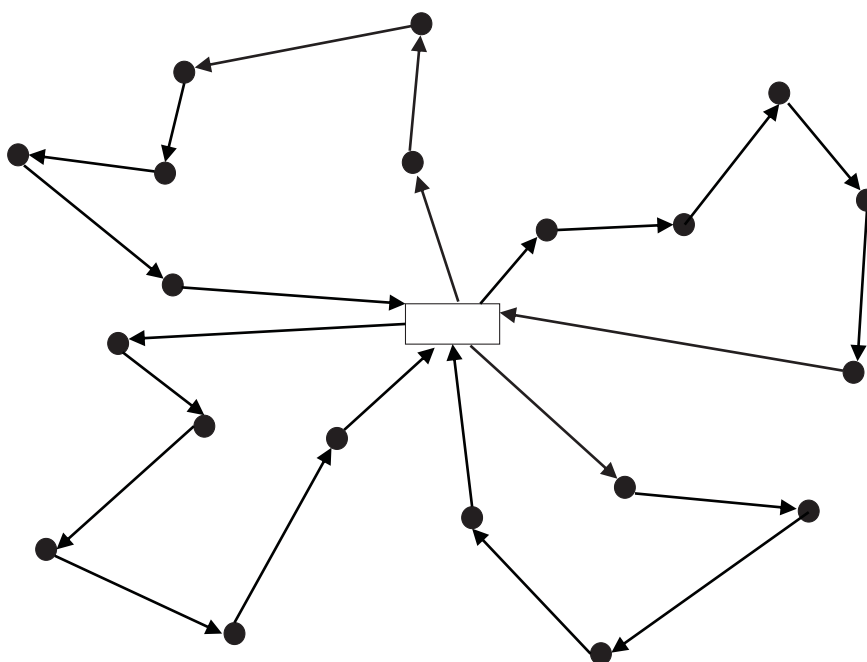


Guilherme Gomes Neves de Oliveira



Uso de algoritmos genéticos na otimização do sistema de coleta do palhico de cana-de-açúcar

Monografia para Conclusão de Curso

Orientadora: Prof^a. Dr^a. Silvely N. A. S. Néia
Co-Orientadora: Prof^a. Dr^a. Helenice de O. F Silva

Guilherme Gomes Neves de Oliveira

*Uso de algoritmos genéticos na otimização
do sistema de coleta do palhiço de
cana-de-açúcar*

Monografia apresentada na Faculdade de Ciências e Tecnologia da Universidade Estadual Paulista “JÚLIO DE MESQUITA FILHO” - UNESP - Câmpus de Presidente Prudente, como requisito obrigatório para a conclusão do Curso de Bacharelado em Ciência da Computação.

Orientadora: Prof^a. Dr^a. Silvely N. A. S. Néia

Co-orientadora: Prof^a. Dr^a. Helenice de O. F Silva

Presidente Prudente

2010

Termo de Aprovação

Guilherme Gomes Neves de Oliveira

Uso de algoritmos genéticos na otimização do sistema de coleta do palhão de cana-de-açúcar

Monografia sob o título “**Uso de algoritmos genéticos na otimização do sistema de coleta do palhão de cana-de-açúcar**”, defendida por Guilherme Gomes Neves de Oliveira e aprovada em 13 de dezembro de 2010, em Presidente Prudente, Estado de São Paulo, pela banca examinadora constituída pelos doutores:

Prof^a. Dr^a. Silvely N. A. S. Néia
UNESP - FCT Presidente Prudente

Prof. Dr. Almir Olivetti Artero
UNESP - FCT Presidente Prudente

Prof. Dr. Maurício Araújo Dias
UNESP - FCT Presidente Prudente

Presidente Prudente, 13 de dezembro de 2010

”Pois como a criação do universo é a suprema perfeição e trabalho de um Criador de suprema sabedoria, nada no universo tem lugar sem que uma regra de máximo ou mínimo se apresente.”

Leonhard Euler, 1707 - 1783

Agradecimentos

A Deus, que me dá forças e está sempre presente em minha vida.

A meus pais e familiares, pelo carinho, amor e estímulo nos momentos mais difíceis.

A minha orientadora, professora e amiga Silvely N. de A. S. Néia pelos ensinamentos transmitidos, apoio, orientação e incentivo durante toda a realização deste trabalho.

A professora Helenice de Oliveira Florentino Silva da Faculdade de Ciências Agronômicas da Unesp, campus de Botucatu, por prestar o seu apoio e viabilizar a realização deste trabalho.

A todos os mestres com quem tive o prazer e oportunidade de aprender e conviver ao longo desses 5 anos e aos queridos amigos.

Resumo

A preocupação com o meio ambiente e as novas leis ambientais têm feito com que empresas produtoras de cana-de-açúcar busquem reduzir a queima do canavial e melhorar as formas de corte e colheita da cana crua. O corte mecanizado dá origem ao palhiço de cana-de-açúcar, formando um ambiente propício para o surgimento de pragas e para o atraso no brotamento da cana. Várias usinas têm transportado o palhiço do campo para o centro de processamento, no intuito de serem reaproveitados como fonte de energia. Porém a inserção deste processo na colheita gera grandes dificuldades às usinas, no que diz respeito à tecnologia de coleta e ao aproveitamento do palhiço, devido ao custo envolvido no sistema de transporte.

Para reduzir tais problemas, propõe-se neste trabalho, uma formulação do problema de coleta dos fardos de palhiço de cana-de-açúcar do campo ao centro de processamento para a geração de energia, como um problema de transporte, que busca minimizar a distância percorrida pelos veículos evitando a formação de filas de carga e descarga na usina.

Para resolver o problema foram estudadas metaheurísticas baseadas no algoritmo genético.

palavras-chave: Otimização. Problemas de transporte. Palhiço da cana-de-açúcar. Algoritmos Genéticos.

Abstract

The Brazilian government has convinced the world that ethanol deriving from sugar cane is a promissory means of sustainable fuel for vehicles. There is a great growth of flex vehicles , i.e, run both by ethanol and gasoline, due to competent automotive industries and efficient alcohol production technology. In 2009 and 2010 the ethanol production was 25.7 billion liters and 53.8% of sugar cane production was destined to alcohol production.

Nevertheless, the sugar production also derived from sugar cane should increase in 2011. Brazil produced 33 million tons of sugar in the last harvest. With sugar cane on the rise production is arising new environmental problems. The harvest using mechanized cut besides improving the logistic transportation system leaves the generating residue in the field. This residue is a mixture of straw, leavings and scrap of sugar cane named sugar cane crop residue and corresponds to 30% of biomass and can be burned and produce electricity by cogeneration.

But the transport the sugar cane crop from the field is expensive due costs involved in the transport system. This work aims to propose a formulation for the bales collecting problem from sugar cane field to mill that minimize the costs involved in the transport system. The computational tests use the C++ language and an algorithm based on genetic algorithms techniques.

keywords: Optimization. Transportation problems. Sugarcane straw. Genetic Algorithms.

Sumário

1	Introdução	p. 2
1.1	Justificativa e relevância do tema	p. 2
1.2	Objetivos do Trabalho	p. 3
1.3	Organização do Trabalho	p. 4
2	Fundamentação Teórica	p. 5
2.1	O Palhiço	p. 5
2.1.1	A coleta e a recepção do palhiço de cana-de-açúcar	p. 6
2.1.2	Legislação do transporte do palhiço	p. 8
2.2	Modelagem de problemas	p. 8
2.2.1	Modelos matemáticos do sistema de produção de cana-de-açúcar	p. 9
2.2.2	Modelos de planejamento e designação de veículos	p. 10
2.2.2.1	O problema de coleta de resíduos de madeira para fins energéticos estudado por Galvão(2004)	p. 11
2.2.2.2	A heurística de Van Vliet; Boender; Rinnooy Kan (1992) para o sistema de distribuição de açúcar	p. 12
2.3	Programação linear inteira (PLI)	p. 13
2.3.1	Problema geral	p. 13
2.3.2	Métodos de solução para problemas de programação linear inteira	p. 14

2.3.3	Métodos Exatos	p. 14
2.3.3.1	Branch and Bound (BB)	p. 14
2.3.3.2	Programação Dinâmica (PD)	p. 15
2.3.4	Métodos Aproximados	p. 15
2.3.4.1	Relaxações	p. 16
2.3.4.2	Relaxação Lagrangeana	p. 16
2.3.4.3	Heurísticas	p. 17
2.4	Algoritmos Genéticos	p. 17
2.4.1	Genética Biológica	p. 18
2.4.2	Propriedades dos algoritmos genéticos	p. 19
2.4.3	Codificação	p. 20
2.4.4	Operadores Genéticos	p. 21
2.4.4.1	Seleção	p. 22
2.4.4.2	Cruzamento	p. 22
2.4.4.3	Mutação	p. 24
2.4.4.4	Elitismo	p. 24
2.4.4.5	Criacionismo	p. 24
2.4.4.6	Inicialização da População	p. 25
2.4.5	Algoritmos genéticos e problemas de transporte	p. 25
2.4.5.1	Algoritmo genético para a expedição de veículos por Leclerc; Potvin (1997)	p. 25
2.4.5.2	Algoritmos genéticos aplicados ao planejamento do trans- porte principal de madeira proposto por Souza (2004) .	p. 26
3	O Problema de Coleta do Palhão de Cana-de-açúcar	p. 27
3.1	Descrição do problema	p. 27
3.2	Definição do problema	p. 28

3.3	Modelo matemático	p. 29
4	Metodologia	p. 32
4.1	A estratégia de determinação das rotas e a construção do algoritmo genético	p. 33
4.1.1	Descrição funcional do algoritmo genético	p. 34
4.1.1.1	População inicial	p. 34
4.1.1.2	Representação do indivíduo	p. 35
4.1.1.3	Avaliação da população	p. 36
4.1.1.4	Cr�terios de parada	p. 37
4.1.1.5	Operador de Sele�o	p. 37
4.1.1.6	Operador de Cruzamento	p. 37
4.1.1.7	Operador de Mutac�o	p. 38
4.1.1.8	Mortalidade	p. 39
4.2	A sele�o dos melhores padr�es gerados pelo algoritmo gen�tico	p. 39
5	Implementa�o, Experi�ncias e Resultados	p. 40
5.1	Os dados do problema	p. 40
5.2	O software	p. 40
5.3	A ferramenta Xpress-Optimizer	p. 42
5.4	Experi�ncia e resultados obtidos	p. 43
6	Conclus�o e Trabalhos Futuros	p. 47
6.1	Trabalhos Futuros	p. 48
	Refer�ncias	p. 49
	Anexo A – Local Talh�es	p. 51
	Anexo B – Talh�es	p. 52

Anexo C - Frota	p. 53
Apêndice A - Código Fonte Calculo Aptidão	p. 54
Apêndice B - Código Fonte Seleção	p. 56
Apêndice C - Código Fonte Cruzamento	p. 57
Apêndice D - Código Fonte Mutação	p. 58
Apêndice E - Código Xpress-Mosel	p. 59

Lista de Figuras

1	Palhiço de cana-de-açúcar	p. 5
2	Queima do palhiço	p. 6
3	Enleiramento do palhiço	p. 7
4	Carregamento do palhiço	p. 7
5	Representação do Cruzamento (<i>crossover</i>)	p. 18
6	Representação da mutação	p. 18
7	Fluxo de execução do algoritmo genético	p. 20
8	Ilustração do método da roleta	p. 22
9	Exemplo de ambiente composto pela usina e os talhões	p. 27
10	Fluxograma da colheita e recepção do palhiço da cana-de-açúcar	p. 28
11	Esquema geral da solução	p. 33
12	Representação do cromossomo	p. 35
13	Exemplo de cruzamento clássico entre dois padrões	p. 37
14	Estratégia de determinação do ponto de corte	p. 38
15	Exemplo de mutação entre dois padrões	p. 38
16	Imagem do software desenvolvido	p. 41
17	Exemplo e ilustração do uso do Xpress-Optimizer	p. 43
18	Representação gráfica dos resultados obtidos	p. 46

Lista de Tabelas

1	Analogia entre a biologia e algoritmos genéticos	p. 19
2	Relacionamento de problemas x forma de representação	p. 21
3	Exemplo do cálculo da aptidão para um único veículo	p.36
4	Padrões individuais obtidos pelo software	p.44
5	Melhores resultados obtidos em um primeiro experimento	p.45
6	Resultado obtido em uma segunda experiência	p.45

Introdução

1.1 Justificativa e relevância do tema

O Brasil é o maior produtor de cana-de-açúcar do mundo. A produção é voltada principalmente para obtenção de álcool e açúcar. Segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), na safra de 2008/2009 foram colhidas 572,57 milhões de toneladas, um aumento de 17,9% em relação a safra de 2007/2008, e ainda, o país possui 354 usinas e destilarias, que geram cerca de 40 bilhões de reais por ano, cerca de 2,35% do PIB nacional e mantém 3,6 milhões de empregos, o que revela a grande importância de tal cultura na economia do país.

A grande preocupação hoje do governo e de ambientalistas é no destino dado ao palhico da cana-de-açúcar gerado após a colheita mecanizada, que até então normalmente durante a noite é queimado antes do início de uma nova colheita. O problema é que tal prática libera gases que poluem a atmosfera e causam problemas de saúde, o fogo gerado danifica redes elétricas, destrói reservas, mananciais e até mesmo residências.

Diversos pesquisadores Spadotto(2006) e Ripoli (2002) mostram como é possível aproveitar os resíduos de cana-de-açúcar não só como fonte de energia através da produção de biodiesel, mas também na forma de ração animal ou até mesmo como energia para as usinas, que levam a vantagens ambientais e econômicas, tais como a redução da utilização de recursos energéticos de fontes naturais e a geração de empregos. Porém, a grande dificuldade para o aproveitamento desse recurso é o alto custo que o processo demanda, com destaque para o custo gerado ao sistema de transporte, no que diz respeito à capacidade da frota, logística de transporte, gastos com combustível e manutenção de veículos.

Neste contexto, com a adição desse novo agravante a agroindústria sucroalcooleira nacional vem, nas últimas décadas, investindo em pesquisa e desenvolvimento, no intuito de aumentar a produtividade e reduzir os custos de produção. As usinas de cana-de-açúcar procuram se adequar ao mercado cada vez mais competitivo, com a implantação de técnicas, equipamentos e recursos que beneficiem o planejamento e o controle do processo produtivo de forma sustentável. Sendo assim a grande dificuldade está em como coordenar esses processos de corte, carregamento e transporte do palhicho e da cana-de-açúcar crua, do campo até a usina, de forma a suprir adequadamente as moendas, e ainda reduzir seus custos.

