simulation at predefined locations to create an interface for the real world. The idea of the environment interaction was investigated further with the inclusion of spatial segments of rules within the simulation, an idea inspired by the development of body plan segments during embryonic development. Differential growth of these segments was investigated briefly but was discarded due to becoming cancerous.

Furthermore, each material was allowed to have separate behaviour to create more complex interactions.

The investigations showed the possibility of expressive solutions that added design knowledge, but also prompted a drastic simplification of the simulation model by removing the evolutionary aspect and switching the focus to the designer's impact on the simulation, with the model environment as a human/computer interface.
The corresponding Grasshopper3D plug-in on the thesis GitHub is "Experiment2.gha", which should be used with the Grasshopper3D file "ScenarioBridge.gh", a few sample environments is contained in the folder "Environments"

8.1 Algorithm

The environment was represented by a 200x200 pixel rgb bitmap with the colour red spawning chemicals, blue culling chemicals in the chemical matrices and green acting as obstacles.

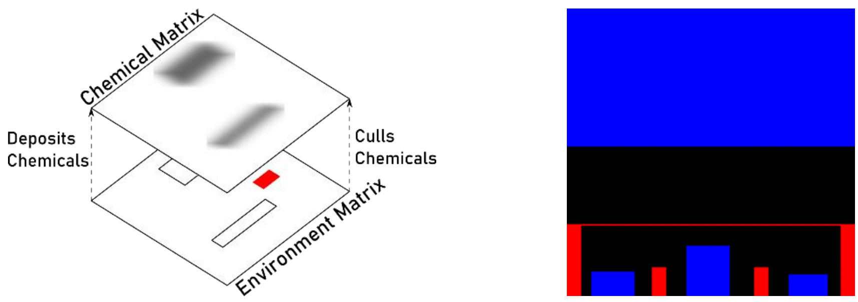


*Figure 16. **(Left)** Environment matrix/chemical matrix interaction. **(Right)** Environment matrix RGB bitmap*

To allow for cell differentiation, the genome was subdivided to contain different rulesets depending on material and body segment. Each ruleset is like the one presented in experiment 1 Figure 13. This genome was initially updated over generations using a genetic algorithm but was later in the experiment defined manually.

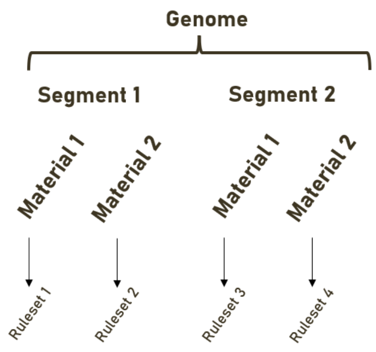


Figure 17. Genome layout.

The idea of a body plan was represented as a mesh. Each face in the mesh represented a segment of the design. Each segment contains two rulesets, one for each material. Each ruleset also included a gene that allows it to switch between materials if the conditions are met. The growth and shrinkage of each segment was briefly investigated, but ruled out, as it was impossible to control the growth of a segment, resulting in cancerous growth, this was also due to the agents not being bound to a single segment. This might have worked better for a collision-based algorithm.

8.2 Case study

For this experiment, a bridge spanning 100m in steel (orange) and concrete (grey) was chosen for the design scenario. The scenario contains areas that are good for supports and areas to avoid allowing traffic to pass underneath. The design had to connect the left and right side, with two optional intermediate supports.

This scenario tested the ability for the model to derive the connectivity of the design and develop a hierarchy of materials. It also tested whether the algorithm could generate effective structural geometry by the power of the underlying shortest path algorithm, or if it needed support by other engineering analysis tools.

A full structural evaluation of the bridge is not included as the algorithm was not ready at this stage.

A good solution to the problem has a clearly defined topology between supports, a geometry that is effective in form, size, and material choice.

Three different levels of benchmarks were designed with increasing complexity. A single span beam bridge of concrete, a continuous beam bridge of concrete utilizing the intermediate supports and a harp bridge utilising the tensile efficiency of steel.

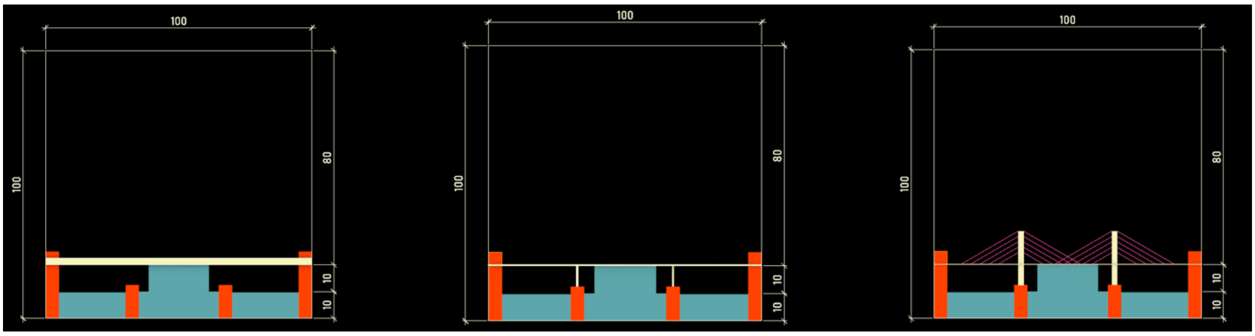


Figure 18. Experiment benchmarks with fitness estimates. ***(Left)*** *Simple beam. Fitness = 298* ***(Middle)*** *Continuous box girder with two intermediate supports. Fitness = 669* ***(Right)*** *Harp bridge. Fitness = 702.*

The fitness function for the evolutionary algorithm gave points for:

- Connection between left/right supports, 1 point per connected meter (0-100)
- Support of driveway, 1 point per meter supported driveway (0-100)
- Stiffness per weight, with a punishment for using steel over concrete. This is evaluated using an automatic FE-model. (Only applied if other two conditions are both equal to 100, 0-infinite)

The fitness function was designed to facilitate an evolution of the genome first to simply connect each side in a straight line before starting to develop efficient structural features, making it easier to find solutions.

8.2.1 Discretization of solutions

Different ways of adapting the simulation output to discreet geometry were investigated. A Delaunay mesh for each group of points where faces with large edge lengths removed was the natural choice. However, bin-lattice and “mesh per point” representations were also investigated. All three methods were more expressive than the simple convex hull used in the first experiment.

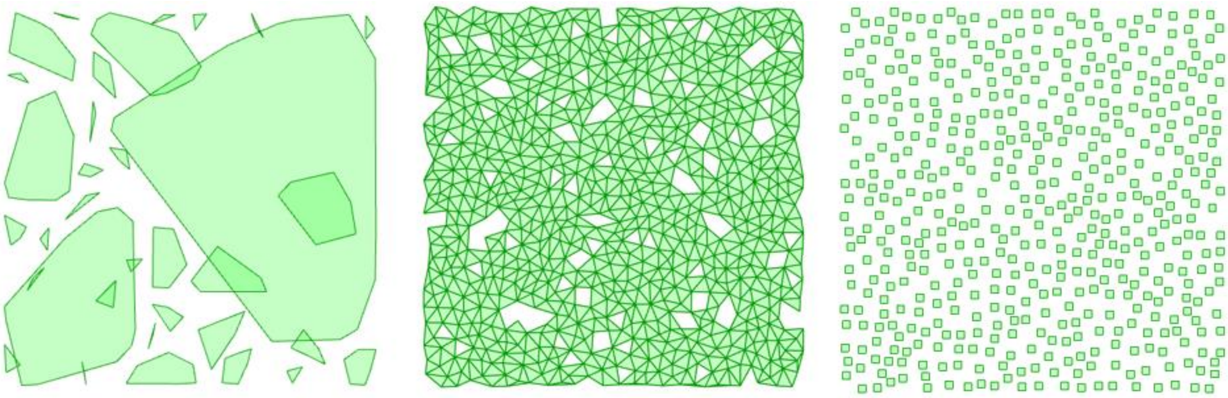


Figure 19. Point discretization methods ***(Left)*** *Convex hull* ***(Middle)*** *Delaunay mesh* ***(Right)*** *Bin-lattice*

8.2.2 Interfaces

As an addition to the actual geometry of the output, an additional level of output was added. The interface map. Each collision between materials is projected onto the map, which works as a heatmap for interfaces. This was useful as it not only showed where the clashes happen, but also the importance of the interface, which could guide the further design process.

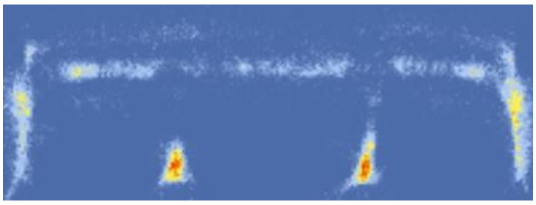


Figure 20. Interface map. Warmer colours signify more clashes throughout simulation.

