



**UNIVERSIDADE FEDERAL DE SERGIPE  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA CIVIL**



**MATEUS CARDOSO OLIVEIRA**

**USO DO PROJETO GENERATIVO COMO FERRAMENTA DE BUSCA DE  
SOLUÇÕES DE PROJETO**

Aplicado a vigas de concreto armado

**São Cristóvão - SE**

**2021**

**Mateus Cardoso Oliveira**

**USO DO PROJETO GENERATIVO COMO FERRAMENTA DE BUSCA DE  
SOLUÇÕES DE PROJETO**

Aplicado a vigas de concreto armado

Trabalho de conclusão de curso apresentado à  
Universidade Federal de Sergipe para  
encerramento do componente curricular e  
conclusão da graduação em Engenharia Civil.

Orientador: Prof. Dr. Marco Antônio Brasiel  
Sampaio

**São Cristóvão - SE**

**2021**

É concedida à Universidade Federal de Sergipe permissão para reproduzir cópias desta monografia e emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste trabalho acadêmico pode ser reproduzida sem a autorização por escrito do autor.

Assinatura

Oliveira, Mateus Cardoso

Projeto generativo como ferramenta de análise de soluções de projeto: aplicado a vigas de concreto armado / Mateus Cardoso Oliveira. – 2021.

61 f.: il. Color

Trabalho de Conclusão de Curso (Graduação em Engenharia Civil) – Ciência Exatas e Tecnologia da Universidade Federal de Sergipe, São Cristóvão, 2021.

“Orientação: Prof. Dr. Marco A. B. Sampaio”.

1. Projeto generativo. 2. Algoritmo genético. 3. BIM. I. Sampaio, Marco A. B. (orientador). II. Título.

**Mateus Cardoso Oliveira**

**Título**

Trabalho de conclusão de curso apresentado à Universidade Federal de Sergipe para encerramento do componente curricular e conclusão da graduação em Engenharia Civil.

Orientador: Prof. Dr. Marco Antônio Brasiel Sampaio

Aprovado em:

Banca Examinadora:

\_\_\_\_\_  
Nota: \_\_\_\_\_

Prof. Dr. Marco Antônio Brasiel Sampaio  
(Presidente - Orientador)

\_\_\_\_\_  
Nota: \_\_\_\_\_

Prof. Dr. Daniel Moureira Fontes Lima  
(Examinador 1)

\_\_\_\_\_  
Nota: \_\_\_\_\_

Prof. Dr. Jorge Carvalho Costa  
(Examinador 2)

*Dedico todo este trabalho a minha irmã, pais, cunhado e sobrinho. A minha amada namorada. Aos amigos Thomaz, Wislan, Hilton, Raphael, Jéssica e Will. Ao professor Marco pelas valiosas orientações.*

## RESUMO

No contexto de avanço de métodos de automação e otimização de projetos, este trabalho dedica-se a avaliar o uso do projeto generativo como ferramenta de busca de soluções de projeto aplicado a vigas de concreto armado. Para tanto, apresenta conceitos básicos presentes no projeto generativo como o algoritmo genético, projeto computacional, otimização multiobjetivo e integração com o BIM. Em seguida, são detalhadas as condições do exemplo de aplicação, este, é desenvolvido para projetos no *Revit*, cujo projeto computacional é desenvolvida no *Dynamo*. O projeto computacional gera uma viga de concreto armado a partir de informações do *Revit*, e então avalia algumas de suas propriedades como custo, deslocamento vertical e comprimento da armadura. Por isso, o projeto computacional é dividido em estrutura de entrada, algoritmo gerador, algoritmo avaliador e estrutura de saída. O projeto generativo gerou os resultados esperados, operou corretamente com o projeto do *Revit*, desenhou a Fronteira de Pareto e explorou o espaço de projeto de acordo com a proposta de cada solucionador. Portanto, atestando o correto funcionamento da estrutura computacional paramétrica e constatando o desempenho do projeto generativo como ferramenta de busca de soluções de projeto.

Palavras-chave: Projeto Generativo. Algoritmo Genético. BIM.

## **ABSTRACT**

Considering automation and optimization methods improvement, this project dedicates to evaluating the employing of Generative Design as a tool capable of finding design solutions to reinforced concrete beams. Therefore, it presents basic concepts of generative design as genetic algorithm, computational design, multi-objective optimization and integration with BIM. Then, the conditions of the applied example are detailed to projects in Revit and the computational project is developed in Dynamo. The computational design generates a reinforced concrete beam from Revit's information and analyzes some of its properties like cost, vertical displacement, and reinforcement length. Hence, the computational project is divided into input structure, genetic algorithm, evaluator algorithm and output structure. The generative project achieved expected results, collaborated correctly with the Revit project, plotted Pareto solutions, and explored the project space according to each solver's characteristics. Ultimately, it was possible to attest the correct functioning of the parametric computational structure, as well as verify the performance of the generative design to search for design solutions.

**Keywords:** Generative Project. Genetic Algorithm. BIM.

# SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	Objetivos .....	13
2	FUNDAMENTAÇÃO TEÓRICA .....	13
2.1	Aprendizagem da Inteligência Artificial .....	13
2.2	Algoritmo Genético .....	14
2.3	Projeto computacional .....	16
2.4	Espaço de projeto.....	18
2.4.1	Viés versus Variância .....	18
2.4.2	Complexidade versus Continuidade .....	19
2.5	Otimização multiobjetivo .....	19
2.6	Estágios do projeto generativo.....	21
2.7	Dimensionamento da armadura da viga .....	24
2.8	Integração com BIM.....	26
2.8.1	Generative Design for Revit.....	28
2.8.2	Dynamo for Revit.....	29
2.8.3	Nós Data.Remember e Data.Gate .....	31
2.9	Exemplos de projeto generativo.....	31
2.9.1	Redução de resíduos na modelação de casa de madeira.....	31
2.9.2	Estudo de caso de bloco residencial.....	32
2.9.3	Otimização de desempenho térmico.....	33
3	METODOLOGIA .....	34
3.1	Exemplo de aplicação.....	35
3.2	Projeto Computacional .....	36
3.2.1	Estruturas de entrada .....	36
3.2.2	Algoritmo Gerador .....	38
3.2.3	Algoritmo Avaliador.....	38
3.2.4	Saídas e Interoperabilidade com o Revit .....	41



3.3	Solucionador.....	42
4	ANÁLISE DOS RESULTADOS.....	43
4.1	Funcionamento do modelo .....	43
4.2	Exploração do espaço de projeto por diferentes solucionadores .....	50
5	CONSIDERAÇÕES FINAIS.....	53
6	REFERÊNCIAS .....	55

## LISTA DE FIGURAS

Figura 1 - Representação cromossomial.....	15
Figura 2 - Esquema básico do AG .....	16
Figura 3 - Crossover e Mutação .....	16
Figura 4 - Função de geração .....	17
Figura 5 - Função de avaliação .....	17
Figura 6 - Relação complexidade vs continuidade. ....	19
Figura 7 - Exemplo de Fronteira de Pareto .....	20
Figura 8 - Frentes ideais descontínuas .....	21
Figura 9 - Etapa de geração.....	21
Figura 10 - Etapa de análise .....	22
Figura 11 - Etapa de classificação .....	22
Figura 12 - Etapa de evolução .....	22
Figura 13 - Etapa de exploração .....	23
Figura 14 - Etapa de integração .....	23
Figura 15 - Fluxograma de estágios do projeto generativo .....	23
Figura 16 - Estrutura otimizada por projeto generativo no <i>Rhino-Grasshopper to ArchiCAD</i> .....	28
Figura 17 - Anatomia de um nó de controle deslizante .....	30
Figura 18 - Nó personalizado de dimensionamento .....	30
Figura 19 - Exemplo de modelação da casa de madeira .....	32
Figura 20 - Definição do espaço de projeto.....	33
Figura 21 - Opções de projeto.....	33
Figura 22 - Fronteira de Pareto .....	34
Figura 23 - Projeto de aplicação.....	35
Figura 24 - Esquema de Carregamento das vigas .....	35
Figura 25 - Gráfico da estrutura paramétrica.....	36
Figura 26 - Representação cromossomial.....	37
Figura 27 - Aplicação do nó <i>Data.Remember</i> .....	41
Figura 28 - Aplicação do nó <i>Data.Gate</i> .....	42
Figura 29 - Configurações da geração .....	42
Figura 30 - Criar estudo .....	43
Figura 31 - Opções de solucionador .....	43
Figura 32 - Metas do estudo com dois objetivos .....	44

Figura 33 - Definição das restrições.....	45
Figura 34 - Resultados modelo 1 .....	46
Figura 35 - Resultados modelo 2 .....	46
Figura 36 - Resultado para um objetivo .....	47
Figura 37 - Frente de Pareto objetivo duplo com 10 gerações.....	48
Figura 38 - Frente de Pareto objetivo duplo com 20 gerações.....	48
Figura 39 - Frente de Pareto objetivo duplo com 40 gerações.....	49
Figura 40 - Frente ideal de objetivo triplo: Cores mais quentes para maiores comprimentos.....	49
Figura 41 - Frente ideal de objetivo triplo: Cores mais frias para os menores diâmetros.....	50
Figura 42 - Integração da opção de projeto no modelo.....	50
Figura 43 - Espaço de projeto do Método Randomizar .....	51
Figura 44 - Espaço de projeto do Método Semelhante .....	51
Figura 45 - Espaço de projeto da otimização .....	52
Figura 46 - Espaço de projeto do produto transversal.....	52

## LISTA DE TABELAS

Tabela 1 - Carregamentos e Solicitações .....	36
Tabela 2 - Custo concreto usinado.....	38
Tabela 3 - Custo da armadura longitudinal .....	39
Tabela 4 - Custo armadura transversal .....	39
Tabela 5 - Peso do aço .....	39

## 1 INTRODUÇÃO

Com o objetivo de solucionar problemas computacionais complexos, é comum aos cientistas recorrer a princípios biológicos como fonte de inspiração, neste caso, trata-se de uma área de pesquisa conhecida como Computação Bioinspirada. São exemplos de modelos bioinspirados as Redes Neurais – que simulam o comportamento do sistema nervoso, Inteligência de Enxame – simula o comportamento de voo de enxame de abelhas, Sistemas Imunológicos Artificiais – simula o funcionamento do sistema imunológico, Recozimento Simulado – simula o processo de recozimento de metais, entre outros que compõem essa linha de pesquisa. Neste sentido, inspirados na Biologia Evolutiva, os Algoritmos Evolutivos (AE) se inspiram na Teoria da Evolução das Espécies proposta por Darwin. Esta, tem como principal classe os Algoritmos Genéticos (AG) (LINDEN, 2008; LACERDA & CARVALHO, 1999).

Os AGs são um tipo de inteligência artificial que opera como uma ferramenta de busca. Estes, procuram por possibilidades de solução direcionados por operadores genéticos, isto faz deles excelentes ferramentas de otimização. Com efeito, os AGs têm papel central no Projeto Generativo (KRISH, 2011; LINDEN, 2008).

A utilização de AEs, e em especial AGs, como método de criação de soluções de projeto é chamada de Projeto Generativo. Como ferramenta de busca o AG procura entre as opções de projeto, aqueles que melhor performam frente aos objetivos do projeto (NAGY, 2017).

Naturalmente, propostas de estruturas de concreto armado envolvem problemas multiobjetivos, pois, devem apresentar uma solução adequada a arquitetura, ao mesmo tempo que atenda aos estados limites e minimize custos. Muitas vezes estes objetivos são conflitantes entre si, complicando a tarefa de escolher a melhor solução. Neste sentido o Projeto Generativo pode ser uma eficiente ferramenta auxiliar na exploração de opções de projeto na área de estruturas.

Outro aspecto importante do Projeto Generativo, é a sua utilização no BIM como ferramenta para potencializar recursos de automação. De fato, a solução do problema de projeto pode ser expressa como um modelo paramétrico e facilmente integrada a um software na plataforma BIM (ABRISHAMI, GOULDING, RAHIMIAN, & GANAH, 2014).

Sendo assim, percebe-se a necessidade de estudar o uso do Projeto Generativo como ferramenta de busca de soluções de projeto de estruturas. Para aplicação deste estudo foi escolhido o projeto de vigas de concreto armado.

### **1.1 Objetivos**

O objetivo geral do presente trabalho é avaliar o desempenho do Projeto Generativo como ferramenta auxiliar para projeto de vigas de concreto armado.

Para isso, foram delimitados os seguintes objetivos específicos:

1. Apresentar fundamentos teóricos básicos do Projeto Generativo;
2. Modelar estrutura computacional paramétrica adequado ao estudo de soluções de projeto de viga de concreto armado;
3. Analisar os resultados do projeto computacional do item 2 e relacioná-los com os conceitos abordado no item 1.

No segundo capítulo é feita a fundamentação teórica. Primeiro sobre Projeto Generativo: contextualização; apresentação de conceitos importantes dos Algoritmos Genéticos e do projeto computacional. Em seguida, sobre projeto de viga, plataforma BIM, sua interoperabilidade com o Projeto Generativo e apresentação de alguns exemplos.

No terceiro capítulo são apresentados os métodos para modelagem do projeto computacional e obtenção dos resultados aplicados a vigas de concreto armado.

Por fim, no quarto capítulo são apresentados e discutidos os resultados do projeto generativo aplicado ao exemplo de vigas e comparando-os com os conceitos vistos nos capítulos anteriores.

Nas considerações finais são resgatados os objetivos e como eles foram atendidos. Por fim, apresentadas as limitações das pesquisas e as perspectivas para futuros trabalhos.

## **2 FUNDAMENTAÇÃO TEÓRICA**

### **2.1 Aprendizagem da Inteligência Artificial**

Com a invenção dos primeiros computadores, programadores e pesquisadores acreditavam que bastaria apenas o avanço tecnológico do hardware

para possibilitar a programação de Inteligência Artificial (IA). O rápido avanço tecnológico dos computadores confirmava as suas aspirações. Assim, velocidade de processamento e memória das máquinas deixou de ser uma limitação, no entanto, o desenvolvimento da IA não avançou como o esperado. O que estava impedindo esse avanço? (NAGY, 2017).

Na verdade, segundo Nagy (2017), programar uma inteligência artificial é mais difícil do que se imaginava, de modo que, um humano não seria capaz de programar diretamente a IA. Por isso, como solução, ao invés de programá-la diretamente poderiam criar um modelo que possibilite o desenvolvimento e aprendizado da máquina por si só. Desde então, uma IA que aprende sozinha se tornou a forma mais comum. Neste cenário, o projetista estrutura objetivos, delimita possibilidades, restrições e a partir disso o computador deve encontrar sozinho a melhor maneira de fazer seu trabalho, se tornando inteligente por conta própria.

Na década de 1950 o cientista da computação Arthur Samuel criou um programa que joga damas capaz de aprender com a experiência e melhorar suas jogadas. O programa era impressionante pois aprendia jogando contra humanos e outros programas. Neste sentido, Buchanan afirma: “Sempre que tentamos identificar o que está no cerne da inteligência, a aprendizagem certamente será mencionada” (BUCHANAN, 2006).

Com efeito a IA envolvida no Projeto Generativo pertence a classe de Algoritmos Metaheurísticos que, por sua vez, trata-se de um Método Estocástico de otimização. É Metaheurístico pois consiste numa busca inicial aleatória e, na sequência, usa o conhecimento adquirido para direcionar uma nova iteração de busca, e assim sucessivamente até alcançar o resultado. Bem como, é estocástico, pois, envolve um certo nível de aleatoriedade que resulta em soluções imprevisíveis (NAGY, 2017). Será abordado nos capítulos seguintes o Algoritmo Genético, pois é o Algoritmo Metaheurístico mais comum no Projeto Generativo.

