



SENAI CIMATEC

PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM

COMPUTACIONAL E TECNOLOGIA INDUSTRIAL

Mestrado em Modelagem Computacional e Tecnologia Industrial

Dissertação de mestrado

Sistema Generativo de Projeto Aplicado ao Desenho e Otimização da Estrutura de um *Shed*

Apresentada por: Fernando Ferraz Ribeiro

Orientador: Davidson Martins Moreira

Co-orientador: Arivaldo Leão de Amorim

Abril de 2015

Fernando Ferraz Ribeiro

Sistema Generativo de Projeto Aplicado ao Desenho e Otimização da Estrutura de um *Shed*

Dissertação de mestrado apresentada ao Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial, Curso de Mestrado em Modelagem Computacional e Tecnologia Industrial do SENAI CIMATEC, como requisito parcial para a obtenção do título de **Mestre em Modelagem Computacional e Tecnologia Industrial**.

Área de conhecimento: Interdisciplinar

Orientador: Davidson Martins Moreira

SENAI CIMATEC

Co-orientador: Arivaldo Leão de Amorim

Faculdade de Arquitetura - UFBA

Salvador
SENAI CIMATEC
2015

Nota sobre o estilo do PPGMCTI

Esta dissertação de mestrado foi elaborada considerando as normas de estilo (i.e. estéticas e estruturais) propostas aprovadas pelo colegiado do Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial e estão disponíveis em formato eletrônico (*download* na Página Web http://ead.fieb.org.br/portal_faculdades/dissertacoes-e-teses-mcti.html ou solicitação via e-mail à secretaria do programa) e em formato impresso somente para consulta.

Ressalta-se que o formato proposto considera diversos itens das normas da Associação Brasileira de Normas Técnicas (ABNT), entretanto opta-se, em alguns aspectos, seguir um estilo próprio elaborado e amadurecido pelos professores do programa de pós-graduação supracitado.

SENAI CIMATEC

Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial

Mestrado em Modelagem Computacional e Tecnologia Industrial

A Banca Examinadora, constituída pelos professores abaixo listados, leram e recomendam a aprovação [com distinção] da Dissertação de mestrado, intitulada “Sistema Generativo de Projeto Aplicado ao Desenho e Otimização da Estrutura de um *Shed*”, apresentada no dia (dia) de (mês) de (ano), como requisito parcial para a obtenção do título de **Mestre em Modelagem Computacional e Tecnologia Industrial**.

Orientador:

Prof. Dr. Davidson Martins Moreira
SENAI CIMATEC

Co-orientador:

Prof. Dr. Arivaldo Leão de Amorim
Faculdade de Arquitetura - UFBA

Membro externo da Banca:

Prof. Dr. Felipe Tavares da Silva
Faculdade de Arquitetura - UFBA

Membro interno da Banca:

Prof. Dr. Renelson Ribeiro Sampaio
SENAI CIMATEC

Membro interno da Banca:

Prof. Dr. Gilney Figueira Zebende
SENAI CIMATEC

Dedico este trabalho a minha mãe,
Aidil Nunes Ferraz Ribeiro,
pelo amor e apoio que nunca me faltaram;
e a memória de meu pai,
Ubirajara Gomes Ribeiro,
pelas inesquecíveis lições de Geometria,
mas principalmente,
pelos esquadros e compassos da vida.

Agradecimentos

Um trabalho desta natureza não pode ser realizado sem a inestimável ajuda de muitas pessoas. É o resultado de experiências profissionais, pessoais e acadêmicas que se acumulam, ganham forma e me fazem olhar para minha trajetória profissional com felicidade e gratidão. Agradeço a meu orientador, Davidson Martins Moreira e meu co-orientador, Arivaldo Leão de Amorim, por todos os conselhos, críticas, orientações e pelo trato amistoso e paciente que sempre me reservaram. Aos professores do programa de pós-graduação PPG-MCTI e aos professores e palestrantes convidados, tanto pelo conhecimento dividido, quanto pela oportunidade de crescimento acadêmico e pessoal que me proporcionaram. Aos funcionários do colegiado e secretaria do mesmo programa, pela maneira solícita que exercem suas funções. Aos colegas de turma, pelo ambiente de amizade e colaboração que marcaram nossa convivência. À equipe do LCAD da Faculdade de Arquitetura da UFBA, e a professora Natalie Johanna Groetelaars, que me receberam durante a atividade de tirocínio docente, tanto pela oportunidade de acompanhar as aulas e o desenvolvimento da turma durante os trabalhos e avaliações, quanto pelos conhecimentos divididos e multiplicados. À equipe do Instituto Brasileiro de Tecnologia do Habitat, representada pelo seu presidente, o arquiteto João Filgueiras Lima (Lelé), pelo período de intenso trabalho e pela visão do papel e da profissão que fui apresentado. À equipe do escritório Eleve Engenharia, onde pude aprimorar meus conhecimentos sobre estruturas. Aos membros das bancas de qualificação e defesa, pelas críticas e correções fundamentais e sempre pertinentes. Aos amigos arquitetos, companheiros de muitos projetos e colaborações: Deraldo Tourinho, Arlindo Esteve, Daniel Gomes, Márcio Targa, Mário Vitor Bastos e Luiz Villa. Aos amigos pesquisadores de diversas áreas, pelas conversas e conselhos: Sandra Maria Ferraz Mello (amiga, tia e madrinha), Alexis Dourado Guedes, Lorena Guedes, Marcos Rodrigues, Lia Seixas e Paolo Bruni. À família de meu tio, Archibaldo Ferraz Melo, com um pedido de desculpas por ter alterado o roteiro de suas férias para que me trouxessem de presente um livro utilizado nesta pesquisa. Ao grande amigo Alessandro Soares, a quem sempre recorro nas dúvidas com o idioma inglês e conselhos sobre muito mais. Aos amigos que não desistiram de mim, me procurando mesmo sabendo que eu passaria os fins de semana imerso em trabalho: Maurício Cerqueira, Rogério Ribeiro (amigo e irmão), Norbert Siering, Carlos Pereira, Miguel Cordeiro e Juliano Pontes. À minha grande amiga Carolina de Britto Fernandes, pelo carinho e apoio em todas as horas. Agradeço também a todos os amigos e familiares, que acreditando e me incentivando tornaram essa jornada possível.

Salvador, Brasil
30 de Abril de 2015

Fernando Ferraz Ribeiro

Resumo

O presente trabalho é uma investigação sobre a teoria e as aplicações dos sistemas gerativos de projeto como ferramenta de auxílio na tomada de decisões para a concepção de formas na indústria da construção civil. Adotando uma metodologia experimental, um conjunto de algoritmos foi proposto como instrumentos de concepção e desenho de um grupo de treliças de suporte de uma cobertura tipo *shed*, inspiradas na obra do arquiteto João Filgueiras Lima (Lelé). Além da geração de formas, o sistema conta com uma ferramenta de análise de estruturas, cuja modelagem foi baseada na Grafostática. Uma metodologia de trabalho e abordagem das questões relativas ao projeto arquitetônico é proposta, e as contribuições deste método para a concepção de elementos construtivos são avaliadas e discutidas.

Palavras Chave: Sistemas Generativos de Projeto, Algoritmos Generativos, Modelagem Paramétrica, Grafostática, Arquitetura.

Abstract

This Dissertation is a research on the theory and applications of the generative design systems as a tool in decision making for defining forms in the construction industry. Adopting an experimental methodology, a set of algorithms was proposed as assisting instruments aimed to design a group of shed trusses inspired by the work of architect João Filgueiras Lima (Lelé). Besides the generation of forms, the system has a structural analysis tool, which was based on Graphical Statics. A working methodology and an approach to architectural design related topics is proposed, and the contributions of this method for the design of building elements are evaluated and discussed.

Keywords: Generative Design Systems, Generative Algorithms, Parametric Modeling, Graphical Statics, Architecture.

Sumário

1	Introdução	1
1.1	Definição do Problema	3
1.1.1	Objeto de Estudo	6
1.2	Objetivos	7
1.3	Importância da Pesquisa	7
1.4	Limites e Limitações	10
1.5	Questões e Hipóteses	11
1.6	Aspectos Metodológicos	11
1.7	Organização da Dissertação de Mestrado	13
2	Algoritmos Generativos e Sistemas Generativos de Projeto	14
2.1	Definição de Algoritmo e Algoritmos Generativos	14
2.2	Principais Modelos de Algoritmos Generativos Aplicados à Arquitetura e ICC	18
2.2.1	Gramáticas de Forma (GF)	18
2.2.2	<i>L-Systems</i> (LS)	19
2.2.3	Autômatos Celulares (AC)	21
2.2.4	<i>Swarm Intelligence</i> (SI)	23
2.2.5	Diagramas de Voronoi (DV)	24
2.2.6	Algoritmos de Otimização Estocástica	26
2.2.6.1	Algoritmos Genéticos	26
2.2.6.2	<i>Simulated Annealing</i> (SA)	28
2.2.7	Fractais	31
2.2.8	Modelos de Parâmetros e Restrições (MPR)	35
3	Grafostática	40
3.1	Estática e Grafostática	40
3.2	Precisão do Método	42
3.3	Conceitos Fundamentais da Estática	43
3.4	Diagramas Recíprocos	44
3.5	Condições de Equilíbrio	46
3.6	Problemas em Equilíbrio	47
3.6.1	Caso Um - Composição de uma Força	47
3.6.2	Caso Dois - Resolução de uma Força em Duas Componentes	48
3.6.3	Caso Três - Resolução de uma Força em uma Componente de Direção Conhecida e Outra que Passe por um Ponto	49
3.6.4	Caso Quatro - Resolução de uma Força em Três Componentes	50
3.7	Cálculo das Reações e Esforços em Treliças Planas Bi-Apoiadas	51
3.7.1	Determinação Estática de Treliças	51
3.7.2	Regras de Desenho para Obtenção de Diagramas Recíprocos	53
3.7.3	Cálculo das Reações	53
3.7.4	Cálculo dos Esforços	53
3.8	Cálculo das Reações e Esforços em Treliças Planas em Balanço	56
3.9	Outros Algoritmos da Grafostática	57
3.10	Ferramentas Computacionais	59

4 Ambiente de Implementação	60
4.1 <i>Rhinoceros 3D</i>	61
4.2 <i>Grasshopper</i>	61
4.3 <i>Python Script</i>	65
5 Trabalho experimental e desenvolvimento da pesquisa	66
5.1 Divisão da Treliça em Elementos e Nomenclatura Utilizada	67
5.2 Algoritmo de Geração dos Eixos	68
5.2.1 Parâmetro 1: Curva	69
5.2.2 Parâmetro 2: Eixo de Simetria	70
5.2.3 Parâmetro 3: HV1	71
5.2.4 Parâmetros 5 e 7: HV2 e HV3	71
5.2.5 Parâmetros 4, 6 e 8: diagV1, diagV2 e diagV3	72
5.2.6 Parâmetro 9: No_Shed	72
5.2.7 Parâmetro 10: VigaDist	72
5.2.8 Parâmetro 11: Plano	73
5.2.9 Função 01: Geração dos Eixos das Vigas v1 e v2	74
5.2.10 Função 02: Geração dos Eixos da Viga v3	75
5.2.11 Teste de Determinação Estática	75
5.2.12 Saídas do Algoritmo	76
5.3 Algoritmo de Geração da Volumetria	76
5.3.1 Parâmetro 1: Mostrar Volume	77
5.3.2 Parâmetro 2: N_Tr	78
5.3.3 Parâmetro 3: Cobertura	78
5.3.4 Parâmetro 4: Eixos	78
5.3.5 Parâmetro 5: Dist_entre_eixos	78
5.3.6 Parâmetro 6: Peso_cobertura	79
5.3.7 Parâmetro 7: Peso_esp_Tr	79
5.3.8 Parâmetro 8: Fator_de_Conv	79
5.3.9 Parâmetros 9 e 10: Diag_e e Diag_l	79
5.3.10 Parâmetros 11 e 12: Bz_e e Bz_l	79
5.3.11 Parâmetro 13: Plano	79
5.3.12 Função 01: Geração dos volumes	80
5.3.13 Funções 02 e 03: Cargas	80
5.3.14 Funções 04 e 05: Áreas e Momentos de Inércia Mínimos	81
5.3.15 Saídas do Algoritmo	82
5.4 Aplicação e Validação Dos Algoritmos de Geração dos Eixos e dos Volumes	82
5.5 Algoritmo de Análise Grafostática	84
5.5.1 Definição do Sentido de Seleção das Linhas de Ação	88
5.5.2 Método Grafostático Aplicado à Viga do <i>Shed</i>	89
5.6 Método Grafostático Aplicado às Vigas v1, v2 e Conector	92
5.7 Validação para Treliças Geradas por Curvas Iniciais no Sentido Dirieta-Esquerda	94
5.8 Teste do Dimensionamento dos Menbros da Treliça	97
5.9 Parâmetros do Algoritmo de Análise Grafostática	98
5.10 Funções do Algoritmo de Análise Grafostática	99
5.11 Saídas do Algoritmo de Análise Grafostática	103
5.12 Aplicação do Algoritmo de Análise Grafostática	103

6 Considerações finais	105
6.1 Conclusões	106
6.2 Contribuições	107
6.2.1 Principais Produtos e Algoritmos Desenvolvidos	108
6.3 Atividades Futuras de Pesquisa	109
A Algoritmos Baseados na Grafostática	110
A.1 Algoritmo de Soma de Vetores e Cálculo da Equilibrante	110
A.2 <i>Cluster</i> Utilizado para a Visualização de Vetores	110
A.3 Algoritmo de Desenho do PF, FG, Funicular e Cálculo da Resultante e Momento	111
B ALgoritmos Generativos no <i>Rhinoceros 3D / Grasshopper</i>	112
B.1 Triângulo de Sierpinski	112
B.2 Cúpula Móvel	113
Referências Bibliográficas	114

Lista de Figuras

1.1	Modelo Teórico	4
1.2	Proposta para o Mercado São Miguel - Autoria: João Filgueiras Lima e Instituto Habitat - Trecho do "Corte AA" - Fonte: Acervo do Instituto Habitat	6
1.3	Consumo de Energia - Gráfico produzido pela ONG Architecture 2030. Com base nos dados coletados pela agência U.S. Energy Information Administration (EIA) referentes ao ano de 2012	9
1.4	Fluxograma de um SGP - fonte: Bohnacker, Gross e Laub (2012)	12
2.1	Exemplo de algoritmo - subdivisão de um seguimento em um número finito (5) de partes iguais	16
2.2	Árvore Evolutiva das Gramáticas de Forma	19
2.3	Exemplos de aplicações dos L-Systems	20
2.4	Inverted Skyscraper Typology - fonte: EVolo (2009)	21
2.5	Ilustração da regra 90 dos AC elementares	22
2.6	Resultados obtidos a partir de sucessivas aplicações da regra 90 dos AC elementares - Triangulo de Sierpinski	23
2.7	Duas propostas para o conjunto de prédios em Aomori/japão - fonte: Herr e Kvan(2007)	23
2.8	Gerados no <i>Software Rhinoceros 3D</i>	25
2.9	Exemplos de Diagrmas de Voronoi na Arquitetura	25
2.10	Exemplo de Aplicações de Algoritmos Genéticos na Arquitetura	27
2.11	<i>Simulated Annealing</i> - Conceituação	29
2.12	Exemplos de Utilização dos Fractais na Arquitetura	32
2.13	Gerados no <i>Rhino3d/Grasshopper</i> utilizando o <i>add-on Hoopsnake</i> para uma e cinco iterações	33
2.14	Dimensão de Haussdorf - fonte: http://en.wikipedia.org/wiki/Fractal_dimension	34
2.15	Cúpulas móveis do Arquiteto João Filgueiras Lima (Lelé)	36
2.16	Cúpulas móveis geradas por MPR - <i>Rhinoceros/Grasshopper</i>	38
3.1	Precisão do cálculo	43
3.2	Diagramas Recíprocos - fonte: Hoskins (1899) (editado)	45
3.3	Caso 2 - forças concorrentes - fonte: Hoskins (1899)	48
3.4	Caso 2 - forças paralelas - fonte: Johnson (1908)	49
3.5	Caso 3 - fonte: Johnson (1908) (editado)	50
3.6	Caso 4 - fonte: Hoskins (1899)	51
3.7	Exemplo de Cálculo de Esforços de Treliças - fonte: Malcolm (1909)	52
3.8	Polígono de Forças - fonte: Malcolm (1909)	54
3.9	Satisfação das Condições de Equilíbrio de Translação - <i>L0</i> e <i>U1</i> - fonte: Malcolm (1909)	54
3.10	Satisfação das Condições de Equilíbrio de Translação - <i>L1</i> e <i>U2</i> - fonte: Malcolm (1909)	55
3.11	Satisfação das Condições de Equilíbrio de Translação - <i>L2</i> e <i>U3</i> - fonte: Malcolm (1909)	56

3.12 Exemplo de Cálculo de Esforços de Treliças em Balanço- fonte: Malcolm (1909)	56
4.1 Ambiente de Implementação	60
4.2 Grasshopper - Componentes e listas	62
5.1 Fluxograma Geral do Experimento	66
5.2 Captura de tela dos componentes onde os algoritmos foram implementados	67
5.3 Fluxograma Geral do Experimento	67
5.4 Algoritmo de Geração dos Eixos	69
5.5 Variação do Vão em Função do Parâmetro Eixo de Simetria	73
5.6 Algoritmo de Geração dos Eixos	77
5.7 Ilustração do Cálculo das Cargas	80
5.8 Comparação entre a treliça do Mercado São Miguel e a gerada pelo algoritmo	83
5.9 Variações na Geração de Treliças	84
5.10 Outros Exemplos de Variações na Geração de Treliças	85
5.11 Componente de Análise Grafostática	86
5.12 Carregamento da Treliça	86
5.13 Diagramas Recíprocos do Shed (viga v3)	88
5.14 Sentido de Seleção das Forças	89
5.15 Resolução de Reações em <i>Sheds</i> com Números Pares e Impares de Diagonais	90
5.16 Diagramas Recíprocos das Vigas v1 e v2	93
5.17 Sentido de Desenho das Curvas Iniciais	94
5.18 Treliças Simétricas	95
5.19 PFs e FGs dos <i>Sheds</i> Simétricos	95
5.20 FGs das Vigas v1 e v2 Simétricas	96
5.21 PFs das Vigas v1 e v2 Simétricas	96
6.1 Variações do Eixo de Simetria de uma Treliça	106
A.1 Captura de Tela do Algoritmo de Soma de Vetores	110
A.2 Captura de Tela do Algoritmo do Cluster de Visualização de Vetores	110
A.3 Captura de Tela do Algoritmo de Desenho do PF, FG, Funicular, Resultante e Momento	111
A.4 Entradas e saídas do componente	111
B.1 Algoritmo Paramétrico do Triângulo de Sierpinski 2D	112
B.2 Algoritmo Paramétrico para Análogo Tridimensional do Triângulo de Sierpinski 3D	112
B.3 Algoritmo paramétrico de uma Cúpula Móvel	113

Lista de Siglas

AC	Autômatos Celulares
AG	Algoritmos Generativos
BIM	Building Information Modeling
CG	Computação Gráfica
DOF	Design Optimization and Fabrication
DV	Diagrama de Voronoi
FG	Forma Geométrica
GF	Gramáticas de Forma
GS	Grafostática (Estática Gráfica)
ICC	Industria da Construção Civil
LS	L-Systems
MPR	Modelos de Parâmetros e Restrições
PF	Polígono de Forças
SA	Simulated Annealing
SGP	Sistema Generativo de Projeto
SI	Swarm Intelligence

Introdução

”Existe geometria no som das cordas;
existe música no espaçamento das esferas”

(Atribuído à Pitágoras de Samos)

No decorrer dos últimos 50 anos, grandes mudanças tem acontecido na Indústria da Construção Civil (**ICC**). As atividades de planejar e transformar o espaço ao redor de si, almejando uma ocupação ideal e deparando-se inevitavelmente com as possíveis apropriações reais, foram subitamente abordadas por novas formas, métodos de trabalho, demandas, críticas e paradigmas, estéticos e técnicos. É interessante notar o quanto pequeno é esse espaço de tempo em uma atividade cuja origem se encontra ligada à das primeiras civilizações humanas.

A força que as questões ecológicas gradativamente ganharam, desde a década de 1960, levaram as atuais preocupações de conservação de energia na fabricação, execução e uso das edificações. A Computação Gráfica (CG) desenvolve-se no mesmo período e colocou a disposição dos projetistas, primeiro novas ferramentas de desenho que rapidamente substituem a prancheta como instrumento prevalente no mercado, para então terem a sua recém conquistada hegemonia desafiada por sistemas paramétricos de projeto, algoritmos generativos (**AG**), Sistemas Generativos de Projeto (**SGP**) e pela aplicação do conceito BIM (***Building Information Modeling***). Algoritmos e programas de análise dos diversos comportamentos do edifício proporcionaram tanto uma maior velocidade de trabalho, quanto critérios de julgamento mais objetivos. A otimização ganha importância pelo valor que se tem dado ao desempenho energético das construções e economia de recursos na execução.

A crise propositiva do Movimento Moderno da Arquitetura, que simbolicamente culmina na demolição do conjunto habitacional Pruitt-Igoe em 1972, pode ser encarada como o rebatimento da crise do modelo mecanicista do universo na Filosofia das Artes. No livro ”Complexidade e Contradição na Arquitetura”, um manifesto e um dos textos chave para o estudo estético deste período de transição, Robert Venturi (1977) transforma o preceito do design minimalista **”less is more”** (menos é mais) do arquiteto Mies van der Rhei em **”less is a bore”** (menos é um tédio), afirmado que uma ”Simplicidade forçada resulta em uma simplificação excessiva”¹. Se parte do modernismo tem sua inspiração nas linhas aerodinâmicas dos carros, trens e aeronaves (**CORBUSIER, 1994**),

¹Forced simplicity results in oversimplification.

Paolo Portoghesi (2002, p. 35), em outro texto fundamental da mesma época, aponta o fato de que ”(...)calculadoras eletrônicas não são máquinas no sentido tradicional, de transformação de energia, mas instrumentos que operam de forma análoga ao cérebro humano”, dentre os motivos da atual inadequação dos valores que nortearam grande parte da produção arquitetônica da primeira metade do século XX.

Oxman (2006) reconhece o Museu Guggenheim de Bilbao como um catalizador da complexidade deste momento da Arquitetura em diversos contextos. Principalmente por introduzir, nas suas superfícies metálicas, curvas detalhadas com o suporte do *software* CATIA, a complexidade de novas formas geométricas, livres de formalismos anteriores². Picon (2011) aponta a retícula das fachadas do *Young Museum*(2005) dos arquitetos Herzog & de Meuron como um convite para uma experiência tátil, diluindo as barreiras entre objeto e observador, conceito que, segundo o autor, muitos dos projetistas que exploraram as potencialidades das ferramentas de projeto digital se identificam. Peter Eisenman (*EISENMAN; DOBNEY; Eisenman Architects.*, 1995, p. 130-135) propõe na sua ”Casa Guardiola” de 1988 um espaço resultante de operações matemáticas entre sólidos sobrepostos (interseção, subtração...) realizadas em projeto através de comandos de um programa CAD 3D. Mitchell (2008, p. 64) identifica nas formas fluidas e dinâmicas de alguns projetos contemporâneos, citando os trabalhos da arquiteta iraniana Zaha Hadid, uma alusão ao ambiente computacional que os originou. O grupo de arquitetos ingleses Archigram, reunidos por volta de 1961 em torno da revista de mesmo nome onde apresentavam sua ”abordagem infraestrutural, leve e *high-tech*”(*FRAMPTON*, 2008) da Arquitetura. Na proposta intitulada *Instant City* (1969), imaginava-se que ”as unidades habitáveis estivessem ligadas(...) a uma megaestrutura informacional que alimentaria e possibilitaria a maleabilidade do cotidiano urbano”(*DUARTE*, 1999, p. 104-105). A plena realização deste ideal estético só aconteceu após o desenvolvimento da tecnologia CAD, como no projeto do Museu de Arte de Graz (*Kunsthaus Graz*, conhecido também pelo apelido de *Friendly Alien*), inaugurado em 2003, de autoria de um dos antigos membros do grupo, o arquiteto Peter Cook. Coincidindo com um período importante na difusão das estruturas de informação atualmente disponíveis.

A presente pesquisa propõe um método de trabalho onde aspectos estruturais e estéticos de uma determinada treliça de sustentação de coberturas são analisados com o auxílio de algoritmos generativos, uma ferramenta de análise estrutural cuja modelagem foi baseada em alguns métodos da Grafostática e a possibilidade de utilização desta análise em um experimento de otimização da forma. Visando a aplicabilidade na prática projetual de alguns dos conceitos que vem mudando a maneira de encarar o projeto arquitetônico, bem como a sempre presente necessidade de comunicação e debate entre alguns dos profissio-

²In architecture, the Guggenheim Museum, Bilbao by Frank Gehry was the most prominent catalyst of theorizing new formal directions and postulating new design methods. Beyond the Postmodern sensibility of complexity through ”heterotopia”, or complex hybrids, the Guggenheim introduced the complexity of new geometric approaches freed from a priori formalisms, such as linguistic formalisms.

nais envolvidos no processo. Um exercício onde a complexidade inerente à Arquitetura, mostrada na mistura de técnica, tecnologia e estética que a define, pode ser apreciada e estudada enquanto procura-se colaborar com os debates em torno dos seus atuais paradigmas.

1.1 Definição do Problema

O experimento realizado propõe um **Sistema Generativo de Projeto** passível de ser implementado por arquitetos como ferramenta de auxílio nos primeiros estágios de concepção e definição de um elemento de projeto a ser desenvolvido, bem como servir de ponto de partida do diálogo com outras equipes, particularmente com os escritórios de cálculo e detalhamento de estruturas. O conceito de projeto generativo pode ser definido de maneira simples como ”(...)um algoritmo ou processo baseado em regras pelo qual várias potenciais soluções de projeto podem ser criadas”³(FASOULAKI, 2008).

O algoritmo em foco pode ser conceitualmente dividido em 3 partes que serão adiante explicadas:

- Um modelo geométrico baseado em parâmetros e restrições;
- Um algoritmo de análise estrutural;
- Um método de otimização.

Toda atividade de projeto, em qualquer área, mas particularmente nas que tem por objetivo a confecção de um produto específico, agrega em si uma complexidade de tópicos e áreas do conhecimento. No caso da construção, principalmente vista pelo prisma da Arquitetura, onde técnica, humanidades e arte se mesclam de forma peculiar, o entrelaçamento de diversos campos do saber apresenta um contorno não menos complexo. O modelo teórico representado na Figura 1.1 procura ilustrar as relações centrais confrontadas e aliadas neste trabalho.

A primeira das partes do algoritmo, o modelo geométrico baseado em parâmetros e restrições, corresponde no modelo teórico às ”flexões possíveis do modelo geométrico”. Imaginemos um retângulo ($L \times P$) onde a medida de um dos lados é definida por uma variável que representa um número real variando entre 2 e 8. Alterando esse valor entre os limites estabelecidos a Figura é redesenhada para que a medida do referido lado corresponda ao valor da variável. Para desenhar a outra dimensão do retângulo estabelece-se como

³*Generative Design can be broadly defined as an algorithmic or rule-based process through which various potential design solutions can be created.*

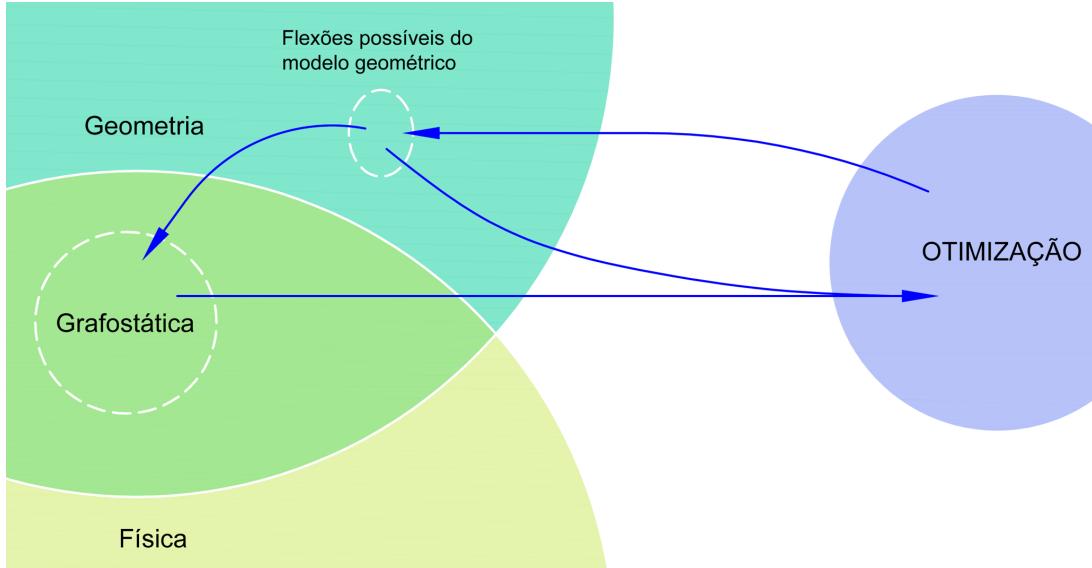


Figura 1.1: Modelo Teórico

regra que a área do polígono deve ser sempre igual a 10 *unidades*. Para $L = 2$, temos $P = \frac{10}{2} \Rightarrow P = 5$. Para $L = 8$, analogamente temos $P = 1,25$, e assim para qualquer valor intermediário ($L = 4 \Rightarrow P = 2,5$; $L = 3,2 \Rightarrow P = 3,125$). A variável que define o valor de L é chamada de parâmetro, e os valores limites desta e a obrigação de ter-se sempre uma área = 10 são denominados de restrições e as alterações nos valores das variáveis produzem as flexões do modelo.

A utilização deste tipo de parametrização modifica o método tradicional de projeto onde se propõe o objeto, desenha, analisa e repropõe. Aqui não se propõe um modelo, mas um conjunto de parâmetros e restrições capaz de gerar uma série de objetos para serem analisados como possíveis soluções para o problema. A etapa de análise não recai mais sobre um único objeto proposto, mas sobre uma gama de flexões. Esta etapa pode levar tanto a uma reconfiguração das variáveis, quanto a redefinição dos critérios de parametrização até que conduza ao objeto final, eleito como solução do problema de projeto.

Esta abordagem auxilia o projetista a explorar uma gama maior de soluções do que seria possível pelo método tradicional, principalmente quando os objetivos do projeto abrangem interesses subjetivos e criatividade ([KRISH, 2011](#)).

A análise de uma proposta de projeto pode ser feita unicamente pela técnica, experiência e subjetividade do projetista, quanto pode contar com o auxílio de ferramentas computacionais específicas para alguns de seus aspectos. Um instrumento de avaliação estrutural é implementado na segunda parte do algoritmo. As demais implicações do problema podem ser avaliadas pelos critérios citados acima. Contudo, uma das grandes virtudes dos modelos baseados em parâmetros e restrições é a velocidade com que se produzem variações

na forma. Utilizando ferramentas de análise de desempenho é possível realizar sucessivas simulações, alterando a configuração das variáveis do SGP. Os resultados de cada uma destas simulações podem contribuir com a experiência do projetista para que as flexões seguintes melhorem os desempenhos investigados.

O cálculo e a análise de estruturas aparecem na ilustração do modelo teórico (Figura 1.1) pertencendo zona de interseção entre a Física e a Geometria. Com particular ênfase na Grafostática, descrita por Baumgart (2000) como:

(...) um quase esquecido método gráfico intuitivo para a resolução de problemas mecânicos (...) Era o método Padrão (...) usado por engenheiros civis pelo menos até os anos 1940. Foi Gradualmente substituído por alternativas analíticas, principalmente por métodos baseados na Tecnologia da Computação.⁴

O aspecto ”quase esquecido” será contraposto no desenvolvimento desta dissertação. Tendo em vista que, pelo aperfeiçoamento da própria Tecnologia da Computação, na medida em que esta ganhou proficiência na manipulação e exibição de formas geométricas (Computação Gráfica), autores e centros de pesquisa se interessaram pela Grafostática, dentre outros motivos pela natureza intuitiva do método.

O método de otimização proposto, nada mais é do que a aplicação de sucessivas flexões ao modelo geométrico, comparando os resultados às suas correspondentes análises Grafostáticas, aos aspectos estéticos das formas resultantes e a anteviçao das demais características técnicas, funcionais, construtivas que a equipe de projeto, por seu estudo e experiência seja capaz de enxergar.

Para um problema real de construção pode-se citar uma série de variáveis cuja quantificação é extremamente complexa (aspectos estéticos, a cultura construtiva, qualificação da mão-de-obra local, etc). O método tradicional de projeto, onde de uma proposição geral de uma forma passa por uma etapa de questionamentos, alterações e descartes, divide-se em elementos menores que passam pelo mesmo processo, até que as soluções de projeto sejam eleitas, oscila seus resultados entre sucessos e fracassos. A competência de uma equipe pode ser entendida, ainda que de maneira extremamente simplista, como a relação entre as ocorrências de cada extremo desta oscilação ao longo de sua atividade. As habilidades técnicas e a experiência da equipe são diretamente ligadas aos relativos êxitos e insucessos, seja no método tradicional ou qualquer outro.

⁴Graphical statics is an almost forgotten, intuitive drawing method for solving plane mechanical problems.”(...)”Graphical statics was the standard method for solving mechanical problems used by civil engineers at least up until the 1940s. It was gradually replaced by analytical methods, especially methods based on computer technology.

Os SGP são uma forma expressivamente mais rápida de se alterar as formas em estudo, possibilitando a experimentação de uma gama maior de variações de possíveis soluções, podendo recorrer diretamente à critérios de escolha utilizados pelo método tradicional sobre uma quantidade maior de propostas.

1.1.1 Objeto de Estudo

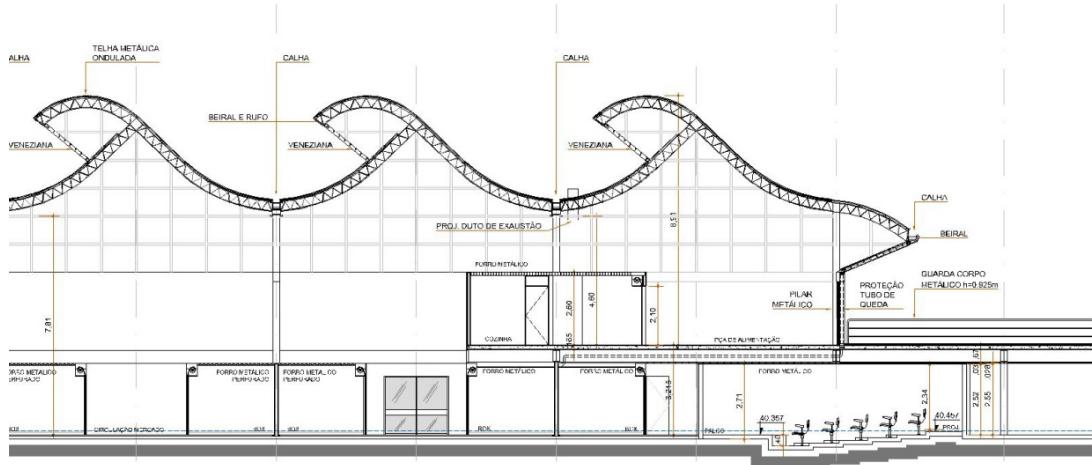


Figura 1.2: Proposta para o Mercado São Miguel - Autoria: João Filgueiras Lima e Instituto Habitat - Trecho do "Corte AA" - Fonte: Acervo do Instituto Habitat

O objeto de estudo eleito foi uma treliça de sustentação de coberturas do tipo ***shed*** com inspiração nas treliças metálicas recorrentemente apresentadas na obra do arquiteto João (Lelé) Filgueiras Lima. No exemplo mostrado na Figura 1.2 vê-se um trecho de um corte da proposta para a requalificação do Mercado São Miguel em Salvador - Ba.

O interesse pelos ***sheds*** vem destes serem, além de estruturas de suporte de coberturas, elementos de iluminação e ventilação capazes de melhorar substantivamente o conforto do ambiente para o uso humano, e assim diminuindo o consumo de energia com iluminação e climatização do espaço. Muito utilizados em projetos de grandes galpões e plantas industriais.

A inspiração na citada obra procura trazer o apuro estético que acompanham as soluções tecnicamente embasadas que marcam de forma singular o trabalho de um dos mais importantes arquitetos brasileiros, trazendo mais complexidade ao experimento.

Aplicando o Modelo Teórico e as partes do algoritmo ao objeto de estudo, o que se propõe é um SGP formado por um modelo baseado em parâmetros e restrições capaz de produzir alternativas de projeto para uma treliça de um ***shed***. Este modelo deve levar em conta que as variáveis devem sintetizar uma forma de desenho onde a criatividade e

outros aspectos subjetivos do projeto tenham espaço para emergir no processo ([SHEA; AISH; GOURTOVAIA, 2005](#)). Uma ferramenta de análise estrutural capaz de calcular as tensões atuantes em cada um dos elementos das diversas configurações possíveis do modelo geométrico, sendo baseada majoritariamente em mecanismos da Grafostática.

1.2 *Objetivos*

O objetivo geral deste trabalho é:

- Analisar a viabilidade prática da utilização dos SGP na atividade de projeto arquitônico.

Como Objetivos específicos temos:

- Implementar um SGP para atuar como ferramenta de auxílio no projeto da cobertura de um *shed*;
- Comprovar a eficácia dos SGP na elaboração de soluções de projeto onde aspectos subjetivos são parte integrante do problema;
- Repensar as virtudes da Grafostática como instrumento de análise estrutural, ressaltando seu aspecto intuitivo e prático.

1.3 *Importância da Pesquisa*

[Kolarevic \(2003\)](#) relata que o distanciamento entre o planejamento e a execução na **ICC** gradativamente transformou a Arquitetura em ”(...)uma profissão incerta quanto ao seu papel na sociedade contemporânea e sua economia, e uma profissão incapaz de responder aos desafios da Era da Informação.”⁵ Entretanto aponta, com bastante otimismo que:

Nos novos processos de produção assistidos por computador, projeto e construção não são mais domínios separados (...) Construtores e fabricantes se envolveram nas primeiras fases do projeto e os arquitetos participam ativamente

⁵*The outcome of this progressive disassociation of architecture from the rest of the building industry is a profession unsure of its role in contemporary society and it's economy, and a profession unable to respond to the challenges and opportunities of the Information Age.*

em todas as fases da construção.⁶

Florio (2012) constata que uma semelhante ”Fragmentação do ensino em disciplinas estanques, sem conexões umas com as outras, sempre gerou sérios problemas para a integração com os conteúdos no currículo.” E encontrou em atividades de modelagem paramétrica desenvolvidas entre estudantes de Arquitetura uma resposta animadora no encurtamento destas distâncias.

E inevitável imaginar o quanto as disrupturas sentidas na prática e no ensino estão relacionadas. A natureza interdisciplinar da aplicação dos SGP se evidencia em ambos os casos. Os benefícios apontados recaem não somente sobre a profissão de arquiteto, trabalha-se com a esperança de levar as virtudes da interdisciplinaridade à ICC como um todo.

O experimento desenvolvido aqui, embora calcado especialmente na etapa de projeto, procura trazer uma contribuição pontual no corrente debate.

Entre tantos desempenhos que poderiam ser escolhidas como objetivo da segunda parte do algoritmo, os argumentos favoráveis à estrutura partem da base do que chamamos de Arquitetura. Vitrúvio, autor do mais antigo tratado arquitetônico conhecido pela História (10 volumes, aprox. 27 a 16 A.C.), embasou a teoria e prática da atividade em 3 pilares:

- Firmitas
- Utilitas
- Venustas

O terceiro pilar refere-se à estética da obra. O segundo sobre o uso. A atividade fim para qual a construção foi dedicada, bem como toda e qualquer apropriação humana que ali ocorra. O primeiro está ligado à propriedade do edifício de permanecer inteiro. É no espaço formado por elementos capazes de se portar que qualquer uso pode ocorrer e qualquer teoria estética sobre a Arquitetura pode ser levantada. No caso dos *sheds*, o benefício que pode ser alcançado, por exemplo, pela exaustão do ”ar quente” pela abertura superior, precisa em primeiro lugar que a solução estrutural seja possível.

A Grafostática aparece como uma ferramenta proposta para o entendimento dos problemas estruturais. Gerhardt, Kurrer e Pichler (2003) defendem o valor dos métodos gráficos no ensino da estrutura para engenheiros porque estes permitem que se experimente visualmente as inter-relações entre, por exemplo, a geometria da estrutura e as forças atuantes.

⁶In the digitally-driven processes of production, design and construction are no longer separate realms but are instead , fluidly amalgamatedBuilders and fabricators become involved in the earlies phases of constructions and architects aactively participate in all phases of construction.

Relatam também a aplicação do método para um estudo de reforço estrutural visando atender uma mudança de uso para um dos espaços do prédio da prefeitura de Berlim. Aplicou-se simplesmente um programa CAD e o conhecimento do método. Os métodos analíticos relacionam expressões matemáticas à formas geométricas, as vantagens cognitivas de se aplicarem formas geométricas para avaliar a solidez de outras formas geométricas são defendidas pelos autores. A interseção de duas retas tem por equivalente analítico a resolução de um sistema de duas equações com duas incógnitas. O CAD funciona como uma calculadora geométrica extremamente interessante para a tarefa: Além de fazer parte do cotidiano profissional dos projetistas, promove uma ligação entre diferentes disciplinas ligadas à construção.

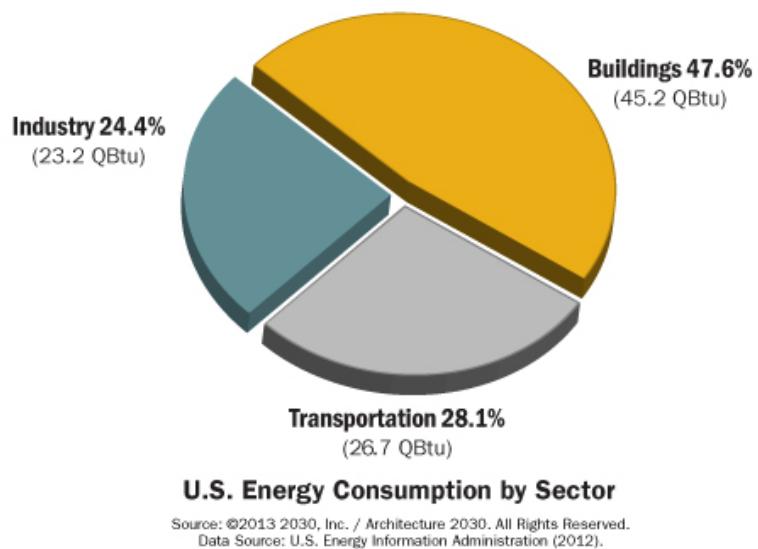


Figura 1.3: Consumo de Energia - Gráfico produzido pela ONG Architecture 2030. Com base nos dados coletados pela agência U.S. Energy Information Administration (EIA) referentes ao ano de 2012

A relevância do estudo e aplicação dos métodos de otimização na ICC está diretamente ligada ao crescimento da força que as preocupações ecológicas e de conservação de energia ganharam. O papel a ser desempenhado pela Indústria da Construção neste cenário não é pequeno. A Figura 1.3 apresenta um gráfico elaborado pela ONG *Architecture 2030* a partir de dados coletados pela agência *U.S. Energy Information Administration* (EIA) sobre o consumo de energia nos EUA dividindo-o em três setores: a indústria, os transportes e os edifícios. A esse último setor cabe a maior das parcelas, acumulando quase a metade da energia consumida no país (47,6%). A contribuição que a otimização por SGP pode oferecer é defendida por Krish (2011):

Os SGP, por outro lado, tem a capacidade de operar nas fases conceituais do projeto, quando a este ainda está sendo formulado. A habilidade de explorar variações nestes primeiros estágios pode produzir benefícios muito maiores que

a otimização realizada entre limites estreitos nas fases finais do processso.⁷

1.4 Limites e Limitações

Uma série de questões deve ser levada em conta na implementação do experimento proposto neste trabalho de pesquisa. Quanto a análise de estruturas vale dizer que, salvo poucas exceções (e.g. cálculo de estruturas pelo método dos elementos finitos planos e espaciais), trabalha-se com simplificações das formas onde linhas de eixo representam elementos tridimensionais a serem dimensionados. O que já representa uma simplificação, apesar de estar na base da teoria das estruturas.

Uma série de outras simplificações foram tomadas quanto as estruturas analisadas pelo trabalho que serão apresentadas a medida que o experimento é descrito. É importante entender contudo que a exequibilidade de uma estrutura depende tanto da sua análise e dimensionamento quanto do seu detalhamento. Uma simples ligação entre elementos construtivos mal estudada, um erro de soldagem pode levar ao colapso do conjunto.

Os SGP são em si um execício de limites e possibilidades. É desejável que os algoritmos sejam capazes de produzir uma gama de soluções factíveis. A natureza de como eles são construídos, baseando-se em parâmetros e restrições, por si já subentende-se que cada algoritmo trabalha dentro de suas próprias limitações. Cria-se um território de fronteiras demarcadas mas que deve ser vasto o suficiente para que aconteça uma satisfatória investigação do que nele pode estar contido.

Embora o pressuposto da pesquisa seja de propor um método de trabalho aplicável em projetos reais, onde estruturas factíveis possam ser propostas e modificadas, dentro de uma estratégia compatível com os recursos e prazos usuais em um escritório de projetos, a pesquisa não trata de um projeto específico. Isso explicita que, no desenvolvimento do experimento, não se conhece o contexto em que uma solução da treliça vai ser escolhida. Sem o programa da edificação e o uso destinado, o vão entre apoios não pode ser determinado. Sem as informações sobre o clima do local de implantação do projeto, a orientação, o regime de chuvas, etc; a necessidade de maior ou menor exaustão não pode ser analisada. A relação entre estética e custo também precisaria de mais dados para nortear uma investigação da forma. O que se apresenta no trabalho é um método de projeto suficientemente versátil para se adaptar as demandas de diversas edificações que admitam algumas das treliças contidas nas flexões do SGP como soluções de cobertura.

⁷ *Generative design on the other hand, can operate at the conceptual stages of design, where the design is still under formulation. The ability to explore design variations at the early stages of design can produce far more beneficial results, than optimizing it within narrow means at the final stages of design.*

Ao tratar de um SGP que procura valorizar os atributos estéticos do desenho, faz-se necessário ressaltar que a Filosofia das Artes é um campo de estudo específico e vasto. Qualquer investigação neste campo demanda um investimento de tempo, conceitualização e literatura específica que ultrapassam o escopo da presente pesquisa.

