

**OTIMIZAÇÃO MULTIOBJETIVO APLICADA À ANÁLISE ESTRUTURAL
DE EDIFÍCIOS**

JONATHAS IOHANATHAN FELIPE DE OLIVEIRA

**DISSERTAÇÃO DE MESTRADO
EM ESTRUTURAS E CONSTRUÇÃO CIVIL**

**FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA**

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL

**OTIMIZAÇÃO MULTIOBJETIVO APLICADA À ANÁLISE ESTRUTURAL DE
EDIFÍCIOS**

JONATHAS IOHANATHAN FELIPE DE OLIVEIRA

ORIENTADOR: ANTONIO CARLOS DE OLIVEIRA MIRANDA

DISSERTAÇÃO DE MESTRADO EM ESTRUTURAS E CONSTRUÇÃO CIVIL

PUBLICAÇÃO: E.DM – 25A/17

BRASÍLIA/DF: AGO/2017

**OTIMIZAÇÃO MULTIOBJETIVO APLICADA À ANÁLISE ESTRUTURAL DE
EDIFÍCIOS**

JONATHAS IOHANATHAN FELIPE DE OLIVEIRA

DISSERTAÇÃO DE Mestrado submetida ao Departamento de Engenharia Civil e Ambiental da Faculdade de Tecnologia da Universidade de Brasília como parte dos requisitos necessários para a obtenção do grau de Mestre em Estruturas e Construção Civil.

APROVADA POR:

Prof. Antonio Carlos de Oliveira Miranda, Dr. (UnB)
(Orientador)

Prof. Francisco Evangelista Júnior, Ph.D. (UnB)
(Examinador Interno)

Prof. Rafael Holdorf Lopez, Ph.D. (UFSC)
(Examinador Externo)

Brasília/DF, 28 de agosto de 2017

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

00L48o Oliveira, Jonathas Iohanathan Felipe de
Otimização Multiobjetivo Aplicada À Análise Estrutural De Edifícios/
Jonathas Iohanathan Felipe de Oliveira; orientador Antonio Carlos de Oliveira
Miranda. -- Brasília, 2017.
76 p.

Dissertação (Mestrado - Mestrado em Estruturas e Construção Civil) --
Universidade de Brasília, 2017.

1. Estruturas. 2. Otimização multiobjetivo. 3.
posicionamento de pilares. I. Miranda, Antonio Carlos de Oliveira, orient. II.
Título.

REFERÊNCIA BIBLIOGRÁFICA

OLIVEIRA, J. I. F. (2017). Otimização multiobjetivo aplicada à análise estrutural de edifícios. Dissertação de Mestrado em Estruturas e Construção Civil, Publicação E.DM-25A/17, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, DF, 76p.

CESSÃO DE DIREITOS

AUTOR: Jonathas Iohanathan Felipe de Oliveira.

TÍTULO: otimização multiobjetivo aplicada à análise estrutural de edifícios.

GRAU: Mestre ANO: 2017

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

Jonathas Iohanathan Felipe de Oliveira
Rua Eufrásio de Oliveira, 976, Bairro Alto da Conceição
59600-410 Mossoró-RN-Brasil
jonathasiohanathan@hotmail.com

AGRADECIMENTOS

Estruturas são sistemas projetados para receber, resistir e transmitir as solicitações que lhe são impostas. Esta obra, em sua totalidade e abrangência, é dedicada àqueles que formam o sistema estrutural do qual faço parte. Os meus maiores exemplos e professores, em aspectos que transcendem a ciência: minha família.

Ionete Maressa, minha querida *Netinha*, *Netson*. Aquela a quem a vida me deu a honra de ser filho, de estar ao seu lado, de ter os melhores abraços, os mais sábios conselhos, as discussões mais nobres; minha definição de resiliência, amor e perseverança. Obrigado por me apoiar em toda jornada até aqui. Me desculpe por minha *ausência*, a saudade que causei e as preocupações que te dei. Acredite: você provocou em mim muito mais do que lhe causei. Meu amor é seu.

Evaristo Filho, meu gordinho favorito. O melhor amigo que tive. Meu companheiro em sonhos, delírios, aspirações e paixões. Uma mente à frente de sua geração: o cara mais moderno e visionário que pude conhecer. Um líder nato, um carisma irrefutável. O responsável por me fazer acreditar que *bons dias estão por vir*. Quem me ensinou a sonhar e, mais tarde, quando tudo parecer perdido, sonhar de novo, e novamente. Minha maior saudade. O baixista da nossa banda. Deus nunca nos esqueceu e não irá. A vida ainda será boa conosco, pode acreditar. Você me fez acreditar. Meu pai. Meu abraço mais forte é para você. *Esse lugar em mim ainda é seu*.

Dedico também aos meus incomparáveis, indescritíveis, inigualáveis, inseparáveis, minhas flores mais lindas: meus irmãos e melhores amigos, Nikolas e Luckas Iohanathan. A vida sem vocês seria um tédio, uma obra prima incompreendida, um filme mudo e sem cores, uma música em braile. Meus queridos e fofos, a obra mais perfeita de meus pais: vocês não sabem a falta que me fizeram. Estendo os sentimentos à Isadora e Erica. Somos nós, para sempre.

Se há motivos na realização deste trabalho, minha família é o único. Nilzete, *Nilzетinha*, minha guerreira e amor incomparável é boa parte disso. Minha avó, fonte de carinho e amor inesgotável. Não há obstáculos para seu amor e cuidado. Se nos dias mais difíceis Deus não nos deixou faltar nada, foi através de você que ele o fez. Minha vida eu dedico a ti. Desculpa por toda a saudade que causei e por não estar por perto para dividirmos os dias, para tomar seu café ou comermos uma boa melancia. Aquele cheiro no cangote!

Jonilze, Janice, minha irmã mais velha disfarçada de tia. Eu não tenho palavras para você. Só posso lhe agradecer por ter cuidado de mim e vivido todos esses dias ao nosso lado. Por ser o

porto seguro da família. Por ser inabalável, incrivelmente bem-humorada. A vida não está à sua altura. Você me dá os melhores sorrisos. Obrigado por deixar a vida mais leve e doce. Espero que ainda haja tempo suficiente para te proporcionar tudo que eu desejo e, aproveito para estender tudo isso ao meu querido *Big Tulio*. Amo vocês mais do que o Mufaza ao Simba! Meu timão, eu sou seu pumba! <3

Vitor, *Vitarugo*, *Vitarela*, meu primo igualmente disfarçado de irmão mais novo. Meu maior desafio, meu grande orgulho. Conte comigo sem hesitar em tudo que precisar. Você alçará voos maiores do que o meu, e comeremos muito para comemorar!

Dona Marly, minha querida avó. O amor sincero de um coração *iceberg*. Obrigado por seus conselhos lúcidos e precisos. Por sua sinceridade ímpar, e força invejável. Exemplo de mulher forte e guerreira, que jamais abaixou a cabeça e me inspirou a ser grande. Meu amor e admiração por você são indescritíveis.

Meu grande e verdadeiro amigo, Rodrigo Fernandes. Companheiro da vida. A única presença garantida em qualquer situação. Fizemos as músicas mais ridículas, jogamos as melhores partidas, dividimos as melhores azias e fizemos as mais particulares ironias. Estaremos juntos sempre. Conte sempre comigo.

Eric Mateus, um grande amigo de vida e profissão. José de Alencar da engenharia. Saiba que muito do que eu sou como profissional foi você que provocou, me instigou e me ajudou a ser. Agradeço pela troca de ideias sobre o trabalho, o qual compartilhamos experiências, sucessos e angústias. Crescemos e continuaremos a aumentar o *arcabouço* de saberes juntos. Por mais que você queira, você não é o motivo do nome: 30/10. *Não obstante*, agradeço seu suporte e da ~~chata insuportável~~ querida Rafaelle, sempre presente.

Agradeço aos meus mestres acadêmicos, a cada um que contribuiu de alguma forma ao meu sucesso. Gostaria de destacar o professor Antônio Miranda por me aceitar como orientando, suportar minha insistência no tema, pela dedicação como orientador e pela admirável disponibilidade e acessibilidade. Acima de tudo, agradeço pelo apoio sempre dado, paciência e confiança. Este trabalho existe por causa do senhor. Aproveito para agradecer ao professor Raul Durand, por quem agradeço a todo o corpo docente do Programa de Pós-Graduação em Estruturas e Construção Civil da Universidade de Brasília pela experiência compartilhada e oportunidades proporcionadas. Agradeço através do professor Raimundo Amorim a todos os

professores da Universidade Federal Rural do Semi-Árido de Mossoró-RN, a quem agradeço também pelos conselhos e amizade.

Agradeço a CAPES e ao MEC pelo programa Ciência sem Fronteiras e ao crescimento por esse propiciado. Em especial, agradeço à professora Caitlin Mueller do Instituto de Tecnologia de Massachusetts pela oportunidade dada e confiança depositada em mim, além disso, principal inspiração desse trabalho. Agradeço também ao professor Rafael Holdorf Lopez, da Universidade Federal de Santa Catarina, cujo contato se deu motivado por essa experiência. A você eu agradeço a loucura de confiar em mim, as oportunidades de aprendizado e parceria que me deste e tem dado.

Dedico este trabalho também aos meus companheiros da Universidade de Brasília: Jerfson Moura (tchow, Brasil!), Iarly Vanderlei (cozinha), Augusto Pippi e Sara Brandão por quem agradeço aos contemporâneos do 2016.2 e toda a nossa turma; Eduardo Fontes e Natali Sarasty, por quem agradeço a todos os colegas de sala; o (Rodolfo) Palhares e Thiago Arnaud, por quem agradeço aos colegas da turma 2017.1 do mestrado e as pizzas; a Danilo, Laís e Luiza por quem agradeço aos demais colegas do PECC.

Por último e mais importante, eu agradeço a Deus por seu amor, cuidado e, especialmente, por sua obra prima: Gisla Freitas, meu anjo, *my pomps*. A pessoa mais doce que existiu e irá existir. O ser humano que eu queria ser. Seu amor indelével, seu cuidado omnipresente, me movem onde quer que eu esteja. Desculpa por me ausentar, pela saudade que causei, pelos sorrisos que não compartilhamos, pelos filmes que não assistimos, pelas viagens que ainda não fizemos e por demorar 18 anos até te encontrar: *Quando a alma quer, ela espera*. Gisla fez possível a conclusão deste trabalho. Gisla é a minha inspiração. Gisla me deixa em dúvidas se eu vivo ou sonho. Seu amor é a *quintessência* da vida. Motivo por querer abandonar tudo e ir criar ovelhas na Islândia ao seu lado ou *um fim de tarde em qualquer lugar!* A vida para nós está apenas começando. *Pra quê temer o futuro quando eu sei que você vai estar lá?* Para você eu quero dar os meus dias. Em seu nome agradeço também a toda a sua família, minha segunda casa.

Faça alguém feliz, não lembrar de você ao sorrir. Gram.

Eu quero um sonho que não acabe com o nascer do dia
Que não ruine com as chuvas desse inverno
Pois eu tenho o mesmo sonho toda noite, todo dia.

RESUMO

OTIMIZAÇÃO ESTRUTURAL COM USO DE ALGORITMO GENÉTICO MULTIOBJETIVO

Autor: Jonathas Iohanathan Felipe de Oliveira
Orientador: Antônio Carlos de Oliveira Miranda
Programa de Pós-graduação em Estruturas e Construção Civil
Brasília, agosto de 2017

A concepção estrutural é um processo em que o projetista deve decidir sobre a topologia de um sistema estrutural, materiais a serem utilizados, disposição dos elementos e suas dimensões. Em sua essência, este é um processo iterativo. Começa com uma escolha feita pelo projetista, com base em sua experiência e recomendações técnicas, e é continuamente alterado até que alguns critérios do projeto sejam alcançados, como os deslocamentos máximos permitidos e a capacidade de resistência. Em estruturas de concreto armado aporticadas, o posicionamento dos pilares é uma das escolhas que o projetista deve fazer. Este trabalho aplica técnicas de otimização, como algoritmos genéticos, para encontrar o melhor posicionamento de pilares em pórticos de concreto armado e fornecer alternativas de projeto. Ao alterar as dimensões dos elementos e as posições dos pilares, um algoritmo genético multiobjetivo é usado para minimizar o custo com concreto, reforço de aço e formas, e deslocamento máximo. Este trabalho difere dos outros na literatura no modo como os pilares são posicionados e sua aplicação em estruturas aporticadas de topologia não-regulares.

Palavras-chave: Otimização multiobjetivo, Estruturas, posicionamento de pilares.

ABSTRACT

STRUCTURAL OPTIMIZATION WITH USE OF MULTIOBJECTIVE GENETIC ALGORITHM

Author: Jonathas Iohanathan Felipe de Oliveira

Advisor: Antônio Carlos de Oliveira Miranda

Programa de Pós-Graduação em Estruturas e Construção Civil

Brasília, August 2017

Structural design is a process in which the designer must decide on the topology of a structural system, materials to be used, arrangement of elements and their dimensions. In essence, this is an iterative process. It starts with a choice made by the designer, based on his experience and technical recommendations, and is continually altered until certain design criteria are met, such as maximum permissible displacements and strength. In reinforced concrete structures, the column's positioning is one of the choices that the designer must make. The present dissertation applies optimization techniques, such a multiobjective genetic algorithm called NSGA-II, to find the best column layout in reinforced concrete frames and provide design alternatives. By changing element dimensions and column positions, a multiobjective genetic algorithm is used to minimize cost with concrete, steel reinforcement and formwork, maximum displacement, maximum absolute bending moment and maximum positive-negative extremes bending moment. This work differs from the others in the literature on the way columns are positioned and their application in non-regular topological framed structures. Finally, some example of applications are provided to test the proposed methodology.

Keywords: Multiobjective optimization, Structures, positioning of columns.

SUMÁRIO

1.	INTRODUÇÃO.....	1
1.1.	OTIMIZAÇÃO APLICADA EM PROJETOS DE ESTRUTURAS.....	2
1.2.	PROPOSTA DO TRABALHO.....	3
1.3.	OBJETIVOS.....	3
1.3.1.	Objetivos específicos.....	4
1.4.	ORGANIZAÇÃO DO TRABALHO.....	4
2.	ESTADO DA ARTE.....	6
2.1.	PROBLEMAS DE OTIMIZAÇÃO.....	6
2.1.1.	A formulação de um problema de otimização.....	7
2.1.2.	Funções objetivo.....	7
2.1.3.	Variáveis de decisão.....	8
2.1.4.	Restrições.....	8
2.2.	APLICAÇÕES DE TÉCNICAS DE OTIMIZAÇÃO.....	9
2.3.	OTIMIZAÇÃO DE ESTRUTURAS.....	10
2.4.	OTIMIZAÇÃO DE ESTRUTURAS DE CONCRETO ARMADO.....	14
2.5.	COMENTÁRIOS GERAIS.....	18
3.	METODOLOGIA.....	20
3.1.	FORMULAÇÃO DO PROBLEMA DE OTIMIZAÇÃO.....	21
3.1.1.	Considerações sobre o modelo estrutural.....	21
3.1.2.	Dimensões das seções transversais.....	22
3.1.3.	Posicionamento dos pilares.....	23
3.1.4.	Vetor de projeto.....	26
3.1.5.	Funções objetivo.....	27
3.1.6.	Restrições.....	29
3.2.	ANÁLISE ESTRUTURAL.....	30
3.2.1.	Robot API.....	32
3.3.	OTIMIZAÇÃO EVOLUTIVA MULTI OBJETIVO: NSGA-II.....	33
3.4.	INTEROPERABILIDADE ROBOT - NSGA-II.....	35
4.	IMPLEMENTAÇÃO.....	38
4.1.	MÓDULO ROBOT.....	39

4.1.1.	Utilizando a <i>RobotAPI</i>	40
4.1.2.	Conexão com o <i>Robot</i>	40
4.2.	MÓDULO <i>NSGA-II</i>	48
4.3.	MÓDULO <i>Otimizacao</i>	48
4.4.	MÓDULO <i>Modelo</i>	49
4.5.	MÓDULO <i>PainelControle</i>	49
4.6.	MÓDULO <i>Flx</i>	52
5.	RESULTADOS	53
5.1.	EXEMPLO 01	53
5.2.	EXEMPLO 02	56
5.3.	EXEMPLO 03	57
5.4.	EXEMPLO 04	59
5.5.	EXEMPLO 05	60
5.6.	EXEMPLO 06	63
6.	CONCLUSÕES	65
6.1.	SUGESTÕES PARA TRABALHOS FUTUROS	67
	REFERÊNCIAS	69

LISTA DE FIGURAS

Figura 3.1- Influência do posicionamento dos apoios nos momentos fletores.	20
Figura 3.2- Exemplo da predefinição das vigas de um pavimento.	21
Figura 3.3 – Esquema dos estágios do posicionamento dos pilares.....	24
Figura 3.4 - Parâmetro de um pilar em uma viga do pavimento.....	26
Figura 3.5 - Representação esquemática do vetor de projeto.....	27
Figura 3.6 - Exemplo de uma Fronteira de Pareto.	29
Figura 3.7 - Interface do programa <i>Autodesk Robot Structural Analysis 2017</i>	30
Figura 3.8 - Propriedades dos elementos das vigas.....	31
Figura 3.9 - Propriedade dos elementos dos pilares.....	32
Figura 3.10 - Pseudocódigo do algoritmo NSGA-II. Fonte: Adaptado de Syberfeldt (2014). 34	
Figura 3.11 - Exemplo da avaliação do hipervolume para dois conjuntos de soluções.	35
Figura 3.12- Módulos do programa em desenvolvimento.	36
Figura 3.13 - Interação entre os módulos.....	37
Figura 4.1 - Detalhe da instalação da API do <i>Robot</i>	40
Figura 4.2 - Exemplo de acesso aos objetos principais da <i>RobotAPI</i>	41
Figura 4.3 - Exemplo de acesso as barras e nós de um modelo.	41
Figura 4.4 - Exemplo de criação de nós e barras.	42
Figura 4.5 - Exemplo de criação e aplicação de seções e apoios.	42
Figura 4.6 - Exemplo de invocação do método de execução da análise.	43
Figura 4.7 - Exemplo da obtenção de deslocamentos e momentos nos elementos.....	43
Figura 4.8 - Exemplo de configuração dos elementos para o dimensionamento.....	45
Figura 4.9 - Exemplo da configuração do motor de cálculo de armadura necessária.....	46
Figura 4.10 - Exemplo da obtenção dos valores de armadura requeridas.....	46
Figura 4.11 - Exemplo do levantamento de quantitativos.....	47
Figura 4.12 - Tela inicial do Painel de Controle na opção das vigas.	50

Figura 4.13 - Tela do Painel de Controle nas opções dos pilares.....	51
Figura 4.14 - Tela do Painel de Controle na aba <i>Otimização</i>	51
Figura 5.1 - Forma do pavimento do edifício do exemplo 01.....	53
Figura 5.2 - Soluções do exemplo 01.....	55
Figura 5.3 - Evolução do hipervolume ao longo de 100 gerações.	56
Figura 5.4 - Soluções do exemplo 02.....	57
Figura 5.5 - Soluções do exemplo 03.....	58
Figura 5.6 - Resultados para o Exemplo 04.	59
Figura 5.7 - Evolução do hipervolume ao longo de 100 gerações.	60
Figura 5.8 - Formas do pavimento tipo do exemplo 05.	61
Figura 5.9 - Resultados do Exemplo 05.	61
Figura 5.10 - Modelo padrão de referência para exemplos 05 e 06.	62
Figura 5.11 - Evolução do hipervolume para o Exemplo 05 ao longo de 100 gerações.....	62
Figura 5.12 - Resultados do Exemplo 06.	63
Figura 5.13 - Evolução do hipervolume para o Exemplo 06 ao longo de 100 gerações.....	64

LISTA DE TABELAS

Tabela 4.1 - Códigos para a obtenção dos valores de armadura.	47
--	----

LISTA DE SÍMBOLOS

Símbolo	Significado
A_{fi}	Área de forma do i -ésimo elemento.
A_{si}	Área de aço do i -ésimo elemento.
C_c	Custo unitário por volume de concreto.
C_f	Custo unitário por área de formas.
C_s	Custo unitário por massa de aço.
$\mathbf{f}(\mathbf{x})$	Vetor de funções objetivo.
f_{desloc}	Função objetivo do deslocamento máximo.
$\mathbf{g}(\mathbf{x})$	Vetor de restrições.
h	Valor do peso da penalidade.
$h_i^{(p)}$	Altura do i -ésimo pilar.
$h_{max}^{(p)}$	Altura máxima da seção dos pilares.
$h_{min}^{(p)}$	Altura mínima da seção dos pilares.
$h_i^{(v)}$	Altura da i -ésima viga.
$h_{max}^{(v)}$	Altura máxima das vigas.
$h_{min}^{(v)}$	Altura mínima das vigas.
j	Índice de posicionamento das vigas na lista de vigas de um pavimento.
L_i	Comprimento do i -ésimo elemento.
M	Momento fletor.
N	Quantidade de elementos da estrutura.
n_{pav}	Número de vigas presente em um pavimento.
$n_{pilares}$	Número de pilares a serem posicionado em um edifício.
P_t	Valor da penalidade.
$S_{pilares}$	Quantidade máxima de seções distintas permitidas para os pilares.
S_{vigas}	Quantidade máxima de seções distintas permitidas para as vigas.
V_{ci}	Volume total de concreto.
\mathbf{x}	Vetor de variáveis de decisão.
δ	Vetor de deslocamentos.
μ_k	Valor de uma determinada grandeza genérica.
μ_k^{max}	Valor máximo permitido para uma determinada grandeza genérica.
γ_s	Peso específico do aço.
λ_i	Parâmetro de posicionamento do i -ésimo pilar.

1. INTRODUÇÃO

O projeto de estruturas usuais de concreto armado pode ser dividido simplificadaamente em quatro etapas iterativas: o lançamento da estrutura, análise, dimensionamento, e o detalhamento. Na primeira etapa, a disposição dos elementos é determinada, definindo a topologia e tipo do sistema estrutural, materiais a serem empregados, bem como as sobrecargas que atuarão sobre a estrutura e as suas condições de apoio. Na etapa seguinte, a análise é realizada de modo a determinar, entre outros, os esforços solicitantes e deslocamentos da estrutura. Com os dados provenientes da análise, pode-se então proceder com o dimensionamento pela verificação da seção dos elementos e a determinação da quantidade de armadura requerida para resistir às solicitações. A armadura deve ser convenientemente detalhada de modo que: atenda a critérios normativos, desenhos possam ser elaborados e as armaduras confeccionadas. Ainda que se trate de um procedimento sequencial, o processo como um todo é iterativo, sendo as etapas aqui definidas interdependentes.

Atualmente, o projeto de estruturas convencionais de concreto armado conta com o auxílio de programas computacionais especializados, que permitem a concepção de modelos complexos e proporcionam maior produtividade, possibilitando ao projetista uma maior liberdade e poder na concepção e estudo do comportamento da estrutura. Os referidos programas, atualmente automatizam grande parte do processo da elaboração de projetos, atuando principalmente nas etapas de determinação dos esforços, cálculos de dimensionamento e elaboração do detalhamento. Entretanto, a etapa do lançamento e concepção estrutural ainda possui um elevado nível de dependência do projetista. Os programas disponíveis ainda necessitam que o usuário determine manualmente a posição dos elementos, suas características geométricas, materiais, vinculações e outras características fundamentais para a definição do modelo da estrutura.

O lançamento estrutural, em especial, é uma das etapas mais importantes do processo de projetar estruturas pois tem influência direta nos resultados das etapas que o sucedem. Esta é a etapa do processo mais dependente do projetista e onde a importância da experiência do mesmo é mais significativa. Nas atividades cotidianas de projeto é comum definir a disposição e dimensões iniciais dos elementos baseados em regras matemáticas simples e recomendações práticas. As posições e as dimensões dos elementos são alteradas ao longo do processo até que se obtenha uma estrutura que apresente um comportamento desejado e satisfaça a critérios

normativos e de segurança, serviço e economia. Embora seja um procedimento bastante usual, o mesmo ainda é realizado com grande dependência do usuário, cabendo ao mesmo realizar cada uma das modificações ativamente.

1.1. OTIMIZAÇÃO APLICADA EM PROJETOS DE ESTRUTURAS

As técnicas de otimização têm sido aplicadas em diversos problemas no projeto e concepção de estruturas. Embora sejam técnicas utilizadas na busca de soluções ótimas, muitas delas não são necessariamente aplicadas com este o intuito. A otimização pode ser também utilizada na busca por soluções alternativas que possam ser mais atraentes frente a soluções já conhecidas e ou adotadas, sendo o motivo de sua utilização neste trabalho.

Quando aplicadas, as técnicas de otimização possibilitam a exploração do espaço de soluções, podendo gerar soluções que não foram imaginadas pelo utilizador ou simplesmente melhorando soluções predefinidas. A otimização tem sido aplicada em diversas áreas de conhecimento científico e prático. Constantemente, novas e mais eficientes técnicas são desenvolvidas e melhoradas, tornando a otimização uma ferramenta atrativa e de grande potencial na aplicação prática no projeto de estruturas.