8.3 Results

8.3.1 Initial results

It quickly became apparent, that an evolutionary approach would not prove fruitful, as a too large number of parameters in comparison with the time required to evaluate a genome. would not enable the algorithm to find a convergence within a realistic time. Even then, no solution was guaranteed. Additional investigations showed that most genomes would result in noise, unable to solve the design scenario.

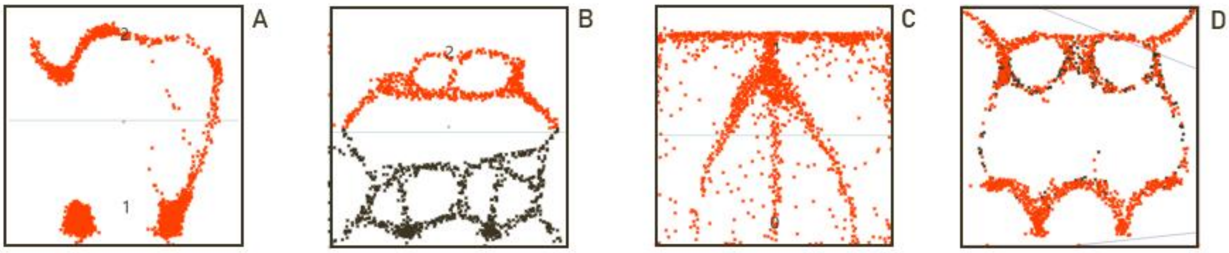


Figure 21. Initial results.

These results prompted an investigation of each model parameter in the genome, to understand exactly which genes had an influence on the phenotype. The evolutionary optimization part of the model was discarded as it was not suitable for the model at its current stage. This gave more responsibility to the designer, to define spatial hierarchies of materials.

8.3.2 Movement and expression

Exploring each parameter revealed how to achieve a design. It was possible to control the hierarchy of each material for negotiation, as well as connecting the supports in an efficient manner. The algorithm showed structurally efficient motifs, such as compression curves and connections without singularities to connect different structural elements. The structural motifs were a result of the inputs from the environment, rather than an inherent property of the algorithm itself.

The arches under the bridge deck in Figure 22 - E were caused by cells trapped between the diffusion of repellent chemicals on either side, while trying to find an efficient way through the chemicals to reach the other support. It was therefore evident that the expression of the cells was closely related to the input geometry provided by the designer.

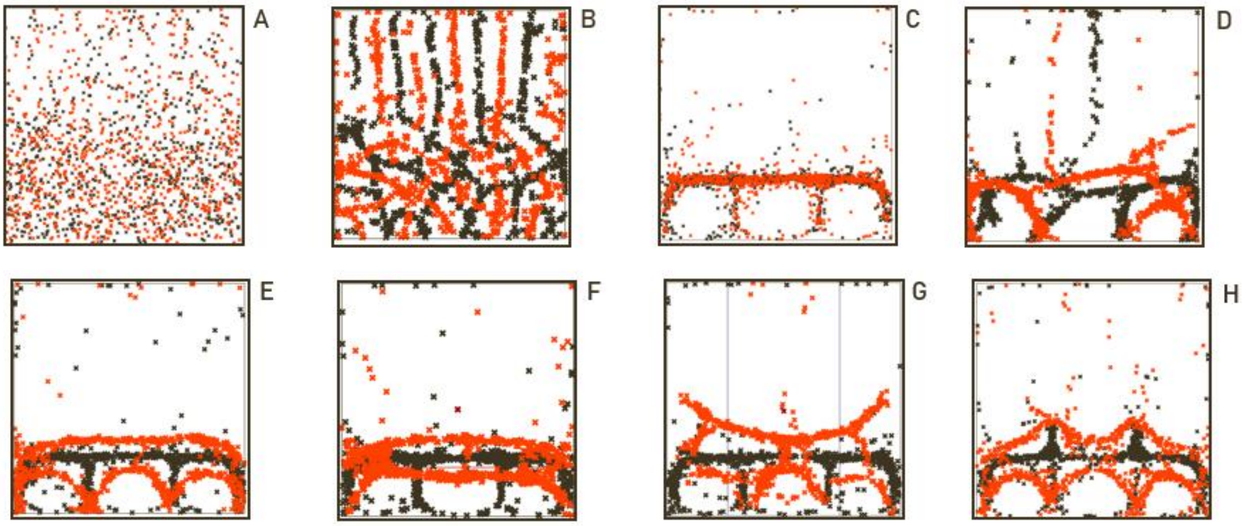


Figure 22. Results of manual genome programming. ***(A)*** *No chemicals and random movement.* ***(B)*** *steer towards own chemical while avoiding the chemical of other material.* ***(C)*** *Steer towards chemicals spawned by environment.* ***(D)*** *Steer towards chemical from environment while avoiding other material.* ***(E)*** *Both materials steer towards chemicals in from environment. Orange avoids grey creating a hierarchy in materials.* ***(F)*** *two segments with different rulesets. Orange does not avoid grey in the lower section and does not steer towards chemicals deposited by environment.* ***(G)*** *Three horizontal segments.* ***(H)*** *Slight alteration to environment map creates new design.*

The body segments allowed for greater control of the geometry, but the growth of body segments often resulted in cancerous behaviour, with segments growing uncontrollably, as it was very hard to control due to the random behaviour of the model.

The interface map worked as intended, returning support areas as interfaces between materials, as that is the major point of physical intersection.

It proved very difficult to control the spacing of agents, resulting in designs unable to solve the fitness function effectively, as the proportions of each design deviated from the benchmarks with magnitude of more than 10, resulting in extremely heavy solutions. Additionally, due to the randomness of the algorithm, the solutions included lots of noise, even after applying a level 2 Catmull-Clark subdivision to the Delaunay mesh.

8.4 Design output

Using a Delaunay algorithm for mesh discretization and Catmull-Clark subdivision for smoothing, the point clouds were discretised, and could be analysed. However, analysis was nonsensical as the noisy nature of the algorithm distorted the analysis output. Figure 22 - D shows agents randomly move away from the established structure to scavenge for better nutrition sources. The materiality of the simulation was also missing resulting in dimensions many times larger than what would have been expected.

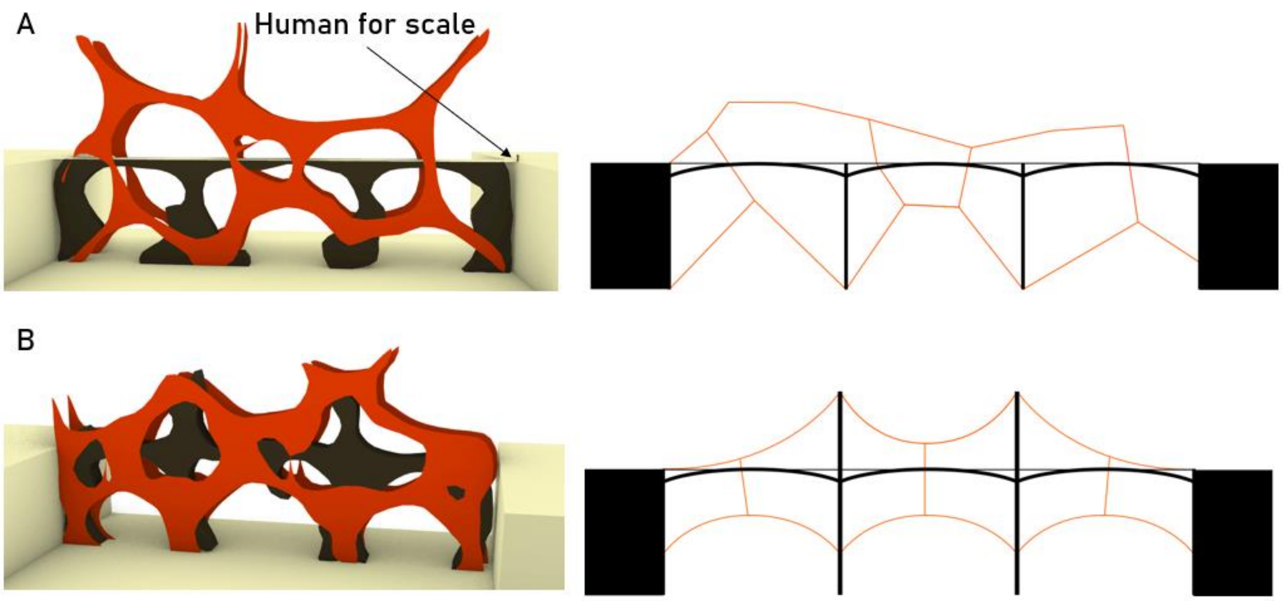


*Figure 23. **(Left)** Delaunay meshes of point clouds extruded. **(Right)** Manual idealized skeleton.*

An idealized version manually approximated from the median line of the results showed some structural potential.

