

UNIVERSIDADE FEDERAL DE ITAJUBÁ
INSTITUTO DE MATEMÁTICA E COMPUTAÇÃO

MARCELO CAVALCA FILHO

ALGORITMOS PARA O GREEN FACILITY LOCATION

ITAJUBÁ
2021

MARCELO CAVALCA FILHO

ALGORITMOS PARA O GREEN FACILITY LOCATION

Trabalho de Curso submetido à Universidade Federal de Itajubá como parte dos requisitos necessários para a obtenção do Grau de Bacharel em Ciência da Computação. Sob a orientação do Professor Dr. Pedro Henrique Del Bianco Hokama e do Professor Dr. Mário César San Felice.

ITAJUBÁ
2021

AGRADECIMENTOS

Aos meus pais, Geniara e Marcelo, e à minha irmã, Marissa, pelo incentivo durante o meu período de universidade e o apoio emocional oferecido a mim em toda a minha formação.

Agradeço à Ruth, minha parceira para a vida, que sempre esteve comigo até nos momentos mais difíceis e está sempre iluminando minha vida.

Agradeço aos meus orientadores, Professor Dr. Pedro Henrique Del Bianco Hokama e do Professor Dr. Mário César San Felice, pela oportunidade e por se manterem pacientes e sempre presentes em todo o desenvolvimento deste trabalho.

Agradeço aos professores pelos seus conselhos e seu vasto conhecimento oferecido, que guiaram e foram essenciais para a minha formação acadêmica.

E por fim, sou grato à UNIFEI pela oportunidade de participar desta instituição que oferece aos alunos uma formação profissional de qualidade e transformação de vida.

RESUMO

Com o agravante aumento de poluentes no meio ambiente e conseqüentemente uma maior preocupação das pessoas com as suas pegadas ecológicas as organizações no mundo todo vêm mudando a maneira que estas distribuem seus produtos aos seus clientes. Um dos problemas que essas organizações enfrentam é o problema de decidir onde serão alocadas as suas instalações e quais destas instalações atenderá cada um dos seus clientes. Este problema é conhecido como o problema de localização de instalações.

Levando em consideração, além do custo financeiro, mas também a emissão de poluentes no meio ambiente, chamado custo verde, nos deparamos com o problema de minimizar ambos os critérios – este é conhecido como o problema verde de localização de instalações.

Este trabalho tem como objetivo implementar algoritmos para o problema verde de localização de instalações gerando soluções para que os tomadores de decisão da organização possam escolher a alternativa que responda às suas necessidades. É apresentada uma modelagem em programação linear inteira para resolver o problema e um método para encontrar soluções da fronteira de Pareto. Resultados experimentais mostraram a viabilidade do nosso método através de instâncias adaptadas da literatura.

Palavras-chave: Alocação de Instalações, Logística Verde, Programação Linear Inteira, Otimização Multiobjetivo.

ABSTRACT

Due to the increase of pollutants in the environment and people's consequent concern about their ecological footprint, organizations around the world are changing the way they deliver their products to their customers. One of the problems that these organizations faces is the problem of deciding where to locate their facilities and which of these facilities will serve each of their customers. This problem is known as the Facility Location Problem.

Taking into consideration not only the financial cost, but also the emission of pollutants into the environment, called green cost, we are faced with the problem of minimizing both criteria – this is known as the Green Facility Location problem.

This paper aims to implement algorithms for the Green Facility Location problem, generating solutions so that the decision makers in the organization can choose the alternative that satisfies their needs. An integer linear programming model for solving the problem and a method for finding Pareto frontier solutions are presented. Experimental results showed the feasibility of our method through instances adapted from the literature.

Keywords: Facility Location Problem, Green Logistics, Integer Linear Programming, Multi-Objective Optimization.

LISTA DE FIGURAS

Figura 1: Ilustração das soluções problema.....	14
Figura 2: Demonstração de Pareto Dominância no espaço de objetivos.....	16
Figura 3: Resultado de execução dos Testes 1 e 2.	25
Figura 4: Resultado de execução dos Testes 3 e 4.	25
Figura 5: Resultado de execução dos Testes 5 e 6.	27
Figura 6: Resultado de execução dos Testes 7 e 8.	27
Figura 7: Resultado de execução dos Testes 9 e10.	28
Figura 8: Resultado de execução dos Testes 11 e 12.	29
Figura 9: Resultado de execução dos Testes 13 e 14.	30
Figura 10: Resultado de execução dos Testes 15 e 16.	30
Figura 11: Resultado de execução dos Testes 17 e 18.	31
Figura 12: Resultado de execução dos Testes 19 e 20.	32
Figura 13: Resultado de execução dos Testes 21 e 22.	33

LISTA DE TABELAS

Tabela 1: Resultado dos testes 1 e 2, e a diferença entre os pontos gerados.....	25
Tabela 2: Resultado dos testes 3 e 4, e a diferença entre os pontos gerados.....	26
Tabela 3: Resultado dos testes 5 e 6, e a diferença entre os pontos gerados.....	27
Tabela 4: Resultado dos testes 7 e 8, e a diferença entre os pontos gerados.....	27
Tabela 5: Resultado dos testes 9 e 10, e a diferença entre os pontos gerados.....	28
Tabela 6: Resultado dos testes 11 e 12, e a diferença entre os pontos gerados.....	29
Tabela 7: Resultado dos testes 13 e 14, e a diferença entre os pontos gerados.....	30
Tabela 8: Resultado dos testes 15 e 16, e a diferença entre os pontos gerados.....	30
Tabela 9: Resultado dos testes 17 e 18, e a diferença entre os pontos gerados.....	31
Tabela 10: Resultado dos testes 19 e 20, e a diferença entre os pontos gerados.....	32
Tabela 11: Resultado dos testes 21 e 22, e a diferença entre os pontos gerados.....	33

LISTA DE ABREVIATURAS E SIGLAS

CFLP – Capacitated Facility Location Problem

FLP – Facility Location Problem

GEE – Gases de Efeito Estufa

GFL – Green Facility Location

IBM – International Business Machines Corporation

OMS – Organização Mundial da Saúde

PL – Programação Linear

PLI – Programação Linear Inteira

UFLP – Uncapacitated Facility Location Problem

WWF – World Wide Fund for Nature

SUMÁRIO

1.	INTRODUÇÃO	5
1.1.	EMISSÃO DE POLUENTES	7
1.2.	OBJETIVOS.....	8
1.3.	TRABALHOS RELACIONADOS.....	8
2.	DESCRIÇÃO DO PROBLEMA	10
2.1.	FACILITY LOCATION PROBLEM	10
2.2.	GREEN FACILITY LOCATION.....	12
3.	FUNDAMENTAÇÃO TEÓRICA.....	14
3.1.	OTIMIZAÇÃO MULTIOBJETIVO.....	14
3.2.	PROGRAMAÇÃO LINEAR INTEIRA	17
4.	ALGORITMO PARA O GFLP.....	20
4.1.	FORMULAÇÃO MULTIOBJETIVO PARA O GFLP.....	20
4.2.	ALGORITMO PARA ENCONTRAR A FRONTEIRA DE PARETO	21
5.	RESULTADOS E DISCUSSÕES	24
6.	CONCLUSÕES.....	34
	REFERÊNCIAS	35
	APÊNDICE A – Implementação em C++	38

1. INTRODUÇÃO

No mundo todo, o cenário econômico tem sido modificado e influenciado pelas novas tecnologias, resultando em empresas que buscam fornecer, de maneira mais eficiente, seus produtos e serviços à uma maior gama de clientes. Entretanto, com a variedade de serviços, que vão desde entrega de compras até aqueles que são mais abrangentes, como distribuição de água e energia, uma das dificuldades encontradas é a alocação de suas infraestruturas. Isto leva em consideração diversos fatores, como: local, quantidade e tipo de clientes, custo financeiros e outros, sempre apontando para providenciar uma entrega rápida, segura e de baixo custo (BRANSKI, 2008).

Para a determinação destes fatores, portanto, as instituições organizam estudos a fim de obter as melhores localizações para fundar suas atividades; deste problema, surgem questões imediatas: de qual local utilizar, qual o custo de abertura, custo de providenciamento do serviço aos seus clientes e sobre o lucro. Em um cenário hipotético, no qual os custos não fossem levados em consideração, a melhor opção seria aproveitar todas as instalações possíveis e, entregar o produto (ou serviço), ao cliente a partir de cada instalação mais próxima. Porém as consequências dessa escolha no mundo real seria um custo absurdo para as empresas e para seus clientes.

No outro extremo da situação, há a opção de minimizar ao máximo o custo de abertura das instalações, abrindo apenas uma instalação responsável por prover o serviço para todos os clientes, todavia, tal escolha não leva em consideração a qualidade de serviço e o custo de conexão do cliente, resultando possivelmente, que muitos dos clientes fiquem insatisfeitos – seja com a demora ou com o custo final do produto, e acabem optando por serviço semelhante na concorrência.

Obter soluções para gerar os melhores pontos para implantar as instalações de uma provedora de serviços é o que o *Facility Location Problem* (FLP) propõe: dado um conjunto de potenciais alternativas e clientes em uma determinada área, a sua solução auxiliará a provedora na tomada de decisão na escolha de quais instalações deverão ser abertas, de modo a oferecer seus serviços à maior quantidade de clientes. O problema leva em consideração o custo financeiro de abertura de cada instalação e o custo do transporte deste serviço até um determinado cliente, tendo como objetivo reduzir ao máximo este custo de maneira que se encontre o equilíbrio de benefício à ambas as partes (AFSHARI et al., 2014).

Outro conceito interessante, que tem se tornado mais relevante e permeando tais assuntos, de forma a questionar a influência das atividades antropogênicas na esfera ambiental, é a logística verde (GOETZINGER; BRANDT; NEUMANN, 2012). Com o avanço das

preocupações climáticas, viu-se necessário reduzir as nossas pegadas ecológicas, que incluem desde as emissões de gases de efeito estufa (GEE), até o consumo de água, passando também pela quantidade de quaisquer recursos naturais consumidos em larga escala, a fim de manter o planeta saudável para as gerações vindouras. Não obstante com a redução dos impactos ambientais – como a redução da emissão de GEE, ao procurar metodologias eficientes de distribuição de serviços, as organizações também apresentam resultados de economia de recursos – como tempo e demais investimentos.

Ainda segundo os autores, a partir deste ponto, surge o *Green Facility Location* (GFL), a extensão do FLP que busca reduzir não apenas o custo financeiro, mas também o denominado custo verde, que é o nível de poluição gerada pelo consumo elétrico e de combustíveis fósseis, tanto da instalação, como do transporte, seja através de veículos ou também do envio do produto através de dutos ou postes.

A pesquisa por trás de soluções para o GFL iniciou-se muito recentemente, portanto, a literatura na área ainda é escassa, mas traz conteúdos influentes para o problema, como sobre quais alternativas são viáveis e quais medidas devem ser tomadas para resolvê-lo. Para um nível de detalhamento mais específico, o custo verde poderia ser dividido entre seus componentes, como combustível e elétrico. Todavia, esta ideia não é aplicada no desenvolvimento deste trabalho, pois considera-se que todo o consumo seja transferido à uma mesma escala de grandeza, de maneira a facilitar a abordagem.

Este trabalho tem como propósito descrever o problema do GFL através da Programação Linear Inteira, uma variação da Programação Linear (PL) onde todas, ou parte das variáveis, devem pertencer ao conjunto dos números inteiros. O objetivo de uma PL é otimizar determinada função linear, seja maximizando ou minimizando um parâmetro específico, que está sujeito a restrições lineares, além das restrições nas variáveis que limitam suas fronteiras (JAIN et al., 2002).

Otimizar um problema com restrições pode ser considerado um desafio, ainda mais em situações que se pode haver múltiplos objetivos. Encontrar uma única solução para problemas multiobjetivo não existe no mundo real – para isso, existem meios de se obter diversas soluções que otimizem gradualmente cada uma das variáveis do problema. Estas soluções, então, devem ser encaminhadas aos tomadores de decisão, que terão alternativas para escolher a melhor e aplicar em seu ambiente (COELLO; DHAENENS; JOURDAN, 2009).

