

Universidade Federal de Minas Gerais
Escola de Engenharia
Curso de Graduação em Engenharia de Controle e Automação

**Planejamento da Colheita de Eucalipto:
Uma Abordagem Via Meta-heurística**

Anilton Cardoso Junior

Orientador: Prof. Lucas de Souza Batista, Dr.
Supervisor: Prof. Ronaldo Santos Sampaio, Dr.

Belo Horizonte, Dezembro de 2015

Monografia

Planejamento da Colheita de Eucalipto: Uma Abordagem Via Metaheurística

Monografia submetida à banca examinadora designada pelo Colegiado Didático do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Minas Gerais, como parte dos requisitos para aprovação na disciplina Projeto Final de Curso II.

Belo Horizonte, Novembro de 2015

Resumo

A Vallourec Florestal, subsidiária do Grupo Vallourec no Brasil, produz carvão vegetal, alimentando os alto-fornos para produção de ferro-gusa, matéria-prima do aço. Contando com mais de 100 mil hectares de florestas de eucalipto, a Vallourec Florestal precisa gerenciar seus recursos de maneira eficiente para que se atenda as demandas dinâmicas de carvão vegetal. Assim, a empresa necessita de pessoal especializado e vem investindo em sistemas de suporte à decisão para gerenciar seus processos.

Este trabalho faz parte dessa iniciativa para o setor de colheita da empresa. Sendo milhares de hectares de floresta e diversas máquinas de corte de madeira para operação paralela, o planejamento de colheita é uma atividade complexa realizada manualmente e sujeita a erros. Dessa forma, desenvolvem-se algoritmos de otimização para planejamento de colheita operacional, i.e. planejamento de curto prazo. À vista disso, requer-se que o método de otimização desenvolvido seja eficiente e rápido com baixo tempo de computação. Portanto, nesse trabalho, realiza-se um estudo das principais variáveis e restrições e uma modelagem matemática e computacional do processo. A partir disso, algumas meta-heurísticas são escolhidas para resolver o problema e, finalmente, são feitas análises dos resultados obtidos pelo método desenvolvido.

Agradecimentos

À Deus, por permitir que essa vitória fosse possível. Aos meus pais, dedico este trabalho e minha graduação pelo seu apoio incondicional. À Thais, por ouvir tanto sobre engenharia e conquistar tudo isso comigo. À minha família, o bem mais valioso que tenho. Ao meu professor orientador, Dr. Lucas Batista, por toda disposição, ajuda e por exercer verdadeiramente o papel de professor. Ao professor, mestre e catalisador, Dr. Ronaldo Sampaio, pelos incontáveis ensinamentos e por me provocar sempre a ser melhor como pessoa e como profissional. Aos meus grandes amigos que fiz nestes cinco anos de engenharia que levarei comigo por toda vida. À equipe DP, por ser a melhor equipe, a qual tive o prazer de fazer parte. À Vallourec, especialmente à Vallourec Florestal, por estes dois anos de estágio e pelo intenso aprendizado.

Sumário

Resumo	i
Agradecimentos	iii
Lista de Figuras	vii
Lista de Tabelas	ix
1 Introdução	1
1.1 Motivação e Justificativa	1
1.1.1 A VFL e Demanda de Carvão Vegetal nos Alto-Fornos	1
1.2 Objetivos do Projeto	3
1.3 Contribuições do Projeto	3
1.4 Setores Envolvidos no Projeto	3
1.5 Estrutura da Monografia	4
2 Revisão Bibliográfica	5
2.1 Técnicas de Otimização	5
2.1.1 Dimensionamento de Lotes e Sequenciamento	6
2.1.2 Meta-heurísticas	7
2.1.3 Otimização Multiobjetivo	9
2.1.4 Otimização Robusta	11
2.2 Planejamento Florestal	12
2.3 Conclusão	13
3 Metodologia	14
3.1 Modelagem Matemática do Planejamento de Colheita	14
3.1.1 Características do Problema de Otimização	14
3.1.2 Funções Objetivo	16
3.1.3 Restrições	17
3.2 Algoritmos de Otimização	19
3.2.1 GRASP	19
3.2.2 Abordagem Escalar Via Método ϵ -restrito e GRASP	25
3.2.3 Análise de Robustez	25
3.3 Estrutura de Testes	26
3.4 Dados de Entrada	26
3.5 Parâmetros Utilizados	27

3.5.1	Modelo	27
3.5.2	Parâmetros da Construção	28
3.5.3	Busca Local	28
3.5.4	Métodos Escalares e Robustez	28
4	Resultados e Discussão	29
4.1	Métodos Escalares	29
4.1.1	Análise de F_1 e F_2 pelo Método da Soma Ponderada	29
4.1.2	Análise de F_{transp} e F_3 pelo Método ϵ -restrito	30
4.2	Qualidade das Soluções	31
4.2.1	Robustez	31
4.2.2	Outras características	32
4.3	Análises Estatísticas das Execuções	33
5	Conclusões	35
5.1	Contribuições do Trabalho	36
5.2	Sugestões para trabalhos futuros	36
5.3	Considerações Finais	37
	Referências Bibliográficas	38

Lista de Figuras

1.1	Disposição das Regionais da VFL	2
2.1	Espaço de Decisão e de Objetivos	10
2.2	Método ϵ -restrito para problema biobjetivo	11
3.1	Estrutura da Solução	15
3.2	Exemplo de Sequenciamento	15
3.3	Estrutura da Solução considerando plantas de carbonização	16
3.4	Sequenciamento de talhões: prioridade por direção	20
3.5	Encontrar ângulo para cálculo de F_g	21
3.6	N_1 – Trocar talhões	22
3.7	N_1 – Trocar Fazendas	23
3.8	N_1 – Troca de períodos	23
3.9	Exemplo de operações de troca para períodos diferentes	23
3.10	N_2 – Deslocar talhão	24
3.11	N_3 – Inverter talhões	24
3.12	N_4 – Inserir talhão	24
3.13	N_5 – Remoção de talhão	25
3.14	N_6 – Trocar destino do talhão	25
4.1	Aproximação à Fronteira Pareto-ótima para F_1 e F_2	30
4.2	Novos testes para Método da Soma Ponderada	30
4.3	Aproximação à Fronteira Pareto-ótima para F_{transp} e F_3	31
4.4	Comparações no uso de máquina médio	32
5.1	Software desenvolvido	37

Lista de Tabelas

2.1	Dimensionamento de Lotes: Variáveis de Decisão	6
2.2	Dimensionamento de Lotes: Parâmetros	6
3.1	Talhões por fazenda	26
3.2	Demandas de madeira	27
4.1	Comparação de robustez	31
4.2	Tempo de estoque médio em meses	32
4.3	Análise de execução do algoritmo	33

Capítulo 1

Introdução

Em um mercado cada vez mais competitivo, as empresas necessitam inovar e investir na qualidade e melhoria de seus processos de produção. Nesse quesito, graças ao aumento do poder computacional nas últimas décadas, os algoritmos de otimização ganharam espaço nas indústrias por serem capazes de lidar com modelos que incorporam e tratam um grande número de variáveis, representando, assim, os processos de forma mais fiel e gerando resultados práticos atrativos [32]. Este trabalho visa a aplicação de algoritmos de otimização para aprimorar o sequenciamento do corte de eucalipto da Vallourec Florestal (VFL), empresa produtora de carvão vegetal para uso siderúrgico.

1.1 Motivação e Justificativa

1.1.1 A VFL e Demanda de Carvão Vegetal nos Alto-Fornos

O grupo Vallourec, no Brasil, produz tubos de aço sem costura para atender o mercado nacional e internacional de soluções tubulares *Premium*. Suas duas usinas no país contam com três alto-fornos para produção de ferro gusa, a matéria prima do aço mais importante para o grupo no Brasil. Esses alto-fornos operam à base energética de carvão vegetal e suas demandas variam dinamicamente com a quantidade solicitada de ferro gusa, operação do forno, carga de minério utilizada e qualidade do próprio carvão.

O fornecedor de carvão para os alto-fornos é a subsidiária do grupo, Vallourec Florestal, que deve estar atenta as demandas de seu cliente, realizando um planejamento efetivo de seus recursos. Contando com mais de 100 mil hectares de floresta de eucalipto divididos em 23 fazendas, a VFL necessita de sistemas de suporte à decisão e de profissionais experientes para conseguir gerir seu complexo processo de produção e suprir toda demanda das usinas.

Introdução ao Problema Gerencial

A VFL é composta de 23 fazendas situadas em três regionais. Juntas possuem mais de 5.000 unidades de floresta conhecidas como talhões. Os talhões são as unidades que são efetivamente colhidas. Cada um possui, essencialmente, uma área e um volume de madeira de eucalipto em certa idade de crescimento. A disposição das regionais pode ser vista na Figura 1.1.

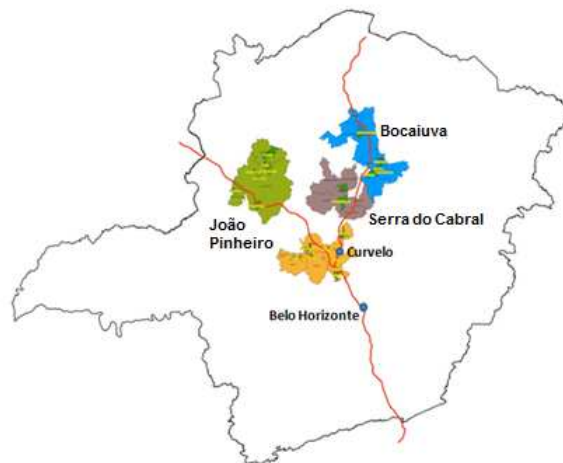


Figura 1.1: Disposição das Regionais da VFL

Em algumas fazendas, há uma unidade de carbonização – planta de produção de carvão vegetal utilizando madeira seca – que deve atender as demandas mensais de carvão vegetal dos alto-fornos. Por conseguinte, devem-se alocar as máquinas de corte disponíveis para colherem talhões em sequência de forma a atingir as demandas de todas as fazendas. Nesse processo, é necessário respeitar, a princípio, duas restrições. Primeiramente, a capacidade da máquina que consiste em número de horas disponíveis de operação em um mês. Em segundo lugar, deve-se respeitar o tempo de estocagem necessário para a madeira atingir a umidade ideal. Estimativas do tempo de estocagem podem ser obtidas, fundamentalmente, em função do clone de eucalipto, das condições plantio e idade das florestas. Outra característica importante a ser analisada é a homogeneização do diâmetro médio das toras de madeira enviadas à carbonização, que é considerado pela VFL um fator essencial para o bom rendimento de produção de carvão. Além disso, tem-se o aumento no custo de transporte em épocas de chuva, pois grande parte do tráfego de máquinas é feito em estradas de terra que se deterioram consideravelmente em período chuvoso.

O Planejamento da Colheita

No momento presente, o planejamento da colheita é feito de forma manual e requer cerca de 200 homens-hora semanais. Com frequência, o tempo de estocagem da madeira foge do ideal, além de não se considerar estimativas do diâmetro médio e da umidade das toras enviadas à carbonização, o que resulta em uma heterogeneidade da madeira, comprometendo a qualidade da produção de carvão. Recentemente, tentou-se abordar o problema para instâncias pequenas, apenas 120 talhões, utilizando otimização linear clássica [29]. Porém, o tempo de computação foi demasiadamente longo e o algoritmo não convergiu para o ótimo. Por se tratar de um problema de otimização combinatória NP-completo, ainda não se conhece métodos capazes de encontrar a solução ótima em tempo polinomial. Por isso, sugere-se que, para esses tipos de problema, os algoritmos meta-heurísticos possam ser mais eficientes, gerando soluções atrativas com menor tempo computacional [20]. Dessa forma, propõe-se neste trabalho modelar o processo de sequenciamento da colheita e otimizá-lo de forma satisfatória por meio de meta-heurísticas.

Um bom planejamento da colheita, ao utilizar algoritmos mais eficientes, tem considerável impacto no custo total da VFL, visto que a colheita representa 19% do custo total de produção do carvão vegetal. Além disso, tem repercussão direta no processo de carbonização que necessita de maior homogeneidade da madeira cortada para produção de carvão de melhor qualidade e representa 38% do custo total. Dessa forma, otimizar o sequenciamento do corte não se limita apenas a reduzir custos do processo de colheita em si, mas também da carbonização e aprimora a qualidade do produto final.

1.2 Objetivos do Projeto

Os objetivos do projeto são:

- a. Descrever e modelar o processo de sequenciamento da colheita de eucalipto, relacionando as variáveis mais importantes envolvidas;
- b. Desenvolver meta-heurísticas eficientes capazes de realizar adequadamente o planejamento e sequenciamento da colheita, garantindo a minimização dos custos envolvidos e melhor homogeneização da madeira enviada para as unidades de carbonização.

1.3 Contribuições do Projeto

Algumas possíveis e esperadas contribuições do projeto para com a VFL são:

- a. Elaboração de um modelo matemático mais representativo do processo de colheita, que reflita o contexto prático vivenciado pela empresa;
- b. Implementação de um arcabouço de técnicas de otimização que possibilite encontrar uma solução prática eficiente, com baixo custo computacional;
- c. Redução expressiva do número de homens-horas semanais envolvido no processo de planejamento, e melhor aproveitamento desse recurso em outras áreas;
- d. Redução de custos da colheita por meio de um planejamento eficaz realizado pelos algoritmos desenvolvidos.

1.4 Setores Envolvidos no Projeto

A VFL é subdividida em gerências responsáveis por cada etapa de seu processo de produção de carvão vegetal. Este projeto se destinou ao setor de colheita em si. Entretanto, o planejamento deve examinar as etapas adjacentes em uma visão integrada do processo, ou seja, também são essenciais para o projeto as considerações do setor de silvicultura e carbonização.

1.5 Estrutura da Monografia

Este trabalho está dividido em cinco capítulos. Este primeiro aborda o contexto no qual o projeto está inserido. Em seguida, o capítulo dois apresenta uma revisão bibliográfica dos temas fundamentais acerca deste trabalho, incluindo planejamento florestal, a modelagem matemática do problema de otimização e dos algoritmos implementados, estendendo-se por todos conceitos necessários para entendimento do projeto. O capítulo três aborda a metodologia de desenvolvimento, seguido pela implementação do método de otimização e planejamento da colheita, além dos critérios de análises que serão realizados. O capítulo quatro apresenta os resultados obtidos pelo método proposto e pelas análises feitas e os discute de forma crítica. Por fim, o capítulo cinco apresenta a conclusão do trabalho, um resumo das contribuições feitas por este projeto e sugestão para trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

Como introduzido na Seção 1.1.1, o planejamento de colheita consiste em escolher os melhores talhões a serem colhidos e sequenciar máquinas de corte de forma a atender as demandas de madeira seca, respeitando as restrições envolvidas. Para solução desse tipo de problema, vasto conhecimento científico foi desenvolvido nas últimas décadas. Dentre esse conhecimento, abrangem-se modelagem matemática de processos e algoritmos de otimização especialistas ou genéricos para diversos casos vivenciados em contextos práticos de indústrias e empresas.