## **2.2 Algoritmo Genético**

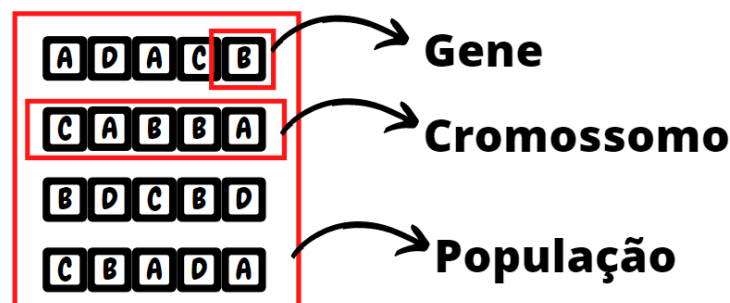
Na década de 1940 cientistas começaram a buscar inspiração na natureza para desenvolver IA, pode-se dizer que este é o marco do início da história dos AGs. Todavia, até a década seguinte as pesquisas se concentraram na área cognitiva e nos processos de raciocínio e aprendizado, logo iniciou-se a busca por soluções mais

genéricas para resolver problemas mais complexos. Foi somente em 1960 que o cientista John Holland inventou AGs capazes de resolver problemas insolúveis até aquela época (LINDEN, 2008).

Os AGs como uma classe de Algoritmo Evolucionário são ferramentas de pesquisa e otimização inspirados na Evolução Natural. Tal como na biologia, a evolução ocorre no código genético e reflete diretamente no fenótipo do indivíduo. Neste caso, o código genético é análogo a representação cromossomial.

A representação cromossomial, de acordo com Linden (2008), é a forma como o indivíduo traduz suas informações de maneira viável ao restante do algoritmo. Cada parte indivisível da representação cromossomial é chamado de um gen, em analogia aos genes dos cromossomos naturais, como na Figura 1. Qualquer gen, por sua vez, representa uma unidade de informação necessária para construção do indivíduo. Em outras palavras, gens são parâmetros de entrada para um algoritmo gerador.

Figura 1 - Representação cromossomial



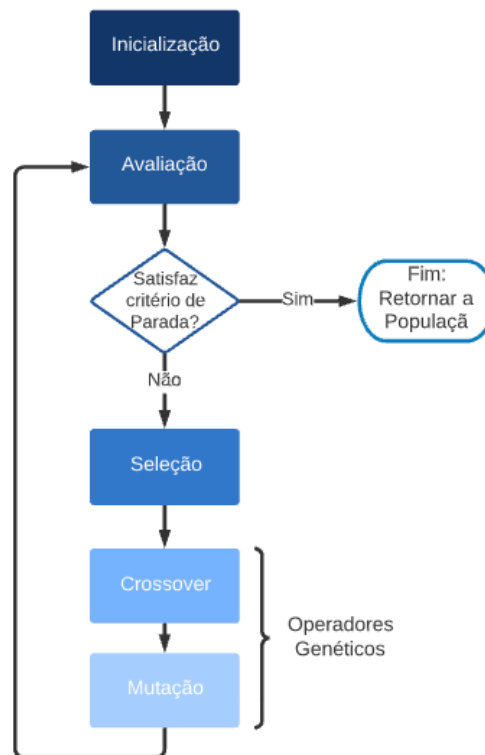
Fonte: Autor (2021)

O gen, como unidade de informação que será traduzida no fenótipo do indivíduo, pode ser qualquer dado variável da solução, como alguma dimensão, localização, material ou cor.

Um esquema de AG básico, Figura 2, é composto por 4 etapas: Inicialização, Avaliação, Seleção e Aplicação dos Operadores Genéticos. Cada uma dessas etapas tem paralelo com processos da evolução natural das espécies (SMITH, 2020; NAGY, 2017; LINDEN, 2008).



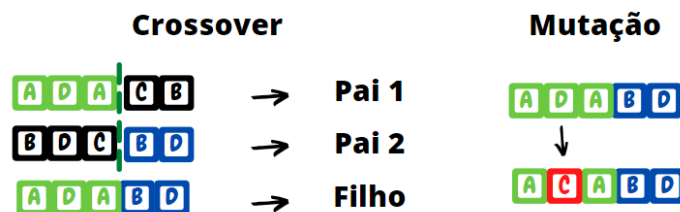
Figura 2 - Esquema básico do AG



Fonte: Autor (2021)

São dois os tipos de Operadores Genéticos, o crossover (ou cruzamento) e as mutações. O crossover, como na reprodução, combina os genes dos pais. Já a mutação, acontece após o crossover, onde alguns dos indivíduos recebem mudanças aleatórias nos seus genes como na Figura 3 (LINDEN, 2008).

Figura 3 - Crossover e Mutação



Fonte: Autor (2021)

### 2.3 Projeto computacional

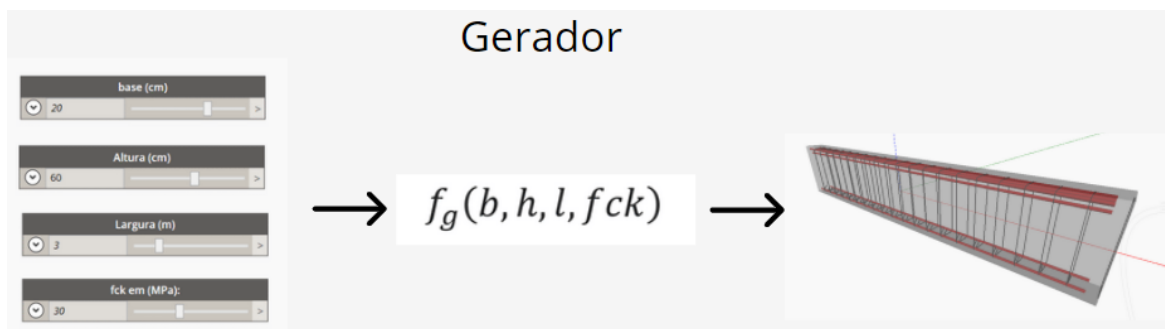
Smith (2020), define o projeto computacional como: “uma abordagem pela qual um projetista define uma série de instruções, regras e relacionamentos que identificam precisamente as etapas necessárias para alcançar um design proposto, seus dados ou geometria resultantes”. Na informática, este conjunto de instruções lógicas é chamado de algoritmo. O termo projeto computacional também pode ser

visto na literatura como projeto algorítmico (algorithmic design) ou projeto paramétrico (parametric design).

Em especial, no projeto generativo, os projetos computacionais apresentam 3 tipos principais de algoritmos: geradores, avaliadores e os solucionadores. Cada um destes tipos de algoritmos é definido por Smith (2020).

O algoritmo gerador é a lógica que cria potenciais soluções de projetos. Ele começa pelos parâmetros de entrada e a partir destes cria o projeto. Seus parâmetros de entrada podem ser: dimensões, posicionamento, quantidade, material ou as suas propriedades. Num exemplo de gerador de vigas, Figura 4, os parâmetros de entrada podem ser suas dimensões e resistência. Já a saída é o projeto da viga.

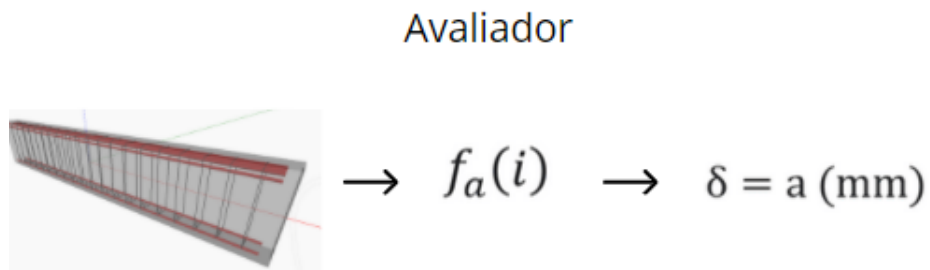
Figura 4 - Função de geração



Fonte: Autor (2021)

O algoritmo de avaliação questiona alguma propriedade em particular do projeto. Sua entrada é o projeto do algoritmo anterior, este projeto é avaliado em função dos objetivos e restrições do problema. Em suma, a função avaliação basicamente mensura o quão próximo a possível solução esta dos seus objetivos.

Figura 5 - Função de avaliação



Fonte: Autor (2021)

No exemplo das vigas, Figura 5, o algoritmo de avaliação pode classificá-las em função de seu volume, custo ou deslocamento. Destaca-se que a função de

avaliação deve necessariamente entregar o resultado numericamente, para que ela possa ser discriminada.

Por fim os algoritmos solucionadores que podem conter ambos, geradores ou avaliadores. Estes normalmente são executados várias vezes pelo solucionador. Suas entradas e saídas são específicas e dependem do método utilizado. Os métodos abordados aqui serão: Randomizar, Produto transversal, Semelhante e Otimizar (SMITH, 2020).

## **2.4 Espaço de projeto**

De acordo com Nagy, Villaggi e Benjamin (2018) o primeiro passo do processo de Projeto Generativo é a delimitação do espaço de projeto. Este, segundo eles, é o conjunto de possibilidades geradas pelo projeto computacional sujeitas a suas restrições.

Na maior parte dos casos o espaço de projeto é  $n$ -dimensional, onde  $n$  é o número de parâmetros de entrada do modelo. Por isso, aumentar o número de parâmetros aumenta consideravelmente o tamanho do espaço de projeto, podendo inviabilizar o processo de pesquisa (NAGY, 2017).

Dito isso, considerando que o papel do projetista também é delimitar o espaço de projeto, indaga-se, qual deve ser o tamanho do espaço de projeto? Não existe uma resposta objetiva para essa pergunta, no entanto Nagy (2017) define duas compensações que podem direcionar o projetista.

### **2.4.1 Viés versus Variância**

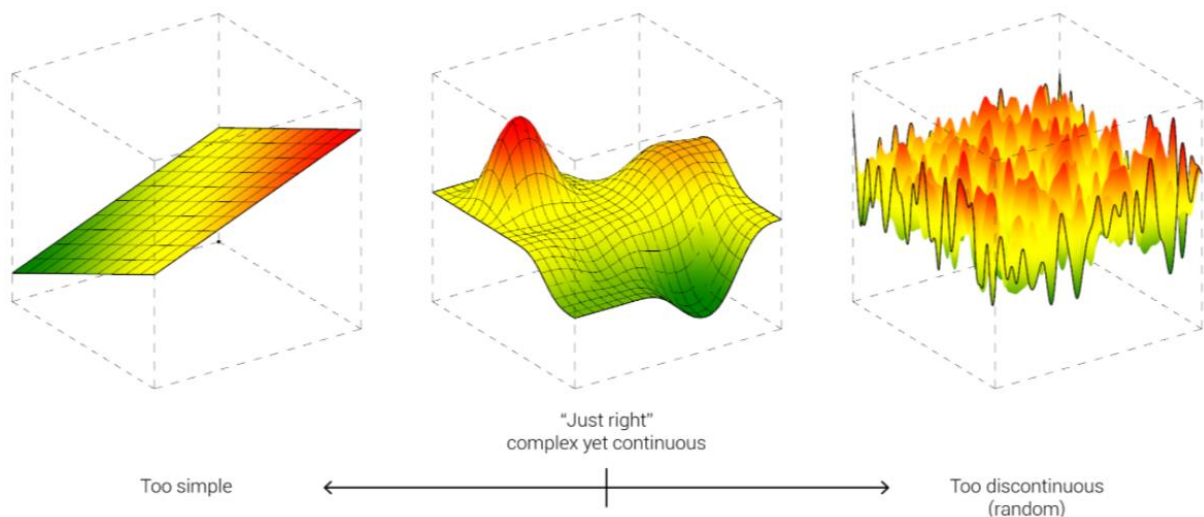
A comparação entre viés e variância se refere a flexibilidade relativa do modelo. Um espaço de projeto com alto viés é muito simples e não captura uma variação de soluções de projetos suficientes, em outras palavras ele é enviesado. Por outro lado, um espaço muito variável é muito flexível e captura um número de possibilidades maior que o necessário para resolver o problema (NAGY et al, 2017).

Cabe ao projetista buscar o equilíbrio entre essas duas partes. Sem limitar muito o espaço de projeto excluindo boas soluções, nem aumentando sua flexibilidade demais a ponto de inviabilizar a busca.

### 2.4.2 Complexidade versus Continuidade

Já complexidade e continuidade referem-se a tendência para que projetos adjacentes no espaço de projeto sejam semelhantes entre si (NAGY, 2017), como na Figura 6. A continuidade se refere a consistência interna do espaço de projeto. É necessário que o espaço de projeto tenha continuidade para que o algoritmo navegue pelas soluções fazendo previsões válidas. Já a complexidade está relacionada ao potencial do espaço de projeto de gerar soluções imprevisíveis e não-intuitivas. Um espaço de projeto pouco flexível não vale a pena o esforço computacional pois é possível determinar as melhores soluções com a intuição (NAGY et al., 2017).

Figura 6 - Relação complexidade vs continuidade.



Fonte: (NAGY, 2017)

Embora essas propriedades sejam muitas vezes conflitantes, o projetista do espaço de projeto tem o desafio de maximizar ambas. Desenvolver um espaço de projeto que seja contínuo suficiente para ser pesquisável e complexo o suficiente para permitir soluções inesperadas (NAGY et al., 2017).

## 2.5 Otimização multiobjetivo

Em geral, os problemas de otimização mais simples têm o objetivo único de maximizar ou minimizar algum parâmetro. Neste tipo de problema é fácil comparar o desempenho entre diferentes opções de projeto. Por isso, muitas vezes a sua solução é intuitiva.

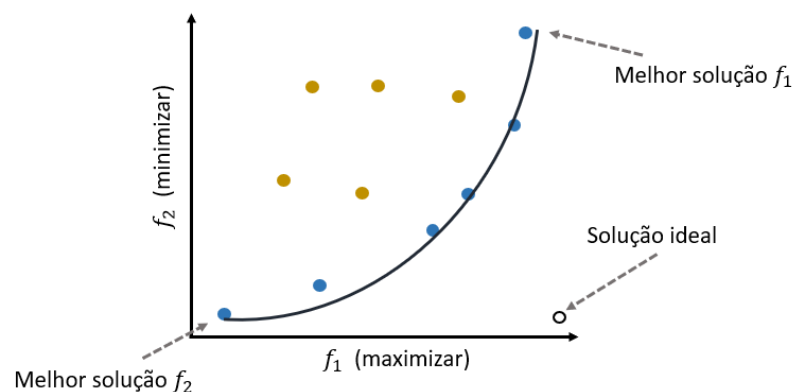
No entanto, os problemas de projeto na engenharia são mais complexos, apresentando mais de um objetivo. Estes objetivos, na maioria das vezes estão

relacionados entre si apresentando resultados conflitantes. Em outras palavras, melhorar um objetivo resulta na piora do outro.

Nestes casos, não existe um projeto único que performa bem para todos os objetivos, na verdade, trata-se de um conjunto de soluções chamada Frente Ótima de Pareto. Onde os projetos ao longo dessa fronteira podem ser considerados ótimos (VELDHUIZEN e LAMONT, 1998).

Observe o exemplo conceitual de Deb (2001), nele deseja-se maximizar a função  $f_1$  e minimizar a função  $f_2$ , considerando esses objetivos concorrentes entre si. Sendo assim, o gráfico de soluções seria como na Figura 7. Observe a formação da Fronteira de Pareto com as potenciais soluções do sistema, note que, não existe solução para o ponto ideal, no entanto as melhores soluções de cada um dos objetivos estão presentes na Fronteira de Pareto.