No que diz respeito aos objetivos a serem alcançados, os problemas de otimização podem ser formulados de duas formas: problemas de único objetivo (mono-objetivo) ou problemas de vários objetivos (multiobjetivo). Nos problemas de otimização mono-objetivo a otimização se dá pela utilização de um único critério. Por exemplo, a busca pelo menor custo, menor massa, menor frequência natural, menor deslocamento ou maior rigidez, entre outros. Em problemas multiobjetivo, mais de um critério são considerados simultaneamente para realizar a otimização. Por exemplo, buscar simultaneamente pelo menor volume de concreto e menor peso de aço, menor deslocamento e menor frequência natural, menor custo, menor deslocamento e menor quantidade de elementos, entre outros.

Problemas mono-objetivo, em sua maioria, resultam em uma única solução e que se sobressai às demais; enquanto que, nos problemas multiobjetivo, é comum se obter um conjunto de soluções ótimas, sem domínio de nenhum dos objetivos sobre os demais, produzindo alternativas de projeto.

1.2. PROPOSTA DO TRABALHO

A proposta do presente trabalho consiste na aplicação de uma técnica de otimização multiobjetivo ao projeto de estruturas, em especial, voltadas a estruturas em pórticos de concreto armado.

Pretende-se expor uma metodologia para a exploração do espaço de soluções de projeto, de forma a apresentar e propor soluções de projeto alternativas, objetivando uma melhor performance da estrutura. A procura por estas soluções se dará através da busca por um melhor posicionamento dos pilares e variação das seções transversais dos elementos, considerando que as disposições das vigas de um pavimento são informadas antes do processo de otimização.

Trata-se, portanto, de um problema de otimização de topologia e dimensionamento, ainda que a primeira seja realizada parcialmente, devido a mudança apenas na configuração dos pilares, não contemplando mudança na disposição do vigamento.

A realização desse trabalho se justifica pela contribuição a oferecer ao estudo de projetos semiautomáticos de estruturas, pela busca e teste de soluções guiadas pelas preferências do projetista. Além disso, o presente trabalho busca propor uma alternativa às metodologias presentes em trabalhos encontrados na literatura, principalmente no que se refere ao posicionamento dos pilares em estruturas aporricadas, com pilares posicionados sob vigas. Diferentemente do encontrado na literatura, o presente trabalho não trata o posicionamento dos pilares em função dos vãos em direções perpendiculares, ou até mesmo em função das coordenadas dos pilares, apresentando uma nova abordagem ao problema. Técnicas de otimização multiobjetivo serão empregadas com o intuito de se obter um conjunto de soluções ótimas, produzindo projetos alternativos para o projetista.

1.3. OBJETIVOS

O objetivo principal deste trabalho consiste na elaboração de uma sistemática para a propor soluções alternativas de projeto, baseado no posicionamento dos pilares e variação das seções transversais, através a integração de um programa de análise estrutural e um algoritmo genético multiobjetivo.

1.3.1. Objetivos específicos

Ao concluir este trabalho, pretende-se ter alcançado os seguintes objetivos:

- Utilizar uma técnica de otimização multiobjetivo, em especial o algoritmo genético NSGA-II (Debs *et al.*, 2002), para otimização de estruturas de concreto armado e busca por soluções alternativas;
- Elaborar um algoritmo que permita a integração de um programa computacional de análise e dimensionamento estrutural, com o algoritmo de otimização multiobjetivo;
- Ser capaz de alterar iterativamente o posicionamento dos pilares e dimensões de seções transversais dos elementos, permitindo explorar soluções alternativas de projeto, inclusive para estrutura de topologia não-regulares;
- Estimar de forma aproximada custo da estrutura baseado no volume de concreto, peso de aço e área de forma;
- Determinar e verificar os deslocamentos máximos atuantes na estrutura;
- Encontrar o ótimo de Pareto para o problema;
- Propor um conjunto de alternativas ótimas para o projeto;
- Desenvolver uma ferramenta computacional que contemple todos os objetivos supracitados.

1.4. ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado de forma a apresentar ao leitor a utilização de técnicas de otimização e como a mesma tem sido aplicada a estruturas de concreto armado.

O Capítulo 2, inicialmente, apresenta aplicações gerais da otimização em diferentes áreas da engenharia, evidenciando sua grande aplicabilidade e versatilidade. Em seguida, trabalhos relevantes relacionados à otimização de estruturas são comentados, destacando suas características principais. Posteriormente, o estado da arte da otimização de estruturas de concreto é apresentado, destacando os trabalhos realizados e suas características principais.

No Capítulo 3 a metodologia é proposta, trazendo a filosofia adotada no trabalho, uma breve apresentação do algoritmo NSGA-II, considerações feitas na análise das estruturas estudadas e

sobre a integração com o programa computacional *Autodesk Robot Structural Analysis*[®] 2017 (*Robot*), utilizado para a análise estrutural.

No Capítulo 4 a implementação computacional é detalhada, explicando o desenvolvimento e a integração entre os módulos de trabalho que compõem o programa desenvolvido. Os resultados obtidos utilizando o programa desenvolvido e a metodologia proposta são apresentados no Capítulo 5. Comentários sobre o programa elaborado e sobre os resultados obtidos, assim como sugestões para trabalhos futuros, são apresentadas no Capítulo 6.

2. ESTADO DA ARTE

A aplicação de técnicas de otimização em problemas de engenharia tem sido explorada como uma ferramenta de auxílio na busca de melhores soluções para produtos. Neste capítulo, são apresentadas relevantes aplicações de técnicas de otimização na solução de diversos problemas, a sua aplicação em estruturas e, em especial, a estruturas de concreto armado.

Inicialmente, serão apresentados alguns conceitos fundamentais à compreensão de um problema de otimização. Algumas aplicações de técnicas de otimização também serão apresentadas, de modo a expor a versatilidade e aplicabilidade dos seus conceitos. Trabalhos envolvendo diferentes aplicações de otimização em estruturas são comentados, enfatizando o potencial de sua aplicação em problemas dessa natureza, com especial atenção a problemas do tipo multiobjetivo. Posteriormente, apresenta-se trabalhos considerados relevantes envolvendo otimização de estruturas de concreto armado, destacando características particulares destes e as técnicas empregadas em cada um. Finalmente, baseado nos trabalhos apresentados, define-se o escopo do presente trabalho, como este se relaciona com os trabalhos apresentados e o que o mesmo apresenta como contribuição.

2.1. PROBLEMAS DE OTIMIZAÇÃO

Projetar é um processo iterativo. Da concepção até a obtenção de um resultado final, o projeto de um produto passa por inúmeras modificações até que se chegue em uma solução que possa ser considerada aceitável.

Na etapa de projeto, o projetista propõe uma solução inicial baseada em sua experiência, intuição ou até mesmo em algumas análises matemáticas simples. A solução é então analisada para determinar sua aceitabilidade e, caso a solução seja aceita, o processo é terminado. Entretanto, no processo de otimização, a solução é modificada e analisada repetidas vezes de modo que seja possível determinar se uma solução é a melhor frente às demais.

Para ser possível avaliar se uma solução é a melhor, é preciso definir critérios que permitam avaliar essa característica. A escolha destes critérios é uma decisão do projetista e das necessidades de cada projeto. De acordo com [Arora \(2012\)](#), geralmente, os critérios envolvem o custo, eficiência, confiabilidade e durabilidade do produto. Conforme será apresentado nas seções seguintes desse trabalho, se tratando de otimização de estruturas, é comum encontrar

trabalhos que busquem: o menor custo com materiais, minimização de deslocamentos e ou frequências naturais, ou até mesmo a redução da quantidade de elementos submetidos a esforços de tração.

2.1.1. A formulação de um problema de otimização

Um problema de otimização genérico pode ser resumido aos seguintes elementos: função objetivo, variáveis de decisão e restrições. Matematicamente, problemas de otimização podem ser representados conforme as equações 2.1 até 2.3,

$$\mathbf{z}(\mathbf{x}) = \min[\mathbf{f}(\mathbf{x})] \quad (2.1)$$

Sujeito a:

$$\mathbf{g}_i(\mathbf{x}) \leq \mathbf{0}, i = 1, 2, 3, \dots, m \quad (2.2)$$

$$\mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_s \quad (2.3)$$

$$\mathbf{x} = x_j \in \mathbb{R}, j = 1, 2, 3, \dots, n$$

onde \mathbf{f} é o vetor das funções-objetivo, \mathbf{g}_i são as restrições impostas e \mathbf{x} é o vetor das variáveis de decisão.

A seguir, uma breve descrição de cada um dos componentes envolvidos na formulação de um problema de otimização é apresentada. Informações detalhadas sobre os tipos e formulação de otimização podem ser encontradas em [Gandomi et al. \(2013\)](#).

2.1.2. Funções objetivo

Nos problemas de otimização, estes critérios de comparação são avaliados através das “funções-objetivo”, $\mathbf{f}(\mathbf{x})$, as quais se deseja encontrar seus valores extremos, isto é, mínimos (quando se tratar de um problema de minimização), ou máximos (quando se tratar de um problema de maximização).

Problemas onde apenas um critério é adotado, isto é, apenas uma função objetivo é avaliada no processo, são chamados de problemas de otimização mono-objetivo. Se o problema busca minimizar ou maximizar mais de uma função objetivo simultaneamente, este é chamado de problema de otimização multiobjetivo. Problemas mono-objetivo geralmente resultam em uma

única solução, que se sobressai às demais. Entretanto, problemas de otimização multiobjetivo podem proporcionar um conjunto de soluções, onde todas podem ser consideradas ótimas.

2.1.3. Variáveis de decisão

As funções-objetivo produzem as respostas de um dado sistema, e dependem diretamente dos valores das chamadas “variáveis de decisão”, \mathbf{x} . Variáveis de decisão são parâmetros que caracterizam um determinado sistema e que, ao terem seus valores alterados, provocarão mudanças no comportamento do sistema e, conseqüentemente, na função-objetivo. Por exemplo, em um problema em que se deseja minimizar o deslocamento máximo de uma viga biapoiada, pode-se considerar como variáveis de decisão as dimensões da seção transversal, a rigidez do material que compõe a viga e os tipos e posições dos seus apoios.

Um sistema físico real tem uma elevada quantidade de variáveis que influenciam seu comportamento. Em problemas de engenharia, é comum fazer simplificações e tornar a quantidade de parâmetros finito, de modo que o sistema possa ser estudado e representado com significativa confiança. O estudo sobre quais variáveis devem ser levadas em consideração é uma das etapas mais importantes da formulação de um problema de otimização. Devendo ser realizado um estudo para identificar quais parâmetros terão maior influência sobre a função objetivo e quais destes deverão ser levados em consideração na análise. Informações sobre o efeito de cada uma das variáveis e quais se considerar em um problema de otimização podem ser encontradas em [Forrester \(2008\)](#).

2.1.4. Restrições

Definidos os objetivos da otimização e as variáveis de decisão, define-se as condições que uma determinada solução deve atender. A estas condições são chamadas “restrições”, $\mathbf{g}(\mathbf{x})$. As restrições podem ser descritas matematicamente por condições de igualdade ou desigualdade. As restrições limitam o domínio do problema, e podem ser manipuladas de modo a restringir as soluções que podem ser consideradas viáveis.

São exemplos de restrições encontradas em problemas de otimização estrutural: deslocamento máximo permitido, diferença entre a frequência natural e a excitante, limites às dimensões das seções dos elementos e comprimento de flambagem.

2.2. APLICAÇÕES DE TÉCNICAS DE OTIMIZAÇÃO

Um dos problemas clássicos da aplicação de otimização é o problema do caixeiro viajante. Considere que um caixeiro viajante possua uma lista de cidades as quais ele deve visitar e as distâncias entre elas, duas a duas. Este problema busca encontrar a melhor rota, isto é, a que minimize a distância por ele percorrida, sem que o mesmo passe duas vezes pela mesma cidade.

Este problema é um problema clássico e tem sido amplamente estudado. Alguns dos trabalhos envolvendo este tipo de problema são [Few \(1995\)](#), [Rosenkrantz \(1977\)](#) e [Dickson \(2016\)](#). Embora seja apresentado como o problema de um caixeiro viajante, a formulação do mesmo é aplicada a diversos problemas que envolvam minimização de percursos.

Outro problema clássico é o problema do empacotamento, onde se deseja empacotar objetos com dimensões variadas em uma determinada quantidade e pacotes (também de dimensões fixas), de modo que a quantidade de pacotes utilizados seja mínima. Os trabalhos de [Levine & Ducatelli \(2004\)](#), [Faina \(2000\)](#), [Stoyan et al. \(2012\)](#) e [Monaci \(2003\)](#), são alguns exemplos de trabalhos que tratam do tema.

Técnicas de otimização também são amplamente aplicadas em problemas de engenharia. Por exemplo, [Hicks & Henne \(1978\)](#) utilizaram o método do gradiente conjugado para a otimização do formato de asas de aeronaves. [Swanson et al. \(1993\)](#) estudaram a otimização de sistemas de suspensão de motores de aeronaves buscando uma melhor isolamento de vibrações e redução de deslocamentos excessivos. [MirHassani & Yarahmadi \(2017\)](#) estudaram a otimização do posicionamento de turbinas em parques eólicos, buscando maior produção energética, considerando as incertezas. [Izui et al. \(2013\)](#), propuseram a aplicação de técnicas de otimização multiobjetivo para a otimização da disposição de células robóticas em sistemas de produção industrial. Técnicas de otimização também foram empregadas em conjunto com redes neurais, no trabalho de [Xiao et al. \(2016\)](#), auxiliando na detecção de propósito de viagens a partir de pesquisas de viagens.

Os trabalhos apresentados mostram o quão versáteis as técnicas de otimização podem ser, sendo aplicadas em diversas áreas do conhecimento para solução de problemas.

2.3. OTIMIZAÇÃO DE ESTRUTURAS

A otimização tem sido amplamente aplicada a estruturas, com os mais diferentes propósitos. De modo geral, o uso de otimização busca melhorar uma ou mais características da estrutura, variando parâmetros que a influenciam. Por exemplo, em uma viga biapoiada sob ação de uma carga uniformemente distribuída, uma das características poderia ser o seu deslocamento máximo, e os parâmetros poderiam ser o vão a ser vencido pela viga, as dimensões de sua seção transversal ou até mesmo o material utilizado na mesma. Outros exemplos de características poderiam ser a sua frequência natural, rigidez, custo, resistência a esforços solicitantes, massa, entre outras. A seguir alguns trabalhos são apresentados de modo a apresentar o domínio de aplicação da otimização em estruturas.

Treliças são um dos tipos de estruturas mais estudadas no que diz respeito a otimização de estrutural. A otimização dessas estruturas é comumente dividida em três tipos principais: otimização topológica, de forma e de dimensionamento.

A seguir são apresentados alguns trabalhos envolvendo o uso de técnicas de otimização em treliças. O uso de técnicas de busca foi proposto por [Fu \(1973\)](#), o qual tinha como objetivo a otimização de treliças planas sujeitas a múltiplas condições de carregamento. A função objetivo considerada foi volume total de material e a topologia da treliça foi mantida fixa durante todo o processo, entretanto, sua forma e as áreas das seções transversais foram tomadas como variáveis de decisão. [Fu \(1973\)](#) concluiu que o uso de técnicas de busca era eficiente na procura por soluções ótimas e que poderia ser aplicada a estruturas estaticamente determinadas e indeterminadas. Outros trabalhos que tratam de otimização de treliças planas utilizando programação matemática podem ser consultados em [Trivedi & Munjal \(1991\)](#) e [Mottl \(1992\)](#),

De forma diferente, [Adeli & Kamal \(1985\)](#) empregaram a técnica de programação geométrica geral na busca do menor peso para treliças espaciais com geometria fixa. Tratava-se de um problema de programação não-linear e foi formulado pelo Princípio dos Trabalhos Virtuais. A função objetivo adotada foi o peso total da estrutura, tendo restrições de tensão e deslocamentos. As áreas dos elementos foram consideradas como variáveis de decisão, e tomadas como contínuas. O algoritmo desenvolvido se mostrou eficiente e robusto.

[Svanberg \(1980\)](#) também estudou o emprego de técnicas de otimização a treliças tridimensionais. Em seu trabalho, a função objetivo e restrições se assemelham as do trabalho

de [Adeli & Kamal \(1985\)](#), porém, com adição de uma restrição sobre a carga crítica de flambagem. Além das áreas das seções transversais, a forma da estrutura variou durante o processo de otimização através da inclusão de determinadas coordenadas nodais como variáveis de decisão. Para a análise estrutural foi empregado o Método dos Elementos Finitos (MEF), e a otimização se deu pela utilização de métodos duais de programação convexa.

Uma otimização baseada em probabilidades de treliça foi estudada por [Józwiak \(1989\)](#). Neste trabalho, o valor médio da massa da estrutura foi considerado como função objetivo. As restrições eram impostas a tensões e deslocamentos limites, considerando também a flambagem. A natureza estocástica das variáveis foi considerada pela variabilidade das resistências dos membros. O problema foi resolvido por técnicas de programação não-linear utilizando programação matemática. O autor destaca que à época, a consideração da natureza estocástica das propriedades da estrutura era complicada para o uso corrente, sendo recomendada a estruturas especiais. Outros trabalhos de otimização de treliças considerando a incerteza incluem [Liu & Moses \(1992\)](#) e [Nakib \(1997\)](#).

[Hajela & Lee \(1995\)](#) realizaram otimização topológica e de dimensionamento de treliças planas utilizando algoritmos genéticos e o método *ground structure* (estrutura base, tradução livre). O método *ground structure* consiste em definir previamente uma estrutura base, comumente contendo a maior quantidade de conexões possíveis entre os nós, de modo que, ao longo do processo de otimização, elementos possam ser removidos sucessivamente da estrutura até que se alcance o peso mínimo, atendendo as restrições impostas. A função objetivo neste trabalho consiste no peso total da estrutura, tendo como restrições a resistência dos membros (incluindo flambagem), limites para deslocamento, frequências e condições cinemáticas de estabilidade. Devido à natureza estocástica dos algoritmos genéticos, não é possível garantir que o mínimo global seja alcançado. Entretanto, pode ser utilizado em problemas com elevada não-linearidade, com descontinuidade no domínio de soluções e, quando bem configurado, não se prende a mínimos locais. Outros trabalhos que utilizam algoritmos genéticos para otimização de treliças podem ser encontrados em [Ohsaki \(1995\)](#) e [Wu & Chow \(1995\)](#).

[Rao \(1986\)](#), utilizou conjuntamente conceitos da Teoria dos Jogos e otimização multiobjetivo para otimização estrutural. Duas funções objetivos foram otimizadas simultaneamente: o peso total da estrutura e o deslocamento máximo. Foram consideradas variáveis a área da seção transversal dos elementos e as posições de determinados nós.

[Liu & Juang \(2013\)](#) aplicou técnicas de otimização multiobjetivo em problemas envolvendo pórticos metálicos rígidos. Em seu trabalho, três objetivos foram otimizados ao mesmo tempo: o peso da estrutura, o valor médio e a variação da demanda sísmica. O algoritmo utilizado foi o NSGA-II, e as seções dos elementos foram tomadas como variáveis de decisão, sendo as seções de vigas em um mesmo pavimento iguais, assim como a de pilares em um mesmo alinhamento em até dois níveis adjacentes. Os autores reportam que o uso da estratégia produziu uma família de soluções competitivas, que são econômicas, seguras e robustas, possibilitando assim o projetista escolher as vantagens proporcionadas por cada uma delas. A otimização de pórticos também foi estudada por [Yang et al. \(2012\)](#).

[Martinez-Martinet al. \(2012\)](#) aplicaram o algoritmo *simulated annealing* (recozimento simulado, tradução livre) multiobjetivo híbrido, no projeto de tabuleiros de pontes de concreto armado. Três funções objetivo foram avaliadas simultaneamente: o custo da estrutura, a quantidade de armadura e as emissões de CO₂.

Alguns trabalhos se dedicam a aplicar a otimização na redução de material de estruturas contínuas, através da otimização topológica. Trabalhos como os de [Zhu e Zhang \(2010\)](#) e [Buhl \(2002\)](#) realizaram otimização topológica de estruturas contínuas considerando variação nos apoios, porém com metodologias diferentes.

A otimização topológica de grelhas utilizando programação linear foi utilizada por [Kirsch e Taye \(1986\)](#), onde, partindo de uma grelha base, os elementos foram removidos continuamente até se obter uma solução ótima. Também sobre grelhas, [Yao et al. \(2015\)](#) utilizaram um algoritmo genético mono-objetivo para otimização do dimensionamento das seções transversais; enquanto [Maciunas e Belevicius \(2012\)](#) aplicaram um algoritmo genético adaptativo para a otimização multiobjetivo de grelhas de fundações, buscando a redução da carga máxima de reação nos apoios e o momento fletor máximo na grelha.

Um dos objetivos do presente trabalho consiste em alterar o posicionamento dos pilares durante o processo de otimização. Em estruturas de pórticos, uma das funções dos pilares é servir de apoio para as vigas de um pavimento. Portanto, alguns trabalhos consideram a mudança de posição dos apoios durante o processo de otimização, como descritos a seguir.

[Wang \(2004\)](#) fez uma análise de sensibilidade sobre o posicionamento de apoios simples em vigas e placas. Tanto apoios elásticos quanto rígidos foram considerados. A otimização também foi realizada de modo a buscar o posicionamento ótimo dos apoios de modo a minimizar o

deslocamento máximo na estrutura. Trabalho semelhante foi realizado por [Sesok et al. \(2010\)](#), ao otimizar o dimensionamento e posição de apoios em grelhas utilizando o algoritmo *simulated annealing*. [Bojczuk e Mróz \(1998\)](#) também estudaram de forma analítica a influência da posição dos apoios em vigas contínuas e pórticos planos, tendo como objetivo a minimização do custo da estrutura. Outras aplicações de otimização considerando a mudança da posição dos apoios em vigas e placas podem ser vistos em [Wang e Chen \(1996\)](#), [Imam e Al-Shihri \(1996\)](#).

Aqui uma atenção especial é dada ao trabalho do [Kripka \(1998\)](#). Nesse trabalho, o posicionamento ótimo de apoios em um pavimento de concreto armado analisado pela analogia de grelhas foi otimizado. A otimização consistia na uniformização dos esforços, em especial o momento fletor, entre as vigas que compõe um pavimento. Para tanto, a função objetivo foi tomada como igual ao somatório dos quadrados dos momentos negativos na grelha. Entre as restrições, encontram-se expressões que forcem a igualdade dos momentos positivos e negativos (em valor absoluto) em vão e nós adjacentes, bem como a garantia da negatividade dos momentos sob os apoios. A altura das vigas ao longo do processo foi mantida constante igual entre todas as vigas do pavimento. Em um trabalho posterior realizado por [Medeiros \(2010\)](#), orientado por Kripka, a otimização do dimensionamento foi contemplada. As coordenadas das posições relativas dos apoios em relação a extremidade inicial da viga foram consideradas como variáveis de decisão. É importante destacar que a quantidade de apoios é previamente definida. Além disso, cada apoio só pode ser deslocado no plano da viga que o contém, o que limita a formulação proposta.

Embora não seja o objetivo do presente trabalho, pesquisadores como [Shaw et al. \(2008\)](#) e [Sharafi et al. \(2017\)](#), aplicaram algoritmos de otimização para encontrar a disposição ideal das vigas de pavimentos. Os trabalhos neste tema se concentram em determinar o arranjo ideal de vigas em um pavimento estudado pela analogia de grelhas. Para tanto, considera inicialmente uma malha ortogonal, devidamente refinada, onde os elementos são retirados ao longo do processo de otimização até que se obtenha uma configuração que atenda a todas as restrições arquitetônicas e estruturais impostas. Alguns trabalhos relevantes sobre o tema são os de [Shaw et al. \(2008\)](#), [Sharafi et al. \(2015\)](#) e [Sharafi et al. \(2017\)](#).

Embora métodos diferentes de otimização tenham sido empregados, é possível perceber que a estrutura dos problemas é semelhante. As variáveis de decisão são comumente adotadas como sendo as dimensões das seções transversais, a presença ou não de um dado elemento, posições

de nós da estrutura ou até mesmo o posicionamento dos apoios. As restrições geralmente envolvem a verificação da capacidade resistente dos elementos e deslocamentos limites, ainda que outras restrições sejam aplicadas em casos especiais. A função objetivo para grande parte dos trabalhos é tomada como o custo da estrutura, que para o caso de materiais homogêneos e lineares é diretamente proporcional ao peso da estrutura.

É importante ressaltar que, na maioria dos trabalhos apresentados, a verificação da capacidade resistente é realizada pela metodologia das tensões admissíveis, considerando o material elástico linear. Portanto, o dimensionamento é realizado apenas de forma aproximada, não sendo considerados comportamentos não-lineares do material e geometria. Esta metodologia tem sido amplamente utilizada nos trabalhos de otimização estrutural existente na literatura, entretanto, não sendo eficiente em problemas de estruturas de concreto armado.

Na próxima seção são apresentados trabalhos relevantes de aplicação de técnicas de otimização a estruturas de concreto armado. Embora os problemas a seguir compartilhem de características da formulação dos problemas apresentados nessa seção, novas características são acrescentadas de modo a considerar as particularidades do material concreto armado, como por exemplo, a consideração da armadura.

2.4. OTIMIZAÇÃO DE ESTRUTURAS DE CONCRETO ARMADO

Diferentemente das estruturas de aço, as estruturas de concreto armado apresentam uma maior variedade e complexidade nas características do material, o que influencia diretamente em seu comportamento e análise. Devido a isto, o processo de otimização de estruturas de concreto tende a ser naturalmente mais complexo do que os envolvendo estruturas de aço.