1.5 Questões e Hipóteses

Como os SGP podem contribuir nas fases iniciais da concepção das formas de um projeto arquitetônico?

Esta é a principal questão investigada por esta pesquisa. Quanto às hipóteses, imagina-se que é possível desenvolver um SGP que sirva como uma ferramenta à ser usada por equipes de Arquitetura que seja:

- de interação direta com o desenho, o traço e os aspectos estéticos da criação;
- forneça subsídio para um julgamento técnico da forma que emerge do processo de projeto;
- apresente uma forma prática de alterar, estudar e eleger uma solução.

Investiga-se também a hipótese de que a implementação do algoritmo de análise baseado na Grafostática apresenta mecanismos semelhantes e compatíveis com os utilizados na construção de um SGP, e apresenta vantagens, do ponto de vista cognitivo, em relação aos seus análogos analíticos.

1.6 Aspectos Metodológicos

Lançando mão de uma metodologia experimental, a organização geral do algoritmo baseou-se no fluxograma proposto por [Bohnacker, Gross e Laub \(2012\)](#), mostrado na Figura 1.4. Pode-se ver no diagrama as etapas de concepção e funcionamento de um SGP além de ilustrar a maneira como a otimização ocorre, sempre através do julgamentos do projetista (*Designer*).

A validação dos resultados segue critérios específicos para cada uma das partes do algoritmo apresentadas na Seção 1.1:

No caso do modelo geométrico baseado em parâmetros e restrições, vistas as limitações quanto a apreciação dos aspectos estéticos, serão avaliadas as capacidades de produzir

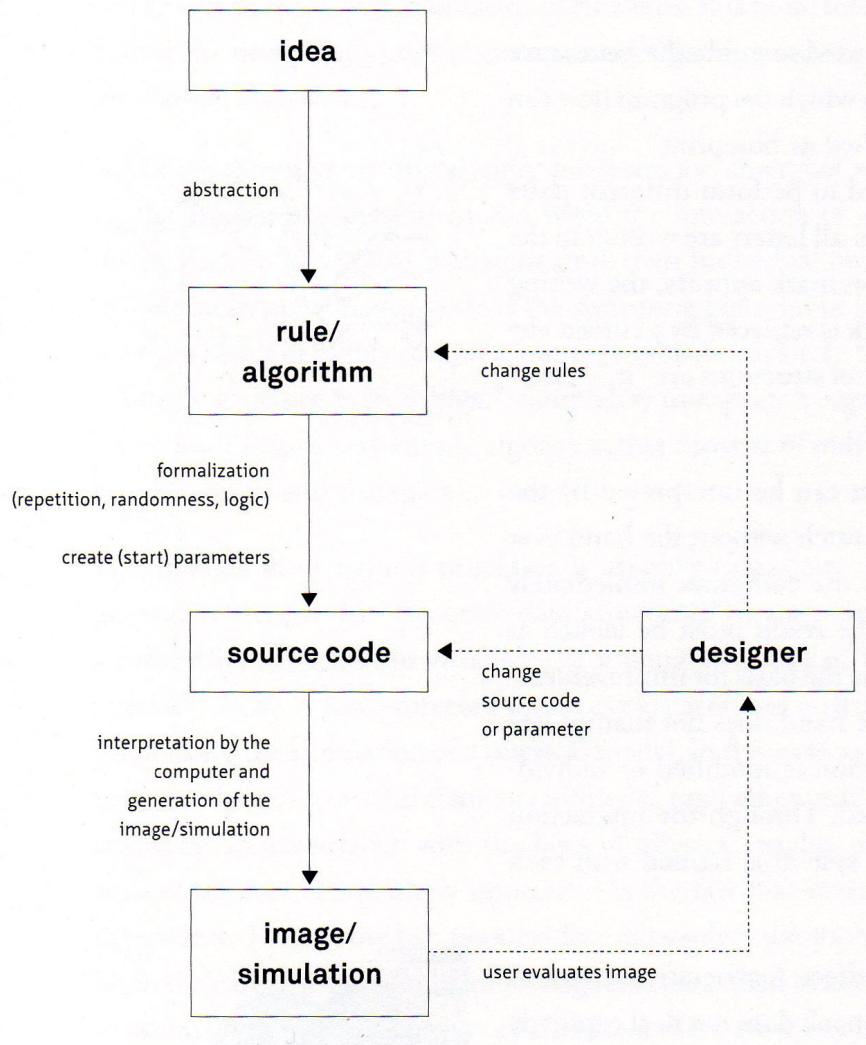


Figura 1.4: Fluxograma de um SGP - fonte: [Bohnacker, Gross e Laub \(2012\)](#)

variações. A facilidade e velocidade com que essas transformações ocorrem. A praticidade da utilização dessas ferramentas no cotidiano do projetista.

O algoritmos de análise Grafostática deve ser construído de maneira coerente com os métodos desta, bem como da Mecânica dos Sólidos e Resistência dos Materiais. Os resultados obtidos serão validados pela coerência dos gráficos gerado pelo algoritmo com a teoria e regras da Grafostática, técnica de cálculo reconhecida e matematicamente provada pelos tratados de [Cremona \(1890\)](#).

O método de otimização passa, inevitavelmente, pelo julgamento do projetista. A relação entre plástica e economia assume diferentes pesos e preferências de acordo com às opiniões de cada projetista e a finalidade de cada projeto. Para que esse julgamento seja feito é preciso assegurar que as informações de forma e análise estrutural sejam coerentes, que

a rápida alteração de uma implique numa também veloz e coerente alteração de outra, e que o conjunto opere sem conflitos entre as partes.

1.7 *Organização da Dissertação de Mestrado*

Este documento apresenta 6 capítulos e está estruturado da seguinte forma:

- **Capítulo 1 - Introdução:** contextualiza o âmbito, no qual a pesquisa proposta está inserida. Apresenta, portanto, a definição do problema, objetivos e justificativas da pesquisa e como esta dissertação de mestrado está estruturada;
- **Capítulo 2 - Sistemas Generativos de Projeto:** apresenta o conceito de algoritmo, investiga o significado dos algoritmos generativos e apresenta os principais modelos dos AGs aplicados ao projeto arquitetônico;
- **Capítulo 3 - Grafostática:** apresenta os fundamentos do método, os procedimentos de cálculo de treliças utilizados na implementação do algoritmo de análise estrutural, as pesquisas em que este vem sendo atualmente aplicado, os *softwares* que utilizam, e um breve levantamento da pesquisa bibliográfica, citando alguns algoritmos apresentados nos manuais práticos da Estática Gráfica;
- **Capítulo 4 - ambiente de Testes do Algoritmo:** uma rápida apresentação do ambiente computacional onde o experimento foi realizado;
- **Capítulo 5 - Implementação do Experimento:** apresenta como os algoritmos que formam o SGP foram criados, como funcionam, como se estabelece a comunicação entre eles, e como podem ser utilizados na exploração das possíveis soluções de projeto;
- **Capítulo 6 - Conclusões:** apresenta as conclusões, contribuições e algumas sugestões de atividades de pesquisa a serem desenvolvidas no futuro.

Algoritmos Generativos e Sistemas Generativos de Projeto

*"A man provided with paper,
pencil, and rubber,
and subject to strict discipline,
is in effect a universal machine."*
 (Alan Turing)

Os componentes essenciais de um SGP são: os algoritmos generativos e o ambiente computacional onde estes serão executados. Envolvendo os sistemas operacionais, os programas, as linguagens de programação, os computadores e demais dispositivos empregados (*hardware*) e as interfaces com usuários, os ambientes de implementação são corresponsáveis pelos limites e possibilidades entre os quais operam os SGP. No caso específico desta pesquisa, a praticidade de se gerar formas em um ambiente CAD utilizado no mercado de desenho de projetos, a capacidade de se implementar algoritmos de maneira rápida e eficiente, e a facilidade de ver os resultados das alterações dos parâmetros em tempo real, tiveram grande importância na escolha. Já os Algoritmos Generativos determinam principalmente a lógica que rege a criação de formas. As características básicas do ambiente computacional escolhido são apresentadas no Capítulo 4. As definições, conceitos e principais os modelos computacionais dos AG aplicados á Arquitetura formam o conteúdo deste capítulo.

2.1 Definição de Algoritmo e Algoritmos Generativos

A palavra ”algoritmo” tem sua origem no nome do matemático e astrônomo árabe Moham-med ibu-Musa al-Khowarizmi (século VII). A tradução latina de um de seus livros, sobre a ”arte hindu de calcular”, foi responsável por difundir o sistema de numeração que chamamos de hindu-arábicos na Europa. Embora não revindicasse a autoria do sistema, a ”nova” notação, começou a ser referida como a de ”al-Khowarizmi” ou ”algorismi” e posteriormente ”algorismo” ou ”algoritmo”. Com o passar do tempo a palavra ”algoritmo” passou a significar processo, regra, método (e.g. O algoritmo de Euclides para encontrar o máximo divisor comum de um número) (BOYER; GOMIDE, 1974, p. 166).

Knuth (1997, p. 1-8) define o conceito atual do termo como um conjunto finito de regras

que fornece uma sequência de operações para a solução de um problema específico, e que deve atender cinco requisitos:

- Finitude
- Precisão
- Entrada de Informação
- Saída de Informação
- Efetividade

Em consonância com os assuntos abordados por esse trabalho, na função de ilustrar o conceito, foi escolhido um exemplo de algoritmo proveniente da geometria e desenho técnico, em lugar de uma abordagem puramente numérica: o método de divisão de uma seguimento de reta \overline{AB} qualquer em uma quantidade finita de seguimentos iguais, representada pelo número inteiro N . O processo é realizado com êxito seguindo os cinco passos da "receita" descrita abaixo e representados graficamente na Figura 2.1:

1. Dado o segmento de reta \overline{AB} desenhar, a partir do ponto A, uma semi-reta não paralela à \overline{AB} .
2. Dado o número N de divisões, utilizando o compasso com uma mesma abertura qualquer, partindo do ponto A, marcar N pontos consecutivos e equidistantes sobre a semi-reta traçada no passo anterior.
3. Traçar um segmento unindo o "N-ésimo" ponto marcado no passo anterior com a extremidade B do segmento \overline{AB} .
4. Traçar paralelas ao segmento desenhado no passo 3, passando pelos pontos definidos no passo 2 e interceptando o segmento de reta \overline{AB} .
5. FIM

Comparando o exemplo com a definição acima, nota-se que o método de divisão de segmentos tem um fim. A precisão diz respeito a clareza das instruções que, no caso citado, são suficientemente claras para o entendimento humano. Por entrada de informações temos o número inteiro de divisões e o objeto seguimento de reta \overline{AB} . Por saída podemos considerar os pontos que dividem o seguimento em N partes ou os N seguimentos de reta justapostos sobre o seguimento \overline{AB} . E a efetividade, a capacidade de lograr êxito na execução do seu propósito, pode ser aferida analisando a Figura 2.1(e) comparada com o

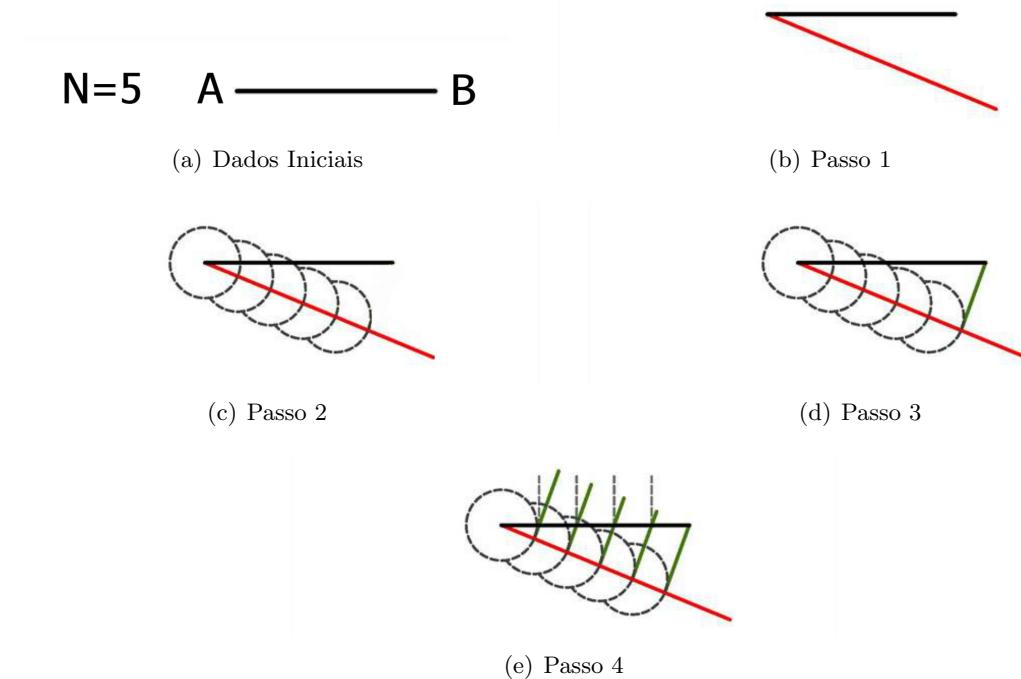


Figura 2.1: Exemplo de algoritmo - subdivisão de um seguimento em um número finito (5) de partes iguais

teorema descrito por Tales: um feixe de paralelas cortado por duas transversais quaisquer as divide em seguimentos proporcionais.

O pioneiro dos sistemas generativos de projeto aplicados à Arquitetura, William J. Mitchell (2008, p.61-64) afirma que todo o trabalho de desenho geométrico/técnico é fundamentado em algoritmos (e.g. os métodos descritos em "Os Elementos" de Euclides). O vocabulário algorítmico do desenho é absorvido pelo projetista e, pela combinação de uma gama de procedimentos, vão surgindo não só formas gradativamente mais elaboradas, como uma linguagem técnica do desenho de projetos. Palavras como perpendicular, paralela, ponto central, tangente, etc passam a resumir uma série de instruções e descrever algoritmos inteiros. Nos primórdios do projeto assistido por computador, esse era o papel dos programadores: A transposição dos algoritmos de desenho para o ambiente digital.

Os Algoritmos Generativos são um campo de estudo novo e, embora estejam sendo abraçados pelos grandes escritórios de Arquitetura do mundo e estejam sendo estudados e ensinados em muitos centros acadêmicos de Arquitetura (KRISH, 2011), ainda está a procura de uma definição clara e completa. A simples formulação de que são algoritmos voltados a geração de forma, cai por terra pelo simples fato de todo o desenho técnico ser construído a partir de algoritmos.

A simples transcrição dos algoritmos de desenho a ferramenta computacional também

não é suficiente para definir o termo. Em seguida, partiu-se para novos domínios como relatado no mesmo texto de [Mitchell](#):

(...) Com o apoio das habilidades de programação mais do que inventividade mecânica, programadores gráficos primeiro replicaram as funções de instrumentos de desenho tradicional, depois foram muito além. Isso disponibilizou para os projetistas um ainda maior vocabulário gráfico junto com uma sintaxe mais elaborada - em suma, uma mais rica e potencialmente mais expressiva linguagem gráfica e espacial.¹

[Terzidis \(2006, p. 37-64\)](#) classifica os problemas de projeto como: passíveis de solução (*problem-solving*), referindo-se ao conceito tradicional de encontrar um caminho entre os pontos *A* e *B*, ou de abordagem (*problem-adressing*). Problemas como a forma inicial, estética ou planejamento não tem uma solução definitiva, mas estratégias de trabalho que podem atingir resultados mais ou menos satisfatórios. Para o autor, projetar é uma forma de pensar. É uma atividade intrinsecamente atada as características mais existencialmente humanas como a lógica, criatividade e identidade. E acaba por conferir à certos algoritmos a capacidade de explorar além, em paralelo ou no lugar de maneiras tradicionais e estabelecidas de se pensar o projeto.

[DIno \(2012\)](#) elenca quatro características presentes nos AG: Os parâmetros de entrada, os mecanismos generativos (regras e algoritmos), a geração de variações de formas (saída), e a escolha da melhor variação. Os dois primeiros requisitos estão presentes de maneira idêntica na definição de algoritmo, os outros estão mais ligados à forma de aplicação das ferramentas. Gerar uma saída de dados processados é também uma característica dos algoritmos. Entradas diferentes produzem saídas variantes em qualquer problema, trabalhar com variações e escolher entre estas, como é pontuado nos dois últimos requisitos, fazendo com que as propriedades de um AG diferenciem-se das características básicas de um algoritmo apenas por uma maior especificidade no que diz respeito ao modo de utilização da ferramenta.

Em suma, os autores concordam que, além da geração de formas, os Algoritmos Generativos possibilitam a aplicação de uma metodologia diferente de trabalho. Parte importante do que caracteriza um AG é a possibilidade de se escolher uma solução de projeto a partir da variabilidade gerada pelos parâmetros e regras. O desafio teórico, que esta pesquisa não encontrou resposta definitiva, é formular uma definição mais precisa, que abarque as referidas novas formas de se trabalhar e eleger uma solução de projeto.

¹ *Relying upon software skills rather than mechanical ingenuity, graphics programmers first replicated the functions of traditional drafting instruments, and then went far beyond them. This has made a wider graphic vocabulary available to designers, together with a more elaborate syntax?in all, a richer and potentially more expressive graphic and spatial language.*

2.2 *Principais Modelos de Algoritmos Generativos Aplicados à Arquitetura e ICC*

Os AG são sim algoritmos capazes de gerar formas, mas também de trazer algo novo para esse antigo jogo. Procurando revelar mais sobre a natureza destes métodos, uma lista dos principais tipos de AGs e algumas de suas aplicações na Arquitetura aparece como o caminho eficaz. Algumas revisões de literatura serviram como base para a elaboração desta lista ([Fasoulaki \(2008\)](#); [Santos \(2009\)](#); [Daly \(2009\)](#); [Singh e Gu \(2012\)](#)), bem como artigos fundamentais e atuais sobre os diversos modelos utilizados. Aplicações dos principais algoritmos foram buscados tanto nos artigos científicos quanto em projetos construídos e propostas.

Curiosamente a maioria desses métodos tem relação direta com aspectos da natureza, a fonte de inspiração dos primeiros construtores, dotados apenas da geometria que estavam formulando e de instrumentos rudimentares de desenho e construção, e repetidamente interpretada por todas as gerações de projetistas. No "Ensaio sobre a Arquitetura" de 1753, o Abade Laugier revisita o conceito fundamental da "cabana primitiva", quatro pilares de trocos de árvore suportando uma cobertura rústica de duas águas. Uma apologia à simplificação das formas. Segundo [Frampton \(2008, p. 5\)](#), foi um ponto crucial na sucessão de fatos que nos levam ao lema "ornamento é crime" do arquiteto Adolf Loos no início do século XX. Um elo na cadeia de eventos que transformou as formas detalhadamente esculpidas de colunas clássicas em sólidos platônicos como o cilindro e o prisma. Porém o que grande parte dos métodos aqui apresentados explora são justamente os elementos que o platonismo, a Geometria, as ciências e a Arquitetura tinham deixado de lado, e que passaram a ser discutidos cientificamente pelos estudos relacionados com o fim do modelo mecanicista do universo (Sistemas Complexos, Teoria do Caos Determinístico, etc).

2.2.1 Gramáticas de Forma (GF)

Introduzido por George Stiny e James Gips em 1971 ([STINY; GIPS, 1972](#)) nos campos da Pintura e Escultura e seguidamente aplicado à Arquitetura por [Stiny e Mitchell \(1978\)](#), é considerado o primeiro Sistema Generativo aplicado ao projeto. Os quatro elementos básicos de uma gramática de formas são:

- um conjunto finito de formas;
- um conjunto finito de símbolos;
- um conjunto finito de regras de formas;

- uma forma inicial.

Partindo da forma inicial, as soluções são geradas pela modificação (adição, subtração, substituição) de elementos (membros do conjunto finito de formas) obedecendo as regras presentes no conjunto finito de regras de formas ([STINY, 2008](#)).

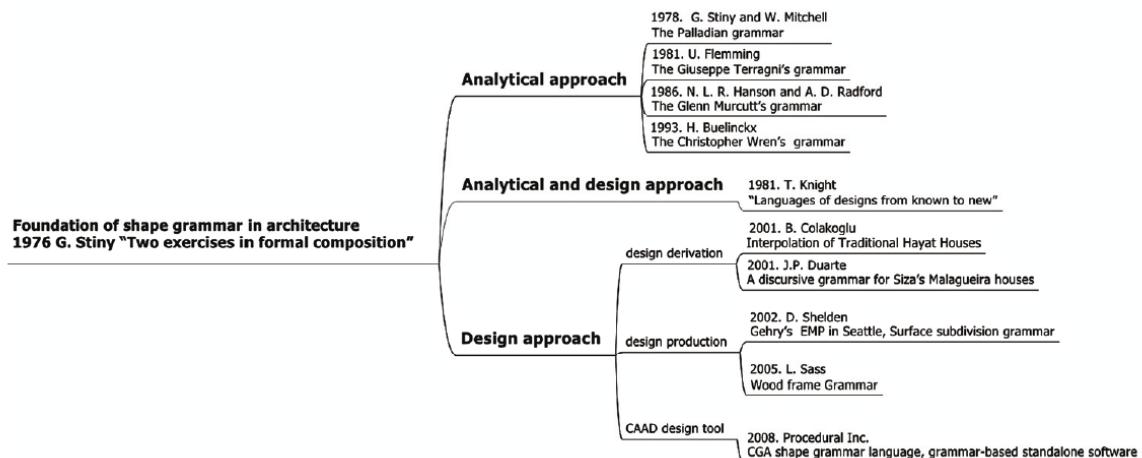


Figura 2.2: Árvore Evolutiva das Gramáticas de Forma

A Figura 2.2 ([TEPAVCEVIC; STOJAKOVIC, 2012](#)) mostra uma árvore evolutiva das GF onde destacam-se as implementações capazes de simular "estilos" de arquitetos notáveis (Palladio, Terragni, Siza...). No caso da GF baseada na Casa Malagueira do arquiteto português Álvaro Siza, [Duarte \(2005\)](#) relata que o próprio autor teve dificuldade de distinguir entre o projeto de sua autoria e algumas das simulações a ele apresentadas.

2.2.2 *L-Systems (LS)*

Os **Lindenmayer Systems**, ou simplesmente **L-Systems**, foram concebidos pelo botânico Aristid Lindenmayer como um modelo matemático do crescimento de plantas. O conceito inicial, embora não tivesse ainda uma representação geométrica, rapidamente evoluiu tornando-se, não só um versátil modelador de plantas e estruturas orgânicas, como uma ferramenta rápida para visualização de fractais ([PRUSINKIEWICZ; LINDENMAYER, 1997](#)). Os LS possuem 3 componentes:

- um conjunto finito de símbolos;
- um conjunto finito de regras de reescrita;
- uma configuração inicial dos símbolos.

Como exemplo podemos ter um conjunto de símbolos composto pelas letras "A" e "B", uma configuração inicial apenas com o símbolo "A" e duas regras de escrita:

- todo símbolo "A" deve ser reescrito como "ABA";
- todo símbolo "B" deve ser reescrito como "BBB".

Partindo da configuração inicial e operando por três gerações geramos:

```

      A
      ABA
      ABABBBABA
      ABABBBABABBBBBBBBBBABABBBABA
    
```

Para gerar gráficos a partir desta ferramenta basta que se associe formas a cada um dos símbolos. Se no exemplo acima o "A" representasse uma linha horizontal ininterrupta e o "B" uma interrupção na linha, este LS desenharia o fractal "Conjunto de Cantor" (Figura 2.3(a)).

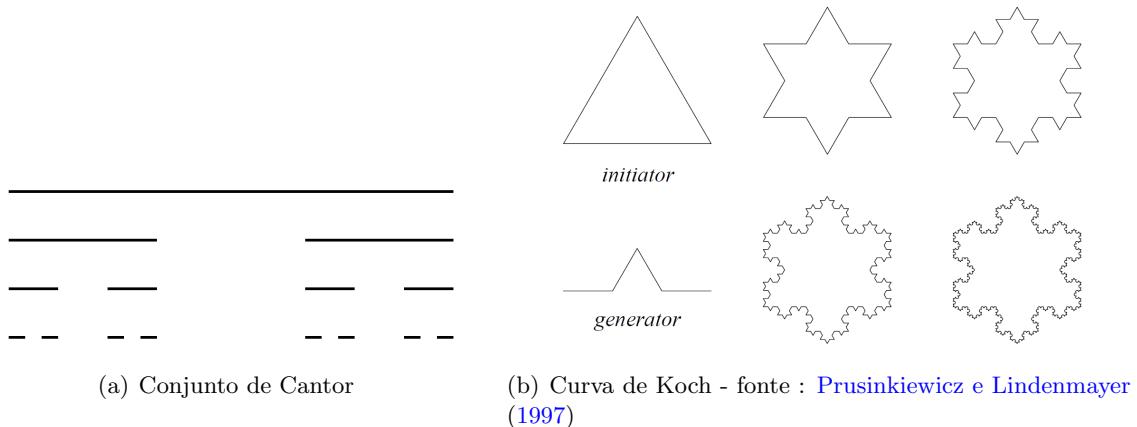


Figura 2.3: Exemplos de aplicações dos L-Systems

Embora os LS sejam definidos como um método de escrita, através de símbolos que representam as formas, uma abordagem gráfica destes modelos foi também exposta no livro "The Algorithmic Beauty of Plants" ([PRUSINKIEWICZ; LINDENMAYER, 1997](#)) coescrito pelo autor da proposta original. A Figura 2.3(b) mostra aa configuração inicial (*initiator*) onde cada linha do triângulo é substituída pelo elemento "*generator*" (que corresponde graficamente à regra de reescrita) na primeira geração. O mesmo acontece nas quatro gerações seguintes, cada linha é redesenhada, mantendo o tamanho e colocando o "*generator*" em escala proporcional ao da linha que este substitui. estas operações resultam no fractal conhecido como Curva de Koch.

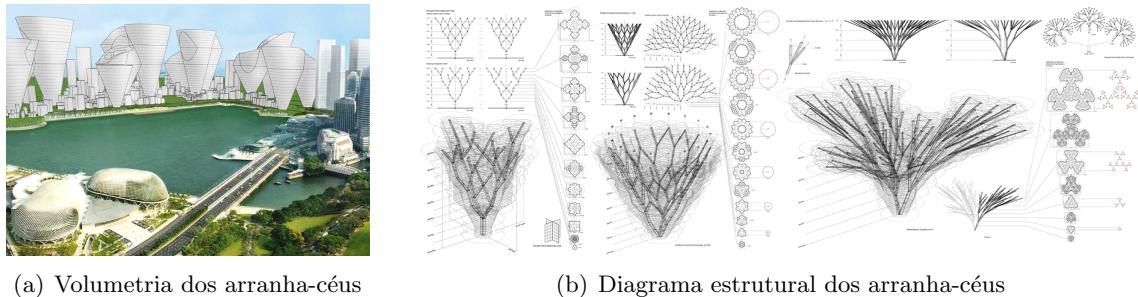


Figura 2.4: Inverted Skyscraper Typology - fonte: [EVolo \(2009\)](#)

Müller et al. (2006) propõem uma estratégia de modelagem procedural de edifícios baseado nas Gramáticas de Chomsky e em L-Systems, objetivando ampla variação das formas e grande nível de detalhe, aplicado a grandes áreas. Daly (2009) cita como aplicação dos LS na Arquitetura a proposta do arquiteto Yi Cheng Pan, premiada com o segundo lugar na edição de 2007 do concurso de arranha-céus promovido pelo periódico de Arquitetura eVolo. A ideia consiste na criação de grandes edifícios que ocupem uma área diminuta em sua base, liberando o terreno como espaço público, e que gradativamente, a medida em que se distancia do chão, a área útil de cada pavimento possa aumentar (Figura 2.4(a)). O diagrama estrutural dos prédios, representado na Figura 2.4(b), baseia-se nas formas encontradas nas ramificações dos galhos de árvores. A aplicação dos LS se justifica neste caso pela eficiência do método na simulação do crescimento de plantas.

2.2.3 Autômatos Celulares (AC)

Os autômatos celulares são um poderoso e simples modelo matemático/computacional com aplicações importantes em diversas áreas do conhecimento (e.g. Física, Biologia, Tecnologia, Artes, Ciências Humanas) ([MASUM, 2002](#)). A teoria dos autômatos celulares tem seu início em meados do século XX, nos trabalhos de von Neumann e Morgenstern. O que se buscava era uma formalização matemática e lógica, capaz de contribuir para o entendimento tanto dos sistemas naturais, quanto da computação. Podem ser definidos como uma regra evolutiva fixa para um grupo de células de uma matriz ou grade, onde o tempo é discreto e o estado de cada célula é alterado dependendo apenas dos estados anteriores (geralmente apenas o imediatamente anterior), dos seus vizinhos e dela própria ([PEREIRA; ZEBENDE; MORET, 2010](#)). Posteriormente os fundamentos desta teoria foram estruturados por Wolfram, tendo por base os autômatos celulares elementares.

Os autômatos celulares elementares consistem em matrizes unidimensionais onde as células podem apresentar apenas dois estados (0 ou 1, "preto" ou "branco"). O estado de cada célula depende apenas do estado anterior dela própria e das vizinhas. Levando em conta

que cada célula pode apresentar apenas 2 estados, existem oito ($2 \times 2 \times 2$) combinações possíveis para o estado de três casas consecutivas. Cada uma destas combinações determinará um dos dois valores possíveis para a célula central, o que totaliza 256 regras ($2^8 = 256$).



Figura 2.5: Ilustração da regra 90 dos AC elementares

A Figura 2.5 ilustra a regra 90 dos autômatos celulares elementares. Na linha superior temos as oito combinações possíveis, e na linha inferior os resultados de cada combinação para a regra 90. Na Figura 2.6 cada linha representa uma passagem de tempo onde o valor de cada célula é alterado (ou mantido) segundo a regra 90. Partindo de um único ponto com o valor "preto", e colocando cada uma das passagens de tempo como se vê na Figura 2.6, o desenho formado é uma imagem semelhante ao fractal "Triangulo de Sierpinski".

Partindo-se da primeira linha da Figura 2.6, procura-se determinar qual valor cada uma das células deve assumir na linha seguinte pela aplicação da regra descrita na Figura 2.5. O procedimento consiste em, para cada célula, considerar o valor dela própria e de suas duas vizinhas imediatas (esquerda e direita), na ordem que elas aparecem na linha (vizinha à esquerda, célula a ser reescrita, vizinha à direita). Sempre que as três células tiverem o valor "branco", aplica-se a oitava posição da Figura 2.5 e escreve-se "branco" no estado subsequente da célula em questão. Ao analisar a célula imediatamente à esquerda do elemento "preto" central da primeira linha da Figura 2.6, aplica-se a sétima posição da regra: para a sequência de valores "branco", "branco", "preto", o estado seguinte da célula analisada será "preto". Para a célula central da primeira linha aplica-se a sexta posição da regra 90 e o novo estado da célula central é "branco". Para definir o novo estado da célula imediatamente à direita do centro aplica-se a quarta posição da regra e o valor "preto" é encontrado. As linhas sucessivas da Figura 2.6 são determinadas da mesma maneira: observa-se o valor da vizinha à esquerda, da própria célula e a da imediatamente à direita, procura-se na regra a mesma sequência de valores, e utiliza-se o valor de novo estado para essa ordem descrito na regra.

[Herr e Kvan \(2007\)](#) apresentam uma adaptação dos AC para proporcionar maior interação do projetista com o processo, ao invés de deixar o desenho ser unicamente guiado pelas regras de reescrita primariamente definidas. O emprego do AC aparece como um "movimento" opcional de projeto fazendo uso da sua capacidade de gerar variedade formal. Durante o processo é oferecido ao usuário a premissa de aceitar as mudanças propostas pelas regaras e/ou intervir no modelo para direcionar a solução do passo seguinte. Um experimento de redesenho de uma proposta para um conjunto de prédios proposto para a

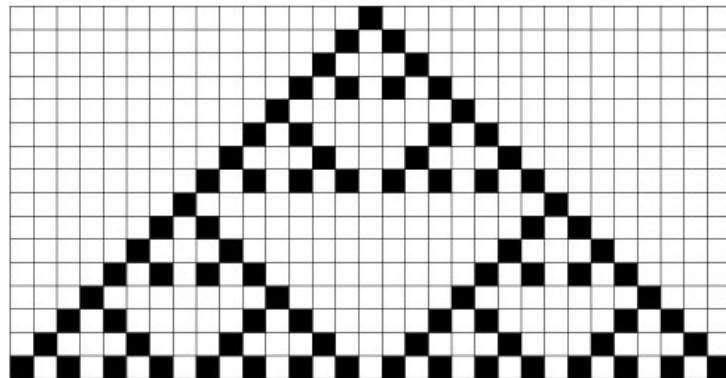


Figura 2.6: Resultados obtidos a partir de sucessivas aplicações da regra 90 dos AC elementares - Triangulo de Sierpinski

cidade de Aomori, no norte do Japão, pelo grupo de arquitetos espanhóis Cero9 em 2001 (Figura 2.7(a)). Um projeto inspirado em CA onde a inclinação de cada andar é alterada em função da inclinação dos andares adjacentes, com os volumes gerados pela adaptação da mesma proposta com o auxílio do algoritmo descrito no experimento (Figura 2.7(b)) obtendo um jogo volumétrico bastante semelhante.

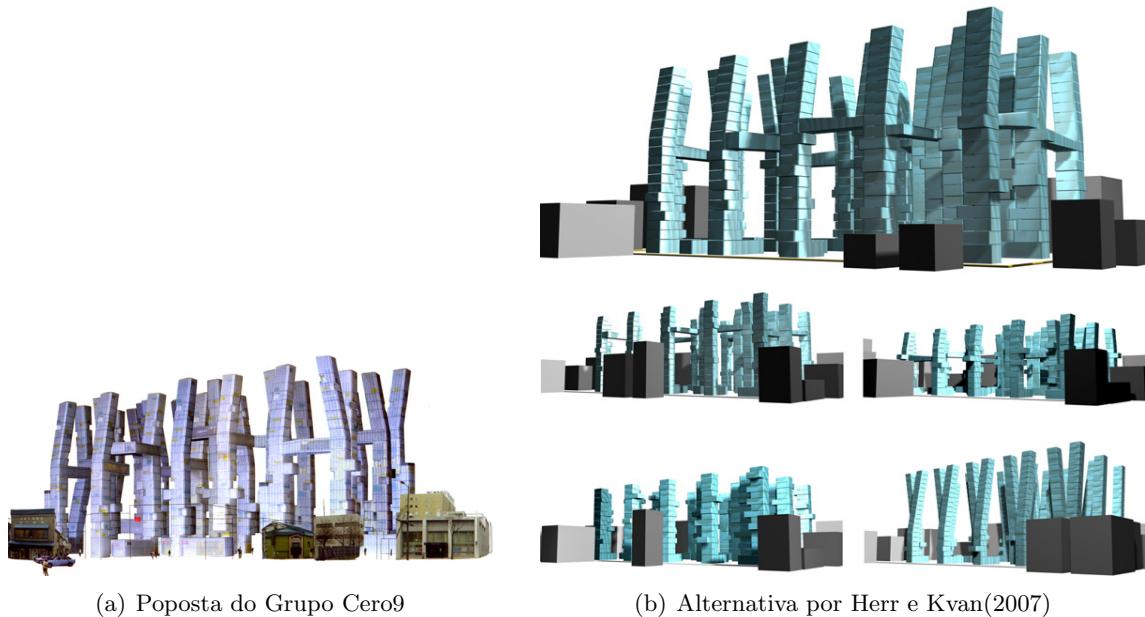


Figura 2.7: Duas propostas para o conjunto de prédios em Aomori/japão - fonte: Herr e Kvan(2007)

2.2.4 Swarm Intelligence (SI)

Swarm Intelligence (inteligência de enxame) é um conjunto de modelos de inteligência artificial baseado no comportamento de coletivos descentralizados e auto-organizados.

Lançando mão de Agentes (elementos de programação capazes de atuar com autonomia e de acordo com seus próprios critérios) que interagem, competem e colaboram e assim levam o sistema global ao objetivo almejado ([SINGH; GU, 2012](#)).

Os algoritmos de SI são comumente nomeados e classificados em referência aos seres ou coletivo que os inspirou, entre eles temos:

- Colônia de Formigas (*Ant Colony*);
- Colônia de Abelhas (*Artificial Bee Colony*);
- Algoritmos das Abelhas (*The Bees Algorithm*);
- Enxame de Partículas (*Particle Swarm*).

As virtudes dos SI nos campos da organização do trabalho, divisão de tarefas e cronogramas, são investigadas também quando esses campos são usados pela Indústria da Construção Civil. A atividade fim da ICC não se passa em um ambiente projetado e planejado como uma planta industrial convencional, mas no terreno escolhido e servido pelo entorno existente (densamente povoado, urbano, rural, etc), o que acarreta em um grande grau de incerteza quanto a qual seria o melhor planejamento para cada caso. [Zhang et al. \(2005\)](#) propõem a aplicação do modelo baseado em "Enxame de Partículas" para a definição do cronograma de trabalho em atividades de construção com limitações no acesso aos recursos necessários. Na mesma linha, porém com um rebatimento formal, [Yahya e Saka \(2014\)](#) fazem uso de um algoritmo "Colônia de Abelhas" para o planejamento dos espaços de um canteiro de obra. Na geração de formas a serem construídas, o trabalho de [Luh, Lin e Lin \(2011\)](#) apresenta um algoritmo para otimização topológica de estruturas, baseando-se em um *Particle Swarm* binário. [Aranda, Lasch e Balmond \(2007, p. 62-71\)](#) discutem estratégias de geração de formas pela aplicação de algoritmos inspirados nos coletivos dos pássaros (*Flock of Birds*).

2.2.5 Diagramas de Voronoi (DV)

Definidos de maneira simples e intuitiva: Para um conjunto finito de pontos isolados em um espaço contínuo (\mathbb{D}^n), associa-se todos os pontos do espaço ao elemento mais próximo do conjunto de pontos ([OKABE; BOOTS; SUGIHARA, 1992, p. 1](#)). Na Figura [2.8\(a\)](#) vemos um DV bi-dimensional com polígonos demarcando a região do espaço em que qualquer dos pontos internos ao perímetro estão mais próximos do ponto gerador contido no mesmo polígono do que de qualquer outro. Os lados do polígono (\mathbb{D}^{n-1}) estão equidistante a dois geradores e os vértices (\mathbb{D}^{n-2}) equidistantes a três pontos. Nos

Diagramas 3D, Figura 2.8(b), as faces dos sólidos (\mathbb{D}^{n-1}) representam a região equidistante à dois, as arestas (\mathbb{D}^{n-2}) à três e os vértices (\mathbb{D}^{n-3}) à quatro pontos geradores.

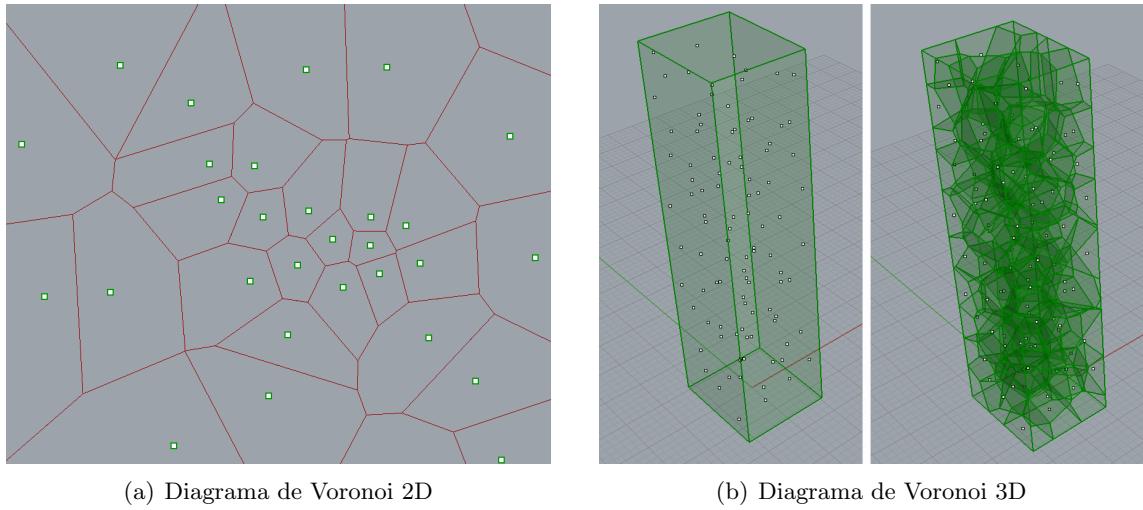


Figura 2.8: Gerados no *Software Rhinoceros 3D*

A descoberta dos DV (também chamados de Tesselagem de Dirichlet) é frequentemente atribuída à Dirichlet (1850) e ao matemático Georgy Voronoi (1907) embora tenham sido tratados por Descartes em um estudo sobre a fragmentação cósmica, publicado em 1644. Uma estrutura geométrica versátil com aplicações em Geografia, Física, Astronomia, Robótica, Biologia e muitas outras áreas (BERG et al., 2008, p. 147-161).

Essas malhas ganharam o interesse de arquitetos pelo seu apelo estético orgânico, complexo e, ao mesmo tempo, ordenado. O desenho começa pela escolha dos pontos geradores do diagrama, o algoritmo desenha a malha do Voronoi, avalia-se o resultado e intervêm na disposição dos pontos para direcionar o desenho. Sua aplicação na arquitetura pode ser observada no Centro de Esportes Aquáticos da olimpíada de Pequim (PWT Architects, Figura 2.9(a)), onde seu uso busca conferir sensação de fluidez ao involucro do prédio.



Figura 2.9: Exemplos de Diagramas de Voronoi na Arquitetura

O escritório californiano [Iwamoto-Scott Architecture](#) propôs a *Jellyfish House* (Figura 2.9(b)) para uma exibição de conceitos sobre o futuro próximo das *smart houses*. O fechamento externo apresenta ramificações orgânicas, desenhadas com a sobreposição de um DV e uma outra malha gerada pela triangulação de Delauney.

2.2.6 Algoritmos de Otimização Estocástica

Tudo que é calculado ou representado em computação; a altura de um prédio, O raio de um círculo, a inclinação de um telhado, a largura de uma rua, o calor do sol, a velocidade do vento, etc; é feito através de variáveis. A otimização é a procura dos valores ideais de variáveis, para que o cenário descrito por elas possa ser configurado da melhor maneira possível. Quando o problema é simples e bem definido (e.g. qual a menor quantidade em m^2 de paredes e laje de uma caixa-d'água para armazenar 5.000 litros de água?), os métodos determinísticos são suficientes para se encontrar com exatidão o resultado ideal. Quando o número de variáveis é grande e/ou a função que relaciona estas à grandeza que se pretende otimizar é muito complexa ou o que se entende por "melhor cenário possível" não é muito bem determinado, a otimização determinística não é suficiente para solucionar o problema e outros modelos ganham importância.

A otimização estocástica refere-se aos métodos de procura por uma solução ideal (em relação a um objetivo determinado) onde a aleatoriedade está envolvida de forma construtiva. Os principais algoritmos de otimização estocástica são ([FOUSKAKIS; DRAPER, 2002](#)):

- Algoritmos Genéticos
- *Simulated Annealing*
- *Tabu Search*

2.2.6.1 Algoritmos Genéticos

Os Algoritmos Genéticos são, dentre os métodos de otimização estocásticos, os mais aplicados na Arquitetura ([SINGH; GU, 2012](#)). Propostos por John Holland em 1975 ([LODDING, 2004](#)), estes algoritmos são modelos computacionais inspirados nos conceitos darwinianos de sobrevivência dos mais aptos ([CALDAS; NORFORD, 2002](#)). Operações de seleção e recombinações são aplicadas entre as soluções candidatas na procura dos que apresentam um melhor desempenho. A terminologia usada também faz analogia com a teoria da evolução das espécies: as soluções candidatas são chamadas de indivíduos, os

grupos de indivíduos existentes em cada estado é denominado de população, cada vez que uma nova população é criada, chama-se de geração. As principais operações realizados por um algoritmo genético são: a reprodução, o *crossover* e a mutação.

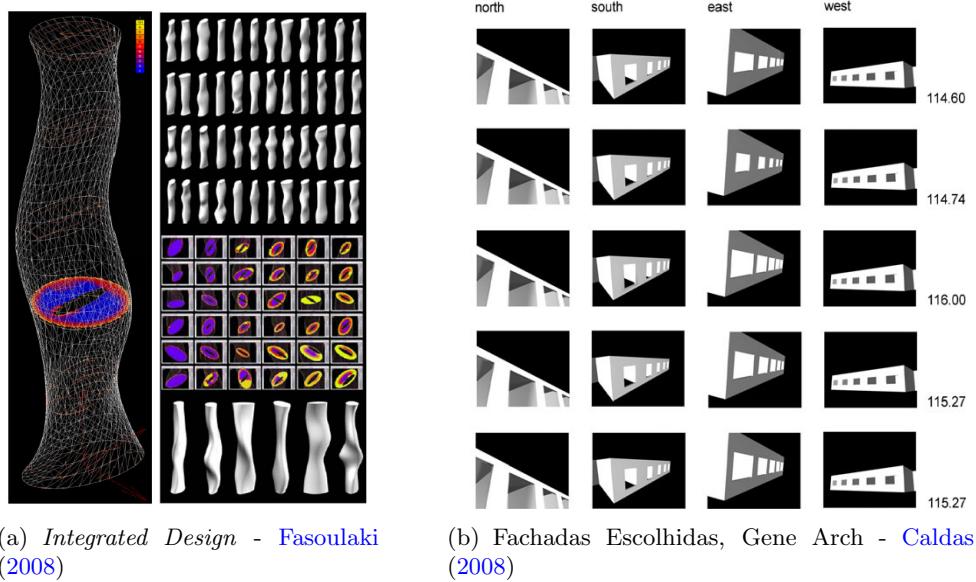


Figura 2.10: Exemplo de Aplicações de Algoritmos Genéticos na Arquitetura

Inicialmente o algoritmo gera aleatoriamente uma população inicial de indivíduos, um conjunto de possíveis configurações das variáveis do problema. Em seguida aplica-se a função objetivo, também chamada de *fitness function* ou função de aptidão, para todos os candidatos gerados no passo anterior. A *fitness function* alude ao conceito evolucionista em que os mais aptos sobrevivem. Na série de trabalhos desenvolvidos por [Caldas e Norford \(1999\), \(2001\), \(2002\), \(2008\)](#) o que se pretende analisar é a temperatura e iluminação resultante das dimensões das janelas de fachadas com orientações distintas, em diferentes cidades do mundo. Os indivíduos são configurações das fachadas com variadas dimensões de abertura para cada fachada. A *fitness function* se alimenta pelos valores de iluminação e temperatura analisada pelo software DOE2.1E. Simulando as características climáticas de uma cidade específica, o *software* recebe os parâmetros de cada individuo e calcula a sensação térmica e a iluminação em um cômodo com essas configurações e a energia gasta para que, com iluminação e controle térmico artificiais, o ambiente se aproxime de condições definidas como ideais. A partir desses dados, ordenam-se os indivíduos de acordo com a eficiência energética de cada análise, medindo-se assim a aptidão à "sobrevivência" de cada candidato. As melhores soluções são eleitas para a próxima fase. O *crossover* permite reorganizar partes de duas soluções eleitas, gerando uma nova configuração, um novo indivíduo integrante de uma nova geração e mutações ocorrem com definida probabilidade sobre os indivíduos da população. Os trabalhos evoluem e um *software* de otimização baseado em algoritmos genéticos para Arquitetura, o *Gene Arch*, é apresentado [2.10\(b\)](#).