Figura 7 - Exemplo de Fronteira de Pareto



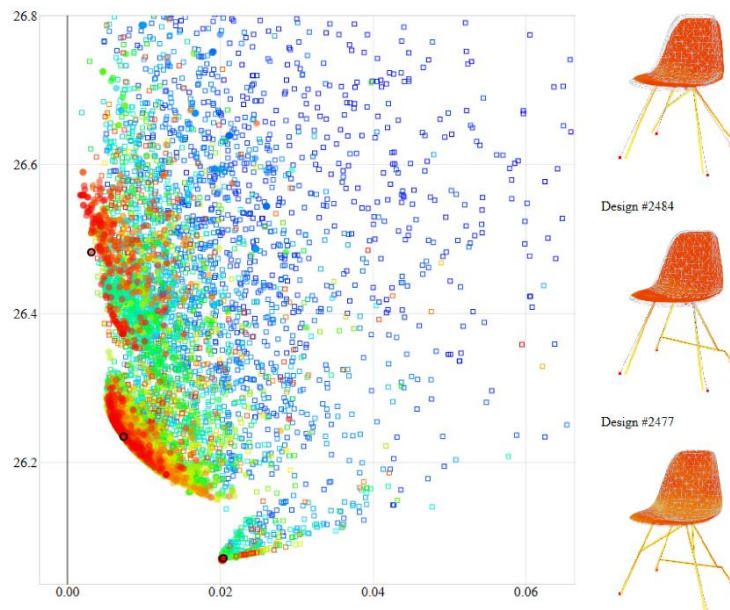
Fonte: Adaptado DEB (2001)

A Fronteira de Pareto, também chamada de Frente Ideal é formada pelos projetos dominantes. Projetos dominantes (em azul na Figura 7) são aqueles que não são superados em ambos os objetivos por nenhuma outra opção de projeto. O contrário também é verdadeiro, projetos não-dominantes (em amarelo na Figura 7) são os projetos superadas em todos os objetivos por alguma outra opção, portanto são excluídos da frente ideal (VELDHUIZEN e LAMONT, 1998).

No exemplo anterior, com dois objetivos, é fácil de identificar a frente ideal. Porém o conceito também é válido para múltiplos objetivos, nesses casos a fronteira é visualizada num espaço n-dimensional, onde n é o número de objetivos do problema.

Outro fator que distorce a frente ideal são as descontinuidades no espaço de projeto. Nesses casos, a fronteira apresenta formato descontínuo. Nagy (2017) exemplifica isso com o projeto de uma cadeira otimizada por dois objetivos concorrentes: minimizar o peso e a deformação, Figura 8. Nela, a descontinuidade ocorre nas regiões onde o espaço de projeto muda o número de barras das pernas.

Figura 8 - Frentes ideais descontínuas



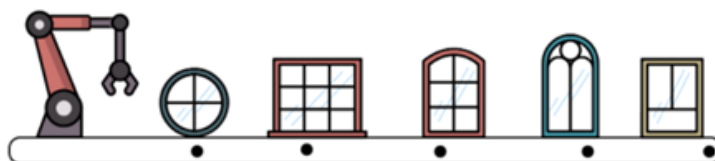
Fonte: (NAGY, 2017)

## 2.6 Estágios do projeto generativo

Para compreender melhor o processo de projeto generativo Smith (2020) dividi-o em 6 etapas: geração, análise, classificação, evolução, exploração e integração.

Na etapa de geração, é executado o algoritmo gerador criando as opções de projeto, Figura 9. Cabe ao projetista fornecer o algoritmo e especificar os parâmetros.

Figura 9 - Etapa de geração



Fonte: (SMITH, 2020)

Na etapa de análise, é executado o algoritmo de avaliação onde os projetos serão avaliados com base nos objetivos e nas restrições, Figura 10. Nesta etapa o projetista fornece o algoritmo avaliador.

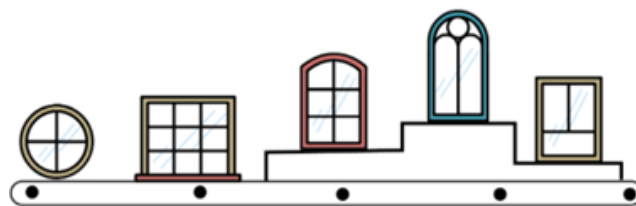
Figura 10 - Etapa de análise



Fonte: (SMITH, 2020)

Na etapa de classificação os projetos são ranqueados em função da sua avaliação e restrições, Figura 11. É papel do projetista escolher objetivos e delimitar restrições.

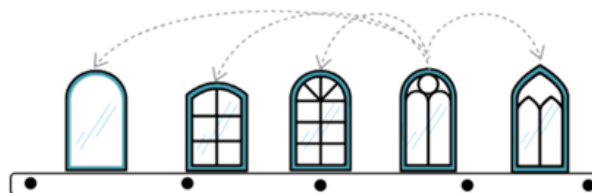
Figura 11 - Etapa de classificação



Fonte: (SMITH, 2020)

Na etapa de evolução são geradas novas opções de projeto, dessa vez combinando os projetos melhor classificados na última etapa, Figura 12.

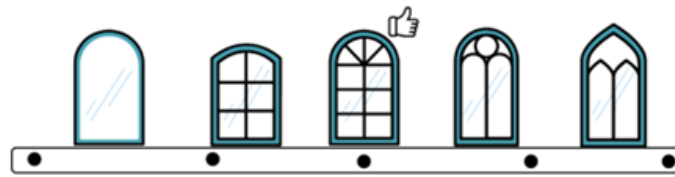
Figura 12 - Etapa de evolução



Fonte: (SMITH, 2020)

Na etapa de exploração é apresentado ao projetista as opções de projeto. O projetista explora a geometria e os resultados das opções, Figura 13.

Figura 13 - Etapa de exploração



Fonte: (SMITH, 2020)

Na última etapa, após a exploração, o projetista escolhe o projeto e o integra ao ambiente de trabalho, Figura 14.

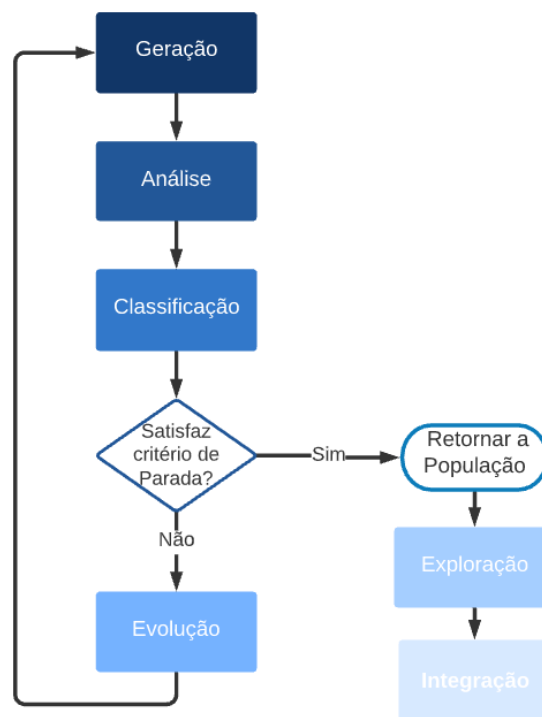
Figura 14 - Etapa de integração



Fonte: (SMITH, 2020)

A Figura 15 apresenta o fluxograma de estágios do projeto generativo. Sua estrutura é semelhante a estrutura do AG, porém com adição de algumas etapas.

Figura 15 - Fluxograma de estágios do projeto generativo



Fonte: Autor (2021)



## 2.7 Dimensionamento da armadura da viga

A norma técnica 6118 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2014) estabelece os requisitos básicos exigidos no projeto estrutural de concreto armado.

De acordo com Carvalho e Filho (2016) é conveniente sempre que possível utilizar fórmulas adimensionais para o dimensionamento de seções retangulares. Sendo assim, o cálculo da área de armadura ( $A_s$ ) é dado por:

$$A_s = \frac{M_d}{(KZ) \cdot d \cdot f_s} \quad (2.1)$$

Onde,

$$KZ = \left(1 - \frac{\lambda}{2} \cdot KX\right) \quad (2.2)$$

$$KX = \frac{1 - \sqrt{1 - \frac{2 \cdot KMD}{\alpha_c}}}{\lambda} \quad (2.3)$$

$$KMD = \frac{M_d}{b_w \cdot d^2 \cdot f_{cd}} \quad (2.4)$$

Para  $f_{ck} \leq 50 \text{ MPa}$ :

$$\lambda = 0,8 \text{ e } \alpha_c = 0,85 \quad (2.5)$$

E para  $f_{ck} > 50 \text{ MPa}$ :

$$\lambda = 0,8 - \frac{f_{ck} - 50}{400} \text{ e } \alpha_c = 0,85 \cdot \left(1 - \frac{f_{ck} - 50}{200}\right) \quad (2.6)$$

Onde,  $M_d$  é o momento de cálculo,  $d$  é a altura útil da seção,  $b_w$  é a largura da seção,  $f_s$  é a resistência de projeto de escoamento do aço,  $f_{cd}$  é a resistência de projeto à compressão do concreto,  $f_{ck}$  é a resistência à compressão característica. Os parâmetros  $\alpha_c$  e  $\lambda$  são respectivamente o parâmetro de redução da tensão e o coeficiente da posição da linha neutra. Por fim os coeficientes  $KMD$ ,  $KZ$ , e  $KX$  são respectivamente coeficientes adimensionais do momento fletor, braço de alavanca e posição da linha neutra.

É importante frisar que o coeficiente de posição da linha neutra deve ser menor que 0,45 para que o problema ocorra no domínio 2.

O item 17.3.5.1 da NBR 6118 (2014) estabelece a armadura mínima de tração ( $A_{s,min}$ ). Já o item 17.3.5.2.4 determina a armadura máxima de tração e compressão na viga ( $A_s + A_s'$ ). Dados por:

$$A_{s,min} = \rho_{min} \cdot A_c \quad (2.7)$$

$$(A_s + A_s') \leq 0,04 \cdot A_c \quad (2.8)$$

Onde,  $\rho_{min}$  é a taxa mínima de armadura e  $A_c$  a área de concreto da seção.

Ainda pela mesma norma as considerações de armadura concentrada, de não esmagamento da biela de concreto e altura útil mínima da seção, são verificadas pelas seguintes equações, respectivamente:

$$y_{cg} < 0,1 \cdot h \quad (2.9)$$

$$T_{sd} \leq T_{Rd2,I} \quad (2.10)$$

$$d_{min} \leq d \quad (2.11)$$

Onde,  $y_{cg}$  é a distância entre o centro de massa da armadura e a barra mais afastada da linha neutra,  $h$  é a altura da viga,  $T_{sd}$  é a tensão de cisalhamento solicitante de cálculo,  $T_{Rd2,I}$  é a tensão de cisalhamento relativa à ruína,  $d_{min}$  é a altura útil mínima e  $d$  é a altura da seção.

Para as verificações do estado limite de serviço são considerados os itens 17.3.3 e 17.3.2 que determinam o cálculo da abertura característica de fissura e a deformações excessivas imediata, dados por:

$$w_k = \text{menor entre} \left\{ \begin{array}{l} \frac{\phi_i}{12,5 \cdot \eta_i} \cdot \frac{\sigma_{si}}{E_{si}} \cdot \frac{3 \cdot \sigma_{si}}{f_{ct,m}} \\ \frac{\phi_i}{12,5 \cdot \eta_i} \cdot \frac{\sigma_{si}}{E_{si}} \cdot \left( \frac{4}{\rho_{ri}} + 45 \right) \end{array} \right. \quad (2.12)$$

$$a_{t,\infty} = a_{t,0} \cdot (1 + \alpha_f) \quad (2.13)$$

Onde  $w_k$  é o tamanho de abertura de fissuras característico e  $a_{t,\infty}$  é a flecha total no tempo infinito. No cálculo da fissuração  $\phi_i$  é o diâmetro da barra,  $\sigma_{si}$  é a tensão de tração no centro de gravidade da armadura considerada,  $\eta_i$  é o coeficiente de conformação superficial  $\eta_1$ ,  $E_{si}$  é o módulo de elasticidade do aço,  $f_{ct,m}$  é a resistência média do concreto a tração e  $\rho_i$  é a taxa de armadura passiva em relação à área da

região de envolvimento. No cálculo da flecha  $a_{t,0}$  é a flecha imediata para a combinação de ações permanentes e  $\alpha_f$  é o fator dado pelo item 17.3.2.1.2 da norma para as cargas de longa duração.

Os valores limites destes são dados nos itens 13.3 e 13.4.2 da norma.

No detalhamento da armadura longitudinal é calculado o deslocamento do diagrama de momentos fletores ( $a_l$ ) e o comprimento de ancoragem ( $l_b$ ) dados nos itens 17.4.2.2c e 9.4.2.4 da norma, pelas respectivas equações:

$$a_l = d \cdot \left[ \frac{V_{sd,max}}{2 \cdot (V_{sd,max} - V_c)} \right] \leq d \quad (2.14)$$

$$l_b = \frac{\phi}{4} \cdot \frac{f_{yd}}{f_{bd}} \quad (2.15)$$

Onde,  $d$  é a altura útil da seção,  $V_{sd,max}$  é a força cortante de cálculo na seção mais solicitada,  $V_c$  é a parcela da força cortante absorvida por mecanismos adicionais de treliça,  $f_{yd}$  é a resistência de projeto de escoamento do aço e  $f_{bd}$  é a resistência de aderência de cálculo entre a armadura e o concreto.

O cálculo da armadura transversal é feito pela eq. 2.16 de espaçamento dos estribos ( $s$ ) e obedecendo o limite máximo de espaçamento ( $s_{max}$ ) dado pela eq. 2.17.

$$s = \frac{A_{sw}}{b_w \cdot \rho_{sw,90}} \quad (2.16)$$

$$s_{max} = \frac{A_{sw}}{b_w \cdot \rho_{sw,90,min}} \quad (2.17)$$

Onde,  $A_{sw}$  é a área da seção transversal de uma barra da armadura de cisalhamento,  $b_w$  é a largura da seção,  $\rho_{sw,90}$  é a porcentagem volumétrica de armadura empregado em estribos verticais e  $\rho_{sw,90,min}$  é o mesmo coeficiente calculado para a armadura transversal mínima.

## 2.8 Integração com BIM

Conforme apontado por Suermann e Issa (2009), o BIM (Building Information Modeling) tem melhorado a eficiência na execução de projetos. De acordo com eles, gestores, projetistas e construtores apontam melhora em diversos

indicadores como os de qualidade, prazo e custo de execução. Provando-se uma excelente ferramenta na Construção Civil.

Além disso, sabe-se que, no início do desenvolvimento das ferramentas generativas, elas não atendiam os princípios básicos da modelagem da informação. Abrishami, Goulding, et al. (2014) propuseram a criação dos 'espaço de trabalho BIM generativo' com objetivo de potencializar a automação.

Argumentava-se que incorporar a modelagem da informação a ferramentas generativas, permitiria a troca de informações importantes do ambiente BIM para as entradas do sistema generativo, bem como os resultados otimizados poderiam ser inseridos diretamente no ambiente BIM.

Neste sentido, a fim de integrar ferramentas de projeto generativo ao BIM, Ferreira e Leitão (2015) conectam a ferramenta para programação de projeto generativo *Rosetta* ao *Revit*. Neste trabalho, destaca-se a facilidade de executar um projeto generativo em plataformas BIM em comparação a abordagem tradicional em CAD (Computer Aided Design).