Figure 23 - B exhibits the function of an arch bridge combined with a suspension bridge. However, Figure 23 - A showed less potential as an actual structure as the topology suggests that it would have to rely on bending and frame action, which is less efficient than pure compression/tension.

The model would also make less sense if the definitions were reversed, and grey represented steel and orange represented concrete utilizing the weakest attribute of each material.

8.5 Discussion

The experiment clarified what the algorithm can and can't do.

8.5.1 Knowledge

The algorithm lacked an inherent ability to solve structural problems by itself as seen in Figure 23 - A. It instead solved shortest path problems which can be seen as a proxy for structural behaviour as it favours efficient compression/tension curves, but structural efficient behaviour is not guaranteed.

A straight line is always the shortest path, which in many cases will be the most efficient solution, but that is not always the case[1]. This adds some amount of knowledge to the algorithm.

Another important point is that the algorithm lacks materiality. Sizing choices were arbitrary and mainly depending on the number of agents in the simulation, however important nodes such as supports still had larger dimensions, suggesting a proportional relationship of dimensions. The

[1] Sidenote: It would be interesting to see the old graphical structural design methods combined with modern shortest path algorithms for geometry generation, as they share some underlying features. This is currently done through classical computational optimization methods by people like Philippe Block.

algorithm showed great potential to react to changes within the environment, interpreting support areas and free zones into geometry.

8.5.2 Expressiveness

The expressiveness was drastically improved with the greater understanding of the parameters of the model. The idea of having a design body plan increased the expressiveness of the model allowing for varying hierarchies across the design but did not improve the algorithm in a fundamental enough way to be considered further. The expression is also limited by the underlying shortest path properties of the algorithm, favouring certain types of geometries.

8.5.3 Collaboration

The introduction of hierarchies also introduced some form of collaboration, having each agent make choices being dependent on the behaviour of the other. However, materials higher in the hierarchy will always have priority to a space, making the collaboration more of an authoritarian regime rather than a holistic negotiation.

8.5.4 Interaction

Initial clues on how to interact with the algorithm results were also shown during the experiment. A Delaunay mesh interpretation of the point cloud improved connection between simulation output and geometry. However, the meshes were still very noisy due to the stochastic nature of the algorithm and the continuous scavenging for nutrition. Idealised graph representations void of dimensions showed promise, improving the usefulness of the simulation output. Interface maps, while not elaborated upon, could prove very useful when integrating into a design process. Environment bitmaps gave the designer a fast way to iterate a design.

8.5.5 Design process

The results of the experiment suggest a place for the algorithm within the design process. It is not realistic that the model can provide a final design due to the lack of material understanding. It is however good at understanding the connectivity of a design problem given some boundary conditions. In this experiment the boundary conditions were structural, but they could just as much be representative of other disciplines if the underlying properties of the problem could be interpreted as a shortest path problem, such as duct paths or building circulation.

Having different chemicals within the environment matrix also allowed for different agents following different goals instead of having two agents competing for the same space. This plays to the strength of the algorithm by having the algorithm coordinate the space for the goals of each discipline. In a design process this could look like Figure 24.

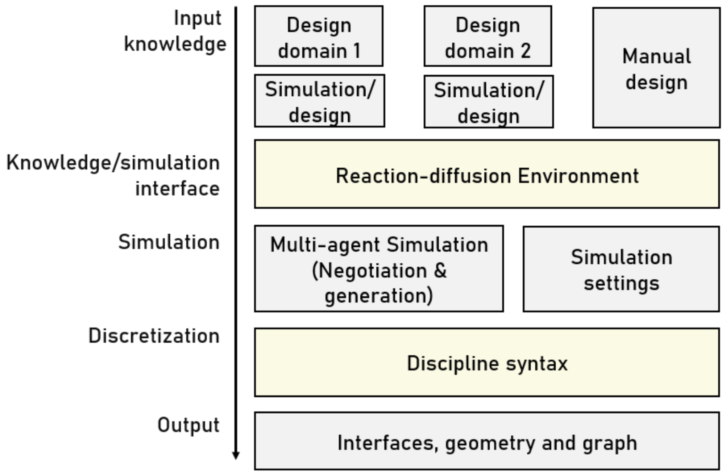


Figure 24. Proposed design process integration.

8.6 Summary

Parameter	Expressiveness	Knowledge	Collaboration	Interaction
Score				
Comment	Better understanding of expressiveness limitations and possibilities. Expression is limited by algorithm shortest-path properties.	Utilizing the underlying shortest path to add knowledge. Reacting to the environment. Proportional relationships of dimensions.	Hierarchy based negotiation	Delaunay meshes and potentially skeleton graphs. Interface maps. Environment map as input.

Table 4. Evaluation of solution objectives for experiment 2.

8.7 Abstraction

The solution objectives guide a selection of a concept within embryologic development to adapt. This concept should help move the model towards a design process application for the third experiment. Missing in the model in Figure 24 is the interface between design intent and simulation. Having a robust interface here could improve interaction, which in turn improves expression, knowledge, and collaboration.

Problem:
How can the designer communicate design intent to the model?

Biological analogy:
During embryonic development chemical gradients are used to control and coordinate the development, being one of the secrets behind the extremely robust development process. Information is expressed from a gene by creating proteins that control the behaviour of other cells. This effectively creates an interface between the design intent embedded in the genes and the expressed morphology.

Abstraction:
Gradients of information is an effective way to control a system. They can be used to express spatial intent. Gradients are effective as design tools because they are non-binary, allowing for exploration and negotiation in the areas where competing gradients meet.

Application in model:
Chemical diffusion is already a part of the model through the median filter applied to the chemical matrix. Gradients as a controlling input is however useful. Information can be embedded in a bitmap as a gradient of pixel values, with a brightness ranging from 0 to 255. Knowledge can be embedded in a way that is better suited for negotiation as targets are less binary. Most engineering analysis also comes in the form of gradient maps, such as stress diagrams and thermodynamics.

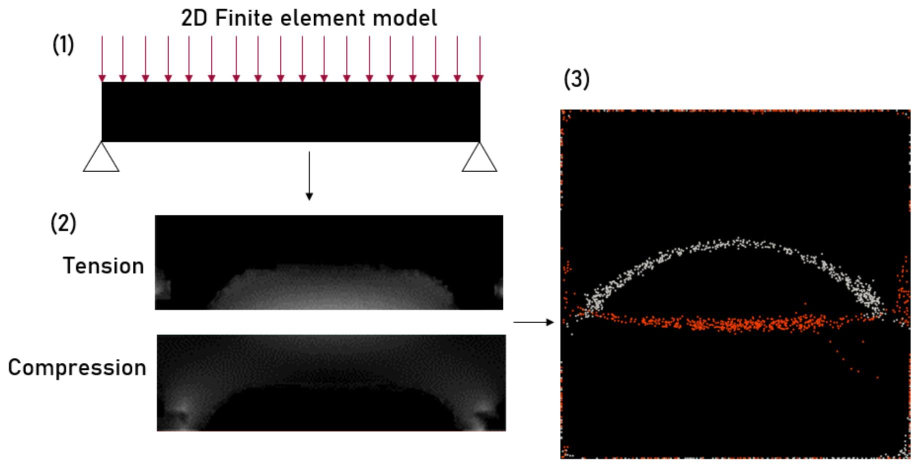


Figure 25. Gradient map proof-of-concept. ***(1)*** *Boundary conditions (Design case)* ***(2)*** *Gradient map of stresses (Knowledge)* ***(3)*** *Agents following gradient maps. (Simulation and negotiation)*

9 Experiment 3

It was established in experiment 2 that the algorithm is able to add useful information that can that be converted into an actual design. This experiment investigates how the information acquired from the model can be adapted into a design process, and how it can be discretized in a useful way, to integrate various disciplines. The experiment was a case study based on a simple architectural proposal for a building. The building had a predefined envelope and guiding internal floors. The model was tested on its ability to produce designs for internal structures, ventilation ducts and floor plan, through interpretation of the simulation results, guiding a design proposal with coordinated disciplines. This section goes in depth of each step from case to design.

The corresponding Grasshopper3D plug-in on the thesis GitHub is “Experiment3.gha”, which should be used with the Grasshopper3D file “ScenarioGradient.gh”. The sample environments are split into two parts. One for section-based simulation and one for plan-based simulation. Each containing environments for each discipline for multiple slices.

9.1 Algorithm

The algorithm was set up with three separate agent types each representing a design discipline, architecture, ventilation, and structures. Each agent type had its own chemical representing the goals of the discipline. No genetic algorithm was used. New concepts introduced for this experiment are explained in the following sections.

9.1.1 Application within the design process

The main goal of this experiment was to incorporate the algorithm into a design process. In this process a designer has given boundary conditions for the model and is given the task of interpreting the results into a design proposal. By adding information from each discipline even before an initial design has been given, the designer should be able to make better decisions about the prioritization of space and have better knowledge of what issues might occur further down the design pipeline, allowing the designer to address them immediately. See Figure 26.