As seguintes seções apresentam os conceitos e o estado-da-arte de estratégias de otimização apropriadas para o tipo de problema proposto e suas aplicações no planejamento florestal.

2.1 Técnicas de Otimização

Otimizar é o ato de obter o melhor resultado sob certas circunstâncias. Matematicamente pode ser definido como o processo de encontrar as condições que maximizam ou minimizam o valor de certa função, definida como função objetivo [32].

A formulação geral dos problemas de otimização, descrita em [3], é dada como:

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimizar}} && \mathbf{f}(\mathbf{x}) \in \mathbb{R}^m \\ &s.a. : && \begin{cases} g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, p. \\ h_i(\mathbf{x}) = 0, \quad i = 1, \dots, q. \\ \mathbf{x} \in \mathcal{X}. \end{cases} \end{aligned} \quad (2.1)$$

Onde $\mathbf{f}(\mathbf{x})$ são as funções objetivo, $\mathbf{x} \in \mathbb{R}^n$ são as variáveis do problema, $\mathbf{g}(\mathbf{x}) \in \mathbb{R}^p$ são restrições de desigualdade, $\mathbf{h}(\mathbf{x}) \in \mathbb{R}^q$ são restrições de igualdade e \mathcal{X} é o espaço de busca definido pelos limites inferiores e superiores das variáveis de otimização. O problema pode ser mono-objetivo, $m = 1$, ou multiobjetivo, $m > 1$. Técnicas de otimização, então, devem minimizar \mathbf{f} para as variáveis x_1, \dots, x_n , respeitando as restrições de igualdade e desigualdade.

Problemas de sequenciamento são caracterizados como combinatórios [2], i.e. as variáveis de otimização \mathbf{x} assumem valores discretos. Técnicas de otimização combinatória para sequenciamento são descritas a seguir.

2.1.1 Dimensionamento de Lotes e Sequenciamento

No contexto prático de planejamento da produção, alguns problemas podem ser modelados por meio de uma abordagem de dimensionamento de lotes seguida de sequenciamento, utilizando programação linear. O dimensionamento de lotes consiste em transformar a demanda do cliente em lotes, i.e. ordens de produção, com objetivo de minimizar custos envolvendo máquinas, produtos e estoques [14], [19].

Tabela 2.1: Dimensionamento de Lotes: Variáveis de Decisão

Variável	Definição
X_t	Quantidade produzida no período t
I_t	Quantidade de estoque no final do período t
Y_t	Variável binária que indica se ocorreu preparação de máquina no período t

O dimensionamento de lotes para um único produto e sem restrições de capacidade é formalmente definido como um problema de otimização [9] (Veja a Equação 2.2), sendo as variáveis de decisão definidas na Tabela 2.1 e os parâmetros na Tabela 2.2.

Tabela 2.2: Dimensionamento de Lotes: Parâmetros

Parâmetro	Definição
d_t	Demanda do produto no período t
p_t	Custo de produção de uma unidade no período t
s_t	Custo de preparação de máquina no período t
h_t	Custo de estoque no período t

$$\text{Minimizar} \quad \sum_{t=1}^T (s_t Y_t + p_t X_t + h_t I_t) \quad (2.2)$$

s.a.:

$$I_{t-1} + X_t = d_t + I_t \quad \forall t, \quad (2.3a)$$

$$X_t \leq Y_t d_t \quad \forall t, \quad (2.3b)$$

$$X_t, I_t \geq 0, Y_t \in \{0, 1\} \quad \forall t. \quad (2.3c)$$

Sendo o período de planejamento $t = 1, \dots, T$, $d_{tT} = \sum_{k=t}^T d_k$, o somatório de todas demandas futuras em relação ao período atual t inclusive. A Função Objetivo 2.2 contabiliza os custos de produção, estoque e preparação de máquina. A Restrição 2.3a realiza o balanço de estoque, considerando o estoque no período anterior $t - 1$ e a produção atual X_t para atenderem a demanda d_t . O restante é armazenado no estoque do fim do período atual I_t . A

Restrição 2.3b limita a produção atual X_t e a relaciona com a variável binária Y_t . Finalmente, a Restrição 2.3c define X_t e I_t como não-negativos e Y_t como binário.

Após o dimensionamento de lotes, tem-se a etapa de sequenciamento de n jobs [10], i.e. tarefas, sujeito a restrições de tempo e objetivando a minimização dos custos envolvidos. Além disso, a função objetivo pode priorizar tarefas e penalizar atrasos caso essas tenham prazo para serem processadas.

Em [30], diversas variantes de problemas de sequenciamento são descritas. Tem-se para máquinas paralelas não-relacionadas, isto é, máquinas com diferentes velocidades de processamento para cada tarefa, o seguinte modelo.

$$\text{Minimizar} \quad \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^n k p_{ij} x_{ikj} \quad (2.4)$$

s.a.:

$$\sum_{i=1}^m \sum_{k=1}^n x_{ikj} = 1 \quad \forall j, \quad (2.5a)$$

$$\sum_{j=1}^n x_{ikj} \leq 1 \quad \forall i, k, \quad (2.5b)$$

$$x_{ikj} \in \{0, 1\}. \quad (2.5c)$$

Sendo $i = 1, \dots, m$ as máquinas paralelas, $j = 1, \dots, n$ as tarefas a serem processadas, x_{ikj} uma variável binária que assume valor 1 se a tarefa j é a k -ésima tarefa processada na máquina i e $k p_{ij}$ é o tempo de processamento da tarefa j na máquina i . A Função Objetivo 2.4 contabiliza o tempo total do sequenciamento. A Restrição 2.5a garante que cada tarefa é processada somente uma vez. Já a Restrição 2.5b assegura que cada máquina não processe tarefas simultaneamente. Finalmente, a Restrição 2.5c define x_{ikj} como variável binária.

Outras abordagens utilizando heurísticas, que consideram m máquinas paralelas, são propostas em [27] e [23]. Este também cita a ineficácia de técnicas de programação inteira e métodos *branch-and-bound* para instâncias maiores, em virtude de muitos problemas de sequenciamento serem caracterizados como NP-completo.

Um modelo de dimensionamento de lotes, considerando restrições de capacidade e máquinas paralelas, foi proposto em [31] e apresentado em [29] como um método capaz de escolher os melhores talhões a serem colhidos em um horizonte de planejamento. Este também apresenta um método exato integrando dimensionamento de lotes e sequenciamento implementado para VFL. Entretanto, o método desenvolvido não é utilizado na prática por demandar tempo excessivo de computação, inclusive para instâncias pequenas com cerca de 120 talhões.

2.1.2 Meta-heurísticas

Problemas combinatórios e NP-completos, como o sequenciamento de colheita [38], são improváveis de serem resolvidos em tempo ‘razoável’ por métodos exatos [8]. Dessa forma, heurísticas foram desenvolvidas para encontrar soluções próximas do ótimo global com menor tempo de computação. Entretanto, estas são especializadas a tipos específicos de problemas. Para contornar essa desvantagem, as meta-heurísticas agregam qualidades de ambas

abordagens, sendo genéricas para problemas discretos e fornecendo soluções razoáveis com baixo tempo computacional, devido a sua natureza, em geral, estocástica e também pela incorporação de outras heurísticas [13].

Em [36] é possível notar um crescimento no uso de meta-heurísticas nos últimos anos, graças a eficiência e flexibilidade dessas estratégias. Dentre esses métodos, pode-se citar os métodos *multi-start* como o *Greedy Randomized Adaptive Search Procedure* (GRASP). Proposto em [16] e [17], o GRASP é uma meta-heurística bastante utilizada em sequenciamento [18].

GRASP

O GRASP é basicamente dividido em duas etapas, sendo a primeira de construção, seguida pela etapa de busca local. A estrutura geral do método é apresentada no Algoritmo 1, sendo $f(\cdot)$ a função objetivo do problema.

Algoritmo 1 Meta-heurística GRASP

```

1:  $min \leftarrow +\infty$ 
2: repete
3:    $x \leftarrow$  CONSTRUÇÃO SEMI-GULOSA
4:    $\hat{x} \leftarrow$  BUSCA LOCAL( $x$ )
5:   se  $f(\hat{x}) < min$  então
6:      $min \leftarrow f(\hat{x})$ 
7:      $x^* \leftarrow \hat{x}$ 
8:   fim se
9: até Condição de parada satisfeita
10: devolve  $x^*$ 

```

Na fase de construção do GRASP, uma solução é construída iterativamente. A cada passo, elementos são incorporados até que a solução esteja completa e factível, i.e. nenhuma restrição é violada. Para se adicionar um elemento, uma lista de possíveis candidatos (RCL, *restricted candidate list*) é construída e ordenada de acordo com alguma função gulosa. O elemento a ser inserido na solução parcialmente construída é selecionado aleatoriamente da RCL. A RCL pode ser limitada pelo número de elementos ou pelo valor da função gulosa. No último caso, apenas elementos com custo menor do que $c_{\min} + \alpha(c_{\max} - c_{\min})$ fazem parte da RCL, sendo c_{\min} e c_{\max} os valores mínimo e máximo da função gulosa, respectivamente. Finalmente, α representa um compromisso entre diversidade e intensificação da construção.

Na etapa de busca local, outras meta-heurísticas podem ser utilizadas [5] ou até mesmo algoritmos exatos híbridos chamados de *matheuristics* [35]. Uma possível meta-heurística, utilizada como busca local, é o *Basic Variable Neighborhood Search* (BVNS) [34]. O BVNS tem sido amplamente utilizado em sequenciamento e apresenta resultados satisfatórios [12].

Essa meta-heurística consiste em explorar dinamicamente a vizinhança de uma solução [8]. Inicialmente, são definidas estruturas de vizinhança $(N_1, \dots, N_{n_{\max}})$ que exploram vizinhos cada vez mais distantes em seu ciclo. Esse ciclo consiste em 3 fases. A primeira, chamada de *shake*, é uma perturbação, utilizando uma estrutura de vizinhança N_n , aplicada à solução. Em seguida, ocorre a fase de busca local que se utiliza de qualquer estrutura de vizinhança para encontrar um ótimo local. Se a solução encontrada é melhor do que a inicial, então n passa a ser igual a 1 e o processo se reinicia. Senão o algoritmo move para a próxima

estrutura de vizinhança $n + 1$. Essa, então, é a terceira fase do BVNS. A estrutura básica desse método pode ser vista no Algoritmo 2.

Algoritmo 2 Meta-heurística BVNS

```

1: procedimento BVNS( $x, k_{max}$ )
2:    $k \leftarrow 1$ 
3:   repete
4:      $\hat{x} \leftarrow \text{SHAKE}(x, k)$ 
5:      $\hat{\hat{x}} \leftarrow \text{BUSCA LOCAL}(\hat{x})$ 
6:     se  $f(\hat{\hat{x}}) < f(x)$  então
7:        $k \leftarrow 1$ 
8:        $x \leftarrow \hat{\hat{x}}$ 
9:     senão
10:       $k \leftarrow k + 1$ 
11:    fim se
12:  até  $k = k_{max}$ 
13:  devolve  $x$ 
14: fim procedimento

```

O GRASP ainda pode ser aprimorado aplicando diversas outras técnicas. Dentre elas, pode-se citar a Reconexão por Caminhos que se resume em utilizar informação das melhores soluções encontradas para criar uma nova superior e o GRASP Reativo que iterativamente adapta o parâmetro α da construção gulosa com base na qualidade das soluções construídas [7].

Em [21] podem ser vistos diversos artigos com as aplicabilidades do GRASP e do BVNS em problemas de roteamento de veículos, sequenciamento e outros.

2.1.3 Otimização Multiobjetivo

Problemas reais, em geral, possuem mais de um objetivo, e.g. reduzir custos e maximizar capacidade de carga em projetos de aeronaves ou minimizar distância percorrida e o número de veículos em problemas de roteamento. Nessa categoria de problemas, tem-se m funções objetivo, em geral contraditórias, que devem ser minimizadas/maximizadas simultaneamente. Dessa maneira, tem-se um conjunto de soluções de compromisso, *tradeoff* entre os objetivos, para esse tipo de problema que é, normalmente, grande e em alguns casos, infinito.

Para se comparar soluções, dados muitos objetivos, define-se as relações de Pareto-Dominância. À vista disso, uma solução \mathbf{x}_1 Pareto-domina \mathbf{x}_2 , ou simplesmente $\mathbf{x}_1 \prec \mathbf{x}_2$, se $\mathbf{f}(\mathbf{x}_1) \leq \mathbf{f}(\mathbf{x}_2)$ e $\mathbf{f}(\mathbf{x}_1) \neq \mathbf{f}(\mathbf{x}_2)$, considerando um problema de minimização de todos objetivos. Por outro lado, soluções são não-dominadas entre si, ou incomparáveis, quando $\mathbf{x}_1 \not\prec \mathbf{x}_2$ e $\mathbf{x}_2 \not\prec \mathbf{x}_1$ [11].