As soluções de problemas multiobjetivo devem, também, ser condizentes com aquilo que se busca otimizar; para o GFL, por exemplo, abrir todas as instalações e fornecer o serviço ao cliente mais próximo de cada uma seria uma resposta, mas enquanto houver a possibilidade

de existir alternativa que se mostre financeiramente e ecologicamente mais viável, essa pode ser, nem de longe a mais correta; portanto deve-se afunilar todas as soluções encontradas ao longo da otimização para fornecer alternativas relevantes. A fronteira de Pareto descreve um limite onde os ponto-soluções podem chegar, ou seja, é um cenário onde não é possível otimizar ainda mais uma das variáveis, ou objetivos, sem que afete o outro negativamente (SILVA NETO, 2011). A concepção deste trabalho visa desenvolver um algoritmo envolto na problemática do GFL, exibindo, ao final de sua execução, um conjunto de soluções encontradas, cada uma com seus custos verde e financeiro.

1.1. EMISSÃO DE POLUENTES

Segundo a Organização Mundial da Saúde (WHO, 2021), a exposição à poluição atmosférica é responsável por cerca de 7 milhões de mortes prematuras por ano, podendo causar, tanto em adultos, quanto em crianças, graves interferências na saúde. As novas diretrizes (WHO, 2021) visam aumentar a qualidade do ar, restringindo os limites para os seis poluentes com maior aumento de evidências sobre os danos à exposição: material particulado fino (MP_{10} e $MP_{2,5}$), ozônio (O_3), dióxido de nitrogênio (NO_2), dióxido de enxofre (SO_2) e monóxido de carbono (CO), que são emitidos a partir de diversas atividades antropogênicas, direta ou indiretamente. No Brasil, tais medidas poderão ter efeitos significativos, uma vez que o país aumentou em 9,6% – entre 2019 e 2018, a emissão bruta de gases de efeito estufa (OBSERVATÓRIO DO CLIMA, 2019).

Considerando que todas as intervenções humanas possuem um histórico de impactos ambientais – podendo ser dimensionados com a pegada ecológica, uma ferramenta capaz de medir os impactos do consumo humano sobre os recursos naturais (WWF-BRASIL, 2013), é de se ponderar que quaisquer instalações, físicas ou virtuais, possuam repercussões no meio ambiente a partir do consumo de determinados recursos – a exemplo da energia elétrica. A partir deste ponto, conceitua-se o custo verde: um parâmetro que estipula a magnitude do impacto ambiental a partir da emissão de todo e quaisquer poluentes, em especial aqueles ligados à poluição atmosférica.

Sendo o *Facility Location Problem* (FLP), um problema que descreve a necessidade de se distribuir, da melhor forma, a localização de instalações a fim de atender os respectivos clientes com a minimização dos custos de instalação, caracterizado como um problema logístico de enorme importância e aplicado no cotidiano a partir da decisão de localização de estruturas como escolas, centros de distribuição, hospitais, postos de correios e outros, incorpora-se,

portanto, os conceitos ambientais no problema de *Green Facility Location* (GFL), que atribui como objetivo, concomitantemente, a diminuição do custo verde (BOUCHERY et al., 2017).

O esperado deste trabalho é fornecer uma solução do problema multiobjetivo, caracterizado acima, aos agentes de organizações provedoras de serviços como uma maneira de avaliar alternativas viáveis para a alocação de suas instalações, além de fornecer os caminhos que as instalações devem realizar para se conectar aos clientes, possibilitando o recebimento do produto fornecido, reduzindo ao máximo os custos financeiros e a poluição gerada pela instalação e transporte.

1.2. OBJETIVOS

1.2.1. Objetivo Geral

Desenvolver um algoritmo para solucionar o problema do *Green Facility Location* minimizando, simultaneamente, o custo financeiro e a emissão de poluentes, a fim de alocar instalações e prover serviços aos clientes de uma determinada área.

1.2.2. Objetivos Específicos

Visando o objetivo geral, dentre os objetivos específicos estão:

- Aplicar os conceitos de programação linear, otimização multiobjetivo, otimalidade de Pareto e de normalização através dos pontos Nadir e ideal.
- Construir um algoritmo utilizando o otimizador CPLEX.
- Analisar os resultados obtidos com a aplicação desenvolvida para os dados propostos.
- Avaliar o algoritmo em questão de performance de execução e validade dos resultados obtidos.

1.3. TRABALHOS RELACIONADOS

Harris, Mumford e Naim (2014) apresentam um estudo e desenvolvimento da formulação do problema do GFL de maneira clara e direta, mas encontram uma forma diferente de alocar as instalações e os clientes, dividindo-as em duas etapas: a primeira consiste em uma decisão de localização multiobjetivo, onde se encontra todas as soluções viáveis para as instalações, visando minimizar ambos os objetivos; já a segunda etapa, a alocação de cliente multiobjetivo, busca, dada uma solução de alocação de instalações, encontrar todas as soluções viáveis de conexão de clientes às instalações.

Mallidis, Dekker e Vlachos (2010), apresentam uma visão abrangente de todos os dados e resultados obtidos pelos autores em seu trabalho, que propõe uma ferramenta para auxiliar gestores de cadeia de suprimentos no design de instalações e clientes, levando em consideração o custo financeiro e os impactos ambientais. A aplicação de sua metodologia se baseia no desenvolvimento de redes de cadeia de suprimentos verde no sudeste europeu.

Goetzinger, Brandt e Neumann (2012), apresentam um estudo de caso do *Facility Location Problem* (FLP) em uma empresa real, porém focando nas emissões de poluentes através do transporte de serviço da empresa, ao invés do custo financeiro. Neste estudo, o problema é formulado em uma P-mediana, utilizando os dados fornecidos pela organização, Greenfinch Ltd., e comparando as melhores soluções quanto à emissão de CO₂ com as melhores soluções do custo financeiro.

O trabalho de Afshari et al. (2014) apresenta uma modelagem mais complexa aplicada a um estudo de caso real, buscando minimizar os custos financeiros das instalações enquanto maximiza a satisfação dos clientes da organização visando a perspectiva sustentável.

2. DESCRIÇÃO DO PROBLEMA

Neste capítulo iremos descrever o problema da alocação de instalações (FLP) e o *Green Facility Location*.

2.1. FACILITY LOCATION PROBLEM

O *Facility Location Problem* possui diversas variações e compreende diversos aspectos que podem distinguir e determinar os modelos e abordagens mais aptas para resolvê-los. Para os autores Klose e Drexl (2005), os modelos para este problema podem ser classificados em algumas formas diferentes a partir de certas características, sendo elas apresentadas a seguir.

- a) **Organização da região geográfica:** podendo ser retratada como um plano ou, a partir das condições do problema, pode ser mapeada para um modelo discreto.
- b) **Objetivos:** podem compreender a minimização de um parâmetro, como o custo de abertura de uma instalação, ou a maximização de outro, como o alcance de uma instalação.
- c) **Capacidade:** os FLP podem levar em consideração, ou não, a limitação de capacidade de demanda da instalação em questão. Ainda de acordo com os autores (KLOSE; DREXL, 2005), se a capacidade de cada instalação for superior ou igual à soma de todas as instalações, um modelo sem restrições de capacidade pode ser considerado um modelo com tais restrições.
- d) **Estágios:** a quantidade de estágios de uma logística de atendimento pode variar conforme suas aplicações. Quando as instalações atendem diretamente aos clientes, são determinados aqueles modelos que possuem apenas um único estágio; caso contrário, ou seja, quando além das instalações que atendem os clientes, existem instalações que atendam à outras instalações, são considerados os modelos multiestágios. De todas as formas é o problema que determina a quantidade de estágios.
- e) **Serviços:** os problemas envolvem examinar instalações responsáveis por fornecer apenas um, ou mais de um, tipo de serviço.
- f) **Influência da demanda:** grande parte dos problemas FLP consideram que a demanda pelo serviço é suficiente, independente da localidade determinada no problema; entretanto, existe uma parcela de abordagens que considera que a demanda pode ser instigada pela ausência ou existência de certa instalação.

- g) **Dinâmica:** os modelos para estes problemas podem ser estáticos, que buscam otimizações para períodos específicos, ou dinâmicos, que consideram o tempo como agente decisivo e que deve ser levado em consideração.

Segundo os autores Farahani, SteadieSeifi e Asgari (2010), os modelos ainda podem ser particularizados a partir da quantidade de critérios a serem otimizados, sendo com múltiplos objetivos ou múltiplos atributos.

Neste trabalho o *Facility Location Problem* (FLP) que irá ser considerado descreve a situação onde se deve alocar um conjunto de possíveis instalações $N = \{1, 2, \dots, n\}$ para prover um determinado serviço a um conjunto $I = \{1, 2, \dots, m\}$ de clientes de uma dada área; há um custo c_j para se alocar uma determinada instalação $j \in N$, além do custo de conexão h_{ij} para satisfazer a demanda de um determinado cliente $i \in I$ para uma instalação $j \in N$. No problema, deve-se minimizar o custo da alocação das instalações somado com o custo das conexões dos clientes. Na literatura muitas das vezes este problema é chamado de *Uncapacitated Facility Location Problem* (UFLP) por não possuir restrições de capacidade nas instalações.

Uma variação do problema acima, denominado *Capacitated Facility Location Problem* (CFLP), se baseia na questão de que as instalações possuem uma capacidade máxima de serviço que podem oferecer e que cada cliente possui uma demanda específica, assim como apresentado por Klose e Drexler (2005).

2.1.1. Modelagem Matemática do FLP

Segundo Wolsey (2021), para representar matematicamente o problema é necessário fornecer algumas variáveis adicionais: dado um conjunto de potenciais instalações $I = \{0, 1, \dots, n\}$ e um conjunto $J = \{0, 1, \dots, m\}$ de clientes, temos $x_j \in \{0, 1\}$, sendo uma variável binária que descreve se a instalação j será aberta ou não, com 0 representando que não será aberta e 1 caso contrário; a variável contínua $y_{ij} \in \{0, 1\}$ descreve a fração de demanda que um determinado cliente i tem para com a instalação j ; a condição para esta variável é apresentada na Equação (1):

$$\sum_{j \in J} y_{ij} = 1, \text{ para todo } i \in I. \quad (1)$$

Outra restrição implícita, é a de que um cliente i só pode receber o serviço de uma instalação j quando esta estiver aberta, apresentada na Equação (2):

$$y_{ij} - x_j \leq 0 \text{ para todo } i \in I \text{ e } j \in J. \quad (2)$$

Tem-se então, na Equação (3), a função objetivo do UFLP:

$$\min \sum_{j \in J} c_j * x_j + \sum_{i \in I} \sum_{j \in J} h_{ij} * y_{ij}. \quad (3)$$

Ainda segundo Wolsey (2021), UFLP ajuda a resolver os problemas relacionados à alocação de instalações mas considerar que uma instalação possa atender uma demanda infinita de clientes não é realista, portanto há o *Capacitated Facility Location Problem* (CFLP); o modelo para descrever o CFLP é o mesmo apresentado para UFLP, com exceção das duas restrições propostas – considere-se que uma instalação j tenha uma capacidade máxima de atendimento u_j e que um determinado cliente i tenha uma demanda b_i , com isso tem-se as seguintes restrições, apresentadas na Equação (4):

$$\sum_{j \in J} y_{ij} = b_i, \text{ para todo } i \in I, \quad (4)$$

Que substituirá a primeira restrição; e a Equação (5), que substitui a segunda restrição:

$$\sum_i y_{ij} - u_j * x_j \leq 0 \text{ para todo } j \in J. \quad (5)$$

O problema do *Facility Location* pode ser dividido em dois níveis estratégicos diferentes: quais instalações poderão ser abertas das disponíveis e quais clientes serão alocados à quais instalações. O primeiro nível é comumente definido como uma decisão estratégica, já que o custo financeiro para se abrir uma instalação é bem considerável, enquanto o segundo nível pode ser definido como uma decisão tática já que, com as instalações definidas, a alocação dos clientes pode ser redefinida em intervalos de tempo variados desde que haja a possibilidade de se diminuir o custo para providenciar o serviço aos clientes disponíveis.

2.2. GREEN FACILITY LOCATION

A preocupação com as alterações climáticas vem se destacando e tornando-se um debate constante a cada dia, em todo o planeta, já que a saúde deste se encontra comprometida por conta dos resultados das principais atividades humanas, se tornando uma das principais responsabilidades das corporações que as executam. Todos os processos de uma indústria – desde a produção, o transporte e até mesmo os setores comerciais, estão inclusos como os principais atores na intensificação do Efeito Estufa ao longo dos anos, através das emissões de gás carbônico, e demais gases característicos que contribuem para este fenômeno (HARRIS et al., 2011).