Segundo os problemas relatados acima, o foco da pesquisa que está sendo proposta será o de reduzir os custos gerados com a inserção do transporte do palhicho, no que diz respeito à otimização do uso da frota disponível na coleta do palhicho. Pretende-se abordar o transporte dos resíduos dos talhões (áreas divididas no plantio), como um problema de roteamento, mais especificamente como um problema de programação e designação de veículos como proposto por Souza (2004).

1.2 Objetivos do Trabalho

O objetivo deste trabalho é estudar o sistema de transporte e coleta dos resíduos de cana-de-açúcar do campo para a usina, visando o seu reaproveitamento desde matéria-prima na produção industrial até combustível para abastecimento de fábricas ou da própria usina. De maneira específica esse problema consiste em transportar os fardos de palhicho, de tal forma que reduza o percurso de coleta dos caminhões, respeitando a capacidade da frota disponível e que toda a demanda seja atendida.

O referido trabalho procura também contribuir na erradicação da queima do palhicho de cana-de-açúcar, assim como estimular a redução do uso de combustíveis fósseis, sendo tais ações importantíssimas para redução da poluição industrial e a preservação do meio ambiente.

Para realizar a modelagem matemática do referido problema utilizamos as teorias do problema de planejamento e designação de veículos, em que a idéia básica será citada mais adiante, podendo ser encontrada também em Galvão (2004).

Os estudos e conhecimentos adquiridos na elaboração deste trabalho puderam agregar e potencializar valores à formação do aluno.

Os resultados do trabalho foram analisados, a fim de se comprovar a melhora, ou não, no sistema de transporte do palhiço da cana-de-açúcar.

1.3 Organização do Trabalho

O presente trabalho está estruturado em capítulos, organizados de maneira a facilitar o entendimento do problema de coleta do palhiço de cana-de-açúcar e expor o estado da arte através da fundamentação teórica, a qual serviu de base para a elaboração da metodologia proposta, tal que:

O primeiro capítulo consiste de uma introdução ao tema, destacando a justificativa de sua escolha e descrevendo aspectos relacionados ao estudo;

O Capítulo 2 corresponde à revisão bibliográfica, onde são apresentados os conceitos básicos sobre palhiço e seu transporte. Traz ainda a aplicação da modelagem de problemas em programação linear inteira e suas áreas de aplicação. É introduzido o conceito de classe de problemas com ênfase para a classe de problemas NP - completos e apresentada a heurística dos algoritmos genéticos, sua origem, fundamentos e componentes, além de exemplos da aplicação dos mesmos em problemas de transporte;

No Capítulo 3, é apresentado o problema de programação e designação de veículos na coleta do palhiço de cana-de-açúcar, tema do presente trabalho; No Capítulo 4, é descrita a metodologia adotada para resolver o problema proposto e sua aplicação, através do método escolhido;

O Capítulo 5 descreve as experiências realizadas, bem como os resultados obtidos pelo método; O último capítulo faz uma discussão dos objetivos propostos e os alcançados, a metodologia utilizada, fazendo uma análise do trabalho desenvolvido bem como a descrição de trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo, é apresentada a logística que envolve o palhiço, os conceitos de modelagem de problemas, exibindo trabalhos relevantes relacionados ao tema desta pesquisa.

2.1 O Palhiço

A denominação utilizada de forma errônea segundo Ripoli (2002), tem sido "palha", quando o certo tecnicamente seria "palhiço", porque após a colheita da cana para produção de álcool e açúcar, ficam depositados, além de folhas de cana de baixa umidade (secas), também folhas verdes, palhas, ponteiros, colmos e reboulos e suas frações, com ou sem terra. O mesmo pode ser visto segundo a Figura 1 abaixo.



Figura 1: Palhiço de cana-de-açúcar
FONTE: Ripoli (2002).

Para preparar a terra para um novo plantio, é comum queimar o palhiço. No entanto,

o crescimento da cultura de cana-de-açúcar nos últimos anos, originou alguns problemas ao setor sucroalcooleiro, tais como os causados pela queima do palhiço, representado pela Figura 2. O referido problema é tema de discussão de ambientalistas e políticos, sendo que algumas leis têm sido impostas, estabelecendo prazos para erradicar a queima do mesmo na pré-colheita. Portanto, o destino desse resíduo tem sido alvo de diversas pesquisas atualmente.



Figura 2: Queima do palhiço
FONTE: Ripoli (2002).

O estudos com base na biomassa residual (palhiço) tem identificado inúmeras formas de destinação e reaproveitamento para a mesma. O seu alto poder calorífico pode ser utilizado como fonte de energia elétrica, por exemplo, em substituição aos resíduos naturais como carvão mineral, além disso a presença de minerais e sacarose, o transformam em matéria prima para fabricação de rações e adubos.

No entanto, uma das dificuldades encontradas em seu aproveitamento está ligada ao carregamento e transporte, pois não existe maquinário específico para manipular esse tipo de resíduo, e normalmente a frota é a mesma utilizada no carregamento da cana crua, o que agrava a logística de colheita e recepção de todo o processo, contribuindo para a formação de filas nos pontos de descarga e diminuição da vida útil do caminhão.

2.1.1 A coleta e a recepção do palhiço de cana-de-açúcar

A coleta do palhiço visando o seu aproveitamento é basicamente realizada da seguinte maneira: primeiro o palhiço é enleirado e posteriormente enfardado nos formatos cilíndrico ou prismático, por fim esses fardos são colocados em caminhões e transportados até o

centro de processamento, que geralmente fica localizado no pátio da usina. Os processos de enleiramento e carregamento podem ser vistos nas Figuras 3 e 4.

No sistema de recepção, compreendem-se as operações de pesagem, amostragem, armazenamento e descarga de cana-de-açúcar e estocagem, limpeza e descarga do palhiço, que operam com o fluxo uniforme de cana e palhiço transportados do campo até a usina a fim de permitir a alimentação adequada das moendas, evitando a paralisação das mesmas, o que é altamente prejudicial à usina.



Figura 3: Enleiramento do palhiço
FONTE: Ripoli (2006).



Figura 4: Carregamento do palhiço
FONTE: Ripoli (2006).

No entanto, o fluxo ideal de transporte do campo à usina pode sofrer variações provocadas por diversos fatores tais como: o tipo de corte, manual ou mecanizado, fatores ambientais como o clima, geográficos como a localização e deformidade dos talhões no

campo de colheita, até operacionais como o tipo de equipamento utilizado, colhedoras, tratores, entre outros.

A dependência entre tais processos contribui para o surgimento de tempos não produtivos, provocados pela ociosidade de caminhões nos pátios de descarga, formando grandes filas, gerando custos com combustível e perda de mão de obra, devido à falta que os caminhões fazem no campo, problema este que envolve máquinas e operários.

Sendo assim é necessário determinar um modelo para o planejamento do transporte dos fardos, de tal forma que maximize a quantidade de palhiço transportado e reduza o percurso dos caminhões.

2.1.2 Legislação do transporte do palhiço

O transporte de palhiço, assim como de outros produtos deve obedecer às resoluções impostas pelo Conselho Nacional de Trânsito (Contran), órgão que estabelece as diretrizes da Política Nacional de Trânsito.

Conforme a Resolução nº12/98 do Contran, o limite máximo autorizado de peso bruto total é de 45t por unidade ou combinação de veículos ou 10t de peso bruto por eixo isolado, nas superfícies das vias públicas. O mesmo estabelece ainda para veículos, com ou sem carga, a largura máxima de 2,60 m e altura de 4,40 m. O limite para o comprimento é de 14 m, para veículos simples, 18,15 m para veículos articulados e 19,80 m para veículos com reboque.

Uma vez não cumpridas tais exigências, não é permitido o registro e licenciamento do veículo, sendo necessária ainda a concessão de uma autorização específica anual, fornecida pela autoridade com a circunscrição sobre a via e considerando os limites desta via. Esta autorização tem validade de um ano podendo ser renovada até o sucateamento da frota, desde que obedeça ao volume de tráfego e traçado da via.

2.2 Modelagem de problemas

A representação da realidade sempre se fez necessária para a humanidade, devido a diferentes aspectos, sejam eles de caráter econômico, prático ou quanto ao grau de complexidade do problema. O processo de modelagem de problemas tornou-se um mecanismo essencial na busca por soluções viáveis, através da formulação de modelos representativos, conforme Arenales (2007).

O processo de modelagem do problema é dividido em três etapas: definição do problema, formulação, solução e validação.

A definição de um problema de programação linear é composta pela identificação do objetivo do problema (maximizar ou minimizar), da identificação e definição das variáveis e restrições.

A formulação consiste na codificação propriamente dita, onde normalmente os objetos são representados matematicamente, por equações, expressões e fórmulas. Esta etapa depende muito da habilidade e experiência do modelador, e por isso para a maioria, é considerada a etapa mais difícil do processo.

A etapa de solução e validação trata da resolução do sistema codificado na fase de formulação, através de técnicas adequadas ao tipo de problema estudado. Tal solução pode ser validada de diferentes maneiras, seja, através de simulações, análise do protótipo ou matematicamente.

A referida abordagem é muito utilizada na expressão de modelos em Pesquisa Operacional, por tratar de problemas de decisão, onde se faz o uso de modelos matemáticos. Este ramo do conhecimento busca a otimização de um processo, seja pela maximização ou minimização de certo objetivo, sujeito a determinadas condições, também conhecidas como restrições. Para a resolução de tais problemas é indispensável o uso de ferramentas computacionais, devido à grande quantidade de variáveis e restrições necessárias.

A representação matemática geral de um problema de otimização é esboçada da seguinte maneira:

(P) Maximizar a função objetivo $f(x)$

Sujeito a restrições:

$$h_i(x) = 0, \quad i = 1, \dots, p$$

$$g_j(x) \leq 0, \quad j = 1, \dots, q$$

$$x \in R^n$$

2.2.1 Modelos matemáticos do sistema de produção de cana-de-açúcar

Existem na literatura diversos trabalhos com o objetivo de estimar a produtividade da cana-de-açúcar, que foram desenvolvidos utilizando modelagem matemática. As justificativas de se utilizar tais modelos são a busca por altos rendimentos e baixos custos,

reduzindo a relação entre os fatores de produção na busca do melhor desempenho.

Curverwell (1984), utilizando o método FRPS (Serviço de Processamento de Recordes de Produtividade), desenvolveu um modelo de previsão de produção, aplicado de forma experimental na África do Sul, onde o mesmo levava em consideração dados climáticos, épocas de corte e teor de sacarose. Com os dados obtidos foi concluído que a ferramenta é muito importante no gerenciamento da lavoura de cana-de-açúcar. Beauclair e Penteado (1984) relataram a importância dos modelos matemáticos na previsão de maturação da cana, que possibilita a definição de formas de manejo de variedades. Pereira e Machado (1986) descreveram um modelo matemático de balanço de carbono, integrando o crescimento, fotossíntese e respiração de uma única equação. Tal equação descreve o crescimento e o desenvolvimento da cana-de-açúcar.

Boote, Jones e Pickering (1996) relataram alguns modelos de produtividade, avaliando o risco de produção da cana em intervalos climáticos de uma determinada região. em que foi possível analisar as melhores datas de plantio, a densidade de plantio, o espaçamento, a escolha da variedade e a aplicação de fertilizante conforme determinado tipo de solo.

Florentino (2005) e Spadotto (2008) apresentaram modelos que respectivamente consistiam em otimizar o processo de transferência e de carregamento do palhico da cana-de-açúcar do campo para o centro de processamento visando o seu reaproveitamento.

Como visto acima, existe uma ampla difusão na utilização de modelos matemáticos na representação de problemas reais, que somados a importância que a cultura da cana-de-açúcar tem para a economia nacional, justificam a proposta deste trabalho.

2.2.2 Modelos de planejamento e designação de veículos

Assim como os modelos citados de produção e transporte de cana-de-açúcar e afins, nesta seção são discutidos os principais modelos de planejamento e designação de veículos de carga. Tal referência é importante devido à semelhança de tais modelos com o problema em questão, sendo estes a base para a modelagem do problema de coleta do palhico de cana-de-açúcar.

Os problemas de planejamento e designação de veículos (carros, caminhões, ônibus, navios e etc..), em geral consistem em programar uma sequência de viagens e designar as respectivas rotas dos veículos, que satisfaçam um conjunto de restrições, e ao mesmo tempo otimizem a função objetivo do problema.

2.2.2.1 O problema de coleta de resíduos de madeira para fins energéticos estudado por Galvão(2004)

Galvão(2004) desenvolveu uma heurística para otimizar a programação e designação diária de veículos na coleta de resíduos de madeira para fins energéticos.

Os resíduos são produzidos de diferentes formas e por uma grande quantidade de empresas como serrarias, centros urbanos com a poda das árvores e indústrias em geral que utilizam embalagens de madeira. Estas, uma vez cientes de que podem lucrar com a venda dos resíduos, bem como dos problemas ambientais e multas causados pela destinação incorreta deste material, estimulam o seu aproveitamento.

O problema no geral consiste em atender a um custo mínimo as necessidades de abastecimento de combustível para uma central, que deve produzir energia continuamente durante todo o ano, sendo que os combustíveis, neste caso, são os resíduos de madeira. Neste caso, dado um conjunto de potenciais geradoras de resíduos, deve-se selecionar um subconjunto que minimize o custo global (percurso), para o abastecimento de uma central, além de dimensionar a frota homogênea mínima necessária, bem como a programação das viagens de cada veículo.

Os veículos fazem viagens de ida e volta entre a central e cada fornecedor. O veículo tem capacidade máxima, respeitando as restrições de jornada de trabalho. Quanto aos fornecedores, é preciso conhecer algumas informações como: a taxa de geração de resíduos, distância até a central, preço unitário do resíduo, o custo de transporte, assim como os tempos de carregamento e descarregamento.

Na metodologia adotada pelo autor, o problema é dividido em dois subproblemas que são tratados separadamente, sendo:

- O **problema 1**: o de selecionar um subconjunto de geradoras que minimize o custo global de transporte e coleta dos resíduos.
- O **problema 2**: o de determinar os dias de coleta e designar a frota diária, segundo as restrições de jornada de trabalho dos veículos.

A heurística adotada pelo pesquisador, no sistema coleta de resíduos de madeira ajuda a entender e criar uma heurística para o problema de coleta do palhiço de cana-de-açúcar.