The goal was to have the disciplines spatially organized in a way that allowed the placement of elements within each discipline, while providing interface maps of potential problem areas and connections. It is important to remember that the model cannot give materiality. Instead, the importance of each element was represented by the concentration of agents at a node. This allows for materiality to be added later, using the output of the simulation as an analytical framework. Materiality here also includes the designer’s aesthetic choices, which can either be added before simulation as boundary conditions or after simulation as an interpretive layer.

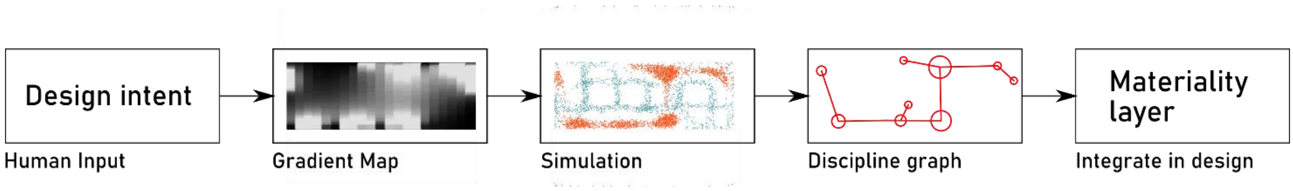


Figure 26. Layers of the design model.

9.1.2 Gradient maps

Concentration gradients are found everywhere in the complex chemical interactions during embryological development, due to the natural law of diffusion. These gradients are functionally local 1-D coordinate systems, denoting the distance from origin as chemical concentration, these coordinate systems are however not just a single line as the cartesian system is defined but rather radial or even non-uniform. These gradients can be discretized through the variance in sensibility of different genes to produce discrete body segments.

Gradients provided a robust way of interpreting space as the exact location of every element was not precisely given but had a strong tendency towards the highest value of a gradient. A version of this was already implemented in the model as the underlying chemical map, where a median filter was responsible for the diffusion of chemicals deposited by agents and the environmental chemical spawners. This experiment expanded on this by having the environmental matrix defined in terms of gradients. This provided an interface between the designer and the model where design intent can be expressed in terms of *gradients-of-usefulness* for each discipline. Ex. One would prefer to arrange columns in a uniform fashion, or one would prefer to have an office where there is a good view as compared to a bad view. The challenge is to understand the underlying spatial logic of each discipline. The location of elements on the gradient map is not promised, but rather a wish list for each discipline.

The gradient maps were defined as grayscale images. The brighter the pixels, the more chemicals are deposited for each timestep of the simulation. Gradient maps could also be combined by adding the pixel values of two images. Note that the agents make decisions based on which chemical has the higher value, the relative proportions of chemical deposits were therefore more important than the numerical values.

9.1.3 Discretization & Graphs

While the simulation can provide useful information for each discipline, there are still several steps between the output of the model showing a point cloud to a finished drawing. A skeleton as defined by a geometric graph proved to be a useful interface. A geometric graph is defined by edges and nodes; nodes have a spatial location, while edges describe connectivity of the nodes. In this implementation information is also embedded in each node describing their relative importance. The importance of a node is denoted by the size of the circle at a given node. This information is extracted by the concentration of agents at a given node. As the model is void of materiality, only the relative size of the nodes is important to the designer, favouring one space over the other. Each discipline has its own graph.

The graphs can either be generated algorithmically through mean axis algorithms or manually by simply drawing on the point clouds in CAD software. A manual approach was chosen as it was easier to implement, giving more robust results. The designer drew median lines on the point cloud as edges and chose the node size to be equivalent to the concentration of agents.

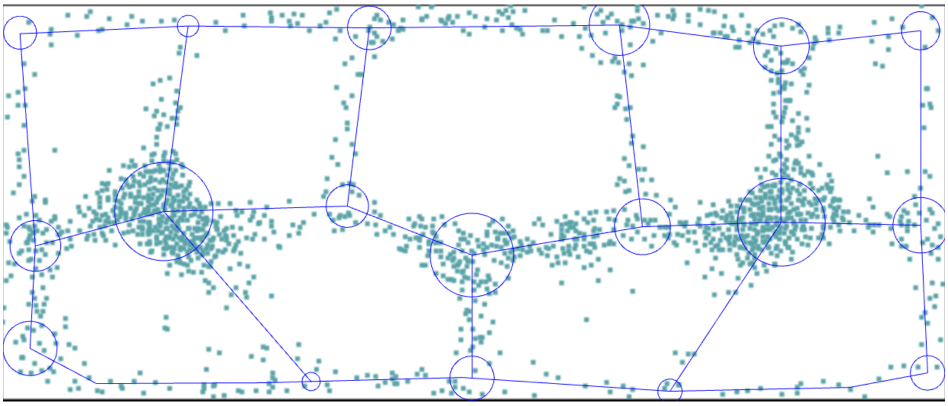


Figure 27. Point cloud discretized to graph with node information.

To assist the designer in interpreting discipline syntax[2] are introduced. They provide a guideline of what each node and edge represent within the model geometry. These should be determined before simulation. For structural elements nodes denote a vertical element, while edges denote a horizontal element.

9.1.4 Meta model - Combining simulations to design

As the model was defined in 2d, a single simulation can represent either a plan or a section. But a building design is typically defined by either several plans and sections or a unified 3d model, so how can a simulation representing a 2d drawing be extrapolated into a full building design? The intuitive solution was to adapt the simulation to 3d to generate a 3d model as output. This was however out of scope for the project and could be explored in further work.

Rather, a design was found by interpolating between a range of plans or sections, each simulated independently with elements perpendicular to the simulation plan being predefined but allowed to be negotiated.

The independent simulations had the disadvantage that each simulation attempted to solve all design challenges. To minimize this problem a careful decision had to be made between plan-based simulation and section-based simulation. Initial results on a section-based model showed that every section generated a vertical shaft, for connecting ventilation on different levels if simulated independently, resulting in three ventilation shafts rather than one. The nature of a plan seems to be more independent and was therefore chosen to be the basis for the experiment.

An alternative meta model is to have a hierarchy within the simulations. For example, have each plan inform the plan below, by generating a new gradient map for each plan based on the results of the plan above.

Other methods were also investigated. The medical industry has long been interpolating 2d images from MR scans into 3d models. This process requires hundreds if not thousands of 2d slices. To apply the same method in construction one would need 2d slices describing the space in between floor plans, which is a radical departure from how plans are typically imagined[3].

[2] To avoid confusion, note that while similar to the architectural space syntax analysis, these are not directly related concepts

[3] Sidenote: Medical organ segmentation software has an interesting overlap with BIM, as medicine often works on an organ-based approach, while AEC often has an element-based approach.

9.2 Case Study

The experiment was based on a case study inspired by building 127 on DTU Lyngby campus, a teaching building housing many of the architectural engineering courses and workshops at DTU. The building should house classrooms, workshops, exhibition space, study space, meeting rooms and supporting rooms. The case study provided a building envelope with walls, windows, and entrances, for a building with the same footprint, but with additional height. Guiding internal floors are also added, to allow for generation of gradient maps of light simulation, the floors were allowed to be negotiated.

The case study was chosen as it presented a complex 3d system with several interdependent interactions that would have to be negotiated. Ventilation had to figure out a way to connect in an efficient way, while avoiding structural elements. Structures had to find good column and beam locations without interrupting good architectural spaces.

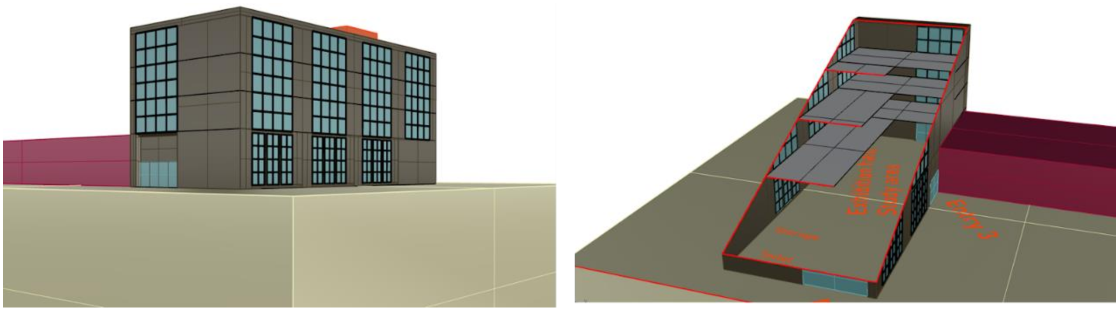


Figure 28. Case study boundary conditions for building 127 proposal.