Os fornecedores de serviço hoje em dia se utilizam de diversas práticas diferentes para reduzir as emissões de gás carbônico nos setores de transporte e logística da indústria, mas poucos levam em consideração também o local de suas instalações - este é um dos pontos mais significativos quando se deseja reduzir tanto os custos monetários quanto às emissões de poluentes (BOUCHERY et al., 2017).

Com essa preocupação em mente, além de reduzir os custos monetários em transporte e instalação, as grandes corporações e governos no mundo todo também focam em reduzir as emissões de gás carbono em todos os seus processos de serviço. O *Green Facility Location Problem* (GFL) surge a partir desta questão – o FLP é visto, também, através das questões ambientais para oferecer novas alternativas àqueles que buscam fornecer seus serviços.

O GFL se modela da mesma maneira que o FLP com exceção de que a emissão de CO₂, que compõe 80% das emissões de gases poluentes na atmosfera em países industrializados, tanto em instalações quanto no transporte de um determinado serviço, será levada em conta em conjunto com o custo deste (GOETZINGER; BRANDT; NEUMANN, 2012).

2.2.1. Modelagem Matemática do GFLP

Este problema, por sua vez, é dividido em 2 problemas diferentes: minimizar os custos de implementação dos serviços, instalação e transporte; e minimizar os impactos ambientais gerados também pelos dois setores nas emissões de gás carbônico. O primeiro é descrito acima, na seção anterior, e o segundo a seguir, com novas variáveis: eg_i descreve as emissões de gás carbônico através do consumo de combustível de uma determinada instalação i ($i \in I$); ee_i descreve as emissões de gás carbônico gerados através do consumo de energia elétrica de uma instalação i ; e et_{ij} descreve as emissões de gases geradas pelo transporte do serviço providenciado por uma instalação i para um determinado cliente j ; por fim, segundo Harris, Mumford e Naim (2014), tem-se a seguinte função objetivo para descrever o custo verde do *Green Facility Location* (GFL), apresentado na Equação (6):

$$\min \sum_{i \in I} (eg_i + ee_i) * x_i + \sum_{i \in I} \sum_{j \in J} et_{ij} * y_{ij}. \quad (6)$$

O ideal em uma resolução do *Green Facility Location* é que sejam fornecidas alternativas diferentes aos tomadores de decisão que poderão optar por um serviço que custe menos à corporação ou que seja menos maléfico ao meio ambiente através da análise quantitativa dos resultados obtidos.

3. FUNDAMENTAÇÃO TEÓRICA

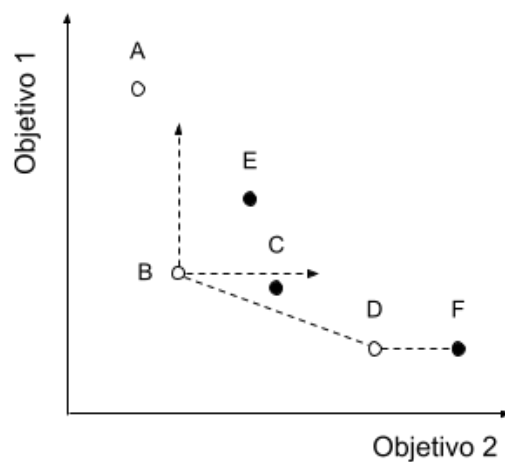
Neste capítulo vamos abordar sobre conceitos fundamentais para o desenvolvimento do trabalho.

3.1. OTIMIZAÇÃO MULTIOBJETIVO

A otimização de um problema consiste em maximizar ou minimizar uma determinada função dado um conjunto de restrições. A preocupação que os tomadores de decisão têm, no entanto, é o de otimizar diversos objetivos simultaneamente, o que traz à tona a Otimização Multiobjetivo. Este tipo de otimização é mais difícil de se resolver, quando comparado aos de único objetivo, já que aqui se busca uma solução focando em diversos objetivos diferentes e que podem ser conflitantes entre si. Em vista disso, nem sempre é óbvio o que seria uma solução ótima para problemas multiobjetivos já que algumas das soluções podem ser melhores para um objetivo enquanto são piores para outros (PIKE-BURKE, 2019).

No problema abordado neste trabalho, deseja-se encontrar uma distribuição ótima de instalações para prover um determinado serviço a um conjunto de clientes. É visto que, frequentemente, os objetivos buscados são conflitantes entre si, e isso deve-se ao fato que é necessário investir em instalações que estejam mais próximas aos clientes para os quais irá oferecer o serviço – minimizar o trajeto, enquanto minimiza-se os custos de se estabelecer uma determinada instalação. Para ilustrar a análise de otimizações multiobjetivo, a Figura 1 apresenta nos eixos os objetivos que se deseja minimizar, e nos pontos, as soluções encontradas pela resolução do problema:

Figura 1: Ilustração das soluções problema.



Adaptado de Daskin, 1995.

Neste caso, a solução A encontrada é a mais bem relacionada a diminuir o Objetivo 2; por outro lado esta é a pior solução para o Objetivo 1, já que é o maior ponto na linha vertical;

as soluções D e F são as melhores para o Objetivo 1, mas quando se trata do Objetivo 2 a solução D é melhor que a solução F – portanto, afirma-se que a solução F é inferior, ou dominada, por D e, portanto, não será levada em consideração.

Considera-se que as soluções B, C e E são soluções de compromisso: estas não são tão boas quanto A ou D em minimizar um objetivo ou outro; vê-se, claramente, que a solução E não é tão boa quanto a solução B, já que esta minimiza, ambos os objetivos, melhor que a outra; portanto, E é dominada por B (DASKIN, 1995).

3.1.1. Otimalidade de Pareto e Fronteira de Pareto

Em problemas multiobjetivos, há uma diferença quanto à otimização, quando se comparado a problemas de único objetivo – já que, nesse caso, se busca encontrar boas soluções de troca levando em consideração os objetivos que se busca otimizar; esse tipo de otimização é denominado “otimalidade de Pareto” (SILVA NETO, 2011).

Ainda segundo Silva Neto (2011), uma solução do problema pode ser denominada como solução eficiente ou não dominada, sendo considerada Pareto-ótima quando não existe outra solução que a domine, com ambas pertencentes ao conjunto de soluções do problema. A partir da comparação de vetores, o conceito de Pareto Dominância no espaço de objetos, com base na definição apresentada pelo autor: sejam $y_1, y_2 \in Y$, dois vetores no espaço de objetivos, afirma-se que y_1 domina y_2 se $y_1 \leq y_2$ e $y_1 \neq y_2$. Em outros casos, quando y_1 não domina y_2 e vice-versa, y_1, y_2 são chamados indiferentes. Qualquer solução considerada globalmente Pareto-ótima, será também localmente Pareto-ótima; a recíproca desta afirmação depende da Análise Convexa: o problema de otimização multiobjetivo definido é convexo caso todas as funções coordenadas da função vetorial f e o conjunto $X \subset \mathbb{R}^n$ forem convexos – neste caso, a recíproca faz-se verdadeira.

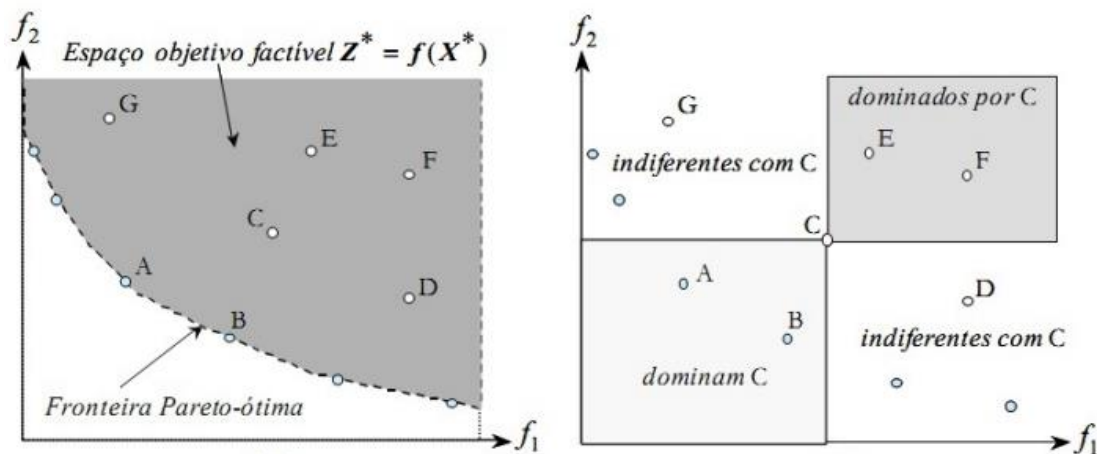
Seguindo este raciocínio, para uma solução $y_1 \in Y$ ser considerada fracamente eficiente, deve-se existir uma solução $y_2 \in Y$ tal que $y_2 < y_1 \in Y$. Antunes e Alves (2012) acrescentam, ainda, que o conjunto de soluções fracamente eficientes abrange as que são estritamente eficientes – ou seja, aquelas que atendem apenas um dos objetivos em totalidade (solução superotimizada para um objetivo); entretanto, por tais soluções comporem, de certa forma, a fronteira de Pareto, por ordem prática, não são consideradas como fracamente eficientes.

A fronteira Pareto-ótima é formada por pontos indiferentes entre si, sendo necessário algum critério de classificação para escolher aquelas soluções que se considerem mais

adaptáveis ao problema. Usualmente, no processo de decisão, solicita-se alguns parâmetros adicionais ao decisor para limitar a quantidade de soluções obtidas.

A Figura 2, no lado esquerdo, demonstra a Fronteira Pareto-ótima e o espaço objetivo factível hipotético ilustrado por $Z^* = f(X^*)$, onde X^* é o conjunto solução e $Z^* = f(X^*)$ é a imagem deste; já no lado direito, sendo uma demonstração cartesiana do espaço objetivo anterior, observa-se que os pontos G, C e D não são comparáveis, enquanto o ponto C domina os pertencentes ao retângulo superior direito, e os pontos A e B dominam o ponto C.

Figura 2: Demonstração de Pareto Dominância no espaço de objetivos.



Fonte: Arroyo (2002).

3.1.2. Ponto Ideal e Nadir

O ponto ideal de um problema é dado por: seja x_1 a melhor solução de $f_1(x)$, tal que $f_1(x_1) = f_1^*$ e x_2 a melhor solução de $f_2(x)$, tal que $f_2(x_2) = f_2^*$, portanto o ponto ideal, ou utópico, será (f_1^*, f_2^*) . Em outras palavras, o ponto ideal é aquele em que não se pode melhorar os valores de um objetivo, sem que afete – negativamente, os outros.

Seguindo a definição acima, os valores do ponto Nadir serão $f_1^N = f_1(x_2)$ e $f_2^N = f_2(x_1)$. A normalização através do ponto ideal e do ponto Nadir é dada pela Equação (7):

$$\frac{f_i(x) - f_i^*}{f_i^N - f_i^*}, \quad (7)$$

E serve para auxiliar na geração dos pesos de maneira normalizada (ARCENCIO, 2021).

3.1.3. Soma Ponderada

Junto aos conceitos de otimalidade de Pareto, o método de otimização de problemas multiobjetivos, mais intuitivo, é chamado de método da soma ponderada (*Weighted Sum*). Este

método possui como princípio atribuir um peso $w_i > 0$ a cada uma das funções objetivo $f_i(x)$, tendo $y_i = f_i(x)$ para todo $i = 1, \dots, p$ e sendo p o número total de funções objetivo a serem otimizadas no conjunto factível X e então, minimizar a função objetivo $\sum_{i=1}^p w_i f_i(x)$ sujeita às restrições do problema em questão. A abordagem da soma ponderada, em geral, apresenta soluções eficientes; porém, deve-se tomar cuidado com a atribuição de pesos, uma vez que ao aplicar determinado peso $w_i = 0$ para uma das funções objetivo, há uma grande chance de que sejam obtidas apenas soluções fracamente eficientes já que, dada a definição da otimalidade de Pareto, não é possível obter uma solução a partir daquele ponto a fim de melhorar pelo menos um dos objetivos sem que os demais sejam afetados Marler e Arora (2010).