2.2.2.2 A heurística de Van Vliet; Boender; Rinnooy Kan (1992) para o sistema de distribuição de açúcar

Segundo referenciado por Galvão (2004), tal sistema de transporte estudado por Van Vliet; Boender; Rinnooy Kan (1992) é um dos poucos que utilizam alocação e sistema de coleta com veículos em viagens ponto a ponto, onde toda a carga do veículo é carregada e entregue.

O problema aborda um sistema de distribuição de açúcar a granel em uma cooperativa da Holanda chamada, Sulker Unie, responsável por processar mais da metade do açúcar comercializado no país, sendo o seu objetivo o de minimizar o custo do transporte segundo a distância percorrida, a remuneração dos motoristas, a disponibilidade e a capacidade dos caminhões, o horário de abertura das fábricas, a quantidade e o estoque diários e a jornada de trabalho dos motoristas. Tal cooperativa possui cinco fábricas que produzem diferentes tipos de açúcar, segundo sua granulação e cor, tal que na entressafra, mais da metade deste é comercializado e o restante é utilizado pela própria usina, na produção de açúcar em pó, cubo e melaço.

A distribuição aos clientes ocorre diariamente, segundo as especificações de granulometria e cor, quantidade, dia e hora da entrega. As fábricas podem produzir qualquer tipo de produto e possuem uma ou mais estações de carregamento de veículos, suprindo seus clientes sempre com a carga máxima do caminhão. A frota de caminhões é própria e heterogênea, podendo ser complementada com a locação de outros veículos, sendo que no final do expediente todos devem retornar as suas bases.

Assim como citado anteriormente o planejamento se dá através da definição da seqüência de viagens fechadas, iniciadas a partir de uma base pré-definida. Em resumo, o veículo é carregado em umas das bases e em seguida dirige-se a um cliente, sendo esta rotina repetida até o final de sua jornada.

Na metodologia adotada pelos autores, o problema é dividido em partes que inicialmente são tratadas separadamente.

Segundo os autores, mesmo após um ano operando na Suiker Unie, não é possível saber exatamente os resultados do sistema adotado, porém, no que diz respeito aos roteiros e à jornada de trabalho dos funcionários, a redução dos custos globais foi de aproximadamente 7%.

Galvão (2004) aponta também que a heurística adotada pelos pesquisadores no sistema de distribuição de açúcar a granel, ajudou a entender e solucionar o seu problema de coleta

de resíduos de madeira no que diz respeito a:

- O sistema de carregamento e descarregamento de um veículo por vez.
- O carregamento e entrega da capacidade total do veículo a um cliente.
- Os tempos de ciclo veiculares ponto a ponto.
- A solução do problema obtida pela sua decomposição.

Sendo assim, o mesmo contribui para o entendimento e solução da pesquisa proposta neste trabalho, que é a de minimizar o custo da coleta do palhiço de cana-de-açúcar.

2.3 Programação linear inteira (PLI)

Os problemas de programação linear inteira estão presentes em diversas áreas do mundo real, tais como energia, finanças, medicina, agricultura, aviação transportes entre outras. Segundo Dias (2004), todos eles visam resolver as mesmas categorias de problemas: otimização da utilização de recursos, sendo todas as variáveis utilizadas do tipo inteiro, pois, por exemplo, não há como alocar meio caminhão em uma viagem, ou decidir fabricar somente parte de um produto. A designação de um caminhão a frente de uma colheita de cana pode ser representada por valores inteiros, como: decidir qual caminhão irá carregar em uma determinada área, segundo sua capacidade.

2.3.1 Problema geral

A representação geral de um problema de programação linear poder ser dada por:

(PL) Maximizar

$$c^T x \tag{2.1}$$

Sujeito a:

$$Ax \leq b \quad x \geq 0$$

onde A é uma matriz $(m \times n)$, c é um vetor linha $(1 \times n)$, b é um vetor coluna $(m \times 1)$ e x um vetor de variáveis $(n \times 1)$.

Se todas as variáveis do problema forem inteiras, temos um problema de programação linear inteira (PLI).

(PLI) Maximizar

$$cx \tag{2.2}$$

Sujeito a:

$$Ax \leq b \quad x \geq 0 \quad x \text{ e } b \text{ são inteiros}$$

2.3.2 Métodos de solução para problemas de programação linear inteira

Existem inúmeros métodos de solução para problemas de programação linear inteira. Podemos dividi-los em métodos exatos de solução e métodos aproximados. Dentre os métodos exatos podemos destacar: o método *branch and bound (BB)* e a *programação dinâmica (PD)*. Já entre os métodos aproximados destacamos: a utilização de relaxações em procedimentos de solução e as soluções heurísticas. Os métodos serão descritos detalhadamente a seguir.

2.3.3 Métodos Exatos

Nesta seção serão discutidos dois métodos exatos de solução do problema da mochila: *branch and bound* e *programação dinâmica*.

2.3.3.1 Branch and Bound (BB)

O método Branch and Bound segundo Arenales (2007) baseia-se na idéia de reduzir o espaço de busca. Ele utiliza informações do problema para executar o procedimento de *enumeração implícita* dos pontos candidatos à solução ótima inteira. Para isso é adotada a estratégia de *dividir para conquistar*, que separa o problema original em problemas menores, que são mais fáceis de encontrar uma solução, daí o motivo do nome *branch*. O termo *bound* busca ressaltar que a verificação da otimalidade da solução, tem como base o limite calculado em cada enumeração (limites inferiores), ou seja, em cada subconjunto de problemas, compondo a solução final aqueles que possuem uma solução ótima e descartando os demais.

Matematicamente, podemos defini-lo da seguinte maneira:

$$(P) = \text{Maximizar}\{cx \mid Ax = b, x \geq 0, x \in Z^+\}$$

$$(P') = \text{Maximizar}\{cx \mid Ax = b, x \geq 0, x \in R^+\}$$

Definindo $V^*(P)$ e $V^*(P')$, como os valores da função objetivo nos pontos ótimos P e P' , têm-se:

$$V^*(P) \leq V^*(P')$$

Se x é uma solução ótima de (P) então:

$$V(x) \leq V^*(P) \implies C^T x \leq V^*(P)$$

O problema (P) pode ser dividido em dois problemas (P1) e (P2), e a união das soluções destes formarão a solução de (P). Essa estratégia poderia ser repetida para cada sub-problema, dando origem à árvore de partição do BB, tornando os sub-problemas criados, posicionados nas extremidades da árvore, cada vez mais fáceis de resolver.

A restrição em se utilizar o referido método é que para problemas NP-completos com quantidade de entradas muito grandes, é provocada uma explosão combinatória nas divisões da árvore de partição, tornando-o computacionalmente lento ou na maioria dos casos inviável.

2.3.3.2 Programação Dinâmica (PD)

Arenales (2007) defini que, programação dinâmica é uma metodologia de solução exata e recursiva para problemas de otimização que podem ser decompostos em partes ou estágios. O processo consiste em se tomar uma decisão de se atingir um ponto em questão (novo estágio) com base na decisão do estágio anterior. Sendo assim, o processo é decomposto em vários estágios, que simplificam a solução do problema. Ele se torna determinístico se o resultado da decisão em cada estágio for exato.

Para problemas NP-completos ou NP-difíceis a metodologia de programação dinâmica pode se tornar complexa.

2.3.4 Métodos Aproximados

Nesta seção também serão discutidos dois métodos de solução de problemas de programação linear inteira, porém agora aproximativos: Relaxações e Heurísticas.

2.3.4.1 Relaxações

O emprego de relaxações é comum quando os métodos exatos, como visto anteriormente, são computacionalmente ineficientes e custosos. Este método é uma alternativa às soluções exatas, pois permite buscar um conjunto de soluções para o problema próximas das soluções ótimas, segundo Arenales (2007). Uma vez que são determinados os limites inferiores ou superiores que a função objetivo pode assumir, respectivamente para os casos de minimização ou maximização.

O principal destaque para o método das relaxações é que o mesmo permite ao analista garantir a relação entre a qualidade da solução e o tempo computacional.

Na maioria dos casos é possível encontrarmos soluções ótimas para problemas NP-difíceis ou NP-completo, a um baixo tempo computacional.

Dado um problema de programação, matematicamente poderíamos representar o problema de relaxação como:

$$z = \max\{f(x) : x \in X \subseteq Z^n\}$$

onde $f(x)$ é a função objetivo, x a variável de decisão pertencentes ao domínio de X e Z^n o espaço de soluções.

Logicamente z será uma solução ótima se estiver entre os limites superior $\bar{Z} \geq z$ e inferior $\underline{Z} \leq z$.

Podemos também de forma iterativa chegar a uma solução aproximada Z_t , utilizando como critério de parada a expressão:

$$|\bar{Z} - \underline{Z}| \leq \epsilon$$

ϵ é uma constante não negativa e pequena.

2.3.4.2 Relaxação Lagrangeana

Muitos problemas possuem uma formulação "simples". Os mesmos tornam-se difíceis com a adição de restrições "complexas". No entanto, podemos transformá-los em problemas fáceis, se retiramos tais restrições e introduzimos na função objetivo, multiplicada por um fator de penalidade. A referida técnica é conhecida como relaxação Lagrangeana e

o fator como multiplicador de Lagrange, muito utilizada devido aos resultados satisfatórios obtidos em diversos experimentos.

É importante que se dê atenção para a escolha dos multiplicadores de Lagrange, devido a sua influência na qualidade dos valores limitantes da função objetivo. Experiências com a adoção desta técnica mostram a obtenção de bons limitantes a um custo computacional baixo.

2.3.4.3 Heurísticas

Os algoritmos heurísticos são a outra vertente dos algoritmos aproximativos. Este se baseia na intuição e nas circunstâncias a fim de gerar conhecimento. A palavra heurística é de origem grega *heuristikein*, que significa "descobrir".

Segundo Goldberg e Luna (2000): *"Uma heurística é uma técnica que busca alcançar uma boa solução, utilizando um esforço computacional considerado razoável, sendo capaz de garantir a viabilidade ou otimalidade da solução encontrada, ou ainda, em muitos casos, ambas, especialmente nas ocasiões em que essa busca parte de uma solução viável próxima ao ótimo"*.

Sendo assim, é possível verificar que uma heurística não garante que o algoritmo seja computacionalmente viável para todas as instâncias do problema.

Podemos utilizar relaxações juntamente com heurísticas, onde tal abordagem permite encontrar soluções ótimas e sub-ótimas. Neste trabalho, é proposto um algoritmo de resolução que envolve o uso de uma heurística denominada algoritmos genéticos, que é abordada no capítulo a seguir.

2.4 Algoritmos Genéticos

A heurística dos Algoritmos Genéticos Goldberg (1989) é baseada no princípio da evolução das espécies, proposta por Darwin (1859) e na genética de Mendel (1863).

Os princípios nos quais os algoritmos genéticos se baseiam são simples. Segundo a teoria de Darwin, o princípio de seleção privilegia os indivíduos mais fortes e com maior longevidade, ou seja, que possuem a maior probabilidade de sobreviver e reproduzir, perpetuando seus códigos genéticos, as próximas gerações. Estes códigos genéticos, também conhecidos como cromossomos formam a identidade de cada indivíduo.

2.4.1 Genética Biológica

Todos os seres vivos existentes na natureza são formados por células. Tais células contêm cadeias de DNA, também conhecidas como cromossomos, que possuem um conjunto de "genes", que caracterizam e diferenciam os genótipos e fenótipos de cada um dos seres vivos.

Nos seres de reprodução sexuada, como os mamíferos, os cromossomos são dispostos em pares, cada qual originado de um dos respectivos pais. Durante a reprodução os cromossomos se dividem, dando origem aos gametas. Logo estes gametas possuem metade dos cromossomos do ser original. Na seqüência estes se unem ao gameta do outro ser, dando origem a um novo, cujos cromossomos possuem o formato semelhante ao de seus pais.

Este princípio tem uma enorme influência na diversidade de seres gerados por esta forma de reprodução.

A reprodução pode passar ainda por algumas operações como a recombinação, ou crossover, que ocorre durante a produção dos gametas, onde parte das cadeias de um par é trocado entre estes, antes da separação. Tal fato faz com que o cromossomo de um determinado gameta passe a ser a combinação de cromossomos dos pais, aumentando ainda mais a diversidade dos seres.



Figura 5: Representação do Cruzamento (*crossover*)

Outra operação que pode ocorrer durante este processo é a mutação, que são alterações em determinadas posições dos cromossomos, causadas por fenômenos químicos, físicos ou biológicos, que também contribuem na diversidade dos seres.



Figura 6: Representação da mutação

Todos estes mecanismos são importantes para que os seres possam se adaptar e sobreviver no meio em que vivem. Segundo o fenótipo, os seres podem estar menos ou mais

adaptados ao ambiente, onde os mais adaptados possuem chances maiores de sobreviver e reproduzir, dando origem a gerações, que carregam em seus genes características dos pais provavelmente melhores, e as tornam tão bem adaptados ao ambiente quanto estes.

A escolha dos indivíduos mais bem adaptados ao ambiente segundo o seu fenótipo é conhecido como seleção natural, que, juntamente com as operações que garantem a variabilidade genética, são responsáveis pela evolução das espécies.

2.4.2 Propriedades dos algoritmos genéticos

Os algoritmos genéticos são fundamentados na genética do seres vivos discutida na seção anterior.

O referido método trabalha com populações de indivíduos, que são uma analogia aos cromossomos, onde cada indivíduo representa uma possível solução do problema, no caso semelhante a um gene.

Com base nisto os algoritmos genéticos buscam de forma aleatória por soluções aproximadas para o problema em questão. Este método normalmente é utilizado em problemas cujo espaço de busca é muito grande e desconhecido. Na construção de algoritmos que buscam a solução ótima de um determinado problema, estes princípios são imitados, através da evolução de populações de soluções representadas por cromossomos artificiais. Nesta abordagem um cromossomo é uma estrutura de dados que representa uma das possíveis soluções para o problema, onde estes são submetidos a um processo de evolução. Os componentes de tal estrutura representam os genes.

Cada indivíduo é avaliado por uma função, conhecida como função de adaptação, que indica a qualidade da solução por ele representada, correspondente à adaptação do indivíduo ao ambiente. A Tabela 1 abaixo mostra a relação entre a Biologia e os Algoritmos Genéticos.