Entre as variáveis consideradas nos problemas de estrutura de concreto se encontram: dimensões das seções transversais, resistência característica do concreto a compressão, resistência característica ao escoamento do aço, taxa de armadura nas seções, forma e topologia da estrutura.

Grande parte dos trabalhos que envolvem otimização de estruturas de concreto utilizam a abordagem da minimização do custo da estrutura, conforme pode ser visto em [Sarma & Adeli \(1998\)](#). Nestes problemas, o custo da estrutura é composto pela composição dos custos relativos às parcelas do concreto, aço e formas. O custo do concreto é calculado em função do volume

de concreto requerido, o do aço depende do seu peso e o das formas de sua área total, multiplicados pelos seus respectivos custos unitários. É importante ressaltar que os custos referentes à mão de obra, por simplificação, podem ser considerados embutidos nos preços unitários, como feito por [Babaei & Mollayi \(2016\)](#).

As restrições em problemas de otimização de estruturas de concreto podem contemplar: verificação da taxa de armadura em uma seção entre valores máximos e mínimos, limitação de deslocamentos, verificação da capacidade resistente, verificação das dimensões das peças entre valores máximos e mínimos, quantidade máxima de diferentes seções, entre outras.

Nas últimas décadas, algoritmos metaheurísticos, entre eles os algoritmos genéticos, têm sido amplamente utilizados em problemas de otimização. Estes algoritmos buscam mimetizar os mecanismos genéticos da seleção natural. De acordo com [Lee & Ahn \(2003\)](#), as principais características destes algoritmos incluem: a não necessidade de informações sobre o gradiente das funções, as funções envolvidas não precisam ser contínuas, e quando bem configurados, conseguem evitar mínimos locais. Entretanto, estes requerem elevada quantidade de avaliações da função objetivo, o que pode ser computacionalmente restritivo. Além disso, por serem de natureza intrinsicamente probabilística, podem não produzir os mesmos resultados em execuções independentes. São, também, altamente dependentes das soluções iniciais adotadas, uma vez que a busca pelo ótimo se dá através da recombinação dos indivíduos da população.

[Leps & Sejnoha \(2003\)](#) utilizou o método *augmented simulated annealing* (recozimento simulado aumentado, em tradução livre), para otimização de vigas de concreto armado. O principal objetivo da otimização era a minimização do custo da estrutura, levando em consideração parâmetros de resistência e serviço. Entre as variáveis da otimização está presente as dimensões das seções transversais da viga. A área de aço requerida em cada vão (superior e inferior) e as quantidade de barras em cada seção também foram consideradas variáveis de decisão. De forma semelhante, a área de aço requerida para a armadura transversal (em três regiões do vão) e o espaçamento necessário, também foram considerados. [Vidosa et al. \(2008\)](#) também estudaram o uso do algoritmo *augmented simulated annealing* na otimização de estruturas de concreto armado, em particular estruturas de contenção.

Algoritmos genéticos simples implementados em FORTRAN foram utilizados por [Govindaraj & Ramasamy \(2005\)](#), para o detalhamento otimizado de vigas contínuas de concreto armado. Diferente de outros trabalhos, apenas as seções transversais das vigas são consideradas como

variáveis de decisão. As armaduras e seu detalhamento são calculados após de definidas as seções transversais para cada iteração. O detalhamento foi realizado em dois estágios, baseados em configurações pré-definidas para a armadura. A função objetivo foi tomada como o custo da estrutura, composta pelo custo com concreto, aço e formas. As restrições foram com respeito à largura mínima de vigas para suportar as barras, limitação da razão entre largura e altura da viga, verificação do momento máximo, verificação da altura da linha neutra, quantidade de armadura mínima que chega no apoio, taxa de armadura máxima e mínima, e dimensões limites para as seções transversais.

A otimização isolada de pilares de concreto armado foi objeto de estudo de [Lee et al. \(2009\)](#). Nesse trabalho, pilares sujeitos a múltiplas condições de carregamento são dimensionadas e otimizadas utilizando técnicas de otimização matemática para problemas não-lineares e um método gráfico.

[Camp et al. \(2003\)](#) implementaram algoritmos genéticos para o dimensionamento e otimização de pórticos planos de concreto armado. O custo total da estrutura foi considerado como função objetivo, tendo como variáveis as dimensões das seções transversais, o diâmetro, número de barras e topologia da armadura. Entre as restrições se encontram limitações a deslocamentos, quantidade de barras por camada de armadura, dimensões limites de seções, capacidade resistente à flexão e cisalhamento, e deslocamentos da estrutura. Trabalho semelhante também foi realizado por [Fadaee e Grierson \(1996\)](#), [Guerra e Kiouisis \(2006\)](#) e [Aga e Adam \(2015\)](#).

[Lee & Ahn \(2003\)](#) também estudaram o dimensionamento otimizado de pórticos planos de concreto armado utilizando algoritmos genéticos. Para a otimização, um banco de dados de candidatas a seções típicas de concreto armado foi criado, contendo configurações preestabelecidas de armação, que já respeitam limitações normativas. Estas configurações diferem nas dimensões da seção, taxa e disposição da armadura. Para a otimização foi utilizado um algoritmo genético simples e a função objetivo foi o custo da estrutura. Uma metodologia semelhante à [Lee & Ahn \(2003\)](#) foi utilizada por [Kwak & Kim \(2008\)](#).

Outros tipos de algoritmos metaheurísticos foram utilizados por [Kaveh & Sabzi \(2012\)](#) (algoritmo *Big Bang-Big Crunch*) e [Kripka et al \(2015\)](#) (algoritmo *Harmony Search Method*, método da busca harmônica, em tradução livre), para otimização de pórticos de concreto armado.

A otimização do custo e da disposição dos elementos para pórticos espaciais de concreto armado foi o objetivo da pesquisa de [Sharafi et al. \(2012\)](#). Em seus estudos foi empregado o algoritmo *Ant System* (sistema de formigas, tradução livre) para a busca do menor custo da estrutura através da alteração das dimensões de suas seções transversais e da disposição dos pilares. É importante ressaltar que a metodologia apresentada por [Sharafi et al \(2012\)](#), no que diz respeito à otimização da disposição dos pilares, simplesmente busca as dimensões ótimas dos vãos entre pilares em duas direções ortogonais, não sendo evidente a possibilidade da disposição em direções arbitrárias. Embora esta metodologia possa ser eficiente para estruturas com formatos retangulares e onde os pórticos são regularmente espaçados, a mesma pode não apresentar a mesma eficiência para topologias irregulares.

Alguns autores utilizaram o mesmo algoritmo de otimização multiobjetivo empregado no presente trabalho, o NSGA-II. O NSGA-II é um algoritmo metaheurístico de otimização multiobjetivo, com a principal característica de apresentar um sistema rápido de classificação de soluções baseado no conceito de não-dominância. Os indivíduos da solução ótima se distribuem em uma região conhecida como Fronteira de Pareto, não sendo apresentado apenas uma solução, mas um conjunto de soluções ótimas.

[Babaei & Mollayi \(2016\)](#) empregaram o NSGA-II na otimização multiobjetivo de pórticos planos de concreto armado. Entre as variáveis de decisão estavam as dimensões das seções transversais e as armaduras dos elementos, com restrições quanto à resistência dos elementos e deslocamentos. Como funções objetivo foram adotadas o custo da estrutura e o deslocamento horizontal máximo.

[Stankovic et al. \(2012\)](#) utilizaram o NSGA-II para a otimização multiobjetivo a partir de uma metodologia conjunta com o algoritmo *NodeSort*, para o projeto de treliças. Como objetivos principais foram considerados o peso da estrutura e o deslocamento máximo da estrutura, enquanto variando sua topologia e seção dos elementos.

[Vo-Duy et al. \(2017\)](#), estudando vigas compósitas de laminados, também utilizaram o NSGA-II para otimização, buscando, simultaneamente, minimizar o peso e maximizar a frequência natural dos elementos. Entre as variáveis de decisão estavam a fração de fibras por volume, a espessura e orientação das camadas. Outro trabalho que empregou o NSGA-II para a otimização multiobjetivo de materiais compósitos laminados foi o apresentado por [Honda et al. \(2013\)](#), aplicado á placas com fibras de formas curvilíneas.

2.5. COMENTÁRIOS GERAIS

Os trabalhos aqui apresentados apresentam algumas características em comum. As variáveis de decisão mais adotadas foram as dimensões das seções transversais e a taxa ou quantidade de armadura na seção, sendo em alguns casos utilizadas seções pré-definidas. As restrições são primariamente relacionadas à capacidade resistente e deslocamentos limites. Outras restrições são relativas a critérios normativos ou construtivos, como dimensões mínimas, abertura de fissuras, espaçamento entre as barras, limitação da quantidade de camadas de armadura ou adoção de uma quantidade máxima de diferentes seções adotadas. A função objetivo geralmente é adotada como a soma dos custos com volume de concreto, peso total de aço das armaduras e área de formas. Em problemas multiobjetivo, além do custo, outros objetivos são: deslocamento máximo da estrutura, frequência natural média, quantidade de elementos, reação máxima nos apoios, entre outras.

É importante destacar que na maioria dos trabalhos apresentados, o dimensionamento das armaduras foi realizado apenas para seções críticas do elemento, adotadas como a seção de momento positivo máximo no vão e a região de momento negativo máximo sob os apoios. Na maioria dos casos levantados a armadura dimensionada para a seção crítica positiva foi adotada para todas as seções do vão, sem escalonamento de armadura. Para os momentos negativos, a armadura encontrada para o momento máximo negativo se estende perto da região de momento negativo, também sem escalonamento. Essa metodologia se deve ao fato de o detalhamento refletir as preferências do projetista e práticas regionais, ainda que o mesmo deva atender a critérios normativos. Na maioria dos trabalhos apresentados o custo com mão-de-obra não é considerado diretamente, tendo os autores incorporados tais custos aos valores unitários dos materiais. A adoção de um detalhamento simplificado, ainda que possa ocasionar um aumento no custo com material, pode levar a uma maior facilidade e velocidade de execução, reduzindo o custo com mão-de-obra.

No presente trabalho, a otimização se dará através da variação das seções dos elementos e mudança do posicionamento dos pilares. Diferentemente dos trabalhos presentes, buscar-se-á a mudança do posicionamento dos pilares para estruturas de topologias não regulares, não se limitando a estruturas de padrões retangulares. Devido a um dos objetivos do trabalho ser produzir alternativas de projeto, um algoritmo de otimização multiobjetivo será aplicado, o NSGA-II (Deb *et al.*, 2002), que tem versões distribuídas com a biblioteca computacional *jMetal* (Durilo & Nebro, 2011). A análise da estrutura será realizada pelo programa *Autodesk*

Robot2017 (Robot), sendo a integração entre o algoritmo de otimização e o *Robot* realizada através do programa desenvolvido neste trabalho.

O *Robot* é um programa para o cálculo de estruturas mantido pela empresa *Autodesk*, que utiliza elementos finitos possui tecnologia BIM (modelagem baseada em informação, da sigla em inglês). Uma das vantagens de sua utilização está na *API* distribuída junto ao programa, que consiste em um conjunto de ferramentas que permitem a integração com programas externos e desenvolvimento de novas funções.

3. METODOLOGIA

Os trabalhos apresentados nas seções anteriores mostram a versatilidade da otimização e de sua aplicação na solução de problemas estruturais, sendo aplicadas em diferentes áreas e propósitos. Especialmente no que diz respeito a estruturas de concreto armado, grande parte dos trabalhos onde a otimização é empregada se dedicam a otimizar elementos isolados. Embora existam trabalhos envolvendo a otimização de pórticos planos e espaciais, estes estão presentes em menor quantidade. Além disso, a maioria destes se resume a minimizar o custo da estrutura através da variação das seções transversais e quantidade de armadura nos elementos. Como ainda apresentado no Capítulo 2, o emprego do NSGA-II e outros algoritmos multiobjetivo em problemas de otimização ainda é muito pequeno.

Uma decisão que deve ser tomada pelo engenheiro de estruturas na concepção de um sistema estrutural diz respeito ao posicionamento dos pilares, que para grande parte das estruturas usuais de concreto armado servem como apoio para as vigas dos pavimentos.

A consideração do posicionamento dos pilares em problemas de otimização é um tema ainda pouco explorado na literatura. Todavia, é sabido que a mudança das condições de suporte das vigas, por exemplo, pode alterar o estado de tensões nestes elementos e, conseqüentemente, os esforços e os deslocamentos em uma estrutura, como apresentado na Figura 3.1.

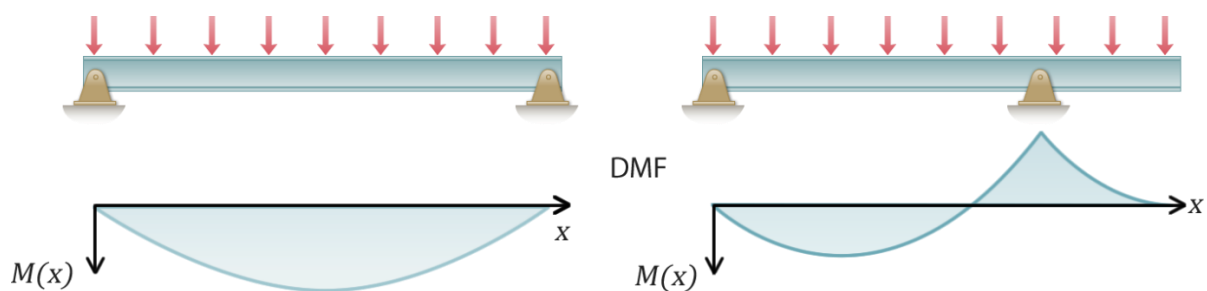


Figura 3.1- Influência do posicionamento dos apoios nos momentos fletores.

Outro fator de grande influência no comportamento da estrutura são as seções transversais. As características das seções transversais têm influência direta na capacidade resistente dos elementos e em sua rigidez. Em especial em vigas de concreto armado, a altura das seções transversais tem importância fundamental para a quantidade de armadura requerida e para os deslocamentos da estrutura.

Neste capítulo, será apresentada a metodologia utilizada neste trabalho para a otimização multiobjetivo de estruturas em pórticos de concreto armado. A seguir é apresentada a sistemática adotada para a formulação do problema de otimização, uma breve apresentação do programa para a análise estrutural (*Robot*), o algoritmo de otimização multiobjetivo utilizado (NSGA-II), e do procedimento para a comunicação entre eles.

3.1. FORMULAÇÃO DO PROBLEMA DE OTIMIZAÇÃO

3.1.1. Considerações sobre o modelo estrutural

Este trabalho contempla apenas o caso de estruturas aporricadas de concreto armado compostas por lajes apoiadas em vigas e as vigas apoiadas diretamente sobre os pilares. Não são contemplados, por exemplo, edifícios onde as lajes possam se apoiar diretamente sobre os pilares.

Nos projetos correntes de edifícios em concreto armado, a disposição das vigas em um pavimento, em sua maioria, é condicionada às posições das alvenarias. Entretanto, é possível que haja o posicionamento das vigas em locais por onde não passem alvenarias, como apresentado no esquema da Figura 3.1.

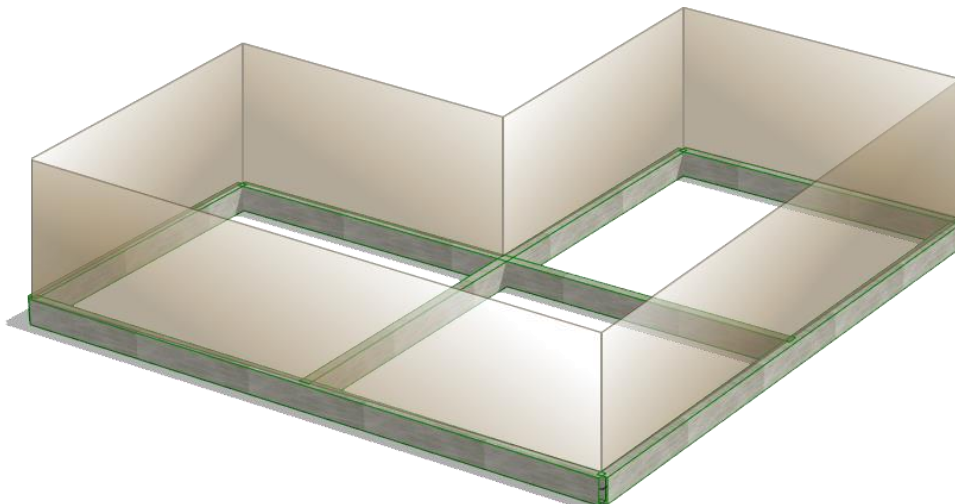


Figura 3.2- Exemplo da predefinição das vigas de um pavimento.

Ainda que esta consideração não seja necessariamente restritiva, para fins de simplificação será adotado neste trabalho que a disposição das vigas é determinada anteriormente ao processo de otimização.

Assim como as vigas, as lajes são consideradas como predefinidas pelo usuário, devido à se considerar neste trabalho que as definições dos seus painéis sejam delimitadas pelas vigas. Assim, as lajes permanecerão inalteradas durante o processo de otimização, ainda que sejam consideradas na análise do modelo.

3.1.2. Dimensões das seções transversais

Embora a disposição das vigas permaneça inalterada durante o processo de otimização, as dimensões de suas seções transversais podem ser alteradas a cada iteração. Para tanto, é necessário definir os limites máximos e mínimos permitidos para as dimensões das seções transversais a serem produzidas durante o processo de otimização. Devido a menor influência das larguras das seções transversais na capacidade resistente à flexão das vigas, a largura das seções é considerada constante e igual a todas as vigas. Assim, apenas os limites máximos e mínimos para a altura das seções transversais devem ser definidos, respectivamente, $h_{max}^{(v)}$ e $h_{min}^{(v)}$.

Devido à facilidade e economia da execução, é comum limitar a quantidade de seções distintas utilizadas em um projeto. Esta consideração proporciona uma economia na confecção de formas, maior facilidade de execução e economia de mão-de-obra. Entretanto, a quantidade máxima de seções distintas (s_{vigas}) a serem consideradas no projeto é uma escolha do projetista, não existindo um consenso ou regra geral que limite o seu uso. Portanto, neste trabalho, a quantidade máxima de seções distintas permitidas durante o processo de otimização é definida pelo usuário.

Uma estratégia semelhante será utilizada para os pilares, onde a quantidade máxima de seções distintas para os pilares ($s_{pilares}$) deverá ser arbitrada pelo usuário. Assim como para as vigas, a menor dimensão do pilar será mantida constante para todos os pilares, sendo a maior dimensão variável, tendo como limites máximos, $h_{max}^{(p)}$ e $h_{min}^{(p)}$, respectivamente.

As condições aqui definidas referentes às dimensões das seções podem ser resumidas pelas equações 3.1 e 3.2, respectivamente para as seções de vigas e pilares.

$$h_{min}^{(v)} \leq h_i^{(v)} \leq h_{max}^{(v)} \quad (3.1)$$

$$h_{min}^{(p)} \leq h_i^{(p)} \leq h_{max}^{(p)} \quad (3.2)$$

Além das quantidades máximas de seções utilizadas para pilares e suas dimensões limite, pretende-se, nas próximas etapas do trabalho, considerar a quantidade de pilares na estrutura ser variável, participando do processo de otimização, e não ter valor fixo.

3.1.3. Posicionamento dos pilares

A estratégia para o posicionamento dos pilares da estrutura é uma das etapas mais complexas da modelagem do problema de otimização. Será considerado que o posicionamento dos pilares estará restrito ao domínio das vigas, isto é, só será possível posicionar pilares que interceptem vigas, não sendo permitido o posicionamento de pilares em contato direto com as lajes.

É importante destacar que esta é uma limitação imposta ao trabalho. Esta metodologia, como apresentada neste trabalho é aplicada apenas à construções onde os pilares devam ser posicionados sob as vigas, não considerando o posicionamento dos pilares sob as lajes. Esta consideração pode limitar sua aplicação em outros casos, principalmente devido à metodologia apresentada no presente item do trabalho, resumida na Figura 3.3.

Nos trabalhos presentes na literatura, o posicionamento dos pilares, quando considerado, é parametrizado por meio da construção da distribuição uniforme em direções perpendiculares, semelhante a matrizes, como o considerado por [Sharafi et al. \(2012\)](#). Em outra estratégia, menos utilizada, o posicionamento se dá pelas coordenadas do centro geométrico da seção do pilar em planta, sendo verificado em seguida se a interceptação com vigas ocorre.

O procedimento para o posicionamento dos pilares neste trabalho consiste em: (1) transformar cada pavimento em uma viga equivalente parametrizada, (2) definir o posicionamento do pilar nesta viga equivalente, (3) identificar em qual elemento de viga do pavimento e (4) em qual posição deste deverá ser posicionado o pilar. Os estágios do procedimento descrito estão esquematicamente representados na Figura 3.3.

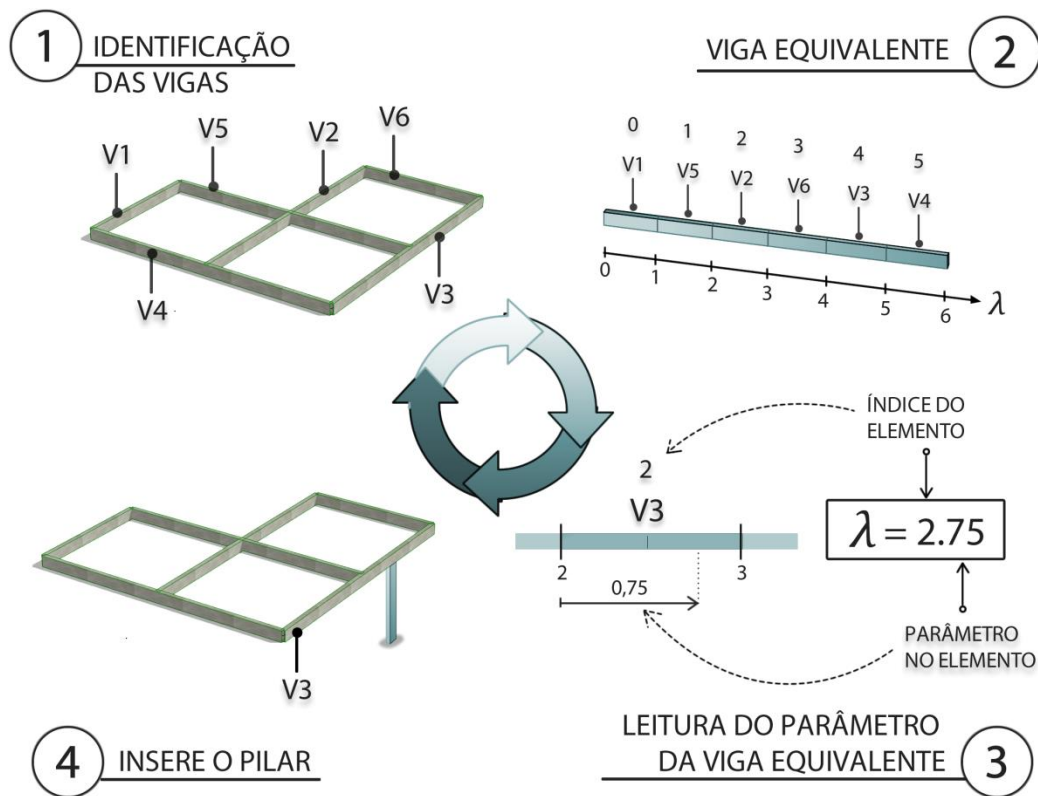


Figura 3.3 – Esquema dos estágios do posicionamento dos pilares.

No estágio 1 da Figura 3.3, as vigas do pavimento são identificadas e guardadas em uma lista ordenada, sendo possível acessá-las posteriormente através de um índice, j , correspondente ao seu posicionamento na lista, onde $0 \leq j < n_{pav}$, sendo n_{pav} a quantidade de vigas em um pavimento.

No estágio 2, as vigas do pavimento são transformadas em uma viga equivalente. A viga equivalente é composta por uma quantidade n_{pav} de vigas de comprimento unitário, posicionadas em série, respeitando a ordem dos elementos na lista das vigas do pavimento criada no estágio 1. É importante observar que a viga equivalente terá comprimento total igual a quantidade de vigas presentes no pavimento.

Devido à adoção da estratégia de conversão de um pavimento em uma única viga equivalente, o posicionamento de um pilar poderá ser definido através de um único parâmetro de posicionamento, λ_i , que identificará em qual posição da viga equivalente o pilar deverá ser posicionado. Esta estratégia possibilita a redução da quantidade de variáveis necessárias para definir o posicionamento dos pilares, sendo uma das contribuições adicional deste trabalho. É importante destacar que nas estratégias presentes na literatura, duas variáveis são necessárias para caracterizar o posicionamento dos pilares. Em uma estratégia, considera-se as coordenadas

dos pilares para definir a sua posição, sendo necessárias duas variáveis (para as coordenadas x_i e y_i do pilar), além da verificação da interseção com as vigas. Em outra estratégia, o posicionamento de um pilar é caracterizado por uma variável para identificar em qual elemento o pilar deve ser posicionado, e uma segunda variável que identifica em qual posição deste elemento deverá ser posicionado. Para exemplificar, suponha que se deseja posicionar 15 (quinze) pilares em uma estrutura. Nas estratégias adotadas na literatura, seriam necessárias duas variáveis para cada pilar, isto é, 30 (trinta) variáveis ao todo para representar o posicionamento dos 15 (quinze) pilares. Na estratégia adotada neste trabalho, será necessária a definição de apenas uma variável por pilar, isto é, 15 (quinze) variáveis, reduzindo pela metade a quantidade de variáveis requeridas para os pilares e, conseqüentemente, a complexidade do problema.