Dadas as relações de dominância, pode-se definir o conjunto de soluções Pareto-ótimo, i.e. o conjunto de soluções não dominadas, como $\mathcal{P} = \{\mathbf{x}^* \in \mathcal{F} \mid \nexists \mathbf{x} \in \mathcal{F} \text{ tal que } \mathbf{f}(\mathbf{x}) \prec \mathbf{f}(\mathbf{x}^*)\}$. Por consequência à definição anterior, tem-se o conjunto Fronteira Pareto-ótima \mathcal{S} que se refere a imagem do conjunto Pareto-ótimo global, sendo $\mathcal{S} = \{\mathbf{y} = \mathbf{f}(\mathbf{x}) : \mathbf{x} \in \mathcal{P}\}$. Para minimização de apenas dois objetivos e duas variáveis de decisão, pode-se ver um

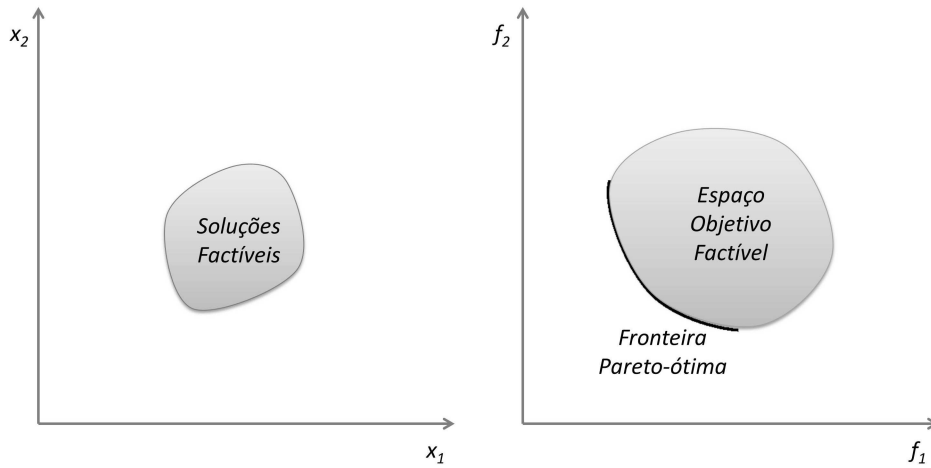


Figura 2.1: Espaço de Decisão e de Objetivos

exemplo de um conjunto de soluções factíveis e da Fronteira Pareto-ótima para um problema de minimização na Figura 2.1.

Método da Soma Ponderada e Método ϵ -restrito

Uma das primeiras tentativas de se solucionar problemas de otimização multiobjetivo foi o Método da Soma Ponderada (WSM, *Weighted Sum Method* em inglês). Seu princípio é transformar o problema em mono-objetivo ao associar cada função objetivo a um peso e somá-las como $F(\mathbf{x}) = \omega_1 F_1(\mathbf{x}) + \omega_2 F_2(\mathbf{x}) + \dots + \omega_m F_m(\mathbf{x})$, sendo m o número de objetivos, ω_i o peso para função objetivo F_i e $\sum_{i=1}^m \omega_i = 1$. Este método é simples e eficiente em termos de computação, porém não consegue encontrar todas as soluções Pareto-ótimas se a Fronteira Pareto-ótima não for convexa. Além disso, relacionar as preferências do tomador de decisão e os pesos atribuídos as funções objetivos é uma tarefa não trivial, uma vez que as soluções encontradas podem não refletir as preferências manifestadas nos pesos [11].

Solucionando alguns problemas do Método da Soma Ponderada, uma outra técnica comum para otimização multiobjetivo é o método ϵ -restrito. Por esse método, o problema multiobjetivo é transformado em mono-objetivo com restrições adicionais [15].

Basicamente, o ϵ -restrito consiste em minimizar um objetivo, sendo os demais considerados como restrições de desigualdade. Assim, tem-se o problema de otimização modificado dado pela Formulação 2.6. Para o caso biobjetivo, representado pela Figura 2.2, tanto soluções Pareto-ótimas globais quanto soluções Pareto-ótimas locais podem ser encontradas.

$$\begin{aligned}
 &\underset{\mathbf{x}}{\text{minimizar}} && f_k(\mathbf{x}) \in \mathbb{R} \\
 &\text{s.a. :} && \begin{cases} f_i(\mathbf{x}) \leq \epsilon_i, \quad i = 1, \dots, m \text{ e } i \neq k. \\ g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, p. \\ h_i(\mathbf{x}) = 0, \quad i = 1, \dots, q. \\ \mathbf{x} \in X. \end{cases}
 \end{aligned} \tag{2.6}$$

Pelo método, é possível encontrar qualquer solução do conjunto Pareto-ótimo. Sendo assim, ao variar sistematicamente os parâmetros ϵ_i , pode-se obter um conjunto representativo dessas soluções e mapear a Fronteira Pareto-ótima. Portanto, a variação dos parâmetros deve ser precisa o suficiente para não se perder soluções desse conjunto. Entretanto, isso

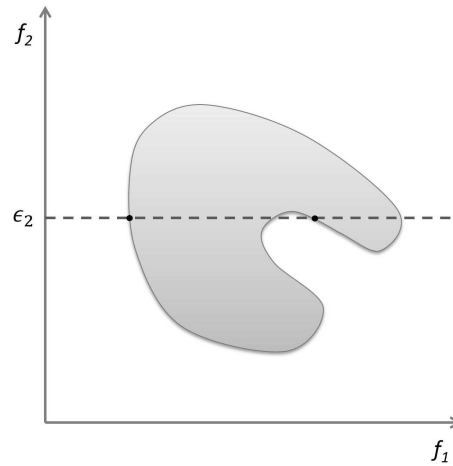


Figura 2.2: Método ϵ -restrito para problema biobjetivo

pode resultar em um grande número de iterações redundantes ao obter soluções que não são significativamente diferentes. Em [26], são descritas metodologias eficientes para variação dos parâmetros do método ϵ -restrito, utilizando informação das soluções já encontradas e, à vista disso, obter a Fronteira Pareto-ótima mais rapidamente.

2.1.4 Otimização Robusta

No âmbito prático, sistemas reais possuem diversas fontes de incerteza, como comentado em [28] para problemas de planejamento florestal. Em [24] são discutidas diferentes categorias de incertezas em computação evolucionária e respectivas abordagens para contornar esses problemas.

Em primeira análise, a robustez de uma solução pode ser definida em termos da sensibilidade da função objetivo em relação à perturbações:

- a. Das variáveis de otimização;
- b. Dos parâmetros;
- c. Da própria função objetivo.

Considerando a definição acima, em [24] são apresentadas metodologias para busca de soluções robustas. Em uma possível abordagem, os algoritmos otimizam o valor esperado da função objetivo, dada uma distribuição de probabilidade de perturbações, como expresso pela Equação 2.7.

$$F(x) = \int_{-\infty}^{\infty} f(x + \delta)p(\delta)d\delta \quad (2.7)$$

Sendo f a função objetivo original do problema, F a nova função objetivo a ser considerada nos algoritmos de otimização, $p(\delta)$ a distribuição de probabilidade dos possíveis distúrbios δ .

Em outro caso, considera-se uma abordagem multiobjetivo quando a Equação 2.7 não se mostra suficiente para encontrar soluções robustas, pois variações positivas e negativas da função objetivo podem se cancelar em certos cenários. Consequentemente, soluções com

função objetivo de alta variância são erroneamente consideradas robustas. Portanto, o valor esperado, descrito anteriormente, a média e o desvio padrão da função objetivo podem, também, ser considerados objetivos diferentes [33], resultando em soluções de compromisso entre desempenho e robustez.

Em [37], é descrito um *framework* para se encontrar soluções robustas utilizando meta-heurísticas. Este define funções de avaliação de robustez como a Equação 2.8.

$$\hat{f}(\xi) = \sum_{i=1}^{n_e} \omega_i \left(\left| f(S_i(\xi), S'_i(\pi)) - f(\xi, \pi) \right| \right) \quad (2.8)$$

Sendo $f(\xi, \pi)$ a função objetivo, ξ é uma solução com variáveis de otimização estocásticas, π são os parâmetros estocásticos de f , S_i e S'_i são funções de amostragem de ξ e π , respectivamente, n_e é a quantidade de testes a serem realizados em f , ω são pesos para realizar a média ponderada do somatório. A Equação 2.8, então, realiza o somatório do módulo das diferenças entre $f(\cdot)$ das soluções perturbadas e da solução nominal ξ .

Algoritmo 3 Definição da Solução Robusta

- 1: $\xi_1, \dots, \xi_{n_s} \leftarrow$ MÉTODO DE OTIMIZAÇÃO
 - 2: $F_\xi \leftarrow \hat{f}(\xi_1, \dots, \xi_{n_s})$ ▷ Avaliar todas soluções utilizando a Equação 2.8
 - 3: **devolve** Solução mais robusta $\min(F_\xi)$
-

Soluções robustas podem ser definidas seguindo o Algoritmo 3, no qual um método de otimização encontra n_s soluções que são avaliadas pela Equação 2.8. Este algoritmo, então, define a solução que obteve menor valor dado pela Equação 2.8 como a mais robusta. Uma alternativa à Equação 2.8, é avaliar o valor da função objetivo para todos os testes n_e e considerar apenas o pior caso. Assim, a solução robusta escolhida seria aquela com melhor desempenho no pior caso.

Além de robustez em relação a variação da função objetivo, pode-se também analisar a robustez de uma solução relacionada a sua factibilidade, i.e. uma solução não viola restrições frente as incertezas do sistema que podem tanto alterar a solução quanto as restrições do modelo.

Uma outra importante característica de uma solução é sua flexibilidade. Em [37], define-se a flexibilidade de uma solução como sua relativa facilidade em ser adaptada, por uma rotina de reparo, devido a mudanças dos requisitos do problema de otimização. Dessa forma, uma solução pode ser flexível e robusta se a solução original e a reparada estão ‘próximas’ no espaço de soluções, i.e. as duas têm alto grau de similaridade.

2.2 Planejamento Florestal

A aplicação de técnicas de otimização no planejamento florestal tem sido amplamente estudada nos últimos anos. Em [2], são descritos métodos e abordagens comuns com restrições e variáveis espaciais para planejamento florestal, como custos envolvendo construção e manutenção de estradas, restrições de adjacência entre áreas colhidas e restrições ambientais. Esse artigo também se refere a ineficácia em otimizar planejamento florestal e sequenciamento de colheita utilizando métodos exatos, devido à incorporação de novas restrições, resultando em problemas maiores, e não-linearidades provindas de problemas cada vez mais complexos.

Em [25], o planejamento florestal é discriminado em estratégico, tático e operacional sendo planejamentos de longo, médio e curto prazo, respectivamente. No artigo também é descrito o planejamento anual – operacional – de colheita, similar ao problema proposto neste trabalho, envolvendo alta complexidade como múltiplas máquinas, fatores climáticos, custos de estradas, estoque e entrega da madeira para unidades de processamento. Além desse artigo, muitos outros foram escritos para planejamento de colheita, como a aplicação de modelo linear inteiro misto [1] e algoritmo genético [22].

Problemas de robustez no planejamento de colheita podem ser vistos em [6] e [28]. Além disso, uma metodologia para validação de heurísticas aplicadas ao planejamento florestal é descrita em [4].

2.3 Conclusão

Tendo em vista a alta complexidade e o grande número de variáveis envolvidos no planejamento de colheita descrito em [25], os métodos exatos demandam excessivo tempo de computação para convergirem, o que os tornam inviáveis para planejamento de curto prazo. Dessa forma, meta-heurísticas são mais apropriadas, apesar de não garantirem otimização global, por serem eficientes e flexíveis para incorporar diversas restrições, e.g. restrições ambientais, espaciais e de capacidade e outros objetivos, como a qualidade da madeira cortada e a robustez das soluções.

As meta-heurísticas escolhidas para solução do problema proposto foram o GRASP e BVNS descritos na Seção 2.1.2. Esses métodos já foram validados em problemas de sequenciamento e são expansíveis com simplicidade para casos multiobjetivo. Além disso, são versáteis para agregar outras heurísticas, que os tornam especialistas no problema proposto, e escaláveis caso, futuramente, deseje-se acrescentar novas variáveis.

Capítulo 3

Metodologia

Neste capítulo, discute-se a abordagem escolhida para solucionar o problema proposto. Primeiramente, é descrita toda modelagem matemática e computacional do planejamento de colheita da VFL. A seguir, o algoritmo de otimização é exposto, juntamente com todas as técnicas utilizadas. Finalmente, todos os testes, avaliações necessárias, dados e parâmetros de entrada para validação da metodologia são apresentados.

3.1 Modelagem Matemática do Planejamento de Colheita

O planejamento de colheita pode ser modelado como dimensionamento de lotes seguido de sequenciamento, como apresentado na Seção 2.1.1. Sendo assim, são consideradas características de ambos modelos para uma melhor descrição do problema proposto. A modelagem, então, é apresentada nas próximas seções.

3.1.1 Características do Problema de Otimização

O planejamento de colheita consiste em sequenciar M máquinas para percorrerem T talhões e, por conseguinte, atender a quantidade de madeira requerida pelas plantas de carbonização que, por sua vez, devem atender as demandas de carvão vegetal dos alto-fornos. Cada máquina m está sujeita a uma limitação de capacidade C_m^p em cada período p . Os planejadores de colheita, então, devem distribuir as máquinas entre as R regionais e sequenciá-las em um plano de corte de curto prazo.

Sendo C a quantidade total de plantas de carbonização, cada planta c possui uma demanda D_c^p de madeira no período p . O conjunto de fazendas F de todas regionais R , então, deve fornecer a madeira suficiente para atender o conjunto de demandas D^p de todas plantas de carbonização em todos os períodos $1, \dots, P$.

À vista disso, a estrutura adotada para representar a solução deste problema de otimização é apresentada na Figura 3.1. As setas indicam o sentido de movimento da máquina. Além disso, o índice i , em cada talhão T_i , indica a posição do talhão no sequenciamento. Para cada máquina m , portanto, tem-se uma lista de talhões T^m a ser percorrida.

Um exemplo de sequenciamento entre fazendas diferentes pode ser visto na Figura 3.2. Neste caso, pode-se constatar que a fazenda F_2 não possui floresta disponível para corte e, logo, as fazendas mais próximas devem fornecer madeira para sua planta de carbonização. Portanto, em cada talhão deve ser especificada para qual planta de carbonização este se

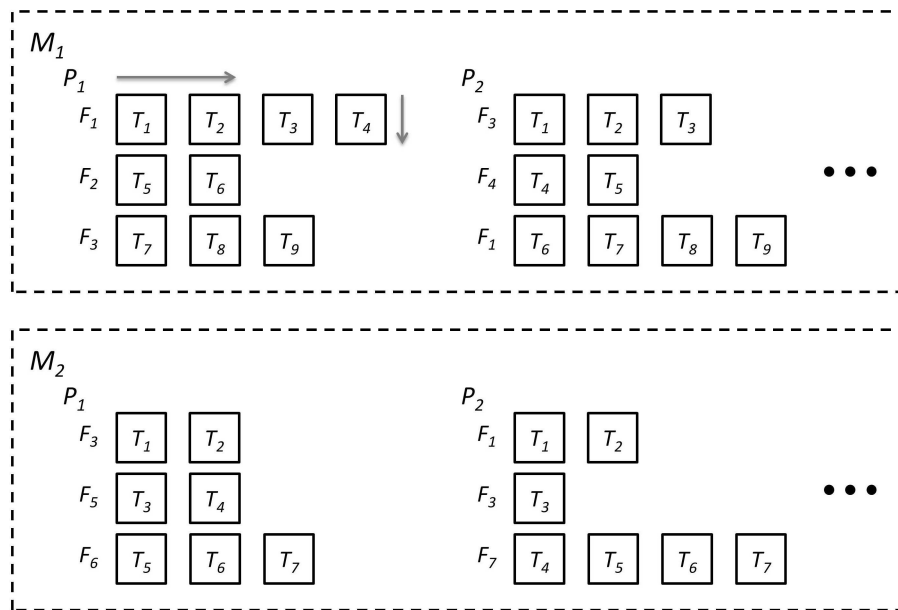


Figura 3.1: Estrutura da Solução

destina e a estrutura de solução deve, também, representar as plantas como mostra a Figura 3.3.

