ESCUELA DE CONSTRUCCIÓN CIVIL
FACULTAD DE INGENIERÍA

Revista de la Construcción
Journal of Construction

Research Article

A generative design-to-BIM workflow for minimum weight plane truss design

Berk Malgıt[1], Ümit Işıkdağ [2], Gebrail Bekdaş [3,*], Melda Yücel [4]

1 Arup Netherlands, Amsterdam (Netherlands), berkmalgit@hotmail.com
2 Department of Informatics, MSGSÜ, İstanbul (Turkey), umit.isikdag@msgsu.edu.tr
3 Department of Civil Engineering, Istanbul University-Cerrahpaşa (Turkey), bekdas@iuc.edu.tr
4 Department of Civil Engineering, Istanbul University-Cerrahpaşa (Turkey), melda.yucel@yahoo.com.tr
*Correspondence: bekdas@iuc.edu.tr

Received: 05.01.2022; **Accepted:** 19.08.2022; **Published:** 30.08.2022

Citation: Malgıt, B., Isikdag, U., Bekdaş, G. and Yücel, M. (2022). A generative design-to-BIM Workflow for Minimum Weight Plane Truss Design. Revista de la Construcción. Journal of Construction, 21(2), 473-492. https://doi.org/10.7764/RDLC.21.2.473.

Abstract: Structural design has a significant impact on the overall cost of truss structures. In order to reduce the cost of a structure it is important to support designers' decision making starting from the early design phase. In this study, an optimization workflow is proposed, developed and implemented using well known generative design and Building Information Modeling (BIM) tools to achieve a cost-optimal design of a truss structure in the early design phase. Generative design aims to develop products that are lighter, stronger, more efficient and tailored to the specific needs of end users. Generative design tools allow users to create efficient designs via optimizing factors such as cost, weight, energy efficiency and performance. The aim of this study was to develop an optimization workflow to find and model the minimum weight / minimum cost design alternative in the early design phase using generative design, structural analysis and BIM tools in an integrated manner. The goal for the optimization was determined as finding the minimum weight (thus minimum cost) structure among the generated design alternatives. The single span steel truss was selected as the structure to be optimized, and optimization scenarios were prepared and implemented to determine the structural components of the truss with minimum weight. The results demonstrated that through integrated use of structural analysis, generative design and BIM tools minimum-weight truss design can be realized easily and practically.

Keywords: Parametric design, generative design, optimization, BIM.

1. Introduction

Modern engineers and architects strive to improve the performance of the design process through computational methods, using parametric design as a generative technique (D. J. Gerber, 2009). Parametric design methods have greatly increased the geometric and form diversity in the design process. Nowadays, automatic methods are needed to evaluate many design solutions as they can be generated quickly(D. Gerber & Lin, 2012). Analyzing the design and its shape in real time or synchronously and providing feedback that enables deliberate pre-decision making for the design process is necessary to facilitate design processes(Bianca, Flora, Robin, John, & Jane, 2011). Parametric design not only allows architects to improve their own designs, but also allows civil engineers to practically automate calculations and analysis of shapes. The concept of parametric design originated in mathematics. Parametric design consists of interconnected algorithms for generating geometries

(Bucci & Mulazzani, 2002). Beams are one of the most important load bearing elements in engineering structures. For this reason, it is extremely important to use optimization techniques for the design of beams. In the design of large and modern structures, minimizing the structural weight is an important necessity along with achieving a certain structural performance. Nowadays, the optimal design of beam structures is mainly based on the optimization of size and shape with topology (Banichuk & Karihaloo, 1976). However, the optimization of topology of beam sections is rarely found in the literature. Yet, the study of topology optimization of beam profiles is so useful and important. Topology optimization is an effective technique for finding a configuration of cross sections that provides scientific guidance for schematic design of structures and ensures that designers make rational decisions for elements or complex structures in the conventional design phase (Rosen & Peters, 1996).

Optimization in the early stages of the design process can provide an advantage in terms of material savings. Creating lightweight structures through weight reduction can be an effective way to combat the high cost of today's high-performance materials. Lower material costs result from the reduced weight of the profiles, and production/operating costs also decrease. On the other hand, engineers and architects think about practical solutions. An automatic structural analysis tool that can work at the early phases of the design will allow designers to find faster and better solutions that increase structural performance and efficiency and reduce costs. In addition, the engineer can provide input at an earlier stage of the project design, which improves design guidance and reduces the amount of structural revision (Kaveh & Shakouri Mahmud Abadi, 2011).

This study presents a workflow to optimize cross sections using parametric and generative design tools that can be implemented at the early design stage. In this context, a parametric steel truss analysis scenario is prepared using Grasshopper and Karamba3D based on design rules for steel structures. In this scenario, different generative design tools/optimizers such as Galapagos, Opposum, and Goat were tested in terms of providing the design option with minimum weight, and their outputs were compared. As a result of this process, has been demonstrated that designers and analysts can define the design parameters and constraints (e.g. material, size, weight, strength, production methods and cost constraints) and design objective in a design software and then can quickly generate hundreds or even thousands of design options and explore all potential solution combinations and discover the best solution.

2. Background

2.1. *Building information modeling*

Building information modelling (BIM) is a process for creating knowledge and managing a construction project along the project life cycle based on semantically rich digital information models. One of the main outputs of BIM process is an information model, a digital representation of a structure. This model benefits from the information that is collectively gathered and updated during all phases of a project. The creation of a digital building information model ensures that the parties interacting with the building can optimize their processes, and in this brings a higher lifetime value to the structure (T. Yousif & Najem, 2014). BIM is a knowledge management and information modelling process conducted in a team environment. For this reason, while implementing BIM method, all team members should work with each other by complying the same standards. At the same time, BIM adds value through the joint efforts of people, processes and technology.

On the other hand, in the BIM processes all information about each component of a building is shared through a central model. This allows any person to use this information for any goal. For example, to combine different aspects of the design in a more effective way. In this way, the risk of errors or inconsistencies is reduced, and unnecessary costs can be minimized. Building Information Modelling supports the continuous and immediate usability of information about the design scope, schedule and cost of a high quality, reliable, integrated and accurately coordinated project. It can also provide many competitive advantages such as faster delivery (time savings), better coordination (fewer errors), lower costs (money savings), higher productivity (efficiency), more skilled labor, new income and career opportunities (Talatahari & Sheikholeslami, 2014). Nevertheless, Building Information Modelling provides access to the following important information for each of the three phases of a building's life cycle: Design, Construction, and Management:

- Design, schedule, and budget information in the design phase.
- Quality, schedule, and cost information in the construction phase.
- Performance, occupancy and financial information in the management phase.

On the other hand, the Building Information Model is more than a 3D model and represents the common data components. Common data components can be expressed through graphical and non-graphical information. There are seven key dimensions of Building Information Modelling according to the established BIM principles (3D - geometry,4D - time,5D - Cost,6D - sustainability, and 7D - facility management).

2.2. *Generative (optimization-focused) design*

Generative design can be described as a rule-based or algorithmic approach in which different possible solutions to a design can be generated. The rules in generative design are likely to include parameters and variables. In generative design, parameters are systematically applied to the design to generate a range of possible outcomes. In generative design an objective/fitness function that can find different solutions with different variables must be defined. In addition, a set of criteria should be described so that the results of the design can be evaluated against the optimum values of the fitness function. The solution is found through increasing or decreasing the input numerical data in cycles to see if the different combinations of input data components would lead to better values of the fitness function.

Generative design tools are inspired by patterns in nature and natural design processes. Generative Design is a design method that takes the designer's goal and generates new solutions. It is based on a highly automated process characterized by a data-centric, collaborative, cloud-based technology. Evolutionary algorithms play an important role in generative design. In evolutionary generative design methods, a set of parameters and rules are accepted as the DNA of the design process; each parameter combination is considered a gene. In these methods genes are combined with evolutionary algorithms for optimization calculations. The application of this type of process in design enables the improvement of new design solutions by changing the rules that define a final design that is difficult or impossible to achieve using other methods (Talatahari & Sheikholeslami, 2014).

2.3. *Design & optimization*

Design is a multidisciplinary activity that requires rigorous research, optimization, and testing until an efficient solution is found and the structure is constructed. Design activities should be well organized and needs to be agile to provide the necessary flexibility to adapt to ever growing markets. For this reason, the basic steps of design activities consist of discovering the numerous alternatives within a solution space, which consists of all possible design solutions for the design task, and the problem space, where all design requirements are explained (Li & Lachmayer, 2018).

Maximizing the desirable factors and minimizing the undesirable factors under the given constraints to find an alternative that costs the least or has the highest available power/output/benefit is called optimization. In contrast, maximization is the attempt to achieve the highest or maximum result without regard to cost or expansion. Optimization is a process of trying to improve or optimize something. Optimization can be described as using the information gained to improve that idea by trying variations of the original concept. Moreover, optimization can be used in any field, from engineering design to financial markets, from everyday activities to vacation planning, in computer science or industrial applications.