Portanto, ao atribuir pesos diretamente para modelar as preferências que se deseja alcançar com o projeto, deve-se levar em consideração que o método da soma ponderada, apesar de eficiente, apenas providencia uma rasa aproximação para tais preferências, sendo fundamentalmente incapaz de incorporar aquelas informações mais complexas, podendo, também, refletir em resultados que não se encaixam no escopo do problema, ou simplesmente não eram esperados. A solução ótima de Pareto resultante de um conjunto específico de pesos, neste método, depende não somente de restrições que fazem parte do conjunto ideal de Pareto, mas também de relações entre os gradientes das diferentes funções objetivas e de suas magnitudes relativas, além do formato da hiper superfície ideal de Pareto (MARLER; ARORA, 2010). Em outros termos, apesar de simples de ser implementado, o método apresenta resultados que são dependentes dos pesos usados e especificados antes que ocorra o processo de otimização.

3.2. PROGRAMAÇÃO LINEAR INTEIRA

Grande parte dos problemas que envolvem alocação de recursos podem ser formulados utilizando a chamada Programação Linear Inteira (PLI) que é um modelo onde todas, ou algumas, das variáveis do problema pertencem ao conjunto de números inteiros; quando todas as variáveis são inteiras, a PLI é denominada pura, enquanto quando apenas algumas pertencem ao grupo dos números inteiros, denomina-se a PLI como mista.

A notação de uma PLI é dada por:

- Entrada: c_j, a_{ij}, b_i – Constantes
- Variáveis de decisão: X_j – Valores que se busca encontrar

Com isso a formulação de uma PLI é construída como na Equação (8):

$$\begin{array}{ll}
\text{MINIMIZAR} & \sum_j c_j X_j \\
\text{SUJEITO A} & \sum_j a_{ij} X_j \geq b_i, \quad \forall i \\
& X_j \geq 0, \quad \forall j
\end{array} \tag{8}$$

Esta formulação é denominada a forma canônica da PLI; entretanto, pode-se buscar resolver os problemas utilizando a forma padrão da PLI, que utiliza a mesma formulação acima, com a adição das denominadas variáveis de excesso em cada uma das limitações:

- Variáveis de excesso: S_i – Variáveis usadas para converter as limitações de igualdade em limitações de desigualdade em uma das formas de uma PLI.

Então, ainda segundo Daskin (1995), tem-se como forma padrão de uma PLI, a formulação da Equação (9).

$$\begin{array}{ll}
\text{MINIMIZAR} & \sum_j c_j X_j \\
\text{SUJEITO A} & \sum_j a_{ij} X_j - S_i = b_i, \quad \forall i \\
& X_j \geq 0, \quad \forall j \\
& S_i \geq 0, \quad \forall i
\end{array} \tag{9}$$

Muitos dos problemas que envolvam a programação linear envolvem a maximização dos objetivos, mas facilmente é possível converter para um objetivo de minimização ao multiplicar por -1 a função objetivo. O mesmo ocorre para as restrições que envolvam um limite inferior, onde é possível transformar em um limite superior ao multiplicar a restrição em questão por -1. No caso de restrições que envolvam igualdades, basta dividir esta restrição em duas partes: a primeira conta com um limite superior, igual ao valor da variável que antes definia a igualdade, e a segunda com um limite inferior, com a mesma variável. Por fim, existem problemas onde as suas variáveis não possuem restrição de sinal e, para que seja possível transformar estes problemas na forma canônica basta substituir as variáveis em questão pela diferença entre duas variáveis e aplicar a restrição de positividade à cada uma delas.

3.2.1. RESOLVEDOR CPLEX

Para auxiliar na otimização do problema do *Green Facility Location*, será utilizado também o resolvedor CPLEX, disponibilizado pela IBM, que fornece resolvedores de programação matemática flexíveis e de alto desempenho para problemas que possam ser otimizados utilizando PLI mista e pura, além de programação quadrática e programação quadraticamente restritos.

Segundo a empresa IBM (2021), o otimizador CPLEX conta com uma ferramenta importante para o desenvolvimento deste trabalho, chamada *Warm Start*, que é apontada como

uma devolução inicial, para que o algoritmo encontre uma resposta mais rapidamente, já que o resolvidor procura por respostas sem ponto de partida toda vez que é iniciado. O programa também conta com o fornecimento do número de camadas em que se deve buscar soluções – estas camadas são nada menos que o número total de ponto-soluções que serão encontrados ao decorrer de sua execução, onde há dois pontos na extremidade: os pontos utópicos para a otimização de cada objetivo, e um ponto normalizado na média entre cada um de outros dois pontos. O resolvidor CPLEX também permite a possibilidade de o programador fornecer o tempo máximo de sua execução, para que este não fique buscando por soluções por um tempo indeterminado.

4. ALGORITMO PARA O GFLP

Neste capítulo apresentamos a formulação em programação linear inteira para resolver o GFLP e o algoritmo para encontrar a fronteira de Pareto do problema.

4.1. FORMULAÇÃO MULTIOBJETIVO PARA O GFLP

A formulação do problema em Programação Linear Inteira se dá por duas funções objetivo, sendo a primeira (10) referente ao custo financeiro e a segunda (11) ao custo verde:

$$\text{MINIMIZAR} \quad \sum_{i \in I} cf_i * x_i + \sum_{i \in I} \sum_{j \in J} tf_{ij} * y_{ij}. \quad (10)$$

$$\text{MINIMIZAR} \quad \sum_{i \in I} cv_i * x_i + \sum_{i \in I} \sum_{j \in J} tv_{ij} * y_{ij}. \quad (11)$$

Ambas sujeitas a (12):

$$\begin{aligned} \text{SUJEITO A} \quad & \sum_{i \in I} y_{ij} = 1, & \forall j & \quad (12) \\ & \sum_{i \in I} y_{ij} \leq x_i * m & \forall i, \forall j \\ & y_{ij} \in \{0, 1\}, & \forall i, \forall j \\ & x_i \in \{0, 1\}, & \forall i. \end{aligned}$$

Sendo $I = \{0, 1, \dots, n\}$ o conjunto de potenciais instalações, o conjunto $J = \{0, 1, \dots, m\}$ de clientes, x_i uma variável binária para descrever se uma determinada instalação i será aberta ou não, y_{ij} uma variável binária que descreve se um determinado cliente j está ligado à instalação i , cf_i o custo financeiro de abertura da instalação i , cv_i o custo verde de abertura da instalação i , tf_{ij} o custo financeiro de transporte de uma instalação i até um determinado cliente j e por fim tv_{ij} o custo verde de transporte de uma instalação i até o cliente j .

De maneira a unir as funções objetivo (10) e (11), é estabelecido um peso α de forma a tornar a formulação mono objetivo. Sendo (10) representada por C_F e (11) por C_V , respectivamente o custo financeiro e verde total, pode-se escrever, portanto, a função objetivo da seguinte maneira, apresentada em (13):

$$\text{MINIMIZAR} \quad \alpha * C_F + (1 - \alpha) * C_V. \quad (13)$$

4.1.1. Warm Start

Para resolver a problemática apresentada neste trabalho, criou-se um algoritmo que transforma a equação do problema em Programação Linear Inteira (PLI), utilizando o resolvidor CPLEX da IBM para encontrar um conjunto de soluções ótimas ao longo da fronteira

de Pareto. Ainda, o resolvidor CPLEX conta com uma ferramenta chamada “*Warm Start*”, que permite, ao desenvolvedor, fornecer uma solução inicial ao programa, para que este possa se aproximar da resposta mais rapidamente.

O programa permite, ao usuário, fornecer o número de camadas que o algoritmo irá utilizar para encontrar as soluções. Este número de camadas irá auxiliar na geração de pesos para cada um dos custos, permitindo realizar até $2^n + 1$ subdivisões de pesos e, conseqüentemente, $2^n + 1$ conjuntos de soluções, onde n é o número de camadas fornecido; o algoritmo, ainda, pode encontrar um número de soluções menor, dependendo das entradas do problema e do tempo de solução fornecido ao resolvidor. Ademais, a fim de tornar a execução do algoritmo adaptável ao tempo disponível para calcular tais informações, é possibilitado, ao usuário, fornecer um tempo máximo de execução do resolvidor, permitindo, também, o algoritmo de encontrar melhores soluções ao decorrer de sua execução. Nota-se que um tempo maior de execução do resolvidor não necessariamente implicará em soluções melhores.

4.2. ALGORITMO PARA ENCONTRAR A FRONTEIRA DE PARETO

O algoritmo desenvolvido é implementado da seguinte forma: um vetor representa as instalações do problema e armazena valores booleanos para dizer se uma determinada instalação será aberta, com valor 1, ou não, com valor 0, em uma solução encontrada. Uma matriz de booleanos representa os clientes do problema, com as suas linhas representando a instalação e as colunas os clientes do problema; portanto, o conjunto linha-coluna diz se o cliente está ligado à instalação em questão, com 1 para mostrar que sim e 0 que não.

Após este passo, deve-se linearizar as variáveis de entrada do problema para fornecê-las ao *Warm Start* do resolvidor CPLEX; portanto, estas devem ser transformadas em um único vetor, que é constituído pelo conjunto Y, seguido do conjunto X.

As restrições do algoritmo são provenientes do *Green Facility Location*: a primeira, é a de que cada cliente deve estar conectado a exatamente uma instalação.

Já a segunda restrição, é a de que um cliente só pode estar conectado a uma instalação que esteja aberta. Existem duas variações implementadas para a segunda restrição: a primeira considera que, devido à sua implementação – que conta com um somatório, pode-se resultar em atribuições de valores muito próximos de 0 ou 1, e assim, encontrar soluções consideradas fracamente eficientes. Para tanto, a segunda variação desta restrição elimina tal possibilidade, adicionando ao resolvidor a restrição de que cada valor do ponto da matriz X deve ser menor, ou igual, ao valor do ponto do vetor Y. Isto é o mesmo que dizer que um cliente j , conectado à uma instalação i , implica que a instalação i deve estar aberta.

A implementação da função objetivo do custo financeiro das instalações, e seus respectivos custos de fornecimento de serviço para cada cliente, fora realizada a partir do seguinte raciocínio: o valor do custo de abertura da instalação é somado ao custo total financeiro, se esta for aberta, juntamente com o valor do custo de conexão de cada cliente, se este estiver conectado a uma determinada instalação. Criou-se uma variável para representar a função objetivo, a fim de facilitar a manipulação quando variar-se os pesos.

Para a função objetivo do custo verde, a implementação ocorre com o mesmo raciocínio, diferenciando-se no ponto que, neste caso, são dispostos os valores de custo verde ao invés do financeiro, e emprega-se outra variável.