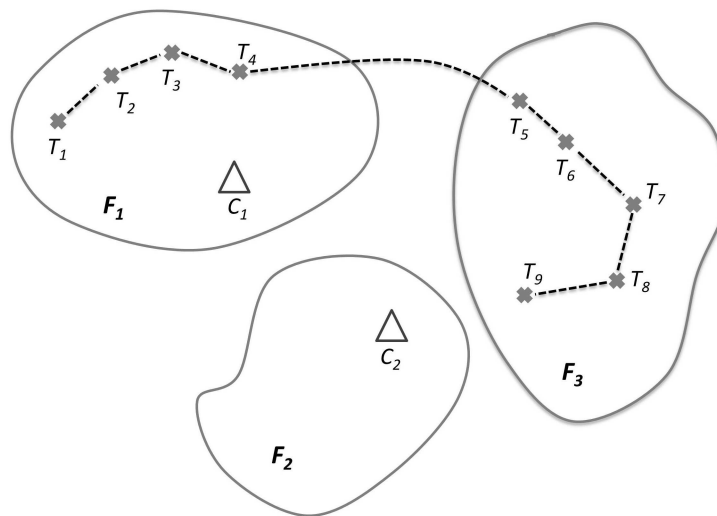


Figura 3.2: Exemplo de Sequenciamento

Após o corte, a madeira deve permanecer em estoque para secagem. Este tempo pode ser estimado considerando, essencialmente, clone e idade das árvores de eucalipto e das condições climáticas enfrentadas pelos talhões. Deste modo, é importante destacar que os talhões, colhidos no período p , são apenas transportados para a planta de carbonização após sua permanência em estoque. Além disso, um talhão pode ser transportado para a planta em múltiplos períodos consecutivos.

Sendo assim, o método de otimização aplicado deve ser capaz de obter a melhor configuração de talhões – sua ordem e, também, o destino – para cada máquina, respeitando todas

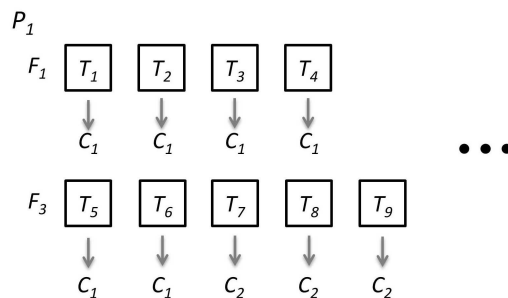


Figura 3.3: Estrutura da Solução considerando plantas de carbonização

restrições envolvidas.

3.1.2 Funções Objetivo

Nesta seção, são definidas funções matemáticas, ou funções objetivo, que têm por finalidade analisar e ranquear soluções quantitativamente. Sendo o problema de planejamento da colheita naturalmente multiobjetivo, são definidas mais de uma função objetivo segundo os critérios mais importantes para a colheita.

Custos de Máquina

Em primeira análise, tem-se o custo total de uso das máquinas, para percorrermos os talhões sequenciados, definido pela Equação 3.1. Como a finalidade deste objetivo é quantificar o custo relacionado à transição das máquinas, são desconsiderados custos envolvendo o corte dos talhões em si, apesar de existirem máquinas com rendimentos e custos de corte diferentes.

$$F_1([T^1, \dots, T^M]) = \sum_{m=1}^M \sum_{p=1}^P \sum_{i=1}^{I^{m,p}-1} [d(T_i^{m,p}, T_{i+1}^{m,p}) K^{m,p}] \quad (3.1)$$

Onde T^i, \dots, T^M são vetores de talhões alocados para cada máquina para todos períodos, $I^{m,p}$ é a quantidade de talhões alocados para máquina m no período p , $d(T_i^{m,p}, T_{i+1}^{m,p})$ é a distância Euclidiana em quilômetros entre talhões $T_i^{m,p}$ e $T_{i+1}^{m,p}$ percorridos pela máquina m no período p , $K^{m,p}$ é o custo por quilômetro percorrido para cada máquina m no período p . Logo, a função objetivo F_1 resulta em custos em reais para transição de máquinas em uma aproximação ao custo concreto pago pela VFL. Existem diversos outros fatores que afetam esses custos, porém a Equação 3.1 representa bem, para fins do modelo de otimização desenvolvido, os custos de máquina.

Distância em Relação às Plantas de Carbonização

O transporte de madeira para as plantas de carbonização é precário em período de chuvas, pois grande parte das estradas disponíveis são de terra. Portanto, o custo de transporte da madeira nesses meses é mais caro em relação ao período de seca. Para quantificar essa característica do problema de planejamento de colheita, tem-se a Equação 3.2.

$$F_2\left([T^1, \dots, T^M]\right) = \sum_{m=1}^M \sum_{p=1}^P \sum_{i=1}^{I^{m,p}} \left[d(T_i^{m,p}, C_{T_i^{m,p}}) F_K(\mathcal{L}(T_i^{m,p})) \right] \quad (3.2)$$

Onde $d(T_i^{m,p}, C_{T_i^{m,p}})$ é a distância Euclidiana em quilômetros entre o talhão $T_i^{m,p}$ e a planta de carbonização $C_{T_i^{m,p}}$ a qual se destina sua madeira, $\mathcal{L}(T_i^{m,p})$ é uma função que retorna as quantidades de madeira do correspondente talhão que são transportadas em cada período para a planta de carbonização, $F_K(\cdot)$ é uma função que retorna o custo por quilômetro de se transportar a madeira para a planta de carbonização, recebendo como parâmetros os períodos em que a madeira é transportada e as quantidades. Assim, F_2 representa o custo total em reais de transporte de madeira para as plantas de carbonização. É necessário salientar, assim como feito para Equação 3.1, que os custos de transporte representam aproximações ao custo real.

Homogeneização da Madeira

A homogeneidade da madeira enviada à carbonização é considerada um fator importante para eficiência do processo. Madeira com distribuição de diâmetros dispersa, i.e. com alto desvio-padrão, pode causar atrasos no processo de carbonização, gerar maior quantidade de perdas e resultar em carvão vegetal de baixa qualidade. Assim, é importante para a VFL que o setor de colheita forneça madeira homogênea para as plantas de carbonização. Entretanto, isso não é considerado, atualmente, ao realizar um planejamento de corte. A Equação 3.3 quantifica a homogeneidade da madeira ao computar o desvio padrão dos diâmetros em todos os períodos.

$$F_3\left([T^1, \dots, T^M]\right) = \sum_{c=1}^C \sum_{m=1}^M \sum_{p=1}^P \sigma(T^m, p, c) \quad (3.3)$$

Onde C é a quantidade total de plantas de carbonização e $\sigma(T^m, p, c)$ é uma função que retorna o desvio-padrão do diâmetro da madeira colhida pela máquina m que será entregue a planta c no período p . Note-se que a variável de talhões sequenciados T^m para máquina m não está indexada pelo período p . Sendo assim, p não significa o período em que os talhões foram cortados, mas sim o período em que eles são transportados para a planta de carbonização. Logo, F_3 retorna a soma de todos desvios padrão de madeira enviadas à carbonização.

3.1.3 Restrições

Uma vez definida todas as funções objetivo, os métodos de otimização devem minimizá-las, porém respeitando algumas restrições impostas. Essas restrições são reais limitações enfrentadas pela equipe de planejamento da colheita e, assim, garantem que o modelo resulte em soluções que possam ser utilizadas na prática.

Capacidade

As máquinas de corte possuem limitação de horas de trabalho em um mês, dependendo de diversos fatores estocásticos como, por exemplo, o tempo gasto em manutenção. Assim, a

equipe de colheita define dois parâmetros principais para mensurarem o tempo total disponível das máquinas por mês. Primeiramente, tem-se a eficiência operacional, $\mathcal{EO} \in [0, 1]$, que mede a fração de tempo em que a máquina pode estar em trabalho, considerando horário de almoço de funcionários, tempo de estagnação em troca de turno, atrasos e outros. Em segundo lugar, tem-se a disponibilidade mecânica, $\mathcal{DM} \in [0, 1]$, que se refere a fração de tempo em que a máquina não está em manutenção. Esses dois parâmetros são estipulados, pela equipe de planejamento, de acordo com dados históricos e com planejamentos realizados pela equipe de manutenção e operação das máquinas e, dessa forma, oferecem estimativas médias.

Assim, a capacidade média $\bar{\mathcal{C}}^{m,p}$ de uma máquina m no período p pode ser calculada pela equação 3.4.

$$\bar{\mathcal{C}}^{m,p} = \mathcal{C}_{\max}^{m,p} \mathcal{EO}^{m,p} \mathcal{DM}^{m,p} \quad (3.4)$$

Onde $\mathcal{C}_{\max}^{m,p}$ é a quantidade máxima total de horas de trabalho da máquina m no período p que depende do número de turnos de trabalho, da quantidade de horas por turno e do número de dias de trabalho no período p , $\mathcal{EO}^{m,p}$ e $\mathcal{DM}^{m,p}$ são a eficiência operacional e a disponibilidade mecânica da máquina m no período p , respectivamente.

Porém, como os parâmetros são valores médios e estimados, é melhor definir a capacidade de máquina disponível para trabalho em um valor em torno da média calculada na Equação 3.4. Por conseguinte, define-se a restrição de capacidade como um intervalo dado pela Equação 3.5.

$$\bar{\mathcal{C}}^{m,p}(1 - \delta_c) \leq \mathcal{C}^{m,p} \leq \bar{\mathcal{C}}^{m,p}(1 + \delta_c) \quad (3.5)$$

Onde $\mathcal{C}^{m,p}$ é a capacidade utilizada pela máquina m no período p e $\delta_c \in [0, 1]$ é uma estimativa da dispersão que a capacidade disponível tem em torno de sua média.

Demanda

Cada planta de carbonização possui, fundamentalmente, demandas mensais de madeira a ser carbonizada e, portanto, o plano de corte deve garantir que todas essas sejam atingidas. Sendo assim, tem-se a restrição de demanda dada pela Equação 3.6.

$$g_d(\mathbf{T}) = \sum_{m=1}^M \mathcal{V}(\mathbf{T}^m, p, c) = \mathcal{D}^{c,p} \quad (3.6)$$

Sendo $\mathcal{D}^{c,p}$ a demanda de madeira da planta c no período p , $\mathcal{V}(\mathbf{T}^m, p, c)$ uma função que retorna a quantidade de madeira colhida pela máquina m que será transportada no período p para a planta de carbonização c .

Estoque

A madeira, após colhida, deve permanecer um período de tempo em estoque para que se seque até o nível ideal de umidade, a fim de ser carbonizada mais eficientemente. O tempo mínimo de estoque é definido pelas características do clone de eucalipto, da idade da floresta e das condições climáticas, sendo estimado para cada talhão. Há também necessidade de se definir um tempo máximo em estoque, quando a madeira passa a apodrecer e perder sua

qualidade para a carbonização. Logo, tem-se a restrição de estoque dada pelas Equações 3.7a, 3.7b.

$$\min_p(\mathcal{L}(T_i^{m,p})) \geq \mathcal{E}_{\min}(T_i^{m,p}) \quad \forall i, m, p \quad (3.7a)$$

$$\max_p(\mathcal{L}(T_i^{m,p})) \leq \mathcal{E}_{\max}(T_i^{m,p}) \quad \forall i, m, p \quad (3.7b)$$

Onde $\mathcal{E}_{\min}(T_i^{m,p})$ e $\mathcal{E}_{\max}(T_i^{m,p})$ são funções que, para cada talhão cortado $T_i^{m,p}$, retornam o tempo mínimo e máximo de estoque, respectivamente, $\mathcal{L}(T_i^{m,p})$ a função que retorna as quantidades de madeira do correspondente talhão transportadas para a planta de carbonização em cada período e as funções \min_p e \max_p retornam o primeiro e último período em que a madeira é enviada para as plantas, respectivamente.

Sabe-se, também, que não há tempo de estoque menor do que quatro meses para todas as fazendas da VFL consideradas neste trabalho. Portanto, para se atender as demandas de madeira no período p , o corte deve ocorrer no máximo até $p - 3$, porém o tempo exato é determinado pela Equação 3.7a.

3.2 Algoritmos de Otimização

Tendo em vista a formulação descrita na Seção 3.1, escolheu-se o GRASP em conjunto com BVNS como método de otimização, sabendo que essas meta-heurísticas foram bem sucedidas para problemas semelhantes na literatura e que possuem confluência com a estrutura de solução adotada.

3.2.1 GRASP

Em relação ao GRASP é importante notar sua generalidade e facilidade para lidar com diversos tipos de problemas de otimização. É possível, também, incluir características específicas dos problemas por meio de heurísticas de construção e de busca local, o deixando especialista e mais eficiente no problema a ser resolvido. Portanto, é importante detalhar cada uma dessas etapas a fim de que seja possível retratar os aspectos mais relevantes da metodologia desenvolvida.

Construção Semi-Gulosa

A etapa de construção semi-gulosa, como já descrita, consiste em adicionar iterativamente elementos à solução até que esta se torne factível. Dessa forma, talhões são acrescentados à solução até que as restrições de capacidade (3.1.3), demanda (3.1.3) e estoque (3.1.3) sejam respeitadas. A escolha desses talhões é feita baseando-se numa lista de candidatos, RCL, que é construída por meio de uma simples adaptação das funções objetivo definidas pelas Equações 3.1 e 3.2.