2.4. *Structural engineering & optimization*

Problems in the real world are usually nonlinear, and there are quite a few of them in structural engineering as well. To date, a variety of methods have been used to solve such problems, including mathematical theorems, formulations or programs, computer-aided structural analysis software, algorithmic coding, construction automation systems. It is important that the solutions are effective enough to provide an optimal design that considers economic efficiency or balanced cost and ensures structural safety without being overly burdensome. For this reason, optimization solutions that pave the way for updatable and repeatable computational processes have been used in the past, and are still used today, for the design of structural systems to achieve these goals. For example, an early effort in the field was focused on weight minimization for rigid frame structures with steel flanges. For this purpose, a nonlinear programming technique known as gradient projection method was used (Brown & Ang, 1966). Similarly, a nonlinear computer programming system(Colin & MacRae, 1984) was used to generate models for the optimal design of various structural beams (concrete). Moreover, a similar nonlinear programming approach has been applied intended for minimum cost design of simply and eccentrically loaded reinforced concrete columns (Zielinski, Long, & Troitsky, 1995).

Other examples include (Balaguru & Courel, 1984) which has improved a programmable calculation tool to determine the cost-optimal design of concrete slabs together with three differently structured beams. Nevertheless, a design with an optimization of the total cost (material and labor cost) was realized as a minimum level for slabs made of wood in combination with concrete. In this context, a calculation algorithm was further developed and used to calculate the costs (Hanna & Senouci, 1995). Based on all these examples, it can be seen that algorithmic developments and applications have been used for a long time and enhanced through the increasing use of computers and technical devices in the field of structural optimization. Accordingly, algorithms or applications for optimization processes should become faster, more talented and even smarter. In this regard, especially in the last thirty years, heuristic, nature-inspired, evolutionary, adaptive or hybrid algorithms have been much more widely used than the traditional methods.

For example, several metaheuristic algorithms have been developed for the design of optimal columns and beams with different objectives (minimum weight, cost). For example, the genetic algorithm (GA) (Coello, Hernández, & Farrera, 1997) and (T. Yousif & Najem, 2014); harmony search (HS) (Nigdeli et al., 2015); differential evolution method (DE) (Hieu & Tuan, 2018); particle swarm optimization (PSO) (Khashi, Dehghani, & Jahanara, 2018); artificial bee swarm (ABC), bat algorithm (BA), modified bat algorithm (MBA) (Yücel, Bekdaş, & Nigdeli, 2020) ; HS (Kaveh & Shakouri Mahmud Abadi, 2011); Enhanced Charge System Search (ECSS) algorithm (Talatahari & Sheikholeslami, 2014); GA (Mohammad & Ahmed, 2018); Jaya algorithm (JA) (Kayabekir, Arama, Bekdaş, & Dalyan, 2020) were used to create an optimal designs. On the other hand, to create an effective vibration damping system with tuned mass dampers (TMD), base isolation or bracing systems in different structures HS and BA(Gebrail Bekdaş & Nigdeli, 2017); improved harmonic search (IHS)(Zhang & Zhang, 2017); center of gravity optimization (CMO) (Gholizadeh & Ebadijalal, 2018); flower pollination algorithm (FPA) (Yucel, Bekdaş, Nigdeli, & Sevgen, 2019); teaching and learning based optimization (TLBO) (Ulusoy, Niğdeli, & Bekdaş, 2019) were implemented.

2.5. *Weight optimization of plane trusses*

There are a large number of construction models in structural and civil engineering. One of the most studied models is plane truss structures, where attempts have been made to develop an appropriate design to achieve various objectives, such as optimal dimensioning and arrangement, optimization of topology, minimization of weight, cost or volume, and adequate seismic resistance. In this context, different studies have been carried out using optimization approaches. One of these studies was conducted by (Lipson & Agrawal, 1974) to achieve optimal topology and sizing, by minimizing weight using a nonlinear optimization application such as Complex method. Moreover, some nonlinear programming methods such as Zoutendijk method, Powell method, Stewart method etc. have been used and compared to optimize the weight of plane trusses alongside plane stress plates (Jóźwiak, 1989). In (Carpenter & Smith, 1977) the optimal designs were achieved by minimizing the average mass value for trusses using a probability-based problem, which was also converted into a nonlinear programming model. In addition, some studies have been carried out to realize more advanced designs, techniques and applications. One example is the use of an optimization approach based on computer-human interaction for the arrangement of truss structures subjected to multiple loads and the constraints of tensile and compressive stresses and displacements (Adeli & Balasubramanyam, 1987).

Apart from this, it can be inferred from the studies that, nowadays metaheuristics are preferred over other techniques and application tools. The recent studies have shown that GA (Dede, Bekirolu, & Ayvaz, 2011), TLBO (Dede, 2014), DE (Vu, 2015), Simulated Annealing (SA) (Millan-Paramo, 2018), Ant Lion Optimizer (ALO) (Salar & Dizangian, 2019), Crow

Search Algorithm (CSA) (Javidi, Salajegheh, & Salajegheh, 2019) and Artificial Coronary Circulation System (ACCS) algorithm (Kooshkbaghi & Kaveh, 2020) have been used for optimal dimensioning of plane truss structures with minimum weight. Also, optimization of layout and geometry/shape with topology besides dimensioning was done using GA (Chen, Shui, & Huang, 2017; Neeraja, Kamireddy, Kumar, & Reddy, 2017; Rahami, Kaveh, & Gholipour, 2008), Firefly Algorithm (FA) (Miguel, Lopez, & Miguel, 2013), PSO (Harsono et al., 2020; Mortazavi & Toğan, 2016), JA (Degertekin, Lamberti, & Ugur, 2019). In addition, in recent studies, Total Potential Optimization (S. Ç. Toklu, Aydin, & Toklu, 2013), TLBO (G. Bekdaş, Temür, & Toklu, 2015), FPA and JA (Y. C. Toklu, Kayabekir, Bekdaş, Nigdeli, & Yücel, 2020) were used to determine the minimum total potential energy for trusses by optimizing the structural model.

3. Material and methods

3.1. The Search for minimum weight truss

In this research, we focus on development and implementation of a workflow for achieving Minimum Weight Truss design by utilizing BIM tools. The workflow started with utilization of Rhino and Grasshopper (a Rhino add-in and coding environment) for optimizing the structural analysis process. The output model generated by these tools was later transformed into a design environment using Speckle and Dynamo. Rhino, Grasshopper and Karamba3D software was used in the first stage for the structural analysis process. Karamba3D works as an embedded package in Grasshopper and accomplishes the structural analysis using the finite element method. In Grasshopper, the truss plane is generated as parametric object set and later is converted into an analysis model to be used by Karamba3D. Then, with combination of Karamba3D and different generative design tools/optimizers thousands of design options were generated and evaluated within the Grasshopper environment, and the one with the minimum weight is determined as the best fit model. Generative design tools/optimizers such as Galapagos, Opossum and Goat helped in generating the design options and determining the best fit model. The best fit model obtained was then transferred into Dynamo environment via the cloud-based software Speckle and finally transformed into a Revit model using a Dynamo script. The schema of the analysis and transformation workflow is depicted in Figure 1.

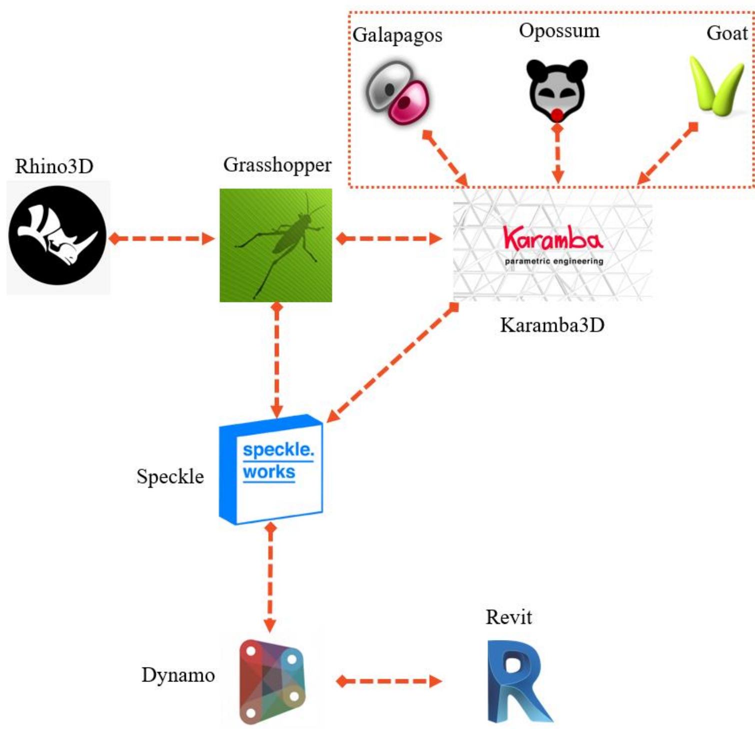

Figure 1. Workflow diagram.

