

MÉTODOS METAHEURÍSTICOS APLICADOS A UM MODELO DE PLANEJAMENTO DE CULTURAS

Cíntia Pimentel de Oliveira

Dissertação apresentada à Universidade
Estadual Paulista “Júlio de Mesquita Filho”
para obtenção do título de Mestre em
Biometria.

BOTUCATU
São Paulo - Brasil
Março - 2013

MÉTODOS METAHEURÍSTICOS APLICADOS A UM MODELO DE PLANEJAMENTO DE CULTURAS

Cíntia Pimentel de Oliveira

Orientadora: Profa. Dra. **Helenice de Oliveira Florentino Silva**

Dissertação apresentada à Universidade
Estadual Paulista “Júlio de Mesquita Filho”
para obtenção do título de Mestre em
Biometria.

BOTUCATU
São Paulo - Brasil
Março - 2013

Agradecimentos

Não há palavras para descrever o sentimento de gratidão pelo aprendizado e experiências que a universidade tem me proporcionado nos últimos anos.

Descobri que os desafios vêm e vão, e diante deles sempre temos algo melhor para buscar e dar de nós mesmos. Enfrentá-los só se torna uma tarefa possível com fé em Deus e ajuda de cada pessoa que passa por nossas vidas, contribuindo de alguma forma para que vençamos mais uma etapa.

Agradeço aos meus pais, João e Inês, por serem meus melhores amigos, e pelo apoio e amor incondicional dado em todos os momentos da minha vida.

Aos amigos da pós-graduação, pessoas inesquecíveis com quem dividi grande parte do tempo, conhecimento, alegrias e angústias. De modo especial, agradeço aos amigos Ronaldo, Thiago, Farid, Livia e Bettina. A vocês desejo muito sucesso.

A minha querida orientadora, Prof^a. Helenice, pela paciência, simpatia, competência, incentivos, conselhos e amizade, características estas indispensáveis para a realização de uma boa orientação e de um trabalho com satisfação.

A todos os professores que ensinaram, aconselharam e compartilharam suas experiências, em especial, o Prof. Carlos Roberto Padovani, que, para mim, é mais que um professor, é um amigo, um exemplo, um educador.

Ao Programa de Pós-Graduação em Biometria do Instituto de Biociência de Botucatu (IBB) por possibilitar o desenvolvimento deste trabalho.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro concedido.

Sumário

	Página
LISTA DE FIGURAS	vi
LISTA DE TABELAS	ix
RESUMO	xi
SUMMARY	xiii
1 INTRODUÇÃO	1
2 REVISÃO DE LITERATURA	3
2.1 Modelagem Matemática	3
2.2 Otimização	6
2.2.1 Formulação geral de um problema de Otimização	7
2.2.2 Otimização global e local	9
2.2.3 Otimização Combinatória	9
2.3 Exato <i>versus</i> Heurístico	10
2.4 Estratégias Heurísticas	12
2.4.1 Metaheurísticas	13
2.5 Algoritmo Genético (AG)	15
2.5.1 Representação de um cromossomo	18
2.5.2 População	21
2.5.3 Aptidão	23
2.5.4 Penalização	24

	iv
2.5.5 Elitismo	25
2.5.6 Seleção	26
2.5.6.1 Proporcional	28
2.5.6.2 Torneio	31
2.5.6.3 Classificação	32
2.5.6.4 Truncada	34
2.5.7 Operadores Genéticos	35
2.5.7.1 <i>Crossover</i>	35
2.5.7.2 Mutação	39
2.5.7.3 Epidemia	40
2.5.8 Parâmetros	40
2.5.9 Critério de parada	42
2.6 <i>Simulated Annealing</i> (SA)	42
2.7 Algoritmos Híbridos	46
3 MATERIAL E MÉTODOS	48
3.1 Modelagem Matemática do Problema Biológico	48
3.2 Heurística Construtiva	54
3.3 Penalização	55
3.4 Algoritmo Genético	56
3.5 <i>Simulated Annealing</i>	59
3.6 Híbrido AG+SA	60
3.7 Instâncias simuladas	60
3.8 Algoritmos testados	62
4 RESULTADOS E DISCUSSÕES	66
5 CONCLUSÕES	81
REFERÊNCIAS BIBLIOGRÁFICAS	83