Inicialmente, a heurística de construção detecta, para cada período, quais são as fazendas mais críticas – fazendas que não possuem estoque suficiente para atender o quarto mês após o período de corte, pois, como discutido na Seção 3.1.3, o menor tempo possível para estoque são quatro meses. Define-se, também, a distribuição de máquinas entre as regionais para que as demandas de todas as plantas de carbonização possam ser atendidas. A seguir, então, é

feito um sequenciamento dessas fazendas e de talhões em cada uma das regionais. Dessa maneira, o sequenciamento é feito em dois níveis: primeiramente as fazendas e depois os talhões. As fazendas são sequenciadas em uma abordagem semi-gulosa baseada na distância Euclidiana entre seus centros de massa. O centro de massa, calculado para uma fazenda f , é dado pela Equação 3.8.

$$\mathbf{x}_{\text{CM}}^f = \frac{\sum_{i=1}^{I_f} \mathbf{x}_i^f v_i^f}{\sum_{i=1}^{I_f} v_i^f} \quad (3.8)$$

Onde \mathbf{x}_{CM}^f é o vetor de coordenadas do centro de massa para fazenda f , I_f é o número de talhões disponíveis – talhões que ainda não foram sequenciados – para a fazenda f , \mathbf{x}_i^f é o vetor de coordenadas do talhão i da fazenda f e v_i^f é a massa de madeira do respectivo talhão. Os centros de massa devem ser re-calculados a cada novo sequenciamento de fazendas, pois a cada período alguns talhões são sequenciados e, assim, o centro de massa não estará condizente com a nova configuração de talhões.

A seguir, os talhões são sequenciados, seguindo a ordem pré-definida de fazendas. À vista disso, constrói-se a RCL, para a escolha dos talhões a constituírem a solução, baseando-se, primordialmente, na distância entre estes e nas outras funções objetivo definidas na Seção 3.1.2. A RCL também considera a próxima fazenda a ser visitada e, dessa maneira, prioriza os talhões escolhidos em direção a ela. Considerando o transporte de madeira para as plantas de carbonização em meses de chuva cujo custo é crítico para VFL, deve-se priorizar os talhões que estão em uma região determinada pela direção à próxima fazenda e pela direção à planta de carbonização. A Figura 3.4 representa um exemplo de região de prioridade para escolha de talhões quando transporte é realizado em meses de chuva. No entanto, se as direções à planta de carbonização e ao centro de massa forem quase paralelas, a região será pequena e, conseqüentemente, poucos talhões serão priorizados. Assim, para se generalizar a escolha de talhões, define-se a função gulosa pela Equação 3.9.

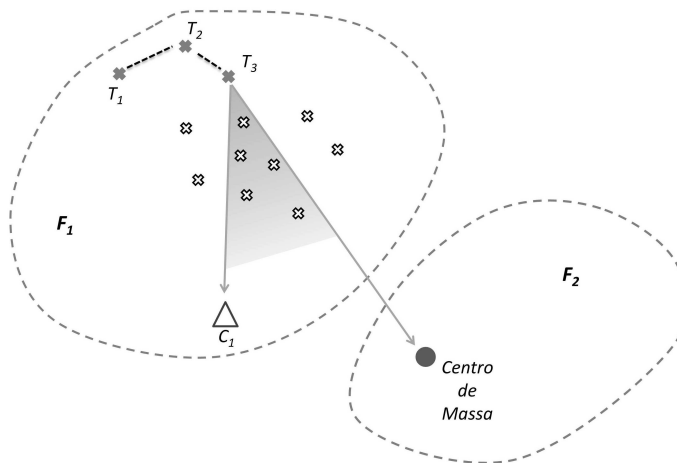


Figura 3.4: Sequenciamento de talhões: prioridade por direção

$$F_g(T_x) = d(T_i^{m,p}, T_x) (1 + |\theta(T_i^{m,p}, T_x, \mathbf{x}_{CM}^{f+1}, C_c)|\beta) \quad (3.9)$$

Onde $F_g(\cdot)$ é o valor da função gulosa para o candidato T_x pertencente à RCL, $d(T_i^{m,p}, T_x)$ é a distância Euclidiana em quilômetros entre o último talhão sequenciado e o candidato T_x e $\theta(\cdot) \in [-\pi, \pi]$ é uma função que retorna o ângulo entre a direção desejada e a direção de $T_i^{m,p}$ para T_x como mostra a Figura 3.5. A direção desejada é calculada ao somar os vetores unitários nas direções à planta de carbonização C_c e ao centro de massa da próxima fazenda \mathbf{x}_{CM}^{f+1} quando se transporta madeira em períodos de chuva, caso contrário a direção desejada é definida apenas pela direção ao centro de massa da próxima fazenda. Finalmente, β é um parâmetro para regular a priorização de talhões, assumindo apenas valores maiores ou iguais a zero. Quando β assume 0, considera-se apenas a distância Euclidiana entre os talhões na função gulosa. A medida que aumenta-se β , talhões próximos a direção desejada são mais priorizados.

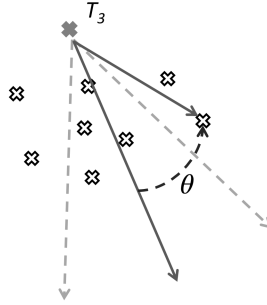


Figura 3.5: Encontrar ângulo para cálculo de F_g

Neste processo de construção, também são determinados os períodos em que a madeira é transportada após secagem. A estrutura geral desse processo é apresentada no Algoritmo 4, sendo os conjuntos R de regionais, F de fazendas, T de talhões, C de plantas de carbonização, M de máquinas e P de períodos do horizonte de planejamento.

Primeiramente, alocam-se máquinas às fazendas de cada regional de forma a atender todas as demandas críticas. A demanda crítica é a demanda do quarto mês após o período de corte e deve ser prioritariamente atendida, pois não há período de secagem menor do que quatro meses no conjunto de fazendas da VFL. A função IDENTIFICAR FAZENDAS CRÍTICAS(.) analisa todas as fazendas relacionadas para a máquina m e divide a capacidade desta entre as consideradas críticas. Caso ainda sobre capacidade de máquina, o restante é dividido entre as fazendas alocadas de maneira proporcional às demandas dos períodos seguintes. Após identificadas e alocadas, as fazendas são sequenciadas e, em cada uma, talhões também são sequenciados até que se utilize toda capacidade designada. Cada talhão é associado a uma planta de carbonização para receber sua madeira e adicionado à solução em construção.

Busca Local

Após a construção, o BVNS (Algoritmo 2), com pequenas adaptações, é utilizado para busca local nas novas soluções a fim de que se encontre ótimos locais. Nessa etapa, é importante definir as estruturas de vizinhança que mais se adequam à modelagem proposta. Assim, é essencial utilizar conhecimento específico do problema para que essas estruturas sejam mais apropriadas, resultando em um método de busca local mais eficiente.

Algoritmo 4 Construção Semi-Gulosa**requer** R, F, T, C, M, P

```

1: para  $p \leftarrow 1$  até  $P$  faça
2:    $\hat{M} = \text{ALOCAR MÁQUINAS EM REGIONAIS}(R, M, C, p)$   $\triangleright$  Relacionar máquinas
   as fazendas de cada regional
3:   para todo  $m \in \hat{M}$  faça
4:      $\hat{F} \leftarrow \text{IDENTIFICAR FAZENDAS CRÍTICAS}(F, C, m, p)$ 
5:      $F_{\text{seq}} \leftarrow \text{SEQUENCIAR FAZENDAS}(\hat{F})$ 
6:     para todo  $f \in F_{\text{seq}}$  faça
7:       repete
8:          $RCL \leftarrow \text{CRIAR RCL}(f, T, C, p)$ 
9:          $\hat{T} = \text{SELECIONAR CANDIDATO}(RCL)$ 
10:         $\text{ASSOCIAR TALHÃO A PLANTA}(\hat{T}, C)$ 
11:         $S \leftarrow \text{ADICIONAR TALHÃO NA SOLUÇÃO}(S, p, m, f, \hat{T})$ 
12:      até Capacidade alocada para  $f$ 
13:    fim para
14:  fim para
15: fim para
16: devolve Nova solução  $S$ 

```

As estruturas de vizinhas foram criadas para alterar as soluções desde o nível mais baixo, alterar sequência de talhões, até os mais altos, fazendas e períodos. As estruturas são semelhantes para cada um desses níveis, portanto, a seguir, são detalhadas mais especificamente para talhões. As estruturas de vizinhança, ou operações, definidas para o BVNS são:

- N_1 – Troca de talhões, fazendas ou períodos;
- N_2 – Deslocamento de talhão, fazenda ou período;
- N_3 – Inversão de talhões, fazendas ou períodos;
- N_4 – Inserção de talhão ou fazenda;
- N_5 – Remoção de talhão ou fazenda;
- N_6 – Troca de destino do talhão.

Primeiramente, têm-se as operações de troca N_1 que consistem em simples permutações de elementos. Na Figura 3.6, pode-se ver a operação de troca de talhões: um primeiro talhão t é escolhido aleatoriamente. A seguir, o segundo deve ser escolhido até uma certa distância δ_{N_1} do primeiro.

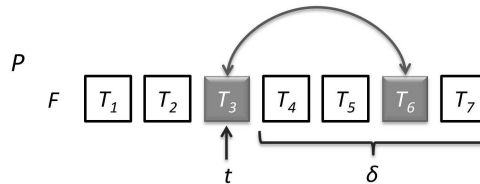
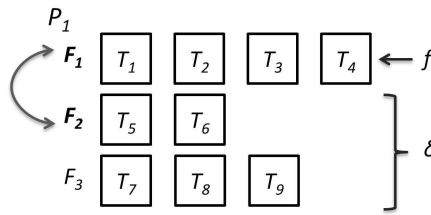
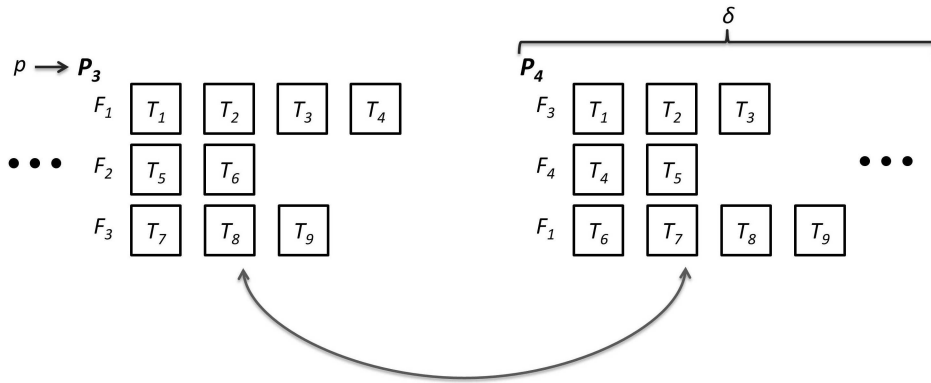


Figura 3.6: N_1 – Trocar talhões

Pode-se, também, permutar fazendas e períodos (Figuras 3.7 e 3.8, respectivamente), sendo similar à troca de talhões: escolhe-se o primeiro elemento (f ou p) aleatoriamente e o segundo a uma distância máxima δ_{N_1} .

Figura 3.7: N_1 – Trocar FazendasFigura 3.8: N_1 – Troca de períodos

A operação N_1 também permite a troca de talhões e fazendas entre períodos diferentes (Figura 3.9). Além disso, ainda é possível realizar a permutação entre talhões que foram sequenciados e aqueles que não foram. Nesses casos, a distância máxima δ_{N_1} não é considerada. Essas trocas, entretanto, geralmente inviabilizam a solução, violando as restrições de capacidade e/ou demanda devido a diferenças de quantidade de madeira entre os elementos. Portanto, é aplicada, após essa operação, uma rotina de reparo que analisa todos períodos e fazendas a fim de identificar violações de restrições e viabilizar novamente a solução por meio de trocas, remoções e inserções sistemáticas de talhões.

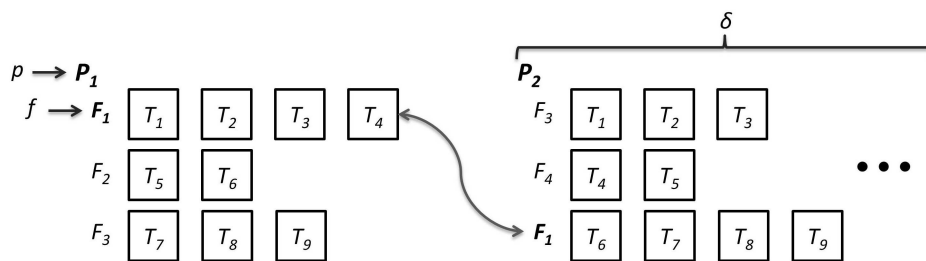
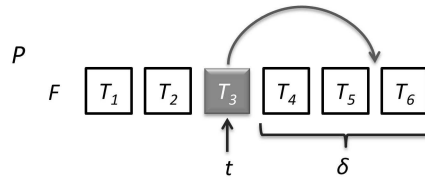


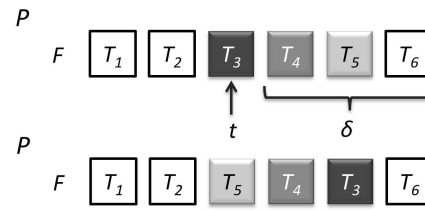
Figura 3.9: Exemplo de operações de troca para períodos diferentes

A segunda operação N_2 realiza o deslocamento de elementos (talhões, fazendas ou períodos). Para talhões, um esquema é apresentado na Figura 3.10. No deslocamento, escolhe-se aleatoriamente o talhão t a ser deslocado e sua nova posição até uma distância máxima δ_{N_2} entre as posições final e inicial. Essa operação, ao deslocar talhões e fazendas entre períodos diferentes, também inviabiliza a solução, sendo necessário aplicar a rotina de reparo.

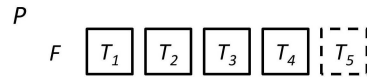
Pode-se, também, inverter elementos pela estrutura de vizinhança N_3 , como mostra a Figura 3.11 para inversão de talhões em um mesmo período. A operação N_3 também é

Figura 3.10: N_2 – Deslocar talhão

limitada por uma quantidade máxima de elementos $\delta_{N_3} + 1$, sendo o primeiro da sequência escolhido aleatoriamente.