The optimizers used in the current study within Grasshopper were Galapagos, Opossum and Goat. These tools are used in the context of Grasshopper (a well-known design-by-coding environment). An optimization-type oriented taxonomy classifying the main generative design tools used in Grasshopper can be found in Figure 2.

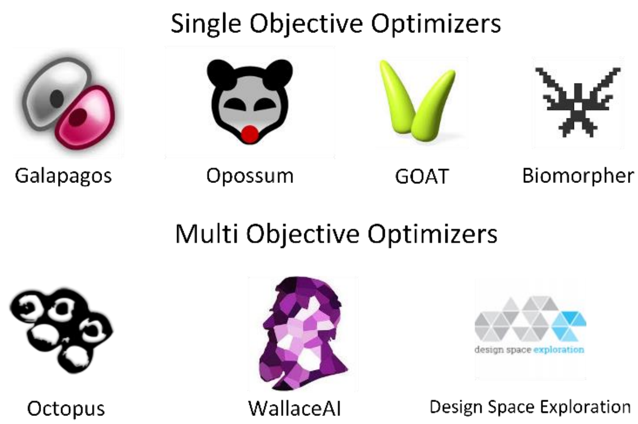

Figure 2. Taxonomy of generative design tools in Grasshopper.

The optimizers differ based on the number of objective (fitness) functions. In terms of the objective function, two different optimization methods exist. The first one is single-objective optimization where a single objective function (e.g. weight) is attempted to be optimized; the other is multi-objective optimization, where there are more than one optimization objectives (e.g. weight and performance).

Many optimization problems can be formulated based on one objective function. However, there are also interesting cases where there are no objective functions in optimization problems. Feasibility problems for example (where the objective is to find the values for variables that satisfy only the constraints of a model) has no specific optimization objective. On the other hand, complementarity problems are widely used in engineering and economics. The objective is to find a solution that satisfies the complementarity constraints. In multi-objective optimization problems optimal decisions must be made in exchange between two or more competing objectives, these problems exist in many fields such as engineering, economics, and logistics.

For example, in developing a new structural component, the objective may be to minimize weight while maximizing performance; or in selecting a portfolio, the objective may be to maximize expected profit while minimizing risk. Also in application scenarios, problems that have more than one objective can be reformulated as a single objective problem by either modifying some objectives with constraints or creating a weighted combination of different objectives. Opossum, Galapagos and Goat were the optimization add-ons for Grasshopper / Rhino that were used in the context of this research. The following elaborates on three optimizers along with Grasshopper and Karamba3D that were also utilized in this study (former as a container of optimizers and latter for 3D Structural Analysis).

3.2. Optimizers, the analysis tool and algorithms

3.2.1. The tools

Grasshopper: Grasshopper is a visual programming language (i.e., a programming language that enables users to create programs by manipulating some elements graphically on a canvas) that operates within Rhinoceros3D CAD. The language is popular among architects and engineers, and generally used for generative and algorithm-based design. There are many plugins to Grasshopper that can be used to facilitate design, structural analysis, performance calculations and optimization.

Opossum: Opossum is the first public model-based optimization tool whose goal is the design optimization. Model-based optimization methods provide good results for a limited number of simulations (Holmström et al., 2008). This high convergence rate is important for design problems where a single simulation is completed in a few minutes or hours. In such cases, it is not practical to run thousands of simulations, as is required for population-based metaheuristic methods such as the genetic algorithm (GA). Other parametric design programs such as Grasshopper, Dynamo, and Design Builder are only equipped with metaheuristic features. However, Opossum can fill this gap by introducing model-based optimization to a wider audience and allowing non-experts to access it. A recent example of the use of Opossum in AEC research can be found in (Wortmann, 2017)which present benchmark of optimization tools for optimizing daylight and glare in Grasshopper.

Galapagos: Galapagos is a component that can achieve a user-defined goal by optimizing a shape in Grasshopper. To work with Galapagos, user would need a set of options or genes (genome) and a defined goal or fitness value. Galapagos uses a typical input-output relation common to multi-objective algorithms, usually genetic algorithms. A "fitness" function and relations of parametric variables are defined as the goal or output to be solved. Galapagos describes the output relations as genomes. These outputs, which Galapagos uses, are constrained by a parametric function in the Grasshopper environment. In Galapagos, fitness functions can be defined with criteria that represent the maximum, minimum, or search for a specific numerical value (Kruchten, 2003). Recent examples of the use of Galapagos in AEC research include (Manni, Lobaccaro, Lolli, & Bohne, 2020) which present a workflow based on an algorithm developed in Grasshopper to parametrically control the building's shape, by maximizing the solar irradiation incident on the building envelope and minimizing the embodied emissions, (Wang, Wu, & Bai, 2022) where the authors propose a method of generating the regular axis from irregular column grids and aim to develop an automatic solution that is repeatable and transplantable.

Goat: This tool complements Galapagos in the best way, as an evolutionary solver. The component follows a rigorous mathematical approach that delivers fast and meaningful results based on unbiased optimization algorithms. (Rechenraum GmbH, n.d.). A recent example of the use of Opossum in AEC research can be found in (Wortmann, 2017) where DIRECT algorithm in Goat was used for performance comparison with RBFOpt / Opossum.

In addition, Karamba3D is used as the structural analysis tool in the implementation.

Karamba3D: Karamba3D (Preisinger, n.d.) is a parametric structural analysis tool that allows the correct and accurate analysis of trusses, frames and shells. Karamba3D is precisely embedded in the parametric design environment of Grasshopper, which is an add-on for Rhinoceros (a 3D modelling tool). Karamba 3D facilitates the combination of geometric models (with parameters), finite element calculations and optimization algorithms such as Galapagos. The structures analyzed by Karamba3D range from simple beams and trusses to bridges and voronoi grid shells. As it works integrated with Grasshopper, Karamba3D can be considered as a BIM tool. The output of Karamba3D can be exchanged and shared by using other BIM tools.

3.2.2. The algorithms

In this section, three optimizers for Grasshopper / Rhino and the variants of algorithms within these tools are presented (Figure 3). Selected algorithms for this implementation are shown in green color. The basics of the implemented algorithms are as follows:

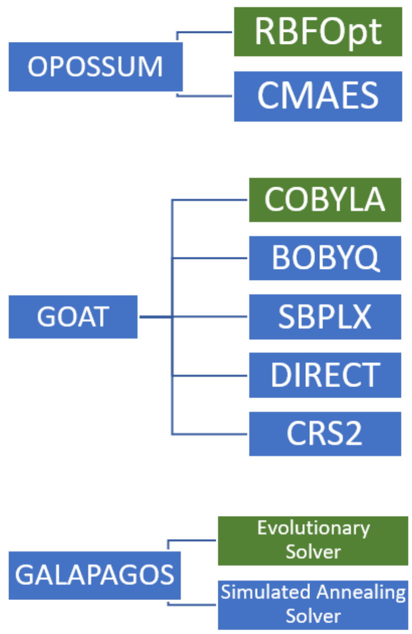

Figure 3. Algorithms of optimizers.

Radial basic function (RBF) algorithm (opossum): Many radial basis function (RBF) algorithms include ε, a free-form parameter that can be adjusted by the user. In the traditional RBD approach, the problem extends to solving a linear system of equations with a system matrix 'A' whose condition number grows as the shape parameter ε approaches zero. On the other hand, standard error predictions show better accuracy via pseudo-power functions to reduce ε. In the literature, this dependence is known as the uncertainty or trading principle and has been accepted as an important issue by many researchers (Fasshauer, Zhang, Fasshauer, & Zhang, 2007).

Constrained optimization by linear approximation (cobyla) algorithm (goat): m.J.D. Powell defined an algorithm that does not use a gradient to solve optimization problems. The algorithm is based on the constrained optimization by linear approximation (COBYLA) method [59]. The optimization problem to be solved is expressed in Eq (1):

$$\begin{gathered} minimize_{\,x}\, J(x) \\ subject\ to\ c_i(x) \geq 0, \quad i = 1{:}m \end{gathered} \tag{1}$$

Here, x is defined as $(x_1,........, x_n)^T \in R^n$, and a vector, where variables take part. Also, J and $c_1,\ldots, c_m$ are real-valued functions. Optimal solution is achieved based on linear iterative approach as Nelder and Mead method. This method aims to calculate the lowest value of a specific $J(x)$ function on variables ($x \in R^n$) without any constraints. R^n is a cluster of $n + 1$ simplex corner points ($x^{(0)}, \ldots, x^{(n)}$), namely $x^{(j)} - x^{(0)}$ vectors are independent as linear for $j = 1 : n$ (Versbach, 2016).

Evolutionary algorithm (Galapagos): Evolutionary algorithms are modeled on nature. An evolutionary algorithm starts with a randomly selected population. Then the population evolves over several generations. In each generation, suitable members are selected as parents. Then these members are interbred to produce new members, the children. After that, the randomly selected child members are subjected to certain mutations. From this process, the algorithm later selects the most suitable members to be passed on to the next generation, according to a predefined survival/selection scheme(de Jong, 2006). The selected members will be among the survivors in the next generation. Such a procedure is used over and over again until a termination condition is met (K. C. Wong, Leung, & Wong, 2010). A typical evolutionary algorithm can be summarized as follows (Table 1):

Table 1. Main components of a typical evolutionary algorithm (K.-C. Wong, 2015).