Mutação é um evento aleatório que ocorre com uma probabilidade definida pelo usuário para alguns indivíduos da população. É aplicada para garantir a diversidade genética da nova geração ([FASOULAKI, 2007](#))². As operações são repetidas seguidamente, de geração à geração, analisando candidatos até que os critérios de parada do algoritmo sejam atendidos e a solução otimizada seja escolhida.

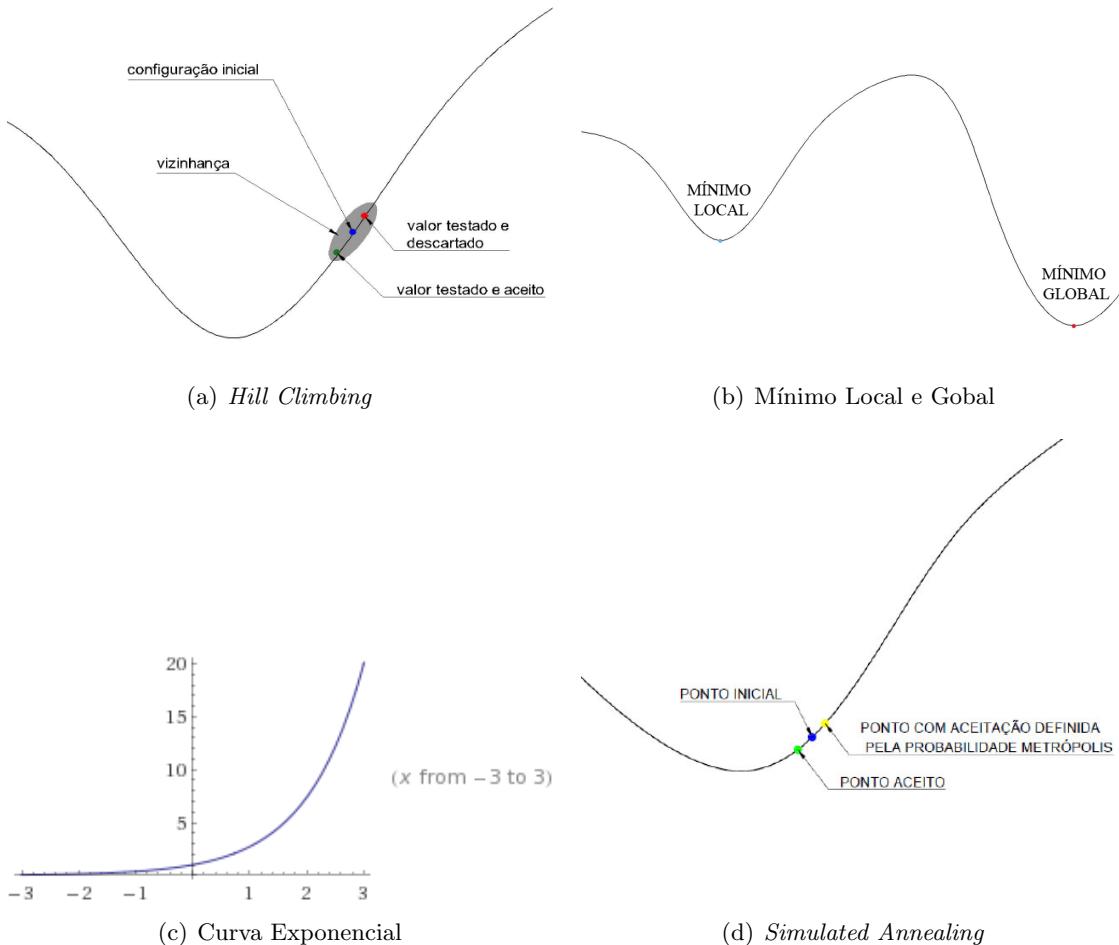
No experimento proposto por [Fasoulaki \(2008\)](#) quatro características de um edifício em Nova York são otimizadas: o insolejamento, a estrutura, a ventilação e maximização da área construída de acordo com as limitações do código de obras da cidade. O modelo é parametrizado em sete pavimentos de controle elípticos igualmente distantes. Neles O desempenho térmico e a iluminação são avaliados. A volumetria é criada a partir da interpolação de uma superfície entre as linhas de contorno de cada pavimento de controle. A área das interseções de planos na altura de cada andar (para este cálculo não são utilizados apenas os pavimentos de controle) é somada para se aferir o valor da área construída . Avalia-se através de uma planilha o custo estimado da estrutura de cada candidato. Os genes controlam a posição e rotação de cada pavimento de controle e as mutações transformam pavimentos elípticos em circulares. A função de aptidão é avaliada após normalizarem-se os valores de cada análise entre zero e um, atribuindo-se pesos para cada uma das quatro características estudadas. O método, intitulado *Integrated Design* ([Figura \)2.10\(a\)](#)), defende uma negociação entre os diversos profissionais envolvidos no projeto, ouvindo especialistas de cada área para que se definam os pesos. A grosso modo, é a tentativa de estabelecer uma proporção entre o benefício de uma melhor insolação com o custo estrutural que isso acarreta, por exemplo.

2.2.6.2 Simulated Annealing (SA)

Supondo que, em um problema de otimização onde se procura o valor mínimo de uma função utilize-se o seguinte método: uma configuração inicial, representada pelo ponto P_1 é fornecida para o problema. Outros pontos são escolhidas aleatoriamente na vizinhança do P_1 , chamados aqui genericamente de P_n . A função objetivo $F(x)$ é avaliada para os novos pontos. Caso o valor da função objetivo para P_n seja menor que o valor da função objetivo em P_1 , a configuração P_n é armazenada como um candidato possível para melhor solução. Se $F(P_1) < F(P_n)$, o ponto P_n é descartado.

O método descrito acima se chama *hill climbing*, ilustrado na [Figura 2.11\(a\)](#). Apresenta grande velocidade de convergência, caso o problema seja bem definido, a função objetivo seja simples ou o ponto inicial seja convenientemente escolhido. Em casos onde a função apresenta uma curva menos previsível, apresentando mínimos locais e um mínimo

²*Mutation is a random event, occurring with a user-defined probability to only some of the new offspring. It is used to maintain genetic diversity by altering only a little piece of the new offspring.*

Figura 2.11: *Simulated Annealing* - Conceituação

global, como no exemplo da Figura 2.11(b), dependendo do ponto inicial escolhido, o procedimento apresentado pode convergir para um mínimo local e ficar estacionado. Em funções objetivos complexas e com número de variáveis maior que dois, a determinação da região que faria o resultado do *hill climbing* convergir para o mínimo global costuma ser impreciso.

O *Simulated Annealing*, ou recosimento simulado, propõe uma alteração no algoritmo *hill climbing* baseando-se no processo metalúrgico do recosimento (*annealing*) e no algoritmo Metropolis-Hastings. O objetivo desta implementação é um método de otimização que consiga executar uma varredura mais completa no espaço de soluções do problema, evitando que o resultado fique sujeito à convergência em mínimos locais.

O recosimento é um processo metalúrgico de se resfriar gradualmente um corpo, com

uma queda lenta e controlada da temperatura para que as moléculas formem cristais, a acomodação que acarreta em uma menor energia para o sistema. Caso a queda da temperatura seja brusca, a tendência é vitrificar, equivalendo a mínimos locais no que diz respeito á minimização da energia do sistema.

O algoritmo Metópolis-Hastings, ou simplesmente o algoritmo Metrópolis, se encaixa perfeitamente na analogia criada por se tratar de uma aplicação do algoritmo probabilístico de Monte-Carlo para simular a configuração energética mais provável das moléculas de um sistema em equilíbrio térmico. O modelo físico utilizado para relacionar energia com temperatura foi a distribuição de Boltzmann, descrita na equação 2.1:

$$P(E) = e^x , \text{ sendo } x = \frac{-\Delta E}{k \times T} \quad (2.1)$$

A distribuição de Boltzmann parte da noção Física de que as partículas de um sistema em equilíbrio térmico não tem todas a merma energia. A distribuição desta energia não acontece por igual, de forma que a soma de pequenas partes iguais resultem na temperatura do sistema. Calcula-se aqui a probabilidade, $P()$, de uma dessas moléculas ter determinada energia, E , em função da variação da energia global, ΔE , e da temperatura do sistema, T . O k é a chamada constante de Boltzmann ($1,38 \times 10^{-23} J/K$), e a curva da probabilidade, $P(E)$ apresenta a forma de uma exponencial, e^x , mostrada na Figura 2.11(c).

Os algoritmos SA possuem três requisitos básicos:

- uma função objetivo;
- uma configuração inicial das variáveis;
- uma agenda de resfriamento.

Semelhante ao modelo *Hill Climbing*, configurações aleatórias são escolhidas na vizinhança da configuração inicial. Caso o valor da função objetivo seja menor em um ponto sorteado na vizinhança do que na configuração inicial, esse ponto é aceito no conjunto das configurações, caso contrario, o ponto não é imediatamente descartado como no *Hill Climbing*, mas se aceita ou refuta o candidato de acordo com uma probabilidade baseada na distribuição de Boltzmann (Figura 2.11(d)).

Para os pontos com função objetivo maior que á inicial, a probabilidade de ingresso no conjunto de candidatos aceitos é definida por uma avaliação da função $P(E)$, tendo por ΔE a variação da função objetivo para $P(1)$ e $P(n)$ e T a temperatura atual da agenda de

resfriamento. A temperatura tem importância determinante no valor da probabilidade e permanência do novo ponto. Se T tem um valor muito alto, x será um valor negativo muito baixo, levando a função e^x assumir posição próxima a 1 no gráfico da Figura 2.11(c). a medida que T diminui, o valor de e^x se aproxima de zero. Com a probabilidade calculada, um número entre zero e 1 é sorteado (algoritmo pseudo-randômico). Se a probabilidade é maior que o número sorteado, o ponto é aceito, caso contrario é descartado. E o processo reinicia tendo os membros do conjunto dos candidatos aceitos como configuração inicial.

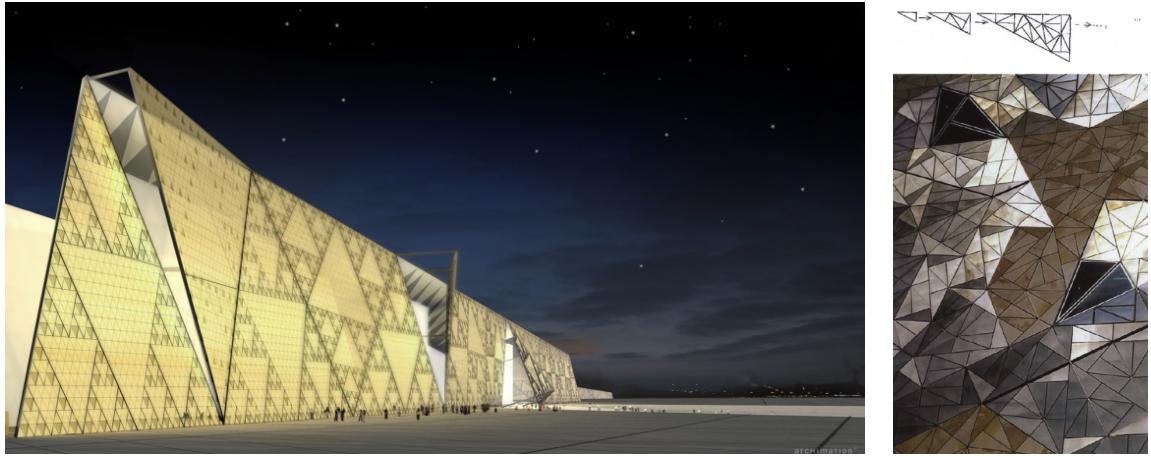
A agenda de resfriamento define de quantos em quantos *loops* a temperatura sera controladamente reduzida, e quais valores vai assumir ao longo do tempo de execução do SA. Os critérios são definidos pelo usuário é podem variar de acordo com o problema a ser otimizado.

No início de sua execução, a probabilidade de aceitação de valores rejeitados pelo *Hill Climbing* é muito grande, permitindo ao SA explorar amplamente o espaço de soluções possíveis. Com o resfriamento simulado no algoritmo, menos candidatos com $P(1) < P(n)$ são aceitos e os candidatos tendem a convergir para diversos mínimos locais, eventualmente encontrando o mínimo global do sistema ([KIRKPATRICK; GELATT; VECCHI, 1983](#)).

[Lamberti \(2008\)](#) propõe uma variação do SA denominada CLMPSA (*Corrected Multi-Level Multi-Point Simulated Annealing*), descrevendo detalhadamente a estrutura do algoritmo para a otimização estrutural de treliças. O método procura mover configurações de maior valor para a função objetivo antes de fazer o teste probabilístico, mostra um critério de definição da agenda de resfriamento em função das variações da energia do sistema e compara os resultados com implementações do SA alternativas e outros métodos de otimização. [Sonmez \(2007\)](#) usa o SA para otimização de forma estrutural submetida a carregamento quasi-estático representada em duas dimensões.

2.2.7 Fractais

Embora não seja exatamente um algoritmo, mas possam ser gerados por uma gama destes, os fractais são uma das mais importantes descobertas científicas do século XX. Uma particularidade interessante é que estas formas geométricas atraíram a atenção não só de cientistas, mas também de artista e *designers*. Na Arquitetura não foi diferente. O escritório [Heneghan Peng Architects \(2014\)](#), por exemplo, no projeto do *Grand Egyptian Museum* em Gizé no Egito, utiliza do Triângulo de Sierpinski em uma fachada lateral translúcida tanto como uma alusão à emblemática imagem histórica das pirâmides quanto como um contraponto estético contemporâneo.



(a) *Grand Egyptian Museum Gizeh*, Egito - fonte: [Heneghan Peng Architects \(2014\)](#) (b) *Federation Square of Melbourne* - fonte: [Hammer \(2006\)](#)

Figura 2.12: Exemplos de Utilização dos Fractais na Arquitetura

O termo fractal foi cunhado pelo matemático Benoit Mandelbrot em um artigo publicado em 1975. Tem por raiz a palavra *fractus*, oriunda do latim, significando "quebrado" ou "fragmentado". Aplicando processos de computação gráfica para estudar o comportamento de métodos iterativos na resolução de certas equações complexas (tendo por base alguns trabalhos do matemático Gastón Julia), gerou-se uma série de gráficos onde propriedades geométricas divergentes das estabelecidas nos estudos euclidianos foram encontradas, estudadas e mesuradas. Ao ramo da Matemática que trata dos elementos que possuem estas propriedades foi dado o nome de Geometria Fractal. O livro "The Fractal Geometry of Nature" de Benoit Mandelbrot ([MANDELBROT, 1982](#)) que, segundo o prefácio do autor, é uma extensão e correção de seus artigos anteriores sobre o tema; parte da ideia de que uma montanha não é um cone e as nuvens não são esferas, propondo uma abordagem científica aplicável ao estudo de formas não euclidianas. Como principais características dos fractais estão a auto-similaridade (ligada ao conceito de escala), replicabilidade infinita e a dimensão de Hausdorff diferente da dimensão euclidiana (muitas vezes não sendo um número inteiro).

Para facilitar a apreensão dos conceitos básicos, tomou-se como exemplo o fractal proposto pelo matemático polonês Waclaw Franciszek Sierpinski em 1915, o Triângulo de Sierpinski. Embora a figura proposta seja anterior à criação deste ramo da geometria, não esta só entre as construções matemáticas preexistentes que se encaixam no conceito.

Sua construção se dá pelo desenho inicial de um triângulo, a partir do qual um segundo triângulo é formado pela ligação dos pontos médios dos lados do primeiro. Considera-se que este segundo triângulo é uma subtração da área do original (quebra ou fragmentação). Tem-se agora três triângulos e um vazio triangular(Figura [2.13\(a\)](#)). Em cada um dos

triângulos o procedimento é refeito, gerando mais três triângulos para cada. Repete-se a operação infinitas vezes e a cada etapa se dá o nome de iteração, numerando-as a partir da primeira vez que a forma inicial é subdividida (primeira iteração).

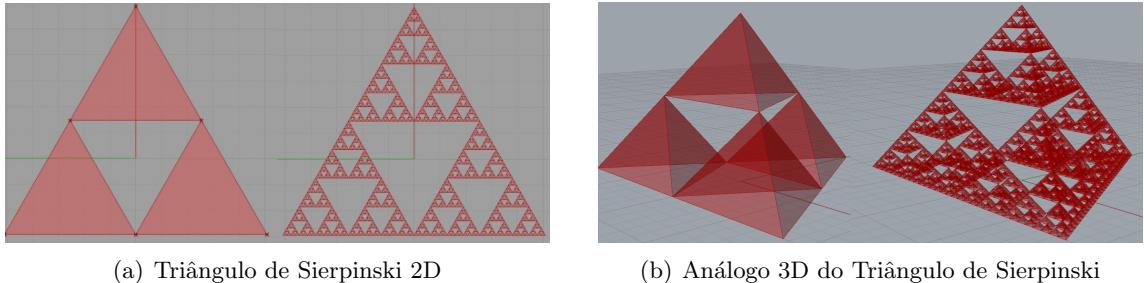


Figura 2.13: Gerados no *Rhino3d/Grasshopper* utilizando o *add-on Hoopsnake* para uma e cinco iterações

A partir da Figura 2.13, gerada no ambiente *Rinoceros / Grasshopper* (ver Apêndice B.1) pode-se aferir pela simples observação duas das características básicas dos fractais e calcular a terceira. Nota-se que pequenos pedaços do desenho reproduzem o todo. Tendo em mente a ideia de uma constante aproximação da figura, de tal forma que a escala dos triângulos menores vai aumentando e, em dado momento, um deles assume o tamanho do triângulo inicial. É intuitivo perceber que a imagem encontrada é semelhante ou mesmo igual (caso não se possa enxergar qualquer pedaço contido nos outros triângulos gerados na mesma iteração, e o número de iterações seja equivalente para se ter o mesmo nível de detalheda imagem) à figura como era vista antes da aproximação. A capacidade dos fractais de carregar em suas partes a representação do todo se dá o nome de auto-similaridade.

Um dos interesses científicos no estudo dos Fractais é a sua ocorrência na natureza. Uma planta onde suas estruturas ramificam-se repetindo em escala reduzida formas semelhantes ao todo (ou a parte do todo) é o que se observa no triângulo de Sierpinski sob o conceito de auto-similaridade (uma parte representa o todo). A replicabilidade infinita, a possibilidade de observar elementos recorrentes independentemente de quantas (infinitas) aproximações (iterações) sejam efetuadas, só é encontrada em fractais da Matemática. O raciocínio que embasa o cálculo da dimensão fractal, também chamada de dimensão de Haussdorf, é ilustrado na Figura 2.14. Para cada uma das dimensões cartesianas desenhou-se uma figura que a representasse (uma reta, um quadrado e um cubo). Cada uma das dimensões de cada figura é dividida em um número qualquer (porem igual para cada dimensão) de vezes.

Sendo r o número de subdivisões em cada dimensão e N o número de figuras semelhantes à inicial geradas por r subdivisões, tem-se $N = r^D$, onde D é a dimensão euclidiana da figura. A variável r pode também ser escrita como L/u , onde L é o tamanho de

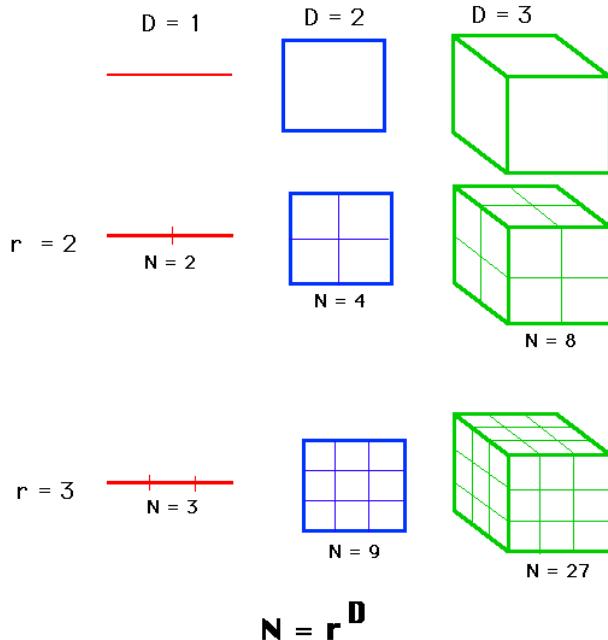


Figura 2.14: Dimensão de Haussdorf - fonte: http://en.wikipedia.org/wiki/Fractal_dimension

cada dimensão da figura e u o tamanho de cada subdivisão. Coloca-se D em evidência encontra-se a equação 2.2.

$$D = \frac{\log N}{\log L/u} \quad (2.2)$$

Para todas as formas e divisões mostradas na Figura 2.14 o valor D calculado será igual ao número de dimensões euclidianas que cada geometria representa (um para a reta, dois para o quadrado, três para o cubo). Aplicando a equação 2.2 para calcular a dimensão fractal do Triângulo de Sierpinski, onde cada lado L é dividido duas vezes por iteração, $L/u = 2$, formando três figuras semelhantes a inicial, $N = 3$. Logo $D = \frac{\log 3}{\log 2}$. Portanto a dimensão de Haussdorf do fractal é de 1,5849.

Mas os números não importam por enquanto. Em um nível mais profundo do estudo, esses números traduzem informações sobre a natureza das coisas. Primeiramente é recomendável entender o significado do que eles medem: a maneira como o espaço se dobra, a quantidade da rugosidade de uma superfície, a textura das formas da natureza, as curvas da linha entre a praia e o mar, o detalhe de como certos tipos de matéria se agrupa. Imagens direta ou indiretamente presentes nos pensamentos sobre a construção. A dimensão fractal é uma quantidade numérica que aproxima as ramificações dos galhos da árvore à seu todo.

Em 1997 o escritório de arquitetura londrino *Lab Architectural Studio* ganhou o concurso

internacional para o projeto de um centro cultural, o *Federation Square* na cidade de Melbourne, Austrália. A Figura 2.12(b) mostra um detalhe da fachada de um três prédios principais do conjunto (abaixo) e a estratégia utilizada na subdivisão dos triângulos sucessivamente em cinco triângulos menores (acima), mantendo sempre a proporção de 1:2 entre os lados perpendiculares. Os triângulos são construídos de três materiais: vidro, arenito e zinco. Um mosaico multifacetado e multicolorido, segmentando, absorvendo, refletindo e enquadrando a luz e definindo os espaços de forma peculiar e fragmentada ([HAMMER, 2006](#)).

2.2.8 Modelos de Parâmetros e Restrições (MPR)

Todos os algoritmos generativos, bem como toda computação, baseiam-se em parâmetros. Os modelos paramétricos, contudo se diferenciam dos demais por colocar a ênfase na manipulações destes ([DINO, 2012](#)). A modelagem paramétrica permite alteração a qualquer momento de algumas variáveis e o automático rebatimento desta alteração no modelo. Desenhando um prisma quadrangular reto (paralelepípedo), deve-se fornecer valores reais para a altura, largura e profundidade. Na modelagem paramétrica é permitido alterar qualquer uma dessas variáveis, redesenhadando o prisma automaticamente.

Os modelos paramétricos de projeto pode ser divididos em dois tipos: conceituais e construtivos. Os últimos tem por principais parâmetros as informações envelopadas em formas 3D predeterminadas. Esses modelos desempenham papel crucial na implementação do conceito de BIM, visto que seu objetivo é agregar o maior número de informações construtivas possíveis dentre os seus parâmetros. Nos modelos conceituais os parâmetros de projeto são definidos em primeira etapa e guiam a procura pela forma. Representando uma mudança mais radical no processo de projetar que os modelos paramétricos construtivos ([MILENA; OGNEN, 2010](#)). Essa mudança no processo coloca os modelos paramétricos conceituais dentro do campo de estudo dos Algoritmos Generativos.

As duas categorias, porém não são estanques. [Donath e Lobos \(2009\)](#), por exemplo, desenvolvem um modelo paramétrico conceitual para projeto da forma externa de prédios residenciais isolados e de grande altura que, desde a instigação inicial, apoia-se em uma ferramenta especializada em modelos paramétricos construtivos, tornando evidente a facilidade com que as informações construtivas poderão ser inseridas durante o processo.

Dentre as principais qualidades de um Modelo Paramétrico está a capacidade de resolver questões bem definidas, com um objetivo claramente descrito, quando de abordar questões pouco definidas e complexas que apresentam diversas soluções viáveis ([DINO, 2012](#)).

Os Modelos Paramétricos Conceituais são aqui tratados como Modelos de Parâmetros e

Restrições por razões técnicas e metodológicas. A ideia é enfatizar na nomenclatura a importância tanto dos parâmetros quanto das regras implementadas na forma de algoritmos, que geram as formas geométricas à serem estudadas. Essas regras, falando especificamente da Modelagem Paramétrica Conceitual, podem ser entendidas como restrições. Esta abordagem mostrou-se eficiente, do ponto de vista metodológico, na elaboração das regras e de sua transcrição nos algoritmos desenvolvidos durante este trabalho.

Do ponto de vista técnico a enfase nas restrições se justifica pelo modelo computacional baseado em *Constraint Solvers*. Este modelo lida com problemas de satisfação de restrições (*constraint satisfaction problems*), onde um conjunto finito de variáveis, cada uma delas restritas á um domínio finito, tem os valores que podem assumir concomitantemente definidos por restrições (BARTÁK, 2001). Modelos estes que, desde sua introdução no mercado, pelo software *Pro/Engineer* na década de 1980, foram adotados pela pelos mais importantes sistemas CAD disponíveis, representando uma grande mudança na maneira que os usuários interagem com suas ferramentas de projeto. A criação e manipulação de formas geométricas através dos *Constraint Solvers* tem forte relação com a demonstração de teoremas em Geometria (HOFFMANN; JOAN-ARINYO, 2005).

Para ilustrar essa características, um exemplo de MPR foi elaborado e será apresentado em seguida. Inspirado em duas obras do arquiteto João Filgueiras Lima (Lelé), este MPR descreve as cúpulas móveis usadas no auditório da unidade do Rio de Janeiro da Rede Sarah de Hospitais (Figura 2.15(a)), e no Auditório e plenário da sede do Tribunal Regional do Trabalho, em Salvador - Ba (Figura 2.15(b)).



(a) Hospital Sarah-RJ fonte: <http://www.sarah.br/>



(b) Perspectiva TRT-Ba fonte: Acervo do Instituto Habitat

Figura 2.15: Cúpulas móveis do Arquiteto João Filgueiras Lima (Lelé)

As cúpulas móveis permitem o controle da iluminação e exaustão nos auditórios. Possibilitando Tanto que ocorram de maneira natural, economizando energia, quanto artificiais, para que, por exemplo, se possa fazer melhor uso de mídias projetadas em telões. O

número de pétalas, o raio da base da cúpula e a altura da calota esférica variam entre os dois projetos. Entre os parâmetros passíveis de otimização estão:

- o número de pétalas;
- o raio da base;
- a altura da cúpula.

O número de pétalas tem relação direta com os mecanismos (motores) de operação das cúpulas. Uma quantidade maior de pétalas, menor o peso de cada uma e um maior número de motores elétricos responsáveis pela articulação de cada peça. O raio implica diretamente na área da abertura de iluminação e exaustão. As três variáveis tem implicação direta no aspecto visual da cúpula, sendo exigidas na abordagem do problema estético. O exemplo aqui proposto apresenta um algoritmo capaz de desenhar as cúpulas de ambos os projetos e variações destas. Além das três variáveis, duas foram acrescentadas: o ponto central do círculo de base da calota, e o ângulo de rotação da abertura das pétalas. O método segue as seguintes etapas:

1. O círculo $C1$ é desenhado partindo do ponto central PC com raio r .
2. $C1$ é dividido em N partes iguais. Os pontos divisores são armazenados em uma lista de pontos $Lp1$, ordenados no sentido anti-horário, partindo de um ponto inicial.
3. Cria-se um ponto PH movendo o ponto PC com uma distância h ao longo do vetor perpendicular ao plano do círculo $C1$.
4. Uma lista de segmentos de reta $Llin1$ é criada ligando o ponto PC a cada um dos pontos da lista $Lp1$.
5. Uma lista de arcos $Larc1$ é criada partindo do ponto PH sendo tangente aos vetores representados pelos segmentos orientados da lista $Llin1$ e com ponto final os da lista $Lp1$.
6. uma lista de superfícies $Ls1$ é criada pelo método de revolução. Rotacionando os arcos da lista $Larc1$ em torno de um eixo definido pelos pontos PC e PH ao longo de um ângulo de $360/N$ e sentido anti-horário.
7. uma lista de pontos $Lp2$ é criada a partir da lista $Lp1$ extraindo o primeiro elemento da lista e colocando no final: sendo a lista $Lp1$ composta por $Lp1_0; Lp1_1; Lp1_2...Lp1_N$ nesta ordem, em $Lp2$ os elementos de $Lp1$ na sequência $Lp1_1; Lp1_2...Lp1_N; Lp1_0$.
8. Uma lista de segmentos de reta $Llin2$ é criada ligando os pontos de mesmo índice das listas $Lp1$ e $Lp2$.

9. O ângulo α é definido pelo usuário.
10. Cada elemento da lista de superfícies $Ls1$ é rotacionado um ângulo α em torno do elemento correspondente da lista de segmentos de reta $Llin2$.
11. Fim.

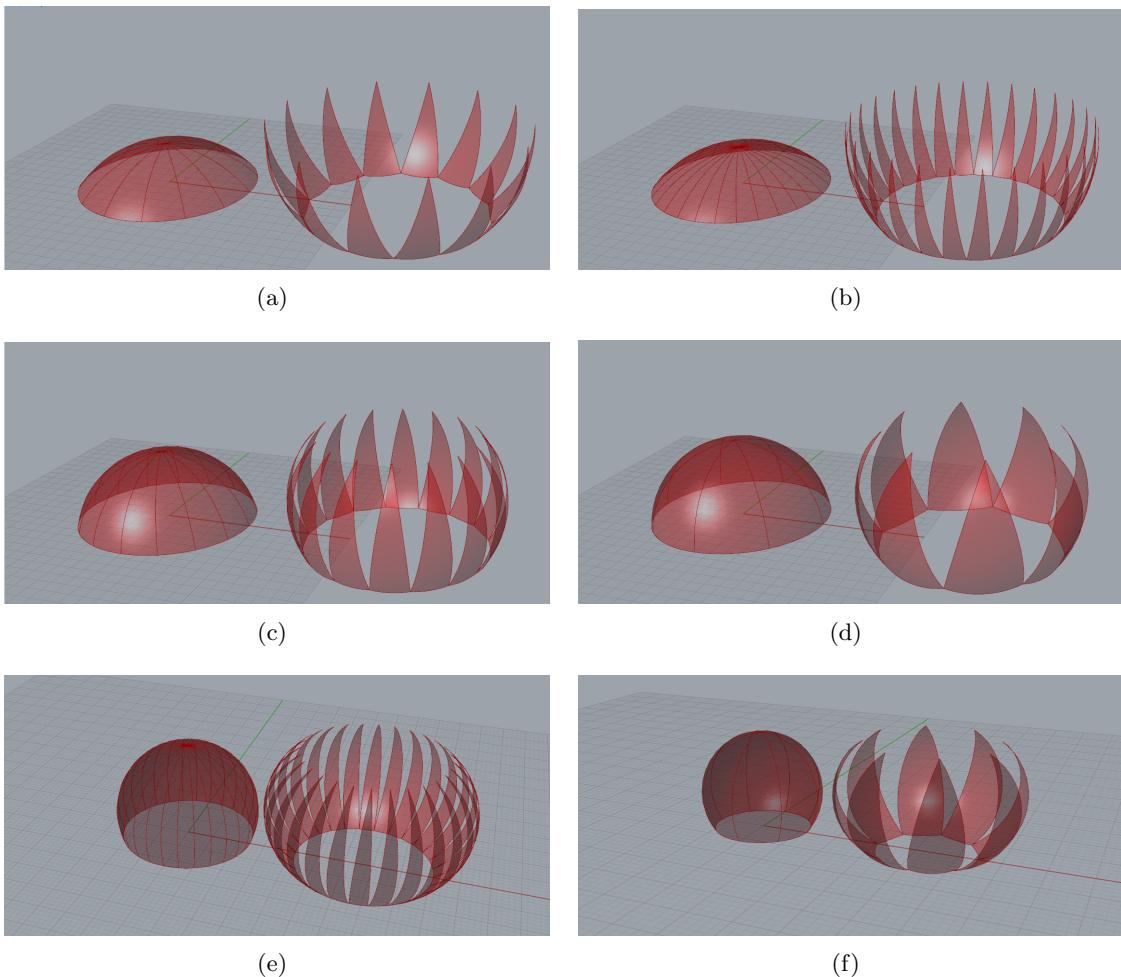


Figura 2.16: Cúpulas móveis geradas por MPR - *Rhinoceros/Grasshopper*

O algoritmo foi implementado no ambiente *Rhinoceros/Grasshopper* e pode ser visto no Apêndice B.2. A Figura 2.16 mostra algumas formas geradas pelo processo. Os parâmetros de entrada são: o ponto PC , o raio r , a distância de deslocamento h , o número de pétalas N e o ângulo de rotação α . A criação do ponto PH , perpendicular ao plano do círculo $C1$, é uma exemplo de restrição. O método de criação das geratriz $Larc1$, em função das tangentes dos vetores em $Llin1$, restringe a forma da cúpula fechada à uma calota esférica, com ponto central pertencente à reta que passa pelos pontos PC e PH .

Aproximando-se do conceito de demonstração de um teorema em Geometria, nota-se uma importante relação entre variáveis r e h :

- sempre que r for igual à h , a calota é uma semiesfera e o centro da esfera coincide com o ponto PC ;
- sempre que r for maior que h , a calota é menor do que a metade da esfera e o seu centro está abaixo do ponto PC ;
- sempre que r for menor que h , a calota é maior do que a metade da esfera seu centro está acima do ponto PC ;
- quando o valor de r ou de h é igual a zero, o algoritmo falha na geração de formas, tornando essas configurações invalidas para a solução do problema.

Ampliando essas relações para todo o conjunto de parâmetros, é possível afirmar que, para cada configuração válida destes, existe uma única cúpula como resposta. O mesmo pode ser dito para qualquer MPR que não utilize, de alguma forma, a aleatoriedade em seu conjunto de regras. Cada pétala é representada computacionalmente como uma curva *nurbs* e, assim como cada uma das geometrias auxiliares utilizadas pelo algoritmo, é composta por um conjunto de variáveis cujos valores são definidos pela atuação do *constraint solvers*. Os parâmetros são variáveis especiais que alimentam e manipulam o conjunto de restrições que guiam a criação das formas.

Enquanto os parâmetros de entrada podem ser dinamicamente modificados pelo usuário, as restrições só podem ser alteradas pela modificação do algoritmo. A criação das formas é guiada por um diálogo entre os valores explicitamente controlados pelos projetistas e as restrições impostas pelas regras transcritas no algoritmo. Enquanto os primeiros garantem a criação de variedade pela simples alteração de valores, também chamadas de flexões do modelo, as últimas garantem que a forma gerada faz parte de um conjunto de resultados específicos, compatíveis com o problema abordado, e apresenta as intenções de projeto, não como forma, mas como ideia.

Grafostática

*"Los que buscan las leyes de la Naturaleza
como un apoyo para sus nuevas obras
colaboran con el creador.
Copiadores no colaboran.
Debido a esto, la originalidad
consiste en regresar al origen."*

(Antoni Gaudí)

A origem da Geometria está intimamente ligada com as primeiras civilizações humanas. Reconhecendo as dificuldades de estabelecer um consenso em torno das origens da Matemática (Aritmética e Geometria), Boyer e Gomide (1974, p. 4) apontam às teorias antagônicas de Heródoto e Aristóteles como as hipóteses mais aceitas para o surgimento da Geometria. Heródoto acreditava que a origem estava ligada à necessidade prática de se fazer novas medidas de terra, após cada inundação anual, no vale do antigo Egito. Enquanto Aristóteles creditava à classe sacerdotal egípcia, que a teria criado para apreciação do belo. E lembram o fato dos geômetras do Egito serem frequentemente chamados de "esticadores de corda" como um apoio comum as duas teorias, já que essas eram esticadas tanto para demarcar terras, quanto para marcar as bases dos templos. Em ambos os exemplos a atividade de construir está presente.

Ceccato (2010) chama atenção para o significado das palavras "arquiteto" (mestre construtor) e Geometria (medida da terra) para afirmar que geômetras e arquitetos eram, primordialmente, a mesma pessoa. A ligação entre Geometria e construção são evidentes e profundas, mesmo tendo em conta a subdivisão em diversas profissões pela qual a atividade de construir passou ao longo de séculos. Antes do desenvolvimento da Mecânica dos Materiais (por volta do século XIX), a construção de estruturas era guiada por um conhecimento empírico das proporções entre esforços e materiais (MCCROMAC, 2009).

3.1 Estática e Grafostática

A Mecânica é o ramo da Física que estuda o movimento e repouso e divide-se em Cinemática e Dinâmica. Enquanto a primeira se debruça sobre o movimento sem abranger as suas causas, a segunda trata do efeito das forças sobre o movimento e repouso dos corpos, e se divide em Cinética e Estática. Johnson (1908) aponta métodos algébricos

ou gráficos podem ser alternadamente mais eficazes para a resolução de determinados problemas da Estática, definida como o ramo da Dinâmica que estuda o equilíbrio dos corpos. Desde então, pouco mudou nas bases e definições dos problemas da Mecânica aplicada às construções, mas diferenças significativas aparecem nos métodos e ferramentas de trabalho dos profissionais envolvidos.

Grafostática, ou **Estática Gráfica**, estuda os problemas de Estática por métodos puramente geométricos. É de grande utilidade em Resistência dos Materiais para a determinação das reações, esforços cortantes, momentos fletores, flechas e, nos sistemas articulados, para a análise dos esforços interiores que acontecem nas barras dos mesmos. ([SANJUAN, 1951](#))

As ferramentas analíticas e gráficas de cálculo estrutural foram desenvolvidas paralelamente ao longo do século XIX. Trabalhos como o de James Clerk [Maxwell \(1965, p.451-525\)](#) são citados tanto como fundamentais na evolução da teoria do cálculo estrutural moderno ([MCCROMAC, 2009](#)), quanto fundador dos procedimentos geométricos de resolução dos problemas de Estática, desenvolvidos principalmente pelos trabalhos de Culmann (1866) e Cremona (1890) ([JOHNSON, 1908](#)) ([GERHARDT; KURRER; PICHLER, 2003](#)) ([LACHAUER; KOTNIK, 2010](#)).

A medida que ferramentas numéricas, tanto mecânicas (e.g. régua de cálculo) mas principalmente computacionais crescem em eficiência ([BAUMGART, 2000](#)) e uma tendência analítica se desenha na engenharia ([GERHARDT; KURRER; PICHLER, 2003](#)), a Estática Gráfica perde espaço e passa a ser encarada quase que apenas como um instrumento didático para o entendimento da Mecânica, como no livro de [Costa \(1974\)](#).

O posterior desenvolvimento da computação gráfica acabou demonstrando que, também no ambiente computacional, o método analítico nem sempre é a maneira mais eficaz de se abordar um problema. O exemplo mais notório provavelmente está na descoberta dos fractais por Mandelbrot. Embora, no experimento que deu origens à este desenvolvimento científico, funções estivessem sendo calculadas através de métodos numéricos e vertidos em gráficos por um processo de mesma natureza, o avanço no conhecimento acontece pela inspeção do olhar humano sobre os desenhos dos resultados, padrões são encontrados, estudados e uma ferramenta matemática é testada e aceita para medir as características destas novas geometrias.

Sobre a Grafostática, surge a hipótese de que o uso das ferramentas computacionais gráficas para o auxílio à tomada de decisões sobre formas geométricas edificáveis, é capaz de trazer uma compreensão mais ampla, objetiva e clara para o estudo, ensino e prática das questões estruturais. As vantagens cognitivas de utilizar formas geométricas (diagra-

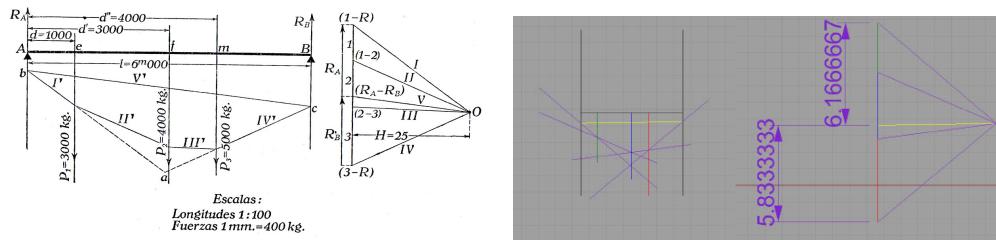
mas), em oposição ao uso de expressões matemáticas, para a análise de outras formas (construção) é evidente.

Entre os defensores desta hipótese, [Gerhardt, Kurrer e Pichler \(2003\)](#) apontam as virtudes do método além dos exemplos didáticos. Estabelece relação direta entre o uso histórico da ferramenta e a maneira como ela é capaz de guiar o projetista pelo espaço de soluções possíveis, destacando os trabalhos de Antoní Gaudí e Gustave Eiffel. Um exemplo de utilização da Grafostática em ambiente CAD é apresentado no projeto de reforço estrutural em um prédio da prefeitura de Berlim, onde a proposta executada é enxergada apenas depois de se analisar os diagramas construídos através do método. [Block, Ciblac e Ochsendorf \(2006\)](#) aplicam mecanismos da Grafostática na elaboração de uma estratégia computacional para análise em tempo real de projetos com estruturas em pedra. [Tessmann \(2008\)](#) elenca a Estática Gráfica entre as ferramentas eficientes para aprimoramento do diálogo entre projetistas de arquitetura e de estrutura. [Zanni e Pennock \(2009\)](#) trabalham em um modelo computacional gráfico para análise de estruturas submetidas à carregamento axial. [Shearer \(2010\)](#) propõe algoritmos para automação do desenho de diagramas da Grafostática e [Lachauer e Kotnik \(2010\)](#) defendem criação de uma ferramenta dinâmica para geração dos mesmos, recebendo informações de um MPR. [Harding e Shepherd \(2011\)](#) aponta que a Grafostática de Culmann serve de base para alguns algoritmos computacionais dinâmicos de exploração interativa 2D de estruturas e propõe um algoritmo que adicione o eixo z no cálculo. O trabalho de [Block \(2009\)](#), aplica a Estática Gráfica como ferramenta para "encontrar" interativamente estruturas submetidas unicamente à compressão (*form finding*).

3.2 Precisão do Método

A precisão dos métodos gráficos aplicados ao cálculo de estruturas é demonstrado por diversos autores. [Panseri \(1968, p. 30\)](#) mostra que apenas duas ou três casas decimais são plenamente satisfatórios quando levadas em conta as tolerâncias de fabricação, as incertezas que envolvem certos esforços e os coeficientes de segurança necessários para a eficiente execução de um elemento construtivo. [Johnson \(1908, p. 2\)](#) afirma que, embora os dois métodos sejam cientificamente corretos, a precisão do processo analítico depende da quantidade de casas decimais e algarismos significativos envolvidos. Na resolução gráfica, as habilidades de desenho e os limites da visão humana tem caráter determinante na acuidade do cálculo.

Utilizando a Grafostática em ambiente computacional as diferenças praticamente desaparecem. A Figura 3.1(a) é um problema apresentado por [Sanjuan \(1951, p. 16\)](#) onde o mesmo cálculo de reações é feito de maneira analítica e gráfica, apresentando pequenas variações entre os resultados. O mesmo problema solucionado graficamente em um pro-



(a) Cálculo de reações em uma viga simples - (b) Cálculo de reações em uma viga simples
fonte: Sanjuan (1951) executado no Rhino3D

Figura 3.1: Precisão do cálculo

grama CAD (Figura 3.1(b)) tem os mesmos valores da resolução analítica se executado em uma calculadora com mesmo número de casas decimais utilizadas pelo CAD.

Embora a Estática Gráfica sirva como base científica do experimento, em um nível mais baixo são operações numéricas que realizam o cálculo. A habilidade com os instrumentos de desenho é substituída pelo cuidado na aplicação do comando correto. O espaço cartesiano implementado no computador pode esconder erros relacionados as variáveis de ponto flutuante, mas felizmente, as precauções que devem ser tomadas são as mesmas que devem ser adotadas para o desenho técnico em CAD. Se, no final do século XIX, muitos engenheiros e arquitetos enxergavam os benefícios de associar seu instrumento de projeto (a prancheta) com uma ferramenta de cálculo, semelhantes vantagens podem ser usufruídas por usuários experientes nos atuais e precisos sistemas de projeto assistido por computador.

3.3 Conceitos Fundamentais da Estática

- **Corpo Rígido:** é um corpo que não sofre alterações de forma ou tamanho sob a ação de uma força ou sistema. Embora não exista na natureza um material com essas propriedades, para efeito prático, a grande maioria dos materiais construtivos funcionam como um corpo rígido, em um contexto maior ou menor.
- **Repouso:** a relação existente entre dois pontos quando uma linha imaginária conectando-os não apresenta mudanças em comprimento ou direção.
- **Movimento:** Quando a linha imaginária descrita acima muda. As mudanças no comprimento são chamadas de **translação**, na direção, de **rotação**
- **Força:** Ação que causa (ou tende à causar) modificação no movimento de um corpo. Entendidas como o ato de puxar ou empurrar, as forças são grandezas vetoriais, e como tais possuem três elementos essenciais:

Módulo ou Magnitude: é um valor numérico real que mede a capacidade de uma força em produzir movimento, quando comparado com uma força padrão escolhida. Por definição, este valor não possui notação de positivo ou negativo.