Figura 3.11: N_3 – Inverter talhões

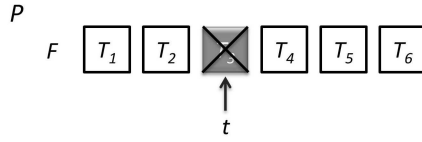
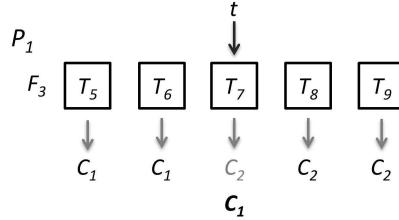
A próxima operação N_4 (Figura 3.12) insere um talhão, que não foi sequenciado, em um período, fazenda e posição escolhidos aleatoriamente. Já a operação N_5 (Figura 3.13) realiza a remoção de talhões. Essas operações podem realizar inserção e remoção de fazendas e também requer o uso de rotinas de reparo.

Figura 3.12: N_4 – Inserir talhão

Finalmente, a operação N_6 (Figura 3.14) altera o destino da madeira dos talhões entre as plantas de carbonização que estão relativamente próximas da fazenda em análise, sendo necessário o reparo após a operação.

As operações N_2, \dots, N_5 são utilizadas no BVNS como perturbações na função *shake*. Já a operação N_1 é utilizada como a busca local propriamente dita, sendo empregado o método *First Improvement*, i.e. Primeira Melhor [21]. Neste método, a operação N_1 é executada aleatoriamente em k_l iterações até que se encontre uma primeira melhora na função objetivo. Se houver melhora, reinicia-se a contagem de k_l iterações. Caso contrário, finaliza-se esta etapa. Assim, o BVNS implementado não utiliza métodos determinísticos para busca local como descrito em [34], mas procura por mínimos locais de forma estocástica, baseando-se no método *First Improvement*.

Nota-se que, após cada operação, as funções objetivo não são re-calculadas em toda solução. Porém, são calculadas somente na região alterada, aumentando, então, a eficiência, em termos de tempo de processamento, do algoritmo de busca local. Além disso, deve-se garantir que as soluções continuem viáveis posteriormente às operações.

Figura 3.13: N_5 – Remoção de talhãoFigura 3.14: N_6 – Trocar destino do talhão

3.2.2 Abordagem Escalar Via Método ϵ -restrito e GRASP

Assim como descrito na Seção 2.1.3, o método ϵ -restrito tem relativa facilidade em ser implementado em estruturas de otimização mono-objetivo ao minimizar/maximizar um único objetivo, transformando os demais em restrições. Assim, na metodologia desenvolvida, combina-se o GRASP com o método ϵ -restrito para obter uma aproximação do conjunto Pareto-ótimo.

De maneira geral, utiliza-se também neste trabalho o Método da Soma Ponderada para combinar os objetivos F_1 (Equação 3.1) e F_2 (Equação 3.2) em somente um F_{transp} . As razões para isso se devem a cultura de planejamento da VFL e, também, a maior facilidade de condução do processo de tomada de decisão ao considerar apenas dois objetivos: F_{transp} e F_3 (Equação 3.3). Ademais, os objetivos F_1 e F_2 representam custo em reais e podem ser naturalmente somados, normalizando e ajustando o parâmetro $K^{m,p}$ da Equação 3.1 e $F_K(\cdot)$ da Equação 3.2 para combiná-los em uma função do tipo $F_{\text{transp}} = \omega F_1 + (1 - \omega) F_2$.

Além disso, escolhe-se o objetivo combinado F_{transp} para ser minimizado e, então, F_3 é transformado em uma restrição de desigualdade. Essa nova restrição deve ser considerada no algoritmo de construção semi-gulosa e no BVNS que deve implementar, também, rotinas de reparo para essa.

O GRASP deve também guardar o conjunto de soluções pertencentes ao conjunto Pareto estimado e atualizá-lo sempre que for obtida uma nova solução Pareto não-dominada.

3.2.3 Análise de Robustez

Como discutido na Seção 2.1.4, é importante analisar soluções de um problema de otimização também pela sua robustez frente a perturbações para que essas continuem práticas e sem alterações significativas em sua qualidade.

Dessa maneira, a busca por soluções robustas utilizada neste trabalho será baseada no Algoritmo 3 e aplicado ao GRASP. O GRASP, ao invés de retornar a solução com melhor desempenho da função objetivo F_{transp} , guarda as n_s melhores soluções e aplica perturbações para que se selecione como solução do problema de otimização aquela que obtiver o menor valor resultante da Equação 2.8, considerando os parâmetros π constantes e a função de amostragem S_i para perturbar a solução por meio da aplicação das operações N_1, \dots, N_6

aleatoriamente.

É importante destacar que as operações N_4 e N_5 são consideradas as perturbações mais frequentes na prática de planejamento da colheita na VFL. Uma vez que talhões podem ser removidos do plano por causa de eventuais queimadas e, assim, devem ser substituídos. Além disso, alguns talhões podem ser adicionados ao plano por meio de requisições do setor de pesquisa da VFL. Portanto, essas operações devem ter maior probabilidade de ocorrência no teste de robustez.

3.3 Estrutura de Testes

A fim de validar a metodologia desenvolvida, o método de otimização foi testado para realizar planejamento de colheita anual, assim como é feito pela VFL, utilizando bases de dados dos planos de colheita de 2014 e 2015. Cada uma das bases de dados possui diferentes talhões disponíveis para corte, demandas de madeira das plantas de carbonização e configurações de máquinas. A seguir, as soluções encontradas pelos métodos deste trabalho são comparadas com os planejamentos de colheita realizados pela VFL.

Além disso, são utilizadas metodologias citadas em [4] como auto-validação. Nessa análise, os principais pontos considerados são os piores e melhores casos, desempenho médio e desvio-padrão dos testes em cada uma das bases de dados com objetivo de avaliar a repetibilidade dos métodos.

3.4 Dados de Entrada

Assim como dito na seção anterior, a metodologia desenvolvida foi testada utilizando bases de dados de 2014 e 2015. Em cada uma, são fornecidos todos os dados descritos na formulação do modelo detalhada na Seção 3.1.1. Como exemplo, os números de talhões disponíveis em uma regional para as bases de 2014 e 2015 podem ser vistos na Tabela 3.1, assim como as demandas de madeira, em toneladas, das fazendas na Tabela 3.1 são apresentadas na Tabela 3.2.

Tabela 3.1: Número de talhões normalizados por fazenda nas bases de dados 2014 e 2015

Fazendas	Talhões
2014	
Pin	0,28
Vgr	0,37
Ext	0,20
NII	0,15
2015	
Cor	0,32
Ext	0,11
Vgr	0,29
Pmo	0,29

Para cada talhão, em torno de 200 em cada base, são fornecidos dados como quantidade de madeira em toneladas, diâmetro médio, coordenadas geográficas e velocidade estimada

Tabela 3.2: Demandas de madeira normalizadas

Fazendas		Demandas (10 ⁻³)											
2014		1	2	3	4	5	6	7	8	9	10	11	12
	Pin	12,94	11,70	13,44	13,52	13,98	13,52	13,98	12,94	12,53	12,94	12,53	13,44
	Vgr	32,08	0,00	33,33	33,53	34,64	31,77	32,82	32,82	33,53	34,65	33,53	33,33
	Ext	26,47	0,00	27,49	27,66	28,58	27,66	28,58	28,58	25,91	26,78	25,91	25,75
	NII	15,62	0,00	16,22	14,95	14,56	16,32	16,87	16,87	16,32	16,87	16,32	16,22
2015													
	Cor	37,99	39,22	36,68	37,90	37,90	67,39	69,64	67,39	70,95	60,58	68,85	67,14
	Ext	27,69	29,12	25,81	27,90	29,12	28,19	29,12	28,19	29,12	28,97	26,17	28,97

de corte em toneladas por hora, i.e. em uma hora quantas toneladas são colhidas neste talhão. Além disso, também são fornecidas as coordenadas de cada unidade de carbonização, dados de operação da máquina como quantidade máxima de horas disponíveis, eficiência operacional e disponibilidade mecânica.

3.5 Parâmetros Utilizados

Divide-se esta seção em parâmetros do modelo, da construção, da busca local e os utilizados nos métodos multiobjetivo.

3.5.1 Modelo

Os parâmetros do modelo dependem exclusivamente dos cenários correntes da VFL. É necessário salientar que alguns parâmetros são definidos em valores relativos para não comprometer o sigilo requerido pela VFL.

Funções Objetivo

Para as funções objetivo F_1 e F_2 , modificam-se $K^{m,p}$ e $F_K(\cdot)$ para se adequar ao Método da Soma Ponderada como ω e $(1 - \omega)$, respectivamente. Ademais, a função $F_K(\cdot)$ se transforma em $(1 - \omega)k_c$, onde k_c assume 1 se a madeira for transportada em meses de chuva e 0 no caso contrário. Isso se deve ao fato de que é válido analisar o custo de transporte de madeira apenas no período de chuva, quando esse cresce consideravelmente em relação ao resto do ano. Além disso, anula-se este custo em tempos de seca para que o algoritmo não escolha apenas talhões próximos às plantas de carbonização durante todo o ano, deixando talhões distantes para o ano seguinte. Portanto, como é necessário equilibrar escolha de talhões próximos ou não à planta, calcula-se o custo de transporte apenas nos meses de chuva que são críticos para o custo total. Escolhendo ω como 0,2, tem-se que a relação de custos se torna proporcional aos valores vistos hoje na operação da VFL.

É válido enfatizar que a função F_{transp} não resulta em custos reais do processo de colheita. Porém, representa bem a dinâmica e variação do custo frente a diversos eventos e cenários vivenciados pela equipe de colheita da empresa.

Restrições

Para a restrição de capacidade, 3.5, tem-se o parâmetro δ_c como 5%. Este valor é usualmente praticado pela VFL em seu planejamento.

3.5.2 Parâmetros da Construção

Para construção, utilizou-se o GRASP Reativo [7] para adaptar os melhores valores do parâmetro α . Entretanto, é necessário fornecer o conjunto de possíveis valores e, portanto, utilizou-se $\mathcal{A} = [0; 0,1; 0,2; \dots; 1]$. Escolheu-se β , parâmetro da Equação 3.9, igual a 1.

Outro parâmetro a ser definido é o número de soluções geradas que foi escolhido entre 50 e 100.

3.5.3 Busca Local

Para as operações N_1, \dots, N_3 , escolheu-se o parâmetro de distância δ_{N_i} , sendo $i = 1, 2, 3$, entre 2 e 5, dependendo da quantidade de elementos a serem manipulados. Por exemplo, nas instâncias de 2014 e 2015 a cada mês são colhidos em média 6 talhões, então, valores razoáveis para N_1 e N_2 seriam 2 e 4, respectivamente.

É necessário, também, definir o parâmetro k_l , o número de iterações da busca local, que foi escolhido entre 20 e 40 iterações.

3.5.4 Métodos Escalares e Robustez

O valor do parâmetro de peso do Método da Soma Ponderada foi estabelecido na Seção 3.5.1. Porém, para que se possa analisar o comportamento das funções F_1 e F_2 frente uma alteração de ω , são feitos testes para gerar aproximações da fronteira Pareto-ótima, variando os valores de ω . Os valores de ω variam de 0 a 1 com passo fixo entre 0,1 e 0,01.

Já para o método do ϵ -restrito, varia-se o valor de ϵ em uma porcentagem do valor obtido em F_3 ao se otimizar apenas a função objetivo F_{transp} . Ou seja, inicialmente, tem-se $\epsilon \rightarrow +\infty$. Após a otimização, calcula-se $\Delta\epsilon$ como uma fração de F_3 , i.e. $\Delta\epsilon = \delta_F F_3$. Utiliza-se δ_F entre 1% a 10%. A cada iteração, atualiza-se o valor de ϵ em $\epsilon^k = \epsilon^{k-1} - \Delta\epsilon$, sendo ϵ^k e ϵ^{k-1} os valores de ϵ nas iterações atual e anterior, respectivamente.

Para o método de seleção de soluções robustas, define-se a quantidade de perturbações n_e proporcional ao número de estruturas de vizinhança, em torno de 5 vezes. Além disso, como descrito na Seção 3.2.3, N_4 e N_5 são as perturbações mais frequentes, assim define-se que cada uma possui 40% de probabilidade de escolha no teste de robustez. O restante é dividido igualmente entre as outras operações. Já a quantidade de soluções para serem avaliadas n_s é definida como percentual das soluções geradas pelo GRASP, em torno de 50%.

Capítulo 4

Resultados e Discussão

Neste capítulo, apresenta-se os resultados e análises referente ao método desenvolvido e descrito no Capítulo 3. Em primeiro lugar, analisa-se as características e eficiência dos métodos multiobjetivo desenvolvidos. A seguir, analisa-se a qualidade das soluções encontradas em relação ao planejamento feito pela VFL. Finalmente, tem-se uma análise estatística das execuções do algoritmo.

4.1 Métodos Escalares

Utiliza-se, neste trabalho, uma abordagem combinada de dois métodos escalares para solução de problemas multiobjetivo, o Método da Soma Ponderada e o Método ϵ -restrito. Testes e análises para cada um deles é feita nas seções seguintes.

4.1.1 Análise de F_1 e F_2 pelo Método da Soma Ponderada

Uma vez que as soluções resultantes de Método da Soma Ponderada podem não refletir as preferências manifestas nos parâmetros de peso, é necessário realizar análises do comportamento das funções objetivo frente a variações dos pesos.

Empregando $\omega = [0; 0,01; 0,02; \dots; 1]$, gerou-se uma aproximação à fronteira Pareto-ótima para 2014 e 2015 que são apresentadas nas Figuras 4.1a e 4.1b, respectivamente.

Ao analisar os resultados para 2014 e 2015, pode-se ver, primeiramente, que todas soluções foram superiores aos planos realizados pela empresa, ou seja, a solução da VFL não dominou nenhuma solução gerada. Há indícios, então, que o algoritmo é superior ao que é realizado atualmente. Na Seção 4.3, apresenta-se estatísticas referentes ao desempenho das soluções geradas e, assim, confirma-se a superioridade do algoritmo nos quesitos considerados nesta monografia.