9.2.1 Gradient maps

Environment matrices and simulation settings were defined for each of the disciplines, Architecture, Structures, and Ventilation. These inputs were a simplification of the spatial logic of the discipline and the interactions with other disciplines. Vertical elements of each discipline were predefined in order to generate the gradient maps, however. As they were contained in the gradient maps, they were able to be negotiated and overruled.

9.2.2 Architectural gradient map

The model defines architecture as how humans use space. The model postulated that the better space has more natural light. This is of course a gross simplification, neglecting many aesthetic-, cultural- psychological- and safety concerns of architecture. The usefulness of space was evaluated using a point-in-time ray trace simulation for a sky with uniform luminance in the Grasshopper3D plugin Ladybug. The higher lux gave more attractive space.

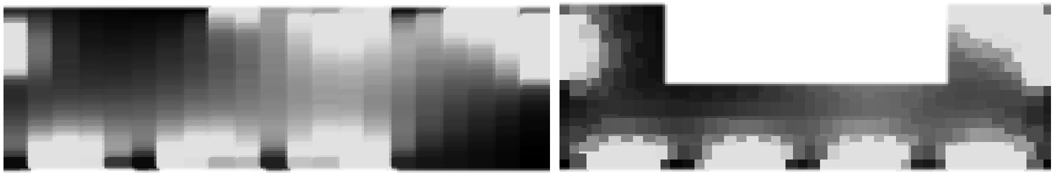


Figure 29. Architectural gradient maps. ***(Left)*** *Ground floor.* ***(Right)*** *Third floor.*

Note that the light map can also be supplemented or replaced by hand drawn attractiveness map giving the power to the designer.

The architectural map had priority of space; this was achieved by culling any other chemical that was present at the same spot as an architectural agent.

9.2.3 Structural gradient map

Structures cover vertical elements, columns/walls, and horizontal elements, beams.

The position of vertical elements were important boundary conditions to consider. They were needed to conduct the analysis needed for generation of a gradient map. However, as the output of the algorithm was the location of the agents, the position could be negotiated, and was not strictly predefined.

The gradient map was generated using a shell mesh FE model to generate a stress diagram of where material was needed. The model was evaluated for three load cases, vertical live load, longitudinal wind load and cross wind load. The resulting stress diagrams were converted to greyscale gradient maps, and then superimposed to create load combinations.

Figure 30. Structural gradient map. Pixel value of each load case is combined additively to form combined gradient map. Relative gradient values can be adjusted.

Structures were attracted by architectural chemicals, as that was most likely where structural support was needed. This was achieved by having the architectural agents deposit structural chemicals into the environment once they achieved a high concentration of agents in any area of the simulation. Structures avoided clashes with ventilation by removing any ventilation chemicals in locations occupied by structural agents.

9.2.4 Ventilation gradient map

Ventilation was simplified to horizontal supply ducts and vertical shafts.

The shaft location was the only thing defined within the gradient map, as ventilation relies on spaces defined by architecture. Ventilation agents therefore follow architectural agents while avoiding structural agents. Ventilation agents additionally avoided being too close to areas with many architectural agents to avoid disturbing the space.

9.3 Simulation & discretization

5 independent plans were simulated based on the gradient maps for the corresponding levels. Each combined point cloud was segmented into each discipline

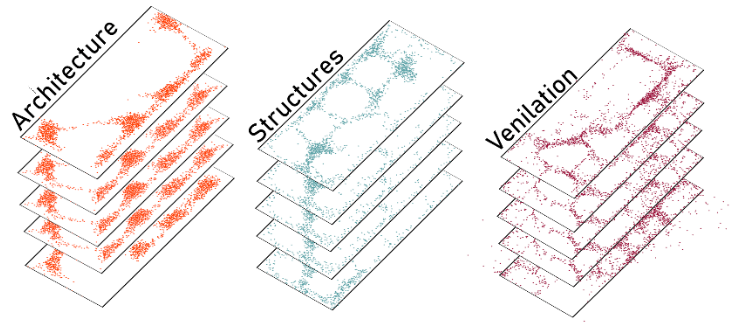


Figure 31. Simulation results for all levels disentangled to each discipline.

Point clouds were manually discretized into graphs containing node information. Node sizes were constructed by drawing the largest circle a node in the point cloud could contain.

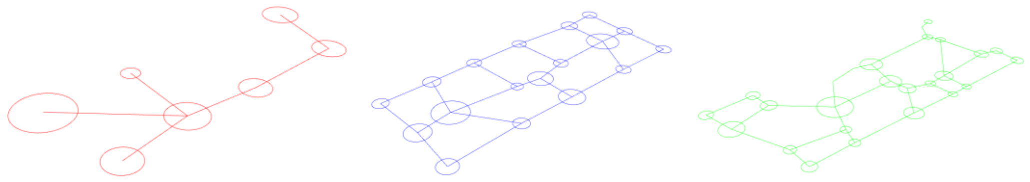


Figure 32. Graph representation of point clouds.

Nodes on each level were connected using the vertical boundary conditions as guides

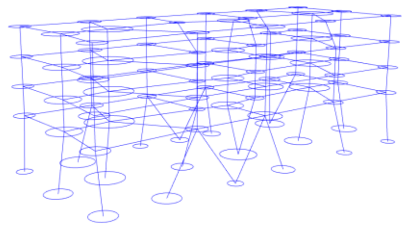


Figure 33. Graphs connected vertically.

Materiality and design could then be added based on edges and nodes embedded in the graph.

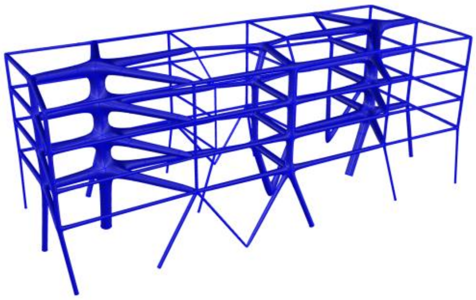


Figure 34. Example of a materiality application. Here SubD pipes were added using edges in graph as guides and node sizes as element thicknesses.

9.4 Design verification

This section is an example of a materiality layer, that can be imposed on top of the simulation results, adapting it into design. This was a manual process, tracing the point clouds in CAD software. Elements were given approximate thicknesses for rendering purposes. The section also explores the feasibility of the design in a more traditional fashion by analysing the geometries given by the materiality layer. This evaluation is only a rough estimate of function and is not meant to be exhaustive.

The information from the simulation informs a rough design proposal for a fictive new building 127 at DTU. Structural agents inform structural elements, ventilation agents inform ventilation ducts and architectural agents inform human preference for space.

9.4.1 Architectural

For the architectural maps the nodes in the graph were translated into rooms or meeting points. A larger node means a more important space. Edges in the graph denote which way people are likely to walk between these rooms. The floor layout was heavily influenced by the window layout having the most attractive spaces close to the windows as the gradient maps were controlled by light simulations.

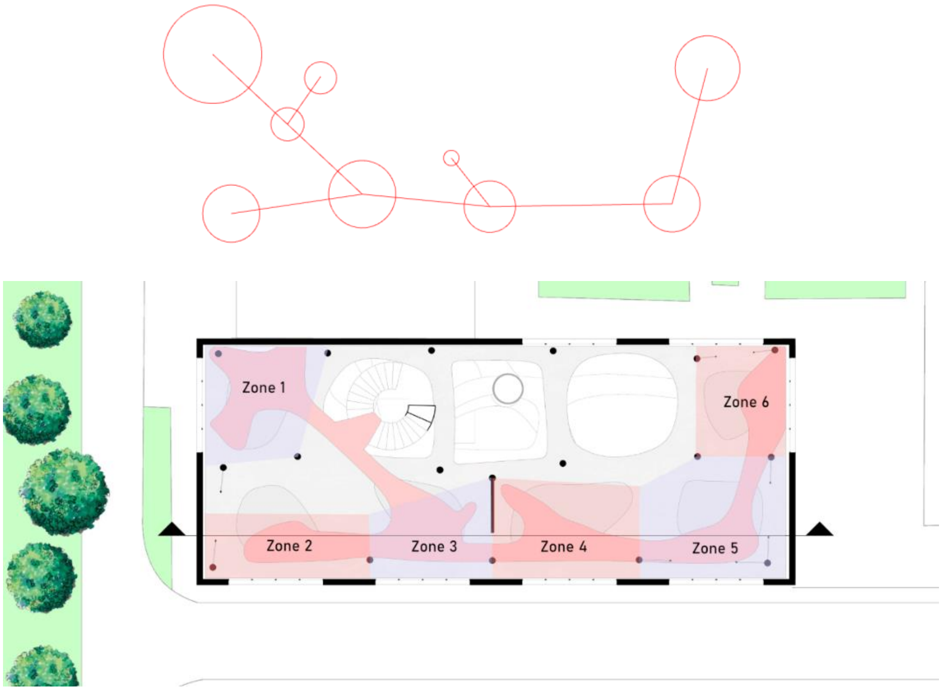


Figure 35. Architectural graph and materiality. Room layout is based on the point cloud.