APÊNDICE

Lista de Figuras

Página

1	Espaço de busca	7
2	Fluxograma do processo de busca do AG em uma abordagem geral . . .	18
3	Itens que compõem o problema do agricultor	19
4	Cromossomo com representação binária	20
5	Cromossomo com representação inteira	20
6	População inicial gerada aleatoriamente para o problema do agricultor .	21
7	Representação esquemática da população inicial e população intermediária	22
8	Fluxograma do Algoritmo Genético com elitismo	26
9	Roleta com aptidão normalizada para a população inicial gerada no pro- blema do agricultor	29
10	Ponteiros construídos pelo método de Amostragem Estocástica Uniforme para o problema do agricultor	31
11	Posições de corte possíveis no cromossomo com representação inteira . .	36
12	Troca de genes entre cromossomos pais realizados a partir do ponto de corte selecionado	36
13	Troca de genes entre cromossomos pais realizados com <i>crossover 2-pontos</i>	37
14	Troca de genes entre cromossomos pais realizados com <i>crossover</i> uniforme	38
15	Mutação aleatória em cromossomo com representação binária	40
16	Probabilidade de aceitação de movimentos que não são de melhora pelo algoritmo SA	45
17	Pseudocódigo do algoritmo <i>Simulated Annealing</i>	46

18	Área hipotética dividida em lotes para planejamento de plantio	50
19	Representação em grafo da região ilustrada na Figura (18)	50
20	Matriz de relacionamento de vizinhança entre lotes	52
21	Matriz de probabilidade de proliferação de praga entre culturas	52
22	Comportamento da função de penalização exponencial com $k = 100$	56
23	Exemplo de matriz solução para o problema abordado	57
24	Comportamento da função logística de probabilidade de mutação	58
25	Possível vizinho da solução corrente S apresentada na Figura (23)	59
26	Geometria e dimensão das áreas consideradas nas simulações	61
27	Boxplot com resultados de homogeneidade e desempenho dos algoritmos testados na instância A para valores da função objetivo	72
28	Boxplot com resultados de homogeneidade e desempenho dos algoritmos testados na instância A para o tempo computacional	73
29	Boxplot com resultados de homogeneidade e desempenho dos algoritmos testados na instância B para valores da função objetivo	74
30	Boxplot com resultados de homogeneidade e desempenho dos algoritmos testados na instância B para o tempo computacional	75
31	Melhor configuração de plantio apresentada pelo algoritmo <i>AG_Torneio_A</i>	76
32	Melhor configuração de plantio apresentada pelo algoritmo <i>AG_Roleta_A</i>	76
33	Melhor configuração de plantio apresentada pelo algoritmo <i>SA_10⁻⁴_A</i> . .	77
34	Melhor configuração de plantio apresentada pelo algoritmo <i>AG+SA_Torneio_A</i>	77
35	Melhor configuração de plantio apresentada pelo algoritmo <i>AG+SA_Roleta_A</i>	77
36	Melhor configuração de plantio apresentada pelo algoritmo <i>AG_Torneio_B</i>	77
37	Melhor configuração de plantio apresentada pelo algoritmo <i>AG_Roleta_B</i>	78
38	Melhor configuração de plantio apresentada pelo algoritmo <i>SA_10⁻⁴_B</i> .	78
39	Melhor configuração de plantio apresentada pelo algoritmo <i>SA_10⁻⁶_B</i> .	78

40	Melhor configuração de plantio apresentada pelo algoritmo <i>AG+SA_Roleta_B</i>	79
41	Curva de melhora realizada pelo SA iniciado com a melhor solução factível do AG com seleção por Torneio Binário para a instância A	79
42	Curva de melhora realizada pelo SA iniciado com a melhor solução factível do AG com seleção por Roleta Viciada para a instância A	80
43	Curva de melhora realizada pelo SA iniciado com a melhor solução factível do AG com seleção por Roleta Viciada para a instância B	80