Além disso, pode-se ver que o GRASP não conseguiu mapear de maneira uniforme uma aproximação à Fronteira Pareto-ótima. Pode-se atribuir esse resultado ao Método da Soma Ponderada que em alguns casos não é capaz de obter toda a fronteira ou pelo GRASP ter convergido previamente para outros mínimos locais. Novas execuções do método resultaram na Figura 4.2, na qual se pode ver as regiões em que estão situadas as soluções obtidas fazendo $\omega = 0,2$, uma vez que em 2014 e 2015, essas soluções foram dominadas e não se encontram na fronteira. Constatase que a função objetivo F_2 é mais priorizada do que F_1 como era necessário, pois F_2 tem maior custo do que F_1 e, assim, é primordial que o

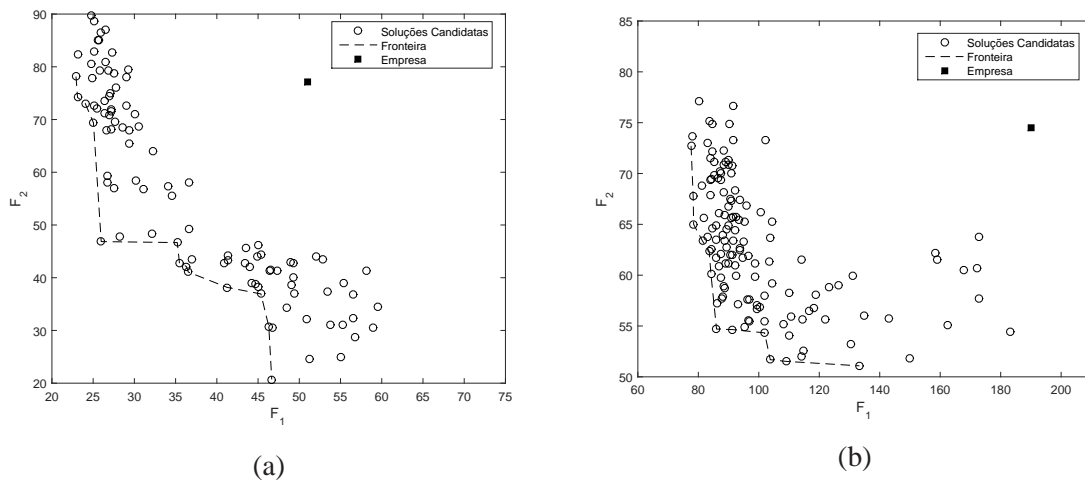


Figura 4.1: Soluções candidatas e aproximação da fronteira Pareto-ótima com objetivos F_1 e F_2 para 2015 (a) e 2014 (b)

peso $\omega = 0,2$, definido na Seção 3.5.1, reflita em soluções nas proximidades das regiões demarcadas na Figura 4.2.

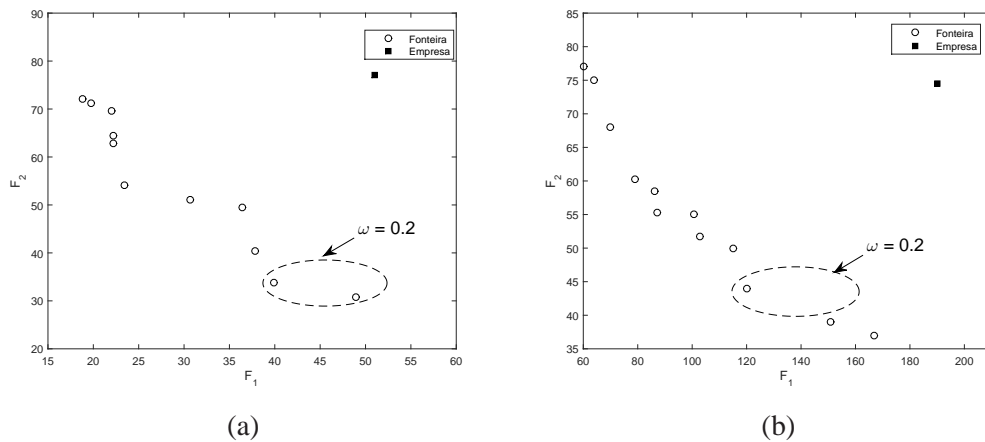


Figura 4.2: Novos testes para Método da Soma Ponderada para dados de 2015 (a) e 2014 (b)

4.1.2 Análise de F_{transp} e F_3 pelo Método ϵ -restrito

Já para o método ϵ -restrito, utiliza-se δ_F como 1% (Veja a Seção 3.5.4). Ou seja, $\Delta\epsilon$ corresponde a 1% de F_3 ao otimizar F_{transp} com $\epsilon \rightarrow +\infty$. Assim, tem-se as aproximações à Fronteira Pareto-ótima apresentadas na Figura 4.3. Mais uma vez, nenhuma solução encontrada pelo algoritmo foi dominada pelo planejamento realizado pela empresa.

Ademais, pode-se ver que o algoritmo conseguiu mapear de maneira razoável uma aproximação à Fronteira Pareto-ótima, apesar de não ter conseguido uma fronteira uniforme. Isto se deve ao fato de que a restrição criada a partir da função objetivo F_3 comprometeu parcialmente a eficiência do método de otimização, pois em muitos casos não se pôde encontrar melhores resultados para a função objetivo F_{transp} dado um valor de ϵ e um número limitado

de iterações na busca local. Em todo caso, crê-se que o algoritmo é capaz de aproximar adequadamente à Fronteira Pareto-ótima, pois o BVNS implementado possui estruturas de vizinhança suficientes para mapear grande parte do espaço de soluções factíveis. As fronteiras encontradas oferecem um conjunto razoável de soluções significativamente diferentes, cabendo ao tomador de decisão escolher qual é a mais apropriada.

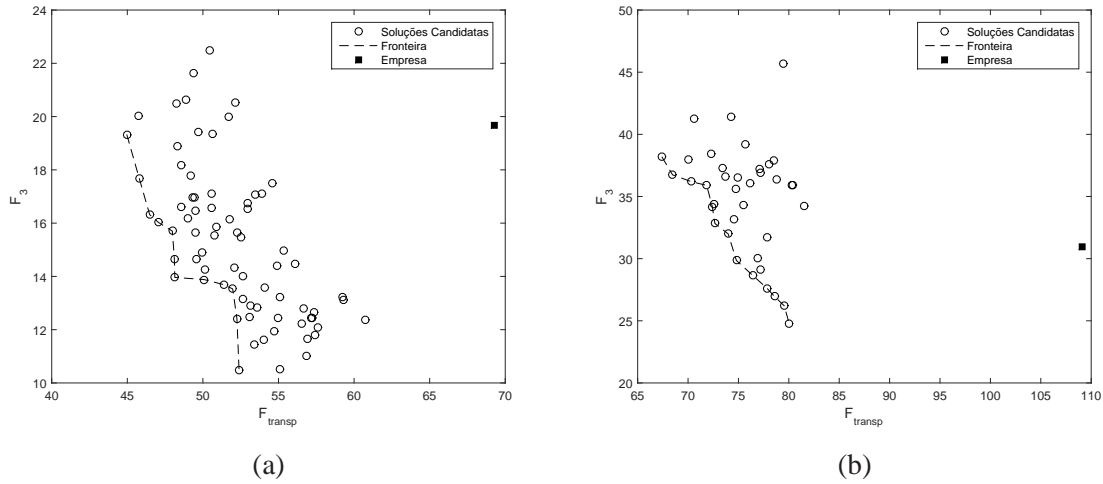


Figura 4.3: Soluções candidatas e aproximação da fronteira Pareto-ótima com objetivos F_{transp} e F_3 para 2015 (a) e 2014 (b)

4.2 Qualidade das Soluções

4.2.1 Robustez

Espera-se que todas as soluções encontradas pelo GRASP tenham um nível razoável de robustez, pois o método só retorna uma solução, entre n_s soluções, com melhor grau de robustez segundo a Equação 2.8. Assim, aplicando os testes de robustez nas soluções da VFL e nas geradas pelo algoritmo, é calculada a porcentagem de soluções com melhor índice de robustez. Os resultados são apresentados na tabela 4.1

Tabela 4.1: Percentual de soluções geradas que foram mais robustas do que as planejadas pela empresa

	2014	2015
Soluções Robustas	73%	81%
Dominadas	0%	0%

Pode-se ver que, em grande maioria, as soluções geradas pelo GRASP são mais robustas do que as propostas pela VFL. Além disso, nenhuma delas é Pareto-dominada pelas soluções da VFL.

4.2.2 Outras características

Após analisar as soluções em relação às funções objetivo e robustez, é também necessário analisá-las em relação às características importantes para o processo de colheita, como o uso das máquinas e o tempo em estoque médio de madeira.

Primeiramente, analisa-se o tempo de estoque médio das soluções da VFL e de 50 soluções geradas pelo algoritmo. Os resultados são apresentados na Tabela 4.2.

Tabela 4.2: Tempo de estoque médio em meses

	2014	2015
VFL	4,74	4,52
Algoritmo	4,93	5,04

Pode-se ver, pela Tabela 4.2, que os estoques médios em soluções dadas pelo algoritmo foram, em média, superiores aos tempos de de estoque das soluções da VFL. Tempos de estoque maior, i.e. maior tempo de secagem, refletem em maior perda de umidade pela lenha. Sabe-se, também, que quanto menor a umidade da lenha, maior a eficiência no processo de carbonização, onde-se transforma a lenha em carvão vegetal. Portanto, conclui-se que o algoritmo também tem maior vantagem nesse quesito.

Além disso, é necessário, também, analisar o uso médio de máquina nas soluções do algoritmo e pelas soluções da VFL. Na Figura 4.4, tem-se tendência de utilização média das soluções geradas pelo algoritmo, em torno de 50 soluções, e planejadas pela empresa em número de horas por mês.

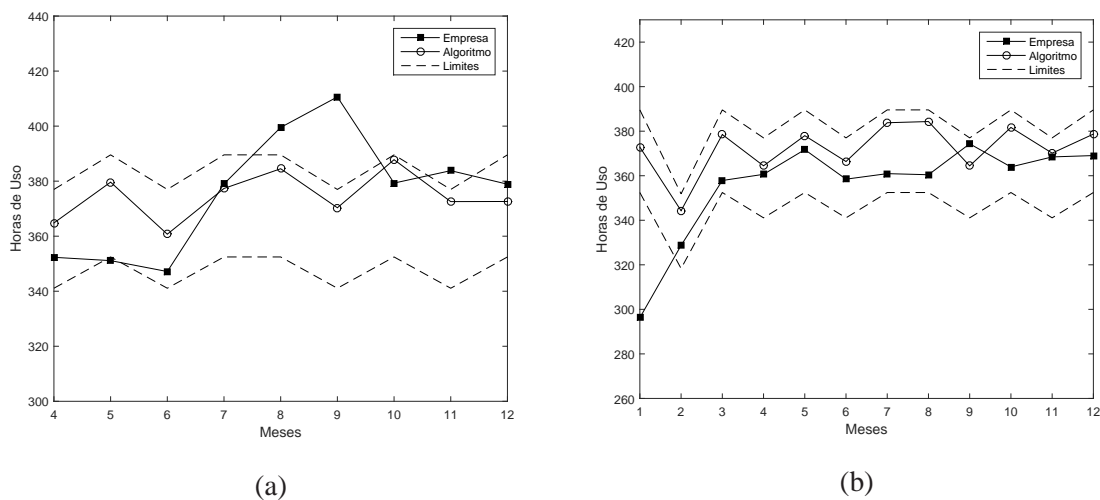


Figura 4.4: Uso de máquina médio das soluções do algoritmo e real do planejamento da empresa em 2015 (a) e 2014(b)

Pode-se ver que, nas soluções da VFL, em alguns momentos a capacidade ultrapassou os limites impostos pelo parâmetro δ_c da Equação 3.5. Entretanto, isso se deve a mudanças dos valores planejados/executados de eficiência operacional e disponibilidade mecânica. Dessa forma, os limites inferiores ou superiores podem ter sido alterados e, portanto, não houve

violação real da restrição de capacidade. Ademais, o planejamento também pode ser baseado em experiência, além dos parâmetros calculados, pois os operadores podem alocar uma máquina além do planejado de horas sabendo que será possível cumprir.

O algoritmo em geral tende a obter soluções com uso de máquina mais uniforme ao longo do ano, o que reflete em tempos de estoque médio maiores como apresentado na Tabela 4.2 e em maiores margens de segurança para atender as demandas e facilita o planejamento de manutenção das máquinas.

4.3 Análises Estatísticas das Execuções

Análise dos tempos de execução do GRASP, feitas em um computador doméstico com processador Intel Core i7-5500U, 2,4 GHz e 8 GB de memória RAM, é apresentada na Tabela 4.3. Em cada uma das execuções, o GRASP gera 50 soluções e são aplicadas 20 iterações k_l na busca local. Além disso, são apresentadas características da solução encontrada pelo GRASP, fazendo ϵ como infinito, pois deseja-se apenas estudar a repetibilidade do método mono-objetivo. Ademais, analisa-se a repetibilidade do método mono-objetivo com o método de seleção de soluções robustas, fazendo $n_s = 25$. Para se analisar uma possível repetibilidade do método multiobjetivo, utilizou-se como fator de tomada de decisão, o método de seleção de soluções robustas aplicado a todas soluções não-dominadas. Os testes foram realizados com a base de dados de 2015.

Tabela 4.3: Análise de execução e das soluções para mono-objetivo (1-obj), mono-objetivo com análise de robustez (1-obj,R) e multiobjetivo (2-obj)

	Tempos (minutos)				Soluções		
	1-obj	1-obj, R	2-obj		1-obj	1-obj, R	2-obj
Média	6,9	7,85	234,31	F_{transp}	48,78	50,30	47,43
				F_3	21,20	20,42	19,31
Desvio Padrão	0,67	0,59	23,04	F_{transp}	1,83	1,07	5,33
				F_3	1,76	2,21	4,94
Melhor Caso	6,07	7,12	219,30	F_{transp}	47,1	48,64	45,03
				F_3	18,63	16,40	13,80
Pior Caso	7,91	9,16	267,21	F_{transp}	52,81	51,97	51,01
				F_3	23,18	23,41	22,67

Os resultados são apresentados na Tabela 4.3. Os resultados para GRASP mono-objetivo estão como "1-obj", para mono-objetivo associado com o método de robustez, "1-obj, R" e para ϵ -retrato, "2-obj".

Analisando primeiramente os tempos de computação, tem-se que os tempos médios para o método mono-objetivo com e sem o método de seleção de soluções robustas foram similares e os testes resultaram em baixo desvio-padrão em relação a média. É válido salientar que os tempos de computação são razoavelmente baixos, respeitando, então, as restrições de tempo levantadas pela empresa, uma vez que os planejamentos realizados são de cunho operacional para o curto prazo. Entretanto, ao analisar os tempos para o ϵ -retrato, tem-se que são, em média, quase 30 vezes maiores do que os tempos de "1-obj" e "1-obj, R". Isto se deve ao fato de que é necessário executar o método mono-objetivo diversas vezes para

uma grande variedade de valores de ϵ . Apesar do tempo médio para o ϵ -restrito ser consideravelmente superior ao método mono-objetivo, ainda se enquadra nas restrições de tempo da empresa que deve possuir os resultados das execuções em menos de um dia. Além disso, o tempo médio do ϵ -restrito pode ser reduzido, escolhendo valores de δ_F maiores que 10%.