Direção: a direção da linha sob a qual a força tende a produzir movimento.

Sentido: para qual dos extremos da direção a força tende a produzir movimento.

- **Representação Gráfica de Vetores:** É dada através de segmentos orientados. Um dos pontos extremos do segmento é tomado como ponto inicial e o outro como final, definindo assim um sentido. O módulo é representado pelo comprimento e a direção pela reta que contem o segmento, bem como as paralelas à ela.
- **Ponto de Aplicação:** o ponto do corpo em que a força age.
- **Linha de Ação:** a direção do vetor força que passa pelo seu ponto de aplicação. No caso de um corpo rígido, um outro **ponto de aplicação** pode ser considerado em qualquer local sobre a linha de ação de uma força, produzindo o mesmo efeito.
- **Composição de Vetores:** para um sistema vetorial qualquer(**componentes**), a composição consiste em encontrar o vetor (**resultante**) que tende a produzir o mesmo efeito quando aplicado sobre um corpo. Diz-se que as componentes e sua resultante são vetorialmente **equivalentes**. Também chamada de **soma de vetores**, a composição é graficamente executada posicionando as componentes de forma que o extremo final do vetor anterior encontre o extremo inicial do vetor seguinte. A resultante é o segmento orientado que tem o mesmo ponto inicial do primeiro vetor e o ponto final do último.
- **Resolução de Vetores:** para um vetor qualquer, encontrar dois ou mais componentes cuja soma tenha por resultante o vetor dado.
- **Equilibrante:** força que equilibra o sistema. Possui o mesmo mesmo módulo e direção e linha de ação do vetor resultante, porém com sentido oposto.
- **Binários:** duas forças iguais (módulo e direção) e de sentidos opostos, com linhas de ação paralelas sobre o mesmo corpo. Os binários apresentam unicamente a tendência de produzir rotação. A medida desta tendência é chamada de **momento do binário**, que é calculada pela multiplicação do módulo de uma das forças pela distância ortogonal entre suas linhas de ação (braço), em unidade apropriada.

3.4 Diagramas Recíprocos

A Grafostática baseia seus algoritmos de cálculo nos chamados diagramas recíprocos, que consistem em apresentar o mesmo sistema de forças em duas figuras complementares. Em uma temos os vetores das forças atuantes sobre um corpo rígido, grafados em magnitude,

direção, sentido e representados na posição de soma (Figura 3.2(a)), essa figura é referida como polígono de forças (PF). Para a determinação do módulo uma escala que relaciona força e medida é eleita (e.g. $\frac{1\text{cm}}{10Kgf}$) de forma semelhante ao procedimento de desenho técnico em escala. No segundo diagrama, chamado de forma geométrica (FG) ou plano de situação (Figura 3.2(b)), apenas as linhas de ação são representadas.

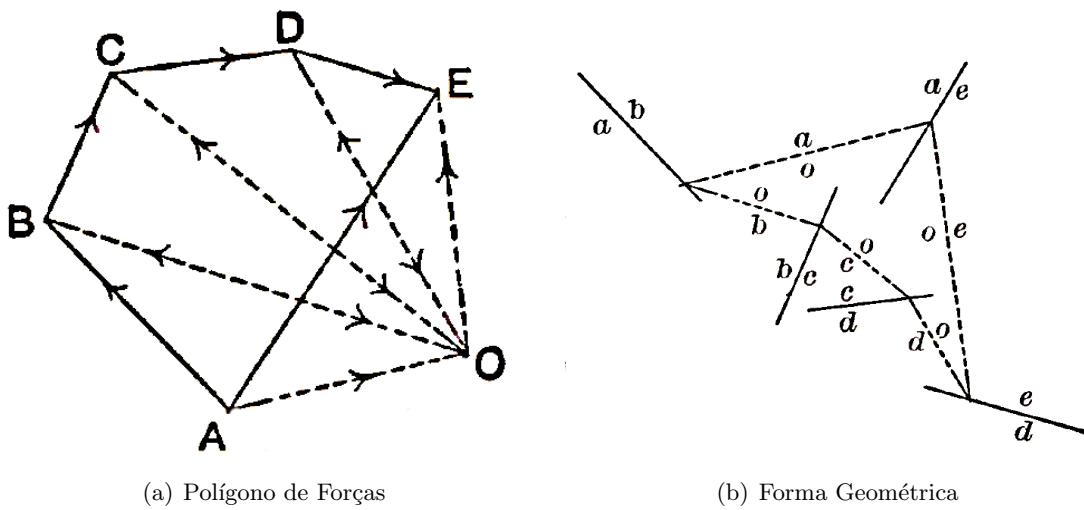


Figura 3.2: Diagramas Recíprocos - fonte: [Hoskins \(1899\)](#) (editado)

Partindo do sistema da Figura 3.2, considerando um conjunto de quatro forças, representadas por AB, BC, CD e DE no polígono de forças, e cujas respectivas linhas de ação são ab, bc, cd e de na recíproca forma geométrica (notação de Bow ([BURR; FALK, 1908](#), p. 5-7)), o cálculo da resultante AE e da sua linha de ação ae é feito pelos seguintes passos:

1. Um ponto qualquer, chamado de polo, representado por O é escolhido no polígono de forças.
2. Os raios polares, linhas partindo do ponto O para cada um dos pontos A, B, C, D e E são desenhados.
3. A reta oa , paralela à OA é traçada na forma geométrica partindo de um ponto qualquer na linha de ação ab .
4. Pelo mesmo ponto que oa intercepta ab , a reta ob paralela à OB é traçada.
5. A reta oc , paralela a OC é traçada partindo do ponto em que ob intercepta bc . O passo é repetido para od partindo da intercessão de oc e cd , e para oe por od e de .
6. O vetor AE é determinado pela ligação dos pontos A e E no polígono de forças.
7. A direção ae é traçada por uma paralela à AE passando pelo ponto intercessão das retas oa e oe na forma geométrica.

A poligonal formada por oa, ob, oc, od , e oe é chamada de polígono funicular, e seus lados são as denominadas de cordas (HOSKINS, 1899, p. 16) ou linhas de pressão (COSTA, 1974). Para cada polo O escolhido aleatoriamente um funicular diferente é construído. Para o sistema apresentado na Figura 3.2, as intercessões de oa e oe referentes ao polo O e a um outro polo qualquer O' estarão necessariamente sobre a linha de ação ae , uma alternativa para o cálculo.

3.5 Condições de Equilíbrio

No método analítico, as condições de equilíbrio de um corpo rígido submetido a um sistema de forças é que, tanto o somatório dos momentos, quanto o das translações seja igual a zero (equação 3.1).

$$\sum M = 0 \quad \sum T = 0 \quad (3.1)$$

Na Grafostática, a análise do equilíbrio é feita pela observação do polígono de forças e do funicular.

- Se PF é um polígono fechado, o sistema está em equilíbrio de translação.
- Se o funicular é um polígono fechado, o sistema está em equilíbrio de rotação.
- Se o PF é fechado e o funicular aberto e de lados opostos paralelos, a resultante do sistema é um binário. O corpo rígido submetido a estas forças está em equilíbrio de translação, mas está submetido a um momento. Neste caso, os raios polares extremos serão iguais, e o valor do momento será dado pelo produto do comprimento de um dos raios polares extremos pela distância entre as paralelas das cordas extremas.

As propriedades dos diagramas recíprocos estão ligadas ao fato destes serem representações gráficas de aspectos complementares da análise de forças. O polígono de forças trata da soma dos vetores, um PF fechado implica em um somatório de resultante nula para o sistema. O equilíbrio de translação pode ser assim verificado, porém a rotação, os binários e seus momentos precisam de informações quanto à posição no espaço, pontos de aplicação e linhas de ação das forças, fornecidas pela forma geométrica. Os raios polares relativos a pontos consecutivos do PF podem ser entendidos como componentes da força representada entre esses pontos (considerando o sentido a partir do polo para o primeiro e sentido em direção ao polo para o segundo). A utilização de intercessões entre linhas de ação do FG corresponde ao princípio da Estática que afirma, sobre o estudo dos corpos

rígidos, que qualquer ponto sobre a linha de ação de uma força pode ser adotado como ponto de aplicação produzindo o mesmo efeito. As propriedades dos diagramas recíprocos podem ser experimentadas através do algoritmo no Apêndice A.3, cujo funcionamento foi registrado e comentado no vídeo 01 - Propriedades dos PF, FG e Funiculares em anexo.

3.6 Problemas em Equilíbrio

Os problemas de Estática consistem em determinar os elementos desconhecidos (magnitudes, direções, sentidos, linhas de ação e ponto de aplicação) em sistemas de forças que estão sabidamente em equilíbrio. Johnson (1908) apresenta os problemas fundamentais da seguinte maneira:

1. encontrar a magnitude, direção e ponto de aplicação de uma força de um sistema;
2. magnitude de duas forças de um sistema;
3. magnitude de uma força e magnitude e direção de outra do mesmo sistema;
4. magnitude de três forças do mesmo sistema.

Uma abordagem didática, que permite ao leitores antever os valores desconhecidos que deveram ser determinados pela Grafostática em cada situação. As operações necessárias para encontrar esses valores, representam a solução dos quatro casos que formam a base dos problemas da Estática, cuja formulações e soluções gráficas serão apresentadas e comentadas em seguida.

3.6.1 Caso Um - Composição de uma Força

O primeiro caso é basicamente o cálculo da resultante apresentado na Seção 3.4. Mas é importante saber que, além da resultante, este problema fundamental também é usado para o cálculo da equilibrante do sistema. Partindo da Figura 3.2, o primeiro passo para encontrar a força que equilibra o sistema é justamente o cálculo da resultante, seguindo a sequência de instruções apresentada. Na forma geométrica, a linha de ação ae é a mesma para a resultante e equilibrante do sistema. No polígono de forças contudo, o equilíbrio se dá pela atuação da força de mesmo módulo e direção que a resultante AE porem com o sentido oposto. Portanto o vetor EA tem módulo, direção e sentido, bem como ponto de aplicação ea para a resolução do caso.

3.6.2 Caso Dois - Resolução de uma Força em Duas Componentes

O segundo caso é dividido em dois problemas: o primeiro trata de forças concorrentes, o segundo, considerado uma situação especial, trata de um conjunto de forças onde todas, inclusive as duas de magnitude desconhecidas, são paralelas.

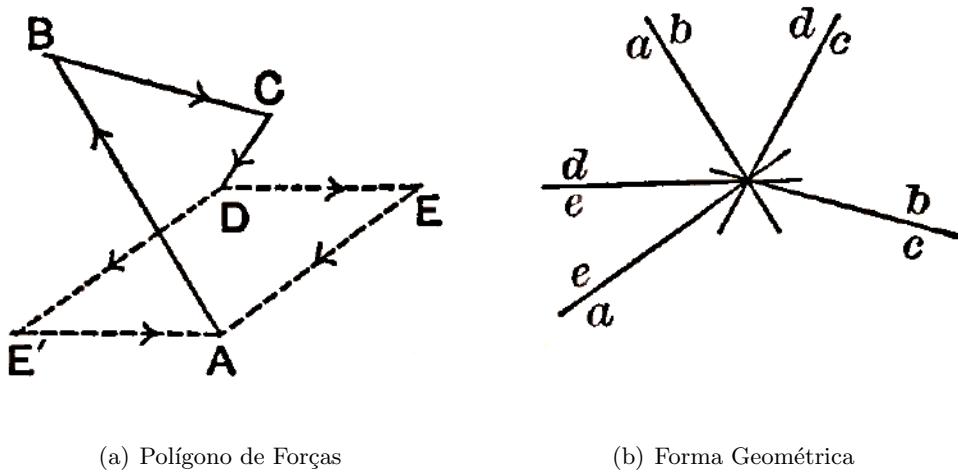
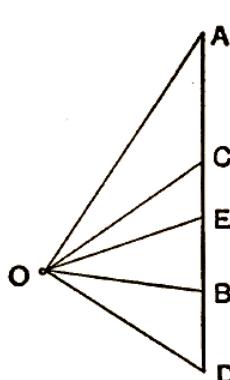


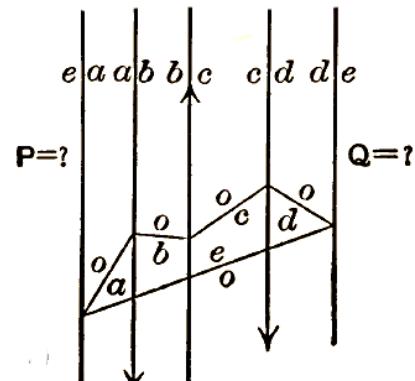
Figura 3.3: Caso 2 - forças concorrentes - fonte: [Hoskins \(1899\)](#)

No caso de forças concorrentes, ilustrado na Figura 3.3, onde são plenamente conhecidas as forças AB , BC e CD , e pretende-se encontrar as magnitudes de DE e EA . O primeiro passo é desenhar todas as linhas de ação no FG (Figura 3.3(b)). Em seguida, desenha-se as forças totalmente conhecidas no PF. Sabendo que o sistema está em equilíbrio, infere-se que o PF tem que ser um polígono fechado. Traçando uma paralela à de pelo ponto D e uma à ea pelo ponto A , observa-se que o ponto de encontro dessas duas retas é o ponto E . O PF se fecha e são determinadas as magnitudes DE e EA . O ponto E' pode ser construído traçando-se uma paralela à EA pelo ponto D e uma paralela à DE pelo ponto A e também é admitido como solução do problema, os mesmos vetores são encontrados em magnitude e sentido, tendo por direção as retas que cada uma delas é paralela. Deve-se contudo, atentar para que não haja conflito com a nomenclatura dos vetores nos diagramas recíprocos.

Para um sistema de forças paralelas (Figura 3.4) tomemos por vetores totalmente conhecidos AB , BC e CD , com suas respectivas linhas de ação traçadas na figura 3.4(b). No PF, traça-se as forças AB , BC e CD , bem como seus respectivos raios polares relativos à um polo O qualquer. As cordas paralelas aos raios polares conhecidos são desenhadas na FG. Para que o sistema esteja em equilíbrio de rotação, a corda eo deve interceptar ea e de nos pontos coincidentes com oa e od , fechando o funicular. Uma paralela à eo e desenhada pelo polo O , dividindo a resultante parcial AE em DE e EA , em magnitude



(a) Polígono de Forças



(b) Forma Geométrica

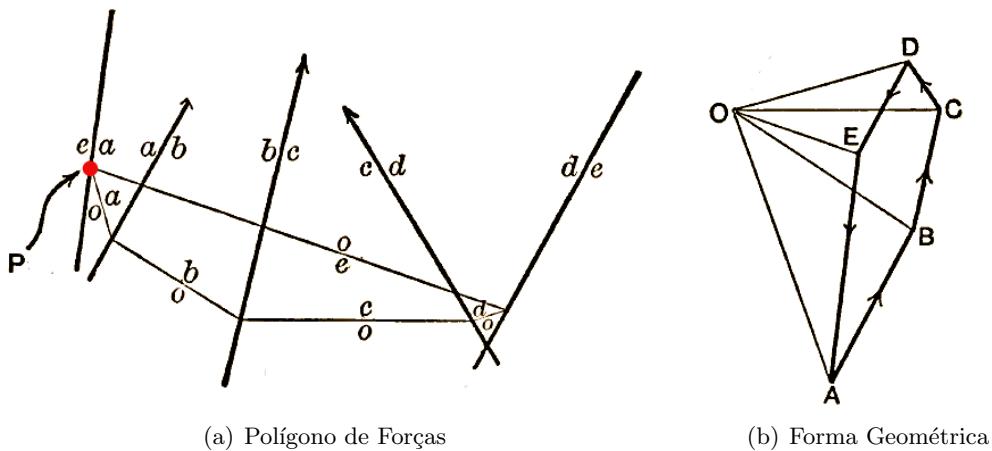
Figura 3.4: Caso 2 - forças paralelas - fonte: [Johnson \(1908\)](#)

e sentido. Essa construção é bastante usada para determinar reações dos apoios de vigas simples, como no exemplo mostrado na Figura 3.1.

3.6.3 Caso Três - Resolução de uma Força em uma Componente de Direção Conhecida e Outra que Passe por um Ponto

A construção mostrada no **caso dois**, quando aplicada no cálculo das reações de apoio para uma viga bi-apoiada, funciona apenas quando o carregamento não possui componentes horizontais, ou se estes são considerados dispensáveis na análise do modelo. Caso algum dos vetores não seja perpendicular ao eixo da viga, uma das possibilidades é admitir que um de seus apoios deve resistir também a esforços na direção axial. Embora seja conhecido o ponto de aplicação desta reação, sua linha de ação permanece desconhecida. Este terceiro caso básico é a forma genérica de resolução destes problemas.

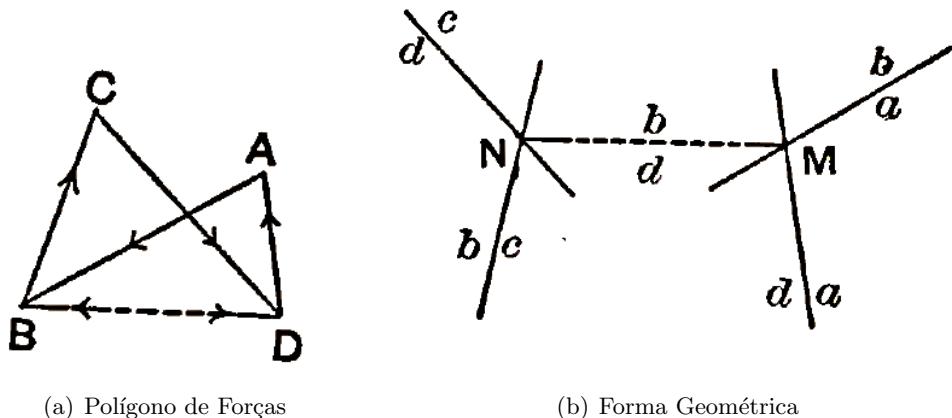
As forças AB , BC e CD são totalmente conhecidas, DE tem sua linha de ação definida por de e de EA conhecemos apenas o ponto de aplicação P (figura 3.5). Desenha-se o PF com as forças conhecidas por inteiro, com os seus respectivos raios polares. O traçado das cordas deve começar pelo ponto P , e segue pelo ponto de interseção entre oa e ab , até que a linha de ação dc . Para que a condição de equilíbrio de rotação seja atendida, a corda oe deve conter o ponto de encontro entre od e de , bem como o ponto P . Traçando as paralelas a oe , partindo do polo, e de , pelo ponto D , encontra-se E que fecha o PF obtendo o vetor EA . A linha de ação ea é determinada desenhandando uma paralela à EA pelo ponto P .

Figura 3.5: Caso 3 - fonte: [Johnson \(1908\)](#) (editado)

3.6.4 Caso Quatro - Resolução de uma Força em Três Componentes

No estudo das treliças, a maior parte das linhas de ação são definidas previamente pela proposta do desenho dos eixos das peças. A análise dos esforços pode ser feita através dos nós (método de Cremona), os pontos de união dos eixos das diagonais e banzos, ou pelo método das seções, tomadas de forma a cortarem até três elementos desconhecidos da treliça (método de Culmann). Estes últimos são resolvidos pelo caso quatro, que pode ser considerado uma combinação dos casos dois e três. Uma força aplicada à um ponto de um corpo rígido, tem o mesmo efeito que a aplicação da mesma força em qualquer outro ponto de sua linha de ação. Por essa afirmação da Estática, pode-se considerar o ponto de interseção entre duas das direções das forças parcialmente conhecidas como o ponto P e resolver o problema conforme descrito no caso três. Assim feito, o vetor calculado no ponto P é concorrente com as duas forças ainda não completamente conhecidas, situação que pode ser resolvida pelo caso dois.

De maneira simples, imagina-se que a força AB é a resultante das forças conhecidas do problema, grafadas no PF e ab é sua linha de ação no FG (Figura 3.6). Na forma geométrica, encontram-se os pontos de interseção das linhas de ação das quatro forças (as três desconhecidas e ab), duas à duas. No ponto M a linha de ação da resultante das forças conhecidas ab encontra da e cd e bc tem por N seu ponto em comum. Para que as condições de equilíbrio sejam satisfeitas, as resultantes de bc e cd precisam ter a mesma linha de ação da composição de ab e da (como mesmo módulo e sentido contrário). Uma paralela à NM traçada em B representa a direção da composição de BC e CD e encontra a paralela à da traçada em A , definindo o vetor DA em módulo e sentido. O encontro de uma paralela à bc , traçada em B , com uma paralela à cd , pelo ponto D define o ponto C que fecha o PF, garantindo equilíbrio de translação para o sistema. É importante notar que o segmento BD na Figura 3.6(a) não representa nenhuma força descrita no problema,

Figura 3.6: Caso 4 - fonte: [Hoskins \(1899\)](#)

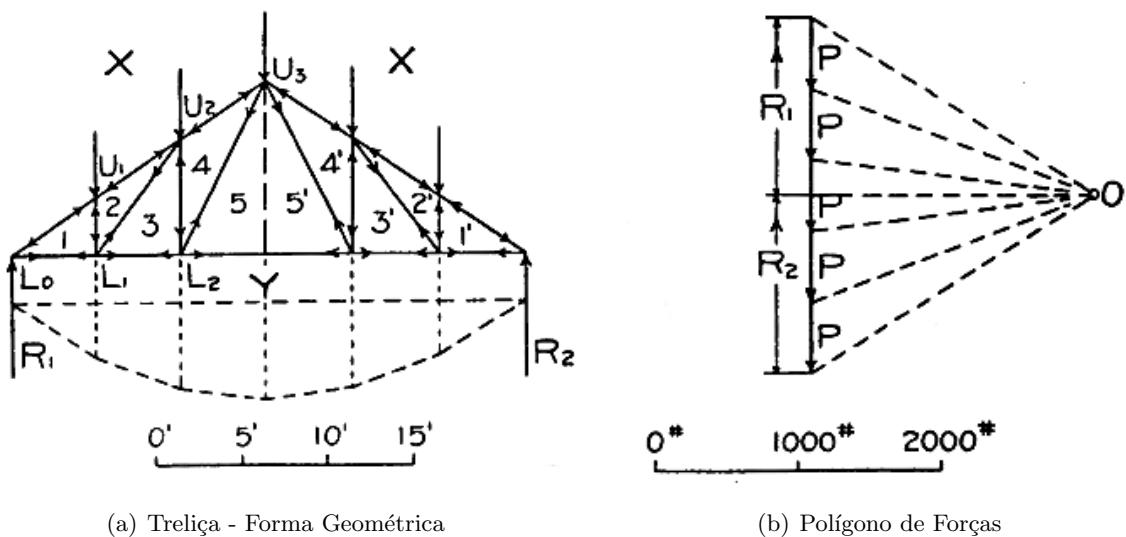
é apenas uma construção auxiliar para completar o desenho do PF. No sentido DB , é a composição de DA e AB , seu inverso, BD é a resultante de BC e CD . Iguais em magnitude e direção, como exige a Estática.

3.7 Cálculo das Reações e Esforços em Treliças Planas Bi-Apoiadas

[Malcolm \(1909\)](#) apontava o método gráfico como o mais conveniente para o cálculo das tensões em treliças de cobertura, tanto pela velocidade, quanto pelo fato das construções geométricas fornecerem maneiras de se verificar os resultados. A Figura 3.7(a) é apresentada como um exemplo ilustrativo do cálculo das reações e esforços. Partindo do desenho dos eixos dos elementos da treliça e conhecendo totalmente os vetores dos carregamentos, pode-se afirmar que todas as linhas de ação das forças no FG são conhecidas.

3.7.1 Determinação Estática de Treliças

Antes de utilizar-se um modelo matemático para a resolução de um problema, é preciso certificar-se que o modelo se aplique adequadamente ao problema. Os métodos gráficos apresentados aqui lidam com problemas da Estática Pura. Definida como o campo da ciência que trata dos problemas da Estática passíveis de solução apenas pela satisfação das condições de equilíbrio dos corpos rígidos ([JOHNSON, 1908](#), p. 45). Na literatura atual é mais comum encontrar os termos isostático e hiperestático para definir os principais tipos de problema do cálculo estrutural. Quando as equações de equilíbrio de um corpo rígido são de mesma quantidade que as variáveis envolvidas, diz-se que o problema

Figura 3.7: Exemplo de Cálculo de Esforços de Treliças - fonte: [Malcolm \(1909\)](#)

é isostático. Quando existem mais variáveis que equações diz-se que o problema é hiperestático ([MCCROMAC, 2009](#)). Para resolver um sistema estrutural hiperestático, são adicionadas equações ao problema, provenientes das características da deformações dos materiais de construção empregados (lei de Hooke). Portanto, dizer que um problema é isostático ou que ele pertence ao escopo da Estática Pura tem o mesmo significado.

A isostaticidade de uma treliça é determinada por duas características e uma condição de igualdade. As características são:

- os elementos devem formar apenas triângulos pelo encontro de seus eixos;
- dois ou mais eixos devem se encontrar apenas em um nó.

A condição de igualdade apresentada na Equação 3.2 ([MCCROMAC, 2009](#), p. 99) deve ser satisfeita, onde m é o numero de barras (membros) da treliça e j é o número de nós.

$$m = (2 \times j) - 3 \quad (3.2)$$

No exemplo em questão, nota-se por inspeção da forma, que as duas características são satisfeitas e que existem 21 elementos e 12 nós. A condição de igualdade ($(12 \times 2) - 3 = 21$) é atingida, portanto a treliça é isostática e pode ser resolvida dentro do escopo da Estática Pura.

3.7.2 Regras de Desenho para Obtenção de Diagramas Recíprocos

Antes de prosseguir com a descrição do algoritmo de desenho, algumas regras devem ser assumidas para que as figuras geradas sejam realmente recíprocas. As propriedades que garantem a reciprocidade das figuras são [Hoskins \(1899, p. 65\)](#):

- Para cada conjunto de linhas que se interceptam em um ponto na FG, são representadas no PF como um polígono fechado com paralelas às respectivas linhas na outra figura.
- A ordem das forças representadas no PF é deve ser a mesma das linhas da FG, se observadas consecutivamente ao redor do nó.

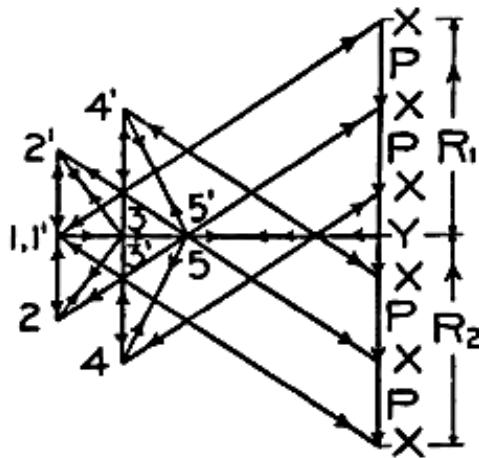
Para que a segunda propriedade seja plenamente atingida é necessário estabelecer um sentido de seleção para as forças conhecidas e linhas de ação que deve ser adotado tanto para o cálculo das reações quanto para os esforços nas barras. O sentido adotado neste exemplo é horário, o sentido anti-horário poderia ser também utilizado, obtendo os mesmos valores para as incógnitas, desde que apenas um desses sentidos possíveis seja utilizado em todas as etapas do processo.

3.7.3 Cálculo das Reações

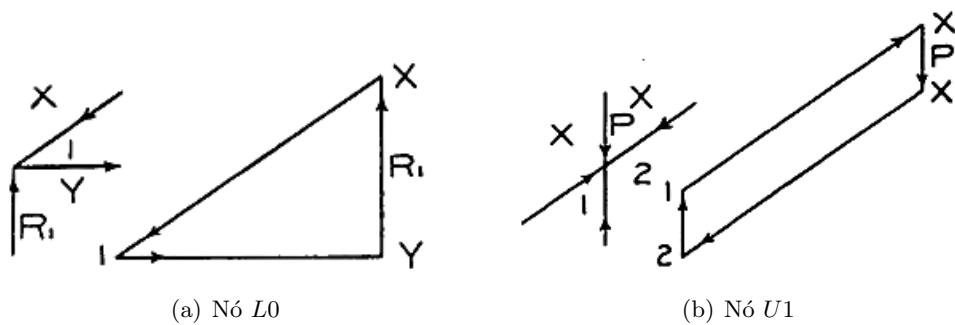
As reações de apoio R_1 e R_2 são as únicas forças externas desconhecidas e devem ser calculadas primeiro. Como apenas carregamentos verticais são aplicados à treliça, pode-se afirmar que as duas reações tem direções também verticais. Neste caso, aplica-se o problema fundamental da Estática chamado de caso dois para forças paralelas, descrito na Seção 3.6.2. A seleção das forças deve seguir o sentido eleito, partindo da força mais próxima de R_1 até R_2 . Os raios polares e o funicular são desenhados (linhas pontilhadas na Figura 3.7) e as reações são definidas. A ordem estabelecida garante que a componente R_1 será a porção do somatório de forças mais próxima do carregamento no nó U_1 . Analogamente ao cálculo analítico, as reações são determinadas pela satisfação da premissa que garante o equilíbrio de rotação do sistema.

3.7.4 Cálculo dos Esforços

Além da obtenção dos módulos e sentidos dos esforços em cada elemento, do ponto de vista gráfico, partindo dos carregamentos e resultantes calculados na Figura 3.7(b), procura-se construir um único polígono de forças que contenha elementos recíprocos à todas as linhas

Figura 3.8: Polígono de Forças - fonte:[Malcolm \(1909\)](#)

de ação da forma geométrica, como mostrado na Figura 3.8. Utilizando-se do método de Cremona, que procura satisfazer as condições de equilíbrio de translação para cada nó, deve-se iniciar por um ponto onde no máximo duas forças não estejam completamente determinadas. Geralmente pelos pontos de apoio, onde atuam as reações R_1 e R_2 .

Figura 3.9: Satisfação das Condições de Equilíbrio de Translação - L_0 e U_1 - fonte: [Malcolm \(1909\)](#)

Aplicando a condição de equilíbrio no nó L_0 , utilizando o sentido de seleção das forças ao redor do ponto, deve-se ordenar os elementos atuantes de forma que as duas forças desconhecidas sejam usadas por último. Analisando o nó em questão, apenas a reação R_1 é completamente conhecida. A partir de R_1 no PF, desenha-se o sentido do elemento X_1 na extremidade final e a direção do elemento $1Y$ no primeiro ponto, como ilustrado na Figura 3.9(a). Para que a condição de equilíbrio seja completamente atendida, os vetores devem descrever um polígono fechado também em suas direções, portanto X_1 tem sentido descendente e $1Y$ da esquerda para a direita. A definição da natureza dos esforços baseia-se nos critérios apresentados em [Malcolm \(1909, p. 99-102\)](#). Movendo as forças X_1 e $1Y$ do PF para a FG pelos seus pontos finais até o ponto do nó, a natureza do esforço sera de compressão, caso haja sobreposição entre os elementos do FG e o seguimento recíproco

movido do PF, como acontece com a força $X1$. Caso os segmentos não se sobreponham, como no caso da força $1Y$, a solicitação é de tração. O procedimento de determinação dos sentidos e natureza dos esforços deve ser realizado para todas as solicitações calculadas.

Determinando todas as solicitações que atuam em $L0$, restam apenas dois valores desconhecidos no nó $U1$. Aplicando o sentido de seleção, partindo das forças conhecidas procura-se os vetores $1X$ ($X1$ e $1X$ representam o mesmo elemento segundo a notação de Bow, que leva em conta o sentido de seleção para determinar a nomenclatura em cada nó) e o carregamento P . Desenha-se as direções dos elementos $X2$ e 21 , partindo respectivamente do ponto final e inicial do polígono aberto das forças conhecidas neste ponto. no ponto de encontro dessas duas direções fecha-se o polígono determinam os sentidos (Figura 3.9(b)).

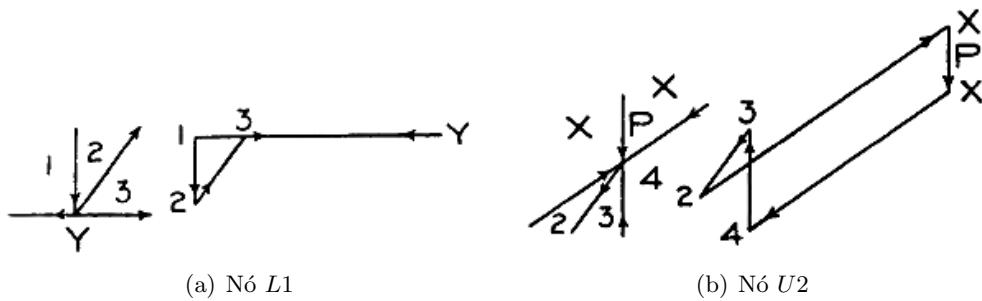


Figura 3.10: Satisfação das Condições de Equilíbrio de Translação - $L1$ e $U2$ - fonte: [Malcolm \(1909\)](#)

A determinação completa das forças no nó $U1$ abre caminho para resolução de $L1$. Pelos critérios de seleção adotados, procura-se por $Y1$ e 12 no PF (elementos completamente conhecidos) e desenha-se a direção de 23 no ponto final de 12 e $3Y$ no início de $Y1$. Como $Y1$ e $3Y$ tem a mesma direção, ocorre uma sobreposição de linhas, como mostra a Figura 3.10(a).

A soma de vetores é uma operação comutativa, independem da ordem em que os vetores são selecionados. Obedecendo o sentido de seleção, contudo, garante-se um único polígono de forças para todos os nós. Analisando $U2$ nota-se que as forças conhecidas 32 , $2X$ e P estão grafadas no PF pelos passos anteriores, $X4$ e 43 , quando determinadas nos pontos final e inicial respectivamente do polígono aberto (Figura 3.10(b)) garantem que os vértices seguintes que dependem dessas forças encontraram suas representações no PF nas posições convenientes para prosseguir com o cálculo de um PF único.

Em $L2$ as forças conhecidas $Y3$ e 34 são identificadas no PF e determina-se as magnitudes e sentidos desconhecidos para 45 e $5Y$ (Figura 3.11(a)). Em $U3$, pode-se calcular as forças desconhecidas $x4'$ e $4'5'$ efetuando os mesmos passos a partir do vértice onde atua $R2$, mas como tanto a forma da treliça, quanto os carregamentos são simétricos em relação

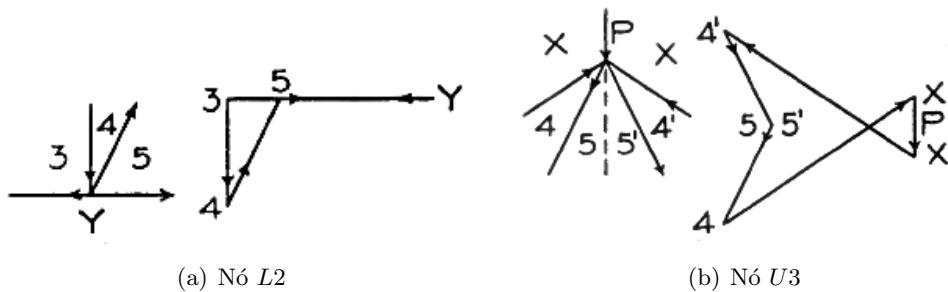


Figura 3.11: Satisfação das Condições de Equilíbrio de Translação - *L2* e *U3* - fonte: [Malcolm \(1909\)](#)

ao eixo $5'5$, os resultados para cada elemento apresentarão mesmo módulo e natureza dos elementos simétricos já calculados. O PF final apresentará também simetria, como grafado na Figura 3.8. O cálculo de $5'5$ é apresentado na Figura 3.11(b) e ocorre pela aplicação do caso um dos problemas fundamentais da Estática. o PF desenhado pelas demais forças que atuam em *U3* já descrevem um polígono fechado, implicando na nulidade da magnitude de $5'5$. Optando por não colocar um elemento $5'5$, as condições de isostaticidade apresentadas na Seção 3.7.1 ainda seriam satisfeitas com 19 elementos e 11 vértices $((11 \times 2) - 3 = 19)$.

3.8 Cálculo das Reações e Esforços em Treliças Planas em Balanço

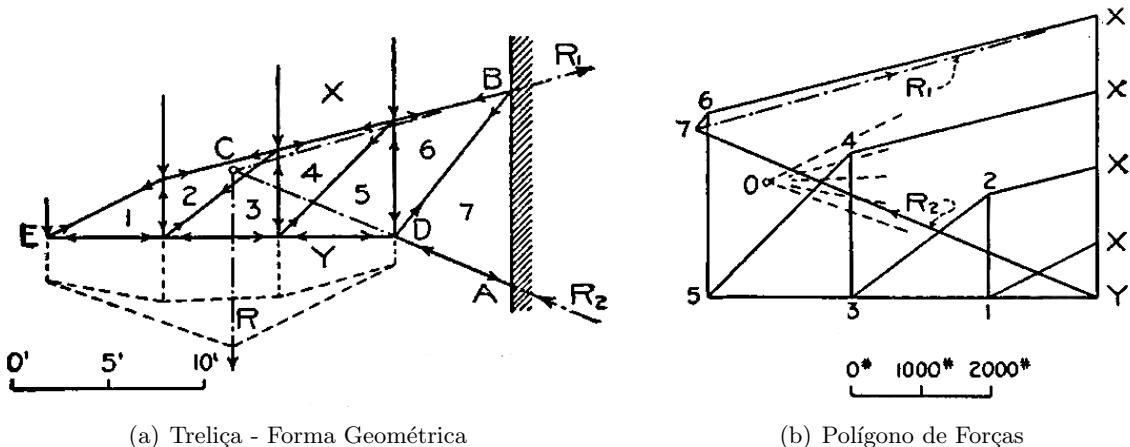


Figura 3.12: Exemplo de Cálculo de Esforços de Treliças em Balanço- fonte: [Malcolm \(1909\)](#)

O método apresentado nesta seção baseia-se nos algoritmos apresentados em [Burr e Falk \(1908, p. 145-159\)](#), [Malcolm \(1909, p. 114-116\)](#)[Gondenhórn \(1943, p. 161-172\)](#) ou [Zalewski e Allen \(1998, p. 132-133\)](#). Para o cálculo das reações na treliça em balanço apresentada na Figura 3.12(a), deve-se primeiro analisar as reações R_1 e R_2 . Enquanto a

segunda tem uma direção definida pelo seguimento AD , a primeira tem direção desconhecida. A resolução usual deste problema parte do cálculo da resultante dos carregamentos ignorando as reações.

Pelo caso um dos problemas fundamentais da Estática, a reação R é calculada pelo fechamento do funicular. As reações $R1$ e $R2$ tem por soma R , prolongando o sentido de $R2$ até encontrar a linha de ação de R no ponto C . Para satisfação das condições de equilíbrio, $R1$ deve ter direção BC . Atendendo os critérios de seleção e sentido (adotado o anti-horário neste exemplo) em torno do ponto R as reações são completamente determinadas, sendo $R1$ tração e $R2$ compressão. O esforço no elemento AD também é determinado, visto que este precisa se anular com a reação $R2$ em torno do nó A , tratam-se de forças com mesmo módulo e direção, mas com sentidos opostos no PF.

A resolução dos demais elementos da treliça deve ser executada a partir do ponto B e seguir em direção ao ponto E , respeitando os mesmos critérios de seleção, sentido e construções geométricas aplicadas na resolução da treliça apresentada na Seção 3.7.

O nó E , entretanto, não chega a ser formalmente calculado, e sim serve como ferramenta de conferência do Cálculo. Em torno deste nó atuam o carregamento YX , a solicitação no elemento $X1$ do banzo superior e em $1Y$ no inferior. Enquanto o carregamento é um dado do problema, $1Y$ é calculado fechando o polígono para o nó imediatamente anterior à E no banzo inferior e $X1$ para o nó imediatamente anterior à E no banzo superior. Quando o elemento $X1$ é definido em magnitude e sentido, ele deve fechar os polígonos de forças ao redor do ponto em que foi calculado e também em torno do ponto E . Se os dois polígonos se fecham com uma linha paralela à $X1$ na FG, o problema foi resolvido corretamente.

3.9 Outros Algoritmos da Grafostática

Apesar das publicações e republicações sobre Grafostática terem sido drasticamente reduzidas na segunda metade do século XX, a quantidade de material disponível sobre o assunto é muito grande. A real dificuldade, na ausência de um levantamento bibliográfico sobre os métodos de solução para diferentes problemas da Estática Gráfica, é filtrar o material existente. Sem a pretensão de realizar um levantamento extensivo, serão listados nesta seção alguns tópicos que merecem destaque, pela clareza da exposição, pelo cuidado com a apresentação da teoria das estruturas ou por apresentarem algoritmos pouco explorados por seus pares, nos trabalhos dos autores que forneceram a fundamentação para o estudo da Grafostática realizado nesta pesquisa.

Dentre os livros técnicos e manuais práticos consultados para a elaboração deste tra-

balho (muitos deles com os direitos de publicação e reprodução já em domínio público) encontram-se disponíveis para consulta e *download* em repositórios na internet (e.g. [openlibrary.org](#)), outros podem ser encontrados em território nacional, no comércio *online* de livros usados.

Os algoritmos fundamentais, os cálculos e diagramas de solicitações e tensões básicas para treliças e vigas de alma cheia são assuntos cobertos por todas as publicações. Destacam-se o livro de [Hoskins \(1899\)](#), pela abordagem objetiva e introdutória das bases da disciplina e [Johnson \(1908\)](#) pela cuidadosa apresentação dos conceitos teóricos e o constante paralelo com os métodos analíticos para os mesmos problemas. O livro de [Burr e Falk \(1908\)](#) tem um capítulo inteiro dedicado à teoria das linhas de influência aplicada nos algoritmos de cálculo e análise, outro sobre deformações calculadas graficamente. [Kneip, Winzer e Killmann \(1939\)](#) apresentam o tratamento gráfico de eixos e gruas, com suas aplicações ao projeto de peças mecânicas. [Henkel \(1953\)](#) trabalha com algoritmos de deformação no cálculo de equilíbrio em vigas continuas. [Eddy \(1877\)](#) trata de tensões provenientes da dilatação térmica e estruturas em forma de cúpula, em metal e pedra. [Malcolm \(1909\)](#) discute a análise e o cálculo de tensões para um série de treliças de cobertura, inclusive estruturas em balanço e assimétricas. [Gondenhórn \(1943\)](#) apresenta dois métodos para o cálculo de vigas continuas e o tratamento gráfico das propriedades das seções planas (incluindo o momento de inércia e o raio de geração). [Sanjuan \(1951\)](#) mostra métodos de desenho para a determinação gráfica de integrais no final de seu livro, com suas aplicações na resistência dos materiais em sistemas isostáticos e hiperestáticos, mostrando a extensão do poder de cálculo desses algoritmos. [Panseri \(1968\)](#) trata, de maneira extensa e cuidadosa com os aspectos teóricos da Física, as propriedades de seções planas, apresentando os métodos de Culmann e Mohr para o cálculo destas propriedades. As aplicações das derivadas e integrais gráficas no tratamento dos esforços cortante e momento fletor abordadas pelo autor, também merecem destaque. [Bayle \(1966\)](#) apresenta objetivamente os aspectos básicos da Grafostática com a particularidade de, ao final de cada capítulo, mostrar de maneira resumida as correlatas aplicações analíticas e métodos de verificação do cálculo gráfico pela utilização de expressões matemáticas. O trabalho teórico de [Crapo \(1979\)](#) é considerado fundamental ([GERHARDT; KURRER; PICHLER, 2003](#)) ([LACHAUER; KOTNIK, 2010](#)) para a formulação do estudo tridimensional da Grafostática. O livro de [Zalewski e Allen \(1998\)](#) é uma publicação recente sobre o tema, largamente ilustrada onde a Estática Gráfica é utilizada como ferramenta para a criação e refino de forma, demonstra o renovado interesse de instituições respeitadas de ensino e pesquisa como o MIT e a Universidade de Yale a respeito do método. A única publicação nacional encontrada nesta pesquisa foi a de [Costa \(1974, p. 3-39\)](#) que, apesar de utilizar a Grafostática com recurso introdutório para o estudo analítico de Resistência dos Materiais, apresenta em poucas páginas os fundamentos básicos da disciplina de maneira clara e objetiva.

3.10 Ferramentas Computacionais

- **ActiveStatics:** Desenvolvido por Simon Greenwold no MIT ([GREENWOLD](#)), é considerada a primeira implementação computacional baseada em Grafostática. Idealizada como ferramenta de ensino, o programa implementa alguns algoritmos gráficos de Estática que podem ser dinamicamente experimentados ([SHEARER, 2010](#));
- **eQUILIBRIUM:** Implementação na plataforma educacional *Open source* GeoGebra de alguns métodos da Estática Gráfica pelo [BLOCK Research Group](#) da Politécnica de Zurique;
- **CADenary:** Uma ferramenta baseada em Grafostática implementada na linguagem de *scripts Processing* ([KILIAN](#));
- **RhinoStatics:** Desenvolvido como parte da dissertação de mestrado de Shearer ([2010](#)), ferramentas Grafostáticas são desenvolvidas sobre a forma de *scripts*, em linguagem *Visual Basic* no ambiente *Rhinoceros*. As soluções são construídas através de funções que manipulam entidades gráficas. Qualquer alteração feita nas geometrias, porem, deve ser redesenhada por novas chamadas das funções do programa;
- Os algoritmos propostos por [Lachauer, Jungjohann e Kotnik \(2011\)](#), a única ferramente computacional baseada em Grafostática, implementada em um ambiente altamente dinâmico como o *Grasshopper*, que esta pesquisa encontrou. Aplicados como modelo de análise e geração *form finding*, os algoritmos são propostos em um ambiente (*Rhinoceros*, *Grasshopper*, *VB.net*) semelhante ao utilizado neste experimento (*Rhinoceros*, *Grasshopper*, *Python*).

Ambiente de Implementação

*"If you want to teach people a new way of thinking,
don't bother trying to teach them.*

*Instead, give them a tool, the use of which
will lead to new ways of thinking."*

(R. Buckminster Fuller)

A ideia de aproximar a programação de formas e análises à atividade de projeto vem ganhando força e espaço, principalmente nos grandes escritórios de arquitetura ao redor do mundo. O *"Specialist Modeling Group"* do escritório *Foster and Partners'* ou o *"CODE unit"* de *Zaha Hadid Architects* são exemplos de departamentos de programação, criados em firmas de arquitetura para trabalhar em cooperação e sincronia com os demais setores, na elaboração de propostas e projetos ([CECCATO, 2010](#)).