A partir do parâmetro de posicionamento, λ_i , é possível definir em qual posição da viga equivalente o mesmo deverá ser alocado, como apresentado no estágio 3 da Figura 3.3. Devido ao parâmetro de posicionamento ser uma variável de decisão, seus limites admissíveis precisam ser definidos obedecendo a equação 3.3.

$$0 \leq \lambda_i < (n_{pav}) \quad i = 1, 2, \dots, n_{pilares} \quad (3.3)$$

Se λ_i respeitar a equação 3.3, o índice na lista de vigas do pavimento do elemento no qual o pilar deverá ser posicionado será dado pela parte inteira de λ_i , enquanto sua posição relativa neste elemento será dada pela parte decimal de λ_i e $n_{pilares}$ a quantidade de pilares considerados. Portanto, o índice na lista de vigas de um pavimento do elemento no qual o pilar deve ser posicionado é dado pela equação 3.4, e a posição relativa do posicionamento neste pilar é expresso pela equação 3.5,

$$j = \lfloor \lambda_i \rfloor = \mathbf{max}\{m \in \mathbb{Z} \mid m \leq \lambda_i\} \quad (3.4)$$

$$\bar{\lambda}_i = \lambda_i - j \quad (3.5)$$

onde j é o índice na lista das vigas do pavimento do elemento onde o pilar deverá ser posicionado, $\lfloor \lambda_i \rfloor$ é a função piso (ou o arredondamento para baixo, ver [Iverson \(1969, p. 12\)](#)) do parâmetro de posicionamento na viga equivalente, e $\bar{\lambda}_i$ é o parâmetro de posicionamento relativo do pilar na j -ésima viga da lista de vigas do pavimento. A Figura 3.4 exemplifica a utilização do parâmetro $\bar{\lambda}_i$ para o posicionamento de um pilar na j -ésima viga de um pavimento.

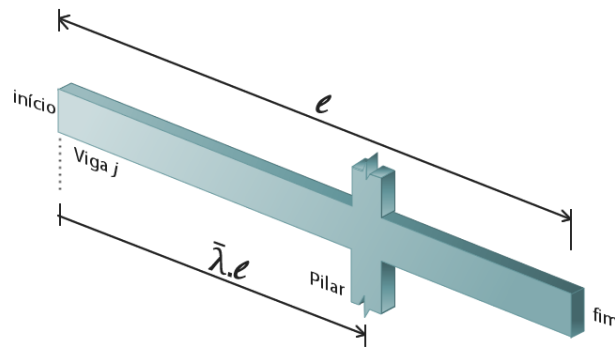


Figura 3.4 - Parâmetro de um pilar em uma viga do pavimento.

A metodologia apresentada aqui não faz nenhuma restrição ao não posicionamento de pilares em determinada viga, isto é, não há necessidade de uma predefinição de quantos pilares existirão em uma viga, ou até mesmo de deverão existir ou não em um elemento. Todo o gerenciamento do posicionamento dos pilares é realizado pela metodologia apresentada anteriormente.

3.1.4. Vetor de projeto

Em um projeto de otimização, as variáveis de decisão do problema devem ser reunidas em um único vetor, chamado aqui de *vetor de projeto*. O vetor solução particularizará uma solução no processo de otimização, sendo modificado a cada interação. Devido o vetor de projeto conter cada uma das variáveis de decisão, a quantidade de informações contidas no mesmo irá variar para cada problema analisado.

Para os problemas apresentados no Capítulo 5, o vetor solução deverá conter as seguintes variáveis de decisão: as alturas das seções das vigas ($\mathbf{x}^{(1)}$), a informação de qual seção deverá ser atribuída a cada viga ($\mathbf{x}^{(2)}$) as dimensões máximas das seções dos pilares ($\mathbf{x}^{(3)}$), a informação de qual seção deverá ser atribuída a cada pilar ($\mathbf{x}^{(4)}$) e os parâmetros de posicionamento λ_i de cada pilar ($\mathbf{x}^{(5)}$). A composição do vetor de projeto, \mathbf{X} , está representado esquematicamente na Figura 3.5.

$$\mathbf{X}^T = \left\{ \begin{array}{l} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \mathbf{x}^{(3)} \\ \mathbf{x}^{(4)} \\ \mathbf{x}^{(5)} \end{array} \right\}^T \begin{array}{l} \longrightarrow \text{dimensão das seções de vigas} \\ \longrightarrow \text{seção das vigas} \\ \longrightarrow \text{dimensão das seções dos pilares} \\ \longrightarrow \text{seção dos pilares} \\ \longrightarrow \text{posição dos pilares} \end{array}$$

Figura 3.5 - Representação esquemática do vetor de projeto.

Uma definição mais formal, o vetor de projeto, \mathbf{x} , é apresentada na equação (3.6, sendo suas componentes definidas pelas equações 3.7 a 3.11,

$$\mathbf{x} = \{\mathbf{x}^{(1)} \quad \mathbf{x}^{(2)} \quad \mathbf{x}^{(3)} \quad \mathbf{x}^{(4)} \quad \mathbf{x}^{(5)}\} \quad (3.6)$$

$$\mathbf{x}^{(1)} = x_i^{(1)} \quad i = 1, 2, \dots, s_{vigas} \quad (3.7)$$

$$\mathbf{x}^{(2)} = x_k^{(2)} \quad k = 1, 2, \dots, n_{vigas} \quad (3.8)$$

$$\mathbf{x}^{(3)} = x_m^{(3)} \quad m = 1, 2, \dots, s_{pilares} \quad (3.9)$$

$$\mathbf{x}^{(4)} = x_p^{(4)} \quad p = 1, 2, \dots, n_{pilares} \quad (3.10)$$

$$\mathbf{x}^{(5)} = x_p^{(5)} \quad p = 1, 2, \dots, n_{pilares} \quad (3.11)$$

Onde $n_{pilares}$ e n_{vigas} são a quantidade de pilares e vigas no modelo, respectivamente.

Em alguns dos exemplos apresentados no Capítulo 5, as seções dos elementos são mantidas fixas, isto é, inalteradas ao longo do processo de otimização. Nestes casos, as componentes do vetor de projeto respectivas às seções dos elementos são suprimidas, restando apenas as componentes referentes ao posicionamento dos pilares.

3.1.5. Funções objetivo

Em problemas de otimização multiobjetivo, a função objetivo é representada por um vetor de funções objetivo. Neste trabalho, são adotadas prioritariamente combinações das seguintes funções objetivo: o custo total da estrutura, o deslocamento máximo entre as vigas, a diferença entre os valores absolutos do máximo momento máximo positivo e negativo e o máximo momento fletor absoluto entre as vigas. Cada combinação é apresentada nos exemplos do Capítulo 5.

O custo total da estrutura, quando considerado, neste trabalho é referente ao consumo de materiais, podendo o custo com mão-de-obra ser indireta e aproximadamente considerado através dos custos unitários dos materiais. A função custo é definida conforme a equação 3.12,

$$f_{custo} = \sum_{i=1}^N C_c \cdot V_{c_i} + C_s \cdot \gamma_s \cdot A_{s_i} \cdot L_i + C_f \cdot A_{f_i} \quad (3.12)$$

onde C_c , C_s e C_f são o custo unitário por volume de concreto, por peso de aço e área de forma, respectivamente; N é a quantidade de elementos no modelo, V_{c_i} é o volume de concreto, A_{s_i} é a área de aço, L_i o comprimento e A_{f_i} a área de forma do i -ésimo elemento da estrutura.

A função do deslocamento máximo entre as vigas consiste em buscar na estrutura qual o deslocamento máximo, em valor absoluto. O deslocamento máximo é obtido navegando nos elementos de viga do modelo e avaliando se os deslocamentos dos mesmos é o maior em valor absoluto na estrutura. A função deslocamento máximo é formalizada na equação 3.13,

$$f_{desloc} = \max(\delta) \quad (3.13)$$

onde δ é o vetor de deslocamentos da estrutura.

A função da diferença entre os valores absolutos do máximo momento fletor positivo e negativo é obtida navegando entre os momentos fletores das vigas da estrutura, em busca do valor máximo dos momentos positivo e negativo. Em seguida, é realizada a diferença entre os seus valores absolutos.

A função do momento máximo da estrutura busca descobrir qual o valor máximo absoluto entre momentos positivos e negativos entre as vigas.

É intuitivo que o aumento das seções em estruturas submetidas a flexão tende a aumentar sua rigidez e, conseqüentemente, reduzir os deslocamentos. Entretanto, o aumento das seções provocará um aumento no consumo de materiais, elevando o custo com materiais da estrutura. Deste modo, percebe-se que as duas funções tendem a apresentar uma correlação inversa, isto é, se o valor de uma aumenta, o da outra tende a diminuir. Portanto, tendo em vista que se deseja minimizar ambas as funções objetivo simultaneamente, buscar-se-á fazê-lo de modo que não melhore uma em detrimento da outra, buscando encontrar a Fronteira de Pareto para o problema, de forma semelhante a apresentada na Figura 3.6.

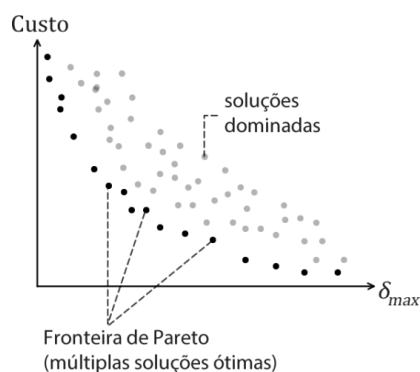


Figura 3.6 - Exemplo de uma Fronteira de Pareto.

Os critérios de resistência não são considerados diretamente na otimização. A partir de uma estrutura sugerida pelo processo de otimização, é verificada para a mesma se esta apresenta algum problema em sua análise e dimensionamento, realizado automaticamente pelo *Robot*. Em caso de alguma inconsistência no dimensionamento ou análise ser encontrada, a solução é penalizada.

3.1.6. Restrições

As restrições impostas no problema de otimização são, neste trabalho, referentes às dimensões máximas e mínimas das seções transversais dos elementos e deslocamentos máximos. Diferentemente de grande parte dos trabalhos apresentados na literatura, não é verificada diretamente a capacidade resistente dos elementos, sendo os elementos dimensionados para resistirem aos esforços solicitantes e, caso dimensionamento não seja possível, as funções objetivo são penalizadas.

Devido à integração com o *Robot*, a verificação de flechas limite, o dimensionamento e a eventual verificação da capacidade resistente dos elementos, podem ser feitas pelo próprio *Robot*, dependendo da norma adotada para o projeto e personalizações do projetista. Ocorrendo a violação dos limites configurados no próprio *Robot*, alertas de cálculo são emitidas pelo mesmo, informando a não conformidade do projeto aos parâmetros definidos. Antecipa-se que alertas quaisquer de violação das condições de dimensionamento e verificação emitidas pelo *Robot*, independentemente de sua natureza, são tratados simplesmente como violação das condições de projeto, penalizando as funções objetivos.

As penalizações às funções objetivos consistem em amplificar o valor calculado das funções por pesos, de modo que o projeto que apresente violações às restrições possa ser desfavorecido

no processo de otimização, diminuindo suas chances de estar presente na solução ótima final. A função penalidade adotada no presente trabalho é semelhante à adotada por Miguel *et al.* (2012), modificada para uma grandeza generalizada, e é apresentada na equação 3.14,

$$P_t(\mathbf{x}) = h \cdot \left[\sum_{k=1}^N \left| \frac{\mu_k(\mathbf{x}) - \mu_k^{max}}{\mu_k^{max}} \right| \right] \quad (3.14)$$

onde N é quantidade de elementos na estrutura, h é o peso da penalidade, comumente definido como um número alto, \mathbf{x} é o vetor de projeto analisado, μ_k é o valor de uma determinada grandeza que se deseja avaliar e μ_k^{max} é o valor máximo permitido para a grandeza em questão.

3.2. ANÁLISE ESTRUTURAL

A análise da estrutura será realizada utilizando o programa computacional *Autodesk Robot Structural Analysis*® 2017 (*Robot*). O *Robot* é um programa de análise estrutural por elementos finitos, com tecnologia *BIM* (modelagem baseada na informação, da sigla em inglês), mantido pela empresa *Autodesk*. Com o *Robot* é possível realizar desde a análise linear de estruturas de barras, até modelos complexos de elementos finitos, com simulação computacional de fluidos para análise dinâmica, considerando efeitos causados pelo vento (ver Figura 3.7).

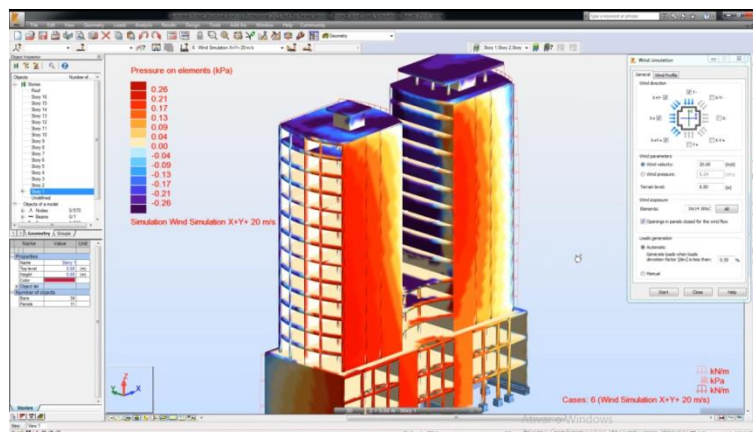


Figura 3.7 - Interface do programa *Autodesk Robot Structural Analysis* 2017.

Fonte: Autodesk.

No presente trabalho, o *Robot* é utilizado como ambiente para a modelagem e configuração da estrutura inicial, processamento dos resultados e visualização dos resultados finais.

É pela entrada gráfica que o usuário deverá definir a posição das vigas e as dimensões iniciais de suas seções, os painéis de lajes e os pilares que o mesmo deseja que se mantenham

inalterados durante a análise. A partir da estrutura modelada é que se dará a mudança das seções dos elementos, a adição e mudança de posição dos novos pilares, baseado em uma quantidade de pilares desejada pelo usuário.

É importante ressaltar que alguns cuidados devem ser tomados na definição do modelo da estrutura para que a metodologia aqui utilizada funcione corretamente. Entre estes cuidados está na definição do tipo do elemento para vigas e pilares. No *Robot*, elementos lineares podem ser do tipo *viga*, *coluna* ou *barra*. Para que o dimensionamento e o posicionamento ocorram como o esperado, os elementos lineares que representam as vigas devem ser definidos no *Robot* como tendo no atributo *Objeto Estrutural* a opção *Vigas*, sendo ainda inseridas como *Vigas de CA* (podendo ser verificado no atributo *Tipo*), em referência a vigas de concreto armado (ver Figura 3.8).

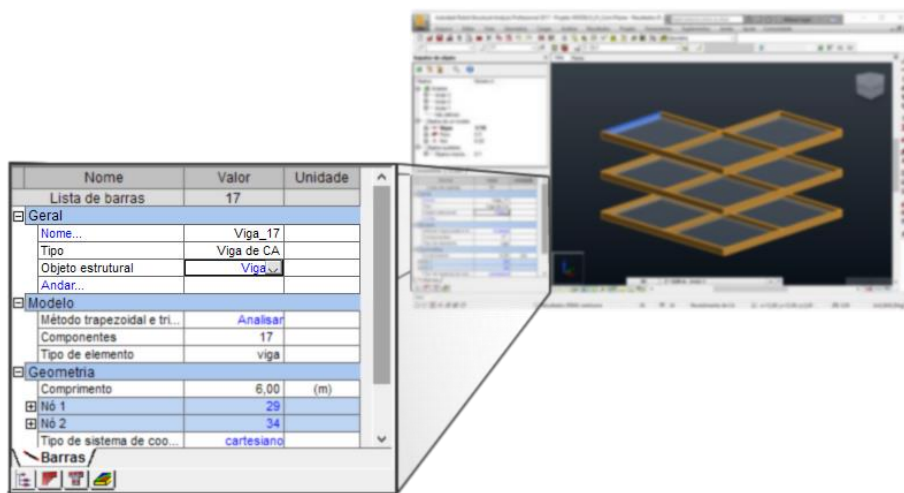


Figura 3.8 - Propriedades dos elementos das vigas.

De forma similar, os elementos lineares que representam pilares devem ser inseridos como *Colunas de CA* (podendo ser verificado no atributo *Tipo*), em referência aos pilares de concreto armado, e verificado se o atributo *objeto estrutural* está definido como *Coluna* (ver Figura 3.9).

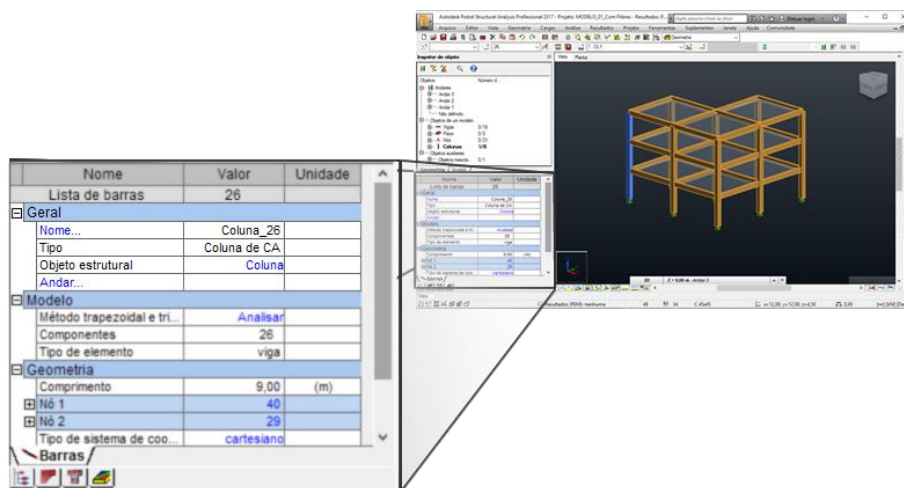


Figura 3.9 - Propriedade dos elementos dos pilares.

Além das apresentadas, outras configurações são importantes para a análise, como a escolha da norma a ser utilizada para as verificações, bem como os parâmetros da norma e de cálculo. Esta versão do trabalho não contempla detalhes sobre as configurações possíveis e as necessárias ao *Robot*, devendo estas informações serem detalhas nas etapas seguintes

3.2.1. Robot API

O principal motivo da utilização do *Robot* para a análise das estruturas neste trabalho se deve ao fato de ser distribuída junto ao programa a *Robot API* (interface de programação de aplicações, da sigla em inglês), isto é, um conjunto de ferramentas para o desenvolvimento de programas que utilizem das bibliotecas e funções do *Robot*, permitindo a interação do *Robot* com aplicações externas. A utilização da *Robot API* é possível através da implementação computacional nas linguagens de programação *C++*, *C#*, *Visual Basic*, ou qualquer outra linguagem de programação com suporte à tecnologia *Component Object Model* (Modelo de objeto componente, em tradução livre). Entre as possibilidades da utilização da *Robot API* estão: automatizar tarefas repetitivas, criar aplicações para facilitar a modelagem de estruturas típicas, analisar projetos, produzir relatórios personalizados, importar dados externos ao *Robot*, integração com outros programas, acessar os resultados e criar funções adicionais.

As principais funções da *Robot API* utilizadas no presente trabalho se dedicam à modificação da geometria da estrutura, criação e alteração de propriedades das seções transversais dos elementos, definição e modificação de apoios, realização do cálculo da estrutura, obtenção de resultados como: deslocamentos, esforços internos, reações de apoio, frequências naturais,

acelerações, levantamento de quantitativo de materiais, dimensionamento e verificação das armaduras. A implementação computacional do uso da *Robot API* e a integração com o algoritmo de otimização são apresentadas no Capítulo 4.

3.3. OTIMIZAÇÃO EVOLUTIVA MULTIOBJETIVO: NSGA-II

Algoritmos evolutivos são algoritmos que mimetizam o comportamento evolutivo dos indivíduos. Na nomenclatura dos algoritmos evolutivos, cada solução é chamada de *indivíduo*. O conjunto de indivíduos gerados em uma iteração do algoritmo (também chamada de *geração*), é chamado de *população*. Durante cada iteração, indivíduos dominantes, com maior aptidão de melhorar a solução, chamados de *pais*, são selecionados e são combinados através do procedimento de *combinação* e *mutação* para gerar novos indivíduos, isto é, novas soluções. Quando uma parte da população é mantida inalterada entre iterações, diz-se que ocorre o *elitismo*. O elitismo é uma estratégia adotada de modo a conservar indivíduos dominantes, que apresentem uma boa aptidão, durante as gerações e contra os procedimentos de mutação e cruzamento. A cada geração, o procedimento a seguir é repetido até que se alcance o fim das gerações ou um critério de convergência estabelecido: uma população aleatória de indivíduos é gerada, a função objetivo é avaliada para cada indivíduo da população, e uma nova população deverá ser criada; a criação de uma nova população se dá através da seleção dos pais, cruzamento entre os pais com ocorrência ou não de mutação (leve alteração nas características herdadas dos pais); em seguida a população anterior é substituída pela nova população, com ou sem elitismo, e o processo é repetido até que se atinja um critério de parada.

O NSGA-II (algoritmo genético de classificação não-dominante 2, na tradução da sigla do inglês), é um algoritmo genético elitista desenvolvido por [Debs et al. \(2002\)](#), para a solução de problemas de otimização multiobjetivo. A principal característica do NSGA-II é o procedimento de classificação rápida não-dominante, onde uma população é classificada em fronteiras formadas por indivíduos não-dominantes, isto é, todos têm chances iguais de reprodução. Em cada iteração deste algoritmo, indivíduos são identificados para formar as Fronteiras de Pareto, baseado no conceito de não-dominância ([GANDOMI et al., 2013](#)), sendo os procedimentos padrão dos algoritmos evolutivos conduzidos em seguida. Outra característica importante do NSGA-II é a preservação da diversidade das soluções ([DEBS et al., 2002](#)), isto é, os indivíduos tendem a estar bem distribuídos sobre as Fronteiras de Pareto em cada geração. O pseudocódigo do NSGA-II está descrito na Figura 3.10.

```

Início
Inicialize Populacao;
Gere N soluções aleatórias e insira em Populacao;
Para i = 1 até MaxGeracoes, faça:
  Gere PopulacaoFilha de tamanho N;
  Selecione Pais da Populacao;
  Crie Filha a partir de Pais;
  Faça a mutação em Filha;
  Combine Populacao e PopulacaoFilha em PopulacaoAtual com tamanho  $2N$ ;
  Para cada individuo em PopulacaoAtual faça
    Atribua rank baseado em Pareto (Classificação Rápida Não-Dominante);
  FimPara
  Gere conjuntos de vetores não-dominantes ao longo da
  FrenteiraParetoConhecida
  Adicione soluções a próxima geração de Populacao iniciando da melhor
  fronteira
  ...até que N soluções sejam encontradas e determine a distancia entre
  ...os indivíduos de cada fronteira;
FimPara
Mostre os resultados;
Fim

```

Figura 3.10 - Pseudocódigo do algoritmo NSGA-II. Fonte: Adaptado de [Syberfeldt \(2014\)](#).

O algoritmo NSGA-II presente neste trabalho está presente na biblioteca *jMetal*, elaborada por [Durilo e Nebro \(2011\)](#), originalmente para a linguagem *Java*, possuindo também uma versão para *C#*, sendo a utilizada neste trabalho. Para mais detalhes sobre o NSGA-II e sua implementação, é recomendado ao leitor os trabalhos de [Durilo e Nebro \(2011\)](#) e [Debs et al. \(2002\)](#), não sendo sua implementação direta um dos objetivos deste trabalho.

No que diz respeito à convergência do NSGA-II, muitos trabalhos buscam encontrar uma forma de se avaliar de forma eficaz: a convergência, isto é, a proximidade da última fronteira de soluções encontradas da Fronteira de Pareto; e a diversidade dos resultados da última fronteira de soluções obtidas, isto é, a diferença entre as soluções, seja no que diz respeito aos valores dos objetivos ou do vetor de variáveis. Alguns autores, como [Zheng et al. \(2012\)](#), relataram que, embora produza resultados eficientes no que diz respeito à proximidade da Fronteira de Pareto ideal, o NSGA-II pode apresentar falta de diversidade nas soluções encontradas.

Embora não haja uma unanimidade, uma métrica utilizada comumente em algoritmos multiobjetivos são o *hipervolume* e a distância entre as soluções em uma fronteira. Entretanto, uma das métricas utilizadas pelo NSGA-II implementados na *jMetal*, é o *hipervolume*, sendo a aqui utilizada com os resultados.

O hipervolume foi proposto por [Zitzler & Thiele \(1998\)](#), avalia a área de soluções coberta (ou dominada) por um conjunto de soluções não-dominadas (pertencentes a uma fronteira), em um

algoritmo multiobjetivo, conforme esquematizado na Figura 3.11, onde H_1 e H_2 são os hipervolumes esquemáticos de duas gerações distintas.