Choose suitable representation methods;
P(t): Parent Population at time *t*
O(t): Offspring Population at time *t*
t ← 0;
Initialize *P(t);*
while not termination condition **do**
temp = Parent Selection from *P(t);*
O(t +1) = Crossover in *temp;*
O(t+1) = Mutate *O(t+ 1);*
if overlapping **then**
P(t + 1) = Survival Selection from *O(t* + 1) U *P(t)*
else
P(t + 1) = Survival Selection from *O(t* + 1) ;
end if
t ← t + 1;
end while
Good individuals can then be found in *P(t);*

The design of an evolutionary algorithm can be divided into several components. These can be described as selection of representative parents, crossover operators, mutation operators, selection of survivors, and stopping conditions.

3.3. The Search for Minimum Weight Truss

In the first step of the workflow for the cross-section optimization to achieve the minimum-weight truss design, it is necessary to model the structure parametrically, so that an analysis can be performed. As mentioned earlier, the geometry of the was created parametrically in the Grasshopper environment in the first stage (Figure 4).

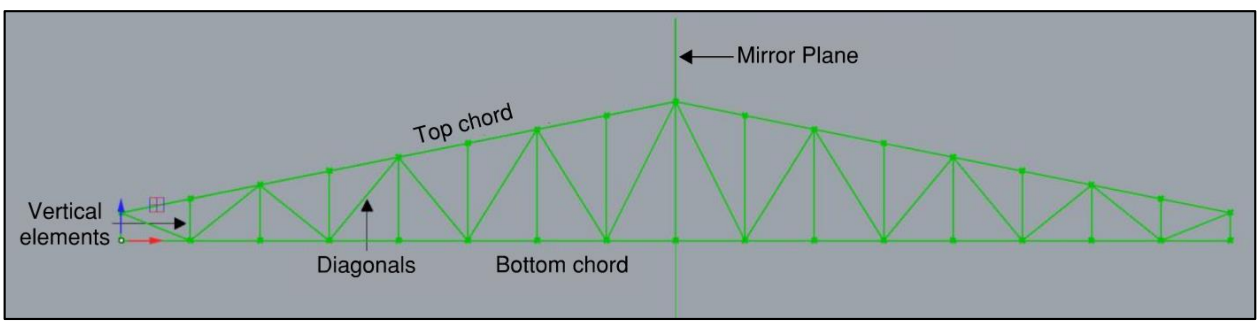

Figure 4. Truss geometry.

Next, each parameter that forms the truss was defined. These were defined using the "Number Sliders" in Grasshopper. These sliders were used to describe the range of values that a parameter can take. First, the minimum and maximum values that can be taken by the parameters used in the scenario were defined. The geometry could only take values in this predetermined range constrained by the parameter boundaries. (Figure 5).

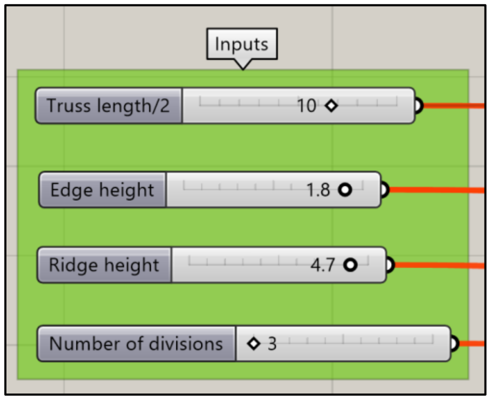

Figure 5. Main parameters for formation of geometry.

For example, the values for the span of trusses were defined between 0 m and 15 m as minimum and maximum respectively; the increment was 1 meter and the parameter was defined as an integer type (Figure 6). For the non-integer values, there were different choices of type depending on the range of use of the parameter.

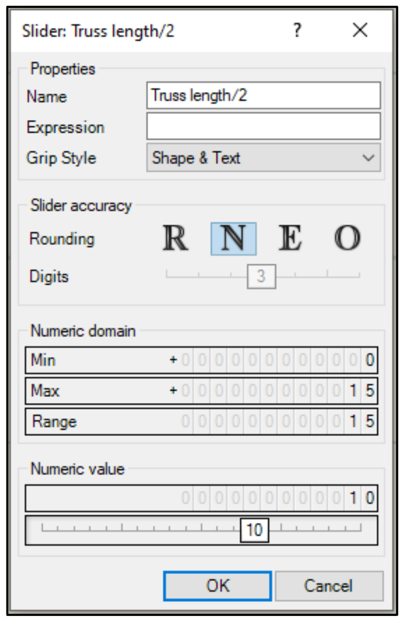

Figure 6. Ranges of values for truss span parameter.

In order to perform a structural analysis in Karamba3D, the geometries from Grasshopper must first be converted into an analysis model. To generate analysis models, there are special commands that Karamba3D provides (i.e. Figure 7). For these special commands, the required information should be defined accordingly and accurately.

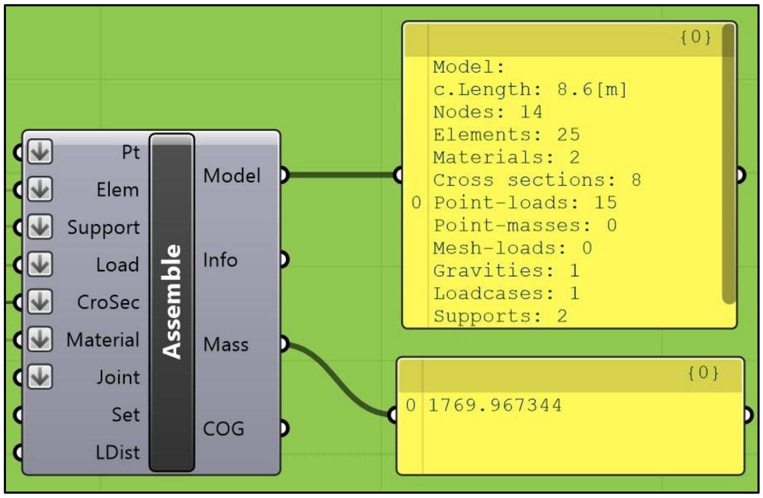

Figure 7. Assemble command.

Having created the elements of the analysis model, the next step was to identify how the model is supported. To identify the support points, special commands of Karamba3D were used, and the values associated with this command were generated as parametric. This special command takes a set of points from Grasshopper and converts them to constant points in the model. For this command to work properly, the identified points should exist as endpoints of beams or corners of shell trusses.

To obtain a valid structural simulation model, one of the most important steps is to define one or more loadings. These loads were calculated parametrically in the scenario by considering engineering rules and pre-defined loading scenarios. The loading scenarios tested with Karamba3D included self-weight, snow loads and wind loads (Figure 8).

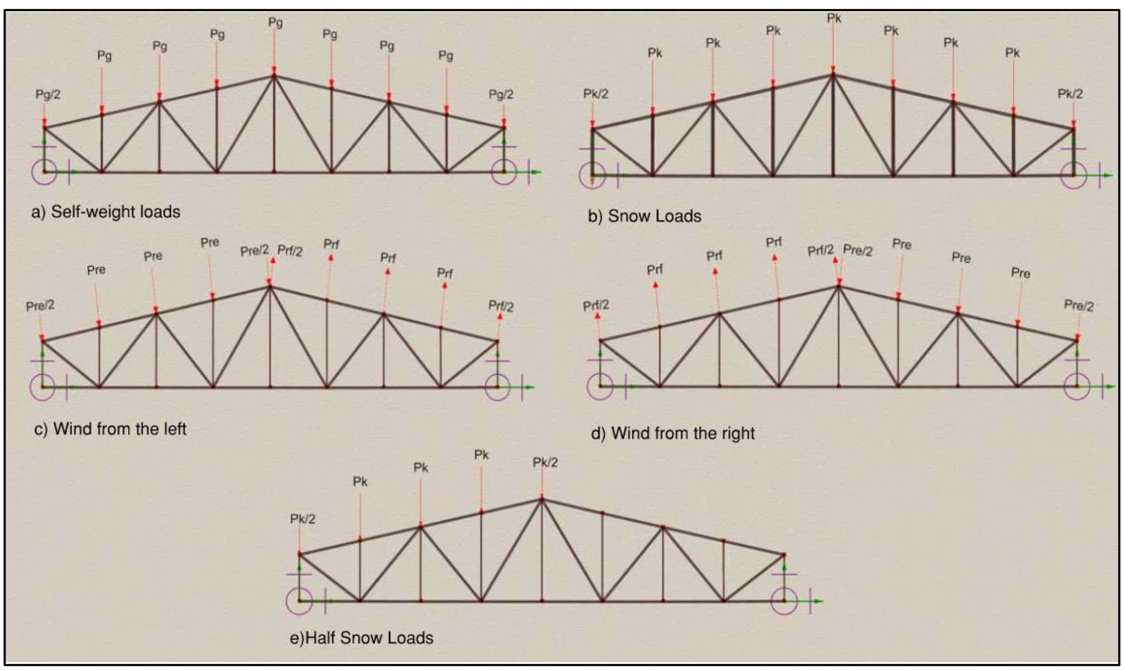

Figure 8. Truss system loads types calculated in scenario.