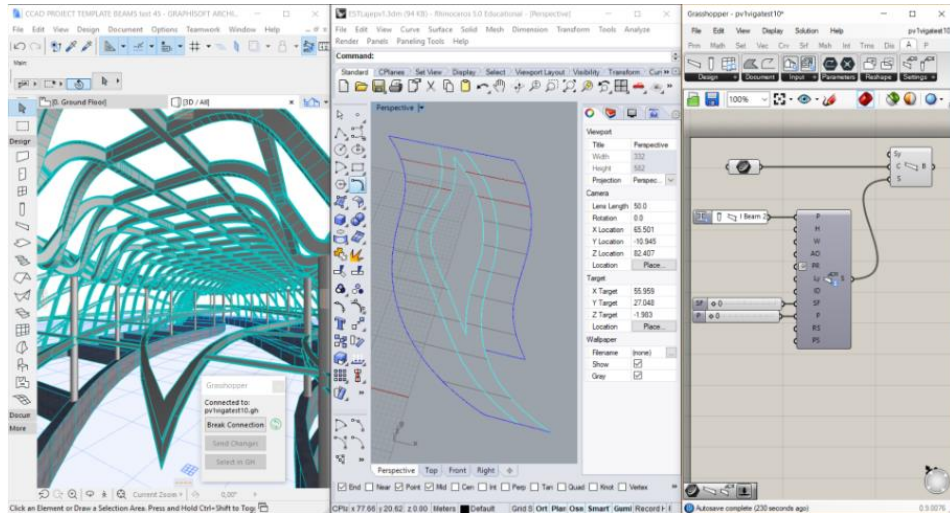
Santiago (2020) aplica projeto generativo integrado ao BIM para gerar soluções otimizadas de uma fachada sustentável considerando a orientação solar. Neste artigo o autor destacou: “A interconectividade em tempo real de BIM e softwares de modelagem algorítmica representa uma vantagem para economia de tempo nas decisões de projetos sustentáveis”.

O projeto ambientado no *Revit* com utilização de scripts generativos construídos no *Dynamo-for-Revit* de Ostrowska-Wawryniu (2019), teve grande ganho de produtividade considerando a complexidade do seu problema. Ele minimizou os resíduos na modelação de peças de madeira pré-fabricadas. O BIM permitiu ao algoritmo a automatização da geração de toda a geometria pré-fabricada e no modelo do *Revit* o algoritmo coletou informações de dimensão das peças.

O trabalho de Silva, Silva e Lacroix (2019) foi além, com a integração de três aplicativos: *FormZ* (Autodesk), *Rhinoceros-Grasshopper* (Robert McNeel & Associates) e *ArchiCAD* (Graphisoft). O processo importa revestimento e eixos estruturais de telhado do *FormZ* para o *Rhinoceros* estes são mais bem desenvolvidos

no *Rhino-Grasshopper-ArchiCAD Live Connection* por fim o *ArchiCAD* reconhece estes elementos. A Figura 16 ilustra esse processo.

Figura 16 - Estrutura otimizada por projeto generativo no *Rhino-Grasshopper to ArchiCAD*



Fonte: (SILVA, SILVA e LACROIX, 2019)

### 2.8.1 *Generative Design for Revit*

*Generative Design for Revit* é um novo recurso disponível na versão do *Revit 2021*. Este recurso facilita a execução de projetos generativos no ambiente do *Revit*.

Smith (2020) destaca “O *Generative Design for Revit* também permite que alguém que não tenha experiência com o *Dynamo* use fluxos de trabalho de projeto generativo, uma vez que executa scripts já criados diretamente da interface do *Revit*.”

*Generative Design for Revit* dispõe de 4 solucionadores para exploração de soluções de projeto são eles: Randomizar, Produto transversal, Semelhante e Otimizar (SMITH, 2020).

O solucionador Randomizar atribui valores aleatórios aos parâmetros de entrada formando uma quantidade pré-definida de projetos aleatórios. Sua entrada é apenas a quantidade de projetos desejados. Esse método é útil em situações em que o projetista tem poucas informações e quer apenas explorar opções diversificadas.

Já o Produto transversal combina todas as possibilidades de projeto. Esse método é eficiente em problemas num pequeno espaço de projeto. Quanto maior o espaço de projeto, mais tempo levará para processar tudo, inviabilizando o método.

O solucionador Semelhante, cria opções de projeto com poucas variações em relação a um projeto pré-definido. Neste método é importante especificar os valores de entrada do projeto de referência. Esse método é eficiente para explorar variações de um projeto que o projetista já gosta.

O método de otimizar é basicamente um AG, por isso, contém tanto o algoritmo gerador como avaliador. Como na lógica do AG o gerador cria potenciais soluções, as quais, em cada iteração, usam o conhecimento adquirido com a população anterior para gerar uma nova população com melhores resultados. O Algoritmo Genético presente no *Generative Design* é o NSGA-II de DEB, PRATAP, *et al.* (2002). Neste método é importante ter como entrada os objetivos que serão otimizados.

### 2.8.2 *Dynamo for Revit*

O *Dynamo for Revit* é um aplicativo de programação visual de código aberto, conectado ao *Revit*. Nele, através de sua interface gráfica o projetista ou programador conecta 'nós' de comando, criando a lógica de programação. Os nós são os objetos que executam tarefas específicas. A conexão entre eles é feita pelos 'fios', eles determinam a relação entre nós criando um fluxo que usualmente ocorre da esquerda para direita. A lógica de programação visual expressa pelos nós conectados por fios é chamada de gráfico.

Assim como o *Generative Design For Revit* a principal vantagem do *Dynamo for Revit* é que um profissional sem conhecimentos profundos de programação pode modelar um problema.

Vale comentar três nós do *Dynamo* importantes para o projeto computacional: nó de controle deslizante, personalizado e o de script em *Python*.

Os gens do projeto generativo são necessariamente nós de controle deslizante. Estes nós são configurados com um valor mínimo, um valor máximo e o passo entre eles. O exemplo da Figura 17 expressa os valores de resistência característica do concreto, contemplando as resistências de: 20, 25, 30, 35, 40 e 45 *Mpa*.



### 2.8.3 Nós *Data.Remember* e *Data.Gate*

Como visto, a estrutura computacional desenvolvida é composta de códigos de importação e exportação de dados no *Revit*. No entanto, esse fluxo de dados entre *Dynamo* e *Revit* exige um alto esforço computacional quando sujeito ao processo iterativo do projeto generativo (SMITH, 2020).

Uma poderosa ferramenta para controle desse fluxo são os nós *Data.Remember* e *Data.Gate* do pacote *GenerativeDesign* do *Dynamo*.

Nós *Data.Remember* armazenam dados em cache, para que o algoritmo que gera esses dados não precise ser executado novamente a cada iteração.

Já os nós *Data.Gate* funcionam interrompendo o fluxo de dados durante as iterações. No entanto, quando, na fase de exploração, o projetista seleciona a opção 'Criar elemento no Revit' ele libera o fluxo, permitindo a execução das estruturas posteriores.

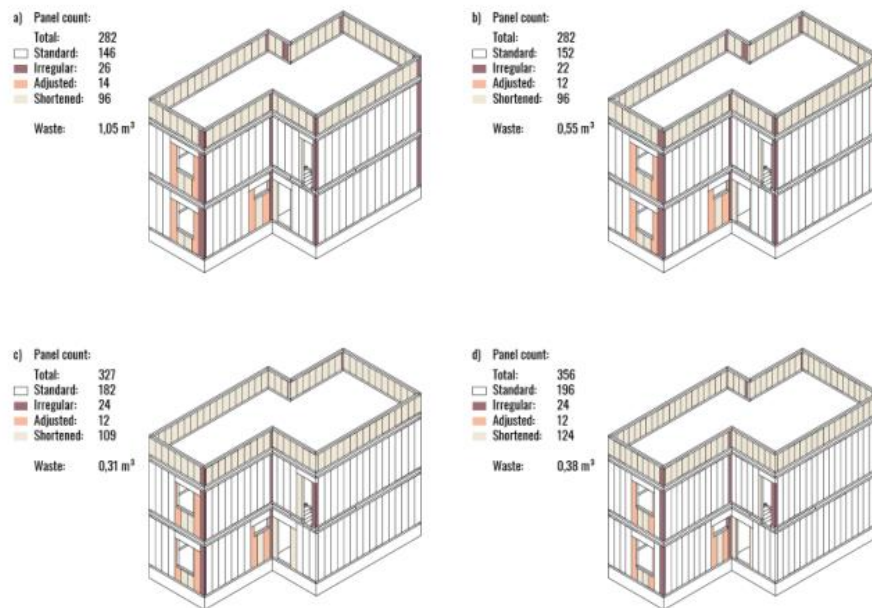
## 2.9 Exemplos de projeto generativo

### 2.9.1 *Redução de resíduos na modelação de casa de madeira*

Um bom exemplo é do já citado Ostrowska-Wawryniu (2019), a sua abordagem generativa considera o projeto de modulação de casas pré-fabricadas de madeira adequadas ao estilo faça-você-mesmo. O projeto de modulação é composto por painéis de madeira, de forma que alguns deles precisam ser ajustados in loco deixando resíduos. A otimização tem o objetivo de minimizar os restos, maximizar a largura dos painéis e aumentar o número de painéis repetidos.



Figura 19 - Exemplo de modelação da casa de madeira



Fonte (OSTROWSKA-WAWRYNIUK, 2019)

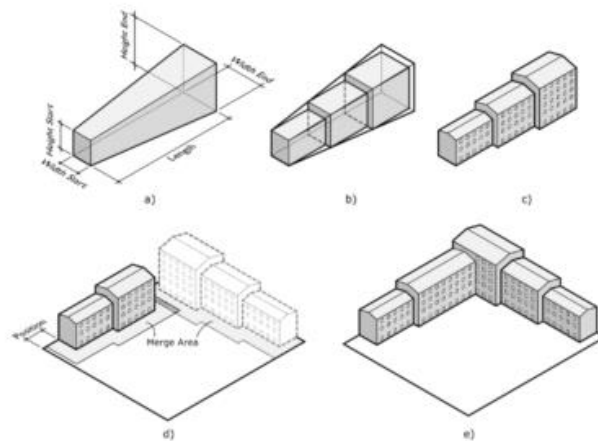
A abordagem foi aplicada a um projeto de casa unifamiliar de dois pavimentos, Figura 19. Então, os resultados do projeto generativo foram comparados com a modulação manual tradicionalmente usada. Disto, observou-se que a abordagem genética reduziu o desperdício em mais de 50%.

### 2.9.2 *Estudo de caso de bloco residencial*

Outro exemplo que explora o projeto generativo no contexto arquitetônico é o de Mukkavaara e Sandberg (2020). O estudo explora aspectos de sustentabilidade no projeto de blocos residenciais em Kiruna, no norte da Suécia.

O projeto generativo explora o formato geométrico dos edifícios do bloco residencial. De forma que os edifícios têm comprimento limitado e são ajustados a prismas trapezoidais, como na Figura 20. O espaço de projeto é restringido pelas dimensões do terreno e os regulamentos da região. As dimensões dos prismas e as bordas que contém os edifícios são variáveis do projeto.

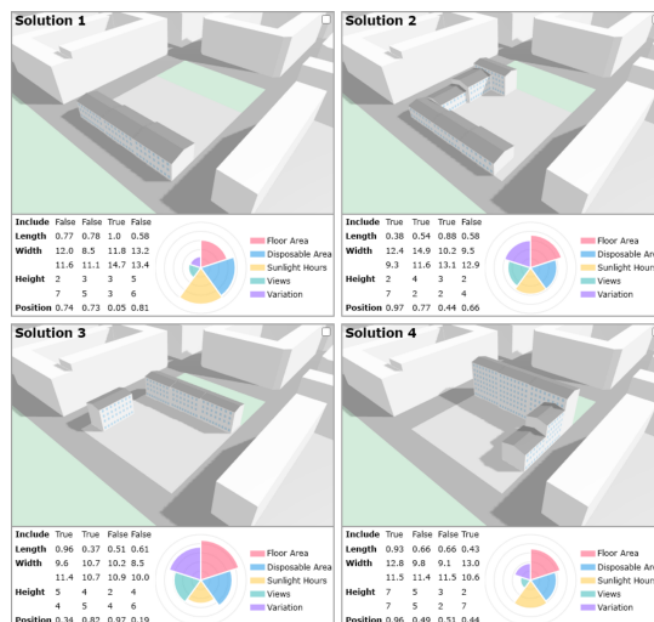
Figura 20 - Definição do espaço de projeto



Fonte: (Mukkavaara e Sandberg, 2020)

As medidas de avaliação utilizadas foram horas de luz solar, vista, área habitável, área descartável e variação. As opções de projetos foram apresentadas junto com seus modelos 3D, resultado das métricas, gráficos e tabelas, como na Figura 21. Portanto, destaca-se como a análise generativa contribui para uma tomada de decisão mais informada e com mais opções.

Figura 21 - Opções de projeto



Fonte: (MUKKAVAARA e SANDBERG, 2020)

### 2.9.3 Otimização de desempenho térmico

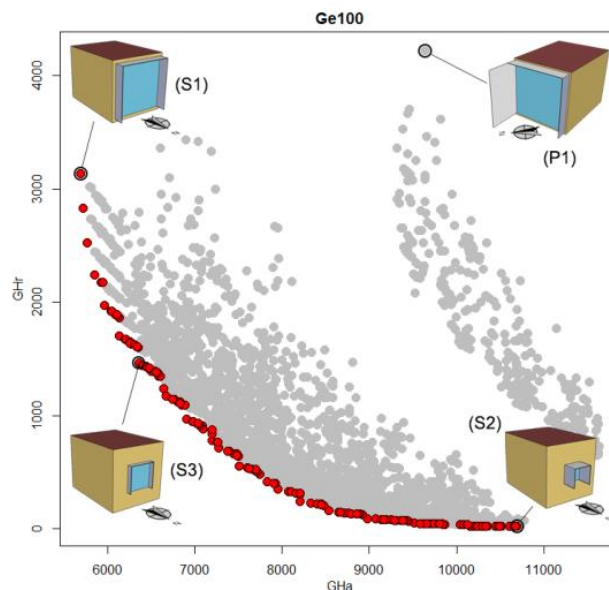
No exemplo de Linczuk e Bastos (2020), é feita otimização orientada ao desempenho térmico e consumo de energia de uma edificação de baixo consumo na

Região Sul do Brasil. A pesquisa foi aplicada a um modelo de edifício hipotético de zona única, sua fachada contém uma janela centrada e anteparos para sombreamento.

As variáveis paramétricas do problema são: a orientação solar, as propriedades dos materiais, as dimensões da janela e dos anteparos. A otimização busca minimizar os indicadores de graus-hora para aquecimento (GHa) e de resfriamento (GHr).

Por fim, em 123 minutos o projeto generativo construiu a Fronteira de Pareto, Figura 22, avaliou 3023 em 100 gerações. Uma observação importante é que o algoritmo teria que explorar cerca de 180.000 opções de projeto se tivesse de cobrir todo o espaço de projeto.