Tabela 1: Analogia entre a biologia e algoritmos genéticos

Biologia x Algoritmos Genéticos	
Biologia	Algoritmos Genéticos
Genótipo/cromossomo	Estrutura/Fila
Gene	Componente da estrutura
Alelos	Valores que um componente pode assumir
Fenótipo/indivíduo	Solução
Adaptação ao ambiente	Avaliação da solução
População	Conjunto de soluções

Podemos considerar os mecanismos de seleção, cruzamento e mutação como operadores que agem sobre uma determinada população. A partir disto podemos descrever o seguinte algoritmo genético básico representado pela Figura 7 à seguir.

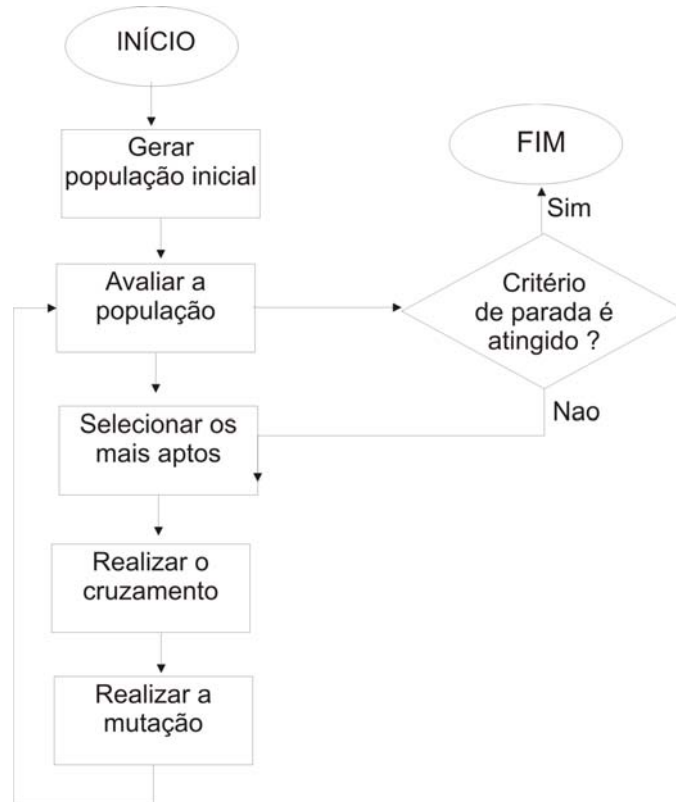


Figura 7: Fluxo de execução do algoritmo genético

A condição de parada normalmente é o número máximo de gerações, mais podemos fazer também com que a execução do mesmo termine quando não houver maiores evoluções, ou seja, quando o mesmo convergir.

2.4.3 Codificação

Consiste na construção da solução real do problema, a partir do cromossomo, para que a mesma seja avaliada. A representação clássica é a binária por facilitar a transformação em números inteiros ou reais, na qual a estrutura do cromossomo consiste em um vetor de bits de tamanho fixo.

Se a solução for composta, por exemplo, por dois números inteiros de 0 a 15, podemos representá-la em um vetor de 8 elementos, onde cada grupo de 4 elementos representa ao valor de uma variável. Sendo assim a codificação binária da solução (11,4) seria representada por

10110100

O problema de tal representação porém, é que a mesma é limitada em relação à capacidade de representação e quantidade de operadores que atuarão sobre o cromossomo codificado. Nestes casos devemos utilizar outra forma de representação como a por valor e a por permutação.

Na codificação por valor os componentes da solução são representados da maneira original, ou seja, por números inteiros ou reais. Este tipo de representação é útil quando só é necessário representar a informação desejada por si só, sem ser necessário dividi-la em partes menores.

Já no caso da codificação por permutação a solução é representada por uma permutação de valores de um determinado conjunto. Tal representação esta ligada o problema que se deseja representar, como no caso de problemas de caminhos em grafos, onde os valores da permutação correspondem à ordem em que os vértices do grafo são percorridos. Na Tabela 2 abaixo relacionamos os problemas e as suas possíveis formas de representação.

Tabela 2: Relacionamento de problemas x forma de representação

Problemas x Representação	
Representação	Problemas
Binária	Numéricos/Inteiros
Números reais	Numéricos
Permutação de símbolos	Baseados em ordem
Símbolos repetidos	Grupamento

A representação que será utilizada neste trabalho para o cromossomo, será a dos valores reais, por se tratar de um problema de codificação simples e fácil de manipulá-lo através dos operadores genéticos.

2.4.4 Operadores Genéticos

Nesta seção, serão apresentados os operadores genéticos mais utilizados, inspirados na genética biológica vista anteriormente.

2.4.4.1 Seleção

O processo de seleção consiste em escolher segundo determinado critério, um ou mais indivíduos de uma população, que passarão a compor uma nova população. O critério de seleção em algoritmos genéticos é baseado na aptidão dos indivíduos, sobre uma função de avaliação, onde os mais aptos possuem a maior probabilidade de serem escolhidos, do que os menos aptos. Sendo assim, a seleção corresponde, à sobrevivência e reprodução dos mais adaptados ao ambiente. Existem diversos mecanismos para determinar o indivíduo a ser sorteado como os métodos aleatório, torneio e roleta, onde o mais utilizado é o Método da Roleta, em que cada indivíduo é representado por uma fatia, proporcional a sua aptidão, como pode ser visto pela Figura 8.

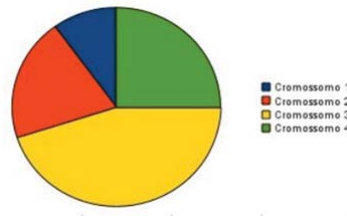


Figura 8: Ilustração do método da roleta

Dessa forma a probabilidade de um indivíduo i ser selecionado é dada por

$$p_i = \frac{f_i}{\sum_{j=1}^n f_j} \quad (2.3)$$

onde f_i é a função de avaliação do indivíduo i na população corrente e n o número de indivíduos na população. É importante observar que para que o referido método funcione f_j dever ser positivo, para $j = 1, \dots, n$.

Uma vez sorteado o indivíduo é adicionado a nova população até que essa atinja o número de indivíduos desejados. É possível observar que pelo referido método um indivíduo pode ser sorteado mais de uma vez enquanto outro pode ser que nunca venha a ser sorteado. Sendo assim optou-se por utilizar neste trabalho o referido método.

2.4.4.2 Cruzamento

Uma vez selecionados, os indivíduos são recombinados através do operador de Cruzamento (*crossover*), também conhecido como, Recombinação, tal operador representa uma característica fundamental dos algoritmos genéticos.

Basicamente é criado um novo indivíduo a partir da troca de material genético entre pares de indivíduos selecionados, segundo suas aptidões. Tal indivíduo é diferente de seus pais, mas possui características genéticas de ambos. Dessa forma espera-se que sejam combinadas as principais informações do problema em questão, de modo a obter uma nova geração indivíduos que representem soluções melhores, até então encontradas.

A operação de cruzamento normalmente ocorre entre dois indivíduos, gerando outros dois, que formarão a nova população. Na representação binária dos cromossomos, é escolhido um ponto de corte, ou seja, ponto a partir do qual as informações dos cromossomos pais serão trocadas. Como na representação abaixo, para dois cromossomos, segundo um corte aleatório

10110100

e

11000101,

escolhendo o terceiro bit como ponto de corte, surgiram os cromossomos:

101—00101

e

110—10100.

O ponto de corte divide os dois genitores, criando outros dois possíveis descendentes.

Nas demais formas de representação do cromossomo, o operador de cruzamento utilizando ponto de corte, em uma simples troca pode gerar soluções inválidas para o problema. Nesses casos, é necessário criar operadores de cruzamento específicos, de modo a garantir que os cromossomos formados pertençam ao domínio da solução do problema.

Geralmente é atribuída uma probabilidade para aplicação do cruzamento, de forma que indivíduos possam ser escolhidos para tal operação, e a mesma não ser aplicada sobre eles, mantendo-os inalterados. As probabilidades de se aplicar o operador de cruzamento normalmente estão na faixa de 60% a 90%.

2.4.4.3 Mutação

O operador de mutação, assim como na biologia, uma vez aplicado altera os elementos de um cromossomo. Espera-se com tal método obter uma variabilidade genética, gerando populações de indivíduos que, possivelmente, seria impossível de se obter utilizando apenas o operador de cruzamento.

O operador de mutação efetua a troca do conteúdo de uma posição do cromossomo, por outro valor que o mesmo possa assumir. A troca é efetuada segundo uma determinada probabilidade, que no geral é baixa ($\leq 1\%$). Na codificação binária, se um bit em uma determinada posição precisa sofrer mutação, basta invertê-lo, ou seja, trocar de 1 para 0 ou de 0 para 1, conforme ilustrado abaixo

11010110.

aplicando a mutação no terceiro e sétimo bit obtem-se

11110100.

O problema, porém assim como no cruzamento é, se utilizarmos outras formas de codificação, a mutação, em uma simples troca, pode gerar indivíduos cuja solução é inválida, ou ainda, gerar indivíduos cuja solução é válida, mais muito pior que até então obtida. Novamente nesses casos é necessário criar operadores de mutação específicos para o problema.

2.4.4.4 Elitismo

Durante a utilização dos operadores citados à cima, pode ser interessante ou necessário, guardar os indivíduos da geração anterior. Tal procedimento é conhecido como elitismo. Nos operadores de seleção o elitismo consiste em copiar os melhores indivíduos da população anterior para a nova população, sem passar pelo sorteio. Nos demais operadores estes indivíduos são ignorados, logo não são modificados.

2.4.4.5 Criacionismo

Através do criacionismo, é possível ainda, incluirmos uma nova população durante a utilização do operador de seleção. São incluídos indivíduos gerados de forma aleatória, da mesma maneira que é feita na primeira geração.

2.4.4.6 Inicialização da População

A inicialização da população impõe a criação dos indivíduos que serão utilizados na primeira iteração do algoritmo. Tal população geralmente é composta por indivíduos criados aleatoriamente, porém quando no início já se conhece bons valores de entrada, a utilização dos mesmos torna o processo evolutivo mais eficiente.

2.4.5 Algoritmos genéticos e problemas de transporte

Cada vez mais os algoritmos genéticos vêm despertam o interesse de pesquisadores na área de problemas de transporte. Apesar da modelagem cromossômica em alguns casos exigir um esforço maior do modelador, várias experiências têm mostrado que heurísticas baseadas em algoritmos genéticos possuem um grande potencial de sucesso para problemas complexos, difíceis de serem resolvidos por pacotes comerciais.

A seguir serão apresentados alguns trabalhos na área de transportes que obtiveram boas soluções através de heurísticas baseadas em algoritmos genéticos.

2.4.5.1 Algoritmo genético para a expedição de veículos por Leclerc; Potvin (1997)

Conforme apontado por Galvão(2004) o trabalho desenvolvido por Leclerc; Potvin (1997) se passa em uma empresa de entregas expressas, que necessitam despachar veículos para a realização do mesmo. O objetivo deste é o de apoiar o operador na tomada de decisões, quando são inseridas novas coletas para aquelas já pré-estabelecidas. A modelagem é feita considerando o problema como de multi-atributos e o algoritmo genético é utilizado em substituição ao operador no processo de tomada de decisão. Para cada veículo é criado um vetor de atributos, onde os mesmos são avaliados de forma que 1 represente o melhor, e 0 o pior. A função de aptidão é dada pela somatória dos atributos de cada veículo. Os veículos são ordenados conforme o valor obtido pela função de aptidão.

Na codificação os cromossomos são representados pelos respectivos valores obtidos na função de aptidão normalizados, sem a necessidade da representação binária.

Os resultados encontrados mostram como os algoritmos genéticos podem beneficiar problemas de tomada de decisão, envolvendo multi-atributos.

A heurística abordada por Leclerc; Potvin (1997) se assemelha ao problema de coleta do palhiço, no que diz respeito à soma e ordenação dos valores da função de aptidão, aqui

representados pela distância percorrida, e ao considerar o valor real do alelo, ao invés da representação binária clássica.

2.4.5.2 Algoritmos genéticos aplicados ao planejamento do transporte principal de madeira proposto por Souza (2004)

O estudo desenvolvido por Souza (2004) aborda a colheita e transporte de madeira e a sua importância no custo final do produto, com o objetivo de ajudar na tomada de decisão quanto ao período de intervenção das equipes no corte e o fluxo de madeira entre os pontos de produção e as fábricas, minimizando tais custos.

Na codificação, cada cromossomo é representado por um vetor que indica a sequência em que os pontos de produção são avaliados, sendo seu tamanho definido pelo número de pontos disponíveis para a colheita.

A função de aptidão avalia a somatória dos custos com a colheita, com o transporte e o tempo total para realizar tal trabalho.

Foram criados oito variedades de algoritmos genéticos em função da combinação de três estratégias de melhoria da solução, sendo as 2 primeiras estratégias visando respeitar a alocação inicial das equipes nos pontos de corte e a terceira permitir que duas equipes realizem a colheita de madeira em um mesmo ponto de produção por período.

Os resultados obtidos com a heurística baseada em algoritmos genéticos foram comparados aos da solução ótima obtida por programação linear, no intuito de verificar o quanto tal abordagem se aproxima da solução exata do problema.

As variedades de algoritmos implementadas com as estratégias três apresentaram custo totais de transportes menores que as demais variedades, logo tal estratégia proporciona maiores ganhos reais. As melhores soluções apresentaram custo entre 2% e 5,2% a mais que a solução exata.

O referido estudo mostrou o quanto heurísticas baseadas em algoritmos genéticos, para problemas de transportes que envolvem tomada de decisões, possuem resultados próximos aos obtidos pelos métodos exatos, servindo assim como uma excelente alternativa para a solução de problemas complexos e de múltiplas variáveis, muitas vezes ineficientes ou até mesmo inviáveis utilizando tais métodos.

Capítulo 3

O Problema de Coleta do Palhiço de Cana-de-açúcar

Nesta capítulo será apresentado o problema geral de transporte do palhiço de cana-de-açúcar, bem como o seu modelo matemático.

3.1 Descrição do problema

Como apresentado no capítulo 2 para que o sistema de coleta e recepção do palhiço de cana-de-açúcar possa trabalhar com o fluxo ideal de transporte existe a influência de diversos fatores, dentre eles podemos destacar a distância dos talhões à usina e o respectivo percurso dos caminhões neste ciclo, problema este que será o foco da pesquisa que está sendo proposta. A Figura 9 ilustra o ambiente no qual ocorre o problema.