One of the most important steps that should be taken before performing the analysis was determining the options for cross section and the materials of the beams. There are three different cross-section libraries that can be used for trusses, which are

"HEA" "IPE" and "RHS". A specific cross-section library with these section types was defined as an input for the analysis model, which was then linked to a variable in order to facilitate the optimizer to make a selection among different cross-sections. These cross-section libraries were then linked to generative design tools through Karamba 3D and the selections from cross-section options were made dynamically in each iteration of algorithms in optimizers with the objective of determining the optimum steel tonnage/minimum weight (Figure 9).

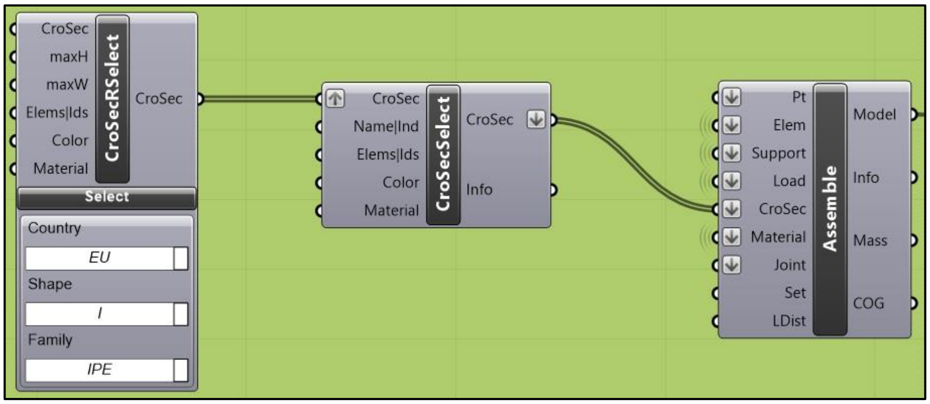

Figure 9. Definition of cross-section.

Other parameter that needs to be defined in the analysis model is the quality of the steel type used. The steel type used in this study was S235. This value is constant throughout the scenario and was not changed by any generative design tool (Figure 10).

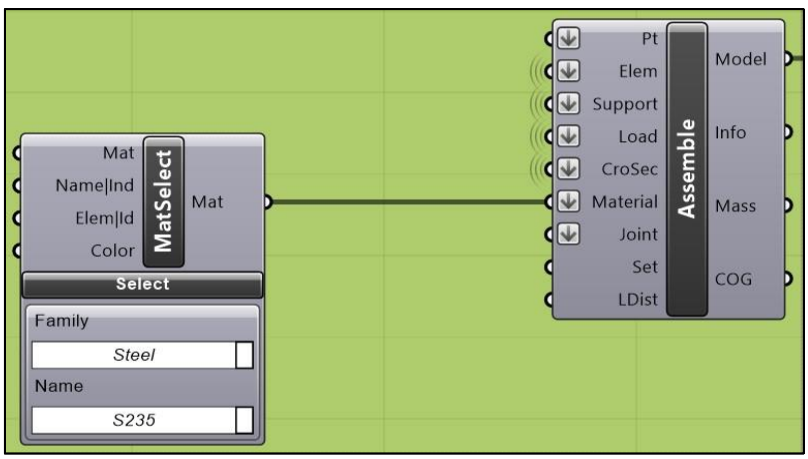

Figure 10. Defining the steel material property.

Following the definition of parameters and design inputs Karamba3D was utilized together with optimizers Opossum, Galapagos (Figure 11) and Goat to detect the best alternatives for each beam section based on the given design options. The Karamba3D commands consider the bearing capacity of the sections and optionally limit the maximum displacement of the structure. In this way, the most optimal sections can be detected considering all the design options and loading scenarios defined in the analysis model. In the end of the analysis, the total tonnage of the optimal sections was calculated and presented by the optimizers.

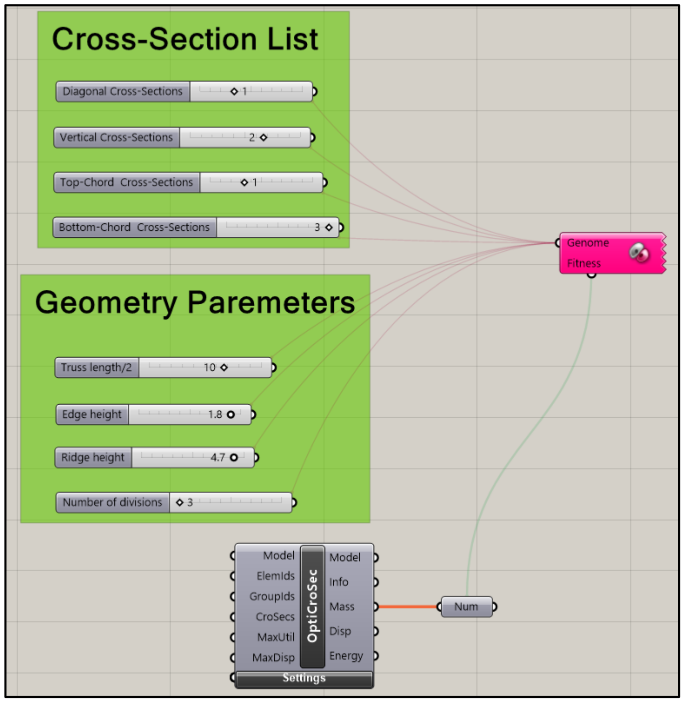

Figure 11. Connections between the design option parameters and the optimizer.

During and following the analysis in Karamba3D, loading set and resulting axial stresses can be visualized and checked in the Grasshopper environment (Figure 12).

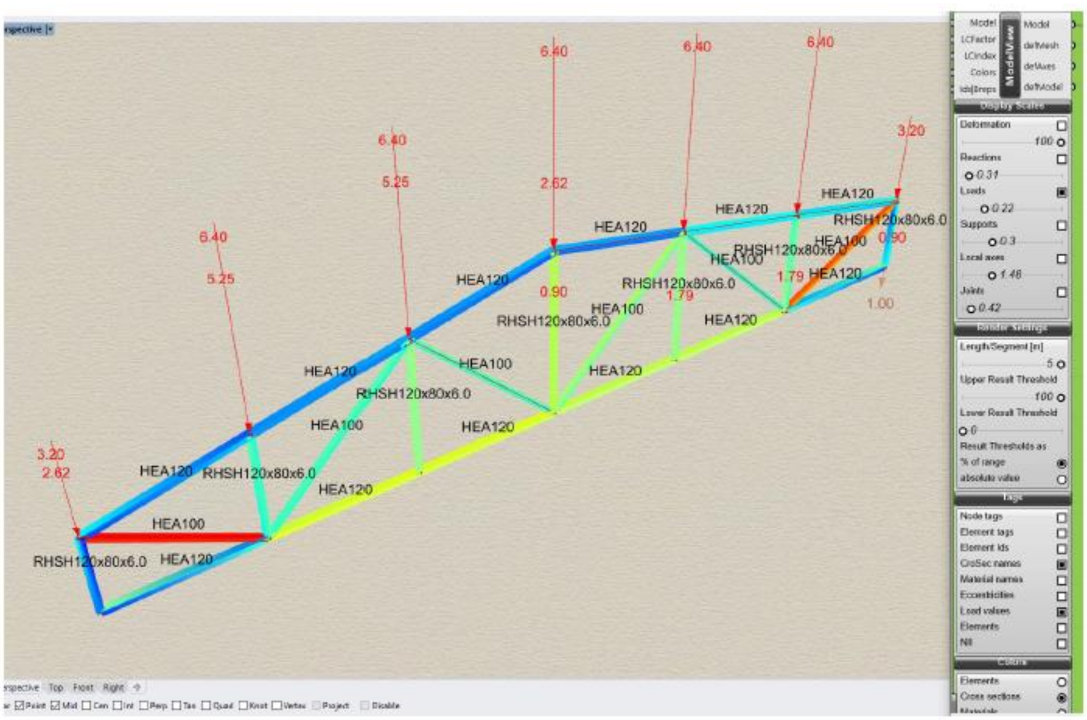

Figure 12. Visualization of a loading set and resulting axial stresses.

The main objective of this study was to determine the minimum weight design alternative by finding out the optimal; i.) cross-sections (for each element) and ii.) overall truss design, for a constant span steel truss system. To achieve the objective of minimum weight (least steel tonnage), the parametric design tools Grasshopper and Karamba3D were used in an integrated manner. Both tools were used to generate a parametric geometry and analysis model. In our scenario, the goal of the generative design tool was to find the most optimal combination (design option) of all sub-combinations with the focus of minimizing the total weight by considering 1.) the side height of the truss, 2.) the ridge height, 3.) the number of truss sections, and 4.) the cross-section library defined for each truss element. Table 2 provides the parameter specifications and their relation with the optimizers.