Figura 22 - Fronteira de Pareto



Fonte: (LINCZUK e BASTOS, 2020)

### 3 METODOLOGIA

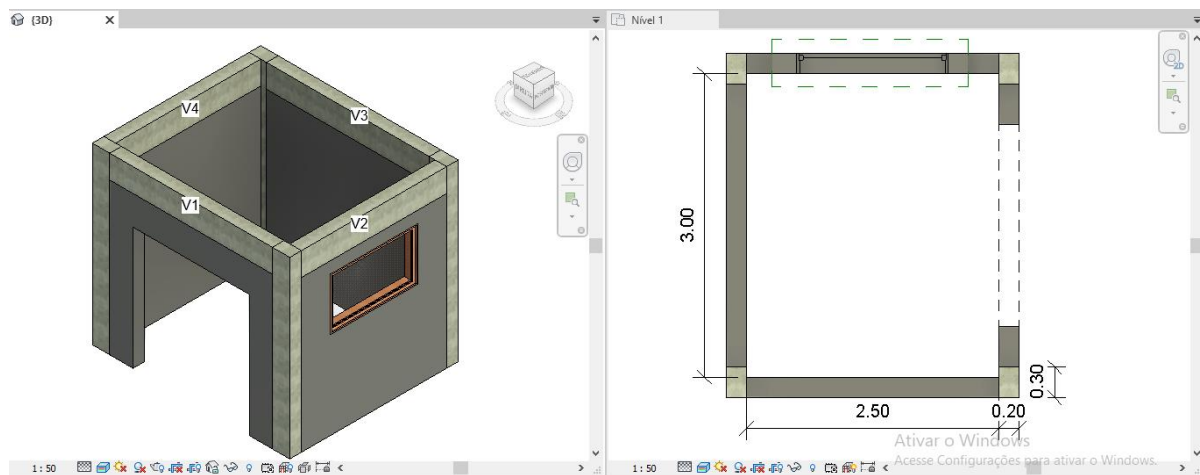
Neste capítulo é apresentado o exemplo de aplicação e os métodos de estruturação do projeto computacional. O modelo se propõe a aplicar os solucionadores do *Generative Design for Revit* na exploração de soluções de projeto para uma viga biapoiada de concreto armado.

### 3.1 Exemplo de aplicação

Para aplicação do solucionador genético foi idealizado um projeto arquitetônico conceitual no *Revit 2021*. Este projeto tem o papel de simular as condições e restrições das vigas em um projeto real.

O projeto consiste em um ambiente de  $3,0 \times 2,5 \text{ m}$  de dimensões internas e  $3,0 \text{ m}$  de altura, como na Figura 23. Os elementos arquitetônicos que vão restringir o modelo são os pilares de  $20 \times 30 \text{ cm}$ , as paredes de  $20 \text{ cm}$  de espessura, uma porta de  $2,1 \text{ m}$  de altura e uma janela de  $90 \text{ cm}$  de altura com peitoril de  $1,5 \text{ m}$ . Esses elementos limitam as dimensões de base e altura das vigas exploradas.

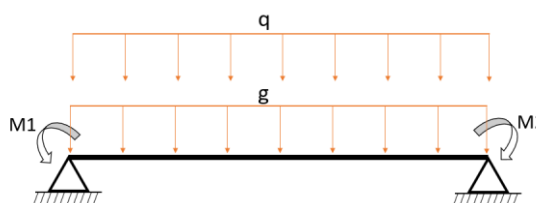
Figura 23 - Projeto de aplicação



Fonte: Autor (2021)

As vigas foram nomeadas de V1, V2, V3 e V4 conforme a Figura 23. O carregamento das vigas é extraído de uma planilha de *Excel*. As informações de carregamento obtidas no *Excel* são: a carga permanente,  $g$  em  $kN/m$ ; a carga variável,  $q$  em  $kN/m$ ; os momentos negativos nas extremidades da viga,  $M1$  e  $M2$  em  $kNm$ . A Figura 24 mostra o modelo de carregamento considerado.

Figura 24 - Esquema de Carregamento das vigas



Fonte: Autor (2021)

Os carregamentos e solicitações referentes as vigas exploradas podem ser verificadas na Tabela 1, a seguir:

Tabela 1 - Carregamentos e Solicitações

Vigas	V1	V2
g (kN/m)	30	40
q (kN/m)	20	40
M1 (kNm)	50	80
M2 (kNm)	100	80

Fonte: Autor (2021)

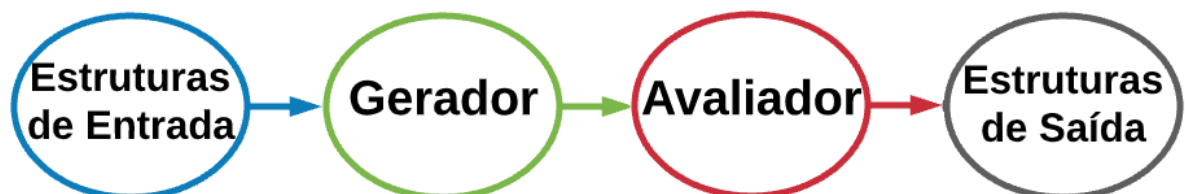
Construiu-se uma estrutura paramétrica para que, ao ser executada, o projetista faça a seleção da viga que será explorada, suas restrições geométricas (limitações de altura de largura) e a escolha do solucionador. Após o processamento são apresentadas pelo algoritmo as opções de projeto e depois da exploração das opções o algoritmo integra a opção escolhida ao restante do projeto.

### 3.2 Projeto Computacional

O projeto computacional é a lógica que direciona o algoritmo de geração e avaliação. Neste projeto ele foi desenvolvido no *Dynamo for Revit*.

O gráfico do projeto computacional está dividido em quatro algoritmos principais como na Figura 25, são eles: a estrutura de entrada que recebe os dados do *Revit* e *Excel* que alimenta o gerador; o gerador e avaliador cuja funções já foram discutidas no item 2.3; por fim, as saídas que exportam a opção de projeto de volta para o *Revit*.

Figura 25 - Gráfico da estrutura paramétrica



Fonte: Autor (2021)

#### 3.2.1 Estruturas de entrada

As estruturas de entrada carregam os dados necessários para o algoritmo gerador do projeto, importando-os do *Revit*, *Excel* e compondo os gens do projeto generativo.

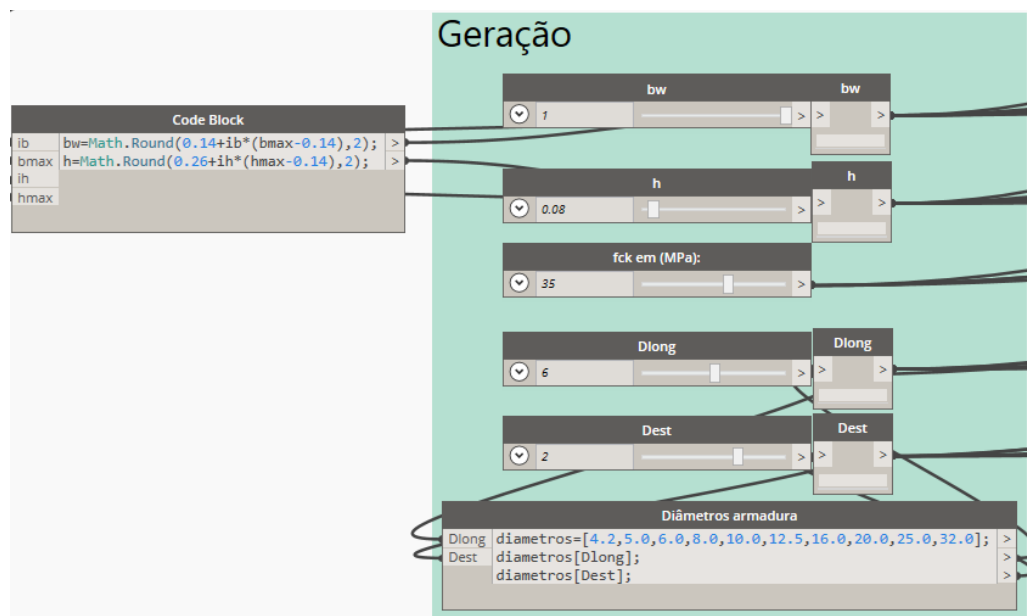
As estruturas de entrada foram divididas em quatro partes: informações da viga, restrições, importação do *Excel* e as variáveis de projeto.

A entrada de importação da viga carrega o nome da viga, seu comprimento e localização. As restrições importadas são a altura e largura. O primeiro está condicionado ao limite superior da porta e janela. A largura é limitada pela espessura da parede ou pilar.

Neste estudo os gens são: a base, altura, resistência característica do concreto, diâmetro da armadura longitudinal e transversal. Vale destacar que os gens são necessariamente nós de controle deslizante.

Os nós de controle deslizantes da base e da altura variam de 14 e 26 cm às suas respectivas dimensões máximas dadas pelas restrições de projeto arquitetônico. O  $f_{ck}$  varia entre as resistências de: 20, 25, 30, 35, 40 e 45 Mpa. O diâmetro da armadura longitudinal varia pelas opções comerciais de: 8.0, 10.0, 12.5, 16.0, 20.0, 25.0, 32.0 mm. O diâmetro da armadura transversal varia pelas opções comerciais de: 4.2, 5.0, 6.0, 8.0 mm.

Figura 26 - Representação cromossomial



Fonte: Autor (2021)

Observe que, neste ponto, o espaço de projeto está definido, uma vez que estão definidos os gens. A Figura 26 apresenta a representação cromossomial do projeto composta pelos gens.

### 3.2.2 Algoritmo Gerador

O algoritmo gerador cria as opções projeto, neste caso, o cálculo da armadura e a sua representação geométrica. O cálculo de armadura por ser uma lógica mais densa e complexa foi programado em nós de scripts *Python*.

O cálculo da armadura foi dividido em três nós principais de scripts *Python*: o nó de cálculo da armadura da seção transversal, comprimentos de ancoragem e armadura transversal. Estes cálculos são feitos de acordo com a NBR 6118 (2014) citada no item 2.7.

Para organizar o gráfico, estes nós de script foram incorporados em um nó personalizado, cujas entradas são os dados recebidos pelas estruturas de entrada, geometria e propriedades da viga e suas saídas são as informações geométricas da armadura. Desse modo, o nó personalizado retorna as coordenadas dos pontos inicial e final das barras de armadura longitudinal e a geometria da armadura transversal.

### 3.2.3 Algoritmo Avaliador

O algoritmo avaliador mensura os objetivos e as restrições entregando o resultado numericamente. O resultado é encaminhado para um nó *watch*.

Definiu-se os seguintes objetivos do problema: a área da seção transversal de concreto,  $A_c$  em  $m^2$ , a área de forma da viga em  $m^2$ , o peso próprio da viga,  $P_p$  em  $kN$ ; o comprimento da armadura longitudinal e transversal em  $m$ ; e o custo da viga em reais.

Para avaliação do custo da viga foi consultado o sistema de Orçamento de Obras de Sergipe (ORSE). Para o concreto, considerou-se o valor do metro cúbico de concreto usinado bombeável com brita 0 e 1, Tabela 2.

Tabela 2 - Custo concreto usinado

Código	Descrição	Unid.	R\$
00636/ORSE	Concreto usinado bombeável b0-b1 fck = 21MPa	m <sup>3</sup>	319,56
00637/ORSE	Concreto usinado bombeável b0-b1 fck = 25MPa	m <sup>3</sup>	328,57
02690/ORSE	Concreto usinado bombeável b0-b1 fck = 30MPa	m <sup>3</sup>	339,52
00638/ORSE	Concreto usinado bombeável b0-b1 fck = 35MPa	m <sup>3</sup>	350,47
02689/ORSE	Concreto usinado bombeável b0-b1 fck = 40MPa	m <sup>3</sup>	365,60
08438/ORSE	Concreto usinado bombeável b0-b1 fck = 45MPa	m <sup>3</sup>	500,35

Fonte: Autor (2021)

A armadura longitudinal de aço CA-50 com valores de custo por quilograma pode ser verificada na Tabela 3.

Tabela 3 - Custo da armadura longitudinal

<b>Código</b>	<b>Descrição</b>	<b>Unid.</b>	<b>R\$</b>
04795/ORSE	Aço CA-50 Ø 8 mm (5/16")	kg	9,33
04796/ORSE	Aço CA-50 Ø 10 mm (3/8")	kg	8,80
04797/ORSE	Aço CA-50 Ø 12,5 mm (1/2")	Kg	7,62
04798/ORSE	Aço CA-50 Ø 16 mm (5/8")	kg	7,62
04800/ORSE	Aço CA-50 Ø 20 mm (3/4")	kg	8,78
04799/ORSE	Aço CA-50 Ø 25 mm (1")	kg	8,78
43057/SINAPI	Aço CA-50 Ø 32 mm, vergalhão	kg	9,65

Fonte: Autor (2021)

Por fim, o custo por quilograma da armadura transversal de aço CA-60, Tabela 4.

Tabela 4 - Custo armadura transversal

<b>Código</b>	<b>Descrição</b>	<b>Unid.</b>	<b>R\$</b>
04801/ORSE	Aço CA-60 Ø 4,2 mm	kg	8,32
04802/ORSE	Aço CA-60 Ø 5 mm	kg	8,32
04803/ORSE	Aço CA-60 Ø 6 mm	Kg	8,32
04805/ORSE	Aço CA-60 Ø 8 mm	kg	7,25

Fonte: Autor (2021)

Como o ORSE fornece o custo da barra por quilograma, esse valor é convertido para custo por metro. Para tanto, a NBR 7480 de aço destinado a estruturas de concreto armado (2007) especifica a massa por metro de cada bitola como na Tabela 5.

Tabela 5 - Peso do aço

<b>Bitola (mm)</b>	<b>Peso aproximado (kg/m)</b>
4.2	0,109
5.0	0,154



6.0	0,222
8.0	0,395
10.0	0,617
12.5	0,963
16.0	1,578
20.0	2,466
25.0	3,853
32.0	6,313

Fonte: Autor (2021)

Define-se também as verificações da NBR 6118 (2014) apresentadas do item 2.7 deste trabalho como restrições de projeto, são elas: verificações do estado limite de serviço, de consideração de armadura concentrada, de esmagamento da biela de concreto, altura útil mínima, altura mínima da linha neutra e área de aço máxima e mínima.

Para as verificações do estado limite de serviço é calculado: a abertura máxima de fissuras características, a deformação excessiva imediata e deferida com o tempo, respectivamente  $w_k$ ,  $a$  e  $a_t$  em metros.

Outra verificação considerada é a de altura mínima da linha neutra limitada em  $KX < 0,45$ . O coeficiente é calculado nos scripts de dimensionamento e encaminhado como saída de projeto.

As demais verificações também foram calculadas no código *Python* do algoritmo gerador. Elas foram convertidas para taxas, a fim de facilitar a verificação das opções de projeto no gráfico final.

As verificações de área de aço máxima e mínima da seção, são verificadas pelo cálculo da taxa de aço da seção  $Tx_{AS}$ , como na Eq. 3.1:

$$Tx_{AS} = \frac{A_{sreal} - A_{smin}}{A_{smax} - A_{smin}} \quad (3.1)$$

Onde  $A_{sreal}$ ,  $A_{smin}$  e  $A_{smax}$ , são respectivamente a área de aço real, mínima e máxima da seção. Então, a armação é verificada para valores de taxa de aço dentro do intervalo entre 0 e 1.

As verificações das eqs 2.9, 2.10 e 2.11 também são convertidas para taxas, essas verificações ficam como nas Eqs 3.2, 3.3 e 3.4:

$$\frac{y_{cg}}{0,1 \cdot h} < 1 \quad (3.2)$$

$$\frac{T_{sd}}{T_{Rd2,l}} \leq 1 \quad (3.3)$$

$$\frac{d_{min}}{d_{real}} \leq 1 \quad (3.4)$$

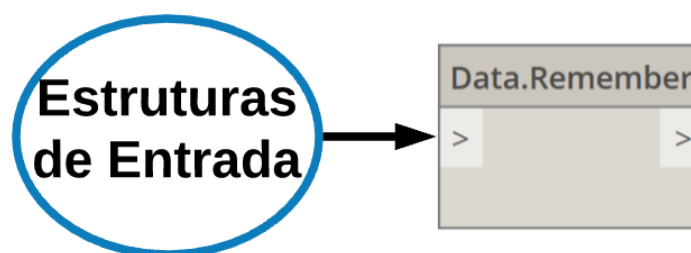
Algumas dessas verificações se repetem por três vezes, pois, no cálculo da armadura são calculadas três seções, uma de momento positivo e duas de momento negativo nas extremidades da viga.