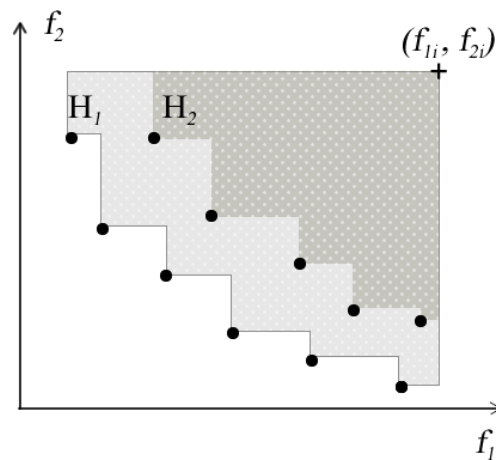


Figura 3.11 - Exemplo da avaliação do hipervolume para dois conjuntos de soluções.

Pode-se observar que, para a avaliação do hipervolume (área coberta), toma-se um ponto de referência no espaço de soluções (f_1, f_2) e, para cada região delimitada por este ponto e a fronteira de interesse, se calcula o hipervolume aproximado (a área, considerando dois objetivos) coberto por esse conjunto de soluções.

Conforme apresentado no trabalho do [Azevedo & Araújo \(2011\)](#), o hipervolume é um indicador interessante por ser uma medida de avaliação tanto da diversidade quanto da proximidade das soluções à Fronteira de Pareto teórica. Logo, quanto maior o hipervolume para um conjunto de soluções, significa que este domina uma quantidade maior de soluções. Além disso, se for adotada uma métrica entre a razão dos hipervolumes teóricos de uma fronteira de soluções qualquer ao longo das iterações e o da Fronteira de Pareto teórica (estimada, para um problema), quanto mais próxima de 1 esta relação for, isto é, quanto mais próximo o hipervolume de uma solução estiver do hipervolume da Fronteira de Pareto teórica do problema, melhor será essa solução. Entre os autores que estudaram a relação entre o hipercubo e diversidade estão [Tahernezhadani et al \(2012\)](#) e [Masin & Bukchin \(2008\)](#).

3.4. INTEROPERABILIDADE ROBOT - NSGA-II

Para que a estrutura definida no *Robot* possa ser otimizada se faz necessária a integração do mesmo com um algoritmo de otimização NSGA-II. A integração pode ser realizada através da

implementação computacional de um programa que utilize a *Robot API* e faça a integração com a biblioteca *jMetal*, que contempla entre seus algoritmos o NSGA-II.

Portanto, rotinas computacionais que permitam a integração entre o *Robot* e o NSGA-II foram desenvolvidas neste trabalho e divididas em módulos, baseados em suas funções. A presente versão, utilizada neste trabalho, consiste em seis módulos principais: *PainelControl*, *Flx*, *RobotConexao*, *Modelo*, *Otimizacao* e *NSGA-II* (ver

Figura 3.12).

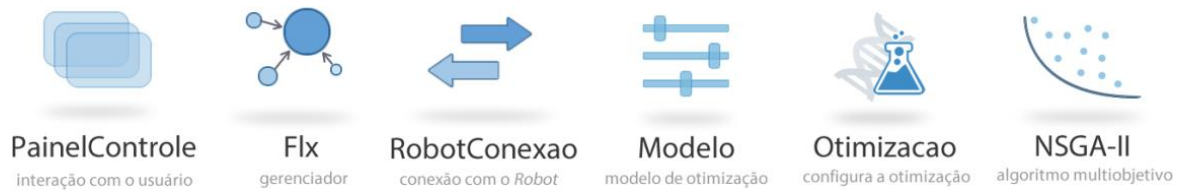


Figura 3.12- Módulos do programa em desenvolvimento.

O módulo *PainelControl* é responsável pela interface gráfica amigável com o usuário, entrada de dados e apresentação dos resultados. O *Flx* é responsável pela comunicação entre a interface e os demais módulos. É neste módulo que os dados da interface são lidos e passados para os outros módulos. O módulo *Modelo* é responsável por armazenar os valores das variáveis de decisão e seus limites, e onde a determinação das restrições e funções objetivos são chamadas.

O módulo *Otimizacao* é responsável pela a estrutura do processo de otimização, sendo nele definido o *Modelo* a ser otimizado e os parâmetros da otimização, como o tamanho da população e quantidade máxima de gerações. Dentro do módulo *Otimizacao* é que o módulo *NSGA-II* é executado, sendo neste último onde são executadas as operações do processo de otimização efetivamente e parâmetros do algoritmo como taxa de mutação, cruzamento e seleção. Entretanto, após realizada a otimização, os resultados e parâmetros da análise são armazenados no módulo *Otimizacao*, sendo os mesmos transferidos sucessivamente aos módulos *Flx* e *PainelControl*, onde são apresentados ao usuário. A integração e relação entre os módulos é apresentada Figura 3.13, sendo detalhada no capítulo 4.

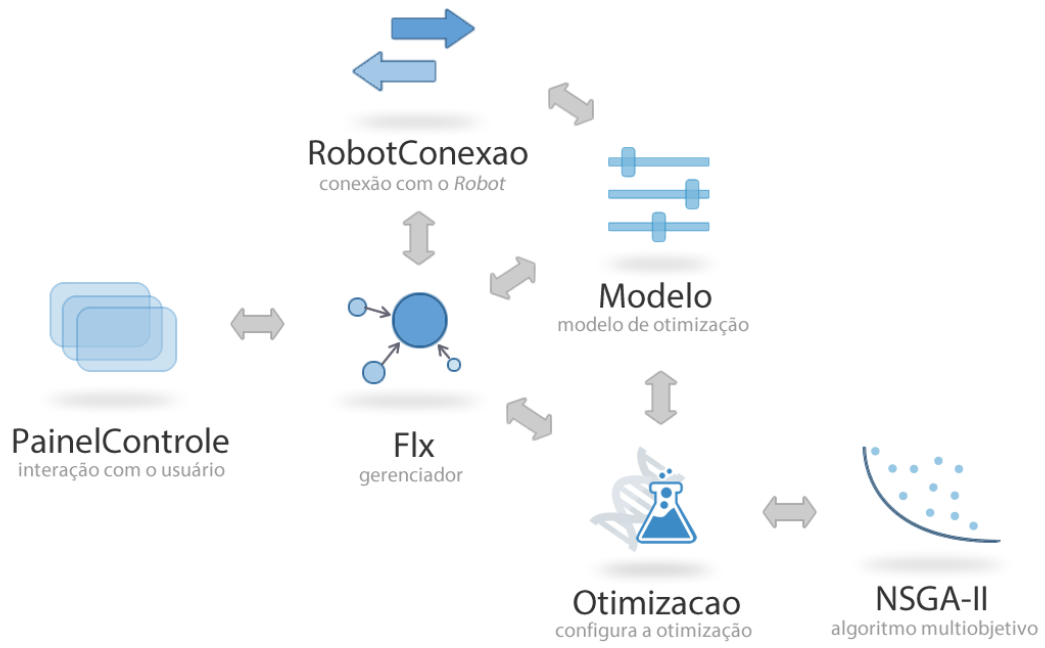


Figura 3.13 - Interação entre os módulos.

4. IMPLEMENTAÇÃO

A interoperabilidade entre o programa computacional de análise estrutural, *Robot*, e a ferramenta de otimização multiobjetivo, o algoritmo NSGA-II, será detalhada no presente capítulo.

Para a integração seja possível, é utilizada a API disponibilizada junto ao *Robot*, a *Robot API* (sendo daqui em diante referida apenas como a *API*), e a biblioteca *jMetal*, que entre seus algoritmos apresenta o NSGA-II e suas variações, além de outros algoritmos de otimização mono e multiobjetivo. Neste presente capítulo, uma maior ênfase será dada a apresentação das funções mais utilizadas da API e sua integração com o NSGA-II. Não se pretende descrever com detalhes o funcionamento do algoritmo NSGA-II, apenas as modificações necessárias à sua interação com a API e os módulos desenvolvidos neste trabalho.

A disposição das vigas e demais parâmetros não envolvidos no processo de otimização e todos os parâmetros que devem permanecer inalterados ao longo do processo de otimização, devem ser configuradas diretamente na interface gráfica do *Robot*, como no cotidiano de projeto. A partir desse modelo básico é que a otimização será conduzida.

A integração entre o *Robot* e o NSGA-II se deu por meio do desenvolvimento de módulos de interação elaborados na linguagem de programação *C#*, utilizando o paradigma da programação orientada a objetos (POO). A escolha da linguagem se deu devido tanto a *API* do *Robot* quanto o NSGA-II distribuído com o *jMetal* possuírem implementações nesta linguagem, facilitando o processo de desenvolvimento da integração. Para o desenvolvimento das rotinas foi utilizado principalmente a versão 2016 do ambiente de desenvolvimento integrado (IDE, da sigla em inglês) *Visual Studio Community*, da empresa *Microsoft*.

Por considerar mais didático, será primeiramente apresentada a descrição da implementação das funções de controle do *Robot* via *API*, destacando as funções básicas utilizadas que possibilitaram a integração, implementadas no módulo *RobotConexao*. Em seguida, serão apresentadas as modificações necessárias nas rotinas do NSGA-II para possibilitar a integração com o *Robot*, principalmente no que diz respeito à definição das variáveis e objetivos dos problemas e restrições, sendo essas modificações presentes no módulo *NSGA-II*. Por último, serão apresentados os módulos que permitiram a integração entre o módulo *RobotConexao* e *NSGA-II*, descritos através do módulo *Modelo, Otimizacao e Flx*.

4.1. MÓDULO ROBOT

A principal função desse módulo é permitir a manipulação do modelo estrutural do *Robot*, execução da análise e a leitura dos resultados por ele produzidos. Para a criação desse módulo foi implementada uma classe homônima, *RobotConexao*, que possui entre os principais métodos:

- *start()*: responsável pela criação da conexão com o *Robot* e leitura da geometria e propriedades da estrutura.
- *optSetup()*: responsável por criar as seções que serão modificadas durante a otimização, verificar a quantidade de variáveis envolvidas no problema e determinar os seus limites numéricos superior e inferior, que definem o domínio de atribuição de seus valores.
- *getStoreyEqBeamTS()*: responsável pela criação da viga equivalente de um pavimento, conforme o procedimento descrito no item 3.1.3 do presente trabalho.
- *placeColumn()*: método responsável pelo posicionamento dos pilares em uma iteração. Este método recebe dois parâmetros: um vetor de valores reais respectivo aos parâmetros de posicionamento na viga equivalente, λ_i , de comprimento igual à quantidade de pilares a serem posicionados; e um vetor de inteiros que informam qual a seção a ser aplicada a cada um dos pilares a posicionar, útil no caso de variação da seção dos pilares durante a otimização.
- *dumpTables()*: método responsável por interpretar, principalmente, os resultados de armadura e comentários sobre o dimensionamento do *Robot*, quando estas informações forem necessárias.
- *Evaluate()*: o método responsável por chamar todos os métodos supracitados durante cada uma das iterações do processo de otimização. É neste método onde todo o procedimento que ocorre em uma iteração é invocado, desde a aplicação das seções aos elementos, posicionamento dos pilares, execução da análise, dimensionamento de armaduras e levantamento de quantitativos.

4.1.1. Utilizando a *RobotAPI*

Para utilizar a API do *Robot*, é necessário verificar se a mesma foi instalada junto ao programa, pois se trata de uma opção que é por padrão desabilitada. Durante o processo de instalação do programa, pode-se optar pela instalação da API conforme apresentada na Figura 4.1 para a versão 2016.

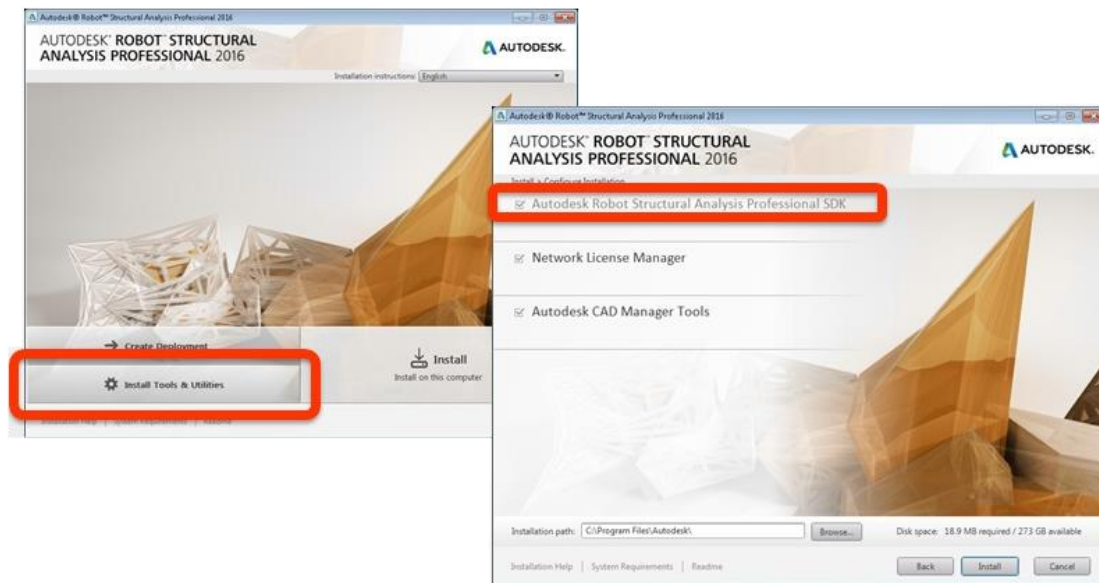


Figura 4.1 - Detalhe da instalação da API do *Robot*.

Quando instalada, a API estará disponível no diretório escolhido para a instalação do programa. Sua utilização se dará através da referência à biblioteca *RobotOM.dll*, presente no diretório da API. A seguir são apresentados exemplos da conectividade com o *Robot* e a implementação das principais funções utilizadas.

4.1.2. Conexão com o *Robot*

Para a conexão com o *Robot*, é necessária a criação de um elemento da classe *RobotApplication*, que dá acesso ao núcleo do programa, permitindo, entre outras, o gerenciamento das janelas, tabelas e dos projetos abertos. O gerenciador de projetos do *Robot* pode ser acessado através do objeto da classe *RobotProject*, permitindo sua criação e manipulação e, portanto, acesso à estrutura presente no projeto. O acesso ao modelo estrutural se dá através do objeto da classe *RobotStructure*, presente no objeto *RobotProject*. É através deste objeto que é possível a manipulação da geometria, definição de carregamentos, execução da análise e obtenção dos

resultados, como deslocamentos, esforços e armaduras. Um exemplo do uso desses objetos é apresentado esquematicamente na Figura 4.2, sendo estes destacados em negrito.

```
using System;
using System.Collections.Generic;
using System.IO;
using RobotOM; // Referencia a biblioteca do Robot

namespace RobotConsole
{
    class RobotConexao
    {
        public void start()
        {
            RobotApplication robot = new RobotApplication(); // Acesso ao Robot
            RobotProject projeto = robot.Project; // Acesso ao projeto atual
            RobotStructure estrutura = project.Structure; // Acesso ao modelo
            estrutural
        }
    }
}
```

Figura 4.2 - Exemplo de acesso aos objetos principais da *RobotAPI*.

4.1.2.1. Gerenciamento de nós e barras

A manipulação da geometria se dá pelo objeto do modelo da classe *RobotStructure*, utilizando os gerenciadores de barras e nós do modelo, respectivamente através dos objetos *RobotBarServer* e *RobotNodeServer*, conforme demonstrado na Figura 4.3.

```
RobotBarServer barras = structure.Bars; // Acesso as barras do modelo
RobotNodeServer nos = structure.Nodes; // Acesso aos nós do modelo
```

Figura 4.3 - Exemplo de acesso as barras e nós de um modelo.

Através dos servidores é possível adicionar nós e barras ao modelo, e também alterar suas propriedades, como atribuição de seções às barras e aplicação de restrições (apoios) aos nós.

A criação de um nó se dá através do informe de suas coordenadas e número do nó, utilizando o servidor dos nós. Já as barras são criadas informando sua numeração no modelo e os números

dos nós que definem seu início e fim. Para exemplificar, na Figura 4.4 são criados dois nós e uma barra utilizando os nós criados.

```
nos.Create(12, 4, 5, 3); // Criação do nó 12, com coordenadas x=4, y=5 e z=3
nos.Create(13, 8, 2, 4); // Criação do nó 13, com coordenadas x=8, y=2 e z=4
barras.Create(2, 12, 13); // Criação da barra 2, através dos nós 12 e 13
```

Figura 4.4 - Exemplo de criação de nós e barras.

4.1.2.2. Seções transversais e apoios

A definição das seções transversais e apoios, via API, no *Robot* se dá por meio da criação de *Labels*, utilizando o objeto *RobotStructure*, e atribuição destes aos nós (no caso dos apoios) e às barras (no caso das seções transversais). Um exemplo da criação de um apoio e sua atribuição a um nó, assim como a criação de uma seção e sua atribuição a uma barra, são exemplificados na Figura 4.5.

```
// Criacao de uma secao de nome "V2_14x30"
IRobotLabel secao=estrutura.Labels.Create(IRobotLabelType.I_LT_BAR_SECTION, "V2_14x30");

// Criacao de uma secao de pilar de concreto armado de forma retangular
secao.Data.ShapeType = IRobotBarSectionShapeType.I_BSST_CONCR_COL_R;

// Definição da seção rectangular b = 14cm, h = 30cm
secao.Data.Concrete.SetValue(IRobotBarSectionConcretDataValue.I_BSCDV_COL_B, 0.14);
secao.Data.Concrete.SetValue(IRobotBarSectionConcretDataValue.I_BSCDV_COL_H, 0.30);

// Atribuição da seção criada a vida V2
barras.SetLabel(2, IRobotLabelType.I_LT_BAR_SECTION, "V2_14x30");

// Criacao de um apoio do tipo engaste
IRobotLabel apoio=estrutura.Labels.Create(IRobotLabelType.I_LT_SUPPORT, "ENGASTE");

// Definindo as restrições dos apoios
IRobotNodeSupportData Apoio_data = apoio.Data;
Apoio_data.UX = true;
Apoio_data.UY = true;
Apoio_data.UZ = true;
Apoio_data.RX = true;
Apoio_data.RY = true;
Apoio_data.RZ = true;

// Atribuindo ao nó 13
nos.SetLabel(13, IRobotLabelType.I_LT_SUPPORT, "ENGASTE");
```

Figura 4.5 - Exemplo de criação e aplicação de seções e apoios.

4.1.2.3. Execução da análise

A configuração do motor de análise do *Robot* pode ser realizada utilizando o objeto da classe *RobotCalcEngine*, acessível por meio do objeto do projeto, *RobotProject*. Entre os parâmetros possíveis de configuração via API estão: a escolha do *solver* a ser utilizado, a configuração da auto geração do modelo computacional, precisão e acesso a janela de *status* da análise. No entanto, no presente trabalho todos esses parâmetros foram configurados na própria interface do *Robot*, sendo a classe *RobotCalcEngine* utilizada apenas para realização da análise. A análise pode ser realizada através da execução do método *Calculate()*, do objeto *RobotCalcEngine*, como exemplificado na Figura 4.6.

```
RobotCalcEngine engine = structure.CalcEngine; // Instância do motor de cálculo
engine.Calculate(); // Chamada da realização dos cálculos
```

Figura 4.6 - Exemplo de invocação do método de execução da análise.

4.1.2.4. Resultados de esforços e deslocamentos

Após realizada a análise, os resultados podem ser acessados através do objeto da classe *RobotStructure*. Entre os resultados disponíveis estão deslocamentos, rotações, esforços, tensões, frequências, entre outros. O exemplo da obtenção dos deslocamentos e momentos no meio do elemento 2, da Figura 4.4 são obtidos conforme a Figura 4.7.

```
// Deslocamentos da barra 2, no parâmetro 0.5 (meio da barra), para o caso de
// carregamento 1.
estrutura.Results.Bars.Displacements.Value(2, 0.5, 1);

// Momento fletor em torno do eixo local y do elemento 2, para o caso 1 de
// carregamento.
estrutura.Results.Bars.Forces.Value(2, 1, 0.5);
```

Figura 4.7 - Exemplo da obtenção de deslocamentos e momentos nos elementos.

Por meio das funções apresentadas na Figura 4.7 e do uso de laços de repetição, foi possível se obter os deslocamentos e momentos máximos nos elementos, utilizados em alguns exemplos.

4.1.2.5. Resultados de dimensionamento

O *Robot* disponibiliza por meio de sua API a realização dos cálculos de armadura necessária para elementos de barra, previamente definidos como vigas ou pilares (colunas, no *Robot* em português), se estes forem ainda definidos como elementos de concreto armado (*Vigas de CA* e

Colunas de CA, na interface do *Robot*). Esta definição dos elementos de barra como elementos de concreto armado é obrigatória para ser possível a estimativa da armadura teórica necessária por meio do *Robot*, ocasionando erro de dimensionamento se tal configuração não for realizada.

Antes de apresentar como devem ser configurados os elementos de barra como vigas e pilares de concreto no *Robot*, é necessário realizar alguns comentários sobre o dimensionamento realizado pelo *Robot*:

1. O dimensionamento realizado pelo *Robot* diz respeito a área de armadura teórica necessária para os elementos de vigas e pilares, e está sujeita a critérios normativos definidos pela a escolha de um código de projeto (uma norma técnica) disponível no sistema.
2. Entre os códigos presentes no *Robot* não está presente a norma brasileira, a NBR 6118, em nenhuma de suas versões. Portanto, na apresentação dos resultados que envolvam armadura nesse trabalho, estes foram feitas conforme o Eurocode 5, por escolha do autor.
3. Quando as execuções dos cálculos de estimativa de armadura foram realizadas utilizando a API, observou-se uma instabilidade no sistema, ocasionando, em alguns casos, a não realização dos cálculos para determinados elementos. Tal instabilidade ocorreu de forma aleatória, sendo investigado e sugerindo que o erro não ocorria por má implementação, e sim por uma deficiência da API. A comparação foi realizada utilizando o mesmo modelo construído inteiramente via API, inclusive suas propriedades, e executadas duas análises: primeiramente via API, que apresentou instabilidade e a não execução de alguns elementos de pilares; e, logo em seguida, no mesmo procedimento, a execução manual do mesmo modelo gerado pela API, sem nenhuma alteração de qualquer forma no modelo, resultando em sucesso. Tentou-se ainda contato com os desenvolvedores para esclarecer este problema, mas não se obteve sucesso.

Realizados estes comentários, a configuração necessária de elementos de barra para se comportarem como vigas ou pilares de concreto armado é apresentada na Figura 4.8.

```
// Referenciando uma barra (2) a qual se deseja alterar as propriedades
RobotBar viga = (RobotBar) estrutura.Bars.Get(2);

// Configurando o tipo estrutural da barra como pilar
Viga.StructuralType = IRobotObjectStructuralType.I_OST_COLUMN;

// Configurando o tipo estrutural da barra como viga
Viga.StructuralType = IRobotObjectStructuralType.I_OST_BEAM;

// Configurando o tipo de parâmetros de armadura para o elemento
Viga.SetLabel(IRobotLabelType.I_LT_MEMBER_REINFORCEMENT_PARAMS, "standard");

// Configurando o tipo de membro estrutural
Viga.SetLabel(IRobotLabelType.I_LT_MEMBER_TYPE, "Viga de CA");
```

Figura 4.8 - Exemplo de configuração dos elementos para o dimensionamento.

Cabe ressaltar que os valores “*standard*” e “*Viga de CA*” dizem respeito, respectivamente, aos “Parâmetros de norma” e “Parâmetros de cálculo”, a serem configurados manualmente na interface do *Robot*, na versão em português, através do menu “*Projeto > Armadura Necessária de Vigas/Colunas – Opções*”. Estes parâmetros controlam as configurações de armadura e modelo estrutural de cada um dos tipos de elementos de viga e pilares. É importante ressaltar, também, que todos os elementos devem ter os parâmetros definidos de acordo com uma mesma norma, do contrário o *Robot* não realizará os cálculos e emitirá uma mensagem de erro. Configurados os elementos, o cálculo da armadura necessária teórica para os elementos pode ser invocado utilizando o objeto da classe *RConcrCalcEngine*, acessível através do objeto gerenciador de projeto *RobotProject*. Através dos procedimentos apresentados na Figura 4.9, pode-se definir os membros envolvidos no dimensionamento, os casos de carregamento e a quantidade de pontos amostrais para o cálculo das armaduras ao longo do elemento.

```

// Instancia do motor de cálculo de armadura necessária
RConcrCalcEngine RCEngine = projeto.ConcrReinfEngine;
// Definição do cálculo de armadura necessária para membros (barras)
RConcrMemberRequiredReinfEngine memberEngine = RCEngine.MemberRequiredReinf;
// Definição da quantidade de pontos amostrais por número de pontos
memberEngine.Params.BeamPointsType =
    IRCOncrBeamCalcPointDefinitionType.I_CBCPDT_POINT_COUNT;
// Definindo a quantidade de pontos igual a 7
memberEngine.Params.BeamPointsNumber = 7;
// Definindo os casos de carregamento do Estado Limite Último (caso 1 de cargas)
memberEngine.Params.CasesULS.FromText("1");
// Definindo os casos de carregamento do Estado Limite Serviço (caso 2 de cargas)
memberEngine.Params.CasesSLS.FromText("2");
// Definindo os membros considerados no dimensionamento. Para o elemento 2...
memberEngine.Params.Members.FromText("2");
// Executando os cálculos
memberEngine.Calculate();

```

Figura 4.9 - Exemplo da configuração do motor de cálculo de armadura necessária.