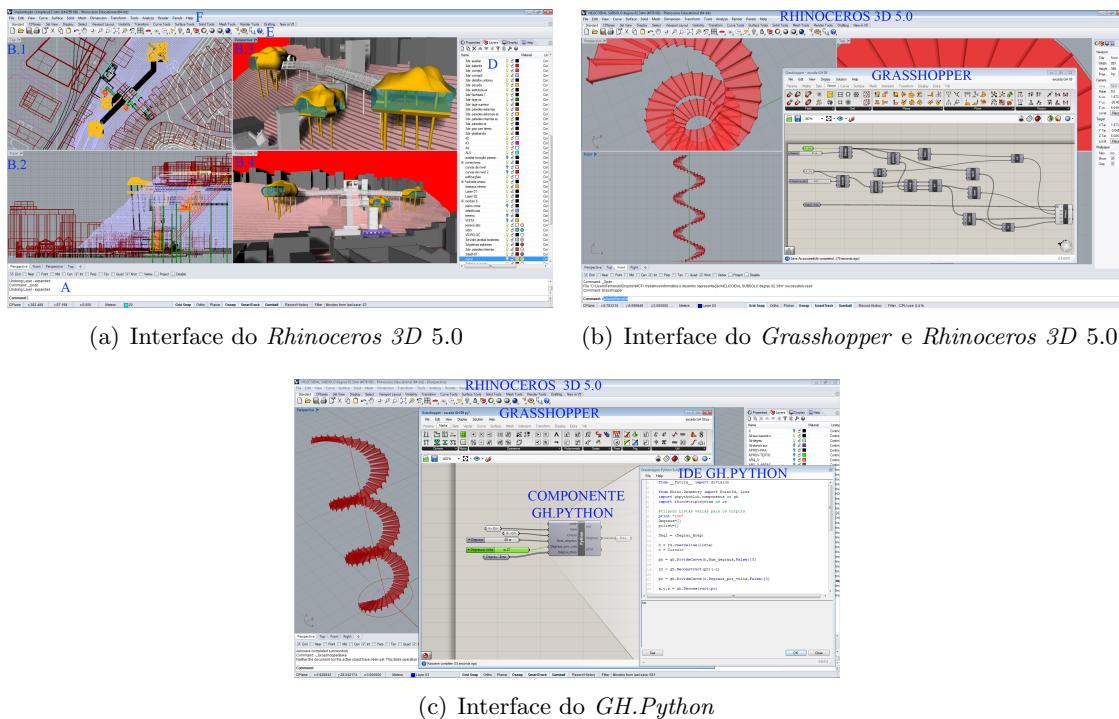


Figura 4.1: Ambiente de Implementação

As demandas de programação entretanto, devem se adaptar aos cronogramas e prazos da atividade fim. Por isso, as linguagens de código e *scripts* devem ter fácil interação com os ambientes computacionais de projeto, permitindo trabalho rápido, eficiente, manipulação de entidades gráficas vetoriais e reaproveitamento de algoritmos em diferentes atividades.

Dentre os muitos possíveis sistemas **CAD** que poderiam ser utilizados para a implementação do experimento, foi escolhido o *Rhinoceros 3D* 5.0 e o *plug-in Grasshopper*, com ênfase no ambiente de *Python Script* suportado por ambos (Figura 4.1).

4.1 Rhinoceros 3D

O *Rhinoceros 3D*, assim como o *Autodesk Autocad*, *BricsCAD*, *LibreCAD*, entre outros, é um **CAD** de interface baseada em linha de comando (marcada com a letra "A" na figura 4.1(a)). Com características mais adaptadas à modelagem 3D, principalmente de curvas e superfícies complexas, é usado em diversos ramos do *design* industrial e foi adotado pelo mercado de arquitetura (CECCATO, 2010), desempenhando papéis importantes nas etapas iniciais de concepção de projeto. Um dos motivos que levaram à esse uso está no fato do programa ser construído como um conjunto de bibliotecas gráficas envoltas por uma interface CAD, potencializada por *plug-ins*. A principal dessas bibliotecas é a *Open-Nurbs*, desenvolvida e mantida pela empresa responsável pelo *Rhinoceros* e *Grasshopper* (*Robert McNeel & Associates*) porem com licença *open source*, nela estão os métodos de gravação e leitura dos arquivos .3dm e a representação das formas geométricas do programa. O desenvolvimento de novas funcionalidades do *software* são feitas através de *plug-ins*, usando o mesmo IDE disponível para programadores terceiros, as funções e métodos de gravação e manipulação de entidades gráficas podem ser acessadas tanto em C++, a linguagem de programação em que o programa é escrito, quanto em linguagens interpretadas (*C#*, *VB.net* e *Python*) que permitem mais rápida criação e manipulação de algoritmos (TIBBITS; HARTEN; BAER, 2011).

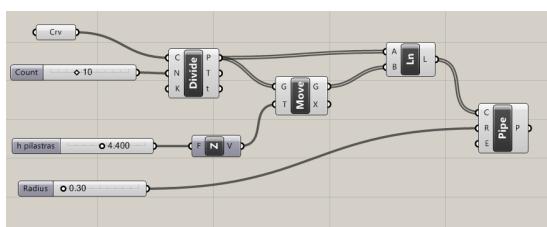
Essas características atraíram também o interesse de trabalhos acadêmicos e pesquisas, que usam a plataforma para implementação e testes de seus algoritmos. (TESSMANN, 2008), (BLOCK, 2009), (VETTORETTI, 2010), (SHEARER, 2010), (STROBBE et al., 2011), (LACHAUER; JUNGJOHANN; KOTNIK, 2011), (DALVI, 2011), (BLOCK; LACHAUER, 2014).

4.2 Grasshopper

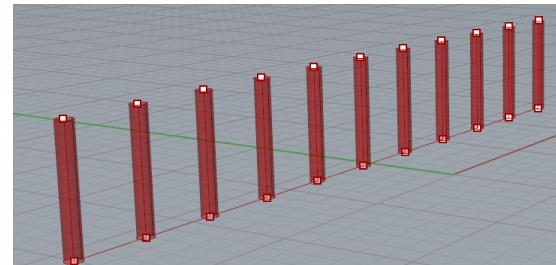
Um dos *plug-ins* desenvolvido pela *Robert McNeel & Associates* para expansão das capacidades do *Rhinoceros 3D* é o *Grasshopper*, uma interface visual de programação e criação de modelos paramétricos (LACHAUER; JUNGJOHANN; KOTNIK, 2011). A interface do *plug-in*, mostrada na Figura 4.1(b), permite a elaboração de algoritmos pela interligação de componentes, também chamados de baterias (*batteries*), que desempenham papéis de armazenamento de informações (variáveis e objetos), ou de funções que

manipulam objetos geométricos. Existem três aspectos importantes da integração entre programa e *plug-in* que precisam ser entendidos:

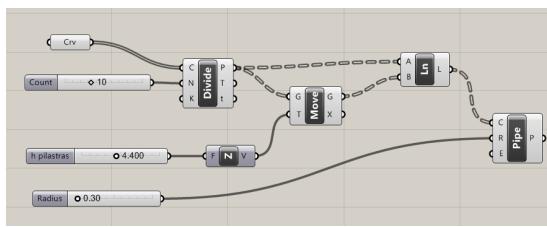
- **Comunicação Rhinoceros/Grasshopper:** todas as geometrias desenhadas no *Rhinoceros*, pontos, curvas, superfícies, sólidos e combinações de elementos, podem ser selecionados a partir de um componente do *Grasshopper*, usadas como variáveis ou objetos, e manipuladas pelos componentes seguintes do algoritmo proposto.
- **Ambiente de Modelagem do Grasshopper:** todas as formas geradas pelos algoritmos do *Grasshopper* são, por padrão, mostradas na janela no *Rhinoceros*, mas apenas os elementos desenhados no *Rhinoceros* podem ser editadas por este. As geometrias criadas pelo algoritmo são dinamicamente alteradas pela edição de uma geometria selecionada em uma bateria do *plug-in*, ou pela alteração de qualquer elemento do algoritmo. Essa velocidade de geração das variações é que torna o sistema interessante como ferramenta para avaliação de alternativas de projeto.
- **Comunicação Grasshopper/Rhinoceros:** qualquer geometria gerada pelos componentes do *Grasshopper* pode ser copiada, em uma configuração específica dos parâmetros que a geraram, como uma geometria do *Rhinoceros*, podendo então, ser editada como qualquer outra.



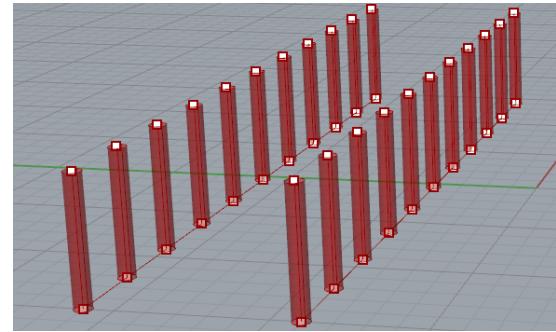
(a) Exemplo 1 - Grasshopper



(b) Exemplo 1 - Rhinoceros 3D



(c) Exemplo 2 - Grasshopper



(d) Exemplo 2 - Rhinoceros 3D

Figura 4.2: Grasshopper - Componentes e listas

Outro conceito importante é que a grande maioria das entradas e saídas dos componentes são tratados como listas numeradas. Cada elemento de uma lista com N valores recebe

uma numeração, chamada de índice, partindo de zero até $N - 1$. A Figura 4.2 mostra o mesmo algoritmo com apenas uma reta armazenada no componente ”*Crv*” (Figura 4.2(a)) e com duas curvas selecionadas no mesmo componente (Figura 4.2(c)), bem como as respectivas geometrias resultantes (Figuras 4.2(b) e 4.2(d)). A organização padrão dos componentes do *Grasshopper* é que as entradas e saídas são feitas pela ”conexão de fios” no sentido esquerda-direita. Nos exemplos da Figura 4.2, o componente ”*Divide*” recebe como entrada a(s) curva(s) do componente ”*Crv*” (segmentos de retas desenhadas no *Rhinoceros*, selecionadas e armazenadas) e um valor inteiro no componente ”*Count*”. Os segmentos selecionados em ”*Crv*” são lidos pela entrada *C* do componente ”*Divide*” e são divididos em N partes iguais, tendo o número inteiro fornecido por ”*Count*” como a quantidade de partes. Se na esquerda do componente ”*Divide*” se conectam as listas de entrada, na direita são lidas as saídas. No exemplo em questão, apenas a saída *P*, com lista(s) de pontos que dividem o(s) seguimento(s), é usada. Os componentes ”*Count*”, ”*h pilastas*” e ”*Radius*” são chamados de *sliders*, pois o valor numérico passado por estes pode ser alterado deslizando o elemento gráfico que marca sua posição. enquanto o primeiro desses é uma variável inteira, os outros são números reais que controlam respectivamente a altura dos elementos cilíndricos e seu raio. Uma diferença visível entre o mesmo algoritmo nas Figuras 4.2(b) e 4.2(d) é a maneira como os ”fios” que conectam as baterias são representados. Na primeira vemos uma linha simples ligando ”*Crv*” à ”*Divide*” representando que apenas um elemento (ou uma lista unitária) é passado como entrada para o componente, na saída *P* uma linha dupla indica que uma lista com $N + 1$ pontos (os pontos que dividem o segmento em N partes contando com os extremos deste). Na segunda, uma linha dupla indica que mais de um elemento estão armazenados em ”*Crv*”, gerando duas listas numeradas (uma para cada conjunto de $N + 1$ pontos que dividem os segmentos) na saída *P*, neste caso, a primeira curva selecionada recebe o índice zero (0), e a segunda é numerada como um (1). As linhas duplas e tracejadas que saem de *P* no segundo exemplo são usadas para indicar que um conjunto de listas numeradas (*Data Tree*) está sendo transmitido entre as saídas e entradas das respectivas baterias.

O componente ”*Move*” recebe a(s) Lista(s) de pontos e um vetor para move-los oriundo da combinação do *slider* ”*h pilastras*” com um vetor unitário na direção ”*Z*”. Cada elemento da lista *P* é movido em módulo, direção e sentido do vetor e a saída *G* é preenchida com lista(s) cuja numeração corresponde aos respectivos índices dos elementos à serem movidos na entrada *G*. A bateria ”*Ln*” cria segmentos de reta (pares ordenados de pontos) entre os valores recebidos pelas entradas *A* e *B*, segundo a ordem dos índices de cada elemento. A bateria ”*Pipe*”, por fim, cria as geometrias equivalentes à translação de um círculo, cujo raio é definido pela entrada *R*, ao longo da(s) curva(s) que alimentam a entrada *C*.

A vantagem do sistema descrito é que a simples alteração do valor de um *slider* ou a edição e/ou substituição de um objeto geométrico selecionado é imediatamente reprocessada pelo *constraint solver* que calcula as geometrias no *plug-in*, proporcionando uma rápida

visualização das alternativas de projeto. Lógicas semelhantes podem ser encontradas no editor paramétrico de texturas do *Software Maya 3D* ou na modelagem paramétrica do pacote de efeitos especiais para o cinema *Houdini*. Na área de arquitetura, o *plug-in* *Dinamo*, para *Revit* e *Vassari*, apresenta interface e lógica semelhantes ao *Grasshopper*. As ferramentas de criação de algoritmos do *plug-in* se aproximam dos paradigmas que originaram a linguagem de programação Lisp, presentes no artigo clássico de [McCarthy \(1960\)](#), no que tange as listas e no aspecto não destrutivo desta manipulação (no exemplo da Figura 4.2, vê-se que os pontos transformados pelo componente "Move" não destroem a lista de pontos que serve de entrada), se afastam porém, no que diz respeito ao conceito de recursividade (uma das formas de se criar um *loop* em programação). O paradigma Lisp é frequentemente associado à criatividade na programação. [Steele e Gabriel \(1993\)](#) relacionam este fato ao perfil dos usuários da linguagem (bastante aplicada à área de inteligência artificial). De maneira otimista, pode-se esperar que criatividade também apareça na manipulação de listas por profissionais de projeto.

Não existe uma maneira de se implementar um *loop* com os componentes do pacote básico do *Grasshopper*. O componente *Hoopsnake*, desenvolvido por [Chatzikonstantinou \(2014\)](#) e instalado como um *Add-on*, foi usado nos exemplos apresentados no Apêndice B.1 para submeter às transformações geométricas à um *loop* semelhante ao *For* com um contador como critério de parada. A manipulação de listas é a principal ferramenta lógica de programação do *plug-in*. No Apêndice A.1 temos um teste feito com os componentes padrão do *Grasshopper*, modelando a soma de vetores e o cálculo da equilibrante. O recurso encontrado para a colocação dos vetores em posição de soma foi:

1. Criar uma lista de segmentos de reta que ligam os vetores com o ponto de coordenadas 0,0,0;
2. Mover os vetores tendo como parâmetro as respectivas linhas que o ligam à origem do espaço cartesiano;
3. Extrair uma lista dos extremos finais dos vetores movidos. Os valores de x,y,z nestes extremos, correspondem às componentes dos respectivos vetores decompostos nas 3 direções;
4. O ponto inicial do PF é acrescentado no início da lista de extremos. O ponto final da mesma lista é retirado;
5. Um componente de soma em massa é utilizado. Ele recebe a lista gerada no passo anterior e retorna uma lista contendo: no índice zero, o valor do índice zero da lista original; no índice um a soma dos valores de índices zero e um; e assim sucessivamente, até que no índice N temos a soma dos N valores na lista inicial.
6. Os vetores são movidos a partir do seu ponto inicial, até cada um dos pontos da lista obtida no passo anterior.

Apesar do interessante desafio lógico de se estabelecer um algoritmo contando unicamente com manipulações de listas, sem utilizar explicitamente um *Loop*, do ponto de vista prático é bastante trabalhoso. Portanto, uma linguagem de *script* foi escolhida para integrar o processo.

4.3 Python Script

Python é uma linguagem de programação *open source*, interpretada, altamente dinâmica, multi-paradigma e multi-plataforma, implementada por Guido van Rossum a partir de 1989 e desenvolvida pela fundação sem fins lucrativos chamada de *The Python Software Foundation*. Possui sintaxe simples e dinâmica, proporcionando fácil implementação, manutenção e adaptação de códigos. Plenamente adaptada as bibliotecas do *OpenNurbs*, *Rhinoceros* e *Grasshopper*, o aspecto multi-paradigma da linguagem é um diferencial do *Python* em relação ás outras linguagens *script* disponíveis. Pode-se dizer, em suma, que tanto *loops* semelhantes aos encontrados em linguagens como *C* e manipulação de listas semelhantes ás do paradigma *Lisp*, podem ser facilmente implementados em *Python* ([The python Foundation, 2014](#)).

Trabalho experimental e desenvolvimento da pesquisa

*” With four parameters I can fit an elephant,
and with five I can make him wiggle his trunk.”*

(John von Neumann)

Analizando o experimento, levando-se em conta as potencialidades e limitações do ambiente de implementação e imaginando uma estratégia de trabalho, optou-se pela subdivisão do problema em três algoritmos interconectados. A Figura 5.1 mostra como esses algoritmos relacionam-se, e como o projetista interage e manipula os parâmetros e regras para explorar as possíveis soluções do problema. Tendo por base o fluxograma apresentado na Figura 1.4, restringindo-se porem as relações entre os algoritmos e o usuário.

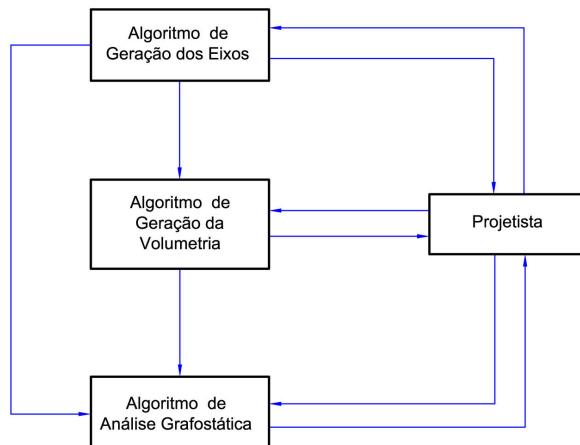


Figura 5.1: Fluxograma Geral do Experimento

Comparando com o modelo teórico, mostrado na Figura 1.1, os componentes ”Algoritmos de Geração dos Eixos”e ”Algoritmo de Geração da Volumetria”controlam as ”flexões possíveis do modelo geométrico”; o ”Algoritmo de Análise Grafostática”corresponde à ”Grafostática”, marcada na área de interseção entre a Física e a Geometria no modelo teórico; ao ”Projetista”cabe escolher a forma final de projeto uma otimização hibrida de objetivos quantitativos e qualitativos.

As formas geradas pelo ”Algoritmo de Geração dos Eixos”são usadas, junto com outros parâmetros, tanto para a geração da volumetria, quanto para a análise da estrutura. O ”Algoritmo de Geração da Volumetria”é lido junto com a forma dos eixos pelo ”Algoritmo

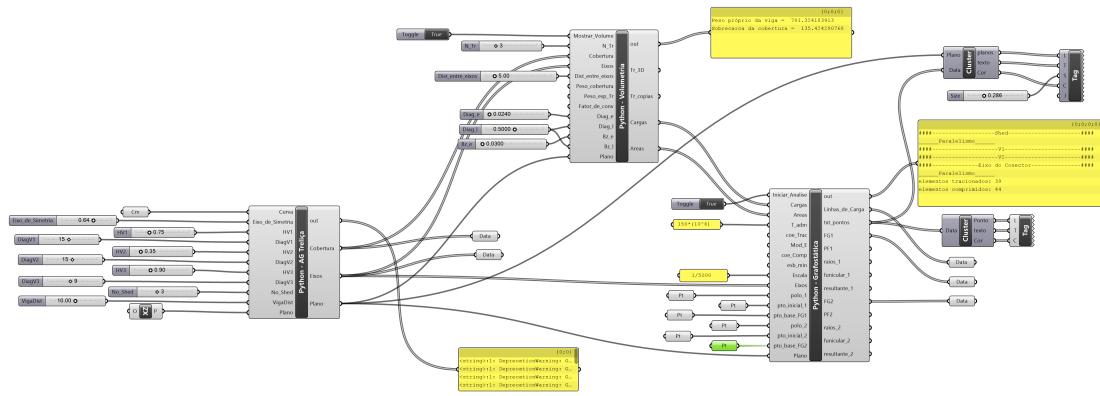


Figura 5.2: Captura de tela dos componentes onde os algoritmos foram implementados

de Análise Grafostática” e o conjunto dos resultados é avaliado pelo projetista para que decisões sejam tomadas. Nesta capítulo serão apresentados cada um dos algoritmos que compõem o experimento, bem como sua atuação em conjunto, validação e uma análise dos resultados obtidos.

5.1 Divisão da Treliça em Elementos e Nomenclatura Utilizada

As treliças desenhadas pelos Algoritmos são subdivididas em elementos para manipulação e estudo. Na figura 5.3(esq.) vemos a nomenclatura padrão dos elementos de uma treliça. Os elementos no limite superior e inferior das vigas são chamados de banzos. os elementos internos são chamados e diagonais.

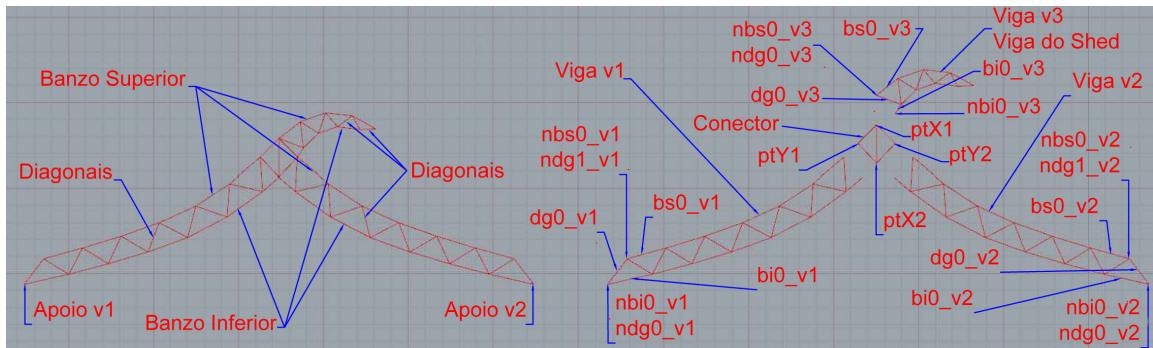


Figura 5.3: Fluxograma Geral do Experimento

A mesma solução é desenhada na Figura 5.3(dir.) com os seus elementos separados em 4 grupos para melhor entendimento, juntamente com a nomenclatura proposta. A treliça foi dividida em três vigas e um elemento de ligação entre estas, denominado de Conector. Este último, ao centro, liga as três vigas geradas e possui cinco elementos, quatro deles de borda, formando um quadrilátero, e um central sempre na orientação vertical, chamado

de eixo do conector. Os pontos do conector de mesma coordenada x são referidos como $ptX1$ (superior) e $ptX2$ (inferior), e os de mesma coordenada y foram chamados de $ptY1$ (coincidindo com o último nó do banzo superior de v1) e $ptY2$ (coincidindo com o último nó do banzo superior de v2). A viga "v3", também chamada de "viga do shed", suporta a cobertura que protege a abertura de ventilação e iluminação. As vigas "v1" e "v2" são construídas a partir de curvas simétricas em relação ao eixo do conector. A simetria, porém, só é completa se ambas apresentarem o mesmo número de diagonais e a mesma altura na porção próxima aos respectivos apoios, informações definidas entre os parâmetros de entrada do algoritmo de geração dos eixos. Nas três vigas são mostrados os elementos dos banzos e diagonais, de acordo com a ordem com que são numerados, assim como os nós.

Para os banzos superiores foi adotado a nomenclatura bs seguida do número, em sequência que parte do elemento mais próximo ao apoio, nos casos das vigas "v1" e "v2", e mais próximos no conector, no caso da viga "v3". A numeração vai de 0 (zero) até $n-1$ (sendo n o número de elementos do banzo superior de cada viga). Por fim, o nome de cada viga é acrescentado separado pelo caractere *underscore*. Para os elementos do banzo superior da viga "v1" tem-se a sequência: $bs0_v1$, $bs1_v1$... $bs(n-1)_v1$.

Para as diagonais e banzos inferiores substitui-se o prefixo "bs" por "dg" e "bi", mantendo os mesmos critérios para o restante da nomenclatura. Para o banzo inferior tem-se: $bi0_v1$, $bi1_v1$... $bi(n-1)_v1$. Pra as diagonais: $dg0_v1$, $dg1_v1$... $dg(n-1)_v1$.

Os nós são nomeados de forma semelhante aos elementos, acrescentando um "n" no prefixo: $nbs0_v1$, $nbs1_v1$... $nbs(n)_v1$. Cada nó pode ser nomeado de duas maneiras, uma em relação ao banzo a que pertence e outra em referência a diagonal. Os nós do conector também coincidem com nós das vigas.

5.2 Algoritmo de Geração dos Eixos

A modelagem de estruturas através dos eixos dos componentes é utilizada pela grande maioria dos métodos de cálculo e também pelas mais comuns estratégias de detalhamento e construção. O primeiro componente deste trabalho tem por principal função o desenho dos eixos dos componentes da treliça, mas também traça os eixos da cobertura, levando em conta a abertura do shed, e testa a determinação Estática da treliça gerada.

Na Figura 5.4(a) vemos uma solução gerada pelo algoritmo com as variáveis ajustadas como mostrado na Figura 5.4(b). As onze entradas (*inputs*) do algoritmo e as implicações destas na forma gerada serão comentadas em seguida.

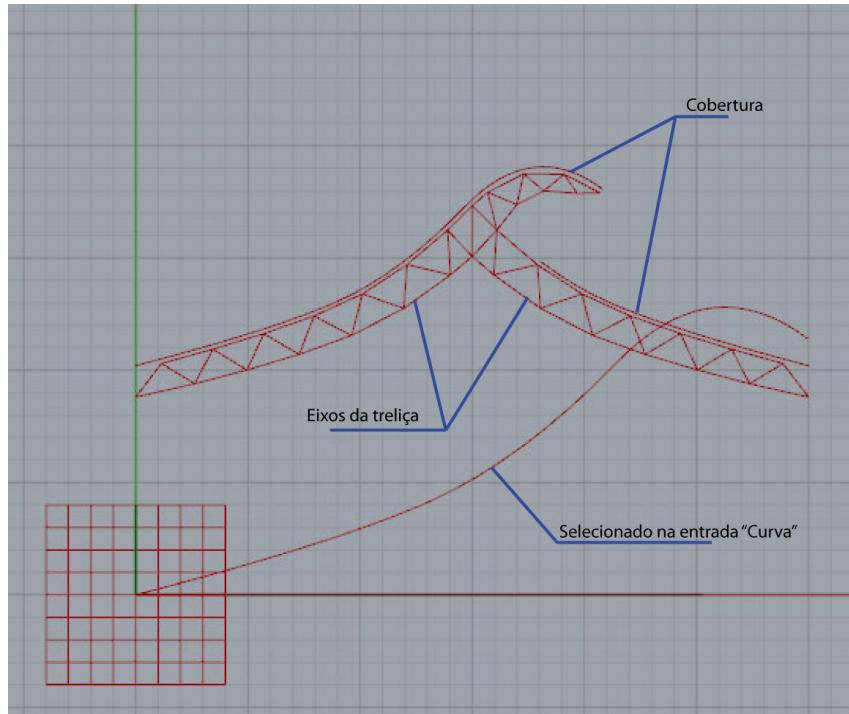
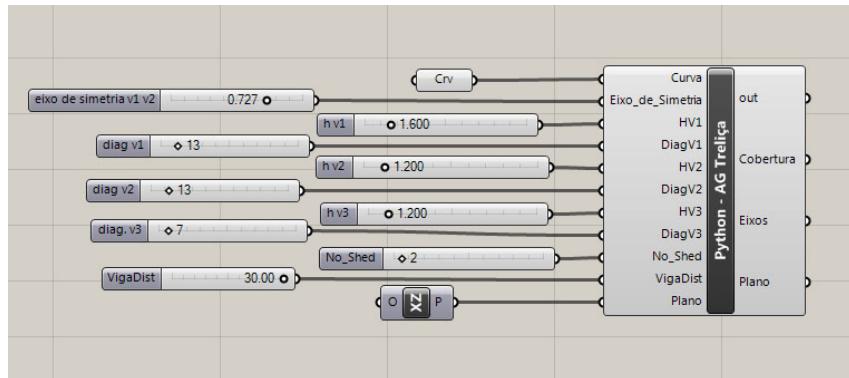
(a) Eixos Gerados pelo Componente - *Rhinoceros*(b) Componente de Geração dos Eixos - *Grasshopper*

Figura 5.4: Algoritmo de Geração dos Eixos

5.2.1 Parâmetro 1: Curva

O primeiro parâmetro (Figura 5.4(b)) de cima para baixo, denominado "Curva", é um dos parâmetros obrigatórios do algoritmo, e de crucial importância para o experimento. Na Figura 5.4(a), vê-se um objeto descrito com o texto "Selecionado na entrada Curva". Inicia-se o trabalho desenhando uma curva no ambiente do *Rhinoceros*, selecionando no componente "Crv" (Figura 5.4(b)) que alimenta a entrada "Curva" do algoritmo.

A curva define o formato dos banzos superiores da treliça. Para melhor visualização e manipulação, optou-se por mover o desenho dos eixos 10 unidades acima da curva selecionada. Isso pode ser visto na Figura 5.4(a), onde a curva inicial está nomeada como

'Selecionado na entrada "Curva"'. Na citada figura, além de ser movida em 10 unidades, a curva inicial está sendo reduzida por um fator de escala definido pelo parâmetro "VigaDist".

O parâmetro aceita tanto curvas *nurbs* (*Non Uniform Rational Basis Spline*); quanto polilinhas, formadas por seguimentos retos e policurvas, que podem ser geradas a partir da união de arcos, curvas e retas. Uma restrição importante é quanto ao sentido de grafia da curva. Em um sistema CAD as linhas e curvas são armazenadas na ordem em que os pontos são inseridos. Para o correto funcionamento dos algoritmos, o sentido da curva deve partir do ponto inicial do banzo superior da viga "v1" e seguir na direção do ponto final do banzo superior da viga "v3". No *Rhinoceros* o sentido da curva pode ser visualizado pelo comando *direction* e invertido pela opção *flip* do mesmo comando.

Esta curva é a principal ligação do método de trabalho proposto com as práticas de projeto tradicionais, e tem efeito predominante sobre o aspecto estético da treliça. A curva pode ser desenhada diretamente no *Rhinoceros*, criada a partir da junção de segmentos, ou rascunhada em papel e digitalizada, seguindo a metodologia preferida do projetista.

Um *offset* (denominação de comando ou função existente em diversas plataformas CAD, que realiza uma cópia paralela de uma reta ou curva à uma certa distância) a 0.2 unidades da curva é gerado e nomeado como *Cob1*, representando o eixo da cobertura sobre as vigas "v1" e "v3".

5.2.2 Parâmetro 2: Eixo de Simetria

O segundo parâmetro opera diretamente sobre o parâmetro anterior. Sua função é definir o ponto *ptX1*, coincidente com o ponto inicial do banzo superior da viga "v3" e que é o primeiro ponto do eixo de simetria *ptX1 – ptX2* e determina, não só o eixo do conector, mas também o eixo de simetria entre as curvas guias dos banzos superiores das vigas "v1" e "v2".

O parâmetro pode receber dois tipos de entrada : um número real entre 0 e 1, ou uma reta. No primeiro caso o ponto *ptX1* é definido pela proporção $L \times p$ onde L é o comprimento total da curva inicial e p o valor numérico entre 0 e 1. No segundo caso, para permitir a seleção de um ponto específico e preciso sobre a curva inicial, o ponto é determinado pela intersecção de uma reta com a curva inicial. Neste caso, a reta e a curva inicial precisam estar no mesmo plano.

5.2.3 Parâmetro 3: HV1

O terceiro parâmetro representa a medida do segmento $ptX1$ $ptX2$. Recebe um valor real positivo. O ponto $ptX2$ é gerado pela cópia no sentido $-y$ do plano de desenho à uma distância correspondente ao valor do parâmetro "HV1", e uma linha é traçada entre os pontos. Esta linha representa um elemento do conector, e também o eixo em que a curva inicial é dividida em duas, uma que serve de guia para o banzo superior da viga "v1", outro como guia do banzo superior de "v3". A primeira parte é espelhada em relação ao eixo do conector gerando assim a linha guia do banzo superior de "v2".

As curvas guias descritas acima fazem parte das "geometrias ocultas" usadas pelo algoritmo para a geração das formas finais, e serão denominadas como gs_v1 , gs_v3 e gs_v2 respectivamente. O termo "geometrias ocultas" aqui empregado refere-se a uma série de objetos geométricos que são gerados pelo algoritmo, e representam estágios intermediários da série de passos que, ao seu final, apresentarão as geometrias que o processo tem por objetivo. Apenas as formas geométricas endereçadas à uma das saídas do componente são visualizadas pelo usuário.

Um offset da curva gs_v2 á 0.2 unidades é gerado e denominado como $Cob2$, representando o eixo da cobertura sobre a viga "v2". Antes de ser endereçada para a saída "Cobertura", a curva $Cob2$ é editada conforme descrito na Seção 5.2.6.

Duas geometrias ocultas são geradas após a definição do ponto $ptX2$. As distâncias entre o ponto $ptX2$ e as curvas gs_v1 e gs_v2 são calculadas e um *Offset* de cada uma das curvas (gs_v1 e gs_v2) é criado de acordo com as distâncias encontradas, respectivamente referidas como $gi1_v1$ e $gi1_v2$, e seccionadas no ponto $ptX2$. As curvas gs_v1 e gs_v2 também são seccionadas, tendo o ponto sobre a corva mais próximo de $ptX2$ como ponto de corte. Esses pontos também são utilizados para o desenho das bordas do conector, o ponto sobre a curva gs_v1 é o ponto $ptX1$, e o ponto sobre gs_v2 é $ptX2$.

Ligam-se os pontos $ptX1$, $ptY1$, $ptX2$, $ptY2$ e fecha-se o quadrilátero no ponto inicial $ptX1$ com uma polilinha, gerando assim as bordas do conector.

Um *offset* da curva gs_v3 é gerado, na distância ela e o ponto $ptY2$, denominada de gi_v3 .

5.2.4 Parâmetros 5 e 7: HV2 e HV3

O parâmetro HV2 recebe um valor real maior que zero, representando a altura da viga "v1" no extremo do apoio.

O parâmetro HV3 recebe um valor real maior que zero, representando a altura da viga "v2" no extremo do apoio.

A altura no extremo do apoio é controlada através das curvas auxiliares $gi2_v1$, para a viga v1, e $gi2_v2$ em v2. Os banzos inferiores das vigas v1 e v2 são determinados com o auxilio de duas curvas guias cada. Assim como a curva guia $gi1_v1$ (cuja criação é comentada na Seção 5.2.3), a curva $gi2_v1$ é gerada por um *offset* da curva guia do banzo superior, gs_v1 , a uma distância correspondente ao valor do parâmetro HV2. O mesmo vale para as curvas $gi1_v2$ e $gi2_v2$, gerados a partir da curva gs_v2 tendo como distância o parâmetro HV3, na viga v2.

Os parâmetros HV2 e HV3 são opcionais, caso nenhum valor seja recebido por uma das entradas citadas, o valor de HV1 é usado, forçando uma viga de altura constante entre o apoio e o conector. A maneira como as linhas guias são usadas para o desenho das treliças esta descrito na Seção 5.2.9.

5.2.5 Parâmetros 4, 6 e 8: $diagV1$, $diagV2$ e $diagV3$

Os parâmetros $diagV1$, $diagV2$ e $diagV3$ definem respectivamente o número de diagonais das vigas "v1", "v2" e "v3". As entradas $diagV2$ e $diagV3$ são opcionais, caso não recebam valor $diagV2$ assume o mesmo valor de $diagV1$ e $diagV3$ é carregado com o valor nove.

5.2.6 Parâmetro 9: No_Shed

Determina em que nó da viga "v2" termina a cobertura sobre esta, e começa a abertura de iluminação e ventilação do *Shed*. Recebe um valor inteiro e maior que zero. O número recebido pelo parâmetro corresponde ao ultimo nó do banzo superio de "v2" á servir de apoio para a cobertura, contado no sentido inverso, ou seja, partindo do nó mais próximo do conector, em direção ao mais próximo do apoio, no citado banzo. A curva *Cob2* é cortada no ponto mais próximo do nó definido por *No_Shed*.

5.2.7 Parâmetro 10: $VigaDist$

O parâmetro *VigaDist* fixa o valor para o vão (distância entre apoios) da treliça. A Figura 5.5 apresenta duas soluções geradas sem que o parâmetro *VigaDist* tenha recebido um valor. Neste caso não existe distância fixa para o vão. As duas soluções foram geradas a partir da mesma curva, apresentada na parte inferior da figura, e respectivos valores de

0.60 e 0.65 para o parâmetro Eixo_de_Simetria. Todos os demais parâmetros de entrada receberam valores iguais nas duas soluções.

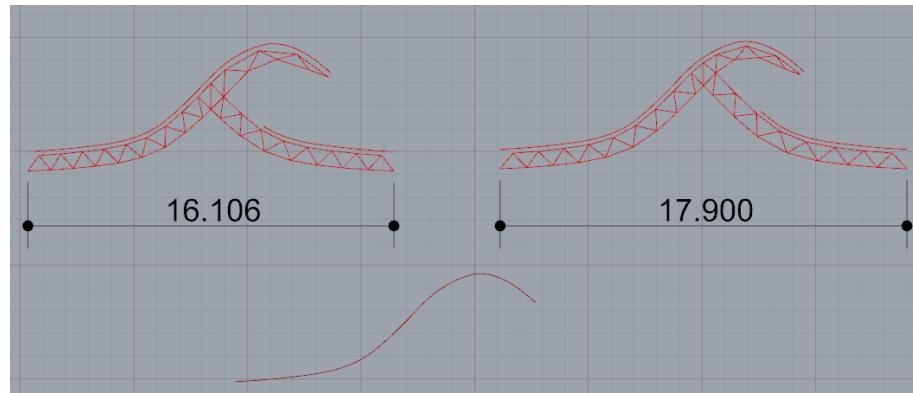


Figura 5.5: Variação do Vão em Função do Parâmetro Eixo de Simetria

Já que a curva guia *gs_v2* da viga "v2" é definida por um espelhamento da porção da curva inicial que se encontra antes do ponto *ptX1*, se o citado ponto encontrar-se mais próximo do ponto de origem da curva inicial (0.60 no exemplo apresentado) as porções simétricas serão menores em relação às geradas com um ponto *ptX1* mais distante da origem (0.65).

Se o parâmetro *VigaDist* recebe um valor (um número real positivo), a curva inicial é submetida à um fator de escala definido por $\frac{VigaDist \times 0.5}{D1}$, onde *D1* corresponde à distância horizontal entre o ponto *ptX1* e o ponto de origem da curva inicial. Desta forma o espelhamento da curva guia *gs_v1* em *gs_v2* definirá um vão com a distância igual ao valor lido na entrada *VigaDist*.

Embora treliças possam ser geradas sem a sua utilização, o parâmetro tem grande importância na aplicação do algoritmo em situações de projeto, visto que, na prática profissional, os vãos vencidos pela treliça são geralmente definidos pela malha de pilares e vigas em um passo anterior. Caso ocorram mudanças no espaçamento desses elementos, pode-se adaptar rapidamente o desenho da treliça.

5.2.8 Parâmetro 11: Plano

O parâmetro Plano, como o nome sugere, define o plano em que a curva inicial é desenhada, portanto, o desenho dos eixos das soluções será gerado. Caso não receba nenhum valor, o algoritmo considera as coordenadas *XY* do ambiente do *Rhinoceros* como plano de desenho.

5.2.9 Função 01: Geração dos Eixos das Vigas v1 e v2

Uma função foi pensada para o desenho dos eixos das vigas "v1" e "v2". As entradas da função são:

- A linha guia do banzo superior gs_v1 ou gs_v2 , aqui chamado simplesmente de gs .
- O número de diagonais, dn .
- A distância HV1.
- A distância HV2 ou HV3, dependendo de qual viga esteja sendo desenhada, identificada aqui como HV2.
- O ponto $ptX2$ é passado como ponto de referência para orientação de direções de desenho.

Duas curvas auxiliares são construídas com o intuito de servirem como guias para o desenho do banzo inferior. Um *offset* da curva gs na distância HV1, denominada $g1$, e outro, da mesma curva, com a distância HV2, referida como $g2$.

Sendo N o número de diagonais, a linha guia do banzo superior é dividida em $2 \times N$. gerando uma lista de pontos. O primeiro ponto é movido na distância HV2 e retirado da lista. A lista de pontos ainda deve ser reduzida a $0.5 \times N$. isso é feito através de uma máscara binária na forma (`False, True, False, False`). Essa lista é aplicada em *loop* até o último elemento da lista de pontos gerando a lista $ptB1$.

A estratégia de dividir o banzo em um número maior de partes e reduzir permite afastar o ponto inicial do banzo superior do apoio no banzo inferior de forma a gerar um desenho melhor distribuído das diagonais. a proporção foi encontrada através de testes e alterações no código fonte.

O ponto de apoio é o primeiro a ser incorporado para a lista de pontos do banzo inferior. Os demais pontos são gerados pelo seguinte processo:

Uma linha é traçada entre os pontos consecutivos da lista do banzo superior. O ponto médio dessas linhas é encontrado e utilizado para calcular os pontos sobre as curvas $g1$ e $g2$ mais próximos deste ponto médio. Uma linha auxiliar liga os pontos sobre as curvas no sentido $g2, g1$. Um ponto escolhido sobre a linha auxiliar na proporção $\frac{d}{l} = \frac{D}{L}$ onde l é o comprimento total da linha auxiliar, d a distância entre o ponto inicial da linha auxiliar e o ponto a ser escolhido, D é o comprimento da curva $g2$ até o ponto a partir de onde foi traçada a linha auxiliar e L o comprimento total da curva $g2$. Com a curva $g2$

e a linha auxiliar traçadas pelos métodos acima descritos, nota-se que a única incógnita desta equação é d . Quanto mais próximo do ponto inicial da curva $gi2$ se encontrar a linha auxiliar, mais próximo da origem da citada linha estará o ponto escolhido, portanto mais próximo do valor de $HV2$. A medida que os pontos se distanciam da origem de $gi2$, se aproximam da curva $gi1$, onde se encontra o ponto final da linha auxiliar, convergindo, portanto, à distância $HV1$.

Com o nó dos dois banzos definidos, o desenho da viga é uma questão de manipulação e criação de polilinhas entre eles.

5.2.10 Função 02: Geração dos Eixos da Viga v3

Uma função foi pensada para o desenho dos eixos da viga "v3", também chamada de viga do *shed*. As entradas da função são:

- A linha guia do banzo superior gs_v3 .
- O número de diagonais, dn .
- O ponto $ptX2$ é passado como ponto de referência para orientação de direções de desenho.

O processo é semelhante ao de geração das vigas "v1" e "v2". Apenas um *offset* da curva gs_v3 é criada, com a distância entre o ponto $ptY2$ e a citada curva. As linhas auxiliares são traçadas entre as duas curvas e o ponto sobre as linhas são encontrados pela fórmula $\frac{d}{l} = \frac{D}{L} \times 0.75$. Desta forma os pontos mais distantes da origem das curvas (na proximidade do conector) apresenta uma solução para o valor de d próximos à zero. Aproximando-se do final da curva, a distância entre os banzos aproximasse de $0.75 \times$ a distância do *offset*.

5.2.11 Teste de Determinação Estática

Como descrito na Seção 3.7.1, duas características e uma condição de igualdade deve ser satisfeitas para verificar a isostaticidade de treliças. As características são analisadas pela inspeção visual das formas. Depois do desenho dos eixos, é bastante simples extrair o número de nós e barras, e a igualdade pode ser checada por uma condicional (*if*) é um aviso de "determinação estática", juntamente com os valores de m e j é impresso na saída *out*. O algoritmo de geração dos eixos foi pensado de forma a gerar sempre treliças que atendam a igualdade, portanto o teste condicional serve apenas para provar

que este aspecto da determinação estática é sempre atingido. As duas características da forma, entretanto, devem ser observadas sempre que houverem alterações na geometria dos eixos. Tendo como principal elemento de definição uma curva as implicações das sinuosidades no desenho das peças deve ser levado em cota e observado a cada caso. O objetivo do algoritmo é desenhar treliças coerentes com critérios construtivos, a combinação de parâmetros que resultam em grandes distorções dos elementos da treliça deve ser devidamente analisado e, sendo o caso, descartado.

Imaginando-se uma tangente à curva inicial no ponto de interseção com o eixo de simetria, nota-se que, quando a inclinação da tangente se aproxima dos 45° , as bordas do conector se aproximam de um quadrado. Quando a referida inclinação se aproxima de 0° ou 90° , a tendência é que o algoritmo distorça o conector e a viga do *shed*, falhando em produzir uma treliça válida.

5.2.12 Saídas do Algoritmo

- **Out** : uma saída padrão dos componentes programáveis. É possível mostrar resultados e textos através dos comandos de exibição de texto (comando *print*). O teste de determinação Estática é mostrado na saída do componente.
- **Cobertura** : linhas de eixo da cobertura são mostradas através desta saída.
- **Eixos** : os eixos dos elementos da treliça.
- **Plano** : retorna o plano utilizado para a construção do desenho dos eixos.

5.3 Algoritmo de Geração da Volumetria

Devido a grande importância dos eixos da treliça, tanto para o cálculo, quanto para o detalhamento e execução, optou-se pelo componente específico para sua implementação. Um dos maiores benefícios de se usar computação gráfica para projetos é a possibilidade de visualizar em 3D as formas em estudo. Enquanto no espaço bidimensional do papel, os volumes podem ser vistos apenas sob um ângulo de visão para cada perspectiva desenhada, os ambientes computacionais, a partir da definição do conjunto de objetos tridimensionais, permitem a análise de suas formas por todos os ângulos possíveis pela simples variação do ponto de observação.