9.4.2 Structural

Beams were materialized by interpreting final agent locations to pseudo-slabs with an arbitrary thickness of 200mm. The stability of the plan was then evaluated for wind loads and vertical loads, to understand how such a plan would function. Holes between slab were defined to be secondary slabs, for distribution of vertical loads, but with no horizontal function.
Columns were defined in the node locations of the graph.
The slab was evaluated in the FE software Karamba3d.

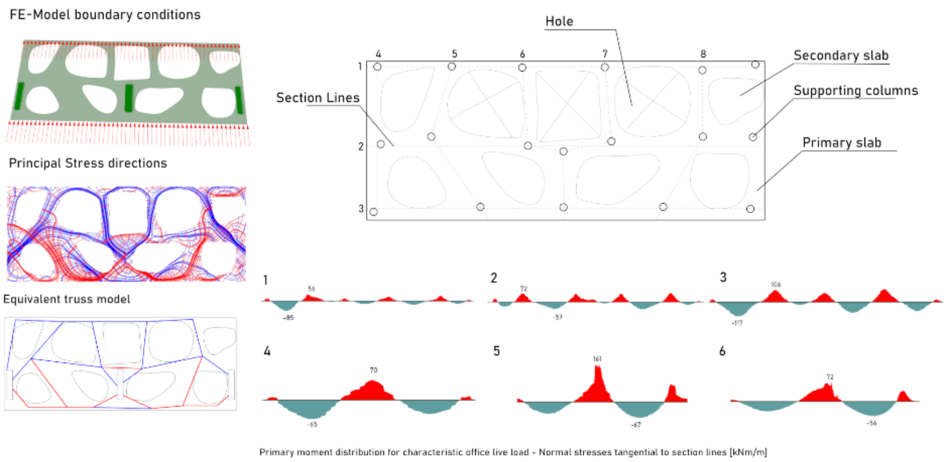


Figure 36. Structural investigations after materiality are applied. ***(Left)*** *Horizontal analysis.* ***(Right)*** *Vertical analysis with moment diagrams.*

The vertical loads were simply evaluated by investigating the moment distribution in the deck. Figure 36 shows the bending curves which would be equivalent to the function of in-situ concrete deck simply supported by either relatively slender columns or hinge-like supports. Shear and punching were neglected. The largest moments in either direction was observed around internal columns. This was also the area that had the largest concentration of agents, indicating a correlation between agent location and a need for structural material. The layout was efficient as no large deviations are created when transferring the loads.

The horizontal structures were evaluated in tension and compression through the floor plate to transfer pressure and suction to support lines. The outlines of the holes in the primary slab correlate with the arches of the stresses, minimizing the need for force transfer through bending. Here an equivalent truss model was applied to understand the function of the design. The truss is a simplification of the principal stress flow but is both kinematically and statically allowable. The morphology of the truss is triangulated and connected to supports, allowing efficient transfer of loads from facade to supports.

9.4.3 Ventilation

By adding a shaft location, the layout of ducts for ventilation supply could be evaluated into duct dimensions for a given air change rate. Return ducts were not considered. Graphs were interpreted into non-circular graphs for duct channels. The node size was interpreted as need for ventilation at a specific node. Therefore, the total flow of a duct will be the sum of all upstream nodes. Node sizes were an arbitrary unit but are converted to a fraction of the total needed ventilation.
The total needed ventilation was found from building code standards and floor area. It was in this case be estimated to an air exchange frequency of 2 1/h resulting in a total ventilation need of 3361 l/s for a given floor. Duct dimensions were then found using a nomogram with a pressure loss limited to 1 Pa/m.

- Main ducts - Ø600
- Primary distribution ducts - Ø450
- Secondary Distribution ducts - Ø180
- Shaft - Intake from 4 floors - Ø900

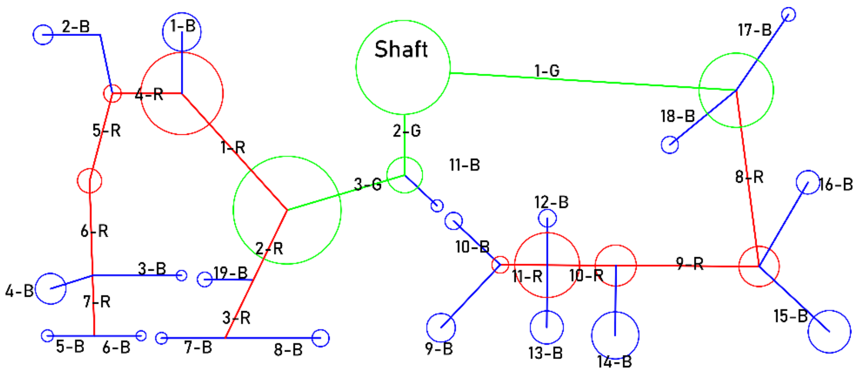


Figure 37. Hierarchical non-circular graph representation. Each color represents a layer of the hierarchy towards the shaft. Main ducts: Green. Primary distribution ducts: Red. Secondary distribution ducts: Blue.

9.4.4 Interfaces and clashes

The simulation provided maps of clashes between disciplines. It is worth noting that the results are simulated within the same height without a specific part of the ceiling dedicated to ventilation, as one would typically have in a 3D scenario.
In the plan there were some inherent clashes between structures and ventilation as both structural elements and ventilation elements must support the whole floor.
This happens even if ventilation agents were instructed to avoid large concentrations of structural agents. In a good design these clashes do not occur at important nodes, where important vertical elements occur. Highlighted areas indicate that many ventilation and structural agents occupied the same space throughout the simulation. This can be understood as a proxy for risk of interface. These interfaces were ranked based on the total number of agent clashes in the interface. The figure below is the result of a simulation for 3rd floor and the following potential clashes.

1. Ducts close to the supporting wall, central column, and beam over open space.
2. Duct passing through column
3. Duct passing through important beam for horizontal stability
4. Duct passing through beam

The largest concentration of structural agents was avoided by the ventilation agents, with clashes happening around the important columns rather than through.

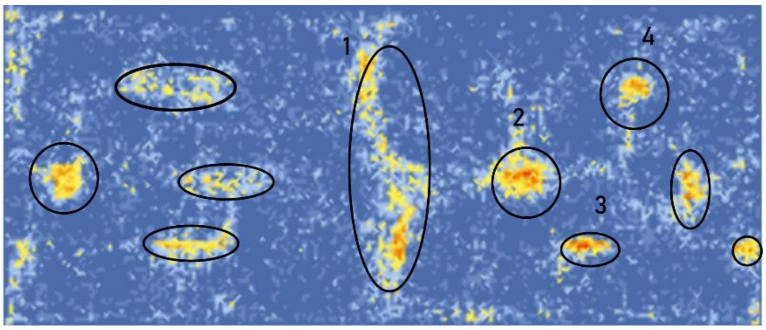


Figure 38. Ventilation/Structures interface map

9.5 Discussion

The algorithm was shown in an integrated design process as an example for how it could be used by a designer as support. It mainly showed the potential for gradient maps and graphs to work as design interfaces to be integrated in an architectural design process.

The experiment contrasted experiment 2 where the boundaries for the simulation were less rigid. The less defined a system is the more gaps the algorithm must fill. To get the best results the designer needs to clearly understand the nature of the multi-agent algorithm, its underlying shortest path characteristics, and the intent of the gradient maps/graphs.

Some interpretations were more straightforward: For a gradient map based on a stress diagram, a large agent concentration will signify a need for material, while edges in the graph show efficient load paths between nodes. Other interpretations were fuzzier: an interpretation based on a lighting diagram could be interpreted as the attractiveness of space, but what to do with that space was still unknown.

9.5.1 Expressiveness

The experiment showed that the expressiveness of the model is closely tied to the designer's choices. Initial boundary conditions had a large impact on the eventual output of the model, restricting the model to choices made by the designer. A 2d model requires many choices from the designer further expanding the responsibilities of the designer.

9.5.2 Knowledge

Gradient maps provide a useful knowledge interface between designer and simulation. The simulation mainly interprets information in the gradient map as opposed to experiment 2 where the simulation had larger agency in adding new information. In experiment 3, the information acquired was about collaboration rather than technical analysis. For the ventilation system where the gradient map was less dominant, more new information was acquired. This suggests that the responsibility of adding knowledge should be by the designer, similar to the design intent described in Figure 26.

9.5.3 Collaboration

Every discipline was able to achieve their design goals, Rooms were located at windows. An efficient structural layout was proposed, and ventilation connected rooms and shafts in a short distance without interfering with the other disciplines. However, the relationships were still hierarchical rather than holistic. The gradient maps however helped disentangle the collaboration process into defined parts for negotiation.

9.5.4 Interaction

Graphs provide a useful interface for the designer, allowing the designer to apply styles and materiality. The process is still manual and prone to misinterpretation by a designer. The fuzzy nature of the algorithm allows the designer to impose his/hers own bias'.