Para ter acesso aos resultados das armaduras requeridas em cada um dos pontos calculados, para cada elemento calculado, é necessário realizar um procedimento semelhante aos apresentados na Figura 4.7 para os momentos e deslocamentos. Todavia, as requisições dos valores de armaduras não são acessadas por funções específicas como no caso dos deslocamentos e momentos, eles são acessados de forma indireta. Até a versão da API que acompanha a versão 2016 do *Robot*, não existiam funções para obtenção direta das armaduras nos membros, fonte de grande dificuldade por parte do autor na busca de como fazê-lo.

A seguir, na Figura 4.10, são apresentados a obtenção do valor da área de armadura teórica requerida por um membro através de um procedimento genérico de obtenção de resultados após a realização dos cálculos.

```

// Armadura da parte superior dos membros
estrutura.Results.Any.ResultId = 1160; // T_BEST_CALC_POINTS = 1160;
// Definindo a barra a qual deseja conhecer a armadura requerida
estrutura.Results.Any.Bar = 2; // selecionando o membro 2
// Definindo o ponto no qual se deseja conhecer os resultados
estrutura.Results.Any.CalcPoint = 3; // Resultados para o 3 ponto de cálculo
// Definindo qual o tipo de resultado se desenha obter
estrutura.Results.Any.ResultId = 919; //919 = código da armadura inferior requerida
// Obtendo o valor do tipo de resultado indicado acima
double armInf = estrutura.Results.Any.ResultValue;
// Definindo qual o tipo de resultado se desenha obter
estrutura.Results.Any.ResultId = 920; //920 = código da armadura superior requerida
// Obtendo o valor do tipo de resultado indicado acima
double armSup = estrutura.Results.Any.ResultValue;

```

Figura 4.10 - Exemplo da obtenção dos valores de armadura requeridas.

Pode-se observar que para a obtenção dos valores das armaduras foi necessário o informar para qual elemento, em que ponto da discretização especificada na Figura 4.9, qual o tipo do resultado desejado (por exemplo, armadura superior ou inferior), para só em seguida requisitar o valor desejado. Além disso, observa-se que o tipo de valor que se deseja obter é informado por um código em forma de um número inteiro. Os valores “919” e “920”, referentes as propriedades de armadura inferior e superior, respectivamente, apresentados no exemplo da Figura 4.10 foram obtidos por tentativa e erro através de exemplos observados em fóruns de discussão sobre o assunto na internet. Outros valores que possam ser importantes são apresentados na Tabela 4.1.

Descrição	Código
Armadura Longitudinal Interior requerida	919
Armadura Longitudinal Superior requerida	920
Armadura Longitudinal Interior fornecida	1145
Armadura Longitudinal Superior fornecida	1144
Área mínima de armadura longitudinal permitida par ao membro	1144
Armadura transversal requerida	1150
Espaçamento da armadura transversal requerida	1201
Taxa de armadura longitudinal	0921

Tabela 4.1 - Códigos para a obtenção dos valores de armadura.

4.1.2.6. Quantitativos de materiais

A avaliação da quantidade de material utilizada com os elementos estruturais foi obtida utilizando uma função nativa da API. Um exemplo do levantamento do volume e peso de concreto do tipo “C25” (criado na interface do *Robot*) utilizando nos elementos é apresentado na Figura 4.11.

```
// Volume de concreto dos membros
double volume =
estrutura.QuantitySurvey.Materials.GetVolume(IRobotObjectType.I_OT_BAR, "C25");

// Peso dos membros de concreto
double peso =
estrutura.QuantitySurvey.Materials.GetWeight(IRobotObjectType.I_OT_BAR, "C25");
```

Figura 4.11 - Exemplo do levantamento de quantitativos.

4.2. MÓDULO *NSGA-II*

O módulo *NSGA-II* é essencialmente composto pelas classes da biblioteca *jMetal*, mais especificamente das partes referentes ao algoritmo *NSGA-II*. As classes disponibilizadas pela *jMetal* foram mantidas praticamente inalteradas, com exceção da classe *NSGA-II*, que foi alterada de modo a permitir através do módulo *PainelControle* o acompanhamento da quantidade. A classe que contém o módulo *NSGA-II* apresenta um único método, *Execute()*, responsável pela execução do algoritmo e suas etapas, como geração de indivíduos e populações, cruzamento, mutação e seleção. Entretanto, sua chamada se dá no módulo *Otimizacao*, responsável por configurar o processo de otimização.

A versão do *NSGA-II* presente na biblioteca *jMetal* e utilizada neste trabalho é a mesma que fora utilizada no trabalho de [Nebro et al. \(2008\)](#), que faz uso de indicadores de qualidade como o hipervolume da fronteira de Pareto de uma iteração para avaliar a velocidade e convergência do algoritmo.

A definição das variáveis e seus limites, funções objetivos, tamanho da população e número máximo de iterações, não são definidos neste módulo, sendo apresentadas no módulo *Otimizacao* e *Modelo*.

4.3. MÓDULO *Otimizacao*

Este módulo é o responsável por configurar e executar o processo de otimização uma vez que o modelo de otimização esteja configurado. A sua definição é realizada através da classe *Otimizacao.cs*, que é bastante semelhante às implementações dos códigos-exemplo distribuídos junto à *jMetal*, que, principalmente, implementa a classe *Problem*, da *jMetal*.

É neste módulo que se dá a definição do problema de otimização, tamanho da população, quantidade máxima de gerações, taxa de probabilidade de mutação, cruzamento e seleção. Além disso, é neste módulo que o módulo do *NSGA-II* é invocado e também onde os resultados são processados.

4.4. MÓDULO *Modelo*

É este módulo que reúne as informações do modelo a ser utilizado no processo de otimização. É por meio desse módulo que as variáveis do modelo são definidas, seus limites estabelecidos e as funções objetivo avaliadas. O mesmo módulo tem conexão direta com o módulo *RobotConexao*, pois os parâmetros aqui definidos, como a chamada das funções objetivos em cada iteração, utilizam das funções presentes no módulo *RobotConexao*. Além disso, o *Modelo* tem fundamental importância na conexão com o módulo *Otimizacao*, pois os intervalos limite das variáveis daqui definidos são utilizados pelo *Otimizacao*.

Estão presentes métodos para a definição nos limites das variáveis, informados como vetores unidimensionais de números reais; um outro método para a avaliação das funções objetivo, que tem como entrada um vetor unidimensional de números reais contendo os valores das variáveis em cada iteração, e produz como saída também um vetor de números reais unidimensional, onde cada entrada representa o valor de uma função objetivo para uma dada iteração; de forma semelhante, há um método de funcionamento similar ao das funções objetivos, porém, para a avaliação das funções de restrição, no caso em que houver.

4.5. MÓDULO *PainelControle*

O módulo *PainelControle* representa o Painel de Controle do programa desenvolvido e é responsável pela interação do usuário com o processo de otimização. Consiste, basicamente, em uma interface amigável para a definição do modelo de otimização e configuração dos parâmetros do algoritmo NSGA-II.

A interface gráfica foi implementada utilizando a *framework* Windows Presentation Foundation (WPF), implementada utilizando a linguagem XAML, todavia, as funções executadas quando ocorre interação com os elementos da interface é implementada em *C#*. A tela inicial do módulo

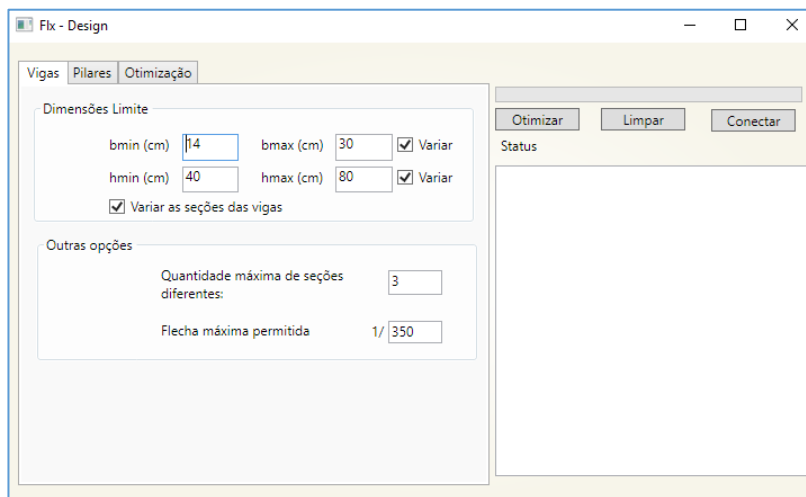


Figura 4.12.

Na Figura 4.12 ainda é possível ver que há 3 abas na parte superior: Vigas, Pilares e Otimização. Na aba Vigas estão presentes as opções referente as vigas no processo de otimização, como as dimensões mínimas e máximas permitidas, se a variação das seções deve ser considerada na otimização, a quantidade máxima de seções diferentes permitidas e a flecha máxima aceitável. De forma semelhante, a Figura 4.13 apresenta as opções disponíveis da aba Pilares.

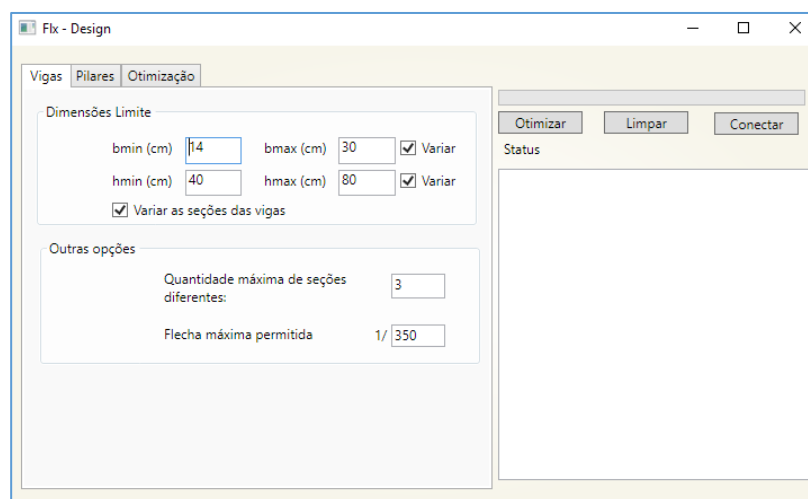


Figura 4.12 - Tela inicial do Painel de Controle na opção das vigas.

Nesta aba estão presentes opções para o fornecimento das dimensões mínimas a serem consideradas na otimização dos pilares, bem como uma escolha quanto se a seção deve ser variada ao longo das iterações, quantidade máxima de seções diferentes permitidas para os pilares e quantidade de pilares a considerar no posicionamento. Além dessas opções, existe a

possibilidade de orientar a maior dimensão da seção transversal com o eixo da viga (por razões arquitetônicas), e se o posicionamento deve levar ou não em consideração a simetria.

Na aba *Otimização*, os parâmetros do processo de otimização podem ser configurados, como o tamanho da população por iteração, a quantidade máxima de iterações, o índice de distribuição da mutação, o índice de distribuição e probabilidade de cruzamento (*crossover*), conforme apresentado na Figura 4.14.

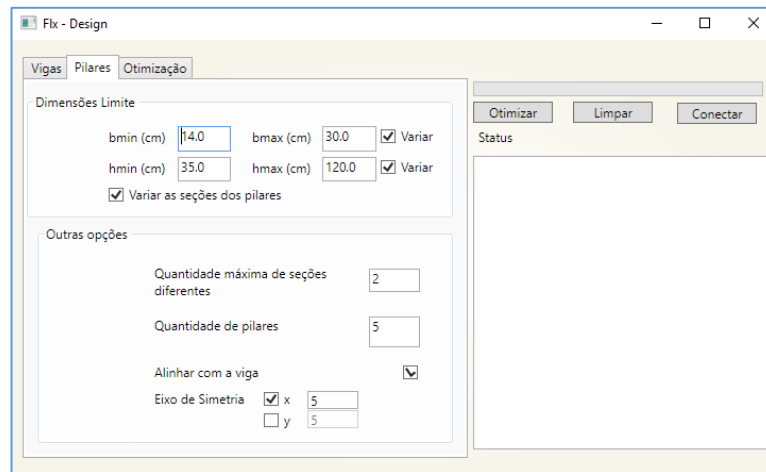


Figura 4.13 - Tela do Painel de Controle nas opções dos pilares.

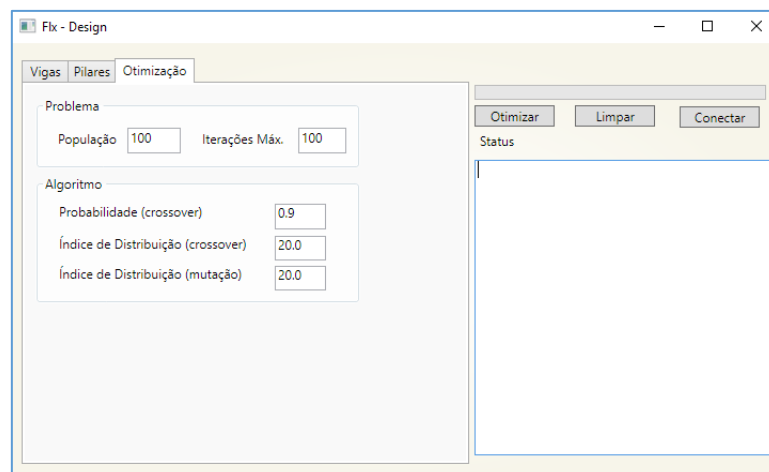


Figura 4.14 - Tela do Painel de Controle na aba *Otimização*.

Ao ser inicializado, o módulo *PainelControle* automaticamente estabelece a conexão com o *Robot*, necessitando o mesmo já estar em execução no momento da tentativa de conexão. A qualquer alteração do modelo na interface do *Robot*, o usuário poderá solicitar a atualização das informações do modelo através do botão *Conectar*. Para iniciar o processo de otimização, o botão *Otimizar* deverá ser pressionado. Neste momento, o módulo *Flx* será invocado, e será

responsável pela coleta dos dados da interface e comunicação com os demais módulos, como apresentado mais adiante.

Após o início da execução, na janela *Status*, serão registrados os valores das variáveis, bem como os valores das funções objetivos e o tempo despendido em cada iteração, podendo o usuário pressionar o botão *Limpar* para apagar os registros. Automaticamente, ao fim do processo de otimização, serão gerados arquivos de valores separados por vírgula no formato *.csv*, com os dados de cada iteração, organizados em arquivos separados para as variáveis e objetivos, bem como para a solução final, de modo que possam ser facilmente manipulados e pós-processados em outras plataformas.

Os módulos de interação são descritos detalhadamente na Figura 3.13, a saber, *PainelControle*, *Flx*, *RobotConexao*, *Modelo*, *Otimizacao* e *NSGA-II*, e como se dá a interação entre esses.

4.6. MÓDULO *Flx*

O módulo *Flx* é o responsável pela interoperabilidade entre os demais módulos, como pode ser observado na Figura 3.13. O *Flx* é invocado no momento em que o botão *Calcular* (Figura 4.14) é pressionado. Neste momento, é realizada a leitura e interpretação de todos os dados fornecidos na interface gráfica (*PainelControle*), de modo que, por exemplo, possam ser definidos os limites das variáveis (*Modelo*), a composição da viga equivalente dos pavimentos (Figura 3.3, *RobotConexao*), preparação do problema de otimização (tamanho da população e quantidade máxima de iterações, *Otimizacao*), bem como a definição dos parâmetros do *NSGA-II* (módulo *NSGA-II*, via *Otimizacao*).

5. RESULTADOS

Neste capítulo são apresentados exemplos para o emprego da metodologia proposta e do programa criado. Em todos os casos a geometria dos pavimentos, através do posicionamento das vigas e lajes, foi pré-determinada na interface gráfica do *Robot*, utilizando suas funções nativas, assim como os tipos, casos e combinações de carregamento da estrutura, tipo e propriedades dos materiais. Em todos os exemplos não foi contemplado o dimensionamento das lajes, sendo elas inseridas no modelo com o único intuito de transferir as cargas a ela impostas para as vigas. Para os casos onde o dimensionamento deverá ser realizado, é necessário também definir qual norma será considerada no dimensionamento e as seus parâmetros configurados, como as comentadas no fim do item 4.1.2.5. É importante também ressaltar que os exemplos e casos apresentados buscam alternativas de projeto para as estruturas propostas.

5.1. EXEMPLO 01

Para os exemplos de 1 a 4 será utilizado um edifício de 3 pavimentos, com a geometria dos pavimentos apresentada na Figura 5.1. A distância entre os pavimentos foi adotada igual a 3 m.

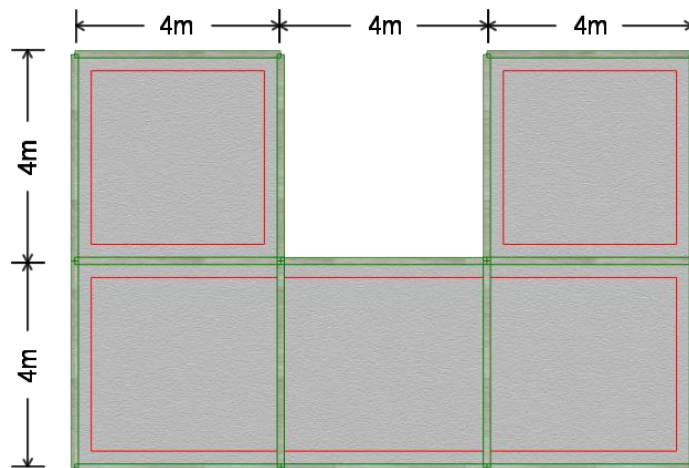


Figura 5.1 - Forma do pavimento do edifício do exemplo 01.

Foi considerado que sobre cada uma das vigas atuará uma carga uniformemente distribuída de $5,85 \text{ kN.m}^{-1}$, equivalente a uma alvenaria de 14 cm de espessura e 3 m de altura em tijolo cerâmico furado. Além disso, uma carga uniformemente distribuída de 3 kN.m^{-2} foi aplicada

nos painéis das lajes, equivalente à ocupação residenciais e o acréscimo de cargas de revestimentos. O peso próprio dos elementos também foi considerado.

No tocante ao posicionamento dos pilares, devido à geometria do pavimento apresentar um plano de simetria, o mesmo foi considerado. Optou-se por essa escolha de modo a diminuir a complexidade do problema e acelerar sua convergência. Desta forma, o posicionamento dos pilares fica limitado a uma das porções em relação ao plano de simetria, sendo a outra porção apenas simétrica à primeira. Assim, deve-se ter o cuidado de perceber que, por exemplo, ao escolher que 12 pilares devem ser posicionados, apenas metade desse valor deverá ser informado, isto é, 6 pilares, uma vez que a outra metade deverá respeitar a simetria, conforme será realizado neste exemplo.

Os parâmetros no NSGA-II (apresentados no item 3.3) foram adotados conforme o utilizado no trabalho de Babaei (2016). Foram utilizados 100 indivíduos (exceto na primeira iteração, onde 200 indivíduos são analisados para aumentar a diversidade inicial) e 100 interações em cada um dos casos. Os parâmetros do NSGA-II utilizados foram os pré-definidos, a saber: taxa de mutação de 0,50; probabilidade de mutação de 0,10; taxa de cruzamento de 0,80; e proporção de elitismo de 0,50, isto é, mantimento de metade da população i na iteração $i+1$.

Neste caso só foi permitida única seção para todas as vigas da edificação, tendo sido fixada a largura a 14 cm, de modo a garantir o embutimento nas alvenarias, e sua altura variada ao longo das iterações entre 30 e 70 cm. O valor máximo de 70 cm para altura das seções foi adotado por ser o valor que representa o limite de que a maior dimensão da seção transversal não deve exceder 5 vezes a sua menor dimensão.

De forma semelhante, apenas uma seção foi considerada para todos os pilares. Pelo mesmo motivo que nas vigas, a menor dimensão da seção transversal foi adotada como sendo de 14 cm. A maior dimensão foi considerada variável entre 30 e 50 cm, sendo esses limites definidos de forma arbitrária, de forma a simular uma preferência de um projetista. Assim, este exemplo apresenta 14 variáveis de decisão: 12 referentes ao posicionamento dos pilares, 1 para a altura das seções das vigas e 1 para a maior dimensão da seção dos pilares.

As funções objetivos escolhidas foram o custo com materiais (concreto, aço e formas) *versus* o deslocamento máximo vertical no pavimento. A única restrição imposta foi a que o deslocamento máximo em cada viga não superasse a razão entre o seu comprimento e 350, que é um dos limites estabelecidos na norma brasileira NBR 6118/2014, para estruturas de concreto

armado. A limitação do deslocamento pode ser configurada no próprio *Robot* que, ao ser violada, emite-se um erro no dimensionamento, sendo o projeto que a gerou penalizado.

Com os parâmetros mencionados anteriormente, após 100 iterações foi obtido o conjunto de soluções apresentadas na Figura 5.2, em que o gráfico mostra a Fronteira de Pareto considerando os deslocamentos em centímetros e o custo em milhares de reais.

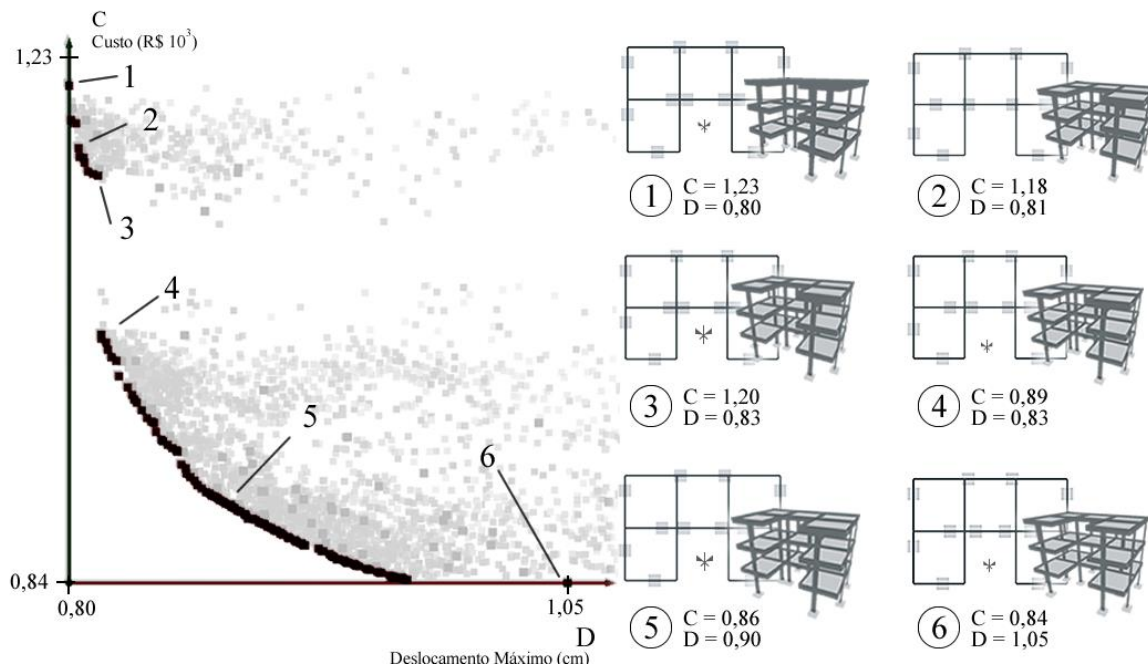


Figura 5.2 - Soluções do exemplo 01.

Os valores apresentados junto às soluções da Figura 5.2 dizem respeito a relação entre os valores obtidos para as funções objetivos em cada uma das soluções e o valor obtido para um projeto padrão de referência. Para os exemplos 01, 02, 03 e 04 deste capítulo, o modelo de referência foi aquele considerando que os pilares são posicionados nas interseções entre as vigas do pavimento analisado. Desde modo, os valores menores do que 1 para as funções objetivos apresentam soluções melhores se comparadas com a obtida considerando os pilares posicionados nas interseções das vigas. De forma semelhante, os valores superiores à 1 são soluções menos eficientes ao obtido considerando os pilares nas interseções.

Ao observar a Figura 5.2, pode-se perceber uma descontinuidade na Fronteira de Pareto para este caso. Após inspeção nos indivíduos das soluções 3 e 4 destacadas na figura, averiguou-se que embora as topologias deles sejam semelhantes, a diferença está na altura das vigas de cada caso. O indivíduo 3 possui vigas de 61 cm de altura, enquanto o indivíduo 4 possui 60 cm de altura. Essa pequena variação na altura das vigas ocasiona uma grande diferença entre o custo

dos dois grupos. Isto ocorre devido vigas com altura maior do que 60 cm necessitarem da utilização de armadura de pele, ocasionando em um aumento considerável de armadura, evidenciada pela descontinuidade da Fronteira de Pareto obtida. O hipervolume para este caso é apresentado na Figura 5.3.

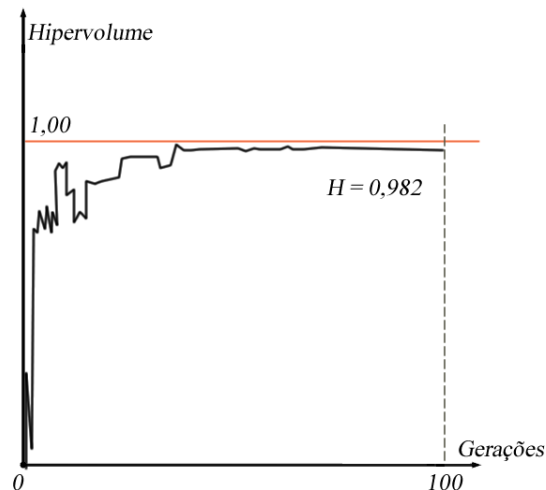


Figura 5.3 - Evolução do hipervolume ao longo de 100 gerações.