4.2.1. Encontrando Valor Ideal e Valor Nadir

Inicialmente, deve-se encontrar os valores da solução ideal para que se possibilite criar o ponto Nadir, que, conforme abordado anteriormente, servirá para aproximar as soluções de maneira ótima na fronteira de Pareto. O ponto ideal considera que se deve otimizar apenas um dos objetivos, atribuindo um peso 0 para o outro. Isto é feito para ambos os objetivos, financeiro e verde, para então, calcular a pior e melhor solução para ambos. Caso o usuário deseje usar o *Warm Start* do resolvidor, o algoritmo armazenará um conjunto de soluções que o auxiliará a encontrar a melhor opção para resolver, a partir de uma solução inicial. Para o custo ideal verde, o mesmo procedimento acontece, com a única diferença na atribuição de um peso nulo para o custo financeiro. O código utilizado para encontrar os pontos Ideal e Nadir (APÊNDICE A) é descrito em seguida.

```
Busca-se uma solução S com o resolvidor CPLEX, com peso 1 para o custo financeiro e
peso 0 para o custo verde
custoNadirVerde <- custoVerde da solução S
custoIdealFinanceiro <- custoFinanceiro da solução S
O par de valores encontrado é adicionado ao vetor de soluções do Warm Start

Busca-se uma solução S' com o resolvidor CPLEX, com peso 0 para o custo financeiro e
peso 1 para o custo verde
custoIdealVerde <- custoVerde da solução S'
custoNadirFinanceiro <- custoFinanceiro da solução S'
Este par de valores é adicionado ao vetor de soluções do Warm Start
```

4.2.2. Geração de Pesos

Antes de encontrar os pontos ideais e pontos Nadir, cria-se uma fila de pesos com um par inicial $\{(1,0), (0,1)\}$ e duplas de pontos (p_F, p_V) , onde p_F é o peso para o custo financeiro

e p_V o peso para o custo verde. O algoritmo para a geração de novos pares (APÊNDICE A) é feito da seguinte maneira: dado um conjunto de pares $C_1 = \{(p_1, p_2), (p_3, p_4)\}$ que se encontra no começo da fila, uma nova dupla é criada calculando $d_1 = (\frac{p_1+p_3}{2}, \frac{p_2+p_4}{2})$, produzindo dois novos conjuntos $C_2 = \{(p_1, p_2), d_1\}$ e $C_3 = \{d_1, (p_3, p_4)\}$ que são inseridos ao final da fila. O conjunto C_1 é, então, removido da fila. O número máximo de pesos é baseado no número de camadas fornecido no início da execução do programa. Estes pesos são usados para que seja possível encontrar soluções enviesadas, considerando um peso maior para o custo financeiro e um menor para o verde, e vice-versa.

```

Seja numeroCamadas o número de camadas fornecido pelo usuário e filaPesos uma fila
em que cada elemento são dois pares de pesos, inicialmente com os valores { (1, 0) ,
(0, 1) }

Para i de 0 até (2^numeroCamadas - 1) faça
  ((a, b), (c, d)) <- primeiro(filaPesos)
  // O ponto (m1, m2) é a média entre os pontos (a, b) e (c, d)
  (m1, m2) <- ((a + c) / 2 , (b + d) / 2)
  push(filaPesos, ((a, b), (m1, m2)))
  push(filaPesos, ((m1, m2), (c, d)))

```

4.2.3. Resolvendo o Problema

Neste ponto, se foi escolhido utilizar o *Warm Start*, o algoritmo irá buscar a melhor solução encontrada até o momento e adicionar ao *Warm Start* para auxiliá-lo a encontrar a próxima solução.

A próxima etapa consiste em iniciar o resolvidor e adicionar o conjunto de soluções encontradas, custo verde e custo financeiro de abertura das instalações, e do providenciamento do serviço, à lista que contém todas as soluções já encontradas até o momento.

Ao final da execução, o programa imprime os conjuntos de resultados obtidos.

5. RESULTADOS E DISCUSSÕES

Este trabalho apresenta um problema real que organizações do terceiro setor vêm enfrentando ao longo dos últimos anos, porém, os dados utilizados foram criados para testar e validar o algoritmo, já que não foi possível encontrar, na literatura vigente, dados que representem a realidade.

Os dois conjuntos de dados utilizados como entrada para o algoritmo foram obtidos, primeiramente, a partir da disponibilização por Sarmet (2019), sendo fundamentados no custo financeiro do problema, contando com um conjunto de instalações, seus respectivos custos de abertura e os custos de fornecimento de seu serviço para todos os clientes do conjunto; já o segundo conjunto de dados, utiliza a mesma estrutura, sendo embasados no custo verde – emissão de poluentes, como gases do efeito estufa (GEE), de cada instalação e do fornecimento do serviço destas para cada cliente. Vale ressaltar que este segundo conjunto de dados foi gerado algoritmicamente, apoiando-se no conjunto de custo financeiro e, portanto, não são considerados valores reais.

A execução do algoritmo, conforme apresentado na metodologia, é condicionada a algumas restrições e opções; portanto, a fim de obter resultados variados e uma análise concisa e embasada destes, realizou-se testes com variações diferentes de tais condições. Para os testes do programa, utilizou-se um conjunto de dados que consiste em 250 instalações e 250 clientes, além dos custos financeiro e verde da abertura de cada uma das instalações e do providenciamento de serviço de cada instalação para cada cliente.

Foram utilizadas quatro variações diferentes para analisar como o algoritmo se comporta quando alguma alteração é feita, principalmente na maneira em como ele busca a sua solução: a primeira mudança é aplicada na decisão de se utilizar a restrição 2.1, onde o valor de um ponto i do vetor de instalações multiplicado pelo número total de clientes deve ser maior ou igual ao somatório da linha i da matriz de clientes, ou a restrição 2.2, onde o valor de cada ponto da linha i da matriz de clientes deve ser menor que o valor do ponto i do vetor de instalações; para a segunda variação, escolhe-se se o algoritmo pode, ou não, adicionar o *Warm Start* para auxiliá-lo a encontrar novas soluções; na terceira restrição, permite-se que o resolvidor procure por solução por período de tempo de até 60 ou 300 segundos; por último, escolhe-se se o programa irá encontrar soluções em até 2 ou 4 camadas, sendo a quantidade máxima de subdivisões dos pesos, que servem para auxiliar na geração de novos pesos. Com isso, para o primeiro grupo de conjunto de dados, que contém um total de 250 instalações e 250 clientes, obtém-se como resultado um total de 17 conjuntos de soluções. Na tabela gerada pelo algoritmo,

“i” representa a ordem em que o ponto foi gerado, “Verde” o custo verde e “Financeiro” o custo financeiro.

Primeiro, comparando quais diferenças são observadas ao alterar o tipo de restrição aplicada. Também será alterado o número de camadas utilizada pelo programa para a busca de soluções, a Figura 3 mostra os testes 1 (à esquerda, com a restrição 2.1) e 2 (à direita, com a restrição 2.2) para execução em 2 camadas; a Tabela 1 apresenta seus respectivos resultados, com a diferença absoluta entre os pontos gerados.

Figura 3: Resultado de execução dos Testes 1 e 2.

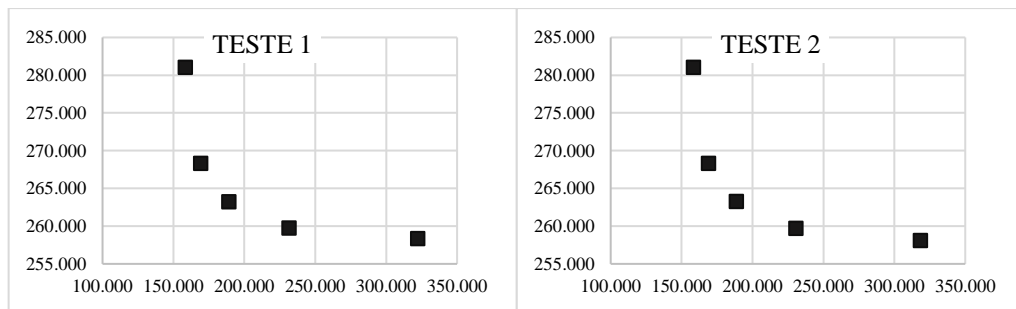


Tabela 1: Resultado dos testes 1 e 2, e a diferença entre os pontos gerados.

i	TESTE 1		TESTE 2		DIFERENÇA	
	Verde	Financeiro	Verde	Financeiro	Verde	Financeiro
0	322250	258362	318517	258092	3733	270
1	158442	281027	158442	281027	0	0
2	189082	263244	188759	263290	323	-46
3	231397	259747	230795	259694	602	53
4	169168	268301	169168	268301	0	0

Já a Figura 4 mostra os testes 3 (à esquerda, com a restrição 2.1) e 4 (à direita, com a restrição 2.2), para execução em 4 camadas. A Tabela 2 apresenta seus respectivos resultados, com a diferença absoluta entre os pontos gerados.

Figura 4: Resultado de execução dos Testes 3 e 4.

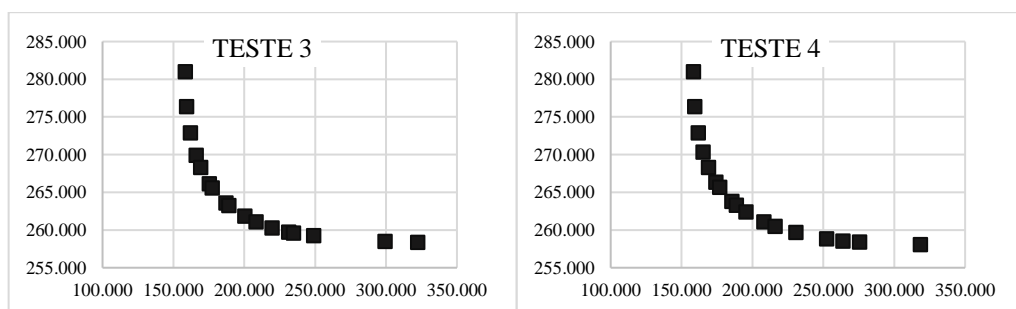


Tabela 2: Resultado dos testes 3 e 4, e a diferença entre os pontos gerados.

id	TESTE 3		TESTE 4		DIFERENÇA	
	Verde	Financeiro	Verde	Financeiro	Verde	Financeiro
0	322250	258362	318517	258092	3733	270
1	158442	281027	158442	281027	0	0
2	189082	263244	188759	263290	323	-46
3	231397	259747	230720	259694	677	53
4	169168	268301	169168	268301	0	0
5	248946	259263	263846	258559	-14900	704
6	208344	261094	208236	261103	108	-9
7	177345	265585	177037	265658	308	-73
8	161889	272882	161889	272882	0	0
9	299166	258499	275612	258439	23554	60
10	234631	259622	252508	258851	-17877	771
11	219466	260287	216171	260483	3295	-196
12	200382	261872	195454	262420	4928	-548
13	186989	263578	185675	263814	1314	-236
14	175205	266128	174409	266374	796	-246
15	165976	269915	165250	270359	726	-444
16	159337	276397	159337	276397	0	0

Nota-se que o número de camadas altera apenas o número de soluções obtidas pelo resolvidor – alguns valores são os mesmos, dada a mesma circunstância, o que mostra que o número de camadas serve apenas para encontrar mais soluções entre aquelas já encontradas. A conclusão do número de camadas é que esta opção serve para que os tomadores de decisão tenham mais escolhas ao realizar sua aplicação.

Nesta comparação também foi alterado o tipo de restrição que foi aplicada ao resolvidor – restrição 2.1 ou 2.2. Aqui nota-se uma diferença, mesmo que mínima, nos resultados obtidos. As soluções são similares, mas o resolvidor que utilizou a restrição 2.2 trouxe uma resposta melhor pois otimizou ainda mais uma das partes enquanto também otimizou a outra. Na Tabela 1 nota-se com mais clareza esta diferença – a iteração 10, de identificador 9, possui a maior diferença nos resultados, onde o resolvidor encontrou uma solução melhor na grandeza de 10^4 para o custo verde, enquanto ainda obteve uma solução melhor, mesmo que mínima, para o custo financeiro. É possível concluir disso que, de fato, a restrição 2.1 é mais fraca e com a integralidade relaxada devido ao uso de um somatório.

Agora será verificado quais mudanças ocorrem nos resultados quando o *Warm Start* é ativado. A Figura 5 mostra os testes 5 (à esquerda, com o *Warm Start* desligado) e 6 (à direita, com o *Warm Start* ligado) para execução em 2 camadas. A Tabela 3 apresenta seus respectivos resultados, com a diferença absoluta entre os pontos gerados.

Figura 5: Resultado de execução dos Testes 5 e 6.

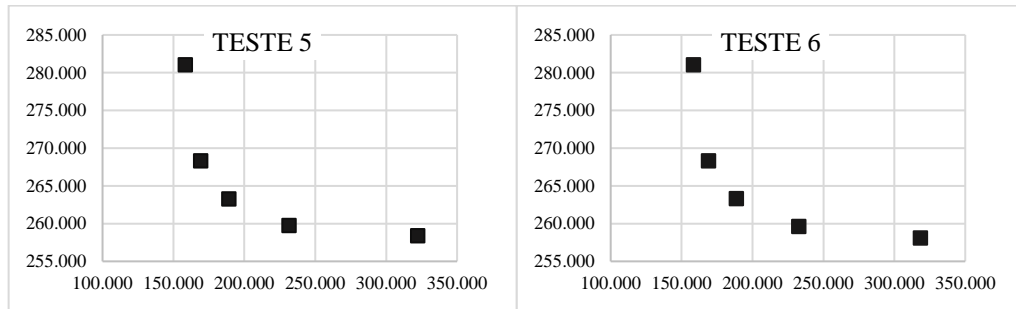


Tabela 3: Resultado dos testes 5 e 6, e a diferença entre os pontos gerados.

i	TESTE 5		TESTE 6		DIFERENÇA	
	Verde	Financeiro	Verde	Financeiro	Verde	Financeiro
0	322250	258362	322250	258362	0	0
1	158442	281027	158442	281027	0	0
2	189082	263244	189082	263244	0	0
3	231397	259747	231397	259747	0	0
4	169168	268301	169168	268301	0	0

A Figura 6, por sua vez, mostra os testes 7 (à esquerda, com o *Warm Start* desligado) e 8 (à direita, com o *Warm Start* ligado), para a execução em 4 camadas. A Tabela 4 apresenta seus respectivos resultados, com a diferença absoluta entre os pontos gerados.