Table 2. Parameter specifications.

	Parameter	Variable type	Handled by optimizer
Geometry	Height of side	Number slider	Yes
	Ridge height	Number slider	Yes
	Span of truss	Number slider	No
	Division number	Number slider	Yes
Loads	Accepted weights and spans	Number slider	No
	H - HZ loading option	Toggle	No
Material	Quality of steel (S235)	Opener control panel	No
Section	RHS or IPE or HEA	Number slider	Yes

Once the optimum (i.e., minimum weight) design option was found, the optimal model were then transferred into a BIM (Revit) environment using Speckle and Dynamo.

Speckle is an online tool/environment that is developed to facilitate interoperability, automation, and collaboration workflows in AEC industry. The main aim of the Speckle is to simplify data sending and receiving operations (and real-time collaboration), while enabling version control and automation of tedious data exchange operations. It is an open-source platform that resides in a cloud environment and through the use of Speckle it is very easy to transfer information between well-known BIM tools such as Revit, Grasshopper/Rhinoceros, Dynamo, Etabs and Tekla Structures. In addition, Speckle provide connectors to non-BIM tools such as Excel, PowerBI, Unity, Blender which contributes to the interoperability of BIM and non-BIM tools. From a local computer, Speckle platform allows data to be transferred not only to another software, but also between networks and different web platforms. The data that is sent through Speckle is completely under the control of the user. Instead of just having a central server, one can decide what happens to his data [64] using this tool. Figure 13 shows the data transfer in an example Speckle workflow. In this study, the Speckle Cloud (Stefanescu & Cominetti, 2019) was used to transfer the analysis results obtained with Karamba3D and Grasshopper to the BIM environment. There were two specific commands used for this purpose. These commands were Data Sender and Data Receiver and conducted the following tasks; Data Sender (Anonymous Stream) sends the connected data to the cloud environment, Data Receiver (Anonymous Stream) Retrieves the information from the cloud environment in the program in use.

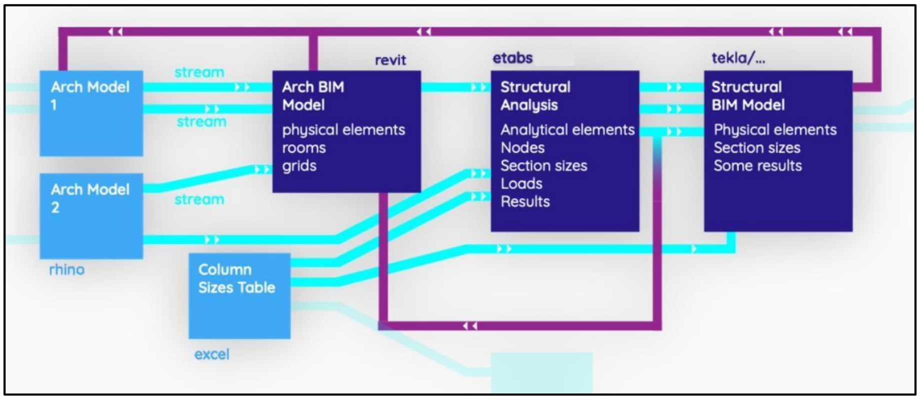

Figure 13. Example speckle workflow.

Following the transfer of information regarding the minimum-weight design option from Grasshopper using Speckle to Dynamo, a Dynamo script was used for generating the objects in Revit. As shown in the Figure 14 Structural Framing. Beam-ByCurve component of the Dynamo has been used to transfer information coming from Speckle into the Revit environment. Three inputs are required to generate the Revit model, "Curve" is used to generate the geometry, "level" is used to denote the Storey and Structural Framing Type is used to define the cross sections of the beams in Revit (it should be noted that in order to generate the beams with correct cross sections in Revit, the components with corresponding cross-sections need to be pre-defined as Families in the Revit environment).

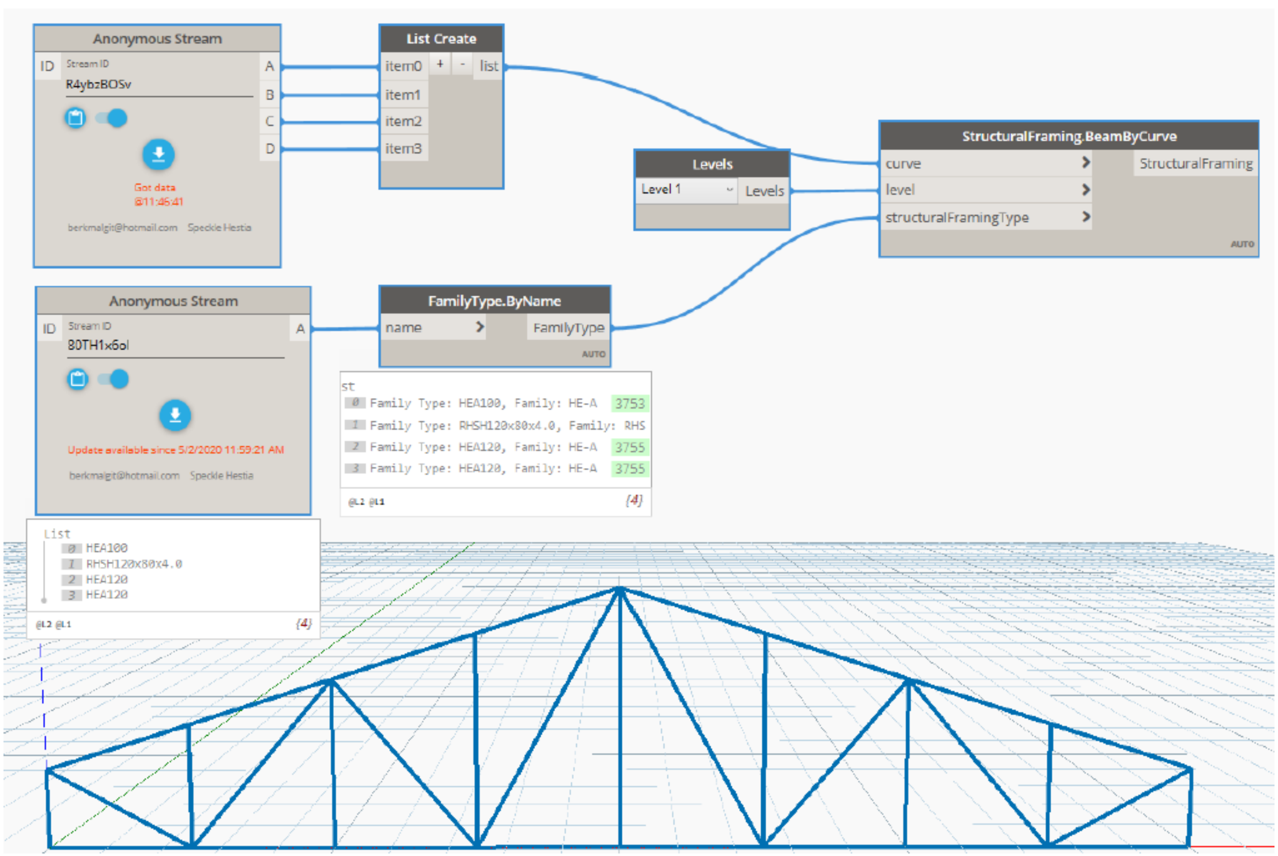

Figure 14. Information transformation by Dynamo.

4. Discussion

In this section, we will provide details on the minimum weight (optimal) models discovered as a result of the integrated structural analysis and generative design process with the use of three different optimization tools, Opossum, Galapagos and Goat. During the analysis/optimization process the hundreds of truss geometry and the cross sections are generated and displayed synchronously in the Rhino/Grasshopper environment. The truss geometry and its cross sections exhibit continuous variability as long as the values are changed rapidly by the optimizers. In particular, when the optimizer is working, observing all different truss types helps to verify the accuracy of the generated design options. The best design options identified by optimizers are provided in this section in Table 3, Table 4 and Table 5. Options are presented as ordered by rank based on minimum weight criterion.

Opossum outputs: The most optimal design found had the weight of 1611 kg (Table 3). It is observed that as the cross-section or geometry change the total weight increases.

Table 3. Optimum cases (minimum-weight truss alternatives) discovered by Opossum.

No	Sub flange	Bay/tie	Diagonal	Top flange	Division number	Ridge height (m)	Height of Side (m)	Span of truss (m)	Total weight (kg)
1	HEA120	RHS120X80X5	HEA100	HEA120	6	5	1.8	20	1611
2	HEA120	RHS120X80X5	HEA100	HEA120	6	5	2	20	1627
3	RHS120X80X6	RHS120X80X6	HEA100	HEA120	6	4.9	2	20	1649
4	HEA120	RHS120X80X6	HEA100	HEA120	6	4.9	1.9	20	1671
5	HEA120	RHS120X80X6	IPE180	HEA120	6	4.9	2	20	1740

Galapagos outputs: The most optimal design option that is identified with Galapagos had the weight of 1624 kg. It is observed that the cross-section values change as the total mass increases. On the other hand, the minimum weight design option discovered by Galapagos were very close to the best result found by Opossum. It was also observed that the results, especially the values of lateral height, were very similar.