Apesar de, através dos eixos, ser possível imaginar as dimensões, proporções e volumes, o modelo tridimensional é a forma mais completa e direta de se analisar os rebatimentos

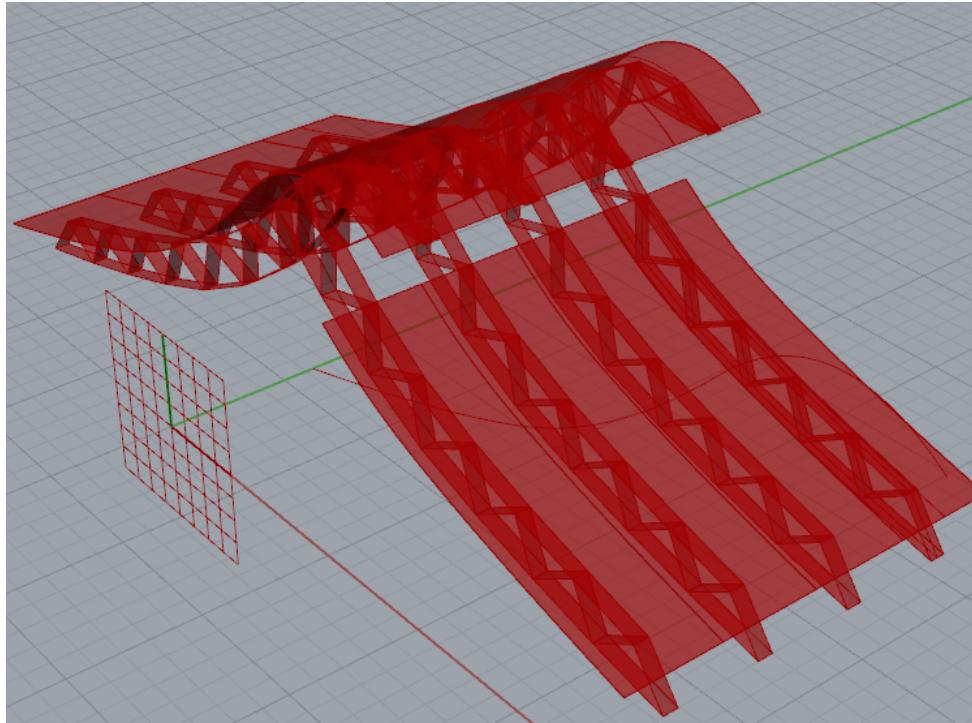
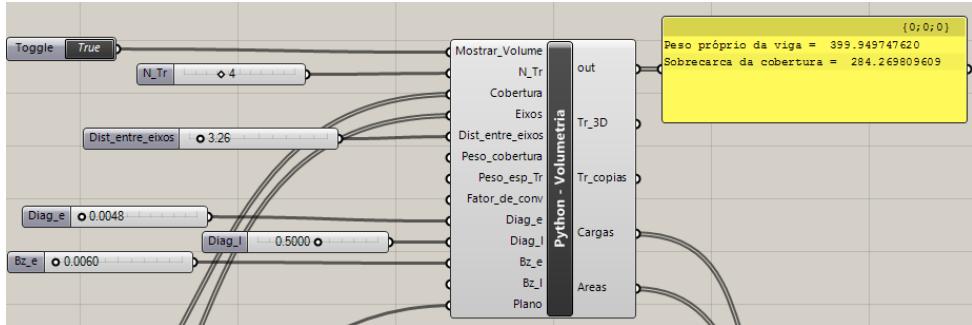
(a) Volumes Gerados pelo Componente - *Rhinoceros*(b) Componente de Geração dos Volumes - *Grasshopper*

Figura 5.6: Algoritmo de Geração dos Eixos

técnicos e estéticos, como pode ser visto na Figura 5.6(a). As dimensões e distâncias necessárias para transformar os eixos bidimensionais em um volume são as mesmas variáveis que seriam necessárias para a determinação das cargas usadas no cálculo das reações devido ao carregamento e peso próprio da estrutura. Por conta disto, optou-se por realizar os cálculos das citadas cargas também neste componente(Figura 5.6(b)).

5.3.1 Parâmetro 1: Mostrar Volume

O primeiro parâmetro recebe um valor booleano: verdadeiro (*True*) ou falso (*False*). Funciona como um controle de exibição e, mais especificamente, como um controle de fluxo do algoritmo. Quando o valor inserido é *True*, ou quando nenhum valor é recebido

pela entrada do componente, a geometria 3D é calculada e exibida. Quando o valor é *False*, não só os volumes não são mostrados, mas todo o código utilizado para sua geração é ignorado. A bateria ligada à entrada "Mostrar_volume" é denominada *Boolean Toggle* e permite a rápida alteração entre os valores *True* e *False* por um duplo clique. Esse controle permite tanto a alternância entre visualizações do volume e dos eixos, quanto a economia de recursos computacionais. Ainda que o tempo de processamento não seja elevado, a economia de alguns segundos na execução dos componentes pode proporcionar um conforto maior para o trabalho em um ambiente *CAD*.

As cargas e áreas dos elementos, contudo, são sempre calculadas, independentemente do valor lido pela entrada "Mostrar_volume".

5.3.2 Parâmetro 2: *N_Tr*

O parâmetro *N_Tr* recebe um valor inteiro e maior que zero. Controla quantas treliças serão geradas. É um recurso de visualização, permitindo analisar um trecho da cobertura onde o número de treliças pretendido é gerado.

5.3.3 Parâmetro 3: *Cobertura*

Recebe os eixos das coberturas gerados pelo componente do algoritmo de geração dos eixos.

5.3.4 Parâmetro 4: *Eixos*

Recebe os eixos dos elementos da treliça (vigas "v1", "v2", "v3" e conector) gerados pelo componente do algoritmo de geração dos eixos.

5.3.5 Parâmetro 5: *Dist_entre_eixos*

A distância entre eixos de duas treliças adjacentes da cobertura. Determina também, juntamente com as curvas da cobertura, a área de influência da cobertura sobre a treliça.

5.3.6 Parâmetro 6: Peso_cobertura

O peso estimado da cobertura por unidade de área. Caso nenhum valor seja recebido, adota a estimativa de 20 Kg/m^2 .

5.3.7 Parâmetro 7: Peso_esp_Tr

O peso específico do material da treliça. Caso não receba valor, é adotado o peso específico do aço: 7800 Kg/m^3 .

5.3.8 Parâmetro 8: Fator_de_Conv

Fator que converte as unidades de massa em unidades de força. Como o quilograma é usado como padrão, caso o parâmetro não receba valores, é utilizado o valor de 10, que converte as unidades para Newtons (o valor de 10 m/s^2 é a aproximação da aceleração da gravidade amplamente adotada no cálculo de estruturas).

5.3.9 Parâmetros 9 e 10: Diag_e e Diag_l

Respectivamente a espessura e largura dos elementos das diagonais

5.3.10 Parâmetros 11 e 12: Bz_e e Bz_l

Respectivamente a espessura e largura dos elementos dos banzos. Caso o parâmetro Bz_l não receba valor, é adotado o mesmo valor de Diag_l.

5.3.11 Parâmetro 13: Plano

Plano do desenho dos eixos da Treliça. Igual ao parâmetro de mesmo nome do componente de geração dos eixos.

5.3.12 Função 01: Geração dos volumes

Uma função foi pensada para o desenho dos volumes dos elementos e da cobertura. As entradas da função são:

- Os eixos dos elementos.
- A espessura dos elementos, e .
- A largura dos elementos, l .

A função cria uma cópia do eixo a uma distância $l/2$ na direção -Z, em relação as coordenadas XYZ do plano de desenho. A partir da cópia do eixo, são gerados dois *offsets* no plano de desenho com distância $e/2$ nas duas direções. Uma superfície é gerada tendo como bordas os dois *offsets*. A superfície é transformada em um sólido sendo extrudado com uma dimensão l na direção Z do sistema de coordenadas XYZ do plano de desenho.

Os elementos dos banzos e das diagonais são separados e recebem os respectivos valores de espessura e largura. O volume do conector é gerado pelos mesmos valores atribuídos aos banzos. A cobertura recebe a distância entre eixos das vigas como largura e um valor fixo de 0.01 m (um centímetro) como espessura. Assim todos os volumes são gerados de forma esquemática, porém suficiente para a visualização e interpretação dos objetos a serem construídos.

5.3.13 Funções 02 e 03: Cargas

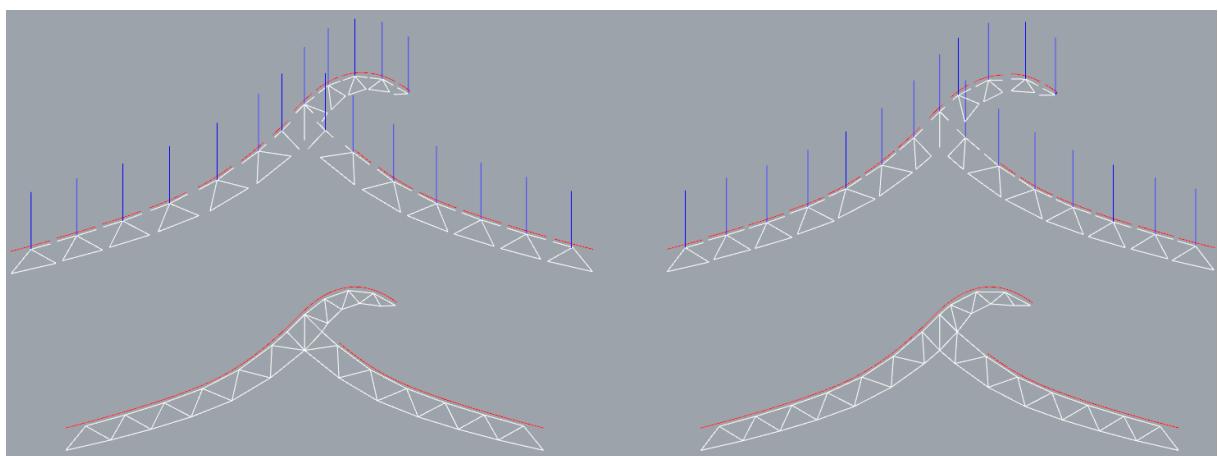


Figura 5.7: Ilustração do Cálculo das Cargas

Duas funções são responsáveis pela definição das cargas nos nós superiores da treliça. Uma calcula as cargas nas vigas "v1" e "v2", e outro para a viga "v3". A Figura 5.7 mostra os critérios usados para determinar as cargas provenientes do peso próprio e do carregamento da cobertura. As linhas azuis representam a direção das cargas (sempre com o sentido da gravidade) as brancas a treliça e as curvas vermelhas a cobertura. O mecanismo básico das funções é separar os elementos de acordo com a proximidade à cada nó. Os banzos superiores são divididos ao meio e esses pontos servem de referência para a divisão da cobertura em trechos que tem maior influência sobre o nó analisado. O comprimento de cada elemento é multiplicado pela área de sua respectiva seção, e este resultado é multiplicado pelo peso específico do material da treliça. As curvas dos eixos das coberturas tem seu comprimento multiplicado pelo parâmetro "Dist_entre_eixos" e pelo peso por m^2 da cobertura.

Os estremos de cada treliça são analisados de acordo com suas particularidades. Nas extremidades próximas ao conector o número (par ou ímpar) de diagonais em cada treliça é levado em conta, como ilustrado na Figura 5.7.

5.3.14 Funções 04 e 05: Áreas e Momentos de Inércia Mínimos

Para a espessura e largura de cada uma das peças, uma função calcula as áreas das seções planas das peças. Esses valores são passados para a saída **Areas** em forma de uma lista, na sequência [Banzo, Diagonal].

Uma função calcula os momentos de inércia das seções planas dos banzos e diagonais. Para cada elemento é aplicada a fórmula do cálculo para seções quadradas (Equação 5.1). Alternando os valores de espessura e largura como base b e altura h para cada peça.

$$I = \frac{b \times h^3}{12} \quad (5.1)$$

Os resultados são comparados e a função retorna o menor valor encontrado para os banzos e o menor para as diagonais. Uma lista na mesma sequência da de áreas é gerada e direcionada também para a saída **Areas**, que acaba por fornecer para o componente de análise, não só a medida quadrada da superfície, como uma importante propriedade da seção plana dos elementos.

5.3.15 Saídas do Algoritmo

- **Out** : Mostra o peso próprio total da treliça e a carga total da cobertura.
- **Tr_3D** : O modelo tridimensional da treliça e do trecho da cobertura que representa a área de influência sobre a treliça.
- **Tr_copias** As cópias da treliça e da área de influência da cobertura.
- **Cargas** : Lista com as cargas em cada nó, ordenadas por viga e na numeração crescente dos nós do banho superior
- **Areas** : Uma lista com a área dos elementos dos banhos e diagonais da treliça e os respectivos momentos de inércia mínimos para cada seção.

5.4 Aplicação e Validação Dos Algoritmos de Geração dos Eixos e dos Volumes

Como estabelecido anteriormente, o conceito de Algoritmo Generativo aplicado à Arquitetura está ligado tanto a capacidade do algoritmo de gerar formas a serem construídas, quanto à ideia de se estabelecer um método de trabalho propício à aplicação em projetos. Nos objetivos da pesquisa foi posto que o experimento deveria resultar em uma ferramenta de trabalho e uma metodologia de atuação. A maneira mais eficaz de se explicar um método de trabalho é mostrando-o. A variedade de treliças válidas é vasta, isso porém, não significa que qualquer curva ou configuração de parâmetros resultará necessariamente em uma treliça válida. A experiência do projetista deve guiar o processo para que as formas geradas resultem em proposta para construção.

As formas geradas pelos algoritmos foram inspiradas na treliça de cobertura do projeto do Mercado São Miguel. A primeira experiência realizada foi a de ajustar os parâmetros para aproximar as formas do algoritmo o máximo possível do desenho que o inspirou. A Figura 5.8 mostra as linhas usadas como entrada (dir. abaixo), a treliça gerada pelo algoritmo (dir. acima), um detalhe do *shed* do projeto do mercado (esq. acima), e as configurações dos parâmetros do algoritmo (esq. abaixo). Algumas simplificações foram adotadas no desenho, principalmente na viga "v3", mas também nos extremos próximos aos apoios banhos superiores de "v1" e "v2". Um eixo foi adicionado ao conector dividindo-o em dois triângulos. A justificativa para essas mudanças estão na intenção de se gerar treliças isostáticas, passíveis de análise estrutural pelos métodos escolhidos.

Outro aspecto da validação é a capacidade de gerar soluções alternativas para o projeto. Os passos necessários para se desenhar e explorar as possíveis soluções foram registrados

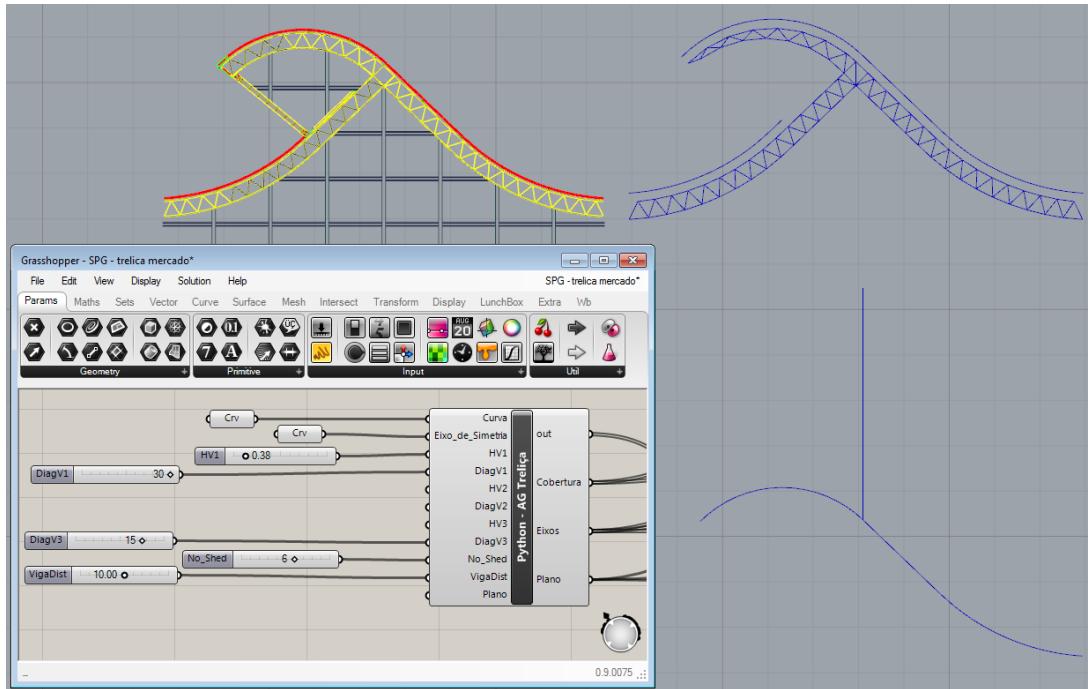


Figura 5.8: Comparação entre a treliça do Mercado São Miguel e a gerada pelo algoritmo

em vídeos. O primeiro, [02 - Modo de trabalho 01](#), mostra aspectos básicos do desenho assistido pelo algoritmo gerativo. O segundo desta série, [03 - Modo de trabalho 02](#), mostra o funcionamento dos parâmetros obrigatórios, o vídeo seguinte, [04 - Modo de trabalho 03](#), apresenta os opcionais e, em sequência, o vídeo [05 - Modo de trabalho 04](#) enfatiza as atividades de desenho e redesenho da curva inicial como ferramenta de criação de formas.

Construções que levam à falha do algoritmo foram apresentadas na Seção [5.2.11](#). Estas representam limitações do código e não necessariamente falhas do SGP. Se uma quantidade considerável de alternativas pode ser rapidamente explorados no processo, pode-se afirmar que o sistema cumpriu seu papel.

A Figura [5.9](#) mostra rebatimentos de flexões dos parâmetros na geração de formas. Os exemplos apresentados são apenas um pequeno conjunto de variações possíveis, baseada em um conceito de cobertura que foi traduzido em algoritmo de geração de formas associado à uma estratégia de trabalho. Juntando os exemplos das imagens e as estratégias e recursos de desenho apresentado nos vídeos, é possível vislumbrar a criação de incontáveis variações de um mesmo tema. A Figura [5.10](#) apresenta quatro direcionamentos diferentes para o desenho das formas, encontradas a partir da exploração de possíveis entradas no parâmetro da curva inicial.

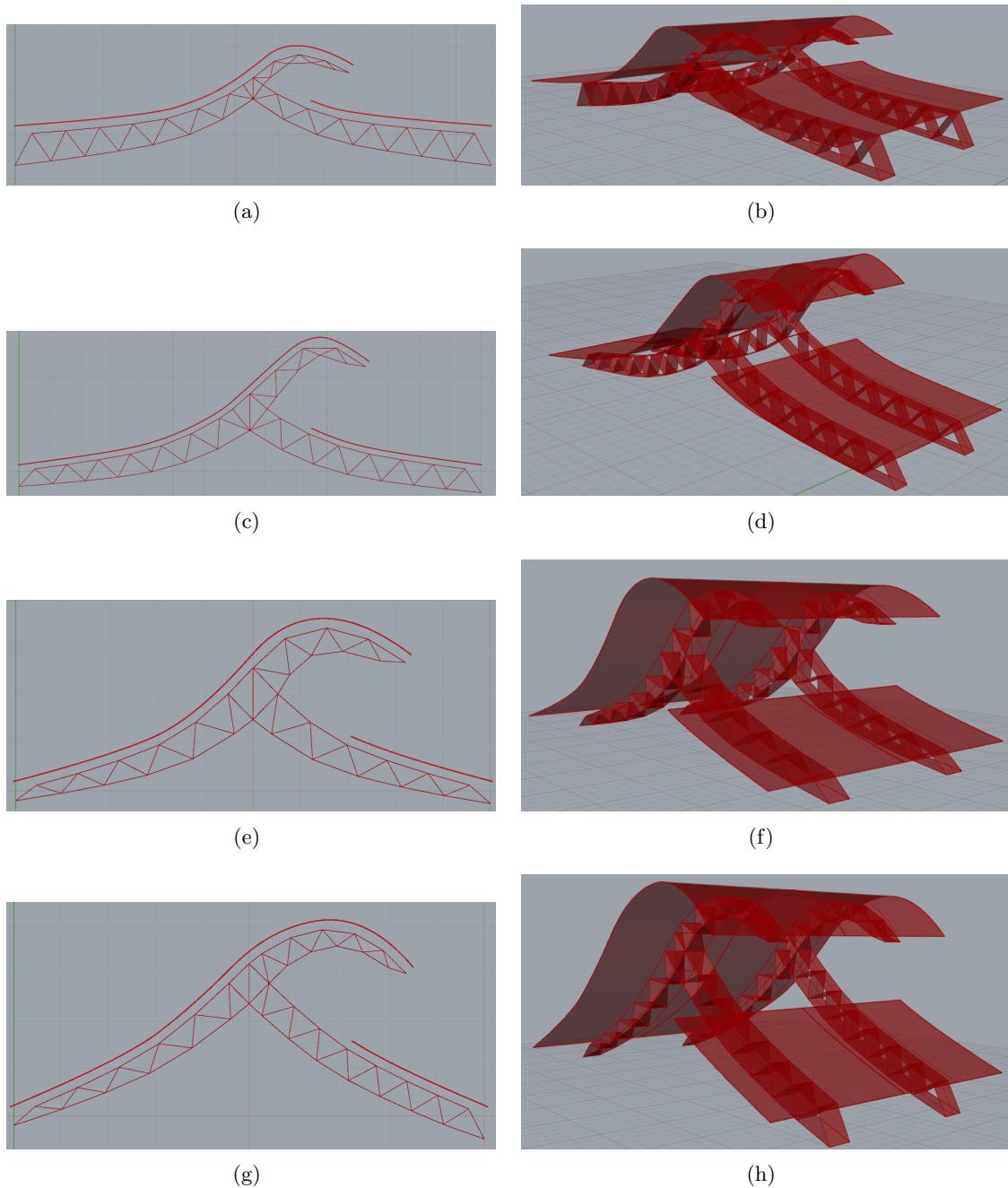


Figura 5.9: Variações na Geração de Treliças

5.5 Algoritmo de Análise Grafostática

Em linhas gerais, o algoritmo de análise grafostática recebe informações sobre a geometria, as cargas, características dos materiais e coeficiente de segurança adotado no projeto, e calcula as tensões nos elementos das barras. Algumas simplificações foram adotadas na execução do cálculo. A principal é considerar a treliça o que é definido por [Johnson \(1908, p. 83\)](#) como treliça ideal. Consistindo em uma estrutura onde: o atrito nas articulações pode ser desconsiderado, que o peso das barras seja modelado considerando que este atue

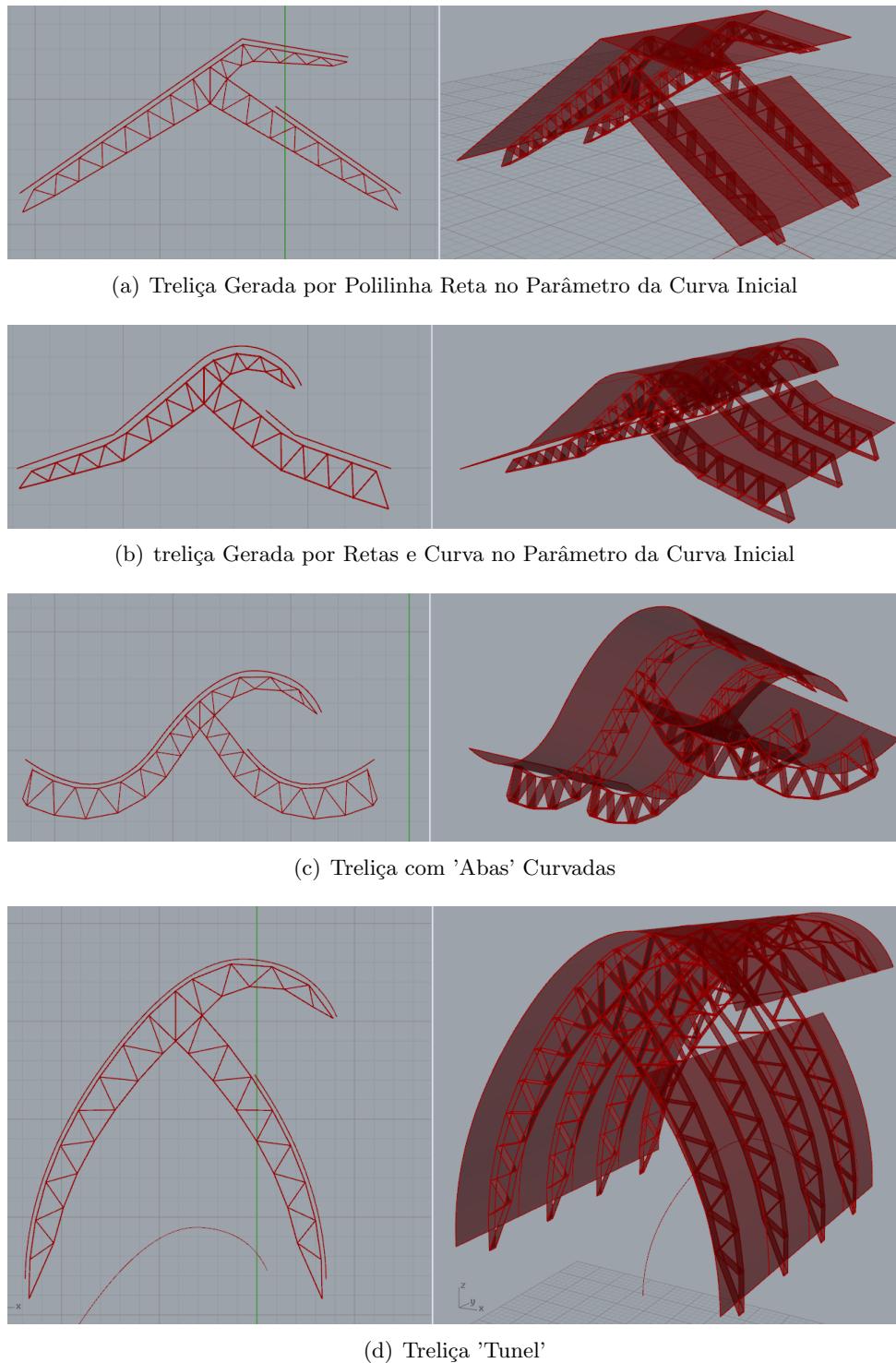


Figura 5.10: Outros Exemplos de Variações na Geração de Treliças

apenas nas suas extremidades e que outras cargas sejam aplicadas exclusivamente nas articulações. Partindo dessas assunções, o autor conclui que apenas compressão e tração pura atuam nos elementos.

A relevância dos deslocamentos na posição dos nós e provocados pelas deformações foi,

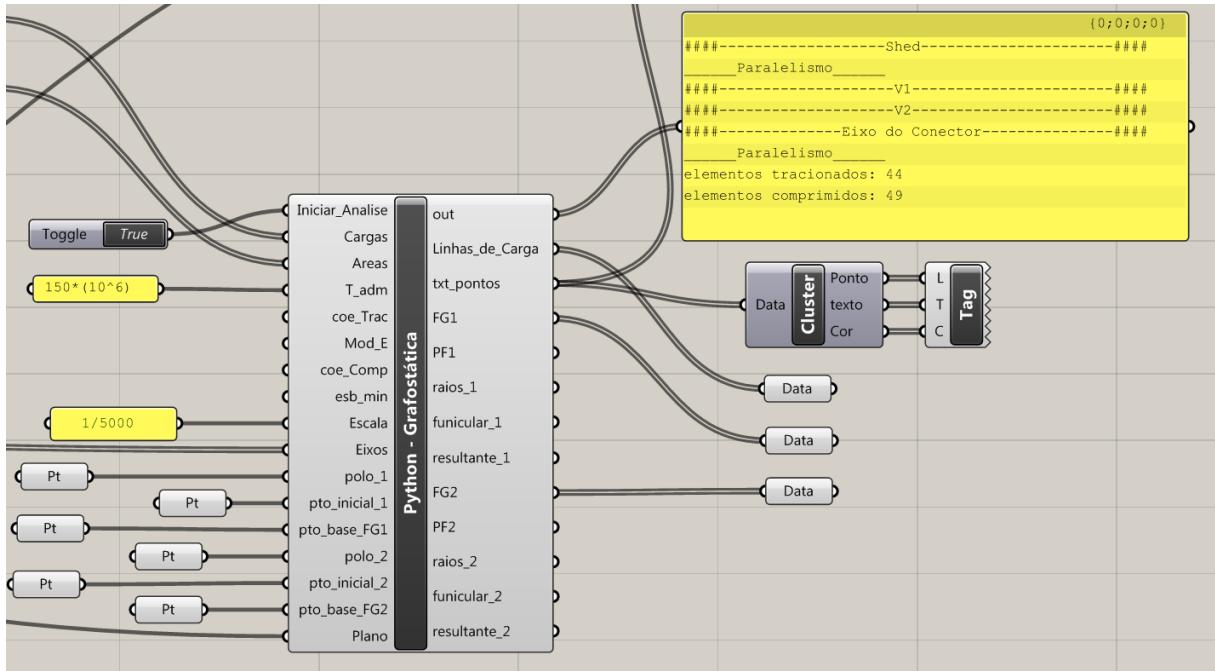


Figura 5.11: Componente de Análise Grafostática

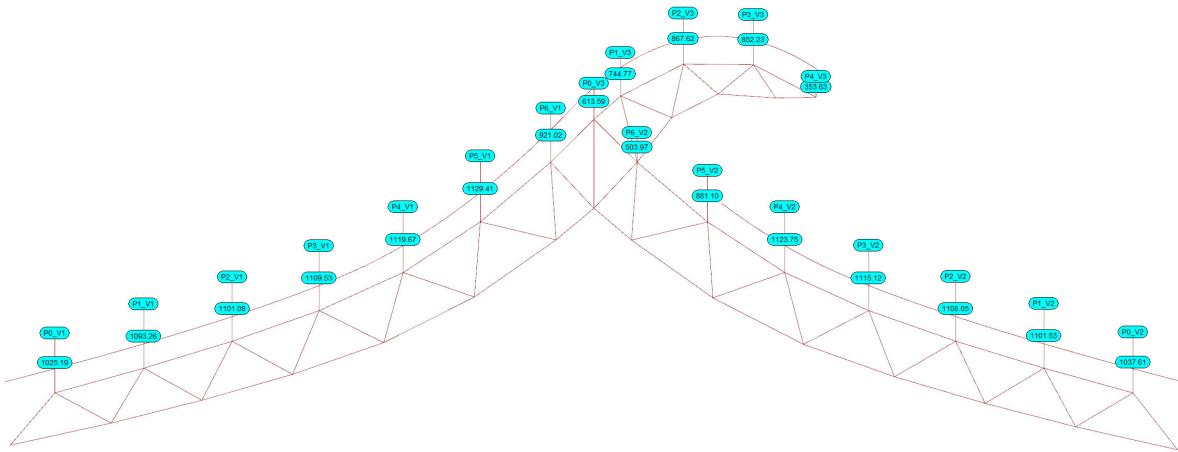


Figura 5.12: Carregamento da Treliça

ao longo da história do cálculo, desconsiderada na investigação de treliças cuja geometria proporciona maior rigidez do conjunto. Uma lista ilustrada de algumas dessas geometrias pode ser vista em [Malcolm \(1909, p. 67\)](#). A treliça em questão não pertence a nenhum dos tipos definidos pela literatura. Sendo gerada a partir de uma curva, simplificada em uma série de seguimentos de reta, a forma da curva inicial é responsável, tanto pela leitura estética particular presente em cada uma das possíveis soluções, quanto pela maior ou menor suscetibilidade à deformações do conjunto inteiro. Além das deformações, os esforços proveniente da ação dos ventos, as cargas móveis (e.g. funcionários de manu-

tenção transitando pela cobertura), a análise dinâmica das vibrações que podem afetar a estrutura, as forças sísmicas, entre outras, são desconsideradas.

A aplicabilidade do algoritmo está direcionada à fase de projeto ligada à definição das formas e pré dimensionamento dos elementos. Na sequencia tradicional da elaboração de projetos, a definição das necessidades da edificação é sucedida por esboços genéricos da relação entre os espaços e o todo. Esses conceitos traduzidos em desenhos são sequencialmente desenvolvidos e, a medida que as formas vão se transformando em detalhes, estes últimos são confrontados com o todo e entre si, com o objetivo de minimizar interferências e resolver conflitos entre elementos de projeto. O fluxo de trabalho raramente é linear. Na busca pela otimização (quantitativa e qualitativa) do projeto não é raro que desenhos anteriormente definidos sejam alterados em prol da harmonia entre as partes.

A Figura 5.11 mostra o carregamento da treliça, calculado pelo componente responsável também pela volumetria das treliças e grafado por linhas em escala definida pelo algoritmo de análise. Dois critérios de simplificação foram usados na definição das cargas.

O primeiro, relativo ao peso próprio da estrutura, assume que as cargas podem ser consideradas atuantes apenas nos nós do banzo superior. essa simplificação é largamente reconhecida e aplicada em modelos de cálculo estrutural. Suficiente para dimensionamento e, por conseguinte, para o almejado pré dimensionamento. Quando calculado por sistemas computacionais de cálculo, contudo, pela natureza *black box* da maioria das ferramentas adotadas pelo mercado, é impreciso afirmar que critério semelhante tenha sido adotado ou não.

O segundo critério de simplificação diz respeito ao pontos de aplicação das cargas provenientes da cobertura. A modelagem de cargas mais próxima do real deveria levar em conta a localização das peças da cobertura que transmitem os carregamentos para a treliça, denominadas terças. Uma gama de possibilidades construtivas pode solucionar a questão, mas não necessariamente teremos uma terça à cada nó. Levando em conta as treliças que serviram de inspiração para este trabalho, a cobertura seria formada por telhas metálicas em folhas curvadas por um equipamento industrial conhecido como calandra. Os critérios de locação das terças levam e conta a posição dos nós do banzo superior e o vão útil máximo das folhas da cobertura. Para atender as necessidades de um dimensionamento preliminar entretanto, a definição dos pesos da cobertura divididos por todos os nós do banzo superior é aceitável e suficiente.

Para a realização da análise, a treliça foi dividida em duas partes: a primeira, mostrada na Figura 5.13, vemos que o *shed* foi isolado das demais partes e, nos pontos onde descarrega sobre o conector, duas reações (*RA* e *RB*) são calculadas. Considerando que apenas tração ou compressão pura atua nas barras, o desenho das treliças com seus carregamentos

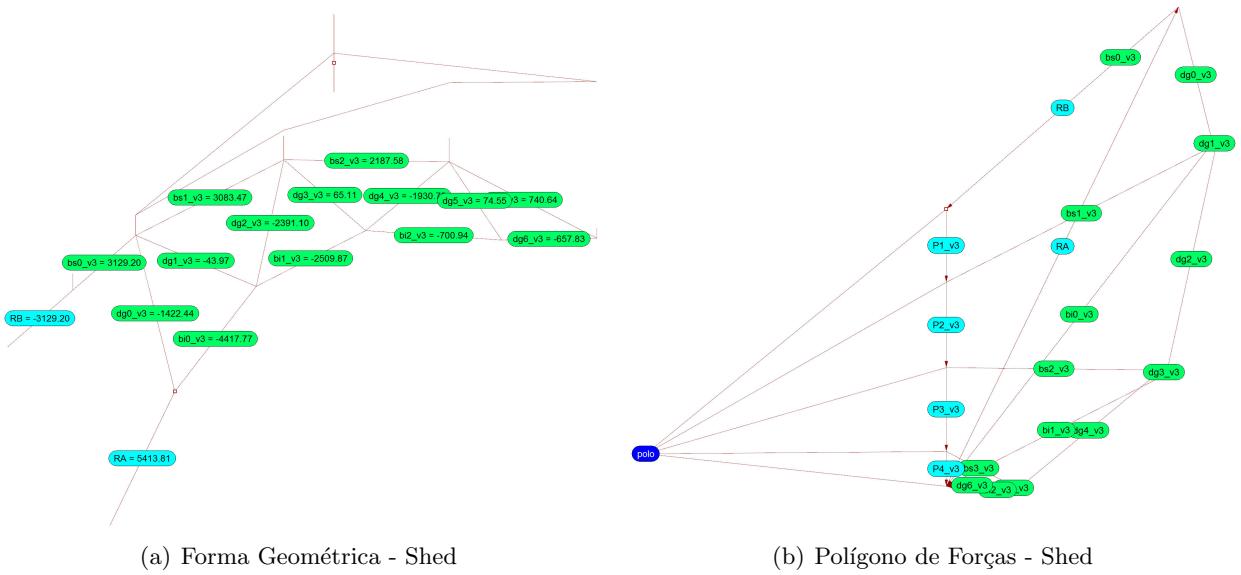


Figura 5.13: Diagramas Recíprocos do Shed (viga v3)

e reações representa completamente o diagrama da forma geométrica (Figura 5.13(a)) e o reciproco polígono de forças aparece representado na Firura 5.13(b).

Enquanto no PF apenas a nomenclatura dos elementos, carregamentos e reações é anotada, na FG a nomenclatura é acompanhada pelo valor da tração (sinal positivo) ou compressão (sinal negativo) atuante no elemento. A viga do *shed* é abordada como uma estrutura em balanço, e sua solução segue os algoritmos de desenho descritos na Seção 3.8 para resolução de treliças em balanços com dois pontos de apoio.

5.5.1 Definição do Sentido de Seleção das Linhas de Ação

A definição de um sentido (conforme descrito na Seção 3.7.2), de natureza intuitiva para a percepção humana, necessita de uma estratégia para o apropriado funcionamento no algoritmo. A Figura 5.14 ilustra como os sentidos de desenho são escolhidos pelo algoritmo. O critério utilizado baseia-se na direção da curva inicial (em azul). Um círculo (em vermelho) é desenhado através de três pontos na seguinte ordem: o apoio da viga "v1", o ponto $ptX1$ e o apoio na viga "v2". Recordando que a curva inicial tem obrigacionalmente direção partindo do ponto de apoio da viga "v1", seguindo para o ponto vinal da viga "v3", caso essa curva seja desenhada da esquerda para a direita, a seleção dos três pontos gera um círculo no sentido horário, no caso inverso, no sentido anti-horário como pode ser visto na Figura 5.14 onde as setas (em branco) indicam o sentido de desenho dos círculos e das curvas iniciais. As forças externas são ordenadas sempre segundo o mesmo critério: para a viga "v3" seguem a ordem crescente de sua numeração. para as vigas

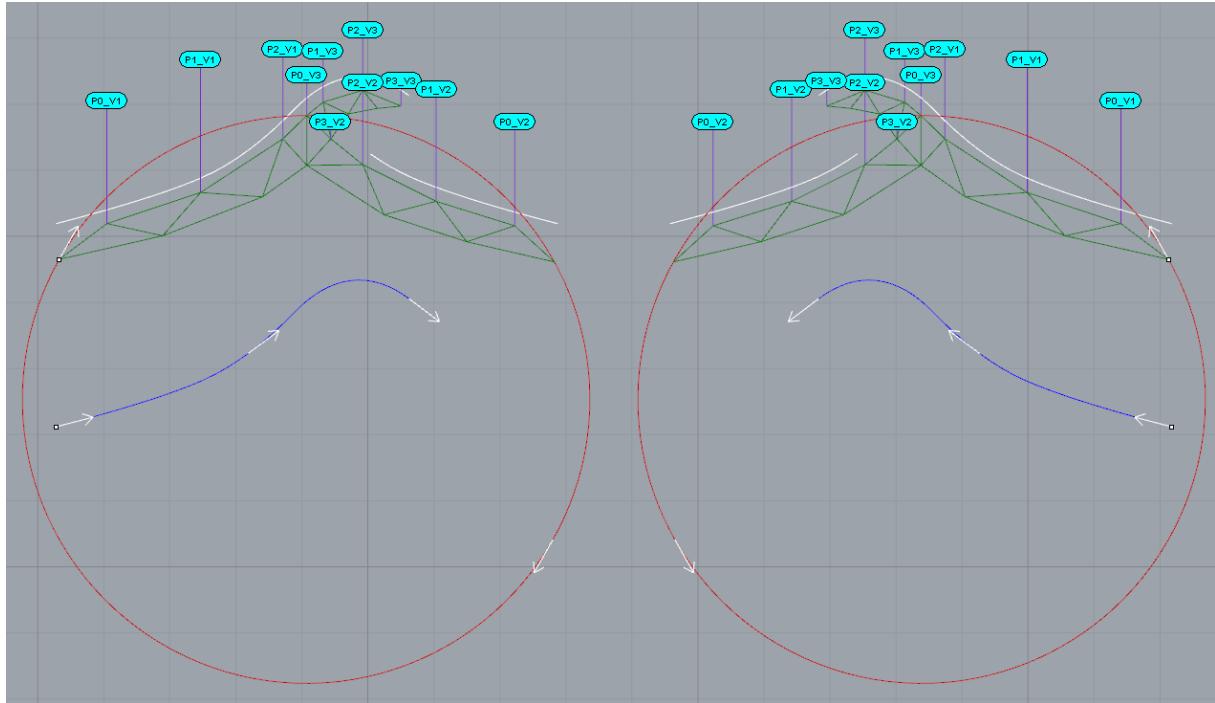


Figura 5.14: Sentido de Seleção das Forças

"v1" e "v2" seguem ordem crescente para "v1", seguindo pela carga $P0_v3$ somada com a reação RB da treliça *dashed*, e ordem decrescente para "v2" (a carga no último nó de "v2" é somada com a reação RA da viga do *shed*). Os círculos fazem parte das geometrias invisíveis e não são grafados no ambiente do *Rhinoceros* pelo algoritmo, mas servem como parâmetro para as funções que ordenam as forças em cada nó.

5.5.2 Método Grafostático Aplicado à Viga do *Shed*

A viga do *shed* é resolvida pela implementação do algoritmo descrito na Seção 3.8. Os vídeos 06 - *Shed impar pt1*, 07 - *Shed impar pt2*, 08 - *Shed par pt1* e 09 - *Shed par pt2* em anexo são ao mesmo um recurso didático para auxiliar a explicação dos métodos utilizados, como importante instrumento de validação dos resultados. A ideia consiste na consciência da Grafostática se tratar de um campo da ciência devidamente comprovado e descrito desde meados do século XIX. Estes métodos são formulações gráficas dos princípios da mecânica clássica que, no escopo e aplicações requisitadas pela engenharia das estruturas, permanece incontestavelmente coerente e eficiente. As operações gráficas modeladas em algoritmo foram testadas manualmente e sequencialmente nos vídeos apresentados e a ordem de suas operações conferidas nos diversos materiais bibliográficos citados. Ao final dos vídeos, os resultados do desenho manual é validado e se mostra coerente com a pesquisa bibliográfica e os diagramas gerados pelo algoritmo são sobrepostos

perfeitamente, mostrando que a sequência de operações realizadas pelo algoritmo está de acordo com as regras estabelecidas pela Estática Gráfica.

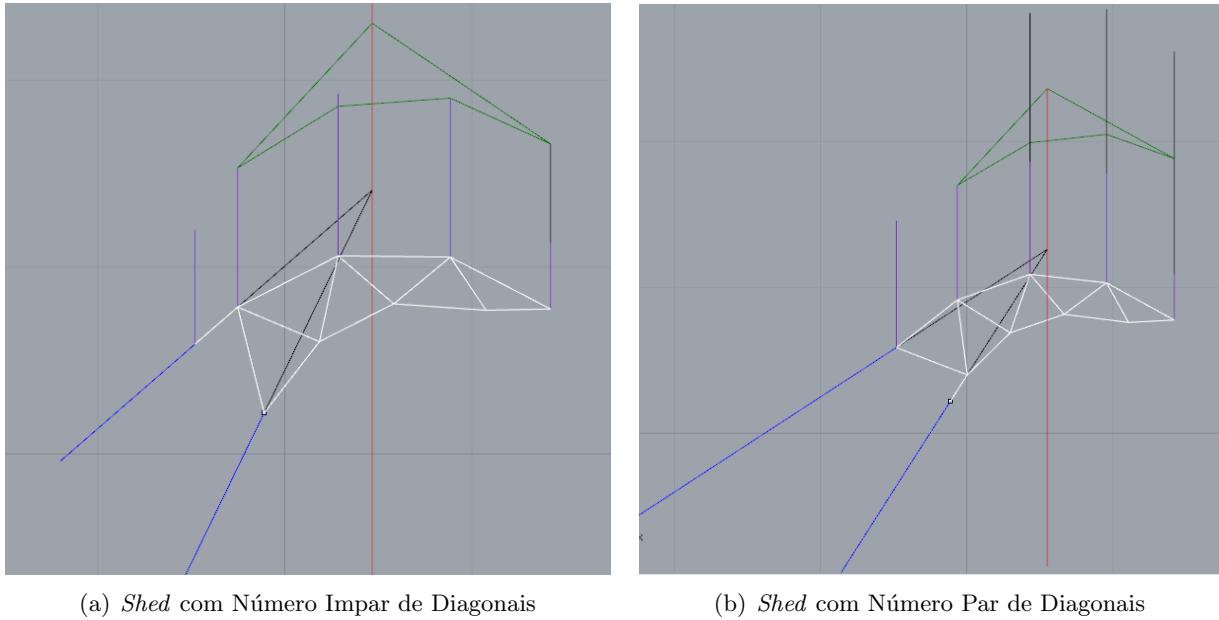


Figura 5.15: Resolução de Reações em *Sheds* com Números Pares e Impares de Diagonais

Os procedimentos para determinação das reações para diferentes números de diagonais respondem às mesmas premissas, porém foram testados e apresentados em sequências distintas, afim de validar os dois caminhos previstos no algoritmo. A Figura 5.15 mostra a FG de uma treliça do *shed* com número ímpar (Figura 5.15(a)) e com número par de diagonais (Figura 5.15(b)). Os primeiros passos, nos dois casos, consistem em transpor os vetores das cargas para o PF, com exceção da força $P0_v3$, cuja ação é considerada diretamente sobre o conector. Na Figura 5.13(b) pode-se ver as forças $P1_v3$ à $P4_v3$ da treliça com sete diagonais mostrada na Figura 5.13(a). Com as forças em posição de soma, um polo é definido e os raios polares são traçados. Com as direções dos raios polares traçase o funicular na FG. Fechando o funicular, um ponto da resultante dos carregamentos é definido, a direção da força é a mesma da soma grafada no PF. na Figura 5.15 as linhas vermelhas representam as respectivas resultantes na FG. Assim como no método analítico, as resultantes são calculadas pela adoção de um momento nulo na estrutura, portanto a resultante e as duas reações (em azul) devem se anular. No caso de treliças ímpares temos a reação RB definida em direção, no outro caso, a reação RA é definida. Estende-se então a reação definida em direção até o seu ponto de interseção com a linha de ação da resultante, para a definição da direção da reação restante, liga-se o ponto de aplicação desta até o ponto de interseção encontrado.

Com as direções definidas, deve-se então utilizar o sentido de seleção para determinar a posição das forças no PF. Pelo sentido horário, partindo da força conhecida (resultante) obtemos a ordem, RA e RB . Na Figura 5.13(b) tem-se, portanto a reação RA posicionada

no extremo final da soma das cargas e RB no ponto inicial. Estendendo cada uma das direções até o ponto de interseção delas, defini-se a magnitude das reações na escala adotada.