9.6 Summary

Parameter	Expressiveness	Knowledge	Collaboration	Interaction
Score				
Comment	Expression controlled by designer	Gradient maps are useful, but moves the added knowledge from the algorithm to the designer	Hierarchy based negotiation. Interface maps helps coordination.	Graphs allow intuitive understanding of simulation results. Prone to misinterpretation.

Table 5. Evaluation of solution objectives for experiment 3.

10 Discussion

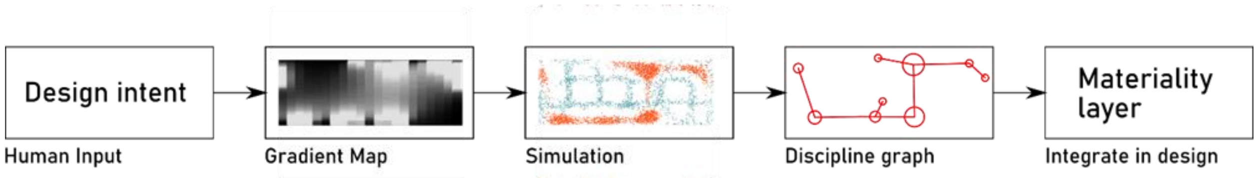
10.1 Experiments and the design process

Throughout the three experiments, three different approaches to algorithmic collaboration were tested. First, an approach purely dependent on a genetic algorithm to generate collaboration rules was proven to be unsuccessful, due to the limited computational power in comparison to a vast search space, with small boundaries for where a gradient applies to the fitness function.
Experiment 2 and 3 proved more successful, being able to exhibit behaviour satisfying the solution objectives.

Both provided usable solutions, but with a different focus:
Experiment 2 relied on the algorithm to solve most of the geometry, given loose boundary conditions, while experiment 3 tied the agents tighter to the gradient map.
This means that experiment 2, was more akin to a geometry generator. Trustworthy solutions with this approach relies on having the design domains share the underlying shortest path properties of the algorithm, which is not well enough understood for many types of engineering problems, many engineering problems rely on materiality simply not present within the algorithm, often a specialized generative algorithm could give better solutions.

Experiment 3 on the other hand relies on the algorithm to connect bits and pieces of knowledge into a unified design, yet it does not add as much new technical information. This creates a branching path in the research, is the model more useful as a technical optimizer or a collaborative tool? I would argue that its place is as a collaborative tool, as it stays the truest to the intentions and inspiration. This is not to say the work as an optimizer does not have potential, but more knowledge would be needed about the fundamental shortest path properties of each field, and it would have to compete with existing specialized algorithms.
In contrast there seems to be a vacuum and a missing space for coordination algorithms. Here the designer should focus on doing design and the algorithm should coordinate that design, allowing the designer to spend less time collaborating and more time designing.

The current framework for the model consists of the steps as presented in section 9.1.1, with the multi-agent algorithm working as the negotiation engine.



In this approach, the algorithm is simply negotiating the knowledge that is put in, abstracting it to a graph, where materials can be applied. The material layer is not dealt with in this thesis, but should consist of sizing, materials, style, spacing, and should use specifications and the nodes and edges within the discipline graph as inputs.

10.2 Revisiting the vision

Each experiment included a discussion of how well the solution objectives were solved. This prompts a concluding discussion for how well the vision was addressed, as the solution objectives works as a proxy for the vision as presented in section 2.2.

In the vision, design decisions were added in a parallel fashion and collaboration is not a factor. This is where this algorithm would apply, as it disentangles the decision process for a designer, and allows for a modular approach for idea-input. However, to be a fully integrated part of the design process, iteration would also need to be included, prompting a designer to explore the design space rather than settling on a specific solution.

The vision also focused on ways to interact with design. The thesis proposes an intuitive human/machine interface to simplify the design process into non-technical, yet meaningful decisions.

A robust way of solving design specifications is still missing. It would be the responsibility of the materiality layer, which is still a huge undertaking to develop. On top of this comes the need for additional levels of detail and integration further in the construction process to manufacturing of components and construction of the building.

Coordination algorithms could change the design process fundamentally bringing the designer closer to the final design, simplifying the human interface with design, and allowing iterations for even complex projects. This reality is still far away, but what it takes to get there is now clearer after this investigation.

10.3 Adaption of concepts from biology

Throughout the thesis, concepts used in the model was appropriated from biology with varying success. The thesis set out trying to adapt the combined design process of nature, with evolution designing the typology and development specifying the design, which needless to say was a too ambitious undertaking.
The thesis ended up exploring the developmental side of the analogy. Agents translate ideas of how a design should perform into a design guide, this approach is different than having the algorithm cover both the development of the idea and the specifics of the project, unifying evolution and development. Biological development is an extremely complex process that varies from species to species, and while this project tries to harness the power of chemical diffusion, that is fundamental to the developmental process, it is almost impossible to find an algorithmic analogy for the numerous and very complex interactions of proteins.

This is partially due to the very computationally expensive algorithm of diffusion, limiting the amount of chemicals in the simulation, and partially due to the very complex model required to describe the complex multi-stage process of genes controlling other genes at certain protein signals and concentrations.

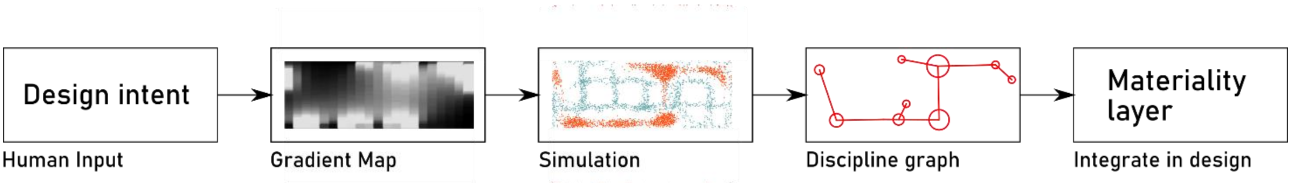
However, focusing on concepts from embryonic development was able to propel the thesis forward. Some concepts were tested and discarded, such as body plan segments and differential growth, while others proved to successfully integrate with the initial multi-agent model such as chemical environments of gradients and the connection between phenotype and environment.

The appropriation of each concept started with a need as defined by the research solution objectives at the end of an experiment. The appropriate literature was then scanned for developmental solutions to this problem. The concepts were then abstracted and integrated in the project as computer code. Biomimetics beyond physical features therefore proved to be a useful idea generating engine for the thesis.

11 Conclusion

This thesis investigated an approach of a multi-agent algorithm based on the slime mould Physarum Polycephalum combined with a genetic algorithm to solve design problems. The algorithm works by having agents interact with diffusing chemicals, by extending the model to include several possible behaviours, the model can solve simple collaboration challenges.

The thesis developed an approach, where the design model is split into layers, of design intent, input interface, simulation/negotiation, output interface, and materiality, with the thesis focusing on the interfaces and simulation. This approach helped disentangle the collaboration process, freeing designers up to focus on design and knowledge.



This approach was found as the result of three iterative experiments:
First a simple proof of concept experiment featuring the design of a concrete cross section to understand how to adapt the algorithm to a problem-solving process. It was found that the algorithm needed to be updated to convincingly solve engineering problems.

Secondly an experiment of a simple bridge subject to uniform loading in steel and concrete. This experiment featured interactions with the environment in the form of a map of boundary conditions for the agents to interact with. It was found that the multi-agent algorithm was ill suited to be combined with the genetic algorithm, due to the large computational demands combined with a very large solution space without sufficient gradients in the fitness function. It was also found that the algorithm was able to come up with convincing solutions to an engineering problem, but that the outputs were very noisy.

Thirdly an experiment of a simple architectural design process featuring three disciplines: Architecture, structures, and ventilation. The experiment investigated how the algorithm might be used in a design process. It featured gradient maps as an input interface and graph representation as an output interface. The experiment showed that the interfaces could be used as part of a design process that could coordinate separate goals leading to the proposed framework. However, solutions were only found in a hierarchical fashion rather than a democratic fashion.

12 Further work

The model used in this thesis can be extended by implementing the model as a whole or revising each of the layers in the proposed model.

- **Design Intent –** How does a team of designers abstract their design intent? Having a team of designers collaborate over a collaboration algorithm could show how design intent is described. This will also reveal the needs a team of designers would have for a model and how it would affect teamwork.

- **Input Interface -** An important part of the design process is iteration, and a useful algorithm would need to support that. This could be done by manipulating the graph output as an iteration interface if the designer wants slight changes and instantly sees the consequences for collaboration and clashes.