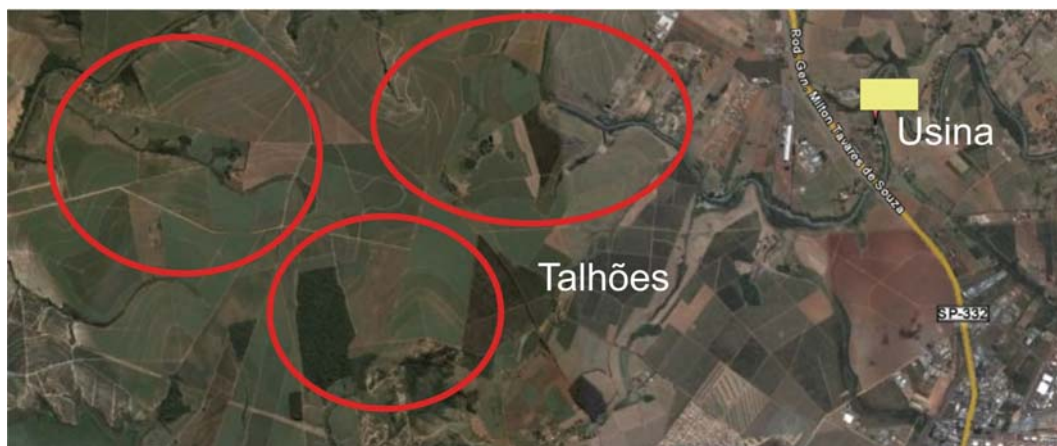


Figura 9: Exemplo de ambiente composto pela usina e os talhões

No que diz respeito ao transporte do palhiço de cana-da-açúcar, a grande questão que o setor sucroalcooleiro busca esclarecer é como retirar o palhiço do campo e disponibilizá-lo para a usina, sem elevar seus gastos. O referido custo vai desde o gasto com combustível, até a ociosidade dos caminhões, pois o mesmo caminhão utilizado para o transporte da cana crua passou a ser utilizado para o transporte do palhiço.

A coleta do palhiço pode ser representada pelo esquema da Figura 10.

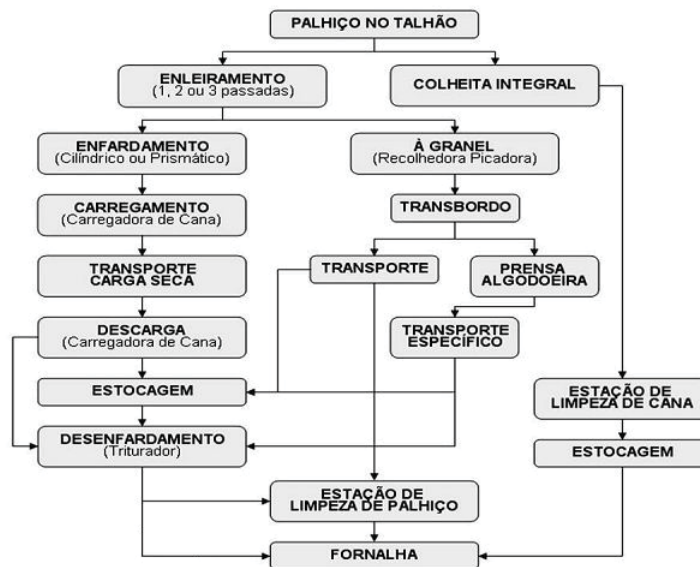


Figura 10: Fluxograma da colheita e recepção do palhiço da cana-de-açúcar
 FONTE: Ripoli & Ripoli (2004).

Diversas usinas do Estado de São Paulo, na busca por tal redução, têm efetuado diferentes ensaios, quanto ao processo que seria mais economicamente viável. Logo, por não existir uma metodologia padrão, torna-se impraticável comparar os resultados obtidos em tais ensaios, pois cada unidade adota padrões diferentes. No entanto, como se trata de uma nova tecnologia de manipulação de biomassa, muitos assuntos como este, são plausíveis para a realização de pesquisas.

3.2 Definição do problema

De maneira geral podemos definir o problema como o de coletar o palhiço a um custo mínimo, que se refere à menor distância percorrida pelos caminhões, de tal forma que toda a demanda seja atingida e a capacidade dos caminhões seja respeitada.

O problema descrito representa uma situação real podendo ser descrito da seguinte maneira: dada uma usina, que possui um conjunto de talhões de diferentes tamanhos, que geram diferentes quantidades de fardos de palhiço, deve-se dimensionar uma frota heterogênea mínima de veículos, assim como elaborar a respectiva programação das viagens de cada um dos veículos. Os veículos farão viagens fechadas de ida e volta entre a usina e os talhões, para coletar os fardos de palhiço correspondendo à sua capacidade máxima. Cada uma das viagens realizadas terá origem na usina, onde o veículo estará vazio, com destino a um ou mais talhões, a fim de coletar os fardos com capacidade máxima de uma vez e retornar à usina. Não existem restrições quanto a horário de carregamento e descarregamento, nem quanto à seqüência de coleta, e sim apenas que o veículo respeite a jornada de trabalho dos veículos.

É pouco provável que ocorra a formação de filas durante a descarga do produto, pois no geral a mesma é rápida, o número de veículos destinados ao recolhimento do palhiço é reduzido.

No presente trabalho, serão conhecidas as características dos talhões e da frota, tais como o número de fardos por talhão, a área de cada um deles, a sua distância até a usina, bem como a distância entre eles e a capacidade dos veículos.

O planejamento do sistema em questão consiste na decisão de como alocar os recursos disponíveis para a coleta da biomassa de cana-de-açúcar, a uma distância menor possível, levando conseqüentemente a uma redução dos gastos com combustível e conservação dos veículos.

3.3 Modelo matemático

Com base em sua definição podemos escrever o problema matematicamente como a seguir.

Seja $T = 1, 2, 3, \dots, n$ o conjunto de talhões da usina de cana-de-açúcar, $v = 1, 2, 3, \dots, V$ a frota heterogênea de veículos disponíveis para coleta da biomassa de cana-de-açúcar e as seguintes variáveis:

$i = 0$ o depóstio, usina;

c_{ij} a distância entre dois pontos i e j de coleta ou recepção com $i, j = 0, 1, 2, \dots, n$;

d_j o número de fardos por talhão com $j = 1, 2, \dots, n$;

y_i^v o número de fardos do talhão i transportados pelo veículo v com $i = 1, 2, \dots, n$ $v = 1, 2, \dots, V$;

$$x_{ij}^v = \begin{cases} 1, & \text{se o talhão } j \text{ é visitado pelo caminhão } v \text{ após ter visitado o talhão } i \\ 0, & \text{caso contrário} \end{cases}$$

O referido problema de dimensionamento e designação de veículo pode ser formulado como:

$$\min \sum_{i=0}^T \sum_{j=0}^T \sum_{v=1}^V c_{ij} x_{ij}^v \quad (3.1)$$

Sujeito a:

$$\sum_{v=1}^V y_i^v = d_i \quad i = 1, 2, \dots, T \quad (3.2)$$

$$\sum_{i=1}^T y_i^v \leq C_v \quad i = 1, 2, \dots, T \quad (3.3)$$

$$y_j^v \leq \sum_{i=0}^T d_j x_{ij}^v; \quad j = 1, 2, \dots, T \quad v = 1, 2, \dots, V \quad (3.4)$$

$$\sum_{i=1}^T x_{ip}^v - \sum_{j=1}^T x_{pj}^v = 0 \quad p = 0, 1, \dots, T \quad (3.5)$$

$$y_j^v \in N \quad j = 0, 1, \dots, T \quad v = 0, 1, \dots, V \quad (3.6)$$

$$x_{ij}^v \in \{0, 1\} \quad i = 0, 1, \dots, T \quad j = 0, 1, \dots, T \quad v = 0, 1, \dots, V \quad (3.7)$$

A função objetivo (4.1) corresponde a minimização da distância percorrida pela frota. A restrição (4.2) garante que toda a demanda do talhão i deve ser carregada pelos caminhões. A restrição (4.3) indica que a carga do caminhão v não deve exceder a sua capacidade. A restrição (4.4) indica a dependência entre as variáveis y e x . A restrição (4.5) assegura a manutenção do fluxo entre os talhões, onde o caminhão que chega, deve sair do talhão. Finalmente as restrições (4.6) e (4.7) asseguram a integridade das variáveis de decisão.

No capítulo a seguir será discutida a metodologia adotada para solucionar este problema que é NP-completo.

Metodologia

Segundo discutido no capítulo anterior os problemas de programação linear inteira são de difícil resolução, pois não existe um algoritmo geral capaz de resolver e obter a sua solução ótima de maneira eficiente.

Os algoritmos exatos ainda que tenham evoluído, em muitos casos ainda são ineficientes computacionalmente, por se tratarem de técnicas de busca exaustiva que geram um número muito alto de soluções para garantir a obtenção da solução ótima.

Devido às diversas restrições que os pacotes comerciais apresentam para resolver problemas de designação de veículos com múltiplas variáveis. Optou-se por dividir a solução do problema em duas partes:

1. Inicialmente são criadas rotas para constituir a população inicial. Aplica-se então algoritmos genéticos por um número fixo de vezes.
2. Na seqüência, são escolhidas dentre as soluções aquelas que forneçam um menor custo sem desrespeitar as restrições do problema e solucionados pelo solver comercial *Xpress-Optimizer*, de resolução de problemas de programação linear, desenvolvido pela Dash Optimizer em sua versão acadêmica. A Figura 11 exhibe o esquema geral da metodologia adotada para a resolução do problema.

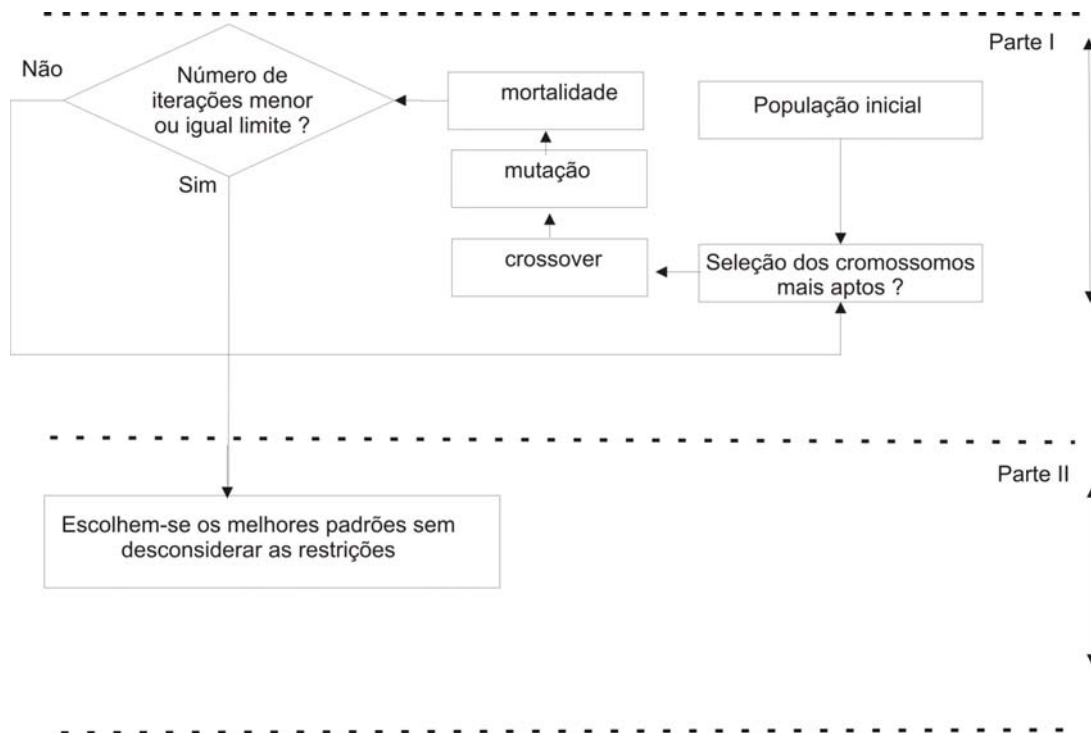


Figura 11: Esquema geral da solução

4.1 A estratégia de determinação das rotas e a construção do algoritmo genético

A determinação das melhores soluções será dada através de um algoritmo genético, que será aplicado ao arquivo de entrada no intuito de gerar o melhor conjunto, ou seja as melhores rotas de cada veículo.

Para isso, é proposta a seguinte estratégia, representada através do modelo matemático a seguir, e que é a base para a projeção do algoritmo genético.

Seja $v = 1, 2, \dots, V$ os veículos disponíveis para a coleta do palhiço, e $i = 1, 2, \dots, T$ a quantidade de talhões existentes na usina, e ainda y_i^v a quantidade de fardos do talhão i transportados pelo veículo v .

Definimos o seguinte vetor $P_j^v = \begin{bmatrix} y_1^v \\ \vdots \\ y_n^v \end{bmatrix}$, que representa a carga total transportada de palhiço por cada veículo v .

Logo, podemos tirar do modelo que, se $y_i^v \geq 0$ significa que o veículo visita o talhão i onde recolhe a carga y_i^v , a partir deste vetor é possível identificar quais talhões foram

visitados por um veículo v .

Portanto a estratégia proposta é a de que para cada talhão visitado pelo veículo v calculamos o roteiro de distância mínima, ou seja:

Dado o conjunto de vetores $P_j^v = \begin{bmatrix} y_1^v \\ \vdots \\ y_n^v \end{bmatrix}$ com $v = 1, 2, \dots, V$ que chamaremos de padrão.

Para cada k_v conjunto de padrões do veículo v , composto pelos padrões de entrada p_{jv} , sendo v o veículo e j o padrão, é criado um conjunto solução n com os p_{jv} padrões em que v percorrer a menor distância total, ou seja, as n melhores rotas de v para o padrão j . O mesmo é aplicado sucessivamente para todos os conjuntos de padrões.

De início será considerada apenas a restrição de capacidade dos veículos, as demais restrições serão tratadas pela ferramenta *Xpress-Optimizer* em uma segunda instância do problema.

4.1.1 Descrição funcional do algoritmo genético

Como descrito na seção anterior, é proposta a utilização de algoritmos genéticos para a resolução do problema de designação dos veículos. Sendo assim, a seguir é descrita a utilização de tal estratégia.

4.1.1.1 População inicial

No presente trabalho optou-se por criar um arquivo contendo todas as informações da população inicial de um problema teste, o mesmo poderia ser gerado aleatoriamente pelo algoritmo genético.