Figura 6: Resultado de execução dos Testes 7 e 8.

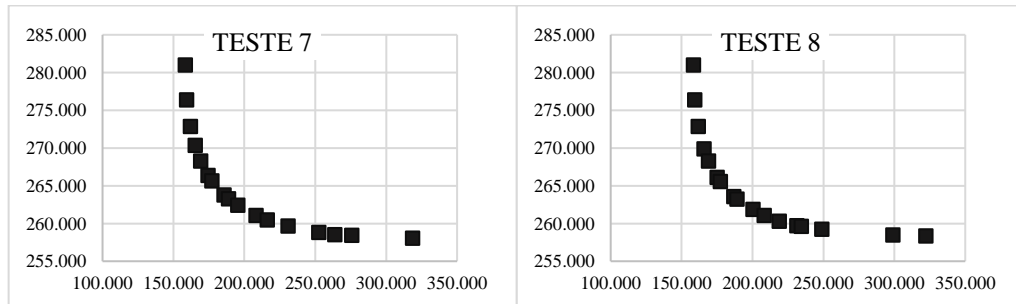


Tabela 4: Resultado dos testes 7 e 8, e a diferença entre os pontos gerados.

id	TESTE 7		TESTE 8		DIFERENÇA	
	Verde	Financeiro	Verde	Financeiro	Verde	Financeiro
0	322250	258362	322250	258362	0	0
1	158442	281027	158442	281027	0	0
2	189082	263244	189082	263244	0	0
3	231397	259747	231397	259747	0	0
4	169168	268301	169168	268301	0	0
5	248946	259263	248946	259263	0	0
6	208344	261094	208344	261094	0	0
7	177345	265585	177345	265585	0	0
8	161889	272882	161889	272882	0	0

9	299166	258499	299166	258499	0	0
10	234631	259622	234631	259622	0	0
11	219466	260287	218973	260305	493	-18
12	200382	261872	200382	261872	0	0
13	186989	263578	186989	263578	0	0
14	175205	266128	175205	266128	0	0
15	165976	269915	165976	269915	0	0
16	159337	276397	159337	276397	0	0

Alternar o *Warm Start* não trouxe os resultados esperados quando o resolvidor utilizou 2 camadas. Pode-se notar na Tabela 4 uma pequena mudança em apenas uma das soluções encontradas quando o resolvidor utiliza 4 camadas, mas nada muito aparente – enquanto com o *Warm Start* ligado foi possível otimizar o custo verde, o resolvidor não conseguiu otimizar o custo financeiro, portanto não se pode concluir qual seja melhor, já que ambas as soluções se encontram próximas a fronteira de Pareto.

Acreditou-se que ao aumentar o tempo que o resolvidor tinha disponível para que pudesse encontrar uma solução, ele iria encontrar soluções mais próximas da fronteira de Pareto, e isso pode ser observado nos gráficos abaixo, onde alterou-se o tempo disponível de 60 segundos, à esquerda, para 300 segundo. Na Figura 7 estão os testes realizados com 2 camadas e, na Figura 8, os testes realizados utilizando 4 camadas. A Tabela 5 e Tabela 6 apresentam seus respectivos resultados, com a diferença absoluta entre os pontos gerados.

Figura 7: Resultado de execução dos Testes 9 e 10.

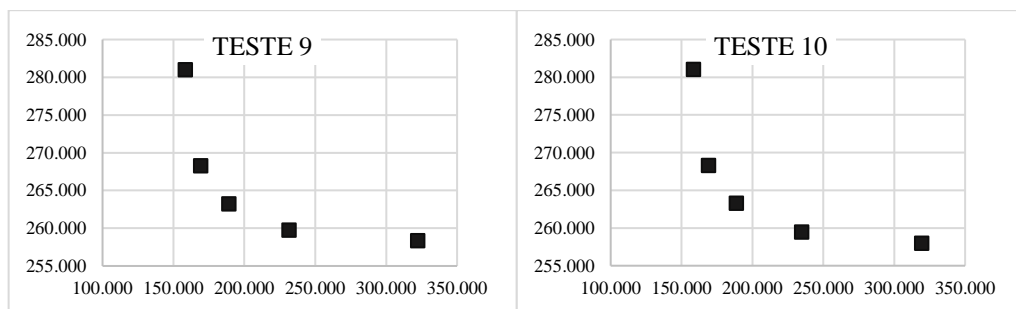


Tabela 5: Resultado dos testes 9 e 10, e a diferença entre os pontos gerados.

i	TESTE 9		TESTE 10		DIFERENÇA	
	Verde	Financeiro	Verde	Financeiro	Verde	Financeiro
0	322250	258362	319477	257990	2773	372
1	158442	281027	158442	281027	0	0
2	189082	263244	188759	263290	323	-46
3	231397	259747	234914	259485	-3517	262
4	169168	268301	169168	268301	0	0

Figura 8: Resultado de execução dos Testes 11 e 12.

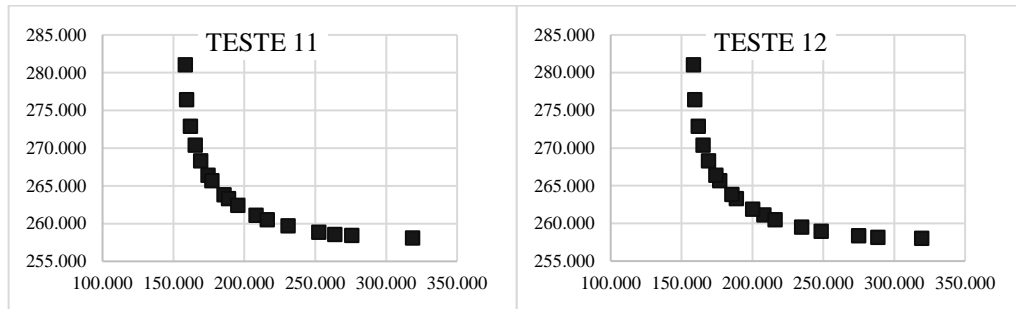
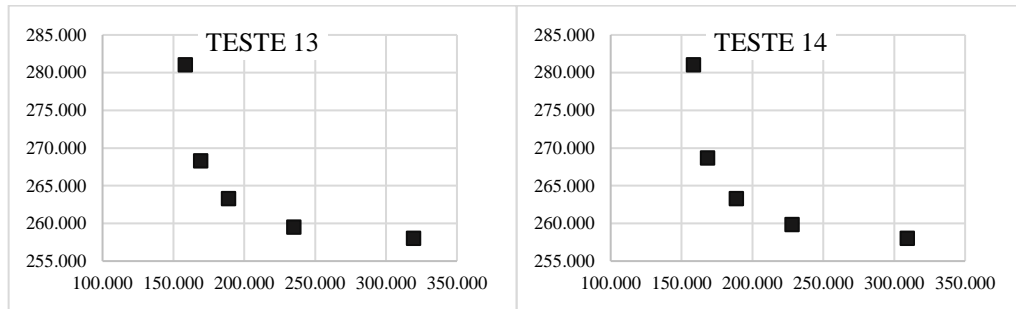


Tabela 6: Resultado dos testes 11 e 12, e a diferença entre os pontos gerados.

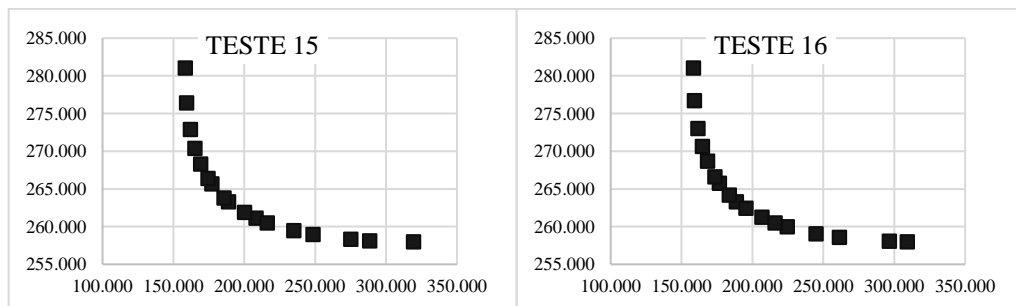
id	TESTE 11		TESTE 12		DIFERENÇA	
	Verde	Financeiro	Verde	Financeiro	Verde	Financeiro
0	322250	258362	319477	257990	2773	372
1	158442	281027	158442	281027	0	0
2	189082	263244	188759	263290	323	-46
3	231397	259747	234914	259485	-3517	262
4	169168	268301	169168	268301	0	0
5	248946	259263	275058	258338	-26112	925
6	208344	261094	208236	261103	108	-9
7	177345	265585	177037	265658	308	-73
8	161889	272882	161889	272882	0	0
9	299166	258499	288472	258112	10694	387
10	234631	259622	248543	258935	-13912	687
11	219466	260287	216171	260483	3295	-196
12	200382	261872	200235	261888	147	-16
13	186989	263578	185675	263814	1314	-236
14	175205	266128	174409	266374	796	-246
15	165976	269915	165250	270359	726	-444
16	159337	276397	159337	276397	0	0

Ao permitir mais tempo ao resolvidor para que este encontrasse uma solução se provou de fato de que ele iria encontrar soluções mais próximas da fronteira de Pareto. Algumas soluções conseguiram otimizar ambos os objetivos. Nota-se isso com mais clareza ao visualizar as tabelas que contém os dados com 4 camadas.

Como um tempo maior o resolvidor conseguiu obter melhores soluções, ele será mantido como base enquanto as outras opções serão alternadas. A Figura 9 demonstra os testes 13 (à esquerda, com a restrição 2.1) e 14 (à direita, com a restrição 2.2); já a Figura 10 apresenta os testes 15 (à esquerda, com a restrição 2.1) e 16 (à direita, com a restrição 2.2). A Tabela 7 e Tabela 8 apresentam seus respectivos resultados, com a diferença absoluta entre os pontos gerados.

Figura 9: Resultado de execução dos Testes 13 e 14.**Tabela 7: Resultado dos testes 13 e 14, e a diferença entre os pontos gerados.**

i	TESTE 13		TESTE 14		DIFERENÇA	
	Verde	Financeiro	Verde	Financeiro	Verde	Financeiro
0	319477	257990	309298	257991	10179	-1
1	158442	281027	158442	281027	0	0
2	188759	263290	188759	263290	0	0
3	234914	259485	227926	259828	6988	-343
4	169168	268301	168348	268669	820	-368

Figura 10: Resultado de execução dos Testes 15 e 16.**Tabela 8: Resultado dos testes 15 e 16, e a diferença entre os pontos gerados.**

i	TESTE 15		TESTE 16		DIFERENÇA	
	Verde	Financeiro	Verde	Financeiro	Verde	Financeiro
0	319477	257990	309298	257991	10179	-1
1	158442	281027	158442	281027	0	0
2	188759	263290	188759	263290	0	0
3	234914	259485	224676	259995	10238	-510
4	169168	268301	168348	268669	820	-368
5	275058	258338	261349	258564	13709	-226
6	208236	261103	206735	261235	1501	-132
7	177037	265658	176714	265736	323	-78
8	161889	272882	161751	273026	138	-144
9	288472	258112	296576	258055	-8104	57
10	248543	258935	244965	259057	3578	-122
11	216171	260483	216171	260483	0	0

12	200235	261888	195454	262420	4781	-532
13	185675	263814	183777	264168	1898	-354
14	174409	266374	173731	266590	678	-216
15	165250	270359	164816	270635	434	-276
16	159337	276397	159194	276711	143	-314

Agora, ao alternar apenas entre o tipo de restrição usado, nota-se uma diferença maior entre os pontos obtidos – as soluções encontradas pelo resolvidor que usou a restrição 2.1 são menos otimizadas por aquele que utilizaram a restrição 2.2, como observadas na Tabela 7 e na Tabela 8, o que reforça ainda mais a ideia de que o somatório usado pela restrição 2.1 torna a sua integralidade relaxada.