#### 3.2.4 **Saídas e Interoperabilidade com o Revit**

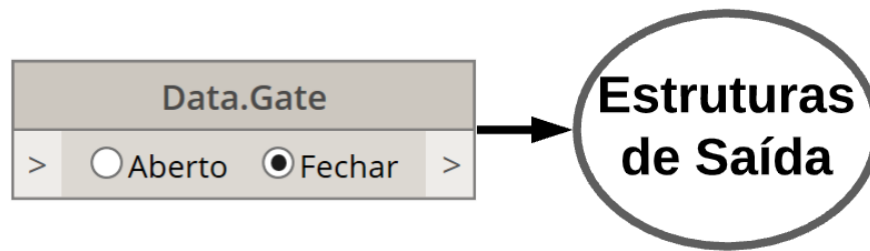
A estrutura de saída é composta por algoritmos que retornam para o *Revit* a opção de viga selecionada na etapa de exploração. Onde, o algoritmo, insere a viga no projeto e configura seus parâmetros de tipo e de instancia. Como parâmetros de tipo são configurados a base, altura e custo. Como parâmetro de instancia os cobrimentos da peça.

A interoperabilidade com o *Revit* fica por conta do pacote *GenerativeDesign*. No fim das estruturas de dados importados são colocados nós *Data.Remember*, como na Figura 27. Já os nós *Data.Gate* são usados antes das estruturas que exportam a opção de projeto para o *Revit*, como na Figura 28. Ver 2.8.3.

Figura 27 - Aplicação do nó Data.Remember



Fonte: Autor (2021)

Figura 28 - Aplicação do nó *Data.Gate*

Fonte: Autor (2021)

### 3.3 Solucionador

O projeto generativo é executado pelo *Generative Design for Revit*. A primeira parte dos resultados apresenta o funcionamento da estrutura para geração de opções de projeto da viga V1 (Figura 23). A viga V1, é explorada pelo método de otimização e comparados os resultados para otimização de um, dois e três objetivos.

Na otimização de objetivo único é minimizado o custo da viga. Na de duplo objetivo são minimizados o custo e a deformação excessiva imediata e deferida com o tempo. Na otimização de objetivo triplo é adicionada a minimização do comprimento da armadura.

Neste projeto para o tamanho da população de cada geração e o número de gerações do estudo, inicialmente foram usadas as configurações padrão do *Generative Design*, Figura 29. Em especial o estudo de duplo objetivo foi repetido com 20 e 40 gerações.

Figura 29 - Configurações da geração

Configurações da geração	
Tamanho da população	20
Gerações	10
Propagar	1

Fonte: Autor (2021)

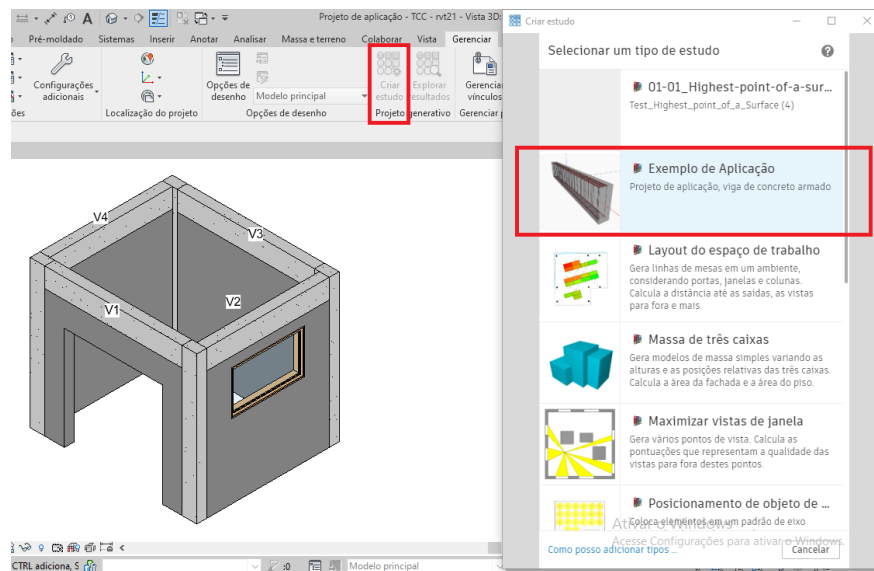
Na segunda parte, são aplicados os demais solucionadores genéticos a viga V2 e discutidas as distribuições das soluções no espaço de projeto de cada solucionador.

## 4 ANÁLISE DOS RESULTADOS

### 4.1 Funcionamento do modelo

O *Revit 2021* vem com o *Generative Design* instalado, facilitando a execução dos estudos do projeto generativo, Figura 30.

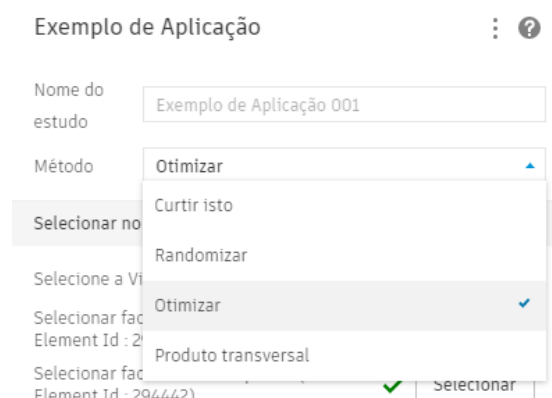
Figura 30 - Criar estudo



Fonte: Autor (2021)

As opções de solucionador do *Generative Design* apresentadas no item 2.8.1 aparecem na Figura 31. O Solucionador "Curtir Isso" foi traduzido erroneamente do termo original que é "*Like This*", por isso neste trabalho optou-se por uma tradução diferente. Neste exemplo é utilizado método de otimizar.

Figura 31 – Opções de solucionador



Fonte: Autor (2021)

Os objetivos e as restrições de projeto descritos no item 3.2.3 são observados na Figura 32, o quadro de seleção dos objetivos.

Figura 32 - Metas do estudo com dois objetivos

Objetivo	Minimizar	Maximizar
<input type="checkbox"/> wt	<input checked="" type="radio"/>	<input type="radio"/>
<input type="checkbox"/> a	<input checked="" type="radio"/>	<input type="radio"/>
<input checked="" type="checkbox"/> at	<input checked="" type="radio"/>	<input type="radio"/>
<input type="checkbox"/> Ac (m <sup>2</sup> )	<input checked="" type="radio"/>	<input type="radio"/>
<input type="checkbox"/> Comp. arm. transv (m)	<input checked="" type="radio"/>	<input type="radio"/>
<input type="checkbox"/> Comp. arm. long. (m)	<input checked="" type="radio"/>	<input type="radio"/>
<input type="checkbox"/> Área de forma (m <sup>2</sup> )	<input checked="" type="radio"/>	<input type="radio"/>
<input type="checkbox"/> Pp (kN)	<input checked="" type="radio"/>	<input type="radio"/>
<input checked="" type="checkbox"/> Custo (R\$)	<input checked="" type="radio"/>	<input type="radio"/>
<input type="checkbox"/> Taxa As Neg1	<input checked="" type="radio"/>	<input type="radio"/>
<input type="checkbox"/> Taxa As pos	<input checked="" type="radio"/>	<input type="radio"/>

Fonte: Autor (2021)

A definição das restrições segue os limites apresentados no item 3.2.3. Observe que, as restrições podem ser aplicadas antes do processamento como na Figura 33 ou após o processamento com a filtragem dos resultados.

Se aplicados antes do processamento o NSGA-II vai desencorajar a exploração de regiões promissoras do espaço de projeto que não atenderem as restrições, já filtrar as opções após o processamento permite a exploração do espaço de projeto sem restrições. Observou-se que o modelo apresentou poucas opções de projeto que não atendem restrições, por isso, optou-se por aplicar as condições de restrições com os filtros após o processamento.

Figura 33 - Definição das restrições

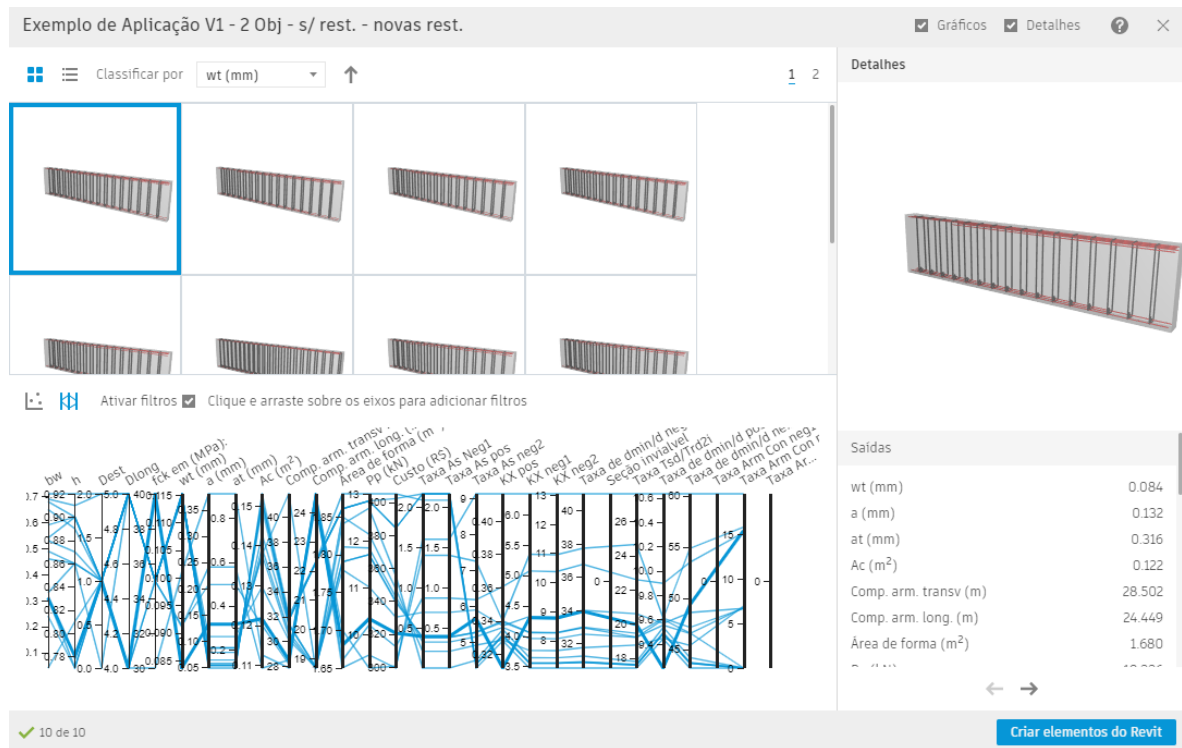
Definir restrições ▲

<input checked="" type="checkbox"/> wt	Mín.	<input type="text"/>	Máx.	<input type="text" value="0.003"/>
<input type="checkbox"/> a	Mín.	<input type="text"/>	Máx.	<input type="text"/>
<input type="checkbox"/> at	Mín.	<input type="text"/>	Máx.	<input type="text"/>
<input type="checkbox"/> Ac (m <sup>2</sup> )	Mín.	<input type="text"/>	Máx.	<input type="text"/>
<input type="checkbox"/> Comp. arm. transv (m)	Mín.	<input type="text"/>	Máx.	<input type="text"/>
<input type="checkbox"/> Comp. arm. long. (m)	Mín.	<input type="text"/>	Máx.	<input type="text"/>
<input type="checkbox"/> Área de forma (m <sup>2</sup> )	Mín.	<input type="text"/>	Máx.	<input type="text"/>
<input type="checkbox"/> Pp (kN)	Mín.	<input type="text"/>	Máx.	<input type="text"/>
<input type="checkbox"/> Custo (R\$)	Mín.	<input type="text"/>	Máx.	<input type="text"/>
<input checked="" type="checkbox"/> Taxa As Neg1	Mín.	<input type="text" value="0"/>	Máx.	<input type="text" value="1"/>
<input checked="" type="checkbox"/> Taxa As pos	Mín.	<input type="text" value="0"/>	Máx.	<input type="text" value="1"/>
<input checked="" type="checkbox"/> Taxa As neg2	Mín.	<input type="text" value="0"/>	Máx.	<input type="text" value="1"/>
<input checked="" type="checkbox"/> KX pos	Mín.	<input type="text"/>	Máx.	<input type="text" value="0.45"/>
<input checked="" type="checkbox"/> KX neg1	Mín.	<input type="text"/>	Máx.	<input type="text" value="0.45"/>
<input checked="" type="checkbox"/> KX neg2	Mín.	<input type="text"/>	Máx.	<input type="text" value="0.45"/>
<input checked="" type="checkbox"/> Taxa de dmin/d neg1	Mín.	<input type="text"/>	Máx.	<input type="text" value="1"/>
<input checked="" type="checkbox"/> Seção invialvel	Mín.	<input type="text" value="-0.5"/>	Máx.	<input type="text"/>
<input checked="" type="checkbox"/> Taxa Tsd/Trd2i	Mín.	<input type="text"/>	Máx.	<input type="text" value="1"/>
<input checked="" type="checkbox"/> Taxa de dmin/d pos	Mín.	<input type="text"/>	Máx.	<input type="text" value="1"/>
<input checked="" type="checkbox"/> Taxa de dmin/d neg2	Mín.	<input type="text"/>	Máx.	<input type="text" value="1"/>
<input checked="" type="checkbox"/> Taxa Arm Con neg1	Mín.	<input type="text"/>	Máx.	<input type="text" value="1"/>
<input checked="" type="checkbox"/> Taxa Arm Con neg2	Mín.	<input type="text"/>	Máx.	<input type="text" value="1"/>
<input checked="" type="checkbox"/> Taxa Arm Con pos	Mín.	<input type="text"/>	Máx.	<input type="text" value="1"/>

Fonte: Autor (2021)

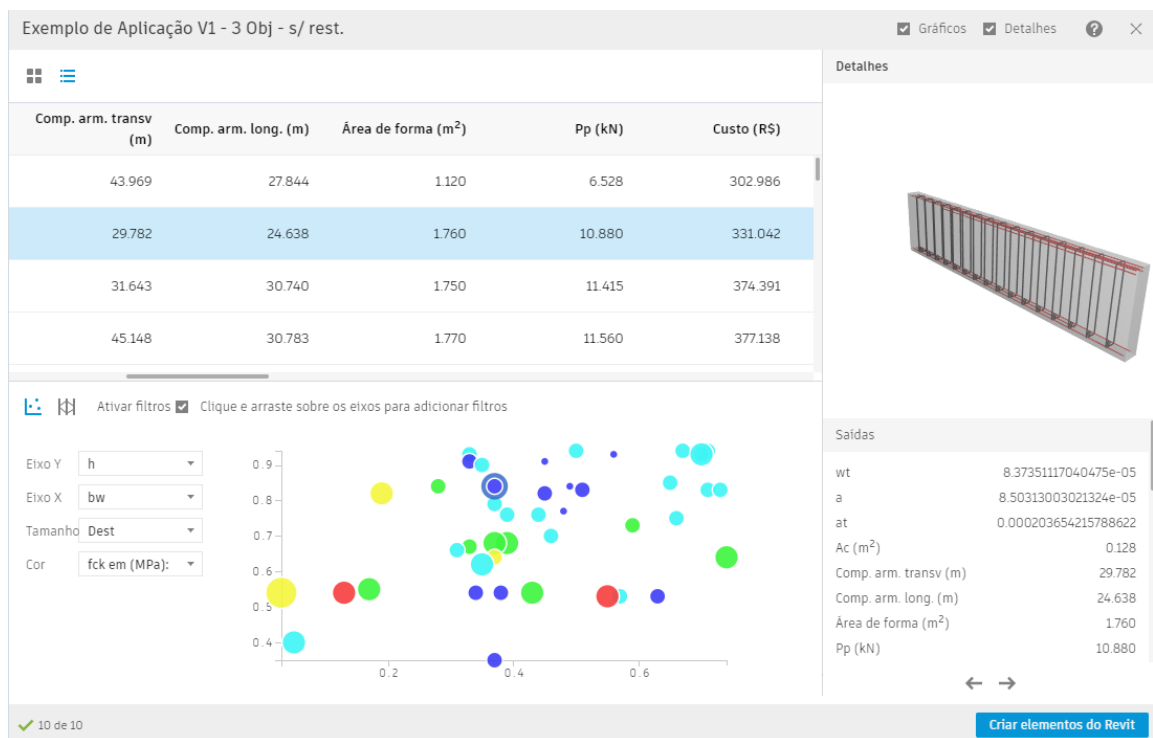
Nos resultados do *Generative Design* são apresentadas as opções de projetos e os gráficos. As opções de projeto, ficam na parte superior a esquerda e podem ser apresentadas em grade como na Figura 34 ou em lista como na Figura 35. Já os gráficos, podem ser apresentados em gráficos de coordenadas paralelas ou gráfico de dispersão, Figura 34 e Figura 35 respectivamente. Qualquer opção de projeto selecionada é imediatamente detalhada no lado direito.