Os sentidos são atribuídos automaticamente pelo algoritmo. A reação RA combate a compressão e reage no sentido ascendente, e RB descendente. Em ambos os casos uma das barras da treliça é determinada imediatamente como igual e contrária a uma das reações. No caso de número ímpar de diagonais, a reação RB se anula com o primeiro elemento do banzo superior e no caso par, RA e o primeiro elemento do banzo inferior são iguais e opostas.

A sequencia de resolução das demais barras é definida pela literatura pelos nós das diagonais pela sua numeração crescente. Para cada nó o sentido das forças é definido e os elementos já conhecidos são seguidos em ordem pelo algoritmo encontrando os pontos inicial e final do polígono aberto. A primeira das forças desconhecidas parte do ponto final até o ponto de interseção com a direção da segunda força desconhecida, com direção que começa no mesmo ponto de interseção até o ponto inicial do polígono aberto. A definição da natureza dos esforços (tração ou compressão) é definida pelos critérios apresentados na Seção 3.7.4. As forças desenhadas no PF são transportadas para a FG utilizando-se uma linha com ponto inicial no extremo final da linha do PF e ponto final no nó em análise. A distância entre o ponto inicial do vetor movido e o ponto médio do eixo recíproco na FG é armazenado em uma variável denominada $D1$. O valor de $D1$ é comparado com o valor de $C1$ obtido por $C1 = L2/2 + L1$, onde $L1$ é o comprimento do vetor no PF e $L2$ o comprimento do eixo recíproco na FG. Caso $C1$ seja menor que $D1$, existe sobreposição das linhas e a natureza do esforço é de compressão, caso sejam iguais, a solicitação é de tração. Optando usar o ponto médio do eixo na FG é a metade no comprimento de $L1$ não é necessário saber se o eixo do elemento está escrito a partir do nó ou em direção ao nó.

O processo segue inalteradamente até o penúltimo nó da diagonal. Neste, apenas uma força permanece desconhecida, bastando unicamente ligar ordenadamente o último ponto do polígono aberto ao primeiro fechando completamente o PF. Neste ponto da execução o método gráfico apresenta dois mecanismos visuais de conferência do modelo construído: a primeira é a aferição do paralelismo entre a ultima força desenhada no PF pela ligação de dois pontos com sua recíproca na FG. Foi implementado no próprio algoritmo um teste de paralelismo para esses casos e os resultados foram positivos em todos as configurações testadas. A segunda diz respeito as forças que atuam no último nó. O fechamento do penúltimo nó implicara necessariamente na construção de um polígono fechado no PF cujos lados correspondem as forças atuantes no último nó, fato que pode ser observado em todas as simulações.

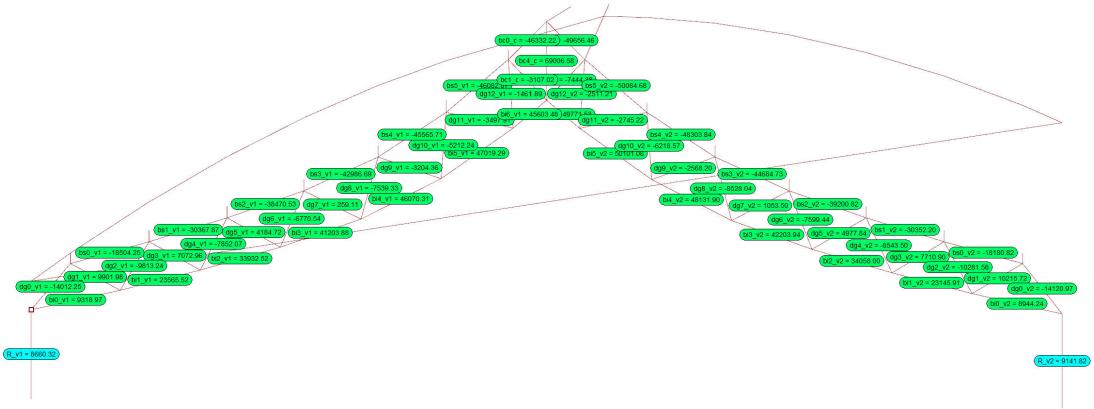
5.6 Método Grafostático Aplicado às Vigas v1, v2 e Conector

A divisão do problema em dois pares de diagramas recíprocos foi a estratégia imaginada para formatar o experimento proposto dentro do arcabouço fornecido pelos diversos algoritmos geométricos encontrados no levantamento bibliográfico. Este tipo de abordagem é largamente utilizada na prática profissional dos escritórios que lidam com estruturas. A própria ideia de análise estrutural carrega em si o conceito de dividir o problema em partes que, embora sejam simplificações do todo, traduzam em suas inter-relações e resultados as características estruturais das formas imaginadas.

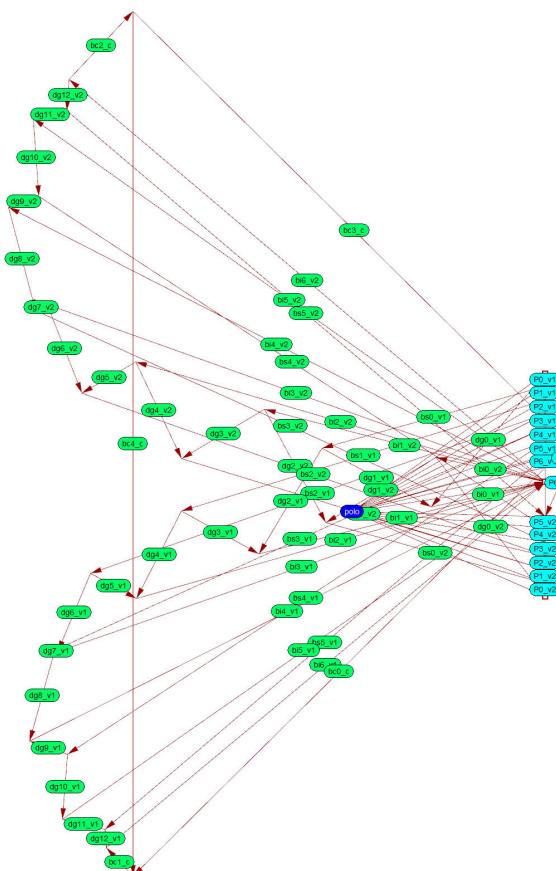
Uma premissa fundamental para o sucesso da análise é adotar uma modelo que leve em consideração as influências de um dos elementos da análise nos demais. No caso específico, as reações RA e RB determinadas para a viga do *shed*, mostradas na Figura 5.13 tem seus pontos de aplicação coincidentes perspectivante com o dos carregamentos do último nó da viga "v2" e com a força $P0_v3$ (que atua diretamente sobre o ponto $ptX1$ do conector). O vídeo 10 - cargas mostra a validação destas duas somas de vetores. O processo é simples e direto: as reações aplicadas aos nós do *shed* são iguais e contrárias às reações exercidas nos mesmos nós quando o restante da treliça é analisado. A determinação da soma é realizada graficamente, as reações são revertidas e movidas pelos seus pontos finais para os pontos iniciais das respectivas forças. uma linha ligando o ponto inicial da reação movida e o ponto de aplicação da carga representa o carregamento em cada um desses dois pontos críticos, como pode ser visto na Figura 5.16(a).

Com as contribuições do *shed* devidamente computadas, o próximo passo é determinar as reações nos apoios da treliça. O método utilizado para determinar as reações de apoios é o mesmo apresentado para a treliça apresentada na Seção 3.7.3. A aplicabilidade deste algoritmo para a presente etapa do experimento foi detectada após o cálculo da resultante dos carregamentos no PF (Figura 5.16(b)), quando resultantes perfeitamente verticais foram verificadas independentemente das combinações dos parâmetros fornecidos. Depois de observado, o motivo de tal comportamento pareceu óbvio: em primeiro lugar, todas as cargas utilizadas no experimento são verticais e as únicas forças com componentes horizontais são as reações RA e RB . Estas contudo, foram determinadas como componentes da resultante vertical das cargas atuantes no *shed*. Portanto é seguro dizer que as componentes horizontais das duas forças se anulam e que as demais cargas não possuem componentes horizontais.

Embora a nulidade das componentes horizontais possa ser verificada tanto pelos procedimentos analíticos quanto gráficos, a simplicidade visual dos diagramas da grafostática tornam a investigação das estruturas intuitiva e acurada. A determinação das reações foi registrada no vídeo 11 - Reações Pilares.



(a) Forma Geométrica - Vigas v1 e v2



(b) Polígono de Forças - Vigas v1 e v2

Figura 5.16: Diagramas Recíprocos das Vigas v1 e v2

Com o conhecimento das reações, os cálculos dos elementos das vigas "v1" e "v2" podem ser efetuados. Assim como o exemplo mostrado na Seção 3.7.4 e a grande maioria das treliças estudadas na bibliografia, os cálculos tem início nos nós onde as reações atuam. Na Figura 5.16(a) podemos identificar o primeiro nó da viga "v1" onde atua a reação R_v1 e os esforços nos elementos $dg0_v1$ e $bi0_v1$ selecionados na ordem apresentada sendo R_v1 a única força conhecida e, as demais definidas na direção e desconhecidas no módulo e sentido. No nó seguinte da mesma viga temos em ordem $dg0_v1$ e o carregamento $P0_v1$

como conhecidas e os elementos $bs0_v1$ e $dg1_v1$ respectivamente. Na viga $v2$ o processo é análogo e parte R_v2 , mas seguindo o mesmo sentido (horário) da viga "v1" o elemento do banzo inferior $bi0_v2$ aparecerá como a primeira incógnita e a diagonal $dg0_v2$ como segunda. No segundo nó a ordem dos componentes conhecidos começa pela carga ($P0_v2$) e tem sequência na diagonal $dg0_v2$, as incógnitas, em ordem são $dg1_v2$ e $bs0_v2$. O diagrama recíproco 5.16(b) mostra as construções resultantes desta seleção. As regras para o fechamento dos polígonos são as mesmas utilizadas no exemplo apresentado na Seção 3.7.4 e a construção do diagrama pode ser conferido nos vídeos 12 - viga v1 e 13 - viga v2.

O processo mostrado nos vídeos segue a ordem dos nós das diagonais de cada viga até o penúltimo. Neste ponto da execução todos os elementos de cada uma das vigas estão determinados e os algoritmos passam para a resolução dos elementos do conector, $bc0_c$ e $bc1_c$ na sequência da viga "v1" e $bc2_c$ e $bc3_c$ para a viga "v2". Para tanto, o último nó do banzo superior de cada viga é analisado. Neste último nó, para as vigas com número ímpar de diagonais, deve-se levar em conta a última diagonal, como mostrado no vídeo 13 - viga v2, fato que não ocorre no exemplo mostrado no vídeo 12 - viga v1.

Após os cálculos dos elementos das bordas das diagonais por ambos os lados, o eixo do conector é resolvido pelo ponto $ptX1$ construindo um polígono na sequência $bc0_c$, $F0$ $bc3_c$, tendo apenas o elemento $bc4_c$ como incógnita. O vídeo 14 - Eixo do Conector mostra o processo que gera uma linha vertical no PF, paralela à sua recíproca no FG. O ponto $ptX2$ é o ponto de conferência, as forças atuantes devem se anular também neste ponto, formando um polígono fechado.

5.7 Validação para Treliças Geradas por Curvas Iniciais no Sentido Dirieta-Esquerda

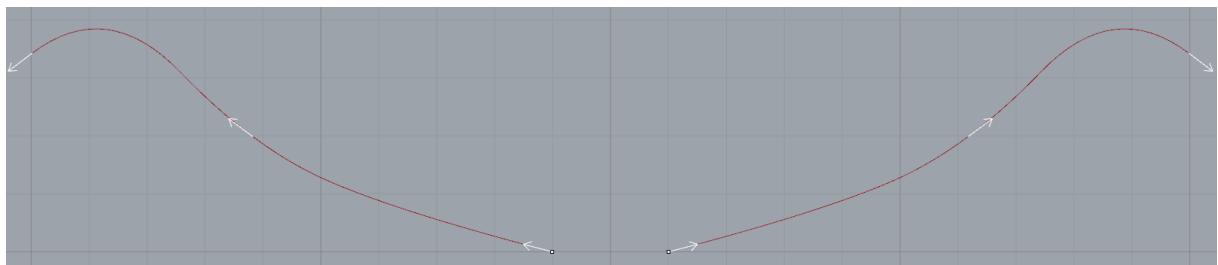


Figura 5.17: Sentido de Desenho das Curvas Iniciais

Nas seções anteriores foi mostrado como o algoritmo implementado atua de acordo com as regras apresentadas nos fundamentos da Grafostáticas, cuja prova matemática pode

ser vista nos tratados de [Cremona \(1890\)](#). A sobreposição exata dos diagramas gerados nos vídeos, obedecendo rigorosamente as regras dos cálculos gráficos com os diagramas gerados pelo algoritmo mostra a correspondência do código com o método. A medida que se adquire prática e fluência nas regras de execução dos desenhos, a aferição dos resultados gerados pelo algoritmo pode ser feita visualmente, observando os sentidos, paralelismos e pontos de interseção das linhas do PF.

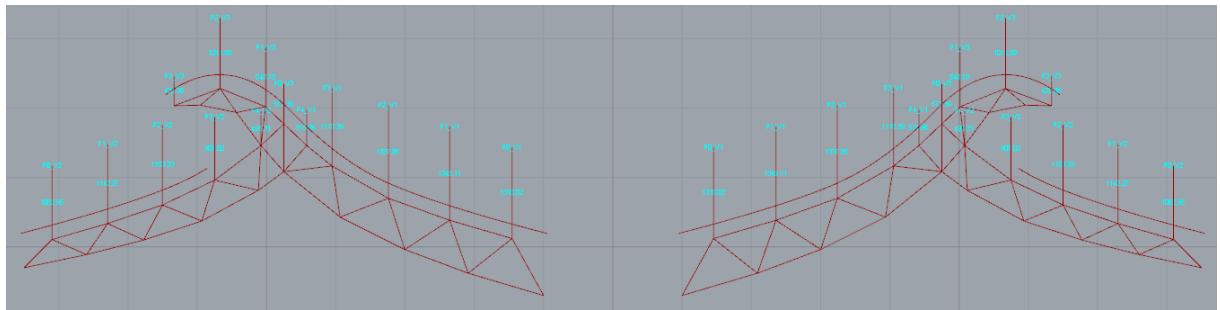


Figura 5.18: Treliças Simétricas

Todos os resultados apresentados até agora mostram diagramas gerados a partir de treliças onde a direção da viga "v1" foram grafadas da esquerda para a direita. Também foi dito que o sentido de grafia das curvas iniciais deve partir do ponto de apoio da viga "v1" na direção do ponto final da treliça do *shed*, podendo ser tanto no sentido esquerda-direita, quanto no direita-esquerda (Figura 5.17). Considerando o segundo caso, como exposto previamente, o sentido de seleção das forças ao redor do nó será anti-horário.

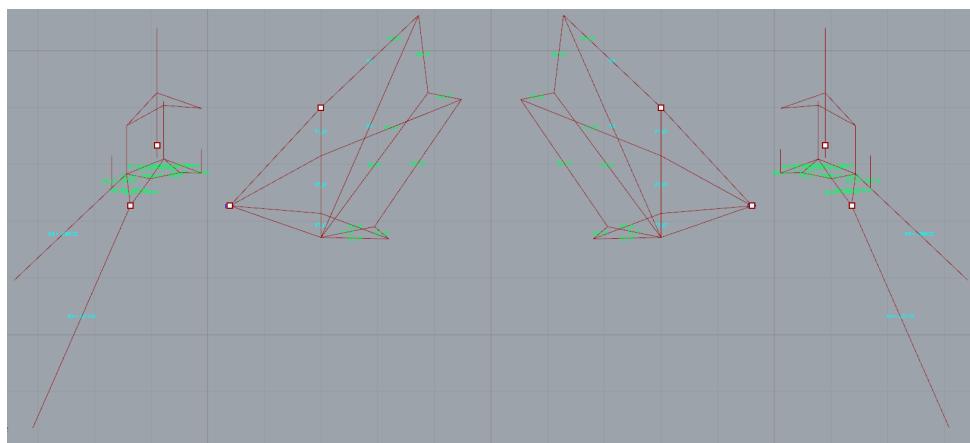


Figura 5.19: PFs e FGs dos Sheds Simétricos

A validação para esses segundos casos pode ser feita pelos mesmos métodos usados no tratamento do primeiro, contudo a inversão do sentido da viga acompanhada da inversão do sentido de seleção das linhas de ação cria uma situação particular e interessante que pode ser usada para verificar o algoritmo: Se espelharmos uma curva inicial em relação ao eixo y do plano de desenho e o mesmo espelhamento seja aplicado aos pontos inicial

e polo dos PFs obteremos diagramas simétricos para as treliças geradas respectivamente para os objetos originais e espelhados.

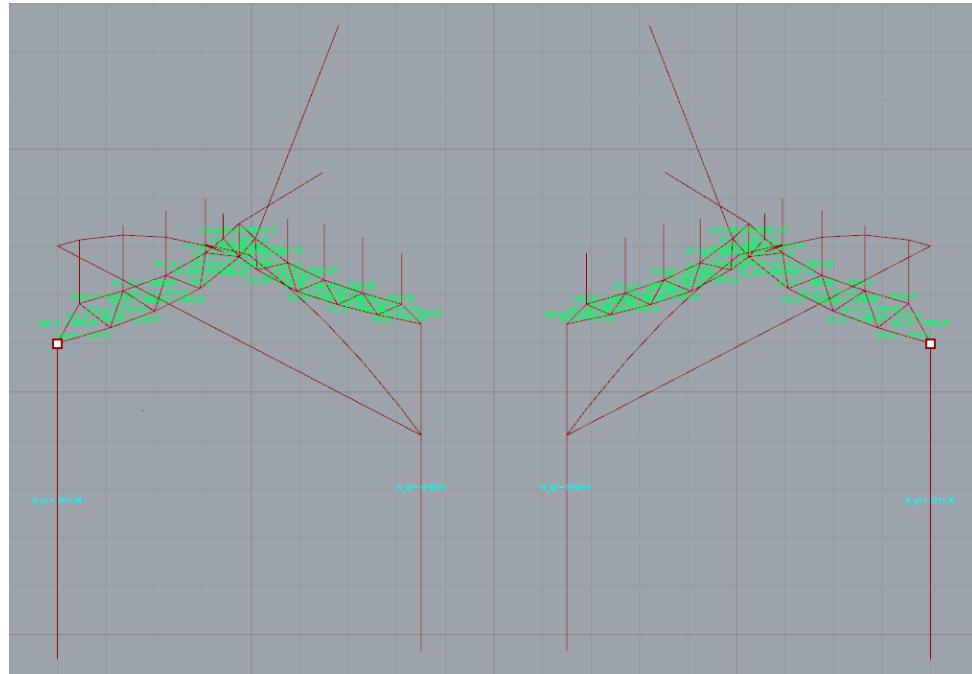


Figura 5.20: FGs das Vigas v1 e v2 Simétricas

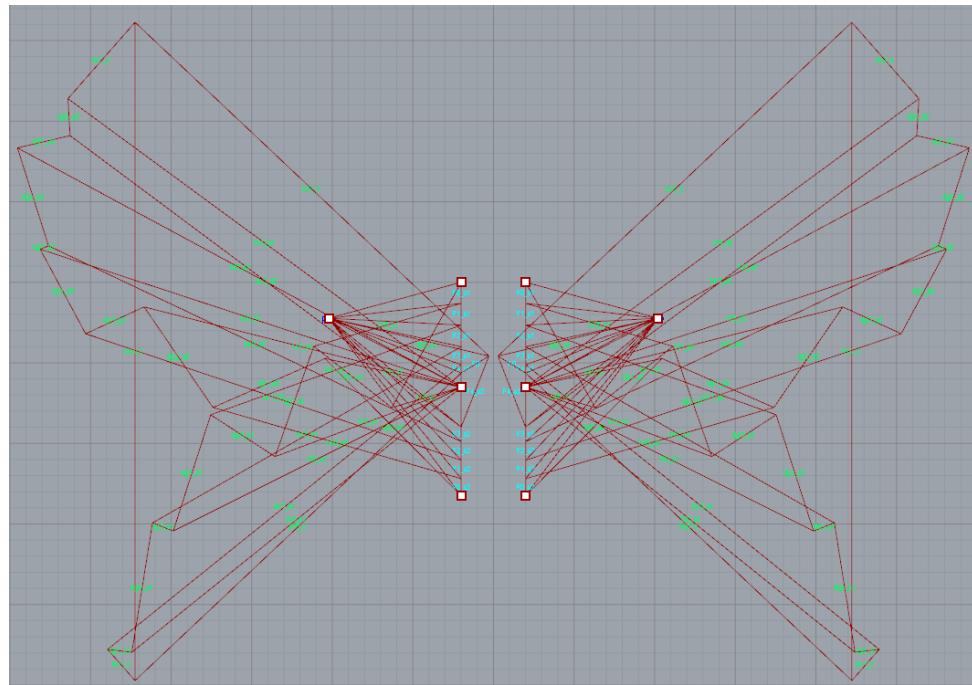


Figura 5.21: PFs das Vigas v1 e v2 Simétricas

Considerando devidamente validadas as construções geométricas nos casos onde a curva inicial está grafada da esquerda para a direita, espelhando os objetos especificados, encontra-se simetria em todas as linhas desenhadas, correspondência nos valores calculados e na

natureza (tração ou compressão) das solicitações atuantes nos componentes. Esse simples procedimento soma-se às vantagens do método gráfico, onde a consciente avaliação dos desenhos é um importante instrumento de correção, verificação e validação dos cálculos realizados.

5.8 Teste do Dimensionamento dos Membros da Treliça

Após o cálculo da força em cada elemento da treliça, uma função (teste_elemento) testa se as dimensões atribuídas as seções de cada membro são capazes de suportar a solicitação que estes estão submetidos. São realizados dois testes: um para as peças tracionadas e outro para as comprimidas.

Para as peças tracionadas, a tensão admissível é calculada a partir da tensão limite de escoamento do material (parâmetro T_Trac) dividido pelo coeficiente de segurança (parâmetro coe_Trac). A carga atuante na peça é dividida pela área da seção e, caso o valor obtido seja menor que a tensão limite, o elemento passa no teste e seu texto é grafado na cor verde, caso contrário a peça precisa ser redimensionada e é escrita na cor magenta.

Para os elementos comprimidos, a tensão crítica de flambagem é comparada com a tensão atuante na peça. O índice de esbeltez da peça (λ) é obtido pela equação 5.2, onde l corresponde ao comprimento de flambagem da peça (para elementos com fixações articuladas em ambos os extremos o comprimento de flambagem corresponde ao comprimento total do eixo da peça), r é o raio de geração, I o menor momento de inércia da peça e A a área da seção.

$$\lambda = \frac{l}{r} , \text{ sendo } r = \sqrt{\frac{I}{A}} \quad (5.2)$$

O valor de λ é comparado com um valor mínimo (lido no parâmetro esb_min). Caso λ seja maior que o mínimo, utiliza-se a fórmula de Euler para determinação carga crítica. A Equação 5.3 apresenta a fórmula de Euler na forma que retorna a tensão admissível ($\bar{\sigma}_{fl}$), sendo E é o módulo de elasticidade do material (parâmetro Mod_E) e s o coeficiente de segurança para a flambagem (parâmetro coe_Comp).

$$\bar{\sigma}_{fl} = \frac{\pi^2 \times E}{\lambda^2 \times s} \quad (5.3)$$

Quando o valor de λ é menor que o mínimo, a fórmula de Euler não pode ser aplicada.

Utiliza-se então uma das fórmulas empíricas para determinação da tensão admissível. Optou-se pela Equação 5.4, conhecida como fórmula parabólica do I.A.S.C. (*American Institute of Steel Construction*). O resultado de $(\bar{\sigma}_{fl})$ obtido já leva em conta coeficientes de segurança, entretanto deve-se ter em mente que os valores assim calculados tem por unidade lb/in^2 , um fator de conversão é utilizado para comparar com os valores em Pa da tensão atuante em cada barra (NASH; SILVA, 1974, p. 267-269).

$$\bar{\sigma}_{fl} = 17000 - 0,485 \times \lambda \quad (5.4)$$

Assim como para as peças tracionadas, a tensão atuante na peça (P/A) é comparada com o valor admissível calculado e as mesmas cores são atribuídas para os casos em que o dimensionamento é confirmado ou descartado.

Além das cores, mensagens são direcionadas para a saída *Out* do componente indicando o nome do elemento que não foi aprovado pelo teste e se a verificação falhou na análise da tensão admissível para tração ou para flambagem.

5.9 Parâmetros do Algoritmo de Análise Grafostática

1. **Iniciar_Analise:** Funcionamento análogo ao do parâmetro Mostrar_Volume do Algoritmo de Geração dos Volumes. Recebe uma variável booleana (True or False) que ativa e desativa a execução da análise estrutural da treliça.
2. **Cargas:** Recebe três listas com os valores dos carregamentos em cada nó do banco superior, provenientes da saída de mesmo nome do Algoritmo de Geração dos Volumes. As listas contêm os valores numéricos ordenados das cargas para cada uma das treliças.
3. **Areas:** Uma lista com a área dos elementos dos bancos e diagonais da treliça e os respectivos momentos de inércia mínimo para cada seção. Recebe os valores da saída de mesmo nome do Algoritmo de Geração dos Volumes.
4. **T_Trac:** Recebe um número real correspondente à tensão de escoamento do material da treliça.
5. **coe_Trac:** O coeficiente de segurança é um valor real que serve como fator de majoração no cálculo da tensão admissível dos elementos submetidos à tração. Os valores dos coeficientes de segurança para o aço recomendados pela literatura costumam variar entre 1,5 e 2,0. Caso nenhum valor seja lido pela entrada, o algoritmo assume o valor do coeficiente como 2,0.

6. **Mod_E:** Módulo de elasticidade do material. Caso nenhum valor seja designado para o parâmetro, assume-se 200 GPa.
7. **coe_Comp:** Coeficiente de segurança para cálculo de elementos submetidos a flambagem. Caso nenhum valor seja lido pela entrada, o algoritmo assume o valor do coeficiente como 2,0.
8. **esb_min:** Índice de esbeltez mínimo, utilizado pelo teste de resistência à flambagem. Caso nenhum valor seja lido pela entrada, o algoritmo assume o valor do coeficiente como 105.
9. **Escala:** Define as escalas em que as forças serão grafadas nos diagramas. Caso nenhum valor seja lido pela entrada, o algoritmo assume o valor de 0,0001 (1/1000 em notação conveccional de escalas). Significando que o comprimento das linhas será 1000 (mil) vezes menor do que a força representada por esta.
10. **Eixos:** Os eixos da treliça. Rebebe a geometria ds saída de mesmo nome do Algoritmo de geração dos Eixos.
11. **polo_1:** Recebe o ponto que sera usado como polo do PF da treliça do *shed*.
12. **pto_inicial_1:** Recebe o ponto que sera usado como ponto inicial do primeiro carregamento transportado da FG para o PF da treliça do *shed*. Os pontos *polo_1* e *pto_inicial_1* devem ser coplanares em um plano paralelo ao plano de desenho.
13. **pto_base_FG1:** Ponto base pelo qual pode-se mover a FG da treliça do *shed*.
14. **polo_2:** Recebe o ponto que sera usado como polo do PF das treliças "v1", "v2"e conector.
15. **pto_inicial_2:** Recebe o ponto que sera usado como ponto inicial do primeiro carregamento transportado da FG para o PF das treliças "v1", "v2"e conector. Os pontos *polo_2* e *pto_inicial_2* devem ser coplanares em um plano paralelo ao plano de desenho.
16. **pto_base_FG2:** Ponto base pelo qual pode-se mover a FG das treliças "v1", "v2"e conector.
17. **Plano:** Plano de desenho. recebe valores da saída de mesmo nome do Algoritmo de geração dos Eixos e tem função Análoga à entrada de mesmo nome do Algoritmo de Geração dos Volumes.

5.10 Funções do Algoritmo de Análise Grafostática

1. **estendeCorda:**

- Entradas:(linha, ponto, indice).
- Função: Quando os raios polares são ordenadamente movidos do PF para a FG, para o desenho correto do diagrama, a nova linha (corda) deve ter seu comprimento determinado pelas interseções de duas linhas de ação de forças consecutivas. A função reescreve a linha para que seja assim delimitada.
- Saídas: A corda em sua posição e dimensões apropriadas.

2. Linha_force_extena:

- Entradas:(no, peso, conv)
- Função: Desenha uma linha em escala representando o vetor de uma força externa, com ponto final coincidindo com o nó.
- Saídas: Uma reta representando a linha de ação da força.

3. Linhas_de_Cargas:

- Entradas:(vugas, cargas, nome)
- Função: Pega os valores coletados no parâmetro **Cargas** e transforma em linhas ordenadas e devidamente locadas no respectivo nó. Usa a função **Linha_force_extena** para cada uma das cargas. Também nomeia as forças em ordem crescente. A entrada "nome" é usada para gerar o sufixo da viga em questão (_v1, _v2, _v3).
- Saídas: Uma lista de retas (vetores) e uma lista de triplas para nomear as cargas

4. Elementos_vigas:

- Entradas:(lista de polilinhas que forma uma das 3 vigas)
- Função: As vigas, recebidas pela entrada **Eixos**, são escritas como polilinhas. A função gera as listas ordenadas dos elementos do banzo superior, inferior e diagonais, bem como a lista de nós dos mesmos.
- Saídas: listas dos elementos e dos nós para o banzo superior, banzo inferior e diagonais de uma viga.

5. SentidoNo:

- Entradas:(CirDir, no, plano , raio)
- Função: desenha um círculo ao redor do nó na mesma direção do círculo mostrado na Seção 5.5.1. para ser usado pela função que ordena as forças ao redor do nó.
- Saídas: círculo

6. TraComp:

- Entradas:(no, linha do PF, recíproca na FG)

- Função: Testa a naturaza do esforço do elemento (tração ou compressão).
- Saídas: retorna o valor 1 para tração e -1 para compressão.

7. teste_elemento:

- Entradas:(nome, carga, linfg)
- Função: Testa se o elemento resiste às forças internas atuantes, como apresentado na Seção 5.8
- Saídas: retorna um código RGB. Caso o elemento seja capaz de suportar a tensão aplicada, o código da cor ciano é passado. Caso contrário, o código RGB da cor magenta é retornado como saída.

8. elemntos_node:

- Entradas: (no, lista de elementos, prefixo da lista, sufixo da viga)
- Função: Dada uma lista de elementos ou carregamentos quaisquer e um nó, a função determina quais dentre os membros da lista atuam diretamente sobre o nó em questão.
- Saídas: Uma lista dos elementos que atuam sobre o nó e uma lsita na mesma ordem com a nomenclatura de cada um dos elementos.

9. OrdenaLinhasDeAcao:

- Entradas: (no, CirDir, Linhas, nomes, dicPF, Plano)
- Função: Coloca as linhas de ação na ordem correta para a determinação dos elementos desconhecidos no PF. Recebe com uma lista de linhas da FG que atuam no nó e seus respectivos nomes. Acha o menor segmento da lista e usa metade do valor do comprimento como raio do círculo desenhado pela função **SentidoNo** com centro no próprio nó. Na interseção das linhas com o círculo, é medido o comprimento do arco entre um ponto da circunferência considerado como inicial e o ponto de interseção com cada reta. A lista de linhas e a de nomes é ordenada de maneira crescente em relação a medida deste comprimento de arco. A entrada dicPF é uma estrutura de dados chamada dicionário, que consiste em associar objetos (no caso, as linhas das forças no PF) á textos (nomenclatura atribuída a cada barra). A função testa se o primeiro item da lista de nomes já está no dicionário do PF. Em caso positivo, o primeiro elemento das listas de nomes e linhas é reposicionado para o final da lista. O teste é repetido até que o primeiro elemento elemento das listas não esteja no PF. Um teste semelhante é feito então, procurando elementos que não estejam no PF. Encontrando assim uma sequência de linhas, no sentido estabelecido, que começa nos elementos já calculados e termina no(s) não calculado(s).
- Saídas: Lista de elementos, lista com os nomes dos elementos, número de elementos já calculados no PF e número de elementos por calcular no nó.

10. Cremona2:

- Entradas:(no, nomes, linhas, countPF , dicPF)
- Função: Caso o número de incógnitas no seja de duas forças, esta função é chamada para solução do nó. O dicionário do PF é usado para , através da lista de nomes ordenada pela função **OrdenaLinhasDeAcao**: encontrar os pontos extremos dos lados abertos do polígono das recíprocas das forças no PF.
- Saídas: Um dicionário com as duas forças calculadas aqui para atualização do dicionário do PF, duas triplas contendo as informações de nomenclatura.

11. Cremona1:

- Entradas:(no, nomes, linhas, countPF , dicPF)
- Função:Caso o número de incógnitas no seja de uma força, esta função é chamada para solução do nó. Um teste de paralelismo é realizado entre as reciprocas da incógnita no PF e na FG. Caso sejam paralelas, a *string* "PARALELISMO" é impressa na saída *out* do componente.
- Saídas:Um dicionário com a força calculada aqui para atualização do dicionário do PF, uma tripla contendo as informações de nomenclatura.

12. PF_funic:

- Entradas:(pto_inicial, polo, carreg, nomes_cargas)
- Função: Desenha os carregamentos no PF, os raios polares e o funicular ainda aberto na FG. Nesta função o dicionário do PF é criado e contém inicialmente apenas os vetores dos carregamentos ligados aos nomes.
- Saídas: Dicionário do PF, lista dos raios polares, lista das cordas do funicular, A resultante no PF e triplas para nomenclatura.

13. Grafo_Shed:

- Entradas:(viga do shed, carregamentos, Plano, DicPF, CirDir, nome da viga)
- Função: Percorre os nós das diagonais, determinando a nomenclatura, ordenando os elementos , calculando as forças internas das barras e atualizando o dicionário do PF.
- Saídas: Dicionário do PF atualizado, triplas para a nomenclatura

14. Grafo_Viga:

- Entradas:(viga , carregamentos, Plano, DicPF, CirDir, nome da viga)
- Função: Mesmo que a função **Grafo_Shed**: para os cálculos das vigas "v1"e "v2"
- Saídas: Dicionário do PF atualizado, triplas para a nomenclatura

5.11 Saídas do Algoritmo de Análise Grafostática

- **Out:** Imprime os testes de paralelismo para as soluções encontradas com a função **Cremona1**.
- **Linhas_de_Carga:** As linhas representando os vetores de carregamento são desenhadas em escala sobre os respectivos pontos dos eixos gerados pelo componente de geração dos eixos.
- **txt_pontos:** Nomenclatura dos elementos. A saída apresenta uma lista de triplas, sendo o primeiro valor um ponto que sera usado para a inserção do texto, a *string* do texto e a cor em que ele deve ser escrito. Um conjunto de componentes, conectados à direita desta saída, escreve os textos no ambiente do *Rhinoceros* através destas 3 informações.
- **FG1:** A FG para a viga do *shed*.
- **PF1:** O PF para a viga do *shed*.
- **raios_1:** Os raios polares para a viga do *shed*.
- **funicular_1:** O funicular desenhado para a viga do *shed*.
- **resultante_1:** As resultantes para a viga do *shed*.
- **FG2:** A FG das vigas v1, v2 e conector.
- **PF2:** O PF das vigas v1, v2 e conector.
- **raios_2:** Os raios polares do PF das vigas v1, v2 e conector.
- **funicular_2:** O funicular desenhado para as vigas v1, v2 e conector.
- **resultante_2:** As resultantes para as vigas v1, v2 e conector.

5.12 Aplicação do Algoritmo de Análise Grafostática

O uso do componente de análise grafostática segue um modelo semelhante ao do desenho das formas. As informações fornecidas para o início dos trabalhos do algoritmo estão apresentadas no vídeo [15 - Análise Grafostática pt1](#). O papel das análises das características, no método de trabalho apresentado, é a de antecipar comportamentos e realimentar etapas de trabalho que, no método tradicional, precederiam a análise em si. O trabalho de modelagem das formas executado antes da análise, caso os resultados desta pretendam ser usados, não simplesmente como avaliação *pass or fail* (como mostradas no vídeo [16 - Análise Grafostática pt2](#)), mas como informações a serem usadas para a melhoria de

aspectos do projeto, teria que ser, em muitos pontos refeito. A maneira imediata com que as informações da análise podem guiar mudanças e ajustes da forma é descrito no vídeo [17 - Análise Grafostática pt3](#).

O entendimento dos significados de cada linha nos diagramas recíprocos, assim como o estudo analítico de estruturas, requer um tempo considerável de dedicação aos seus conceitos fundantes e ao conhecimento das sequências de execução, objetivos e inter-relações dos seus algoritmos de desenho. A medida que os parâmetros são flexionados e instantaneamente os rebatimentos tanto da forma quanto da análise são gerados, é possível observar e concluir pela simples tendência de contração ou dilatação do polígono de forças os efeitos sobre a forma e o comportamento analisado. Embora não abranja todo o escopo de cálculo necessário para a completa verificação da treliça, o método se apresenta aberto ao julgamento da equipe de projeto, sendo submetido à critérios subordinados ao conhecimento teórico, prático e estético destes profissionais.

Considerações finais

”Learn from yesterday, live for today, hope for tomorrow.

The important thing is to not stop questioning.”

(Albert Einstein)

Uma considerável gama de conhecimento foi abordada ao longo do processo que culminou na proposição do Sistema Generativo de Projeto desenvolvido ao longo do experimento. Esse SGP, que engloba os três algoritmos propostos e as possibilidades e limitações de um ambiente computacional capaz de dar suporte ao projeto, foi desenvolvido com o objetivo principal de apresentar uma forma de trabalho exequível. Especialmente voltada para as primeiras etapas de elaboração de projetos, levou-se em consideração a profundidade das implicações que estas etapas exercem sobre todo o resultado.

No caso específico do desenho de uma cobertura tipo *shed*, imaginou-se que a medida do vão vencido pela treliça estaria previamente determinado em uma etapa ainda anterior de projeto. Uma funcionalidade recorrente dos algoritmos baseados em parâmetros e restrições é a possibilidade de rapidamente atualizar uma etapa de trabalho em função das demandas de outra. Caso a modulação dos pilares seja alterada, é fácil notar que, com poucas alterações nos parâmetros de controle da forma, é possível atingir esse objetivo. Outros elementos construtivos serão adicionados e os conflitos gerados não podem ser completamente previstos. Nesses casos conta-se basicamente com a capacidade do algoritmo de produzir variações.

No que diz respeito ao Algoritmo de Análise Grafostática, com as limitações do método e do escopo teórico devidamente postas, algumas questões devem ser levantadas. O componente de análise grafostática foi escrito sob medida para as saídas dos outros dois. Dentro do ponto de vista dos SGP é uma estratégia possível. Cabe uma rápida observação sobre a extensão de cada algoritmo: enquanto, entre linhas de comando e comentários, os dois primeiros algoritmos possuem menos de 300 linhas cada um, o terceiro tem pouco menos de 900. Considerando que o objetivo de um SGP deve ser o suporte ao trabalho na atividade fim, e não a elaboração de um *software*, o tempo despendido na programação deve ser pesado de acordo com os benefícios oferecidos ao projeto. Caso a grafostática seja eleita como ferramenta de análise estrutural, é apropriado direcionar a programação para a elaboração de uma biblioteca de funções que possam ser reaproveitadas em códigos futuros. Entre as funções apresentadas na seção 5.10, apenas as duas últimas são especialmente escritas para o problema em foco, e portanto precisariam passar por reescrita para sua aplicação, as demais poderiam fazer parte de análises para diferentes problemas.

6.1 Conclusões

Os algoritmos de geração da forma tem como principal ferramenta de variação a curva inicial. O desenho desta pode partir do que costuma chamar-se de gesto: um movimento de desenho grafado por um lápis no papel ou em bites no ambiente computacional. O gesto representa o que existe de mais humano na concepção de um projeto e relaciona-se com algumas propostas artísticas de criação arquitetônica. Os algoritmos de parâmetros e restrições propostos são calcados em um sequência de construções geométricas semelhantes às apresentadas na Figura 2.1, estendendo-se porem, até esboçarem uma ideia de um objeto a ser construído. As flexões dos parâmetros representam etapas do algoritmo em que decisões podem ser imediatamente revistas. Seguindo a mesma partitura, as mudanças nessas decisões produzem significativas variações do mesmo tema e transformam os dados em formas geométricas coerentes para a análise e construção. O parâmetro “eixo_de_simetria” produz uma variação particular nas formas. As pequenas mudanças no seu valor geram surpreendentes rebatimentos no desenho da treliça, principalmente se levarmos em conta que tudo que o parâmetro faz é escolher um ponto sobre a curva inicial e fornecê-lo para os passos seguintes do algoritmo. A Figura 6.1 mostra quatro treliças iguais em todos os parâmetros, exceto pelo eixo de simetria, que assume os valores de .75, .70, .65 e .60.

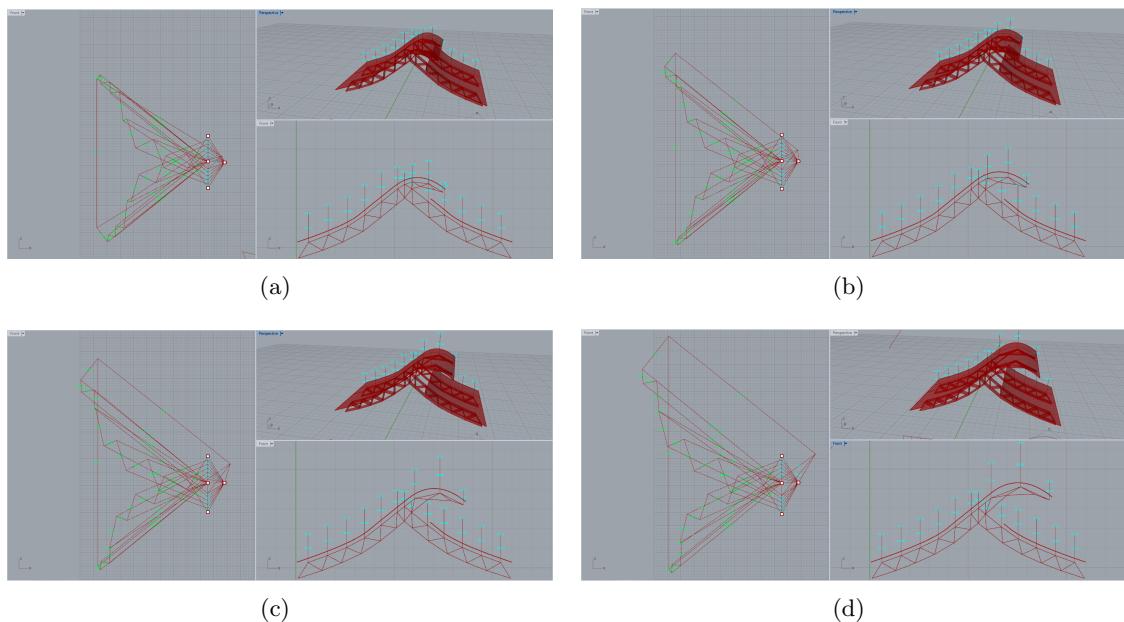


Figura 6.1: Variações do Eixo de Simetria de uma Treliça

As aplicações computacionais para Arquitetura e Engenharia marcaram seu território nos escritórios e empresas de construção e projeto por todo o mundo, e a computação, na sua relação com o complexo mundo contemporâneo serve de fonte de inspiração para muitos projetos. Esta pesquisa porem, concorda com os autores que afirmam que os algoritmos generativos não definem um estilo arquitetônico. Embora as apreciações estéticas não

façam parte do escopo do trabalho, os exemplos de utilização de algoritmos na Arquitetura perpassam a linha do tempo das vilas de Paládio às formas orgânicas contemporâneas, mostrando métodos de trabalho aplicáveis à concepção de projetos dos mais variados estilos.

Do início das civilizações aos dias atuais a atividade de construir contou com a constante companhia da Geometria. Alguns dos modelos de algoritmos apresentados na seção 2.2, como os autômatos celulares, os diagramas de Voronoi e as gramáticas de forma, possuem uma clara representação geométrica, os algoritmos de otimização e de inteligência artificial só são viáveis como algoritmos generativos quando apresentam um rebatimento geométrico, já o termo "Geometria Fractal" fala pro si.

Na análise estrutural, os elementos construtivos são representados por formas geométricas, as forças, definidas como vetores, podem ser assim representados. A Grafostática contudo, transforma o próprio cálculo em um processo de construção geométrica. Ao invés de traduzir uma geometria em equações, os métodos da Estática Gráfica relacionam formas a serem construídas a diagramas que representam o comportamento estrutural do objeto projetado.

Muitas informações podem ser capturadas pela análise do modelo e seus diagramas. Na Figura 6.1, temos quatro treliças geradas apenas pela variação do valor do parâmetro do eixo de simetria. As vigas "v1" e "v2" da treliça são perfeitamente simétricas. Os diagramas do PF mostrados seriam também simétricos (em relação a um eixo horizontal no ponto de interseção das reações), caso os carregamentos também fossem. A maior contribuição de forças assimétricas vem das reações da viga do *shed*. Quanto menor o valor do parâmetro analisado, maior será o *shed* e maiores serão os carregamentos provocados por suas reações, a assimetria do PF e a magnitude das solicitações (principalmente nos elementos mais próximos do *shed*). Este conjunto de análises pode ser alcançado por métodos gráficos, analíticos e numéricos (ou pelo conhecimento bem estruturado dos fundamentos da mecânica das estruturas), porém a noção de como a estrutura se comporta em relação à mudanças (anda que pequenas) em seus parâmetros pode ser observada em tempo real no SGP proposto. A informação das consequências, nas formas e análises, destas interferências fornecem ao projetista uma visão única sobre o rebatimento de pequenas decisões na forma e comportamento do elemento construtivo analisado.

6.2 Contribuições

O trabalho contribui no debate sobre os Sistemas Generativos de Projeto nos seus aspectos teóricos e práticos. Apresentando um quadro geral dos modelos e procurando uma definição para o conceito de Algoritmos Generativos, onde se ve alguns modelos computacionais que despertaram grande interesse científico nas últimas décadas, aplicados na

geração de formas para a indústria da construção civil. O painel formado pelas diversas abordagens de modelagem dos problemas tratados pelos SGP evidenciam a abrangência e multidisciplinaridade destas aplicações, e corroboram com o já alardeado potencial desses modelos na resolução e tratamento de questões complexas.