Por último serão verificadas as mudanças ocorridas ao ativar ou desativar o *Warm Start* do resolvidor, mantendo o uso da restrição 2.2 e permitindo um tempo de execução de até 300 segundos para o resolvidor.

A Figura 11 apresenta os testes 17 (à esquerda, com *Warm Start* desligado) e 18 (à direita, com *Warm Start* ligado), com duas camadas; a Figura 12 apresenta os testes 19 (à esquerda, com *Warm Start* desligado) e 20 (à direita, com *Warm Start* ligado), com quatro camadas. A Tabela 9 e Tabela 10 apresentam seus respectivos resultados, com a diferença absoluta entre os pontos gerados.

Figura 11: Resultado de execução dos Testes 17 e 18.

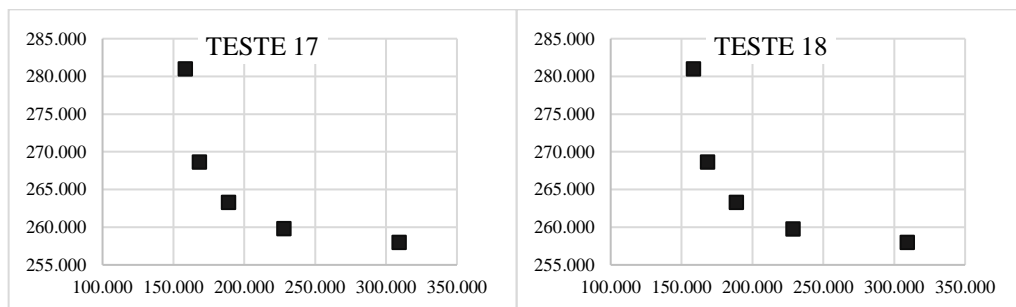


Tabela 9: Resultado dos testes 17 e 18, e a diferença entre os pontos gerados.

i	TESTE 17		TESTE 18		DIFERENÇA	
	Verde	Financeiro	Verde	Financeiro	Verde	Financeiro
0	309298	257991	309298	257991	0	0
1	158442	281027	158442	281027	0	0
2	188759	263290	188759	263290	0	0
3	227926	259828	228660	259790	-734	38
4	168348	268669	168348	268669	0	0

Figura 12: Resultado de execução dos Testes 19 e 20.

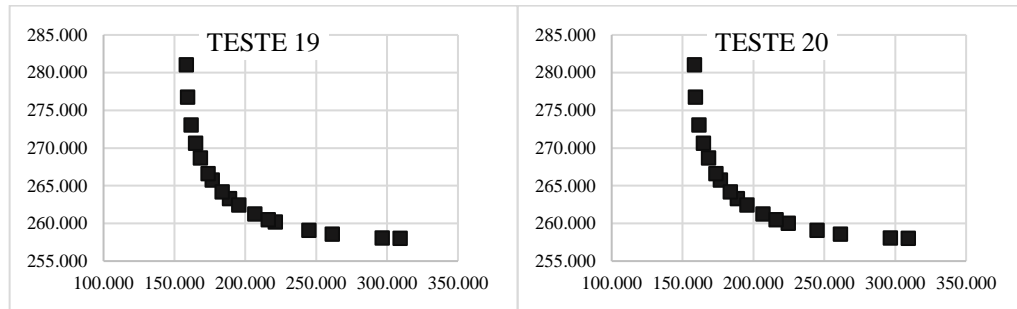


Tabela 10: Resultado dos testes 19 e 20, e a diferença entre os pontos gerados.

i	TESTE 19		TESTE 20		DIFERENÇA	
	Verde	Financeiro	Verde	Financeiro	Verde	Financeiro
0	309298	257991	309298	257991	0	0
1	158442	281027	158442	281027	0	0
2	188759	263290	188759	263290	0	0
3	221213	260173	224676	259995	-3463	178
4	168348	268669	168348	268669	0	0
5	261349	258564	261349	258564	0	0
6	206735	261235	206735	261235	0	0
7	176714	265736	176714	265736	0	0
8	161751	273026	161751	273026	0	0
9	296576	258055	296576	258055	0	0
10	244965	259057	244965	259057	0	0
11	216171	260483	216171	260483	0	0
12	195454	262420	195454	262420	0	0
13	183777	264168	183777	264168	0	0
14	173731	266590	173731	266590	0	0
15	164816	270635	164816	270635	0	0
16	159194	276711	159194	276711	0	0

Não é possível notar uma mudança significativa no uso do *Warm Start*, o que reforça que o seu uso não altera muito as soluções obtidas. Novamente, para ambos os conjuntos de tabelas, com 2 ou 4 camadas, apenas uma solução apresentou mudança na ordem de 10^3 , o que torna o *Warm Start* não tão relevante para obter soluções mais eficientes.

Por fim, realizou-se comparações entre testes que possuíam características opostas, exceto a quantidade de camadas. Para o teste 21, utilizou-se a restrição 2.1, com o *Warm Start* desligado e tempo de 60 segundos; já para o teste 22, utilizou-se a restrição 2.2, com o *Warm Start* ligado e tempo de 300 segundos. A Figura 13 ilustra a diferença visual entre eles, e a Tabela 11 apresenta seus respectivos resultados, com a diferença absoluta entre os pontos gerados.

Figura 13: Resultado de execução dos Testes 21 e 22.

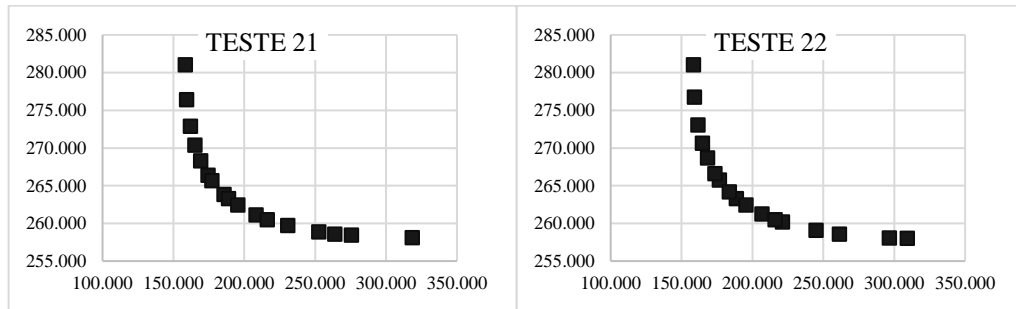


Tabela 11: Resultado dos testes 21 e 22, e a diferença entre os pontos gerados.

i	TESTE 21		TESTE 22		DIFERENÇA	
	Verde	Financeiro	Verde	Financeiro	Verde	Financeiro
0	322250	258362	309298	257991	12952	371
1	158442	281027	158442	281027	0	0
2	189082	263244	188759	263290	323	-46
3	231397	259747	221213	260173	10184	-426
4	169168	268301	168348	268669	820	-368
5	248946	259263	261349	258564	-12403	699
6	208344	261094	206735	261235	1609	-141
7	177345	265585	176714	265736	631	-151
8	161889	272882	161751	273026	138	-144
9	299166	258499	296576	258055	2590	444
10	234631	259622	244965	259057	-10334	565
11	219466	260287	216171	260483	3295	-196
12	200382	261872	195454	262420	4928	-548
13	186989	263578	183777	264168	3212	-590
14	175205	266128	173731	266590	1474	-462
15	165976	269915	164816	270635	1160	-720
16	159337	276397	159194	276711	143	-314

Nota-se, na Tabela 11, que as diferenças entre os testes chegaram até a grandeza de 10^5 , o que reforça a discussão sobre os resultados obtidos até então: a escolha das opções influencia, consideravelmente, os resultados. A maior diferença, quanto ao custo verde, é notada logo na primeira iteração, que, no teste 22, conseguiu uma solução com um custo verde muito menor comparado com o teste 21, enquanto conseguiu minimizar ainda mais o custo financeiro.

6. CONCLUSÕES

Este trabalho apresentou uma pesquisa importante para quaisquer organizações que prestem serviços as suas comunidades e buscam encontrar as melhores soluções possíveis para que estas possam reduzir o custo financeiro da alocação de suas instalações e do providenciamento do seu serviço aos seus clientes, além também, é claro, reduzir a emissão de poluentes seja da instalação quanto do transporte do produto. Também deve-se fixar que o trabalho busca encontrar não uma, já que isso seria utópico, mas um conjunto de soluções para que os tomadores de decisão da organização possam escolher a melhor dentre várias opções para alocar suas instalações.

O algoritmo construído sofreu diversas modificações ao longo de sua construção e acabou se tornando robusto o suficiente para encontrar soluções mais próximas da fronteira de Pareto. Alterar uma das restrições mostrou que o algoritmo irá encontrar soluções melhores. Independentemente do número de camadas, alguns dos valores obtidos serão sempre os mesmos, alterar este número afeta apenas na quantidade de conjunto de soluções que o usuário irá obter ao final da execução do programa. Uma das maiores divergências na concepção deste trabalho foi que se acreditou que o *Warm Start* teria um impacto maior ao encontrar as soluções.

O algoritmo requer uma implementação melhor em como lida com os arquivos de entrada, já que os que foram utilizados possuem uma descrição das variáveis muito específica, portanto uma formalização destes dados de entrada é necessária para que o algoritmo seja utilizado por um usuário comum.

Espera-se que o trabalho desenvolvido seja útil para as organizações do terceiro setor no futuro, já que a preocupação com o meio ambiente vem se tornando um assunto cada vez mais relevante no dia a dia de todos.

REFERÊNCIAS

- AFSHARI, H. *et al.* Optimizing Multi-objective Dynamic Facility Location Decisions within Green Distribution Network Design. **Procedia CIRP**, Amsterdam, v. 17, ed. 47, p. 675-679, 11 jul. 2014. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2212827114004260>. Acesso em: 9 jun. 2021.
- ANTUNES, C. H.; ALVES, M. J. Programação Linear Multiobjetivo – Métodos Interativos e Software. *In*: CONGRESSO LATINO-IBEROAMERICANO DE INVESTIGACIÓN OPERATIVA E SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 16. e 44., 2012, São Paulo. **Anais eletrônicos**[...]. São Paulo: Sociedade Brasileira de Pesquisa Operacional (SOBRAPO), 2012. Disponível em: <http://www.din.uem.br/~ademir/sbpo/sbpo2012/pdf/arq0127.pdf>. Acesso em: 15 nov. 2021
- ARCENCIO, G.G., **Programação Linear Inteira e o Problema do Roteamento de Veículos**. 2021. Relatório Final (Iniciação Científica) – Processo FAPESP #2020/06105-2 – Universidade Federal de São Carlos, São Carlos, 2021.
- ARROYO, J.E.C. **Heurísticas e Metaheurísticas para Otimização Combinatória Multiobjetivo**. Campinas, 2002. Tese (Doutorado em Engenharia Elétrica). Faculdade de Engenharia Elétrica e Computação. Universidade Estadual de Campinas, Campinas, 2002.
- BOUCHERY, Y. *et al.* **Sustainable Supply Chains: A Research-Based Textbook on Operations and Strategy**. [S. l.]: Springer, 2017. v. 4.
- BRANSKI, R. M. **O papel da Tecnologia da Informação no processo logístico: estudo de casos com operadores logísticos**. 2008. Tese (Doutorado em Engenharia de Produção) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2008. Disponível em: <https://www.teses.usp.br/teses/disponiveis/3/3136/tde-01102008-144646/en.php>. Acesso em: 4 jun. 2021.
- COELLO, C. A. C.; DHAENENS, C.; JOURDAN, L. Multi-Objective Combinatorial Optimization: Problematic and Context. *In*: COELLO, C. A. C.; DHAENENS, C.; JOURDAN, L. **Advances in Multi-Objective Nature Inspired Computing**. [S. l.]: Springer, 2009. v. 272, cap. 1, p. 1-23.
- DASKIN, M. S. *et al.* **Network and Discrete Location: Models, Algorithms and Applications**. Toronto: Wiley, 1995.
- FARAHANI, R. Z.; STEADIESEIFI, M.; ASGARI, N. Multiple criteria facility location problems: A survey. **Applied Mathematical Modelling**, Swansea, v. 34, n. 7, p. 1689-1964, 2 jul. 2010. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0377221703008191>. Acesso em: 9 jun. 2021.
- GOETZINGER, M.; BRANDT, T.; NEUMANN, D. Green Facility Location – A Case Study. *In*: AMCIS 2012 PROCEEDINGS, 17., 2012, Seattle. **Anais eletrônicos**[...]. Seattle: AMCIS, 2012. Disponível em: <https://aisel.aisnet.org/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1157&context=amcis2012>. Acesso em: 19 maio 2021

HARRIS, I. et al. Assessing the impact of cost optimization based on infrastructure modelling on CO₂ emissions. **International Journal of Production Economics**, Munchen, v. 131, n. 1, p. 313-321, 1 mai. 2011. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925527310000873>. Acesso em: 4 mai. 2021.