Figura 34 - Resultados modelo 1



Fonte: Autor (2021)

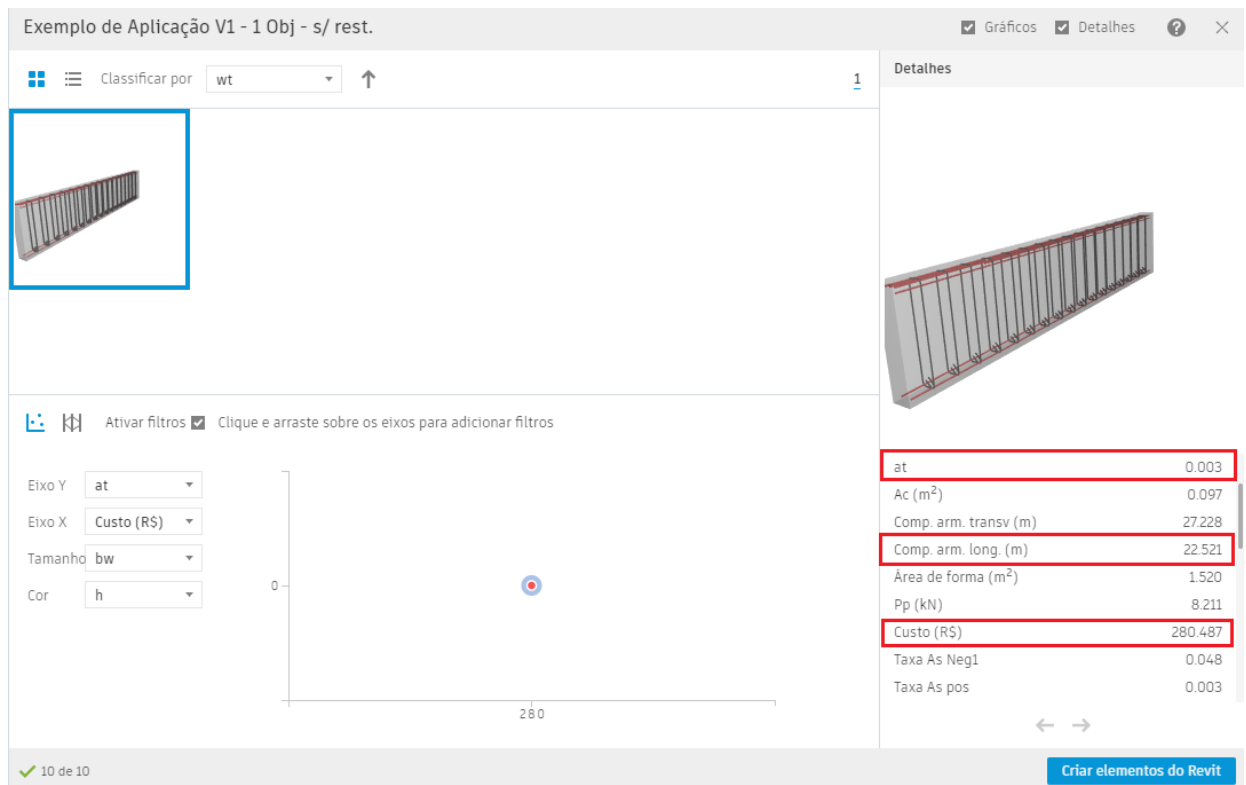
Figura 35 - Resultados modelo 2



Fonte: Autor (2021)

O resultado da otimização com apenas um objetivo é apresentado na Figura 36. Note que, para um objetivo o algoritmo naturalmente apresenta apenas uma opção de projeto. Observe que estão destacados os valores de flecha total e comprimento de armadura dessa opção. Esses valores são importantes para serem comparados com os estudos com mais objetivos.

Figura 36 - Resultado para um objetivo

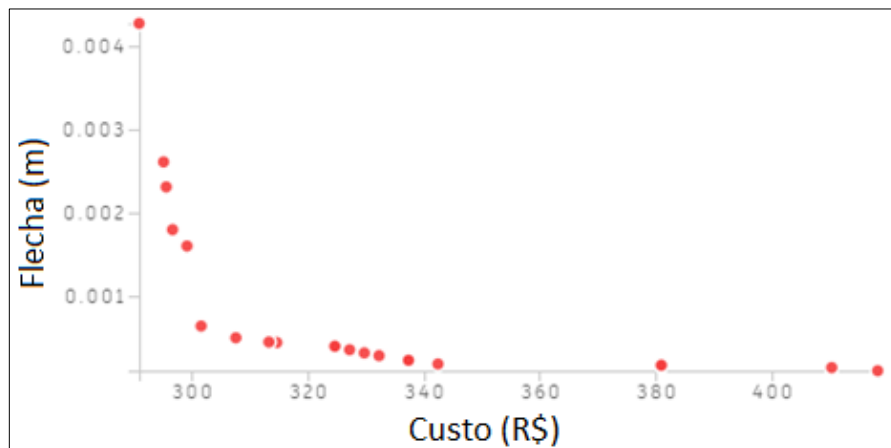


Fonte: Autor (2021)

O gráfico obtido da otimização de objetivo duplo, Figura 37, apresenta um conjunto de soluções ideais expressos pela Fronteira de Pareto. Além disso, também não existe nenhuma solução na frente ideal que seja superada em ambos os objetivos por uma outra solução, ou seja, não apresenta as soluções não-dominantes.



Figura 37 - Frente de Pareto objetivo duplo com 10 gerações

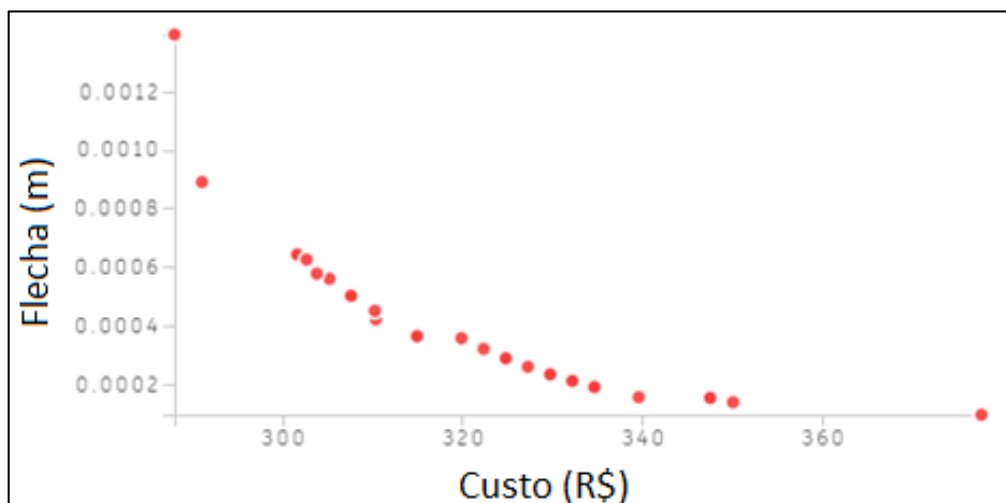


Fonte: Autor (2021)

Observe que, o estudo com objetivo duplo apresentou a maioria das soluções mais custosas que o estudo com um único objetivo, porém estas soluções apresentam uma flecha total menor. Esse comportamento é esperado uma vez que a otimização de objetivo único está comprometida apenas com o custo, portanto entrega resultados melhores deste avaliador.

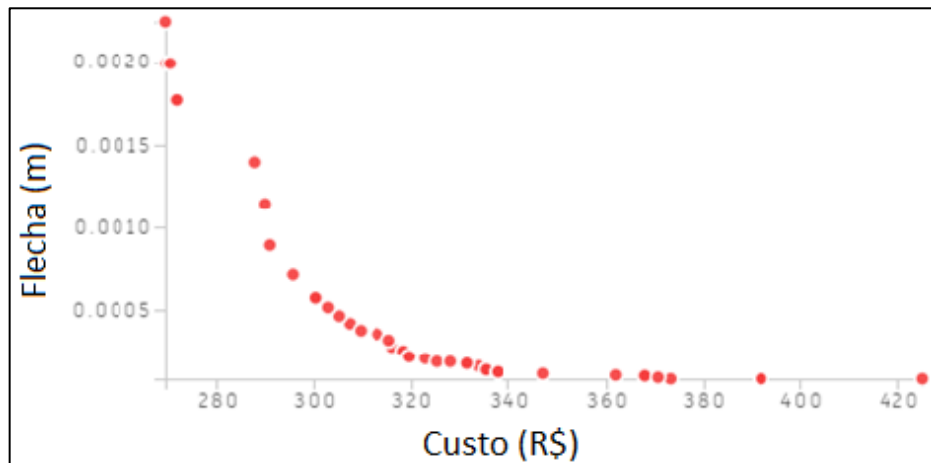
O estudo foi repetido com 20 (Figura 38) e 40 gerações (Figura 39), desse modo, o estudo permite a exploração de opções de projeto melhores e com menores valores de custo e flecha total. Além disso, um maior número de gerações vai formar uma Frente de Pareto mais clara.

Figura 38 - Frente de Pareto objetivo duplo com 20 gerações



Fonte: Autor (2021)

Figura 39 - Frente de Pareto objetivo duplo com 40 gerações



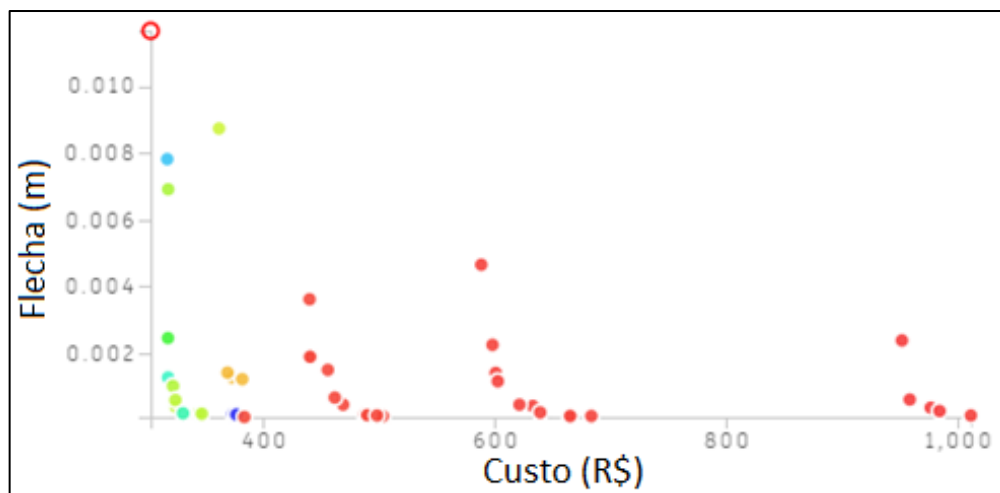
Fonte: Autor (2021)

O gráfico resultante do estudo de objetivo triplo naturalmente apresenta uma Frente de Pareto no espaço tridimensional, Figura 40

O gráfico da Figura 40, tem alguns formatos de Fronteiras de Pareto, estas são fronteiras ideais de diferentes regiões do espaço de projeto. Esse efeito é causado devido a tridimensionalidade dos objetivos e a presença de alguma descontinuidade.

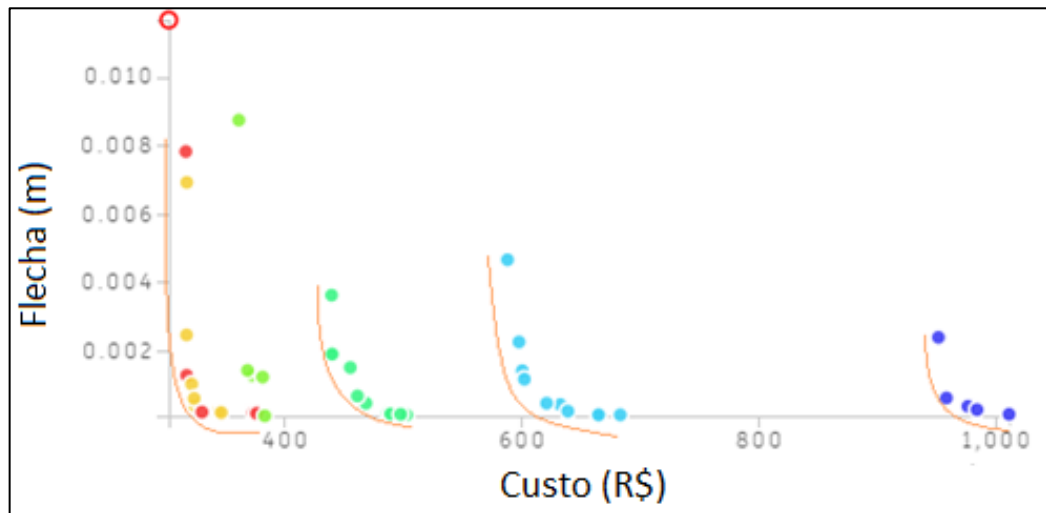
O gráfico da Figura 41 é semelhante ao anterior, no entanto, a cor dos pontos representa o diâmetro da armadura. Nota-se que cada uma das frentes ideais tem soluções com diâmetros diferentes. Isto aponta a descontinuidade que ocorre na mudança de diâmetro de uma barra para outra.

Figura 40 - Frente ideal de objetivo triplo: Cores mais quentes para maiores comprimentos.



Fonte: Autor (2021)

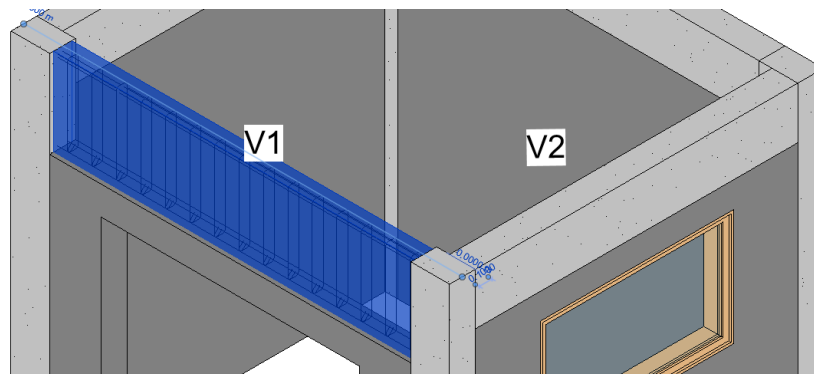
Figura 41 - Frente ideal de objetivo triplo: Cores mais frias para os menores diâmetros.