Além disso, pode-se observar que na parte inferior da Fronteira de Pareto, referente às soluções com vigas com menos de 60 cm, que pequenas variações do custo (resultando em economia), provoca um aumento proporcionalmente maior nos deslocamentos, enquanto a parte superior apresenta uma variação praticamente proporcional entre deslocamentos máximos e custo.

Quanto ao posicionamento dos pilares, observa-se que a diferença entre as soluções 3 e 4 da Figura 5.2 é praticamente nula, fato que foi também observado nas demais soluções que compõem a Fronteira de Pareto da Figura 5.2.

Conclui-se que essa formulação do problema não foi eficiente na produção de alternativas no que diz respeito ao posicionamento dos pilares, ainda que tenha produzido soluções que apresentem alturas de viga que variam de 70 a 50 cm, respectivamente para as soluções 1 e 6 da Figura 5.2. Esse comportamento pode não ser útil em auxiliar o projetista no que diz respeito ao posicionamento dos pilares, ainda que produza alternativas que possam auxiliar na escolha das seções.

5.2. EXEMPLO 02

De modo a verificar se a variação das alturas de vigas e pilares influenciaram na falta de diversidade das soluções do Exemplo 01, no que diz respeito ao posicionamento dos pilares, o Exemplo 02 apresenta o mesmo problema do Exemplo 01, porém considerando que as seções dos elementos são pré-estabelecidas e invariáveis ao longo do processo de otimização. Desta forma, a altura das vigas foi mantida igual a 50 cm e a maior dimensão dos pilares em 40 cm. Assim, as partes do custo referentes ao concreto e formas ficarão fixas, podendo variar apenas o custo com armadura. A Figura 3.1 apresenta os resultados referentes a este caso.

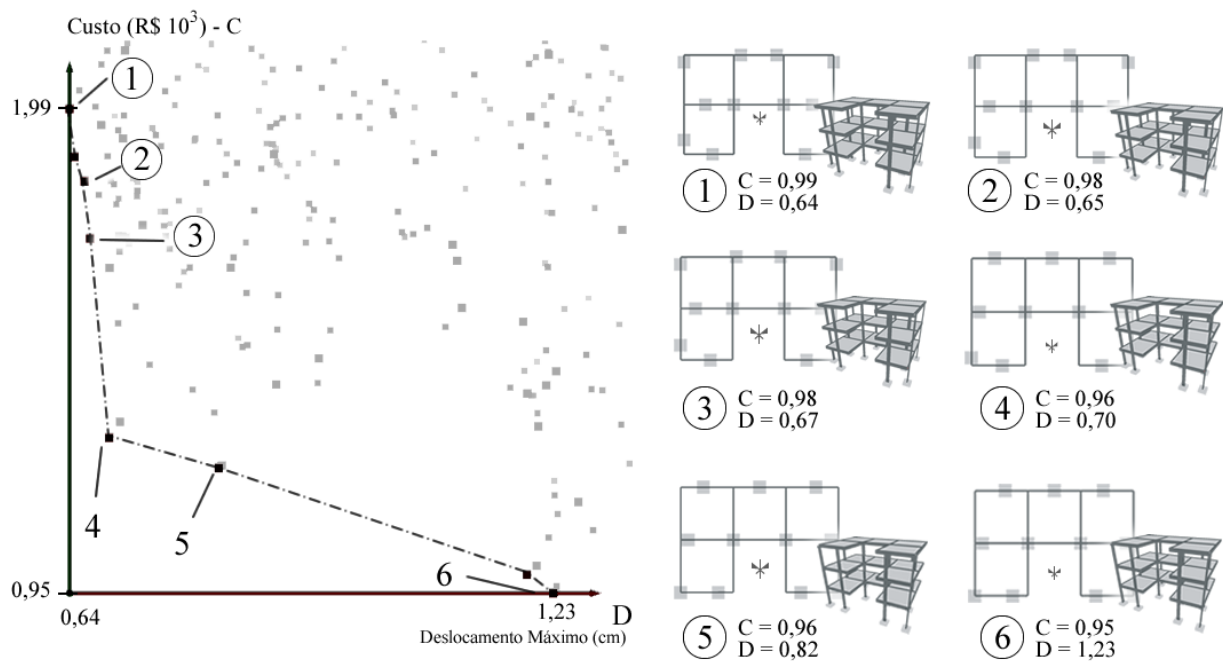


Figura 5.4 - Soluções do exemplo 02.

Com base na Figura 5.4 é possível perceber que a fixação das dimensões das vigas e pilares não produziu significativas alterações na topologia da estrutura, apresentando um posicionamento dos pilares bastante semelhante aos do Exemplo 01 (Figura 5.2).

5.3. EXEMPLO 03

A partir da experiência com o Exemplo 02, no Exemplo 03 é apresentada a consideração do momento máximo absoluto como uma das funções objetivos, em substituição ao custo da estrutura. Dessa forma, a otimização é realizada buscando a minimização do momento fletor absoluto nas vigas, em conjunto com o deslocamento máximo. Uma vez que a armadura longitudinal é função dos momentos fletores das vigas, acredita-se que minimizando o

momento fletor absoluto máximo da estrutura, esteja-se indiretamente limitando a área de aço máxima requerida nas vigas.

Os resultados apresentados na Figura 5.5 foram obtidos considerando a altura das vigas e maior dimensão dos pilares constantes e iguais a 50 cm, respectivamente, e variando-se a posição dos pilares.

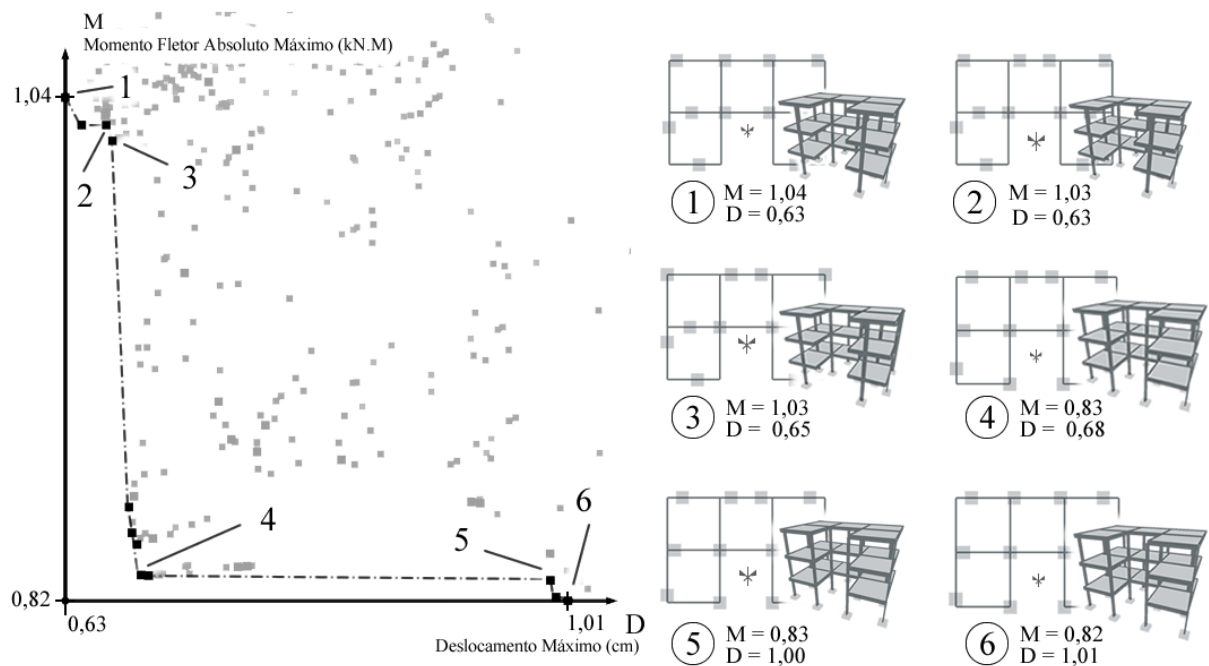
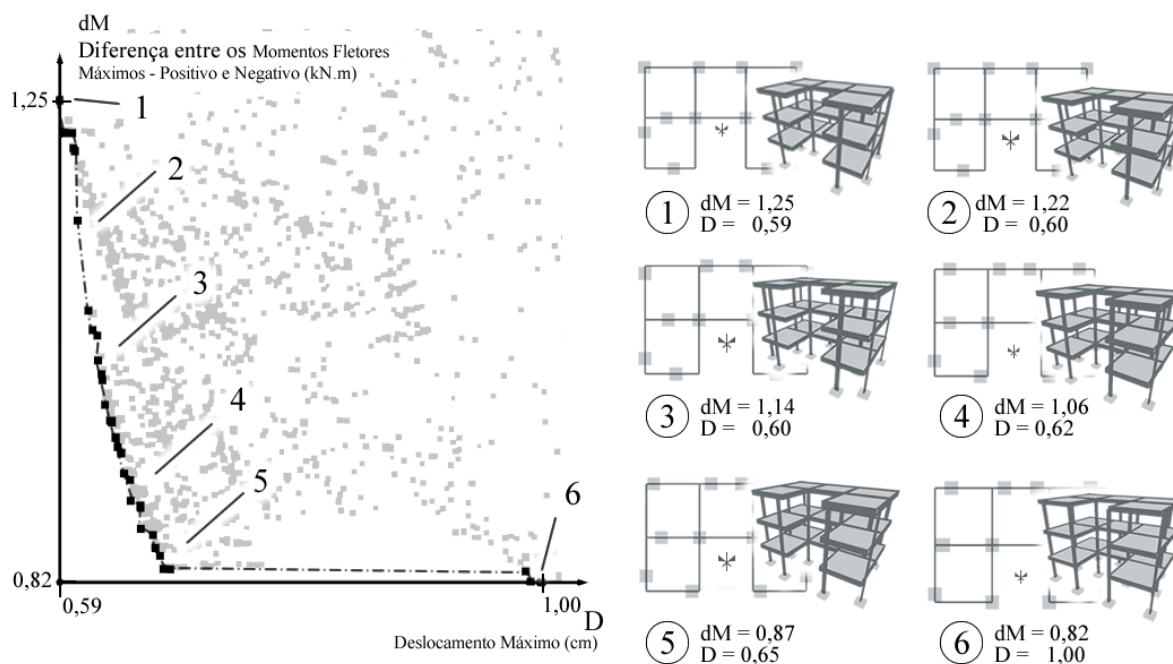


Figura 5.5 - Soluções do exemplo 03.

É possível perceber que não foi possível encontrar uma Fronteira de Pareto bem definida para o problema, sendo ela aproximada pela linha pontilhada no gráfico da Figura 5.5. As soluções não-dominadas neste problema estão nitidamente agrupadas em 3 regiões distintas. Na primeira região, onde estão destacadas as soluções de 1 a 3, tem-se soluções de topologia semelhante, com uma leve variação dos deslocamentos máximos obtidos, praticamente irrelevantes em situações práticas. Na região onde se encontra a solução 4, é possível notar uma mudança mais significativa na topologia da estrutura se comparadas com as soluções da primeira região, além da diminuição considerável do momento fletor máximo absoluto. As soluções 5 e 6 estariam na 3 região do conjunto de soluções, apresentando topologias mais próximas à solução, porém com deslocamentos consideravelmente maiores do que a solução 4. A provável causa dessa considerável diferença entre os deslocamentos esteja no posicionamento mais afastados entre pilares adjacentes na solução 6, se comparado as demais.

5.4. EXEMPLO 04

Neste último caso, buscou-se utilizar um critério semelhante ao utilizado por Kripka (1998). Além do deslocamento máximo nas vigas, foi adotada como função objetivo a diferença entre os momentos positivos e negativos máximos, de modo a se buscar uma uniformidade nos valores dos momentos fletores na estrutura, variando as seções das vigas, que tem influência direta nos deslocamentos. Os resultados para esse caso são apresentados na Figura 5.6.



Indivíduo	1	2	3	4	5	6
h (cm)	51	49	46	50	49	40

Figura 5.6 - Resultados para o Exemplo 04.

Ao observar a Figura 5.6 é possível concluir que grandes variações da diferença entre os momentos resultam em variações discretas nos deslocamentos, com exceção do indivíduo 6. O indivíduo 6 apresenta a menor seção entre as apresentadas na Figura 5.6, o que certamente influenciou na discrepância do seu deslocamento em relação aos demais indivíduos da solução.

É importante destacar que para este caso, as soluções diferiram mais entre si no que diz respeito ao posicionamento dos pilares, sendo o caso 6, possivelmente, uma solução próxima ao que seria proposto por um projetista em um projeto cotidiano. A evolução do hipervolume para a otimização do Exemplo 04 é a apresentada na Figura 5.7.

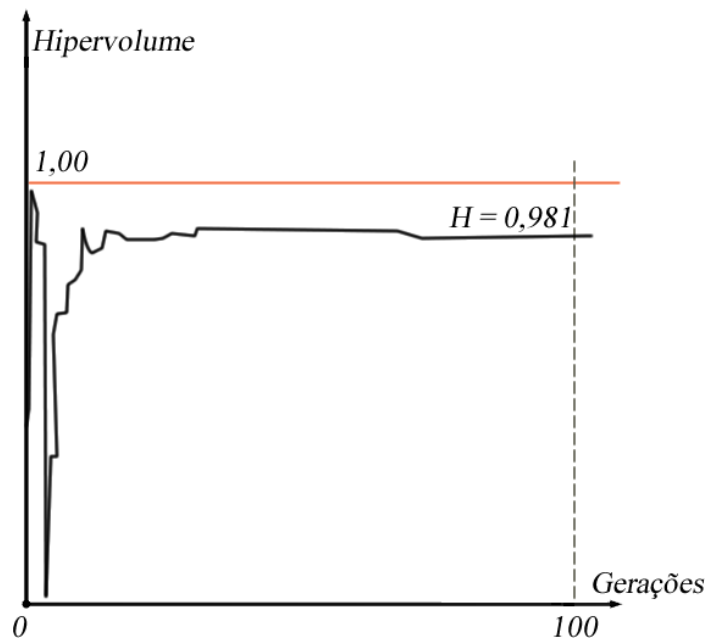


Figura 5.7 - Evolução do hipervolume ao longo de 100 gerações.

5.5. EXEMPLO 05

O exemplo 05 tem como objetivo mostrar a versatilidade da metodologia proposta no que diz respeito as formas do pavimento em que pode ser empregada. Nos trabalhos presentes na literatura, como mostrado no capítulo 2, as metodologias para a busca do posicionamento ótimo dos pilares só são empregadas em estruturas de topologia regular, frequentemente com pavimentos com as vigas de espaçamento regular e perpendiculares entre si. A partir deste exemplo se pretende mostrar que a metodologia é capaz de lidar com pavimentos de formas não regulares, como, por exemplo, contendo elementos não-ortogonais e até vigas curvas.

A forma do pavimento tipo da estrutura é apresentada na figura, sendo o espaçamento entre pisos igual a 3 m. Foi considerada uma carga uniformemente distribuída em todas as lajes igual a $3,75 \text{ kN.m}^{-2}$, e sobre as vigas, uma carga uniformemente distribuída de $5,85 \text{ kN.m}^{-1}$, referente às alvenarias, conforme o exemplo 01.

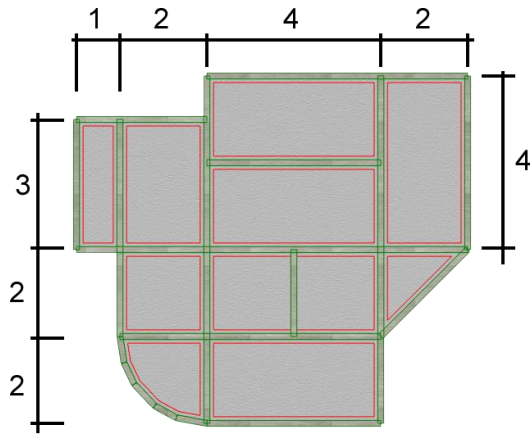


Figura 5.8 - Formas do pavimento tipo do exemplo 05.

De forma semelhante ao exemplo 04, as funções objetivos adotadas foram a diferença entre o momento positivo e negativo máximos, e o deslocamento máximo das vigas. Foi escolhido utilizar 16 pilares no lançamento. Além disso, a altura das vigas foi considerada variável ao longo das iterações, com intervalos semelhantes aos do exemplo 01.

Optou-se por alterar os parâmetros do algoritmo, de modo a avaliar sua influência nos resultados. Assim, para o Exemplo 05, foi considerada uma probabilidade de mutação de 0,20 (20%), diferentemente dos 0,10 (10%) adotados em todo o exemplo 01. A proporção do elitismo foi mantida fixa em 0,50 (50%). Os resultados para este caso são apresentados na Figura 5.9.

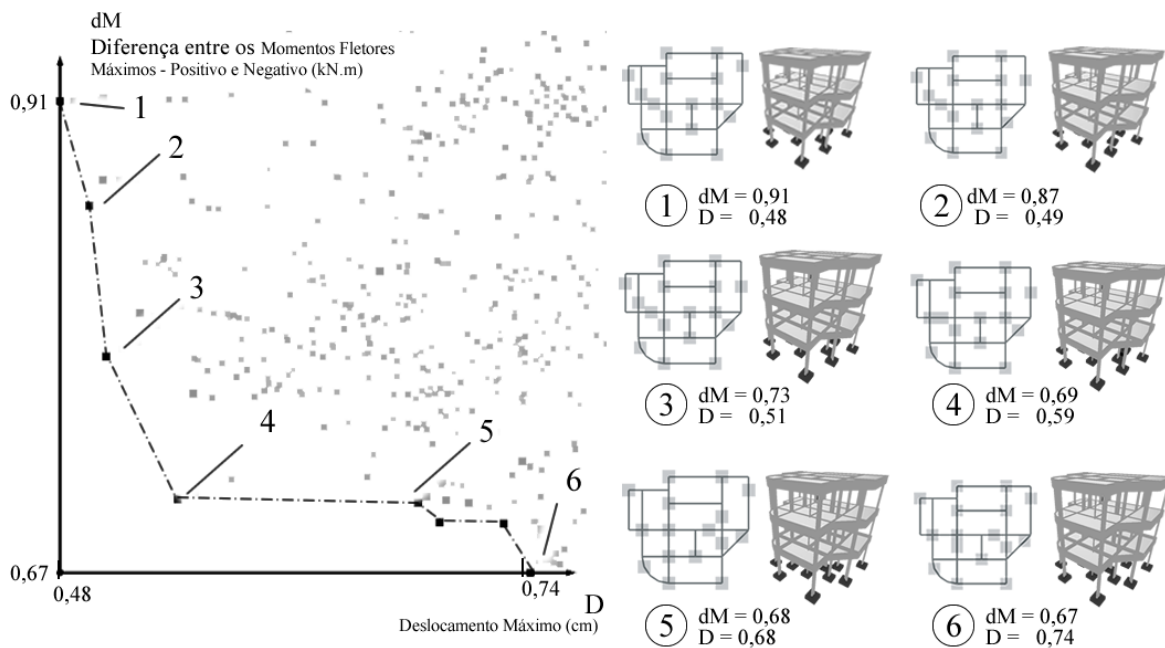


Figura 5.9 - Resultados do Exemplo 05.

Assim como nos exemplos 01 até 04, os valores apresentados para o exemplo 05 e 06 são referentes à um projeto padrão de referência. O projeto padrão de referência adotado para os exemplos 05 e 06 está apresentado em planta na Figura 5.10.

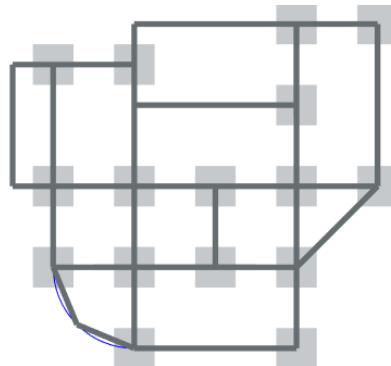


Figura 5.10 - Modelo padrão de referência para exemplos 05 e 06.

Observa-se na Figura 5.9, que a mudança nos parâmetros de mutação não proporcionou um aumento da diversidade dos resultados, principalmente no tocante ao posicionamento dos pilares, produzindo resultados muito semelhantes entre si. Entretanto, considera-se que as soluções obtidas pela metodologia produziram resultados exequíveis com pequenas alterações práticas, desde que não haja limitação quanto ao posicionamento dos pilares. A evolução do hipervolume para o exemplo 05 pode ser observado na Figura 5.11 - Evolução do hipervolume para o Exemplo 05 ao longo de 100 gerações.

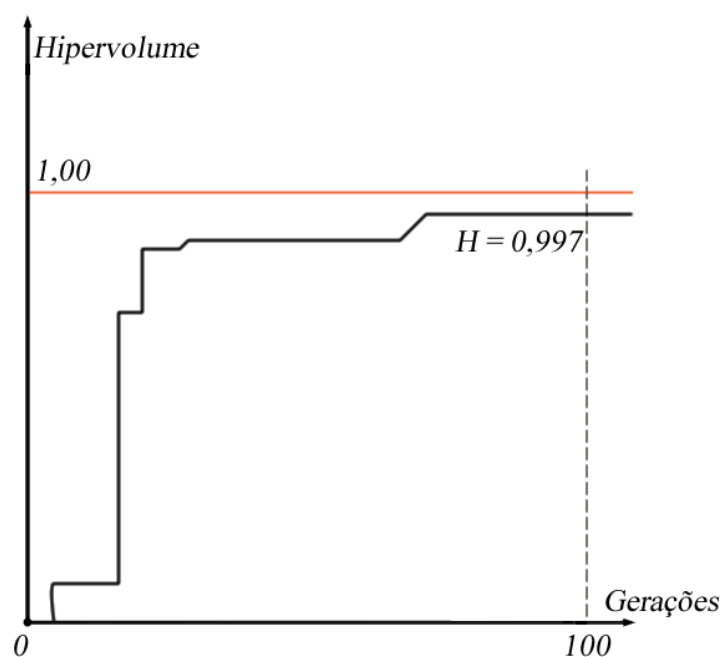


Figura 5.11 - Evolução do hipervolume para o Exemplo 05 ao longo de 100 gerações.

5.6. EXEMPLO 06

O Exemplo 06 compartilha de todas as características do Exemplo 05, diferenciando apenas nas taxas de mutação e cruzamento, sendo neste presente exemplo 0,10 (10%) e 0,90 (90%), sendo respectivamente, o elitismo mantido fixo e igual a 0,50 (50%). Os resultados obtidos são apresentados na Figura 5.12.

Ao observar os resultados da Figura 5.12 é possível perceber que os valores de deslocamentos e diferença entre momentos obtidos foram, em geral, maiores do que no exemplo 05. A influência da taxa de mutação pode ter influenciado nesse resultado, uma vez que ela tende a aumentar o tempo até a convergência. Isso ocorre porque, devido a maior mutação, os indivíduos resultantes de cruzamento, por mais que herdem as características de seus pais, tendem a apresentar variações nas mesmas, tornando-os mais diferentes. Essa diferença pode ser positiva, no sentido de diversidade, mas prejudicial para a convergência do problema.

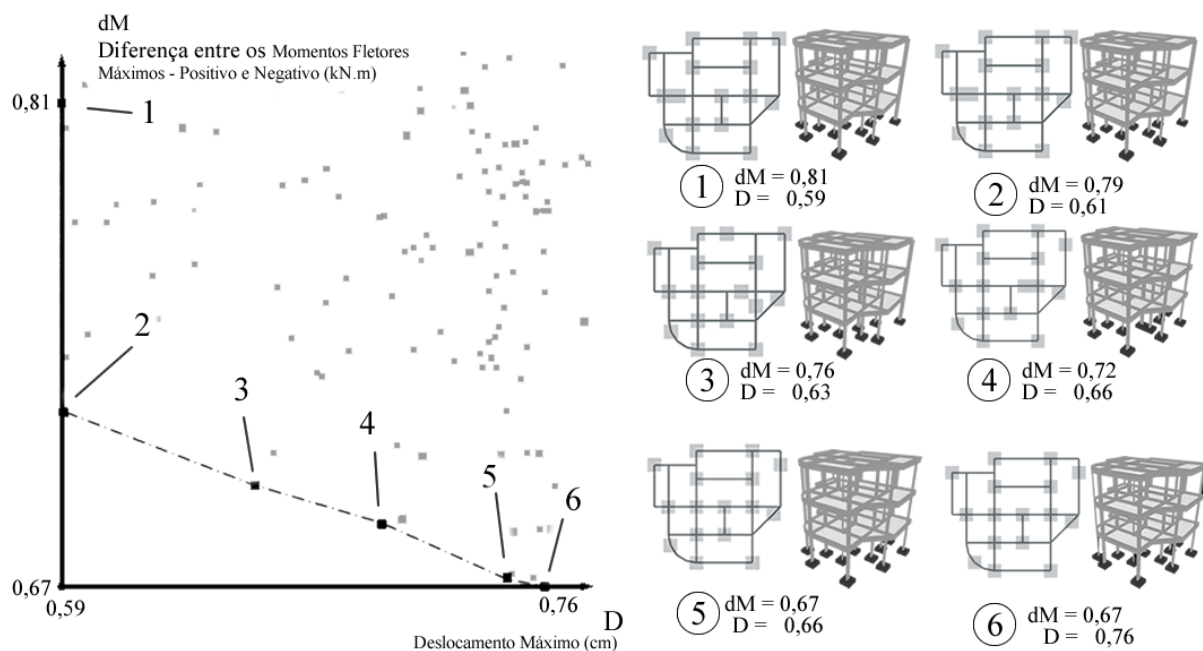


Figura 5.12 - Resultados do Exemplo 06.