Table 4. Optimum cases (minimum-weight truss alternatives) discovered by Galapagos.

No	Sub flange	Bay/tie	Diagonal	Top flange	Division number	Ridge height (m)	Height of Side (m)	Span of truss (m)	Total weight (kg)
1	HEA120	RHS120X80X6	HEA100	HEA120	6	4.5	1.8	20	1624
2	HEA120	RHS120X80X4	HEA100	HEA120	8	4.3	1.6	20	1633
3	HEA120	RHS120X80X4	HEA100	HEA120	8	4.5	1.9	20	1684
4	RHS120X80X6.0	HEA120	HEA100	HEA120	8	4.3	1.7	20	1797
5	HEA120	RHS120X80X5	HEA120	HEA120	8	4	1.9	20	1798

Goat outputs: In contrast with Opossum and Galapagos which provide a list of optimal results (i.e. Table 3, Table 4) in a rank order, Goat provides a (single) optimum result only (Table 5). For Goat, i.e. the last method used in this exercise, the weight of the most optimal combination was determined as 1611 kg, with the section types and geometries presented in Table 5. The output of Goat was exactly identical to the best (most optimal) result obtained with Opossum. Also, the most optimal result obtained with Galapagos was similar to the result obtained with Goat.

Table 5. Optimum cases (minimum-weight truss alternatives) discovered by Goat.

No	Sub flange	Bay/tie	Diagonal	Top flange	Division number	Ridge height (m)	Height of Side (m)	Span of truss (m)	Total weight (kg)
1	HEA120	RHS120X80X5	HEA100	HEA120	6	4.5	1.8	20	1611

5. Conclusions

Building Information Modelling (BIM) is a process of information management that spans through the entire life cycle of a project. The BIM process is carried out based on semantically rich digital information models, where geometry is represented with 3D. In the BIM process, there are several different software working in an integrated and interoperable manner to accomplish different tasks related to the design and construction of a building. In this process, seamless transfer of data between different software is considered as one of the key success factors. The data transfer involves both sharing or exchange of data, either locally, over LAN/WAN or Cloud. Through this seamless data transfer, design/construction, communication, coordination, and information management processes are greatly facilitated. Today structural design is mainly handled by software specifically developed for this purpose. Integration of structural analysis software with other analysis and design applications within a BIM environment, not only facilitates communication and coordination between different stakeholders, but also ensures that robust, reliable and verifiable outputs are produced in the structural analysis phase. In recent years, optimization has become an important aspect of the structural design. The method can be implemented for topology optimization, size optimization and shape optimization. The objective of optimization can be defined as minimizing the weight of the structure under certain material and constraint conditions.

This study aimed to develop and test a new integrated workflow for design of minimum weight trusses and through utilization of the established rules, a cross-section optimization scenario was implemented. Visual programming tools were used to create parametric models in this cross-section optimization scenario. Then, these parametric models were converted into structural models according to the structural design rules. The objective of the exercise was to find the most optimal solution among thousands of parameter combinations that satisfies the design objective (minimum weight) by adjusting the parameters of the structural models using the generative design/optimization tools. According to the analysis results, the Opossum and Goat optimizers found the optimum truss weight as 1611 kgs, while the best result achieved by Galapagos was 1624 kgs. In this context, although the optimum values of the methods are close to each other (the difference between the methods is below 1%), it can be said that Opossum and Goat are considered to be slightly more effective.

Following this, the structural model was transferred into a design environment through utilization of cloud-based integration enablers. An earlier but similar effort (Gerbo & Saliklis, 2014) presented an approach to optimization of efficiency in a building trussed frame. As explained by the authors the optimization was achieved through the repeated construction of models to explore a structure with minimal weight in its columns and diagonal braces, coupled with minimal lateral deflection at the roof level, for a given applied lateral wind load on one side of the trussed frame. The optimization was conducted by Rhino, Grasshopper, Karamba and the evolutionary algorithm Galapagos. This parametric model created a loop that automatically redesigned columns, and braces as well as the width of the bay at each story level. This process is completed with an Evolutionary Solver that uses genetic fitness to eliminate unwanted characteristics and achieve genetic success (i.e., minimum weight as well as minimum lateral roof movement). Although the optimization approach and technique applied by (Gerbo & Saliklis, 2014) were similar to those reported in this paper, our research has the following advantages and unique contributions:

1. The workflow presented in this paper utilized three different optimization tools (Galapagos, Opossum and Goat) instead of one (i.e., Galapagos), and provided an evaluation and comparison of the outputs of these three different tools;

2. The analysis presented in this paper has taken wind and snow loads into account, while the previous research has been focused on weight optimization only by considering the wind-loads;
3. Instead of only an integrated use of Rhino, Grasshopper, Galapagos and Karamba, our research presented the integrated use of additional BIM tools such as Revit (architectural and structural design tool), and integration/interoperability enabler tools such as Speckle and Dynamo (Figure 1);
4. The workflow explained in this paper not only demonstrate an optimization workflow but demonstrates an "optimization and information integration workflow" within a cloud-based BIM environment, thus providing a proof-of-concept for achieving interoperability of several software tools along with providing an optimized design solution (i.e., minimum weight truss).

The implementation of the workflow demonstrated that minimum weight / minimum cost design can be achieved at the early design phase by utilizing generative design, structural analysis and BIM tools in an integrated manner.

Author contributions: Malgıt, B. design and development of the software tools, Isikdag,U. literature review and research design, control of the findings, Bekdaş,G and Yücel.M. control of findings and literature review.

Funding: None

Acknowledgments: None

Conflicts of interest: None

References

Adeli, H., & Balasubramanyam, K. V. (1987). Interactive layout optimization of trusses. Journal of Computing in Civil Engineering, 1(3), 183–196.

Balaguru, P., & Courel, M. (1984). Cost optimum design of reinforced concrete slabs and beams using programmable calculators. Civil Engineering for Practicing and Design Engineers, 3(4), 363–394.

Banichuk, N. V., & Karihaloo, B. L. (1976). Minimum-weight design of multi-purpose cylindrical bars. International Journal of Solids and Structures, 12(4), 267–273.

Bekdaş, G., Temür, R., & Toklu, Y. C. (2015). Analyses of truss structures via total potential optimization implemented with teaching learning based optimization algorithm. 3rd International Conference on Optimization Techniques in Engineering.

Bekdaş, Gebrail, & Nigdeli, S. M. (2017). Metaheuristic based optimization of tuned mass dampers under earthquake excitation by considering soil-structure interaction. Soil Dynamics and Earthquake Engineering, C(92), 443–461.

Bianca, T., Flora, S., Robin, D., John, F., & Jane, B. (2011). Closing the loop of design and analysis: Parametric modelling tools for early decision support. In C. M. Herr, N. Gu, S. Roudavski, & M. A. Schnabel (Eds.), Circuit Bending, Breaking and Mending: Proceedings of the 16th International Conference on Computer-Aided Architectural Design Research in Asia. Hong Kong: Association for Computer-Aided Architectural Design Research in sia (CAADRIA).

Brown, D. M., & Ang, A. H.-S. (1966). Structural optimization by nonlinear programming. Journal of the Structural Division, 92(6), 319–340.

Bucci, F., & Mulazzani, M. (2002). Luigi Moretti: works and writings. New York: Princeton Architectural Press.

Carpenter, W. C., & Smith, E. A. (1977). Computational efficiency of non-linear programming methods on a class of structural problems. International Journal for Numerical Methods in Engineering, 11(8), 1203–1223.

Chen, S. yan, Shui, X. fang, & Huang, H. (2017). Improved genetic algorithm with two-level approximation using shape sensitivities for truss layout optimization. Structural and Multidisciplinary Optimization, 55(4), 1365–1382.

Coello, C. C., Hernández, F. S., & Farrera, F. A. (1997). Optimal design of reinforced concrete beams using genetic algorithms. Expert Systems with Applications, 12(1), 101–108.

Colin, M. Z., & MacRae, A. J. (1984). Optimization of structural concrete beams. Journal of Structural Engineering, 110(7), 1573–1588.

de Jong, K. A. (2006). Evolutionary Computation. A Unified Approach. MIT Press.

Dede, T. (2014). Application of teaching-learning-based-optimization algorithm for the discrete optimization of truss structures. KSCE Journal of Civil Engineering, 18(6), 1759–1767.

Dede, T., Bekirolu, S., & Ayvaz, Y. (2011). Weight minimization of trusses with genetic algorithm. Applied Soft Computing, 11(2), 2565–2575.

Degertekin, S. O., Lamberti, L., & Ugur, I. B. (2019). Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm. Applied Soft Computing, 79, 363–390.