Fonte: Autor (2021)

Após a exploração das opções de projeto o *Generative Design* exporta a solução selecionada para o restante do projeto, integrando-o a arquitetura como na Figura 42.

Figura 42 - Integração da opção de projeto no modelo



Fonte: Autor (2021)

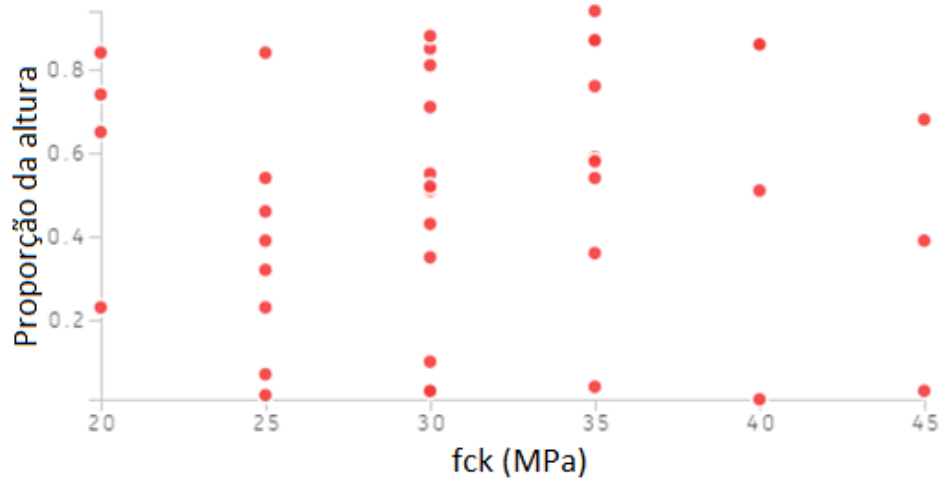
## 4.2 Exploração do espaço de projeto por diferentes solucionadores

Neste subcapítulo será discutido como os diferentes solucionadores genéticos do *Generative Design* exploram o espaço de projeto aplicados a viga V2. Os gráficos deste capítulo são construídos com a porcentagem da altura da viga e o  $f_{ck}$  do concreto. A altura da viga é dada pela proporção decimal onde 0 é a altura mínima de 26 cm e 1 é a altura máxima que vai até o limite da restrição.

Cada solucionador ou método irá explorar o espaço de projeto de formas diferentes.

O método de 'Randomizar' vai explorar todo espaço de projeto de forma aleatória. Note que, os eixos de altura e de  $f_{ck}$  compreendem todo o espaço de projeto e a distribuição de pontos é aleatória, Figura 43.

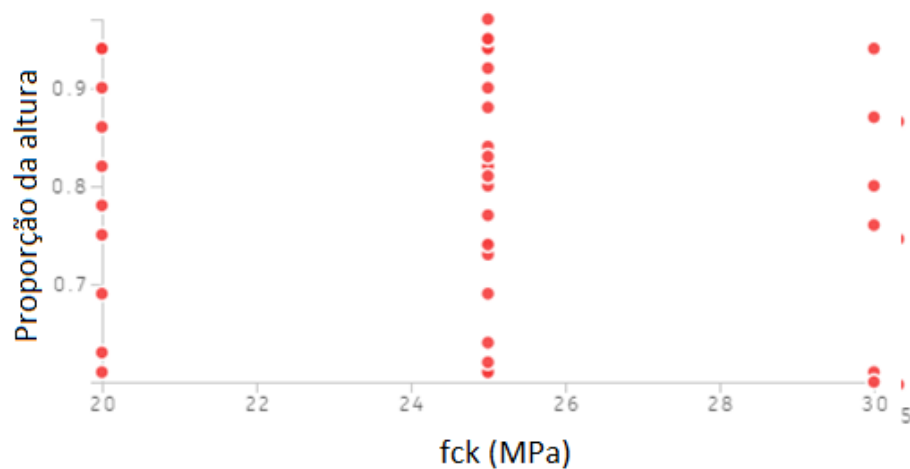
Figura 43 - Espaço de projeto do Método Randomizar



Fonte: Autor (2021)

Já o método Semelhante, cria opções de projetos semelhantes a uma opção inicial que o projetista já gosta. Para esse exemplo foi atribuído um valor aleatório inicial para os parâmetros de altura e  $f_{ck}$ : 0,8 e 25 *Mpa* respectivamente. Neste caso, as opções estão concentradas numa região do espaço de projeto, o  $f_{ck}$  só varia em 3 valores e a porcentagem de altura varia num intervalo limitado no espaço de projeto, como na Figura 44.

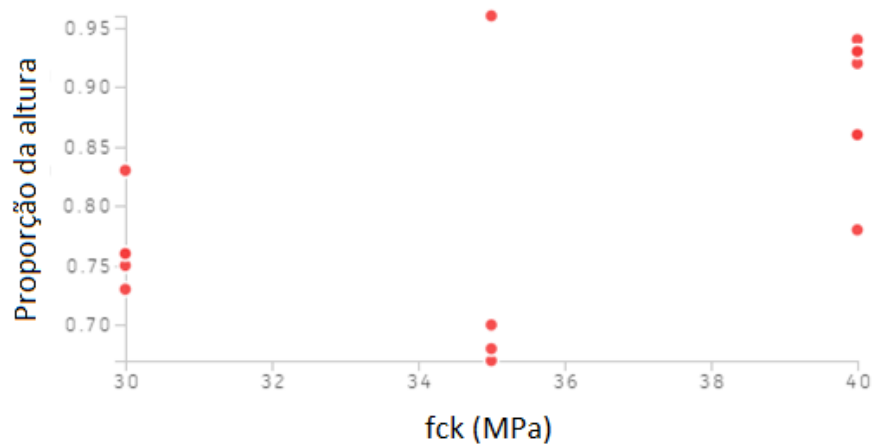
Figura 44 - Espaço de projeto do Método Semelhante



Fonte: Autor (2021)

O método de Otimização também retorna opções pouco dispersas no espaço de projeto, no entanto este método vai escolher de forma inteligente as regiões ótimas. Neste gráfico os eixos também estão limitados, Figura 45, no entanto, diferente do método anterior, o algoritmo genético escolheu a região que irá se concentrar, por exemplo, procura valores de  $f_{ck}$  mais altos. Vale destacar que a otimização embora teste 200 opções de projetos, somente 13 opções foram consideradas ideais e pertencem a solução deste método.

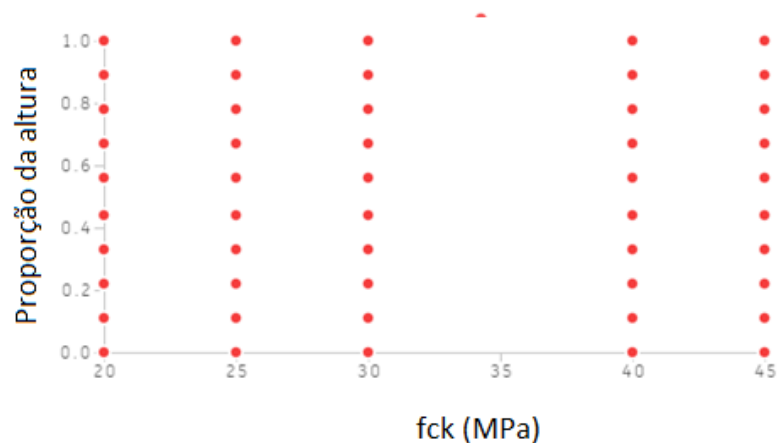
Figura 45 - Espaço de projeto da otimização



Fonte: Autor (2021)

Por fim, o produto transversal explora todo espaço de projeto e dispersa as opções de projeto de forma padronizada e equidistantes como na Figura 46. Uma vez que o espaço de projeto definido no item 3.2.1 é muito grande, para tornar esse método exequível, o estudo aumentou o passo das variáveis.

Figura 46 - Espaço de projeto do produto transversal



Fonte: Autor (2021)

## 5 CONSIDERAÇÕES FINAIS

Ao início deste trabalho de pesquisa constatou-se a necessidade de avançar no estudo de métodos exploração de soluções de projetos na área de estruturas. Por isso, o presente trabalho dedicou-se ao estudo do projeto generativo aplicado a vigas de concreto armado.

Diante disso, a pesquisa teve como objetivo geral avaliar o desempenho do Projeto Generativo como ferramenta auxiliar para projeto de vigas de concreto armado. Por isso, inicialmente, apresentou-se os fundamentos básicos do projeto generativo, desenvolveu-se projeto computacional e por fim foi apresentado seus resultados.

Na análise dos resultados observou-se que o projeto computacional se comportou como esperado em relação a sua interoperabilidade com o *Revit*. Pois identificou corretamente elementos e dados do projeto, gerou opções de projeto otimizadas, apresentou estas opções e integrou a opção escolhida no projeto.

Além disso, os gráficos de otimização das opções de projeto apresentaram os resultados esperados. Apresentaram a formação da Fronteira de Pareto nas otimizações multiobjetivo e as descontinuidades do problema também foram verificadas nos gráficos.

Por fim, os resultados da exploração do espaço de projeto pelos solucionadores genéticos. Nestes, todos solucionadores apresentaram gráficos que atendiam as expectativas particulares de cada método.

Portanto, uma vez que foram encontrados os resultados esperados constata-se o correto funcionamento do projeto computacional e o bom desempenho do projeto generativo como ferramenta auxiliar na exploração de soluções de projeto. Assim sendo, os objetivos propostos foram verificados.

Vale ressaltar algumas limitações da estrutura paramétrica desenvolvida. O projeto computacional se limita a um detalhamento simplificado da armadura, pois calcula apenas a armadura mínima nos momentos máximos da viga. Além disso, resolve somente vigas biapoiada submetidas ao modelo de carregamento da Figura 22.

Outra limitação do projeto computacional natural da simplicidade do projeto de viga proposto é a pouca flexibilidade dos parâmetros, ou seja, apresentando baixa variância.

Propõem-se para trabalhos futuros o refinamento no cálculo e detalhamento da armadura, adicionando armadura de pele, dobras de ancoragem, possibilidade de usar diferentes bitolas e escalonamento.

Outra perspectiva é de trabalhos que exploram espaços de projeto com maior variância. Isto pode ser alcançado ampliando o espaço de projeto, permitindo soluções de vigas contínuas e em balanço ou submetida a carregamentos mais variados. Também o desenvolvimento de projetos computacionais de outras peças estruturais e de projetos de otimização de problemas de outras disciplinas que apresentam maior complexidade.

## 6 REFERÊNCIAS

- ABRISHAMI, S. et al. Integration of BIM and generative design to exploit AEC conceptual design innovation. **Journal of Information Technology in Construction (ITcon)**, Preston, v. 19, n. 2014, p. 350-359, setembro 2014. ISSN 1874-4753.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6118: Projeto de estruturas de concreto — Procedimento**. Rio de Janeiro. 2014.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 7480: Aço destinado a armaduras para estruturas de concreto armado — Especificações**. Rio de Janeiro. 2007.
- BUCHANAN, B. G. A (Very) Brief History of Artificial Intelligence. **AI Magazine**, v. 26, n. 4, p. 53-60, 2006.
- CARVALHO, R. C.; FILHO, J. R. F. **Cálculo e Detalhamento de Estruturas de Concreto Armado**: Segundo a NBR 6118:2014. 4ª Ed. São Carlos: EdUFSCar, 2016.
- DEB, K. **Multi-Objective Optimization Using Evolutionary Algorithms**: An Introduction. New York: John Wiley & Sons, 2001.
- DEB, K. et al. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. **IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION**, v. 6, n. 2, p. 182-197, Abril 2002.
- DYNAMO. The Dynamo Primer – Disponível em: <<https://primer.dynamobim.org/index.html>>. Acesso em 01 mar. 2021.
- FERREIRA, B.; LEITÃO, A. Generative Design for Building Information Modeling. **Real Time - Proceedings of the 33rd International Conference on Education and Research in Computer Aided Architectural Design in Europe**, Vienna, Austria, v. 1, p. 635-644, 2015.
- KRISH, S. A Practical Generative Design Method. **Comput. Aided**, v. 43, p. 88-100, Janeiro 2011.
- LACERDA, E. G. M.; CARVALHO, A. C. P. L. F. Introdução aos Algoritmos Genéticos. In.: GALVÃO, C. O. (org.); VALENÇA, M. J. S. (org.): **Sistemas Inteligentes: aplicações a recursos hídricos e sistemas ambientais**, Porto Alegre, p. 99-150.
- LINCZUK, V. C. C.; BASTOS, L. E. G. Otimização multiobjetivo orientada ao desempenho térmico para o projeto de edificações de baixo consumo de energia na Região Sul do Brasil. **Ambiente Construído**, Porto Alegre, v. 20, n. 4, p. 509-529, dezembro 2020. ISSN 1415-8876.



LINDEN, R. **Algoritmos Genéticos - Uma importante ferramenta da Inteligência Computacional**. 2ª. ed. Rio de Janeiro: Brasport, 2008.

MUKKAVAARA, J.; SANDBERG, M. Architectural Design Exploration Using Generative Design: Framework Development and Case Study of a Residential Block. **Buildings**, Luleå, v. 10, n. 11, p. 1-17, novembro 2020. ISSN 20755309.

NAGY, D.; VILLAGGI, L.; ZHAO, D.; BENJAMIN, D. Beyond Heuristics: A novel design model for generative space planning in architecture. **SimAUD**, New York, junho 2018

NAGY, D.; VILLAGGI, L.; BENJAMIN, D. Generative Urban Design: Integrating Financial and Energy Goals for Automated Neighborhood Layout. **ACADIA**, p. 436-445, 2017.

NAGY, D. Generative Design. **Medium**, 2017. Disponível em:  
<<https://medium.com/generative-design>>. Acesso em: 01 mar. 2021.

ORSE Sistemas de Orçamentos de Obras de Sergipe / Insumos – Disponível em:  
<<http://orse.cehop.se.gov.br/>>. Acesso em 01 mar. 2021.

OSTROWSKA-WAWRYNIUK, K. BIM-Aided Prefabrication for Minimum Waste DIY Timber Houses. **Proceedings of 37 eCAADe and XXIII SIGraDi Joint Conference, “Architecture in the Age of the 4Th Industrial Revolution”**, Porto, v. 1, p. 251-258, setembro 2019.

SANTIAGO, P. Evolutionary Optimization of Building Facade Form for Energy and Comfort in Urban Environment through BIM and Algorithmic Modeling A case study in Porto, Portugal, v. 2, p. 153-160, 2020.

SILVA, L.; SILVA, N.; LACROIX, I. Integrating Parametric Modeling With BIM Through Generative Programming for the Production of Nurbs Surfaces and Structures, Brasília, v. 1, p. 635-644, 2019.

SMITH, L. Generative Design Primer. **Generative Design**, 2020. Disponível em:  
<<https://www.generativedesign.org/>>. Acesso em: 01 mar. 2021.

SUERMAN, P. C.; ISSA, R. R. A. Evaluating industry perceptions of building information modeling (BIM) impact on construction. **Journal of Information Technology in Construction (ITcon)**, v. 14, p. 574-594, Agosto 2009. ISSN 1874-4753.

VELDHUIZEN, D. A. V.; LAMONT, G. B. Evolutionary Computation and Convergence to a Pareto Front. In: **Late Breaking Papers at the Genetic Programming Conference**, p. 221 – 228, Stanford University, California, 1998.