A formulação a seguir demonstra um exemplo da composição da população inicial do problema, que facilita o seu entendimento.

$$\underbrace{\begin{bmatrix} 20 \\ \vdots \\ 30 \\ \vdots \end{bmatrix}}_{p_{11}} \underbrace{\begin{bmatrix} \vdots \\ 50 \\ \vdots \\ \vdots \end{bmatrix}}_{p_{21}} \underbrace{\begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ y_i^v \end{bmatrix}}_{p_{jv}} \dots \underbrace{\begin{bmatrix} 100 \\ \vdots \\ \vdots \\ \vdots \end{bmatrix}}_{k_2} \dots \underbrace{\begin{bmatrix} 10 \\ \vdots \\ 5 \\ \vdots \end{bmatrix}}_{k_v} \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ 35 \\ \vdots \end{bmatrix} \begin{bmatrix} 65 \\ \vdots \\ \vdots \\ \vdots \end{bmatrix}$$

Com p_{jv} igual ao padrão j do veículo v , k_v o conjunto de padrões do veículo v e y_i^v a carga recolhida no talhão i pelo veículo v .

4.1.1.2 Representação do indivíduo

Cada indivíduo de uma população inicial, representa uma rota diferente, formada pela sequência de viagens de um veículo e a quantidade de fardos por este carregado em um determinado talhão, como descrito anteriormente, da qual chamamos de padrão. Este por sua vez representa um cromossomo do algoritmo genético, onde ao contrário da abordagem clássica, que utiliza notação binária, o gene neste caso é representado pelo valor real no formato inteiro de cada peso.

A Figura 12 a seguir ilustra a representação de um cromossomo do problema.

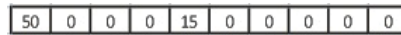


Figura 12: Representação do cromossomo

A partir disto é possível determinar o percurso percorrido por cada veículo, bem como os custos despendidos neste, da usina aos talhões, pois o cromossomo indica em quais talhões o veículo carregou.

Vale salientar que o cromossomo de cada padrão terá sempre entre 1 ou 2 pesos não nulos somente, pois no ambiente real de coleta do palhiço, levando-se em consideração o menor percurso, capacidade do veículo e demanda do talhão, o veículo sempre fara uma viagem a um talhão próximo, onde é carregado até seu volume máximo e retorna a usina. Apenas em algumas ocasiões em que a demanda do talhão for atingida é que o veículo ira completar sua carga no talhão seguinte mais próximo.

4.1.1.3 Avaliação da população

Segundo a metodologia proposta no presente trabalho a função de aptidão F baseia-se na distância total percorrida pelos veículos durante um dia todo de trabalho, ou seja, ao cálculo da função objetivo. Por simplificação, a distância aqui considerada, será a distância euclidiana.

$$F = \sum_{i=0}^T \sum_{j=0}^T c_{ij} x_{ij} \quad (4.1)$$

A Tabela 3 a seguir demonstra como é efetuado o cálculo da função de aptidão e a formação do cromossomo. Cada gene é composto por um único alelo, que representa a quantidade de fardos carregados em um determinado talhão. A partir disto é possível descobrir o trajeto do veículo, bem como a distância deste. Neste exemplo o valor da função de aptidão 20km, corresponde ao menor percurso do padrão i , neste caso apenas de exemplo desconsidera-se as distâncias percorridas até a origem e destino. Este é valor que pretende-se descobrir para cada um dos padrões de cada veículo.

Tabela 3: Exemplo do cálculo da aptidão para um único veículo

Cálculo da Função de Aptidão											
Talhão visitado	i										
Número de fardos transportados	Yiv										
Distância entre os pontos i e j	Cij										
	Aptidão	Talhões									
Valor de aptidão/cromossomo	20	1	2	3	4	5	6	7	8	9	10
i - Padrão/Trajeto		Número de fardos transportados									
i - 1/1		10	0	0	0	0	0	5	0	0	0
i - 1/2		0	15	0	0	0	0	0	0	0	0
i - 1/3		0	0	0	9	0	0	0	0	0	6
i - 1/4		0	0	15	0	0	0	0	0	0	0
i - 1/5		0	0	0	0	0	8	0	0	7	0
	Aptidão	Distância entre os pontos i e j									
i - 1/1	50	23	0	0	0	0	0	27	0	0	0
i - 1/2	30	0	30	0	0	0	0	0	0	0	0
i - 1/3	49	0	0	0	18	0	0	0	0	0	31
i - 1/4	20	0	0	20	0	0	0	0	0	0	0
i - 1/5	48	0	0	0	0	0	23	0	0	25	0

4.1.1.4 Critérios de parada

Dentre os critérios de parada existentes, podemos destacar o número máximo de repetições, utilizado neste trabalho. Um limite é definido manualmente para o número de repetições, sendo a execução do algoritmo interrompida quando este valor é atingido. De uma forma geral o número máximo de repetições, refere-se a quantidade de vezes que é aplicada a função de aptidão sem que o resultado da solução apresente alguma melhora significativa.

4.1.1.5 Operador de Seleção

Para a realização da seleção, o algoritmo genético proposto foi implementado utilizando-se o Método da Roleta, apresentado na seção 2.4.4.1, e aplicado segundo os valores obtidos pela função de aptidão definida na seção 4.1.1.3, deste trabalho.

4.1.1.6 Operador de Cruzamento

Nesta etapa, os indivíduos que foram selecionados na etapa anterior são cruzados segundo uma taxa ρ de cruzamento, de forma que: o usuário informa o valor da taxa, que indica a probabilidade de ocorrer o cruzamento.

A Figura 13 ilustra o cruzamento clássico entre os cromossomos do referido problema, segundo um ponto de corte selecionado aleatoriamente.

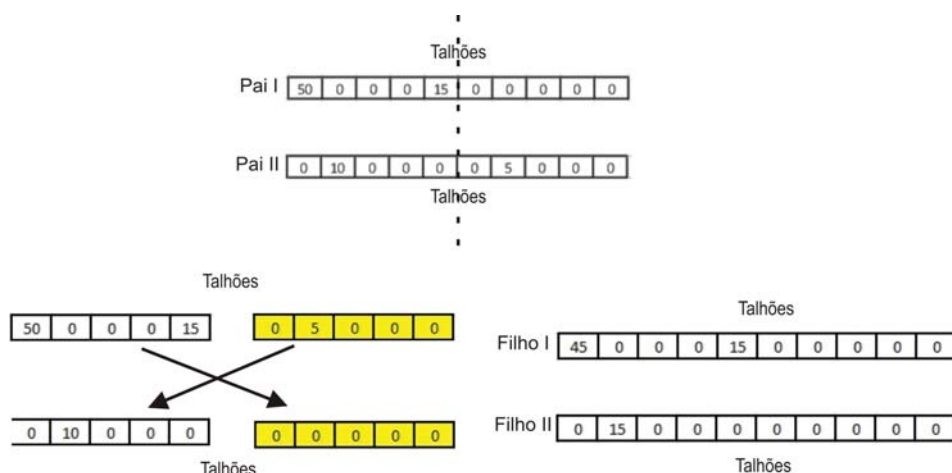


Figura 13: Exemplo de cruzamento clássico entre dois padrões

No entanto para a determinação do ponto de corte é definida uma estratégia segundo a especificação do problema, pelo fato de se utilizar a representação real do cromossomo.

Tal estratégia consiste em determinar o primeiro talhão visitado pelo cromossomo pai e o último visitado pelo cromossomo mãe, sendo o ponto de corte qualquer reta neste intervalo que divide pai e mãe em duas partes. A Figura 14 ilustra o funcionamento da referida estratégia.

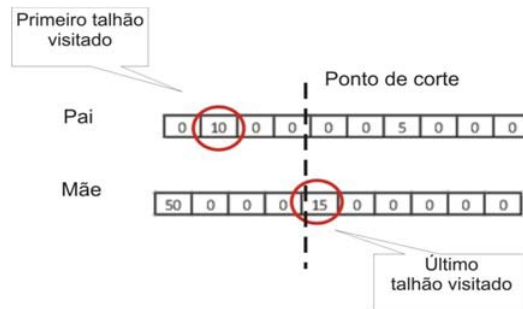


Figura 14: Estratégia de determinação do ponto de corte

Cabe destacar que após o cruzamento, quando necessário, deve-se corrigir as quantidades carregadas a fim de respeitar a capacidade do veículo.

4.1.1.7 Operador de Mutação

O operador de mutação implementado neste trabalho, é o da mutação por substituição. Na representação binária clássica, um bit no qual ocorre a mutação, tem seu valor invertido. O mesmo se aplica a este caso, em que valor é representado por um número real, porém é preciso respeitar as restrições do problema, tais como a de capacidade do veículo.

Assim como no operador de cruzamento, o usuário fornece o valor da taxa de mutação, que indica a probabilidade de ocorrer a troca aleatória do valor de um gene.

A Figura 15 abaixo representa a referida mutação.

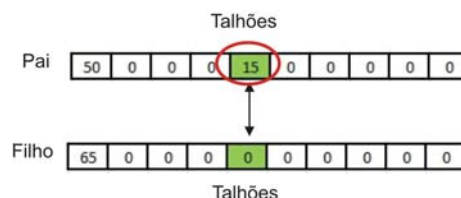


Figura 15: Exemplo de mutação entre dois padrões

A mesma regra que preserva a capacidade dos veículos vista no cruzamento é utilizada para a mutação.

4.1.1.8 Mortalidade

Quando a população atingir np padrões para um veículo elimina-se os menos aptos.

4.2 A seleção dos melhores padrões gerados pelo algoritmo genético

A fim de selecionar os padrões que formam as melhores soluções, é utilizado o software *Xpress-Optimizer* que resolve o problema de:

$$\min \sum_{j=1}^V \sum_{i=1}^{k_j} d_{ij} \lambda_{ij} \quad (4.2)$$

Sujeito a:

$$\lambda_{ij} = \begin{cases} 1, & \text{se o padrão } p_{ij}, \text{ for escolhido} \\ 0, & \text{caso contrário} \end{cases}$$

$$\sum_{j=1}^V \sum_{i=1}^{k_j} d_{ij} \lambda_{ij} \leq D_t \quad j = 1, 2, \dots, V \quad (4.3)$$

D_t igual a demanda do talhão t .

que consiste em escolher um padrão para cada veículo, com a soma das distâncias no menor valor possível, dentre os padrões gerados pelo algoritmo genético, de forma a atender todos os talhões, conforme representado no esquema abaixo:

$$\underbrace{\begin{bmatrix} 20 \\ \vdots \\ 30 \\ \vdots \end{bmatrix}}_{p_{11}} \lambda_{11} \underbrace{\begin{bmatrix} \vdots \\ 50 \\ \vdots \\ \vdots \end{bmatrix}}_{p_{21}} \lambda_{12} \underbrace{\begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ 50 \end{bmatrix}}_{p_{jv}} \lambda_{1j} + \dots + \underbrace{\begin{bmatrix} 10 \\ \vdots \\ 5 \\ \vdots \end{bmatrix}}_{k_v} \lambda_{v1} \underbrace{\begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ 35 \end{bmatrix}}_{k_v} \lambda_{v2} \underbrace{\begin{bmatrix} 65 \\ \vdots \\ \vdots \\ \vdots \end{bmatrix}}_{k_v} \lambda_{vn} \geq \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_t \end{bmatrix}$$

No capítulo a seguir será mostrado como foi realizado a implementação e os testes da metodologia aqui definida.

Capítulo 5

Implementação, Experiências e Resultados

A fim de realizar testes e analisar os resultados, foi implementada uma ferramenta em linguagem C, que utiliza a heurística de algoritmo genético e alguns procedimentos para codificação dos dados e aplicação dos operadores, definidos no capítulo anterior. Além disso, para validar a solução, utilizamos a ferramenta de resolução de problemas de programação linear *Xpress-Optimizer* em sua versão gratuita, destinada à pesquisa.

5.1 Os dados do problema

Para a realização das experiências foram criados cinco arquivos de entrada para o problema, representado cada um deles: a localização dos talhões e da usina no campo; os dados de cada talhão tais como: área, quantidade de fardos, distância da usina; os dados da frota disponível: capacidade, tipo; o arquivo de distância entre os talhões e por fim o arquivo contendo a população inicial do problema, optou-se por utilizar tal mecanismo ao invés da geração de um arquivo de forma aleatória.

5.2 O software

Para a implementação do software proposto, foi utilizado a linguagem C, e o ambiente de desenvolvimento Builder C++ 6.0. O objetivo de utilizar tais ferramentas foi o de fornecer uma interface intuitiva para solução, onde é possível visualizar um plano cartesiano que representa a região do problema, bem como a localização dos talhões e da usina e as rotas da melhor solução encontrada.

A Figura 16 mostra o programa em execução.

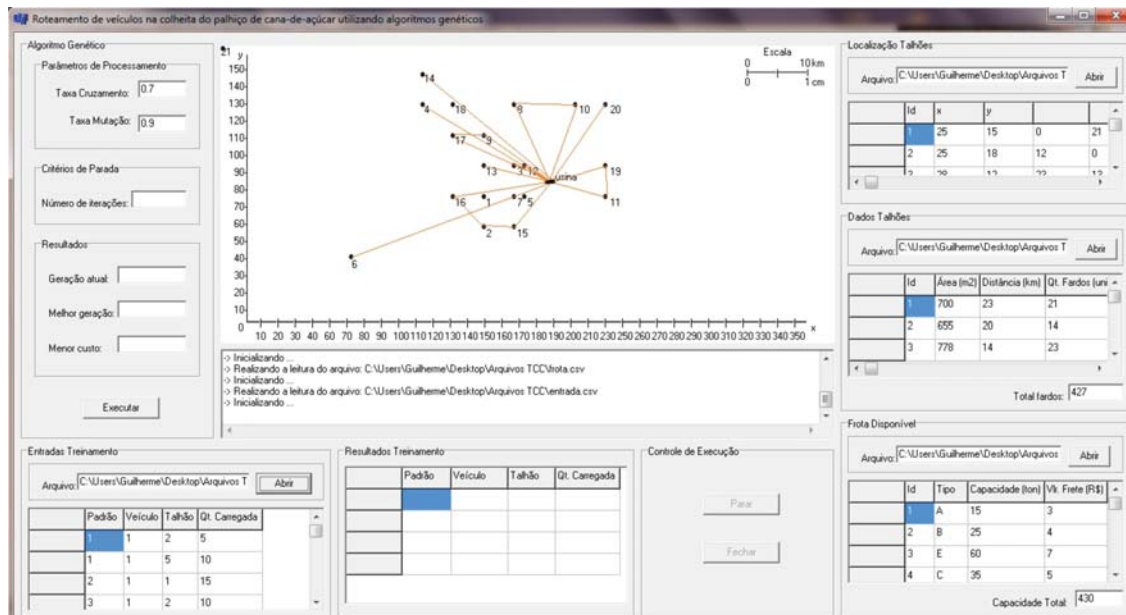


Figura 16: Imagem do software desenvolvido

Na utilização da ferramenta é possível definir o número máximo de iterações, as taxas de cruzamento e mutação. Para realizar um experimento o usuário deve seguir as seguintes etapas:

1. Entrar com o arquivo de localização dos talhões;
2. Entrar com o arquivo de distância entre os talhões;
3. Entrar com o arquivo de dados dos talhões;
4. Entrar com o arquivo de dados da frota disponível;
5. Definir as taxas de cruzamento de mutação do algoritmo genético;
6. Definir o valor do critério de parada;
7. E finalmente clicar no botão Executar para inicializar o algoritmo;

Após a sua execução, o programa gera um arquivo contendo padrões individuais para os veículos. Cada padrão como já citado, consiste dos talhões a serem visitados pelo veículo respeitando a sua capacidade de carga.