Fasshauer, G. E., Zhang, J. G., Fasshauer, G. E., & Zhang, J. G. (2007). On choosing "optimal" shape parameters for RBF approximation. Numerical Algorithms, 45(1), 345–368.

Gerber, D. J. (2009). The parametric affect: Computation, innovation and models for design exploration in contemporary architectural practice.

Gerber, D., & Lin, S.-H. (2012). Designing-in performance through parameterization, automation, and evolutionary algorithms. In T. Fischer, K. de Biswas, J. J. Ham, R. Naka, & W. X. Huang (Eds.), Beyond Codes and Pixels: Proceedings of the 17th International Conference on Computer-Aided Architectural Design Research in Asia. Hong Kong.

Gerbo, E. J., & Saliklis, E. P. (2014). Optimizing a Trussed Frame Subjected to Wind Using Rhino, Grassho...: Ingenta Connect. Proceedings of IASS Annual Symposia, IASS 2014 Brasilia Symposium: Shells, Membranes and Spatial Structures: Footprints – Historical Spatial Structures, 1–7.

Gholizadeh, S., & Ebadijalal, M. (2018). Performance based discrete topology optimization of steel braced frames by a new metaheuristic. Advances in Engineering Software, 123, 77–92.

Hanna, A. S., & Senouci, A. B. (1995). Design optimization of concrete-slab forms. Journal of Construction Engineering and Management, 121(2), 215–221.

Harsono, K., Prayogo, D., Prasetyo, K. E., Wong, F. T., Tjandra, D., Harsono, K., … Tjandra, D. (2020). Comparative study of particle swarm optimization algorithms in solving size, topology, and shape optimization. JPhCS, 1625(1), 012015.

Hieu, N. T., & Tuan, V. A. (2018). Weight optimization of composite cellular beam based on the differential evolution algorithm. Journal of Science and Technology in Civil Engineering (STCE) - NUCE, 12(5), 28–38.

Holmström, K., Quttineh, N.-H., Edvall, M. M., Holmström, K., Quttineh, N.-H., & Edvall, M. M. (2008). An adaptive radial basis algorithm (ARBF) for expensive black-box mixed-integer constrained global optimization. Optimization and Engineering, 9(4), 311–339.

Javidi, A., Salajegheh, E., & Salajegheh, J. (2019). Enhanced crow search algorithm for optimum design of structures. Applied Soft Computing, 77, 274–289.

Jóźwiak, S. F. (1989). Probability-based optimization of truss structures. Computers & Structures, 32(1), 87–91.

Kaveh, A., & Shakouri Mahmud Abadi, A. (2011). Harmony search based algorithms for the optimum cost design of reinforced concrete cantilever retaining walls. International Journal of Civil Engineering, 9(1), 1–8.

Kayabekir, A. E., Arama, Z. A., Bekdaş, G., & Dalyan, İ. (2020). L-shaped reinforced concrete retaining wall design: cost and sizing optimization. Challenge Journal of Structural Mechanics, 6(3), 140.

Khashi, K., Dehghani, H., & Jahanara, A. A. (2018). An optimization procedure for concrete beam-column joints strengthened with FRP. Iran University of Science & Technology, 8(4), 675–687.

Kooshkbaghi, M., & Kaveh, A. (2020). Sizing optimization of truss structures with continuous variables by artificial coronary circulation system algorithm. Iranian Journal of Science and Technology - Transactions of Civil Engineering, 44(1), 1–20.

Kruchten, N. J. (2003). Galapagos: A Distributed Parallel Evolutionary Algorithm Development Platform.

Li, H., & Lachmayer, R. (2018). Generative design approach for modeling creative designs. IOP Conference Series: Materials Science and Engineering, 408(1), 012035.

Lipson, S. L., & Agrawal, K. M. (1974). Weight optimization of plane trusses. Journal of the Structural Division, 100(5), 865–879.

Manni, M., Lobaccaro, G., Lolli, N., & Bohne, R. A. (2020). Parametric design to maximize solar irradiation and minimize the embodied ghg emissions for a zeb in nordic and mediterranean climate zones. Energies, 13(18).

Miguel, L. F. F., Lopez, R. H., & Miguel, L. F. F. (2013). Multimodal size, shape, and topology optimisation of truss structures using the Firefly algorithm. Advances in Engineering Software, 56, 23–37.

Millan-Paramo, C. (2018). Modified simulated annealing algorithm for discrete sizing optimization of truss structure. Jordan Journal of Civil Engineering, 12(4), 683–695.

Mohammad, F., & Ahmed, H. (2018). Optimum design of reinforced concrete cantilever retaining walls according Eurocode 2 (EC2). Athens Journal of Technology & Engineering, 5(3), 277–296.

Mortazavi, A., & Toğan, V. (2016). Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer. Structural and Multidisciplinary Optimization, 54(4), 715–736.

Neeraja, D., Kamireddy, T., Kumar, P. S., & Reddy, V. S. (2017). Weight optimization of plane truss using genetic algorithm. IOP Conference Series: Materials Science and Engineering, 263(3), 032015.

Nigdeli, S. M., Bekdas, G., Kim, S., & Geem, Z. W. (2015). A novel harmony search based optimization of reinforced concrete biaxially loaded columns. Structural Engineering and Mechanics: An International Journal, 54(6), 1097-1109.

Preisinger, C. (n.d.). Karamba3D. Vienna: Bollinger und Grohmann ZT GmbH.

Rahami, H., Kaveh, A., & Gholipour, Y. (2008). Sizing, geometry and topology optimization of trusses via force method and genetic algorithm. Engineering Structures, 30(9), 2360–2369.

Rechenraum GmbH. (n.d.). GOAT. Wien. Retrieved from https://www.rechenraum.com/en/goat.html

Rosen, D. W., & Peters, T. J. (1996). The role of topology in engineering design research. Research in Engineering Design, 8(2), 81–98.

Salar, M., & Dizangian, B. (2019). Sizing optimization of truss structures using ant lion optimizer. 2nd International Conference on Civil Engineering, Architecture and Urban Management in Iran.

Stefanescu, D., & Cominetti, M. (2019). Speckle Cloud. Speckle Systems.

T. Yousif, S., & Najem, R. M. (2014). Effects of materials properties on strength and deflection of optimized r.c. beams: genetic algorithms. College of Engineering, University of Mosul, 22(5), 149–165.

Talatahari, S., & Sheikholeslami, R. (2014). Optimum design of gravity and reinforced retaining walls using enhanced charged system search algorithm. KSCE Journal of Civil Engineering, 18(5), 1464–1469.

Toklu, S. Ç., Aydin, N., & Toklu, Y. C. (2013). Analyzing structures: Total potential optimization using meta-heuristic algorithms (TPO/MA). Proceedings of International Symposium on Computing in Science & Engineering.

Toklu, Y. C., Kayabekir, A. E., Bekdaş, G., Nigdeli, S. M., & Yücel, M. (2020). Analysis of plane-stress systems via total potential optimization method considering nonlinear behavior. Journal of Structural Engineering, 146(11), 04020249.

Ulusoy, S., Niğdeli, S. M., & Bekdaş, G. (2019). Optimization of PID controller parameters for active control of single degree of freedom structures. Challenge Journal of Structural Mechanics, 5(4), 130.

Versbach, L. M. (2016). Evaluation of a gradient free and a gradient based optimization algorithm for industrial beverage pasteurization described by different modeling variants.

Vu, V. T. (2015). Weight minimization of trusses with natural frequency constraints. 11th World Congress on Structural and Multidisciplinary Optimisation. Sydney.

Wang, X., Wu, C., & Bai, C. (2022). Generating the regular axis from irregular column grids through genetic algorithm. Applied Sciences, 12(4).

Wong, K. C., Leung, K. S., & Wong, M. H. (2010). Effect of Spatial Locality on an Evolutionary Algorithm for Multimodal Optimization. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6024 LNCS (PART 1), 481–490.

Wong, K.-C. (2015). Evolutionary Algorithms: Concepts, designs, and applications in bioinformatics: Evolutionary algorithms for bioinformatics.

Wortmann, T. (2017). Model-based optimization for architectural design: optimizing daylight and glare in Grasshopper. Architecture + Design, 1(2), 176–185.

Yücel, M., Bekdaş, G., & Nigdeli, S. M. (2020). Minimizing the weight of cantilever beam via metaheuristic methods by using different population-iteration combinations. WSEAS Transactions in Computers, 19, 69–77.

Yucel, M., Bekdaş, G., Nigdeli, S. M., & Sevgen, S. (2019). Estimation of optimum tuned mass damper parameters via machine learning. Journal of Building Engineering, 26, 100847.

Zhang, H. Y., & Zhang, L. J. (2017). Tuned mass damper system of high-rise intake towers optimized by improved harmony search algorithm. Engineering Structures, 138, 270–282.

Zielinski, Z. A., Long, W., & Troitsky, M. S. (1995). Designing reinforced concrete short-tied columns using the optimization technique. Structural Journal, 92(5), 619–626.