Pode-se observar que os resultados obtidos neste exemplo, no que diz respeito ao posicionamento dos pilares, produziu resultados semelhantes aos comumente adotados no cotidiano pelos projetistas: pilares posicionados nas interseções dos elementos. Baseado na

complexidade geométrica desse exemplo, observa-se que a metodologia é capaz de produzir resultados exequíveis e possíveis, podendo auxiliar na proposta de soluções. Isto é um fator muito importante principalmente para os projetistas com menor experiência de projetos.

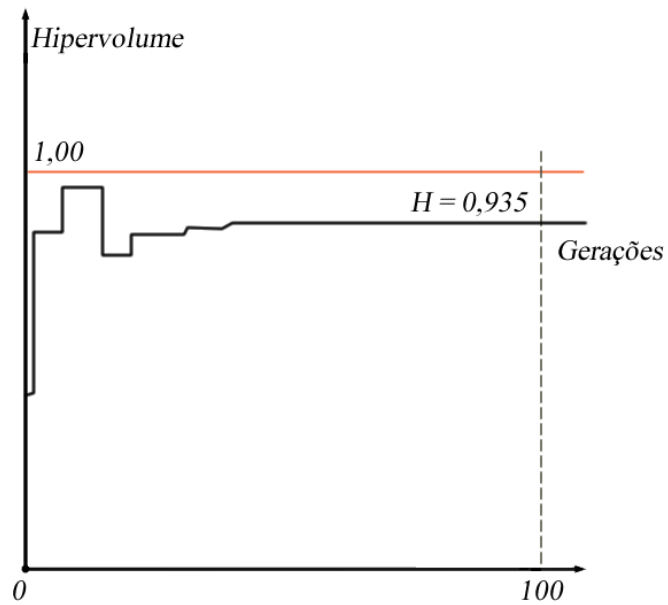


Figura 5.13 - Evolução do hipervolume para o Exemplo 06 ao longo de 100 gerações.

6. CONCLUSÕES

Neste trabalho, uma metodologia para a exploração do espaço de soluções aplicado ao projeto de estruturas aporticadas foi apresentado, especialmente para estruturas de concreto armado. Para tanto, desenvolveu-se um programa computacional que possibilitou a integração de um algoritmo de otimização multiobjetivo, o NSGA-II, com um programa de análise estrutural comercial, o *Autodesk Robot*. O programa foi composto por módulos cuja as funções possibilitaram desde o controle da geometria da estrutura no ambiente do *Robot*, quanto a configuração e execução do *NSGA-II*.

A partir da definição do posicionamento desejado para as vigas de um pavimento tipo de uma estrutura, definido pelo usuário, e definição de limites máximos e mínimos desejados para as seções dos elementos, a metodologia proposta buscou o melhor posicionamento dos pilares sobre as vigas do pavimento e as melhores dimensões para as seções transversais. Ao fim do processo de otimização um conjunto de soluções ótimas foi proposta ao usuário, produzindo diferentes alternativas de projeto para auxiliar o projetista.

Entre as contribuições propostas pelo presente trabalho se encontra uma alternativa ao posicionamento dos pilares. A metodologia proposta se restringe ao caso de estruturas aporticadas onde os pilares devem ser posicionados sobre as vigas, não sendo possível o posicionamento dos pilares diretamente sobre as lajes. A partir dessa simplificação, foi possível realizar a parametrização do posicionamento dos pilares em função dos comprimentos das vigas de um pavimento. Para tanto, foi proposto o conceito de *viga equivalente* de um pavimento, onde todas as vigas de um dado pavimento foram justapostas de modo a formar uma única viga "equivalente". Assim, foi possível a adoção de um sistema baseado em uma única variável para a definição do posicionamento de cada pilar.

A metodologia proposta para o posicionamento dos pilares apresentada neste trabalho é capaz de reduzir à metade a quantidade de variáveis necessárias para o posicionamento dos pilares em estruturas aporticadas com pilares posicionados sobre vigas, o que pode reduzir significativamente a quantidade de iterações necessárias à convergência do problema. Além disso, essa metodologia permite a busca pelo posicionamento dos pilares em estruturas onde a disposição das vigas do pavimento e, conseqüentemente, o posicionamento dos pilares não seja regular uniforme. Os trabalhos presentes na literatura, em sua maioria, contemplam o

posicionamento dos pilares em *grids* regularmente espaçados, o que pode ser um limitante para sua aplicação à maioria das estruturas usuais.

Uma outra contribuição deste trabalho está na apresentação de uma sistemática para a integração de um programa comercial de análise estrutural e um algoritmo de otimização. Esta integração pode contribuir para a utilização de técnicas de otimização na prática de elaboração de projeto cotidianos.

Os resultados obtidos neste trabalho sugerem que a metodologia tem um potencial de ser empregada como auxílio na concepção estrutural, principalmente no tocante ao posicionamento dos pilares em estruturas aporticadas. A escolha pela utilização de um algoritmo de otimização multiobjetivo permitiu a produção de um conjunto de soluções ótimas, ao invés de uma única solução, que podem servir de opções ao projetista na concepção estrutural.

É evidente que mais trabalhos precisam ser realizados de modo a aprimorar a metodologia proposta. Embora os resultados apresentem diversidade nos valores das funções objetivos, estes ainda apresentam uma baixa diversidade no posicionamento proposto dos pilares. Principalmente no Exemplo 1 do Capítulo 5, pode-se perceber pela Figura 5.2 que as soluções apresentam posicionamentos de pilares semelhantes, sendo a principal diferença entre as soluções ocasionadas pela variação das seções dos elementos. Entretanto, ao serem mantidas fixas as seções dos elementos, como apresentado no Exemplo 2, mantendo variável apenas o posicionamento dos pilares, observa-se uma maior diversidade no posicionamento dos pilares entre as soluções ótimas apresentadas, fato evidenciado na Figura 5.4. Estes resultados indicam uma sensibilidade do problema relacionadas à escolha das variáveis do problema, principalmente no que diz respeito ao posicionamento dos pilares, um dos objetivos deste trabalho.

A falta de diversidade topológica de algumas soluções pode ser proveniente de diversos fatores. Entre eles a escolha do algoritmo NSGA-II, sob o qual há relatos sobre a produção de soluções com baixa diversidade genética. Logo, a adoção de outros algoritmos deve ser investigada de modo que possa ser avaliada a performance de outras técnicas frente aos resultados apresentados.

Para os casos onde as armaduras foram levadas em consideração, foi considerado somente a quantidade de armadura teórica necessária. Entretanto, é sabido que em projetos usuais a quantidade de armadura necessária e a adotada após o detalhamento diferem significativamente.

Esta consideração pode acarretar em quantitativos de armadura que não se aproximam os valores reais utilizados ao fim do projeto. Portanto, seria necessário a consideração do detalhamento mais fiel das armaduras de forma que as soluções se aproximem mais do projeto final.

Por fim, ainda há aprimoramentos a serem realizados no que diz respeito à adoção das funções objetivos. No presente trabalho se percebeu uma sensibilidade dos resultados à adoção de diferentes funções objetivo. A não consideração de fatores construtivos, por exemplo, pode produzir soluções inexecutáveis, o que limitaria mais ainda a aplicação da metodologia proposta. Critérios relacionados às distâncias limites entre pilares, restrições ao posicionamento dos pilares (por exemplo, vãos de porta, janela, vagas de garagem, etc.), limitações dos balanços, entre outros, são alguns dos que poderiam ser melhor estudados.

6.1. SUGESTÕES PARA TRABALHOS FUTUROS

O autor considera que o presente trabalho se trata de um trabalho inicial que, a partir da metodologia apresentada, permite a exploração de outras possibilidades em trabalhos futuros e uma alternativa na exploração de soluções direcionadas a um futuro lançamento automático de estruturas. Desta forma, entre as principais sugestões para trabalhos futuros, destacam-se as seguintes:

1. Melhoria no processo de dimensionamento das estruturas: embora produza resultados de armadura requerida para os elementos estruturais, o *Robot* não é um programa dedicado a essa função. Desta forma, sugere-se um refinamento da forma como as armaduras são computadas e consideradas, se possível, com a implementação da NBR 6118.
2. Consideração do detalhamento das estruturas: é sabido que o detalhamento das armaduras é quem dita o que deverá ser executado, logo, a consideração do detalhamento das armaduras no levantamento dos quantitativos possibilitaria uma maior avaliação da quantidade de aço requerida por uma estrutura, resultando num custo mais fiel.
3. Um estudo sobre técnicas de melhoramento da diversidade em algoritmos multiobjetivo: como se pode perceber nos exemplos do Capítulo 5, embora a

metodologia produza alternativas, estas são muito semelhantes entre si. O ideal seria se obter uma Fronteira de Pareto que apresentasse maior diversidade e, verdadeiramente, aumentasse as alternativas propostas ao projetista.

4. Um estudo sobre o emprego de outros algoritmos multi e mono-objetivo para a busca de alternativas de projeto.
5. Uma outra abordagem no que diz respeito a exploração do espaço de soluções, mais voltado às preferências do projetista, semelhante a metodologia apresentada por Danhaive & Mueller (2015). Esta é uma das sugestões que se acredita ser mais interessantes para um trabalho futuro.
6. Os processos de otimização baseados em heurísticas, requerem um elevado número de avaliações das funções objetivo. Este elevado número de execuções é muito custoso computacionalmente, fazendo com que o processo demore muitas horas. Uma alternativa seria a utilização de metamodelos em substituição ao sistema físico custoso, como é o método dos elementos finitos. O presente autor realizou experimentações sobre o tema durante a elaboração deste trabalho, mas optou por sua não inclusão por considerar que esta aplicação merecesse seu próprio trabalho.

REFERÊNCIAS

- ADELI, H.; KAMAL, O. Optimization of Space Trusses. *Computer and Structures*, v. 24, n. 3, p. 501-511, 1986.
- AGA, A. A. A.; ADAM, F. M. Design optimization of reinforced concrete frames. *Open Journal of Civil Engineering*, v. 5, p. 74-83, 2015.
- ARORA, J. S. *Introduction to optimum design*. 3 ed. Waltham: Elsevier, 2012. 896 p.
- ARTEAGA, D. I.; *Otimização de Estruturas Reticuladas Utilizando Algoritmos Genéticos*. Dissertação de Mestrado, Publicação E.DM-006A/15, Departamento de Engenharia Civil e Ambiental. Universidade de Brasília. Brasília, DF, 97p.
- AZEVEDO, C. R. B.; ARAÚJO, A. F. R.; Correlation between diversity and hypervolume in evolutionary multiobjective optimization. *Proceedings of the IEE Congress on Evolutionary Computacion*. Nova Orleans, Estados Unidos. 2011.
- BABAEI, M.; MOLLAYI, M. Multi-objective optimization of reinforced concrete frames using NSGA-II algorithm. *Engineering Structures and Technologies*, v. 8, n. 4, p. 157-164, 2016.
- BLOCK, P.; Ochsendorf, J. Thrust network analysis: a new methodology for three-dimensional equilibrium. *Journal of the IASS*, v. 48, n. 3, p. 167-173, 2007.
- BOJCZUK, D.; MRÓZ, Z. On optimal design of supports in beam and frame structures. *Structural Optimization*, v. 16, p. 47-57, 1998.
- BUHL, T. Simultaneous topology optimization of structure and supports. *Structural multidisciplinary optimization*, v. 23, p. 336-346, 2002.
- CAMP, C. V.; PEZESHK, S.; HANSSON, H. Flexural design of reinforced concrete frames using a genetic algorithm. *Journal of Structural Engineering*, v. 129, n. 1, p. 105-115, 2003.
- COELLO, C. A.; CHRISTIANSEN, A. B. Multiobjective optimization of trusses using genetic algorithms. *Computers and Structures*, v. 75, p. 647-660, 2000.

- DANHAIVE, R.; MUELLER, C.; Combining parametric modeling and interactive optimization for high-performance and creative structural design. *International Association for Shell and Spatial Structures (IASS) Symposium*. 2015.
- DEBS, K.PRATAP, A.; AGARWAL, S. *et al.* A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182-197. 2002.
- DICKSON, M.M; TILLÉ, Y. Ordered spatial sampling by means of the traveling salesman problem. *Computational Statistics*, v. 31, n. 3, p. 1359-137, 2016.
- DURILO, J. J.; NEBRO, A. J. jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software*, v. 42, n. 10, p. 760-771, 2011.
- FADAEI, M. J.; GRIERSON, M. J. Design optimization of 3D reinforced concrete structures. *Structural Optimization*, v. 12, p. 127-134, 1996.
- FAINA, L., A global optimization algorithm for the three-dimensional packing problem. *European Journal of Operational Research*, v. 126, n. 2, p. 340-354, 2000.
- FEW, L. The Shortest Path and the Shortest Road through n Points. *Mathematika* 2, no. 2, 141–144, 1955.
- FORRESTER, A. I. J.; KEANE, A. J. *Engineering design via surrogate modeling*. Chichester: John Willey & Sons. 2008. 288 p.
- FU, K. C. An application of search technique in truss configurational optimization. *Computers and Structures*, v. 3, p. 315-328, 1973.
- GANDOMI, A. H.; YANG, X. S.; TALATAHARI, S. *et al.* Metaheuristics applications in structures and infrastructures. Waltham: Elsevier. 2013.p. 556.
- GUERRA, A.; KIOUSIS, P. D. Design optimization of reinforced concrete structures. *Computers and Concrete*, v. 3, n. 5, p. 313-334, 2006.
- GOVINDARAJ, V.; RAMASAMY, J. V. Optimum detailed design of reinforced concrete continuous beams using Genetic Algorithms. *Computers and Structures*, v. 84, p. 34-48, 2005.

- HAJELA, P.; LEE, E. Genetic Algorithms in Truss Topological Optimization. *Journal of Solids Structures*, v. 32, n. 22, p. 3341-3357, 1995.
- HICKS, R; HENNE, P. A. Wing Design by Numerical Optimization, *Journal of Aircraft*, v. 15, n. 7, p. 407-412, 1978.
- HONDA, S., IGARASHI, T., NARITA, Y.; Multi-objective optimization of curvilinear fiber shapes for laminated composite plates by using NSGA-II. *Composites: Part B*. v. 45. p. 1071-1078. 2013.
- IMAM, M. H.; AL-SHIHRI, M. Optimum topology of structural supports. *Computers & Structures*, v. 61, n. 1, pp. 147-154, 1996.
- IVERSON, K. E. A programming language, Nova Iorque: John Wiley & Sons, 1962, p. 286.
- JÓŹWIAK, S. F. Probability-based optimization of truss structures. *Computers and Structures*, v. 32, n. 1, p. 87-91, 1989.
- KAVEH, A.; BEHNAM, A. F. Design optimization of reinforced concrete 3D structures considering frequency constraints via a charged system search. *Scientia Iranica: Transaction A: Civil Engineering*, v. 20, n. 3, p. 387-396, 2011.
- KAVEH, A.; SABZI, O. Optimal design of reinforced concrete frames using big bang-big crunch algorithm. *International Journal of Civil Engineering*, v. 10, n. 3, p. 189-200. 2012.
- KIRSCH, U.; TAYE, S. On optimal topology of grillage structures. *Engineering with Computers*, v. 1, p. 229-243, 1986.
- KHOT, N. S; BERKE, L.; VENKAYYA, V. B. Comparison of Optimality Criteria Algorithms for Minimum Weight Design of Structures. *American Institute of Aeronautics and Astronauts Journal*, v. 17, n. 2, p. 182-190, 1979.
- KRIPKA, M. *et al.* Optimization of reinforced concrete frames by Harmony Search Method. In: WORLD CONGRESS ON STRUCTURAL AND MULTIDISCIPLINARY OPTIMIZATION, 11, 2015. Sydney. *Anais...* Sydney, 2015.

KRIPKA, M. Determinação do posicionamento ótimo dos apoios em edificações analisadas pelo modelo de grelha. 1998. p. 131. Tese (Doutorado em Engenharia de Estruturas) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 1998.

KWAK, H. G.; KIM, J. Optimum design of reinforced concrete plane frames based on predetermined section database. *Computer-Aided Design*, v. 40, p. 396-408, 2008.

LEE, C.; AHN, J. Flexural design of reinforced concrete frames by genetic algorithm. *Journal of Structural Engineering*, v. 129, n. 6, p. 762-774, 2003.

LEE, H. J. *et al.* Optimum RC column reinforcement considering multiple load combinations. *Structural Multidisciplinary Optimization*. v. 39, p. 153-170, 2009.

LEPS, M.; SEJNOHA, M. New approach to optimization of reinforced concrete beams. *Computers and Structures*, v. 81, p. 1957-1966, 2003.

LEVINE. J.; DUCATELLE, F. J. Ant colony optimization and local search for bin packing and cutting stock problems. *Journal of the Operational Research Society*, v. 55, n. 7, p. 705-716, 2004.

LIU, Y. W., MOSES, F. Truss Optimization including reserve and residual reliability constraints. *Computer and Structures*, v. 42, n. 3, p. 355-363, 1992.

LIU, Z; ATAMTURKTUR, S; JUANG, C. H. Performance based robust design optimization of steel moment resisting frames. *Journal of constructional steel research*, v. 89, p 165-174, 2013.

MACIUNAS, D.; BELEVICIUS, R. Multi-objective optimization of grillages using adaptive genetic algorithm. *International Review of Mechanical Engineering*, v. 6, n. 3, p. 432-439, 2012.

MASIN, M.; BUKCHIN, Y.; Diversity maximization approach for multiobjective optimization. *Operations Research*. v. 56, n. 2. p. 411-424, 2008.

MARTINEZ-MARTIN, F. J.; *et al* Multi-objective optimization design of bridge piers with hybrid heuristic algorithms. *Journal of Zhejiang University-Science A (Applied Physics & Engineering)*, v. 13, n. 6, p. 420-432, 2012.

- MEDEIROS, G. F. Otimização de grelhas de concreto armado. 2010. p. 109. Monografia – Faculdade de Engenharia e Arquitetura, Universidade de Passo Fundo, Passo Fundo, 2010.
- MIGUEL, L. F. F.; LOPEZ, L. H., MIGUEL, L. F. F., Multimodal size, shape and topology optimization of truss structures using the Firefly Algorithm. *Advances in Engineering Software*, v. 56, p. 23-37, 2012.
- MIRHASSANI, S. A.; YARAHMADI, A. Wind farm layout optimization under uncertainty. *Renewable Energy*, v. 107, p. 288-297, 2017.
- MONACI, M. Algorithm for packing and scheduling problems. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, v. 1, n. 1, p. 85-87, 2003.
- MOTTL, J. Truss system optimization using the voting method. *Computers and Structures*, v. 45, n. 1, 127-142, 1992.
- NAKIB, R. Deterministic and reliability-based optimization of truss bridges. *Computers and Structures*, v. 65, n. 5, pp. 767-775, 1997.
- NEBRO, A. J. *et al.* A study of convergence speed in multi-objective metaheuristics. *Parallel Problem Solving from Nature Conference*. Dortmund. 2008.
- OHSAKI, M. Genetic Algorithm for Topology Optimization of Trusses. *Computers and Structures*, v. 57, n. 2, p. 219-255, 1995.
- RAO, S. S. Game theory approach for multiobjective structural optimization. *Computers and Structures*, v. 25, n. 1, p. 119-127, 1987.
- ROSENKRANTZ, D. J.; STEARNS, R. E; LEWIS II, P. M., An Analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, v. 6, n. 3, p. 563-581, 1977.
- SARMA, K. C.; ADELI, H. Cost optimization of concrete structures. *Journal of Structural Engineering*, v. 124, n. 5, p. 570-578, 1998.
- SESOK, D.; MOCKUS, J.; BELEVICIUS, R.; KACENIAUSKAS, A. Global optimization of grillages using simulated annealing and high performance computing. *Journal of Civil Engineering and Management*, v. 16, n. 1, p. 95-101, 2010.

SHARAFI, P.; HADI, M. N. S.; TEH, L. Heuristic approach for optimum cost and layout design of 3D reinforced concrete frames. *Journal of Structural Engineering*. v. 138, n. 7, p. 853-863, 2012.

SHARAFI, P.; HADI, M. N. S.; TEH, L. Conceptual design optimization of rectilinear building frames: a knapsack problem approach. *Engineering Optimization*. v. 47, n. 10, p. 1303-1323, 2015.

SHARAFI, P.; HADI, M. N. S.; TEH, L. Form finding for rectilinear orthogonal buildings through charged system search algorithm. *International Journal of Optimization in Civil Engineering*. v. 7, n. 1, p. 129-142, 2017.

SHAW, D.; MILES, J.; GREY, A. Determining the structural layout of orthogonal framed buildings. *Computers and Structures*. v. 86, p. 1856-1864, 2008.

STANKOVIC, T., STORGA, M., MARJANOVIC, D. Synthesis of truss structure designs by NSGA-II and NodeSort Algorithm. *Journal of Mechanical Engineering*. v. 58. p. 203-212. 2012.

STOYAN, Y. G.; ZLOTNIK, M. V.; CHUGAY, A. M. Solving an optimization packing problem of circles and non-convex polygons with rotations into a multiply connected region. *Journal of the Operational Research Society*, v. 63, n. 3, p. 379-391, 2012.

STROMBERG, L. L. *et al.* Topology optimization for braces frames: combining continuum and beam/column elements. *Engineering structures*, v. 37. p. 106-124, 2012.

SVANBERG, K. Optimization of Geometry in Truss Design. *Computer Methods in Applied Mechanics and Engineering*, v. 28, p. 63-80, 1981.

SYBERFELDT, A. Multi-objective optimization of Real-World Manufacturing Process using cuckoo search. In: YANG, X. S., Ed. *Cuckoo Search and Firefly Algorithm: Theory and applications*. Suíça: Springer, 2014, p. 360.

SWANSON, D. A.; WU, H. T.; ASHRAFIUON, H. Optimization of aircraft engine suspension systems. *Journal of Aircraft*, v. 30, n. 6, p. 979-984, 1993.

- TAHERNEZHADIANI, K.; HAMZEH, A.; HASHEMI, S.; Towards enhancing solution space diversity in multi-objective optimization: a hypervolume-based approach. *International Journal of Artificial Intelligence & Applications*. v. 3. n. 1. 2012.
- TRIVEDI, H. V.; MUNJAL, B. S. Proposed approach for tubular truss optimization using graphics. *Computers and Structures*, v. 39, n. 5, p. 399-413, 1991.
- VIDOSA, F. G. *et al.* Optimization of reinforced concrete structures by simulated annealing. In: CHER MING TAN (Ed.). *Simulated Annealing*. Vienna, Austria: 2008. p. 307-320.
- VO-DUY, T. *et al.* Multi-objective optimization of laminated composite beam structures using NSGA-II algorithm. *Composite Structures*. v. 168. p. 498-509. 2017.
- XIAO, G.; JUAN, Z.; ZHANG, C. Detecting trip purposes from smartphone-based travel surveys with artificial neural networks and particle swarm optimization. *Transportation Research Part C: Emerging Technologies*, v. 71, p. 447-463, 2016.
- WANG, B. P.; CHEN, J. L. Application of genetic algorithm for the support location optimization of beams. *Computers & Structures*, v. 58, n. 4, pp. 797-800, 1996.
- WANG, D. Optimization of support positions to minimize the maximal deflection of structures. *International Journal of Solids and Structures*, v. 41, p. 7445-7458, 2004.
- WU, S.-J.; CHOW, P.-T. Integrated discrete and configuration optimization of trusses using genetic algorithms. *Computers and Structures*, v. 55, n. 4, p. 696-702, 1995.
- YANG, B.; BLETZINGER, K. U., ZHANG, Q.; ZHOU, Z. Frame structural sizing and topological optimization via a parallel implementation of a modified particle swarm algorithm. *Journal of Civil Engineering*, v. 16, n. 6, p. 1359-1370, 2013.
- YAO, X.; ZHOU, Y.; ZHANG, X. Grillage structural optimization based on genetic algorithm and mechanical direct calculation. *Advanced Materials Research*, v. 1061-1062, p. 1135-1139, 2015.
- ZITZLER, E.; THIELE, L. Multi-objective Optimization Using Evolutionary Algorithms – a Comparative Case Study. Springer, 1998.

ZHENG, J.; SHEN, R.; ZOU, J; Enhancing Diversity for NSGA-II in Evolutionary Multi-Objective Optimization. *International Conference on Natural Computation*. 2012.

ZOU, X. K.; CHAN, C. M.; WANG, Q. Multiobjective optimization for performance-based design of reinforced concrete frames. *Journal of Structural Engineering*, v. 133, n. 10, p. 1462-1474, 2007.

ZHU, J. H.; ZHANG, W. H. Integrated layout design of supports and structures. *Computer Methods in Applied Mechanics and Engineering*, v. 199, n. 9-12, p. 557-569, 2010.