Vale salientar que, na geração de padrões, da mesma forma que existe o botão Executar para inicializar o algoritmo, existem também os botões Abortar e Fechar, onde o primeiro

deve ser utilizado para interromper a execução do algoritmo quando este for necessário, devido por exemplo: a um valor de entrada indevido, ou ainda, a divergência da solução, e o último botão por sua vez simplesmente fecha a instância atual do programa.

A parte principal do sistema como foi exibido, é a implementação de algoritmos genéticos para resolver o problema de roteamento dos veículos de coleta do palhiço.

O Apêndice A contém partes do programa implementado e dos operadores genéticos para ilustração.

5.3 A ferramenta Xpress-Optimizer

O pacote comercial *Xpress-Optimizer* foi criado pela empresa Dash Optimizer, com o objetivo de auxiliar empresas e demais instituições, a solucionarem problemas de otimização. O referido programa possui vários módulos como o *Xpress-BCL*, *Xpress-IVE*, *Xpress-Mosel* entre outros. Neste trabalho foram utilizados os módulos IVE e Mosel em ambiente Windows.

O módulo IVE encontra-se disponível apenas para plataforma Windows, sendo a sua utilização bastante intuitiva. O Mosel, é uma linguagem e ambiente de modelagem de problemas de programação linear, possui uma estrutura semelhante ao Pascal que facilita a manipulação e criação de modelos de problemas. Os primeiros comandos são o "model" para definirmos o nome para o modelo e uses para incluirmos os módulos necessários. Os comentários devem ser precedidos pelo caractere '!'.
!'

O próximo bloco é o de declarações das variáveis, que deve começar com **declarations** e terminar com **end-declarations**. Depois é necessário modelar todas as restrições e equações e informar o procedimento a ser tomado, maximizar ou minimizar.

A Figura 17 mostra a interface do programa.

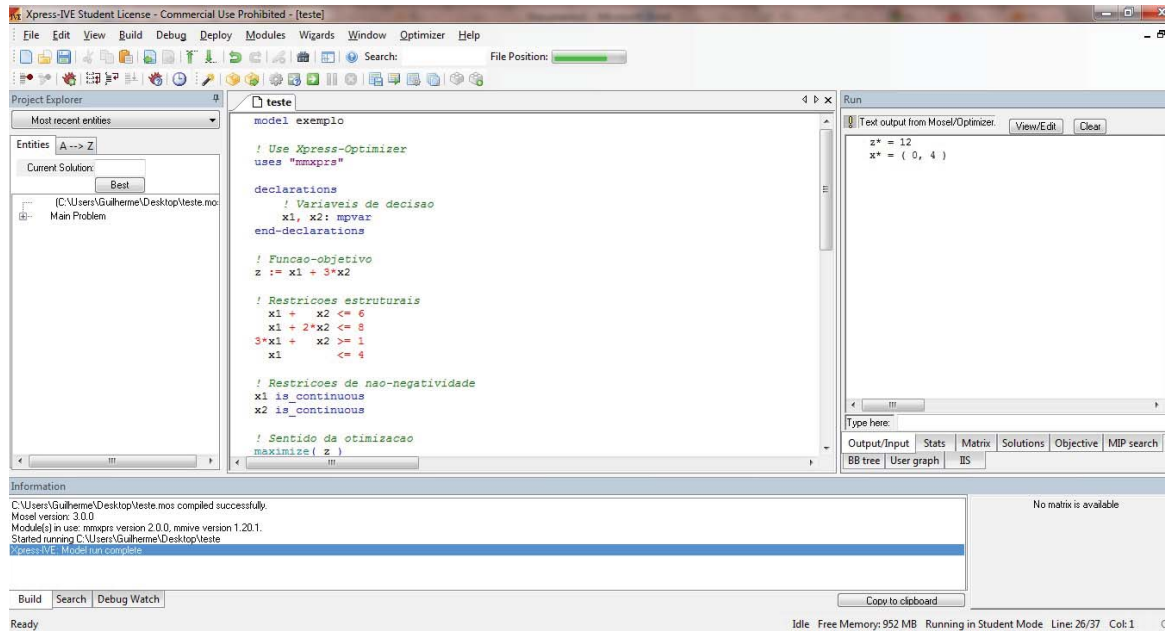


Figura 17: Exemplo e ilustração do uso do Xpress-Optimizer

No presente trabalho foi utilizado a versão acadêmica 20.00.05 do Xpress-Optimizer, bem como a versão 3.0.0 da linguagem Xpress-Mosel, para a modelagem do problema de coleta do palhiço de cana-de-açúcar, com a finalidade de obter uma solução viável com os padrões gerados pelo método dos algoritmos genéticos. Tal análise diz respeito ao suprimento da demanda.

No Apêndice E, é possível encontrar a modelagem do problema de coleta do palhiço, na linguagem Xpress-Mosel.

5.4 Experiência e resultados obtidos

Para a realização de experiências os dados aqui utilizados foram gerados de forma aleatória, como citado na subseção 6.1, com base nos dados apontados por Spadotto(2008), a fim de se verificar a qualidade do método proposto e em um futuro próximo realizar experiências com dados reais. Da mesma forma foi desenvolvido um software utilizando as técnicas do algoritmo genético para gerar um arquivo contendo padrões com os percursos percorridos pelos veículos para o recolhimento dos fardos de palhiço. E finalmente criado um modelo do problema no *solver* de otimização *Xpress-Optimizer* para determinar a solução que contenha um padrão para cada veículo e que não deixe nenhum talhão sem ser atendido.

Como padrão, para entrada dos dados foram utilizados 10 veículos de 5 capacidades diferentes e 20 talhões com tamanhos e quantidade de fardos diferentes.

Para um primeira experiência foram fornecidos ao software um arquivo contendo 10 padrões para cada veículo, a uma taxa de cruzamento igual a 0.9 e mutação 0.01. Como critério de parada, foram limitadas a 100 iterações.

Alguns dos padrões individuais gerados pelo software, de menor percurso dos veículos, podem ser vistos na Tabela 4 a seguir.

Tabela 4: Padrões individuais obtidos pelo software

Veículo	Padrão				
1	13	11	21	6	14
2	11	19	7	9	11
3	23	13	18	10	13
4	17	13	13	9	14
5	9	20	17	21	15
6	11	1	15	19	8
7	21	12	10	13	13
8	17	4	14	9	13
9	12	15	19	12	12
10	19	20	18	10	8

Os dados com a população de padrões, compostos pelo padrão, veículo, talhão visitado e quantidade carregada, gerados após a execução do programa, são armazenados em um arquivo texto, onde em uma segunda etapa é lido e resolvido pelo *Xpress-Optimizer*, para a resolução do problema de formação das rotas respeitando a demanda dos talhões.

Em um primeiro experimento, utilizando os dados a cima, gerados pelo programa, e uma vez aplicados ao *Xpress-Optimizer*, deram origem a um novo arquivo contendo os possíveis resultados do problema. A Tabela 5 a seguir exhibe os cinco melhores resultados encontrados neste experimento.

Tabela 5: Melhores resultados obtidos em um primeiro experimento

Veículo	P1	Custo	P2	Custo	P3	Custo	P4	Custo	P5	Custo
1	13	23	11	18	21	20	6	15	14	21
2	11	18	19	19	7	14	9	20	11	18
3	23	27	13	23	18	15	10	25	13	23
4	17	25	13	23	13	23	9	20	14	21
5	9	20	20	22	17	25	21	18	15	18
6	11	18	1	23	15	26	19	19	8	23
7	21	18	12	16	10	25	13	23	13	23
8	17	25	4	13	12	16	13	23	9	20
9	12	16	15	23	19	19	12	15	12	15
10	19	19	20	22	18	15	10	25	8	23
TOTAIS	#	209	#	202	#	198	#	203	#	205

Em uma segunda experiência foram fornecidos ao software um arquivo contendo 20 padrões para cada veículo, mantendo-se as taxas de cruzamento e mutação, bem como o critério de parada. Neste cenário verificou-se um aumento na divergência entre os valores obtidos como resultado, bem como o tempo de processamento, como pode ser visto na Tabela 6 abaixo.

Tabela 6: Resultado obtido em uma segunda experiência

Veículo	P1	Custo	P2	Custo	P3	Custo	P4	Custo	P5	Custo
1	19	19	10	25	21	18	6	15	18	15
2	23	15	13	23	12	16	9	20	10	25
3	12	16	19	19	16	20	13	23	12	16
4	29	25	17	25	21	18	18	15	15	23
5	31	20	21	18	15	23	21	18	19	19
6	12	16	18	15	12	16	19	19	20	22
7	20	22	18	15	19	19	15	23	21	18
8	16	20	15	23	22	13	7	14	9	20
9	21	18	20	22	12	16	15	23	14	21
10	15	23	17	25	23	15	20	22	13	23
TOTAIS	#	194	#	210	#	174	#	192	#	202

Tal fato se justifica pois quanto maior a população de entrada, mais dados serão computados, o que exige um processamento maior da máquina utilizada. É possível

verificar também um maior refinamento da solução obtida, pois quanto maior a entrada, ou seja, o espaço de busca, mais nos aproximamos da solução ótima do problema.

A Figura 18 ilustra os resultados obtidos nas duas experiências.

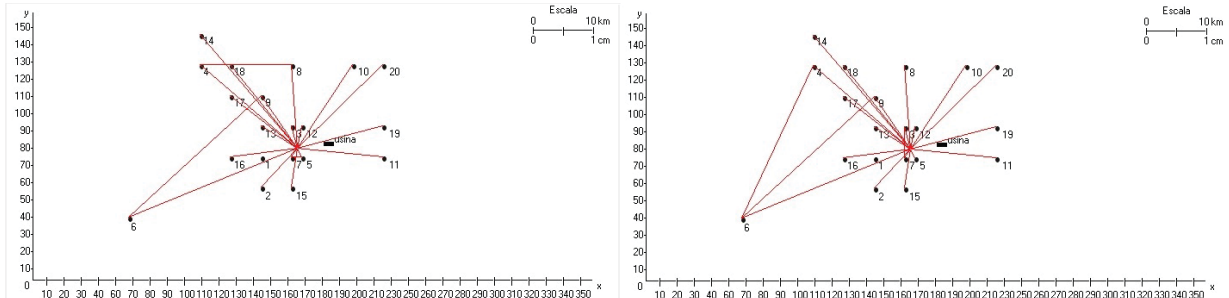


Figura 18: Representação gráfica dos resultados obtidos

O tempo de processamento no entanto diminuiu quando variou-se apenas a taxa de cruzamento para valores abaixo de 0.5, ou apenas a taxa de mutação acima de 0.01, porém aumentou a divergência dos resultados, comparados com os obtidos anteriormente, devido ao aumento no número de mutações e houve também uma redução no tamanho da população gerada, devido a diminuição da ocorrência de cruzamentos.

De uma forma geral para os demais intervalos os resultados se mostraram concisos e ainda quando comparados aos obtidos por Spadotto(2008) que busca maximizar a quantidade de fardos transportados in loco, a fim de reduzir o número de viagens dos veículos, em alguns casos foram satisfatórios, mostrando-se como uma excelente alternativa para o recolhimento do palhicho.

O sistema de transporte e coleta do palhicho de cana-da-açúcar é um tema pouco explorado por pesquisadores, o que dificulta a análise e comparação dos dados. A grande parte dos trabalhos encontrados na literatura são voltados ao sistema de transporte da cana-da-açúcar, ou no reaproveitamento do palhicho como fonte de energia, os únicos trabalhos encontrados sobre o seu transporte foram em Spadotto(2008) como visto anteriormente e em Franco(2003).

É necessário que sejam feitos outros estudos, utilizando outros modelos e metodologias como proposto no capítulo seguinte.

Capítulo 6

Conclusão e Trabalhos Futuros

Neste trabalho de conclusão de curso foi estudado a otimização do processo de coleta do palhiço de cana-de-açúcar, para o seu reaproveitamento.

O problema consiste em designar uma determinada frota heterogênea de veículos na coleta de palhiço nos talhões, buscando minimizar o percurso total percorrido. A estratégia proposta de reduzir o custo de transporte do palhiço entre a central de processamento localizada na usina e os talhões, considera as restrições de capacidade da frota, demanda dos talhões, a distância percorrida em outras.

A solução do problema foi dividida em duas partes. Na primeira parte foi implementada uma ferramenta em linguagem C, utilizando o ambiente de desenvolvimento Builder C++ 6.0. Tal ferramenta foi desenvolvida com base na heurística dos algoritmos genéticos, aplicados ao problema em questão conforme apresentado no capítulo 5. Os resultados obtidos pelo algoritmo serviram de base para a aplicação da segunda parte da solução, que consiste em resolver cada padrão do referido arquivo através do pacote *Xpress-Optimizer*.

Como pode ser visto no Capítulo 6, os resultados gerados pelo algoritmo genético fornecem boas soluções. Outra característica interessante da utilização de algoritmos genéticos é que o aproveitamento de uma certa quantidade de melhores indivíduos, gerados a partir da aplicação dos operadores, contribuem para a divergência e diversidade da solução.

No entanto os poucos testes que foram realizados com um conjunto de entradas maiores, variando apenas alguns de seus valores, apontaram divergências significativas, sem falar no tempo computacional exigido na execução de tal estratégia.

Portanto, se a aplicação da estratégia de solução por meio de algoritmos genéticos, para o problema de coleta do palhiço, for aplicada de maneira adequada, produzirá resultados satisfatórios.