HARRIS, I.; MUMFORD, C. L.; NAIM, M. M. A hybrid multi-objective approach to capacitated facility location with flexible store allocation for green logistics modeling. **Transportation Research Part E: Logistics and Transportation Review**, Taipei, v. 66, p. 1-22, 1 jun. 2014. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1366554514000179>. Acesso em: 4 mai. 2021.

IBM. **IBM CPLEX Optimizer**: Resolvedor de programação matemática de alto desempenho para programação linear, programação inteira mista e programação quadrática. [S. l.], 2021. Disponível em: <https://www.ibm.com/br-pt/analytics/cplex-optimizer>. Acesso em: 12 maio 2021.

JAIN, K. Greedy facility location algorithms analyzed using dual fitting with factor -: revealing LP. **Journal of the ACM**, New York, v. 50, n. 6, p. 795-824, 2003. Disponível em: <https://dl.acm.org/doi/10.1145/950620.950621>. Acesso em: 12 maio 2021.

KLOSE, A.; DREXL, A. Facility location models for distribution system design. **European Journal of Operational Research**, Poznan, v. 162, n. 3, p. 4-29, 1 abr. 2005. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0377221703008191>. Acesso em: 9 jun. 2021.

MALLIDIS, I.; VLACHOS, D.; DEKKER, R. Greening Supply Chains: Impact on Cost and Design. **Econometric Institute Research Papers EI 2010-39a**, Erasmus University Rotterdam, Erasmus School of Economics (ESE), Econometric Institute, 16 ago. 2010. Disponível em: https://www.researchgate.net/publication/46434297_Greening_Supply_Chains_Impact_on_Cost_and_Design. Acesso em: 12 out. 2021.

MARLER, T. R.; ARORA, J. S. The weighted sum method for multi-objective optimization: new insights. **Structural and Multidisciplinary Optimization**, Iowa, v. 41, p. 853-862, 1 jun. 2010. Disponível em: <https://www.researchgate.net/publication/225485886>. Acesso em: 4 mai. 2021.

OBSERVATÓRIO DO CLIMA. **Análise das emissões brasileiras de Gases de Efeito Estufa e suas implicações para as metas de clima do Brasil 1970-2019**. SEEG (Sistema de Estimativa de Emissões de Gases de Efeito Estufa), 2020. Disponível em: https://seeg-br.s3.amazonaws.com/Documentos%20Analiticos/SEEG_8/SEEG8_DOC_ANALITICO_SI_NTESE_1990-2019.pdf. Acesso em: 15 nov. 2021.

PIKE-BURKE, C. **Multi-Objective Optimization**. 2019. STOR601: Research Topic I, London. Disponível em: https://scholar.google.com/citations?view_op=view_citation&hl=en&user=H11vu1MAAAAJ&citation_for_view=H11vu1MAAAAJ:u-x6o8ySG0sC. Acesso em: 21 maio 2021.

SARMET, R. **ICAlgoritmos**. [S. l.], 2019. Disponível em: <https://github.com/renatasarmet/ICAlgoritmos>. Acesso em: 12 maio 2021.

SILVA NETO, G.C., **Um Método para Solução de Problemas de Otimização Multiobjetivo em Ambiente Fuzzy**. 2011. Dissertação (Mestrado em Matemática) – PPGM – Universidade Federal do Amazonas, Manaus, 2011.

WOLSEY, L. A. **Integer Programming**. 2. ed. Hoboken: Wiley, 2021.

WORLD HEALTH ORGANIZATION. **WHO global air quality guidelines: particulate matter (PM_{2.5} and PM₁₀), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide**. Geneva: WHO, 2021. Disponível em: <https://apps.who.int/iris/handle/10665/345329>. Acesso em: 15 nov. 2021.

WWF-BRASIL. **Pegada Ecológica: Nosso estilo de vida deixa marcas no planeta**. Brasília: WWF, 2013. Disponível em: https://www.wwf.org.br/natureza_brasileira/especiais/pegada_ecologica/?35722/Cartilha-Pegada-Ecologica. Acesso em: 15 nov. 2021.

APÊNDICE A – Implementação em C++

Seguem abaixo as implementações em C++ do algoritmo.

Criação das variáveis do problema:

```
IloIntVarArray Y(env, num_facilities, 0, 1);
IloIntVarArray2 X(env);

for(int i = 0; i < num_facilities; i++){
    X.add(IloIntVarArray(env, num_customers, 0, 1));
}
```

Linearização das variáveis do problema:

```
IloNumVarArray variaveis(env);

for(int i = 0; i < num_facilities; i++) {
    variaveis.add(Y[i]);
}

for(int i = 0; i < num_facilities; i++) {
    for(int j = 0; j < num_customers; j++) {
        variaveis.add(X[i][j]);
    }
}
```

Primeira restrição do problema:

```
for (int j = 0; j < num_customers; j++) {
    IloExpr cliente_conexao(env);
    for (int i = 0; i < num_facilities; i++) {
        cliente_conexao += X[i][j];
    }
    modelo.add(cliente_conexao == 1);
}
```

Segunda restrição do problema com suas duas variantes:

```
// Restricao 2.1
for (int i = 0; i < num_facilities; i++) {
    modelo.add(IloSum(X[i]) <= num_customers * Y[i]);
}

// Restricao 2.2
for (int i = 0; i < num_facilities; i++) {
    for (int j = 0; j < num_customers; j++) {
        modelo.add(X[i][j] <= Y[i]);
    }
}
```

Função objetivo do custo financeiro:

```
IloExpr objFin(env);
IloNumVar custoFinanceiro(env);

for (int i = 0; i < num_facilities; i++) {
    objFin += custoAbertura[i] * Y[i];
    for (int j = 0; j < num_customers; j++) {
        objFin += custoConexao[i][j] * X[i][j];
    }
}

modelo.add(custoFinanceiro == objFin);
```

Função objetivo do custo verde:

```
IloExpr objVerde(env);
IloNumVar custoVerde(env);

for (int i = 0; i < num_facilities; i++) {
    objVerde += custoAberturaVerde[i] * Y[i];
    for (int j = 0; j < num_customers; j++) {
        objVerde += custoConexaoVerde[i][j] * X[i][j];
    }
}

modelo.add(custoVerde == objVerde);
```

Buscar a melhor solução encontrada até o momento para ser usada pelo Warm Start:

```
modelo.remove(minimizar);
minimizar = IloMinimize(env, (media.first * custoFinanceiroNormalizado
    + media.second * custoVerdeNormalizado));
modelo.add(minimizar);

resolvedor.setParam(IloCplex::TiLim, tempoLimiteResolvedor);

if (ligaWarmStart) {
double minValue = (solucoesWarmStart[0].custoFinanceiro - custoIdealFinanceiro)
    / (custoNadirFinanceiro - custoIdealFinanceiro) * media.first
    + (solucoesWarmStart[0].custoVerde - custoIdealVerde)
    / (custoNadirVerde - custoIdealVerde) * media.second;
int minPos = 0;

for (int i = 1; i < solucoesWarmStart.size(); i++) {
    double value = (solucoesWarmStart[i].custoFinanceiro
        - custoIdealFinanceiro)
        / (custoNadirFinanceiro - custoIdealFinanceiro) * media.first
        + (solucoesWarmStart[i].custoVerde
        - custoIdealVerde)
        / (custoNadirVerde - custoIdealVerde) * media.second;

    if (value < minValue) {
        minValue = value;
        minPos = i;
    }
}

solucoesAux = solucoesWarmStart[minPos];

resolvedor.addMIPStart(variaveis, solucoesAux.valoresSolucao);
}
```

Iniciar o resolvedor e adicionar a solução ao conjunto de soluções obtidas:

```

if (resolvedor.solve()) {
    custosSolucoes.push_back(make_pair(resolvedor.getValue(custoVerde),
resolvedor.getValue(custoFinanceiro)));

    if (ligaWarmStart) {

        solucoesAux.chave = media.first;
        solucoesAux.custoFinanceiro = resolvedor.getValue(custoFinanceiro);
        solucoesAux.custoVerde = resolvedor.getValue(custoVerde);
        solucoesAux.valoresSolucao = IloNumArray(env);

        for(int i = 0; i < num_facilities; i++) {

            solucoesAux.valoresSolucao.add(resolvedor.getValue(Y[i]));
        }

        for(int i = 0; i < num_facilities; i++) {
            for(int j = 0; j < num_customers; j++) {

                solucoesAux.valoresSolucao.add(resolvedor.getValue(X[i][j]));
            }
        }

        solucoesWarmStart.push_back(solucoesAux);
    }
}

```

Criação da Fila de Pesos:

```

for (int i = 0; i < (pow(2, numeroCamadas) - 1); i++) {
    parAux = filaPesos.front();
    filaPesos.pop();
    media = make_pair(
        (parAux.first.first + parAux.second.first) / 2.0 ,
        (parAux.first.second + parAux.second.second) / 2.0);
    filaPesos.push({ parAux.first, media });
    filaPesos.push({ media, parAux.second });
}

```

Encontrando o Ponto Ideal e Ponto Nadir:

```

vector<Solucao> solucoesWarmStart;
Solucao solucoesAux;

double custoNadirVerde, custoNadirFinanceiro;
double custoIdealVerde, custoIdealFinanceiro;

IloObjective minimizar = IloMinimize(env, custoFinanceiro);
modelo.add(minimizar);
IloCplex resolvedor(env);
resolvedor.setParam(IloCplex::TiLim, tempoLimiteResolvedor);
resolvedor.extract(modelo);
resolvedor.solve();
custoNadirVerde = resolvedor.getValue(custoVerde);
custoIdealFinanceiro = resolvedor.getValue(custoFinanceiro);
custosSolucoes.push_back(
make_pair(resolvedor.getValue(custoVerde), resolvedor.getValue(custoFinanceiro)));

if (ligaWarmStart) {
    solucoesAux.chave = 1;
    solucoesAux.custoFinanceiro = custoIdealFinanceiro;
    solucoesAux.custoVerde = custoNadirVerde;
    solucoesAux.valoresSolucao = IloNumArray(env);

    for(int i = 0; i < num_facilities; i++) {
        solucoesAux.valoresSolucao.add(resolvedor.getValue(Y[i]));
    }

    for(int i = 0; i < num_facilities; i++) {
        for(int j = 0; j < num_customers; j++) {
            solucoesAux.valoresSolucao.add(resolvedor.getValue(X[i][j]));
        }
    }

    solucoesWarmStart.push_back(solucoesAux);
}

modelo.remove(minimizar);
minimizar = IloMinimize(env, custoVerde);
modelo.add(minimizar);
resolvedor.setParam(IloCplex::TiLim, tempoLimiteResolvedor);
resolvedor.solve();
custoNadirFinanceiro = resolvedor.getValue(custoFinanceiro);
custoIdealVerde = resolvedor.getValue(custoVerde);
custosSolucoes.push_back(make_pair(resolvedor.getValue(custoVerde),
resolvedor.getValue(custoFinanceiro)));

if (ligaWarmStart) {
    solucoesAux.chave = 0;

```

```
solucoesAux.custoFinanceiro = custoNadirFinanceiro;
solucoesAux.custoVerde = custoIdealVerde;
solucoesAux.valoresSolucao = IloNumArray(env);

for(int i = 0; i < num_facilities; i++) {
    solucoesAux.valoresSolucao.add(resolvedor.getValue(Y[i]));
}

for(int i = 0; i < num_facilities; i++) {
    for(int j = 0; j < num_customers; j++) {
        solucoesAux.valoresSolucao.add(resolvedor.getValue(X[i][j]));
    }
}

solucoesWarmStart.push_back(solucoesAux);
}
```