No aspecto prático, o experimento testa uma das maneiras de se direcionar a investigação das soluções possíveis de projeto com o auxílio dos SGP. A forma inicial é estabelecida junto com o conjunto de regras que permitem a flexão das formas e a criação de variações.

O algoritmo de análise grafostática, escolhido como instrumento de avaliação do desempenho estrutural da treliça, apresenta uma modelagem bastante particular e interessante por ser também um modelo de parâmetros e restrições. Não pode ser considerado como Algoritmo Generativo já que as formas geradas não tem objetivo de serem construídas, mas de servirem como instrumentos de análise e tomada de decisões. A natureza geométrica do método permite uma fácil conexão com a variação das formas. A pesquisa procurou no seu levantamento bibliográfico tanto autores atuais que defendem a importância e aplicação dos métodos da Estática Gráfica, quanto os manuais que apresentam os diversos algoritmos desenvolvidos ao longo da história do método.

A associação da Grafostática com um SGP em um ambiente de implementação dinâmico enfatiza as características cognitivas do método. As flexões dos parâmetros da treliça implicam diretamente na alteração da forma do polígono de forças. Essa constatação valida a hipótese que os mecanismos do algoritmo baseado na Estática Gráfica teriam semelhanças com os SGP. Pela intercomunicação destes modelos, as implicações das mudanças da formas nos desempenhos estruturais é percebida visualmente pelas alterações no comprimentos das linhas do PF. A noção das consequências que implicam cada alteração de parâmetros é percebida dentro do contexto das interdependências das características modeladas.

6.2.1 Principais Produtos e Algoritmos Desenvolvidos

- O SGP e os vídeos registrando seu funcionamento e operação, apresentados no Capítulo 5, constituem os produtos fundamentais desta pesquisa.
- Os algoritmos de geração do Triângulo de Sierpinski e seu análogo tridimensional, utilizados para ilustrar a Seção 2.2.7 e apresentados no Apêndice B.1.
- O algoritmo de geração de cúpulas móveis, exemplificando os conceitos dos MPR na Seção 2.2.8 e apresentado no Apêndice B.2.
- O algoritmo de Desenho do PF, FG, Funicular e cálculo da Resultante e Momento, e o

vídeo que registra seu funcionamento, apresentado na Seção 3.5 e Apêndice A.3 como recurso didático para o entendimento das propriedades dos polígonos funiculares.

- O algoritmo de soma de vetores e cálculo da equilibrante, descrito na Seção 4.2 e Apêndice A.1.

6.3 Atividades Futuras de Pesquisa

Os sistemas generativos de projeto são um campo de pesquisa vasto e complexo. Grande interesse tem se mostrado, particularmente na criação de ferramentas de desenho para formas orgânicas e complexas. O desejo de contribuir neste contexto passa pela ideia de, não simplesmente desenhar e detalhar essas formas, mas de propor SGPs que tenham a capacidade de analisar os desempenhos destas em toda a sua complexidade.

A Grafostática despertou bastante interesse principalmente pela sua natureza geométrica e seu caráter intuitivo. Sob o ponto de vista cognitivo, a aplicação destes métodos no ensino das disciplinas de estrutura deve ser explorado, e potencializado com as estratégias de programação. Dos muitos algoritmos encontrados na pesquisa bibliográfica, os que tratam das flechas e deformações de estruturas seriam o próximo passo para ampliar a eficácia da análise.

Das formas geradas pelos algoritmos muitos passos devem ser seguidos até a execução plena da treliça. Folgas entre as peças, chanfros e recortes nas peças metálicas deverão ser considerados para possibilitar os encaixes e as soldas que manterão a estrutura coesa. Uma das linhas de pesquisa atuais no campo da construção, denominada de *Design Optimization and Fabrication* (DOF), incorpora etapas de construção controladas por computadores (cortes de chapa, impressões 3D) e aproxima o projeto da atividade fim.

Algoritmos Baseados na Grafostática

A.1 Algoritmo de Soma de Vetores e Cálculo da Equilibrante

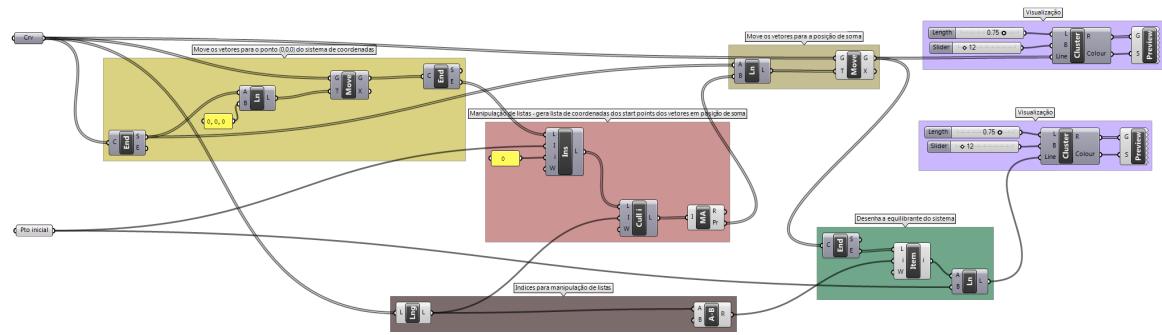


Figura A.1: Captura de Tela do Algoritmo de Soma de Vetores

A.2 Cluster Utilizado para a Visualização de Vetores

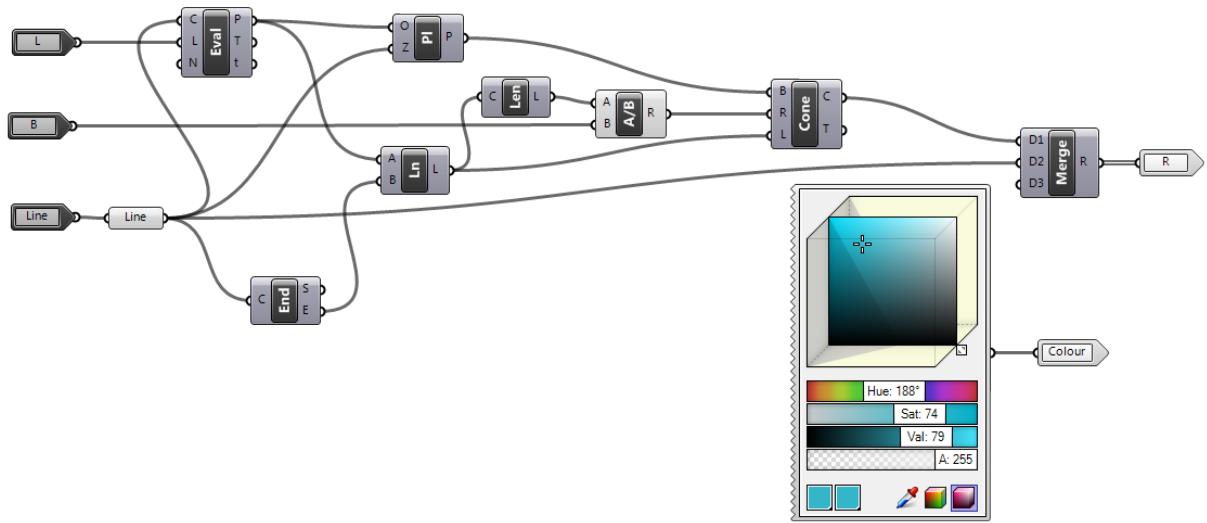


Figura A.2: Captura de Tela do Algoritmo do Cluster de Visualização de Vetores

A.3 Algoritmo de Desenho do PF, FG, Funicular e Cálculo da Resultante e Momento

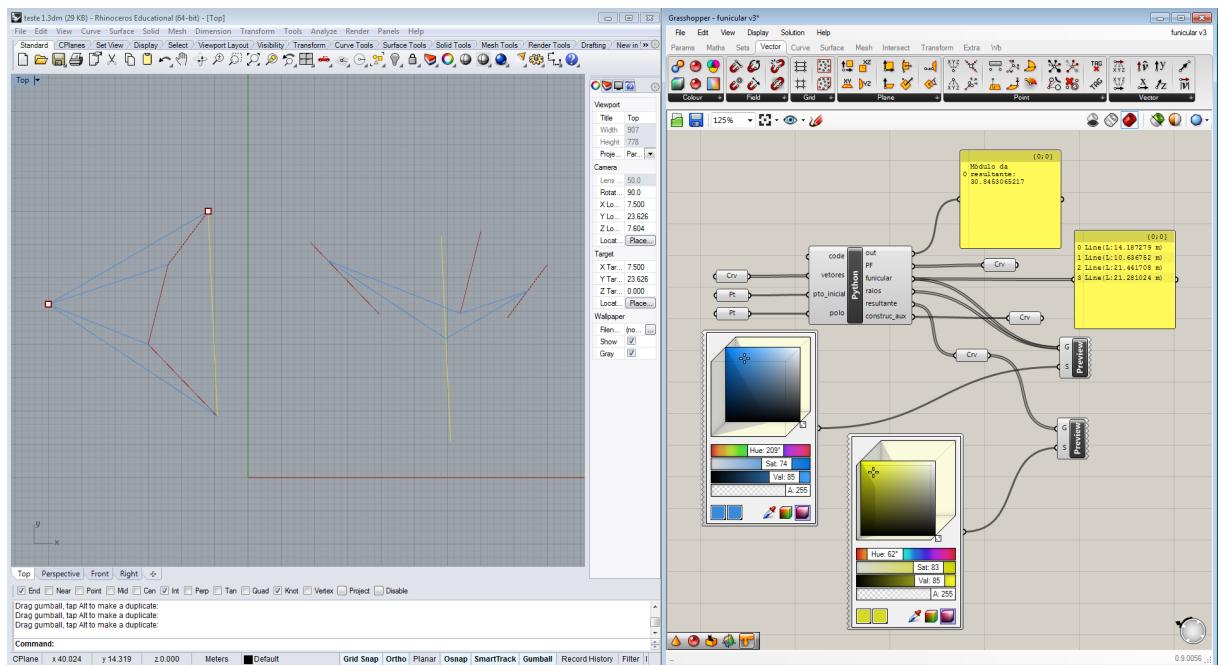


Figura A.3: Captura de Tela do Algoritmo de Desenho do PF, FG, Funicular, Resultante e Momento

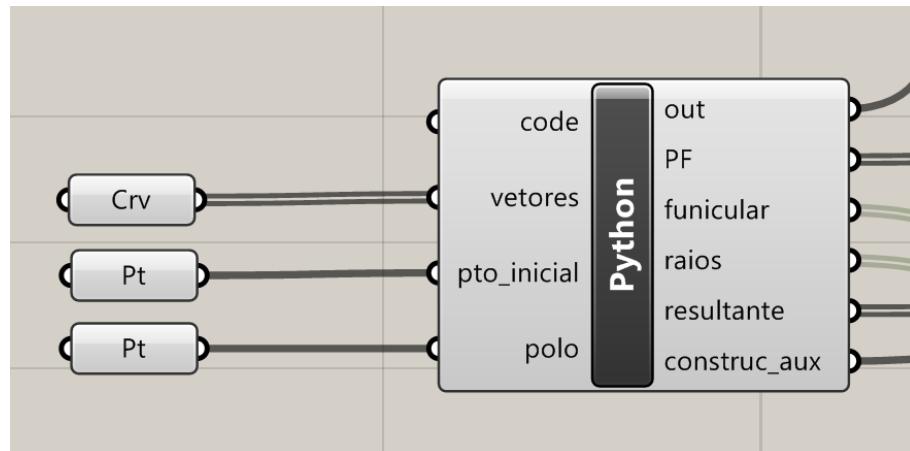


Figura A.4: Entradas e saídas do componente

Vídeo registrando o funcionamento do algoritmo disponível em:

<https://www.youtube.com/watch?v=pIa0o2M0-s>

Algoritmos Generativos no *Rhinoceros 3D / Grasshopper*

B.1 Triângulo de Sierpinski

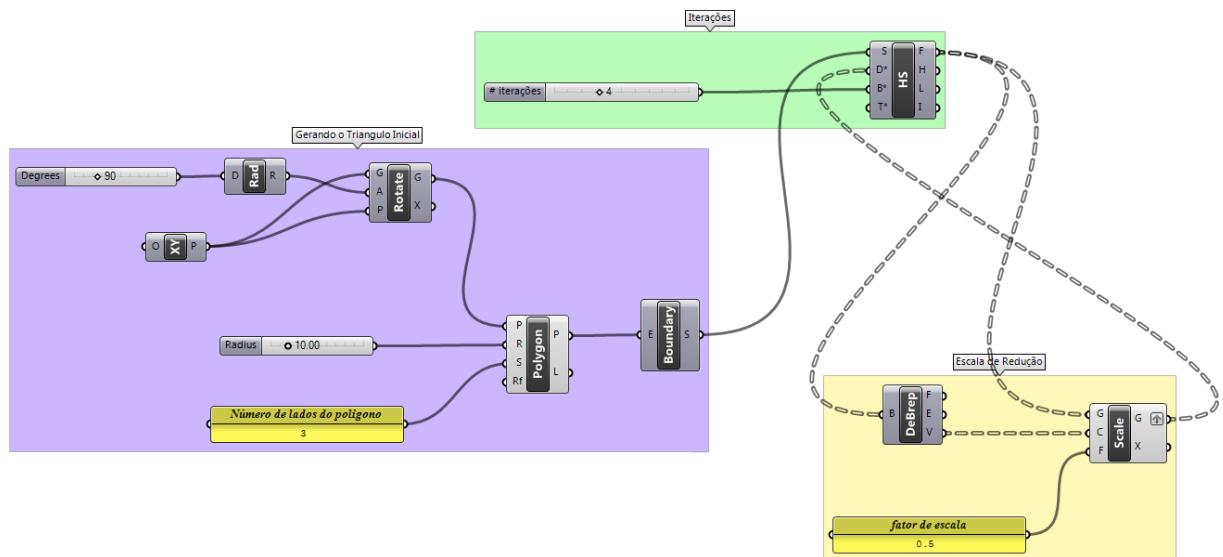


Figura B.1: Algoritmo Paramétrico do Triângulo de Sierpinski 2D

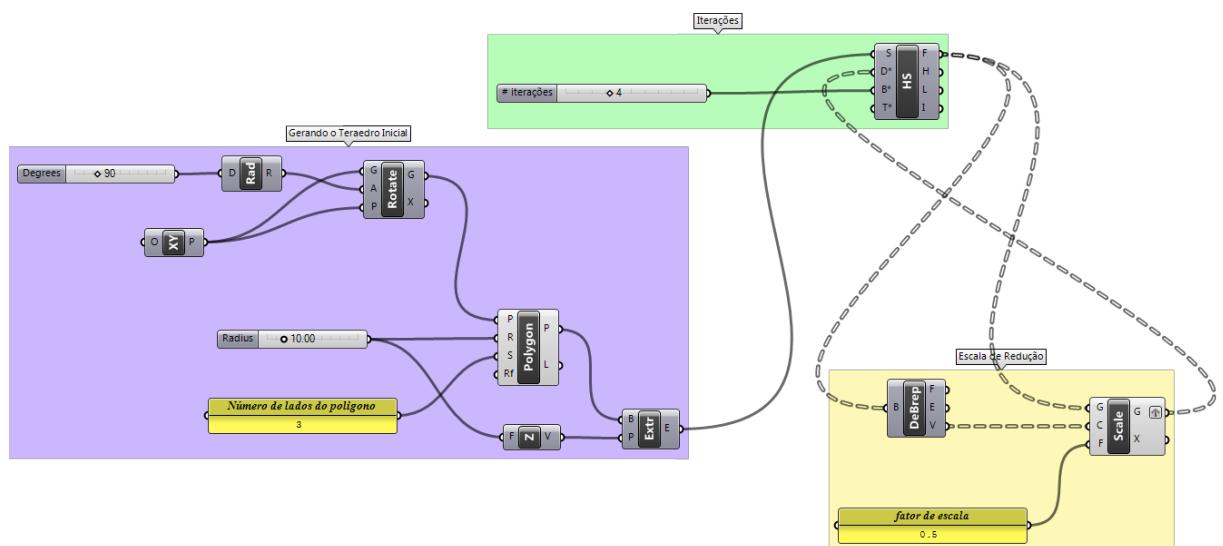


Figura B.2: Algoritmo Paramétrico para Análogo Tridimensional do Triângulo de Sierpinski 3D

B.2 Cúpula Móvel

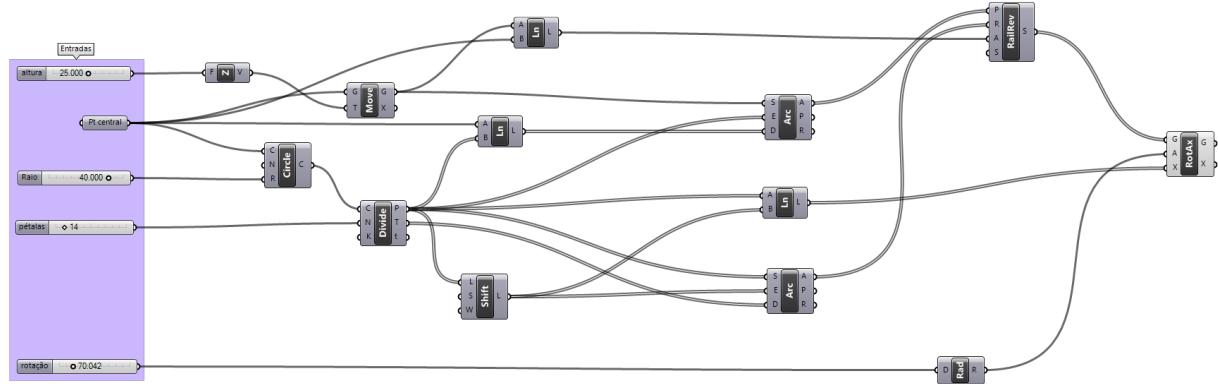


Figura B.3: Algoritmo paramétrico de uma Cúpula Móvel

Referências Bibliográficas

- ARANDA, B.; LASCH, C.; BALMOND, C. *Tooling (Pamphlet Architecture Series #27)*. Princeton Architectural Press, 2007. (Pamphlet architecture). ISBN 9781568985473. Disponível em: [⟨http://books.google.com.br/books?id=MG8xfIe55t0C⟩](http://books.google.com.br/books?id=MG8xfIe55t0C).
- BARTÁK, R. Theory and Practice of Constraint Propagation. In: *In Proceedings of the 3rd Workshop on Constraint Programming in Decision and Control (CPDC 2001)*. [S.l.: s.n.], 2001. p. 7–14. ISSN 00029629.
- BAUMGART, F. Graphical statics a forgotten tool for solving plane mechanical problems. *Injury*, v. 31, p. 24–85, may 2000. ISSN 00201383. Disponível em: [⟨http://www.sciencedirect.com/science/article/pii/S0020138300800418⟩](http://www.sciencedirect.com/science/article/pii/S0020138300800418).
- BAYLE, J. *Estática Gráfica*. Barcelona: Editorial Hispano Europea, 1966. 248 p.
- BERG, M. D. et al. *Computational Geometry: Algorithms and Applications (3rd)*. 2. ed. Springer, 2008. v. 40. 388 p. ISSN 14337851. ISBN 9783540779735. Disponível em: [⟨http://books.google.com.br/books?id=tkyG8W2163YC⟩](http://books.google.com.br/books?id=tkyG8W2163YC).
- BLOCK, P. *Thrust Network Analysis : exploring three-dimensional equilibrium*. 155 p. Tese (Doutorado) — MIT, 2009. Disponível em: [⟨http://hdl.handle.net/1721.1/49539⟩](http://hdl.handle.net/1721.1/49539).
- BLOCK, P.; CIBLAC, T.; OCHSENDORF, J. Real-time limit analysis of vaulted masonry buildings. *Computers & Structures*, v. 84, n. 29-30, p. 1841–1852, nov 2006. ISSN 00457949. Disponível em: [⟨http://linkinghub.elsevier.com/retrieve/pii/S0045794906002227⟩](http://linkinghub.elsevier.com/retrieve/pii/S0045794906002227).
- BLOCK, P.; LACHAUER, L. Three-dimensional funicular analysis of masonry vaults. *Mechanics Research Communications*, v. 56, p. 53–60, mar 2014. ISSN 00936413. Disponível em: [⟨http://linkinghub.elsevier.com/retrieve/pii/S0093641313002012⟩](http://linkinghub.elsevier.com/retrieve/pii/S0093641313002012).
- BLOCK Research Group. *eQUILIBRIUM*. Disponível em: [⟨http://block.arch.ethz.ch/equilibrium/⟩](http://block.arch.ethz.ch/equilibrium/).
- BOHNACKER, H.; GROSS, B.; LAUB, J. *Generative Design: Visualize, Program, and Create with Processing*. Princeton Architectural Press, 2012. 472 p. (Princeton Architectural Press). ISBN 9781616890773. Disponível em: [⟨http://www.amazon.com/Generative-Design-Visualize-Program-Processing/dp/1616890770/ref=pd{_\}sim{_\}sbs{_\}b{_\}1?ie=UTF8{\&}refRID=10QEF⟩](http://www.amazon.com/Generative-Design-Visualize-Program-Processing/dp/1616890770/ref=pd{_\}sim{_\}sbs{_\}b{_\}1?ie=UTF8{\&}refRID=10QEF).
- BOYER, C. B.; GOMIDE, E. F. *História da Matemática*. São Paulo: Edgard Blucher, 1974.
- BURR, W. H.; FALK, M. *The Graphic Method by Influence Lines for Bridge and Roof Computations*. New York, NY, USA: JOHN WILEY & SONS, 1908. Disponível em: [⟨https://ia600602.us.archive.org/11/items/graphicmethodbyi00burr/graphicmethodbyi00burr.pdf⟩](https://ia600602.us.archive.org/11/items/graphicmethodbyi00burr/graphicmethodbyi00burr.pdf).

- CALDAS, L. Generation of energy-efficient architecture solutions applying GENE_ARCH: An evolution-based generative design system. *Advanced Engineering Informatics*, v. 22, n. 1, p. 59–70, jan 2008. ISSN 14740346. Disponível em: [⟨http://linkinghub.elsevier.com/retrieve/pii/S1474034607000493⟩](http://linkinghub.elsevier.com/retrieve/pii/S1474034607000493).
- CALDAS, L.; NORFORD, L. Architectural Constraints in a Generative Design System: interpreting energy consumption levels. *Proc. of Building Simulation 2001*, Rio de Janeiro, p. 1397–1404, 2001. Disponível em: [⟨http://www.ibpsa.org/proceedings/bs2001/bs01{_\}1397{_\}14⟩](http://www.ibpsa.org/proceedings/bs2001/bs01{_\}1397{_\}14).
- CALDAS, L. G.; NORFORD, L. K. A multi-objective genetic algorithm for robust design optimization. In: *ACADIA '99*. Salt Lake City: Acadia'99, 1999. p. 250–261. ISBN 1595930108. ISSN 1615147X. Disponível em: [⟨http://portal.acm.org/citation.cfm?doid=1068009.1068140⟩](http://portal.acm.org/citation.cfm?doid=1068009.1068140).
- CALDAS, L. G.; NORFORD, L. K. A design optimization tool based on a genetic algorithm. *Automation in Construction*, v. 11, n. 2, p. 173–184, feb 2002. ISSN 09265805. Disponível em: [⟨http://linkinghub.elsevier.com/retrieve/pii/S0926580500000960⟩](http://linkinghub.elsevier.com/retrieve/pii/S0926580500000960).
- CECCATO, C. The Master-BUILDER-Geometer. In: CECCATO, C.; HESSELGREN, L.; PAULY, M. (Ed.). *Advances in Architectural Geometry 2010*. [S.l.]: Springer, 2010. p. 9–14.
- CHATZIKONSTANTINOU, Y. *Hoopsnake for Grasshopper*. 2014. Disponível em: [⟨http://yconst.com/software/hoopsnake/⟩](http://yconst.com/software/hoopsnake/).
- CORBUSIER, L. *Por uma arquitetura*. São Paulo: Perspectiva, 1994.
- COSTA, E. V. *Exercícios de resistência dos materiais com elementos e grafostática e de energia de deformação*. São Paulo: Editora Nacional, 1974. v. 2. 365 p.
- CRAPO, H. Structural Rigidity. *Structural Topology Vol.1*, v. 1, n. 1, p. 26–45, 1979. ISSN 8750-7587.
- CREMONA, L. *Graphical statics: two treatises on the graphical calculus and reciprocal figures in graphical statics*. Clarendon press, 1890. Disponível em: [⟨http://books.google.com.br/books?id=IvNJAAAAMAAJ⟩](http://books.google.com.br/books?id=IvNJAAAAMAAJ).
- DALVI, S. *THE SUBSEQUENT EVOLUTION OF SPACE PLANNING The power of selecting and delivering discoveries*. Tese (Doutorado) — Savannah College of Art and Design, Savannah, GA, 2011.
- DALY, H. M. T. E. *Revisiting Algorithms in Architectural Design: Towards New Computational Methods*. 261 p. Tese (Doutorado) — Ain Shams University, Cairo, 2009.
- DINO, I. G. Creative Design Exploration By Parametric Generative Systems In Architecture. *Metu Journal of the Faculty of Architecture*, v. 1, p. 207–224, jun 2012. ISSN 02585316. Disponível em: [⟨http://jfa.arch.metu.edu.tr/archive/0258-5316/2012/cilt29/sayi{_\}1/207-224.⟩](http://jfa.arch.metu.edu.tr/archive/0258-5316/2012/cilt29/sayi{_\}1/207-224.)
- DONATH, D.; LOBOS, D. Plausibility in Early Stages of Architectural Design: A New Tool for High-Rise Residential Buildings. *Tsinghua Science & Technology*, Tsinghua University Press, v. 14, n. 1, p. 327–332, jun 2009. ISSN 10070214. Disponível em: [⟨http://www.sciencedirect.com/science/article/pii/S1007021409700483⟩](http://www.sciencedirect.com/science/article/pii/S1007021409700483).

DUARTE, F. *Arquitetura e Tecnologias de Informação: da Revolução Industrial à Revolução Digital*. Annablume, 1999. (Selo universidade). ISBN 9788574190624. Disponível em: [⟨http://books.google.com.br/books?id=AgLyViiB65UC⟩](http://books.google.com.br/books?id=AgLyViiB65UC).

DUARTE, J. P. A discursive grammar for customizing mass housing: the case of Siza's houses at Malagueira. *Automation in Construction*, v. 14, n. 2, p. 265–275, mar 2005. ISSN 09265805. Disponível em: [⟨http://linkinghub.elsevier.com/retrieve/pii/S0926580504000810⟩](http://linkinghub.elsevier.com/retrieve/pii/S0926580504000810).

EDDY, H. T. *New Constructions in Graphical Statics*. New York, NY, USA: D. van Nostrand, 1877. ISBN 978114155536. Disponível em: [⟨http://books.google.com.br/books?id=kUJWRRAACAAJ⟩](http://books.google.com.br/books?id=kUJWRRAACAAJ).

EISENMAN, P.; DOBNEY, S.; Eisenman Architects. *Eisenman Architects : selected and current works*. The Images Publishing Group, 1995. 256 p. p. ISBN 1875498214\r1320-7253 ;. Disponível em: [⟨http://www.amazon.com/Eisenman-Architects-Selected-Current-Architect/dp/1875498214?SubscriptionId=0JYN1NVW651KCA56C102{&}tag=techkie-20{&}linkCode=xm2{&}camp=2025{&}creative=165953{&}creative⟩](http://www.amazon.com/Eisenman-Architects-Selected-Current-Architect/dp/1875498214?SubscriptionId=0JYN1NVW651KCA56C102{&}tag=techkie-20{&}linkCode=xm2{&}camp=2025{&}creative=165953{&}creative).

EVOLO. *Inverted Skyscraper Typology*. 2009. Disponível em: [⟨http://www.evolo.us/competition/inverted-skyscraper-typology/⟩](http://www.evolo.us/competition/inverted-skyscraper-typology/).

FASOULAKI, E. Genetic Algorithms in Architecture : a Necessity or a Trend ? *Design*, 2007. Disponível em: [⟨http://www.generativedesign.it/on/cic/papersGA2007/09.pdf⟩](http://www.generativedesign.it/on/cic/papersGA2007/09.pdf).

FASOULAKI, E. *Integrated Design: A Generative Multi-Performative Design Approach*. 72 p. Tese (Doutorado) — MIT, Cambrige, 2008.

FLORIO, W. Modelagem Paramétrica, Criatividade E Projeto: Duas Experiências Com Estudantes De Arquitetura. *Gestão & Tecnologia de Projetos*, v. 6, n. 2, p. 43–66, 2012. ISSN 1981-1543. Disponível em: [⟨http://www.revistas.usp.br/gestaodeprojetos/article/view/51010⟩](http://www.revistas.usp.br/gestaodeprojetos/article/view/51010).

FOUSKAKIS, D.; DRAPER, D. Stochastic Optimization: A Review. *International Statistical Review / Revue Internationale de Statistique*, v. 70, n. 3, p. 315, dec 2002. ISSN 03067734. Disponível em: [⟨http://www.jstor.org/stable/10.2307/1403861?origin=crossref⟩](http://www.jstor.org/stable/10.2307/1403861?origin=crossref).

FRAMPTON, K. *História crítica da arquitetura moderna*. 1a. ed. São Paulo: Martins Fontes, 2008. 470 p. ISBN 9788533624269 8533624263.

GERHARDT, R.; KURRER, K. E.; PICHLER, G. The methods of graphical statics and their relation to the structural form. In: HUERTA, S. (Ed.). *Proceedings of the First International Congress on Construction History*. Madrid: Instituto Juan de Herrera, SEHC, COAC, CAATC, 2003.

GONDENHÓRN, S. *Elementos de Estatica Grafica*. Buenos Aires: Talleres Gráficos Argentinos L.J. Rosso, 1943. 320 p.

GREENWOLD, S. *ActiveStatics*. Disponível em: [⟨http://acg.media.mit.edu/people/simong/statics/Start.html⟩](http://acg.media.mit.edu/people/simong/statics/Start.html).

HAMMER, J. From Fractal Geometry to Fractured Architecture : The Federation Square of Melbourne. *The Mathematical Intelligencer*, v. 28, n. 4, p. 44–48, 2006.

HARDING, J.; SHEPHERD, P. Structural Form Finding using Zero-Length Springs with Dynamic Mass. In: *2011 IASS Annual Symposium: IABSE-IASS 2011: Taller, Longer, Lighter*. London: [s.n.], 2011. p. International Association for Shell and Spatial St.

Heneghan Peng Architects. *Heneghan Peng Architects - Website*. 2014. Disponível em: [⟨www.hparc.com⟩](http://www.hparc.com).

HENKEL, O. *Estática Gráfica*. 4^ª edição. ed. Barcelona: Editorial Labor S.A., 1953. 402 p.

HERR, C. M.; KVAN, T. Adapting cellular automata to support the architectural design process. *Automation in Construction*, v. 16, n. 1, p. 61–69, jan 2007. ISSN 09265805. Disponível em: [⟨http://linkinghub.elsevier.com/retrieve/pii/S0926580505001494⟩](http://linkinghub.elsevier.com/retrieve/pii/S0926580505001494).

HOFFMANN, C. M.; JOAN-ARINYO, R. A brief on constraint solving. *Computer-Aided Design and Applications*, v. 2, n. 5, p. 655–663, 2005. ISSN 16864360.

HOSKINS, L. *Elements of Graphic Statics*. London: The Macmillan Company, 1899.

Iwamoto-Scott Architecture. *Iwamoto-Scott Architecture - Website*. Disponível em: [⟨http://www.iwamotoscott.com/⟩](http://www.iwamotoscott.com/).

JOHNSON, L. J. *Statics by Algebraic and Graphic Methods: Intended Primarily for Students of Engineering and Architecture*. J. Wiley & sons, 1908. Disponível em: [⟨http://books.google.com.br/books?id=jlRDAAAAIAAJ⟩](http://books.google.com.br/books?id=jlRDAAAAIAAJ).

KILIAN, A. *CADenary*. Disponível em: [⟨http://www.designexplorer.net/newscreens/cadenarytool/cadenarytool.html⟩](http://www.designexplorer.net/newscreens/cadenarytool/cadenarytool.html).

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by Simulated Annealing. *Science*, v. 220, n. 4598, p. pp. 671–680, 1983. ISSN 00368075. Disponível em: [⟨http://www.jstor.org/stable/1690046⟩](http://www.jstor.org/stable/1690046).

KNEIP karl; WINZER, G.; KILLMANN, P. *Mecânica Resistência de Materiais Grafostática*. 2^ª. ed. Buenos Aires: Editorial Labor S.A., 1939. 530 p.

KNUTH, D. *Fundamental Algorithms. The Art of Computer Programming*. 3rd. ed. Massachusetts: Pearson Education, 1997. v. 1. 435–455 p. ISBN 9780321635747. Disponível em: [⟨http://books.google.com.br/books?id=x9AsAwAAQBAJ⟩](http://books.google.com.br/books?id=x9AsAwAAQBAJ).

KOLAREVIC, B. e. *Architecture in the Digital Age: Design and Manufacturing*. London: Spon press, 2003. 320 p. ISSN 15905896. ISBN 9780415278201. Disponível em: [⟨http://www.routledge.com/books/details/9780415278201/\\$\delimitet" id="H3LRfJJQlkAC{&} printsec=frontcover{&} hl=pt-PT{&} source=gbs{_}ViewAPI{&} redir{_}e⟩](http://www.routledge.com/books/details/9780415278201/$\delimitet)

KRISH, S. A practical generative design method. *Computer-Aided Design*, Elsevier Ltd, v. 43, n. 1, p. 88–100, jan 2011. ISSN 00104485. Disponível em: [⟨http://linkinghub.elsevier.com/retrieve/pii/S0010448510001764⟩](http://linkinghub.elsevier.com/retrieve/pii/S0010448510001764).

LACHAUER, L.; JUNGJOHANN, H.; KOTNIK, T. Interactive Parametric Tools for Structural Design. In: *Proc IABSE-IASS*. London, UK: [s.n.], 2011. Disponível em: [⟨http://www.schwartz.arch.ethz.ch/Publikationen/Dokumente/InteractiveTools.pdf⟩](http://www.schwartz.arch.ethz.ch/Publikationen/Dokumente/InteractiveTools.pdf).

LACHAUER, L.; KOTNIK, T. Geometry of Structural Form. In: CECCATO, C.; OTHERS (Ed.). *Advances in Architectural Geometry 2010*. [S.l.]: Springer, 2010. p. 193–203.

LAMBERTI, L. An efficient simulated annealing algorithm for design optimization of truss structures. *Computers & Structures*, v. 86, n. 19-20, p. 1936–1953, oct 2008. ISSN 00457949. Disponível em: [⟨http://linkinghub.elsevier.com/retrieve/pii/S0045794908000448⟩](http://linkinghub.elsevier.com/retrieve/pii/S0045794908000448).

LODDING, K. N. The Hitchhiker's Guide to Biomimetic Software. *Queue*, ACM, New York, NY, USA, v. 2, n. 4, p. 66, 2004. ISSN 15427730. Disponível em: [⟨http://doi.acm.org/10.1145/1016978.1016985⟩](http://doi.acm.org/10.1145/1016978.1016985).

LUH, G.-C.; LIN, C.-Y.; LIN, Y.-S. A binary particle swarm optimization for continuum structural topology optimization. *Applied Soft Computing*, Elsevier B.V., v. 11, n. 2, p. 2833–2844, mar 2011. ISSN 15684946. Disponível em: [⟨http://dx.doi.org/10.1016/j.asoc.2010.11.013⟩](http://dx.doi.org/10.1016/j.asoc.2010.11.013).

MALCOLM, C. W. *A Text Book on Graphic Statics*. 3. ed. New York, NY, USA: THE MYRON C. CLARK PUBLISHING CO., 1909. ISBN 9781293168318. Disponível em: [⟨http://books.google.com.br/books?id=F\\$\\delimiter"026E30F\\${__}-ZTP7wR⟩](http://books.google.com.br/books?id=F$\\delimiter).

MANDELBROT, B. B. *The fractal geometry of nature*. 1982. Henry Holt and Company, 1982. (Einaudi paperbacks). ISBN 9780716711865. Disponível em: [⟨http://books.google.co.uk/books?id=0R2LkE3N7-oC⟩](http://books.google.co.uk/books?id=0R2LkE3N7-oC).

MASUM, H. *Review of a new kind of science*. [s.n.], 2002. v. 33. 15 p. ISSN 01635700. ISBN 1579550088. Disponível em: [⟨http://austms.org.au/Jobs/Reviews2.html⟩](http://austms.org.au/Jobs/Reviews2.html).

MAXWELL, J. C. *Maxwell Scientific Papers*. [S.l.]: Dover Publications, 1965. 607 p.

MCCARTHY, J. Recursive functions symbolic expressions and their computation by machine, Part I. *Communications of the ACM*, v. 3, n. 4, p. 184–195, 1960. ISSN 00010782. Disponível em: [⟨http://dl.acm.org/citation.cfm?id=367199⟩](http://dl.acm.org/citation.cfm?id=367199).

MCCROMAC, J. C. *Análise Estrutural usando métodos Clássicos e Métodos Matriciais*. 4^ª ed. Rio de Janeiro: LTC, 2009. 500 p. ISBN 978-85-216-1686-3.

MILENA, S.; OGNEN, M. Application of Generative Algorithms in Architectural Design. In: *Advances in Mathematical and Computational Methods*. [s.n.], 2010. p. 175–180. ISBN 9604742434. Disponível em: [⟨http://www.wseas.us/e-library/conferences/2010/Faro/MACMESE/MACMESE-27.pdf⟩](http://www.wseas.us/e-library/conferences/2010/Faro/MACMESE/MACMESE-27.pdf).

MITCHELL, W. J. *World's Greatest Architect MAKING, MEANING, AND NETWORK CULTURE*. Cambrige: MIT Press, 2008. v. 40. 147 p. ISBN 9780262265409. Disponível em: [⟨http://books.google.com.br/books?id=DxszH9whFSIC⟩](http://books.google.com.br/books?id=DxszH9whFSIC).

MÜLLER, P. et al. Procedural modeling of buildings. *ACM Transactions on Graphics*, v. 25, n. 3, p. 614, 2006. ISSN 07300301. Disponível em: [⟨http://portal.acm.org/citation.cfm?doid=1141911.1141931⟩](http://portal.acm.org/citation.cfm?doid=1141911.1141931).

NASH, W. A.; SILVA, J. F. *Resistência de materiais*. São Paulo: McGraw-Hill, 1974. (Schaum).

- OKABE, A.; BOOTS, B.; SUGIHARA, K. *Spatial Tesselation: Concepts and Applications of Voronoi Diagrams*. 2nd ed.. ed. [S.l.]: John Wiley and Sons, Inc., 1992. (Series in Probability and Statistics).
- OXMAN, R. Theory and design in the first digital age. *Elsevier*, v. 27, n. 3, p. 229–265, may 2006. ISSN 0142694X. Disponível em: [⟨http://linkinghub.elsevier.com/retrieve/pii/S0142694X05000840⟩](http://linkinghub.elsevier.com/retrieve/pii/S0142694X05000840).
- PANSERI, E. *Curso medio de estática gráfica*. 4. ed. Buenos Aires: Construcciones Sudamericanas, 1968. 445 p.
- PEREIRA, H. B. D. B.; ZEBENDE, G. F.; MORET, M. a. Learning computer programming: Implementing a fractal in a Turing Machine. *Computers and Education*, v. 55, n. 2, p. 767–776, sep 2010. ISSN 03601315. Disponível em: [⟨http://dx.doi.org/10.1016/j.compedu.2010.03.009⟩](http://dx.doi.org/10.1016/j.compedu.2010.03.009).
- PICON, A. Ornament and its users: from the Vitruvian tradition to the digital age. *Le Visiteur*, n. 17, p. 176–180, 2011. Disponível em: [⟨http://dash.harvard.edu/handle/1/12638041⟩](http://dash.harvard.edu/handle/1/12638041).
- PORTOGHESI, P. *Depois da Arquitetura Moderna*. São Paulo: Martins Fontes, 2002. (Coleção A). ISBN 9788533615762. Disponível em: [⟨http://books.google.com.br/books?id=QBiWPgAACAAJ⟩](http://books.google.com.br/books?id=QBiWPgAACAAJ).
- PRUSINKIEWICZ, P.; LINDENMAYER, A. *The algorithmic beauty of plants*. Springer-Verlag, 1997. v. 122. 109–110 p. (The Virtual Laboratory, 1). ISSN 01689452. ISBN 0387972978. Disponível em: [⟨http://books.google.com.br/books?id=QWKjQgAACAAJ⟩](http://books.google.com.br/books?id=QWKjQgAACAAJ).
- PWT Architects. *PWT Architects - Website*. Disponível em: [⟨http://www.ptw.com.au/⟩](http://www.ptw.com.au/).
- SANJUAN, R. I. *Elementos de Grafostática Integración Gráfica y sus Aplicaciones a la Construcción*. Barcelona: Editorial Labor S.A., 1951.
- SANTOS, L. F. B. S. D. *Sistemas Generativos de Projecto : Integração de Ferramentas Digitais no Projecto de Arquitectura Aplicação ao Caso de Estudo : a Casa Pátio da Medina de Marrakech*. 155 p. Tese (Doutorado), 2009.
- SHEA, K.; AISH, R.; GOURTOVAIA, M. Towards integrated performance-driven generative design tools. *Automation in Construction*, v. 14, n. 2, p. 253–264, mar 2005. ISSN 09265805. Disponível em: [⟨http://linkinghub.elsevier.com/retrieve/pii/S0926580504000809⟩](http://linkinghub.elsevier.com/retrieve/pii/S0926580504000809).
- SHEARER, M. S. *Analyzing and Creating Forms : Rapid Generation of Graphic Statics Solutions through RhinoScript by*. Tese (Doutorado), 2010.
- SINGH, V.; GU, N. Towards an integrated generative design framework. *Design Studies*, Elsevier Ltd, v. 33, n. 2, p. 185–207, mar 2012. ISSN 0142694X. Disponível em: [⟨http://dx.doi.org/10.1016/j.destud.2011.06.001⟩](http://dx.doi.org/10.1016/j.destud.2011.06.001).
- SONMEZ, F. O. Shape optimization of 2D structures using simulated annealing. *Computer Methods in Applied Mechanics and Engineering*, v. 196, n. 35-36, p. 3279–3299, jul 2007. ISSN 00457825. Disponível em: [⟨http://linkinghub.elsevier.com/retrieve/pii/S0045782507000722⟩](http://linkinghub.elsevier.com/retrieve/pii/S0045782507000722).

STEELE, G. L.; GABRIEL, R. P. The Evolution of Lisp. In: Bergin Jr., T. J.; Gibson Jr., R. G. (Ed.). *ACM SIGPLAN Notices*. New York, NY, USA: ACM, 1993. v. 28, n. 3, cap. The Evolut, p. 231–270. ISBN 0-89791-570-4. Disponível em: <<http://doi.acm.org/10.1145/234286.1057818>>.

STINY, G. *Shape: talking about seeing and doing*. Cambridge: Mit Press, 2008. 432 p. ISBN 978-0262693677.

STINY, G.; GIPS, J. Shape grammars and the generative specification of painting and sculpture. In: PETROCELLI (Ed.). *Information Processing 71 Proceedings of the IFIP Congress 1971. Volume 2*. Auerbach: [s.n.], 1972. v. 71, p. 1460–1465. ISBN 0 7204 2063 6. Disponível em: <GotoISI://INSPEC:466862>.

STINY, G.; MITCHELL, W. J. The Palladian grammar. *Environment and Planning B*, Pion Ltd, v. 5, n. 1, p. 5–18, 1978. ISSN 0265-8135. Disponível em: <<http://www.envplan.com/abstract.cgi?id=b050005>>.

STROBBE, T. et al. Metaheuristics in architecture. In: Van Wittenberghe, J. (Ed.). *Sustainable Construction and Design*. [S.l.]: Ghent University, Laboratory Soete, 2011. v. 2, p. 190–196. ISBN 9789490726010. ISSN 2032-7471.

TEPAVCEVIC, B.; STOJAKOVIC, V. Shape grammar in contemporary architectural theory and design. *Facta universitatis - series: Architecture and Civil Engineering*, v. 10, n. 2, p. 169–178, 2012. ISSN 0354-4605. Disponível em: <<http://www.doiserbia.nb.rs/Article.aspx?ID=0354-46051202169T>>.

TERZIDIS, K. *Algorithmic architecture*. [S.l.]: Taylor & Francis, 2006. v. 1. 159 p. ISBN 9780750667258.

TESSMANN, O. *Collaborative Design Procedures for Architects and Engineers*. 1–176 p. Tese (Doutorado), 2008.

The python Foundation. *Python.org*. 2014. Disponível em: <<https://www.python.org>>.

TIBBITS, S.; HARTEN, A. van der; BAER, S. *Python 101 for Rhinoceros*. Robert McNeel and Associates, 2011. Disponível em: <<http://wiki.mcneel.com/developer/python>>.

VENTURI, R. Book; Book/Illustrated. *Complexity and contradiction in architecture*. 2nd ed. ed. [S.l.]: London : Architectural Press, 1977. 132 p. ISBN 0870702815.

VETTORETTI, A. C. *Bancos para ler e conversar: parâmetros de projeto para sistemas de design generativo*. 226 p. Tese (Doutorado) — UFRGS, Porto Alegre, 2010. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/31395/000780166.pdf?sequence=1>>.

YAHYA, M.; SAKA, M. Construction site layout planning using multi-objective artificial bee colony algorithm with Levy flights. *Automation in Construction*, v. 38, p. 14–29, mar 2014. ISSN 09265805. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0926580513001945>>.

ZALEWSKI, W.; ALLEN, E. *Shaping structures: statics*. Wiley, 1998. ISBN 9780471283386. Disponível em: <<http://books.google.com.br/books?id=x-BRAAAAMA AJ>>.

ZANNI, G.; PENNOCK, G. R. A unified graphical approach to the static analysis of axially loaded structures. *Mechanism and Machine Theory*, Elsevier Ltd, v. 44, n. 12, p. 2187–2203, dec 2009. ISSN 0094114X. Disponível em: <<http://dx.doi.org/10.1016/j.mechmachtheory.2009.07.002>>.

ZHANG, H. et al. Particle swarm optimization-based schemes for resource-constrained project scheduling. *Automation in Construction*, v. 14, n. 3, p. 393–404, jun 2005. ISSN 09265805. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0926580504001128>>.

*Sistema Generativo de Projeto Aplicado ao Desenho e Otimização da Estrutura de um
Shed*

Fernando Ferraz Ribeiro

Salvador, Abril de 2015.