Em segundo lugar, analisa-se o desempenho referente a minimização dos objetivos. Tem-se que os métodos mono-objetivo com e sem seleção de soluções robustas tiveram resultados similares. Porém, o método com seleção de soluções robustas teve desempenho inferior, o que era esperado, pois há um compromisso entre robustez e desempenho. Além disso, salienta-se que os métodos apresentam repetibilidade adequada, pois o desvio-padrão das funções F_{transp} e F_3 são pequenos em relação a média. Para o método multiobjetivo, apesar do desvio-padrão ser pequeno em relação a média, a gama de soluções geradas compreende praticamente a fronteira vista na Figura 4.3a, o que dificulta a avaliação de repetibilidade do método, pois soluções robustas podem estar situadas em qualquer região das fronteiras obtidas. Há na literatura, melhores indicadores para avaliar qualidade e desempenho de métodos multiobjetivo, porém não coube, ao escopo deste trabalho, analisá-los e aplicá-los.

Capítulo 5

Conclusões

Nesta monografia, desenvolveu-se todos passos para solução de um trabalho prático vivenciado na indústria. Por meio de intenso estudo do processo e da literatura, pôde-se propor um método que resolvesse o problema dadas as restrições de eficiência computacional requeridas pela empresa.

No Capítulo 3, fez-se uma completa análise do processo de planejamento de colheita e uma estruturação matemática e computacional a ser utilizada por métodos de otimização. Dada a estruturação, pôde-se encontrar meta-heurísticas, GRASP e BVNS, que pudessem resolver o problema de otimização. Essas meta-heurísticas, apesar de genéricas para problemas de otimização combinatória, são poderosas por serem flexíveis para incorporar heurísticas especialistas no problema a ser resolvido. Neste caso, as heurísticas de construção e busca local foram desenvolvidas com foco único no sequenciamento de talhões para produção de carvão vegetal. Entretanto, a estrutura pode ser facilmente adaptada para outros processos, como produção de celulose ou biomassa para geração de energia. Além disso, apresentou-se uma fácil adaptação ao GRASP para lidar com problemas de otimização multiobjetivo. Isso, juntamente com a seleção de soluções robustas, deram aos algoritmos desenvolvidos poder para se encontrar soluções satisfatórias que estivessem condizentes com a realidade do processo de colheita.

Já no capítulo 4, apresenta-se os resultados dos métodos implementados frente a diversos testes para averiguar eficiência, robustez e desempenho. Conclui-se que o método teve melhor desempenho em relação às funções objetivo do que o planejamento feito pela empresa, tanto nas demonstrações do Método da Soma Ponderada, quanto na utilização do método ϵ -restrito, isto é, nenhuma solução gerada foi dominada pelo planejamento realizado pela empresa. Ademais, a solução da empresa encontrou-se razoavelmente distante das fronteiras. Além disso, as fronteiras obtidas pelo método do ϵ -restrito foram razoáveis e possuem boa variabilidade de soluções, garantindo um bom leque de opções para o tomador de decisão.

Entretanto, deve-se considerar que o conjunto Pareto-ótimo não é conhecido e, assim, não se sabe se as soluções encontradas estão realmente próximas à Fronteira Pareto-ótima. No entanto, para fins práticos, os resultados atingidos têm potencial de ganho considerável em relação ao cenário atual.

Além disso, averiguou-se que o GRASP resulta em soluções efetivamente robustas em relação às soluções dadas pela empresa, sendo 73% e 81% das soluções geradas pelo algoritmo para 2014 e 2015, respectivamente, mais robustas do que as soluções geradas pela

empresa.

Em relação as características como tempo de estoque médio e uso de capacidade de máquina, as duas abordagens também mostraram diferenças. Os valores médios de tempo de estoque se mostraram quase 12% e 4% a mais do que o praticado pela empresa nos anos 2015 e 2014, respectivamente. Já o uso de máquina se mostrou superior, em média, em 2014 e equivalente em 2015.

Conclui-se que, relativamente, os resultados para 2015 foram melhores do que para 2014. Visualmente, as fronteiras geradas foram melhor distribuídas, as soluções foram mais robustas e os tempos de estoque maiores para a base de dados de 2015.

Finalmente, averiguou-se o tempo computacional do método. O tempo de computação do GRASP para um único objetivo com seleção de soluções robustas está em média de 7,85 minutos com desvio-padrão de 0,59 minutos. O método multiobjetivo, ϵ -restrito, demanda tempo computacional superior, pois executa o GRASP várias vezes. Já analisando a média e desvio-padrão dos valores das funções objetivo encontradas, pode-se ver que o algoritmo mono-objetivo possui certa repetibilidade, pois o desvio-padrão é relativamente baixo em relação a média. Além disso, quando se combina o GRASP com o método de seleção de soluções robustas, vê-se um leve aumento no desvio-padrão das funções objetivo, pois o método prefere soluções robustas, podendo comprometer o desempenho.

Para se analisar a repetibilidade do método multiobjetivo, aplicou-se, como tomada de decisão, o método de seleção de soluções robustas e, portanto, com apenas uma solução para se analisar, foi possível realizar as estatísticas presentes na Tabela 4.3. Verificou-se que, com esta análise, a repetibilidade está pequena, pois a gama de soluções encontradas compreende quase completamente a fronteira na Figura 4.3a. Entretanto, sabe-se que existem na literatura indicadores mais eficazes para se determinar o desempenho de métodos multiobjetivo. Dentre eles, pode-se citar o método do hiper-volume ou outros diversos métodos descritos em [11].

5.1 Contribuições do Trabalho

1. Estruturação matemática e computacional do processo de planejamento de colheita;
2. Desenvolvimento de um algoritmo rápido e eficiente para realizar planejamento e, conseqüentemente, espera-se que o número de homens-hora necessário seja reduzido;
3. Ganho de eficiência ao setor de colheita devido a melhores soluções geradas pelo algoritmo. Dessa forma, espera-se redução de custo de diesel, manutenção de estradas e máquinas, pois são utilizadas com maior eficiência.

5.2 Sugestões para trabalhos futuros

1. Investigar e acompanhar os impactos das soluções obtidas pelo algoritmo nos resultados da VFL;
2. Analisar a sensibilidade do método em relação aos seus parâmetros e obter, então, um conjunto razoável para cada instância do problema, i.e. ano de planejamento;
3. Analisar e comparar os resultados com outras meta-heurísticas, como os algoritmos genéticos, e utilizar indicadores de desempenho próprios para métodos de otimização multiobjetivo;

5.3 Considerações Finais

Devido ao bom desempenho dos algoritmos desenvolvidos frente aos planejamento atual da empresa, estes foram integrados a um software desenvolvido, em C# e framework .Net 4.0, exclusivamente para a equipe de colheita. Algumas imagens do software se encontram na Figura 5.1, sendo a Figura 5.1a, a tela de abertura do software e a Figura 5.1b, uma das telas de análise da solução. Esse software, além de executar os algoritmos e fornecer uma interface para lidar com alguns parâmetros, oferece ao usuário ferramentas para gestão e acompanhamento do processo de colheita. Assim, é possível gerar soluções otimizadas, as modificar e visualizar por meio de mapas, Gráficos de Gantt e tabelas. Também, é possível realizar a gestão do planejado e realizado e acompanhamento do processo como um todo.

Os algoritmos de otimização também foram implementados em *multithreads*, o que garante maior rapidez e eficiência, até mesmo em computadores domésticos.



Figura 5.1: Software desenvolvido para planejamento e gestão do processo de colheita da empresa

Referências Bibliográficas

- [1] José Rodrigo Banhara, Luiz Carlos Estraviz Rodriguez, Fernando Seixas, José Mauro Magalhães Ávila Paz Moreira, Lana Mirian Santos da Silva, Silvana Ribeiro Nobre, and Andrew Cogswell. Agendamento otimizado da colheita de madeira de eucaliptos sob restrições operacionais, espaciais e climáticas. *Scientia Florestalis*, 38(85):85—95, 2010.
- [2] Emin Zeki Baskent and Sedat Keles. Spatial forest planning: A review. *Ecological Modelling*, 188(2-4):145–173, 2005.
- [3] Mokhtar S Bazaraa, Hanif D Sherali, and Chitharanjan Marakada Shetty. *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.
- [4] Pete Bettinger, John Sessions, and Kevin Boston. A review of the status and use of validation procedures for heuristics used in forest planning. *Mathematical and Computational Forestry & Natural-Resource Sciences (MCFNS)*, 1(1):Pages—26, 2009.
- [5] Christian Blum, Jakob Puchinger, Günther R Raidl, and Andrea Roli. Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*, 11(6):4135–4151, 2011.
- [6] Carlos Bohle, Sergio Maturana, and Jorge Vera. A robust optimization approach to wine grape harvesting scheduling. *European Journal of Operational Research*, 200(1):245–252, 2010.
- [7] Mourad Boudia, Mohamed Aly Ould Louly, and Christian Prins. A reactive GRASP and path relinking for a combined production–distribution problem. *Computers & Operations Research*, 34(11):3402–3419, 2007.
- [8] Ilhem Boussaïd, Julien Lepagnot, and Patrick Siarry. A survey on optimization metaheuristics. *Information Sciences*, 237:82–117, 2013.
- [9] Nadjib Brahimi, Stéphane Dauzere-Peres, Najib M Najid, and Atle Nordli. Single item lot sizing problems. *European Journal of Operational Research*, 168(1):1–16, 2006.
- [10] Jacques Carlier. The one-machine sequencing problem. *European Journal of Operational Research*, 11(1):42–47, 1982.
- [11] Yann Collette and Patrick Siarry. *Multiobjective optimization: principles and case studies*. Springer Science & Business Media, 2003.
- [12] Mateus Rocha De Paula, Martín Gómez Ravetti, Geraldo Robson Mateus, and Panos M Pardalos. Solving parallel machines scheduling problems with sequence-dependent setup times using variable neighbourhood search. *IMA Journal of Management Mathematics*, 18(2):101–115, 2007.
- [13] Johann Dréo. *Metaheuristics for hard optimization: methods and case studies*. Springer Science & Business Media, 2006.

- [14] Andreas Drexl and Alf Kimms. Lot sizing and scheduling - survey and extensions. *European Journal of operational research*, 99(2):221–235, 1997.
- [15] Matthias Ehrgott. *Multicriteria optimization*. Springer Science & Business Media, 2006.
- [16] Thomas A Feo and Mauricio G C Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters*, 8(2):67–71, 1989.
- [17] Thomas A Feo and Mauricio G C Resende. Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133, 1995.
- [18] Paola Festa and Mauricio G C Resende. An annotated bibliography of GRASP–Part II: Applications. *International Transactions in Operational Research*, 16(2):131–172, 2009.
- [19] Bernhard Fleischmann and Herbert Meyr. The general lotsizing and scheduling problem. *Operations-Research-Spektrum*, 19(1):11–21, 1997.
- [20] M Gen and R Cheng. *Genetic Algorithms and Engineering Optimization*. A Wiley-Interscience publication. Wiley, 2000.
- [21] Michel Gendreau and Jean-Yves Potvin. *Handbook of metaheuristics*, volume 2. Springer, 2010.
- [22] Lucas Rezende Gomide, Julio Eduardo Arce, and Arinei Carlos Lindbeck da Silva. Uso do algoritmo genético no planejamento florestal considerando seus operadores de seleção. *Cerne*, 15(4):460, 2009.
- [23] Johnny C Ho and Yih-Long Chang. A new heuristic for the n-job, M-machine flow-shop problem. *European Journal of Operational Research*, 52(2):194–202, 1991.
- [24] Yaochu Jin and Jürgen Branke. Evolutionary optimization in uncertain environments-a survey. *Evolutionary Computation, IEEE Transactions on*, 9(3):303–317, 2005.
- [25] Jenny Karlsson, Mikael Rönnqvist, and Johan Bergström. An optimization model for annual harvest planning. *Canadian Journal of Forest Research*, 34(8):1747–1754, 2004.
- [26] Marco Laumanns, Lothar Thiele, and Eckart Zitzler. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169(3):932–942, 2006.
- [27] Muhammad Nawaz, E Emory Enscore, and Inyong Ham. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1):91–95, 1983.
- [28] Cristian D Palma and John D Nelson. A robust optimization approach protected harvest scheduling decisions against uncertainty. *Canadian journal of forest research*, 39(2):342–355, 2009.
- [29] Fábio Pavinato and Guilherme Souza. *Desenvolvimento de um Modelo Matemático para Otimização do Planejamento do Setor de Colheita em uma Empresa do Ramo Florestal*. Trabalho de graduação, Universidade Federal de Minas Gerais, 2013.
- [30] Michael L Pinedo. *Scheduling: theory, algorithms, and systems*. Springer Science & Business Media, 2012.
- [31] Daniel Quadt and Heinrich Kuhn. Production planning in semiconductor assembly. In *Proceedings of the Fourth Aegean International Conference on Analysis of Manufacturing Systems*, pages 181–189, 2003.

- [32] S S Rao. *Engineering Optimization: Theory and Practice*. Wiley, 2009.
- [33] Tapabrata Ray. Constrained robust optimal design using a multiobjective evolutionary algorithm. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 1, pages 419–424. IEEE, 2002.
- [34] Amir Salehipour, Kenneth Sörensen, Peter Goos, and Olli Bräysy. Efficient GRASP+ VND and GRASP+ VNS metaheuristics for the traveling repairman problem. *4OR*, 9(2):189–209, 2011.
- [35] Patrick Schittekat, Joris Kinable, Kenneth Sörensen, Marc Sevaux, Frits Spieksma, and Johan Springael. A metaheuristic for the school bus routing problem with bus stop selection. *European Journal of Operational Research*, 229(2):518–528, 2013.
- [36] Ozlem Senvar, Ebru Turanoglu, and Cengiz Kahraman. *Meta-Heuristics Optimization Algorithms in Engineering, Business, Economics, and Finance*. IGI Global, 2013.
- [37] Kenneth Sörensen. A framework for robust and flexible optimisation using metaheuristics. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1(4):341–345, 2003.
- [38] Jeffrey D Ullman. NP-complete scheduling problems. *Journal of Computer and System sciences*, 10(3):384–393, 1975.