6.1 Trabalhos Futuros

Como visto anteriormente embora os algoritmos genéticos tenham obtido bons resultados, alguns problemas e modificações na estratégia podem ser tratados e reformulados na continuidade deste estudo, a fim de melhorar os resultados, tais como:

- Utilizar outras ferramentas na segunda etapa de solução do problema no lugar da ferramenta comercial *Xpress-Optimizer* tais como, Gurobi, Cplex.
- Implementar um algoritmo exato para comparação.
- Fazer um estudo de caso em um ambiente real de aplicação do modelo, a fim de aprimorar a pesquisa.
- Utilizar a distância real do problema no lugar da distância euclidiana, aproximando assim a pesquisa do problema real. O mesmo pode ser tratado em um estudo de caso, conforme citado no item anterior.
- Implementar uma nova estratégia, incluindo janelas de tempo ao problema, referente a jornada de trabalho dos motorista.

Referências

- ARENALES, M. e. a. *Pesquisa Operacional*. [S.l.]: 4^a reimpressão, 2007.
- BOOTE K.J.; JONES J.W.; PICKERING, N. Potencial uses and limitations of crop models. *Agronomy Journal*, v. 88, p. 704 – 716, 1996.
- DIAS, R. A. *Engenharia de Tráfego em Redes IP sobre Tecnologia MPLS: Otimização Baseada em Heurísticas*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis - SC, 2004.
- FRANCO, F. N. *Alguns parâmetros de desempenho operacional de um sistema de recolhimento de palhiço de cana-de-açúcar*. Dissertação (Mestrado) — Escola Superior de Agricultura Luiz de Queiroz, Universidade de São Paulo, Piracicaba-SP, 2003.
- GALVÃO, F. D. A. *Otimização do sistema de coleta de resíduos de biomassa de madeira para fins energéticos*. Dissertação (Mestrado) — Escola Politécnica, Universidade de São Paulo, São Paulo - SP, 2004.
- GEORGE J. A.; ROBINSON, D. F. A heuristic for packing boxes into a container. *Computers and Operations Research*, v. 7, p. 147 – 156, 1980.
- GOLDBARG M.C.; LUNA, H. *Otimização Combinatória e Programação Linear - Modelos e Algoritmos*. [S.l.]: 2^a edição, 2000.
- GOLDBERG, D. E. Genetic algorithms in search, optimization and machine learning. In: _____. [S.l.]: 4^a edição, 1989. cap. 1, p. 432.
- MACHADO, E. PEREIRA e. Um simulador dinâmico do crescimento de uma cultura de cana-de-açúcar. *Bragantia*, v. 45, p. 107 – 122.
- MARTELLO S. E TOTH, P. Knapsack problems: Algorithms and computer implementations. *John Wiley & Sons*, 1990.
- PENTEADO, C. BEAUCLAIR e. Cronograma de corte da cana-de-açúcar através da programação linear. *Seminário de Tecnologia Agrônômica*, p. 424 – 434.
- RIPOLI, M. L. C. *Mapeamento do palhiço enfardado de cana-de-açúcar (Saccharum spp.) e do seu potencial energético*. Dissertação (Mestrado) — Escola Superior de Agricultura Luiz de Queiroz, Universidade de São Paulo, Piracicaba-SP, 2002.

SILVA, M. R. *Uma contribuição ao problema de localização de terminais de consolidação no transporte de carga parcelada*. Dissertação (Mestrado) — Escola Politécnica, Universidade de São Paulo, São Paulo - SP, 2004.

SOUZA, D. O. D. *Algoritmos genéticos aplicados ao planejamento de transporte principal de madeira*. Dissertação (Mestrado) — Universidade Federal do Paraná, Curitiba-PR, 2004.

SPADOTTO, A. F. *Desenvolvimento de Programa Computacional Aplicado ao Empacotamento do Palhão de Cana-de-açúcar*. Dissertação (Mestrado) — Faculdade de Ciências Agrônomicas, Universidade Estadual Paulista "Júlio de Mesquita Filho", Botucatu-SP.

SPADOTTO, A. FLORENTINO e. *O Problema da Mochila no Carregamento do Palhão da Cana-de-açúcar*. [S.l.], 2006.

ANEXO A - Local Talhões

Tabela de Dados de Localização dos Talhões

Talhão	X	Y	Distância do talhão i ao j (km)																			
1	25	15	0	21	5	23	15	13	10	15	11	13	14	7	21	18	14	6	2	14	8	16
2	25	18	12	0	1	6	20	2	4	11	20	9	13	15	14	10	19	10	5	21	8	28
3	28	12	23	13	0	6	9	5	4	6	3	10	17	14	25	17	7	26	5	25	17	26
4	19	6	34	5	1	0	9	5	16	6	3	12	9	17	26	13	7	21	26	20	12	13
5	29	15	44	12	23	5	0	9	13	5	15	4	9	7	24	6	15	19	14	12	11	17
6	12	21	5	6	12	25	5	0	21	19	17	16	10	21	21	9	15	31	31	31	16	16
7	28	15	12	12	4	28	5	18	0	20	13	16	11	24	29	9	12	15	19	16	11	4
8	28	6	2	12	1	21	23	35	15	0	13	15	5	7	21	6	19	18	10	27	4	4
9	25	9	12	21	7	12	19	19	3	2	0	4	11	7	20	17	16	30	12	27	3	6
10	34	6	11	4	5	13	14	12	4	9	12	0	18	10	25	4	13	33	11	10	3	2
11	37	15	10	5	18	23	11	14	12	9	22	21	0	12	19	4	14	13	15	9	17	25
12	29	12	22	11	12	1	11	20	34	2	20	12	4	0	21	17	29	2	19	9	24	22
13	25	12	14	21	33	23	17	23	10	22	23	10	4	12	0	17	21	2	21	24	28	14
14	19	3	7	16	23	11	15	25	18	32	10	8	8	16	23	0	13	24	14	11	32	17
15	28	18	5	6	11	12	10	6	18	22	8	6	21	14	34	12	0	18	3	11	15	15
16	22	15	21	7	21	23	11	6	32	12	8	15	21	14	27	22	11	0	5	8	16	3
17	22	9	13	7	20	23	8	18	32	12	18	10	7	4	16	26	7	21	0	19	13	3
18	22	6	3	14	22	19	5	15	9	22	33	7	11	4	13	23	5	6	15	0	13	14
19	37	12	4	14	12	10	11	15	9	25	31	3	12	18	11	19	24	18	23	32	0	21
20	37	6	6	21	26	5	12	10	10	12	21	12	13	20	25	10	20	11	13	19	4	0

ANEXO B – Talhões

Tabela de Dados dos Talhões			
Talhão	Área (m2)	Distância da usina (km)	Quantidade de fardos
i	area	Cij	Qij
1	700	23	21
2	655	20	14
3	778	14	23
4	534	11	12
5	654	11	10
6	675	22	21
7	876	22	32
8	713	35	31
9	544	27	13
10	695	27	15
11	498	14	14
12	738	12	35
13	756	12	22
14	867	33	29
15	511	27	26
16	456	35	18
17	657	15	11
18	834	15	29
19	788	26	32
20	896	31	19

ANEXO C – Frota

Tabela de Dados da Frota			
Veículo	Tipo	Capacidade (ton)	Custo
1	A	15	3
2	B	25	4
3	E	60	7
4	C	35	5
5	E	60	7
6	D	45	6
7	E	60	7
8	C	35	5
9	C	35	5
10	E	60	7

APÊNDICE A – Código Fonte Calculo Aptidão

```

1 //Função que calcula aptidão
2
3 void __fastcall TForm_Principal::calcula_aptidao(ENTRADA *
4     e_entrada, TALHOES *t_talhoes, LOCALTALHOES *l_talhoes){
5
6     int i,j, padrao_atual;
7
8     for(i = 0; i < nr_entradas; i++){
9
10        padrao_atual = 1;
11
12        do{// Bloco de leitura padrões
13
14            if (padrao_atual != e_entrada[i].id_padrao){
15                for(j=0; j <= nr_talhoes; j++){ //Percorre
16                    talhões
17                        if(t_talhoes[j].id == e_entrada[
18                            i-1].id_talhao){
19                            distancia = distancia +
20                                t_talhoes[j].
21                                    distancia_usina;
22                            custo[padrao_atual][e_entrada[i].id_veiculo]
23                                = distancia;
24                        }
25                    }
26                }
27                padrao_atual++;

```

```

21         distancia = 0;
22     }
23
24     if (distancia == 0){//Partindo da Usina
25         for(j=0; j <= nr_talhoes; j++){
26             //Percorre talhões
27             if(t_talhoes[j].id == e_entrada[i].id_talhao
28                 ){
29                 distancia = distancia + t_talhoes[j].
30                     distancia_usina;
31             }
32         }
33     }
34     else{
35         for(j=0; j <= nr_talhoes; j++){ //Percorre
36             talhões
37             if(l_talhoes[j].id == e_entrada[i].id_talhao
38                 ){
39                 distancia = distancia + l_talhoes[j].
40                     distancia[j][e_entrada[i-1].id_talhao
41                         + 1];
42             }
43         }
44     }
45     i++;
46     }while(e_entrada[i].id_veiculo == e_entrada[i
47         -1].id_veiculo);
48 }

```

Listagem 1: 'Código Fonte - Cálculo da Aptidão'

APÊNDICE B – Código Fonte Seleção

```

1 //Função seleção
2
3 void __fastcall TForm_Principal::selecao () {
4     //Variáveis locais
5     ENTRADA *melhor [MAX] [MAX];
6     double prob_aptidao [MAX] [MAX] = {0.0};
7     int i, j;
8     //Calcula probabilidade de seleção das entradas
9     for (i = 1; i <= nr_veiculos; i++){
10         for (j = 1; j <= nr_veiculos; j++){
11             prob_aptidao [j] [i] = (1.0/double(custo [j
12                 ] [i]));
13         }
14     }
15     //Seleciona cromossomos pelo método da roleta
16     for (i = 0; i < nr_entradas; i++)
17         critério_selecao [i] = roleta (prob_aptidao [i] [j])
18         ;
19     //Atualiza melhores cromossomos
20     for (j = 0; j < nr_entradas; j++)
21         for (i = 0; i < nr_veiculos; i++)
22             melhor [j].padrao = entrada [
23                 critério_selecao [j] ].padrao [i
24                 ];
25 }

```

APÊNDICE C – Código Fonte Cruzamento

```

1 //Função cruzamento
2 void __fastcall TForm_Principal::cruzamento(){
3     //Variáveis de entrada
4     int i,j, ponto_de_corte;
5     ENTRADA *auxiliar, *anterior;
6     // Percorre cromossomos de entrada
7     for(i = 0; i < nr_entradas; i+=2){
8         //Segundo a taxa de cruzamento
9         if (randomico(0,1) <= taxa_cruzamento){
10            //Escolhe ponto de corte aleatoriamente
11            ponto_de_corte = irandomico(1,entrada[i
12                ].padrao-1);
13            //Efetua o cruzamento a partir do ponto de corte
14            for (j = ponto_de_corte; j < nr_talhoes; j++) {
15                auxiliar = entrada[i].qt_fardos;
16                entrada[i].qt_fardos = entrada[i
17                    +1].qt_fardos;
18                entrada[i+1].qt_fardos =
19                    auxiliar;
20            }
21        }
22        //Atualiza os pesos dos cromossomos
23        acerta_capacidade_caminhao(anterior, entrada);
24    }
25 }

```

APÊNDICE D – Código Fonte Mutação

```
1 // Função mutação
2
3 void __fastcall TForm_Principal::mutacao() {
4
5     //Variáveis locais
6     int    i, j;
7
8     //Percorre cromossomo de entrada
9     for (i =0; i < nr_entadas; i ++)
```

Listagem 1: 'Código Fonte - Mutação'

APÊNDICE E – Código Xpress-Mosel

```
1 !supondo que o arquivo de dados criado pelo software ja esteja
   ordenado
2
3 model coleta_palhico
4 uses "mmxprs"; !gain access to the Xpress-Optimizer solver
5
6 parameters
7     DATAFILE="resultado.txt"
8     FILEOUT="ffdresultadomosel.txt"
9 end-parameters
10
11 !variáveis do modelo
12 declarations
13     acabou: boolean
14     p: integer
15     i: integer
16     j: integer
17     v: integer
18     c: array(i, j) of integer
19     x: array(i, j) of boolean
20     y: array(i) of integer
21     d: array(i) of integer
22     C: array(i) of integer
23 end-declarations
24
25 !declarações do arquivo
26 declarations
27     padrao: integer
```

```

28         veiculo: integer
29         custo: integer
30 end-declarations
31
32 !lendo dados do arquivo
33 initializations from DATAFILE
34         padrao as 'PADRAO'
35         veiculo as 'VEICULO'
36         custo as 'CUSTO'
37 end-initializations
38
39 ! Funcao-objetivo
40 z := sum(i)*sum(j)*sum(v)c(i,j)*x(i,j)
41
42 ! Restricoes estruturais
43 sum(1 in v)y(i) = d(i)
44 sum(i)y(i) <= C(i)
45 y(i) <= sum(i)d(j)*x(i,j)
46 sum(1 in i)x(i,p) - sum(1 in j)x(p,j)
47
48 ! Restricoes de nao-negatividade
49 i is_continuous
50 j is_continuous
51 v is_continuous
52
53 ! Sentido da otimizacao
54 minimize( z )
55
56 ! Exibe o valor da funcao-objetivo
57 writeln("z* = ", getobjval)
58
59 !----- inicializacao -----
60
61 !--- lendo arquivo -----
62 initializations from DATAFILE

```



```

63         [padrao, veiculo, custo] as 'DADOS'
64 end-initializations
65 !variavel acabou
66 acabou := false
67 !variavel padrao: p
68 p :=0
69 !----- fim -----
70
71 !--- escrevedno arquivo -----
72
73 !abrindo um arquivo para escrita
74 fopen(FILEOUT, FOUTPUT)
75 !escrevendo no arquivo
76 writeln(" Solução: \n\n")
77 forall(j in 1..p) do
78 forall(i in 1..p) do
79 write(x(i,j), " ")
80 end-do
81
82 write("\n\n veiculo:", veiculo, " padrão:", padrao, " custo", custo)
83
84 !fechando o arquivo
85 fclose(FOUTPUT)
86
87 !----- fim -----
88
89 end-model

```

Listagem 1: 'Código Mosel - Coleta do palheiro'