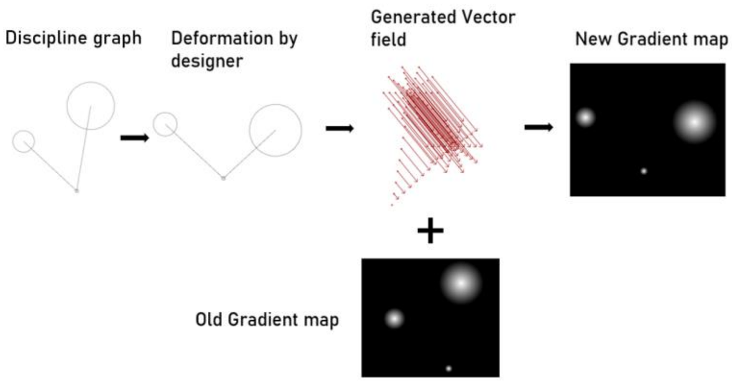


Figure 39. Proposal for iteration method.

- **Simulation/Negotiation -** The underlying negotiation mechanism would have to be revised. Here game-theory could propose mechanisms for reaching consensus between disciplines in a design problem.
  To understand the problem-solving capability of the algorithm, a better understanding of the underlying shortest path properties of various fields is needed to understand where the algorithm can be applied convincingly. Is an element-based approach better? What would be the best implementation for HVAC? How can materiality be considered for structures?

- **Output interface -** Graph representation of design and using graphs as an active design tool would be a useful way to interact with an algorithm. BIM-graph integration would be a logical step. The Grasshopper3D plugin “Topologic” could be a starting point for this. Investigating hierarchical graphs could also prove to be a useful entry point; in most design problems, hierarchies are an important property, such as a hierarchical relationship between columns and beams or ventilation ducts and shafts.

- **Materiality -** What does it mean to design without materials and how can materials later be applied? A materiality system based on graphs would need to be developed. The material system would need to be able to be swapped out, providing several possible solutions to a single graph, without affecting the underlying negotiated graph.

13 References

Adamatzky, A. (2019). *Slime Mould in Arts and Architecture*. River Publishers.

Almeida, J. E., Rosseti, R. J. F., & Coelho, A. L. (2013). *Crowd Simulation Modeling Applied to Emergency and Evacuation Simulations using Multi-Agent Systems*. http://arxiv.org/abs/1303.4692

Bae, J., & Yoo, H. (2018). *Fast Median Filtering by Use of Fast Localization of Median Value*. *13*(12), 10882–10885.

Casares, F. (2000). The Art of Genes: How Organisms Make Themselves . Enrico Coen . *The Quarterly Review of Biology*, *75*(3), 320–321. https://doi.org/10.1086/393536

Chalmers, C. (2008). Chemical Signaling as a Model for Digital Process in Architecture. *ACADIA 08: Silicon + Skin: Biological Processes and Computation: Proceedings of the 28th Annual Conference of the Association for Computer Aided Design in Architecture*.

Collopy, A. X., Adar, E., & Papalambros, P. Y. (2020). On the use of coordination strategies in complex engineered system design projects. *Design Science*, *6*, e32. https://doi.org/10.1017/dsj.2020.29

Doniec, A., Mandiau, R., Piechowiak, S., & Espié, S. (2008). A behavioral multi-agent model for road traffic simulation. *Engineering Applications of Artificial Intelligence*, *21*(8), 1443–1454. https://doi.org/10.1016/j.engappai.2008.04.002

Gerber, D., & López, R. S. (2014). COntext-aware multi-agent systems negotiating intensive fields. *ACADIA 2014 - Design Agency: Proceedings of the 34th Annual Conference of the Association for Computer Aided Design in Architecture*, *2014-Octob*, 153–162. https://doi.org/10.5151/despro-sigradi2014-0024

Hazelrigg, G. A. (1997). On irrationality in engineering design. *Journal of Mechanical Design, Transactions of the ASME*, *119*(2), 194–196. https://doi.org/10.1115/1.2826235

Jones, J. (2015). *From pattern formation to material computation : multi-agent modelling of physarum polycephalum*. http://dx.doi.org/10.1007/978-3-319-16823-4

Klemmt, C., Gheorghe, A., Pantic, I., Klemmt, A., & Sebestyen, C. (2019). Discrete vs. Discretized Growth. *Acadia*, *39*, 2–13.

Lee, J., & Jeong, Y. (2012). User-centric knowledge representations based on ontology for AEC design collaboration. *CAD Computer Aided Design*, *44*(8), 735–748. https://doi.org/10.1016/j.cad.2012.03.011

McGinley, T. (2015). A morphogenetic architecture for intelligent buildings. *Intelligent Buildings International*, *7*(1), 4–15. https://doi.org/10.1080/17508975.2014.970120

McGinley, Tim. (2022). Morphogenetic Prototyping for Coders and Carvers. *Unreleased*.

McGinley, Tim, Hoshi, K., & Iacopetta, L. (2015). MorphoCarve : Carving morphogenetic prototypes. *Iasdr 2015*, *2015*. https://doi.org/10.13140/RG.2.1.3995.0803

Meyboom, A. L., & Reeves, D. (2013). Stigmergic space. *ACADIA 2013: Adaptive Architecture - Proceedings of the 33rd Annual Conference of the Association for Computer Aided Design in Architecture*, 199–206.

Pantazis, E., & Gerber, D. (2018). A framework for generating and evaluating façade designs using a multi-agent system approach. *International Journal of Architectural Computing*, *16*(4), 248–270. https://doi.org/10.1177/1478077118805874

Pawlyn, M. (2016). *Biomimicry in Architecture* (Vol. 1, Issue).

Schumann, A., & Pancerz, K. (2016). *Physarum Chip: Growing Computers from Slime Mould. Logical Aspects*.

Speck, T., & Speck, O. (2008). *Process Sequences In Biomimetic Research*.

Tero, A., Takagi, S., Saigusa, T., Ito, K., Bebber, D. P., Fricker, M. D., Yumiki, K., Kobayashi, R., & Nakagaki, T. (2010). Rules for biologically inspired adaptive network design. *Science*, *327*(5964), 439–442. https://doi.org/10.1126/science.1177894

Veloso, P., Rhee, J., & Krishnamurti, R. (2019). *Multi-agent Space Planning A literature review (2008-2017) Multi-agent Space Planning*. *January*.

vom Brocke, J., Hevner, A., & Maedche, A. (2020). *Introduction to Design Science Research*. *September*, 1–13. https://doi.org/10.1007/978-3-030-46781-4_1

14 Image References

Any illustration in the thesis not cited is created by the author.

[1] McGinley, T., Hoshi, K. and Lacopetta L. (2015). 'MorphoCarve: Carving morphogenetic prototypes', *Iasdr 2015*, (2015). doi: 10.13140/RG.2.1.3995.0803.

[2] McGinley, T., Fotia, A. and Abroe, B. (2015). 'From Hopeful Monsters to Morphogenetic Prototypes', *Architectural Theory Review*, 20(3), pp. 365–375. doi: 10.1080/13264826.2016.1177563.

[3] Bentley, W. (1912). Studies among the snow Crystal…, *Monthly Weather Review*

[4] The National Archives (2014). Bank_Run_on_American_union_Bank [Online image] https://unwritten-record.blogs.archives.gov/2014/10/29/black-tuesday-85-years-gone-by/ Accessed 10-01-22

[5] Dussatour, A. (CNRS) (2016). https://www.quantamagazine.org/slime-molds-remember-but-do-they-learn-20180709/ . Accessed 10-01-22

[6] PTV Group (2021). PTV Viswalk: all Rimea 3.0 Test Cases, [Video]. YouTube. https://www.youtube.com/watch?v=tq1hoQt0fkU

[7] PTV Group (2021). [Untitled Illustration] https://www.ptvgroup.com/en/solutions/products/ptv-visum/ Accessed 21-01-22

[8] Tero, A., Takagi, S., Saigusa, T., Ito, K., Bebber, D. P., Fricker, M. D., Yumiki, K., Kobayashi, R., & Nakagaki, T. (2010). Rules for biologically inspired adaptive network design. *Science*, *327*(5964), 439–442. https://doi.org/10.1126/science.1177894

[9] Jones, J. (2015) *From pattern formation to material computation : multi-agent modelling of physarum polycephalum*. Available at: http://dx.doi.org/10.1007/978-3-319-16823-4.

[10] Jones, J. (2015) *From pattern formation to material computation : multi-agent modelling of physarum polycephalum*. Available at: http://dx.doi.org/10.1007/978-3-319-16823-4.

[11] Gerber, D. and López, R. S. (2014) 'COntext-aware multi-agent systems negotiating intensive fields', *ACADIA 2014 - Design Agency: Proceedings of the 34th Annual Conference of the Association for Computer Aided Design in Architecture*, 2014-Octob, pp. 153–162. doi: 10.5151/despro-sigradi2014-0024.

[12] Bozsik, A. (2020). Agricultural Entomology (a teaching draft for crop protection students). 10.13140/RG.2.2.11362.27842.