Lista de Tabelas

	Página
1 Aptidão das soluções geradas para o problema do agricultor	23
2 Valores de aptidão normalizada e acumulada para os indivíduos gerados no problema do agricultor	29
3 Vocabulários utilizados no processo de recozimento físico e simulado . . .	43
4 Parâmetros utilizados no <i>Simulated Annealing</i>	60
5 Áreas dos lotes	62
6 Valor da função objetivo e tempo computacional (em segundos) médios para os algoritmos testados na instância A	66
7 Valor da função objetivo e tempo computacional (em segundos) médios para os algoritmos testados na instância B	67
8 Número de soluções factíveis encontradas pelos algoritmos testados na instância B	68
9 Homogeneidade e desempenho dos algoritmos obtidos para a instância A com relação ao valor da função objetivo	69
10 Homogeneidade e desempenho dos algoritmos obtidos para a instância B com relação ao valor da função objetivo	69
11 Homogeneidade e desempenho dos algoritmos obtidos para a instância A com relação ao tempo computacional	71
12 Homogeneidade e desempenho dos algoritmos obtidos para a instância B com relação ao tempo computacional	71
13 Informações sobre culturas: nome, família, época de plantio e duração do ciclo de vida, adaptado de Santos (2009)	88

14	Produtividade mensal por m^2 das culturas, adaptado de Aliano Filho (2012)	89
15	Demanda e períodos de demanda das culturas, adaptado de Aliano Filho (2012)	90

MÉTODOS METAHEURÍSTICOS APLICADOS A UM MODELO DE PLANEJAMENTO DE CULTURAS

Autora: CÍNTIA PIMENTEL DE OLIVEIRA

Orientadora: Profa. Dra. HELENICE DE OLIVEIRA FLORENTINO SILVA

RESUMO

No contexto atual de sustentabilidade, formas alternativas à utilização de produtos que podem levar o planeta a uma crise ambiental estão sendo fortemente estudadas. Na agricultura, uma preocupação brasileira é com a redução do consumo de agrotóxicos, já que o país é um dos maiores consumidores mundiais do produto.

Neste sentido, o planejamento das atividades agrícolas é uma ação preventiva que pode colaborar com a redução da proliferação de pragas entre as culturas plantadas em lotes e, conseqüentemente, a redução no consumo de produtos químicos. Ele visa determinar a melhor forma de ocupação de áreas de plantio de modo que a disposição das culturas e/ou variedades em talhões favoreça o controle eficiente de pragas.

Neste trabalho é proposto um modelo de otimização 0-1 para minimizar a probabilidade de proliferação de pragas entre as culturas plantadas, respeitando

restrições de demanda, ocupação de lotes, período de cultivo das culturas e tempo de planejamento. Nesta modelagem, considera-se uma área de plantio genérica, com lotes irregulares e culturas já plantadas em fazendas vizinhas.

Para resolução do modelo foram utilizadas as estratégias Meta-heurísticas Algoritmo Genético (AG), *Simulated Annealing* (SA) e a abordagem híbrida Algoritmo Genético com *Simulated Annealing* (AG+SA) para obtenção de boas estimativas para o planejamento otimizado das atividades agrícolas.

Aplicações práticas a duas instâncias distintas foram realizadas e as melhores soluções factíveis encontradas pelos algoritmos são apresentadas, bem como comparação e discussão dos resultados e desempenho computacional dos métodos.

Os resultados computacionais indicam que os algoritmos meta-heurísticos propostos encontraram soluções factíveis de boa qualidade em um tempo computacional reduzido, e em especial, as estratégias híbridas demonstraram ser excelentes ferramentas para auxílio no planejamento de plantio.

METAHEURISTIC METHODS APPLIED TO A PLANNING CULTURES MODEL

Author: CÍNTIA PIMENTEL DE OLIVEIRA

Adviser: Profa. Dra. HELENICE DE OLIVEIRA FLORENTINO SILVA

SUMMARY

In the sustainability context we are living nowadays, alternative forms to the utilization of products that can lead the Earth to an environmental crisis have been strongly studied. In agriculture, the Brazilian concern is about the reduction of pesticide consumption, once the country is one of the world's major consumers of it.

In this sense, the planning of the agricultural activity is a preventive action that can contribute to reduce the proliferation of pests among crops planted in lots and, consequently, decrease the excessive consumption of such chemicals. This approach aims to determine the best way to use the planting areas arranging the crops and/or varieties in support to an efficient pests control.

The research's objective is to propose an optimization model 0-1 in order to minimize the likelihood of pests proliferation among planted crops, considering constraints demand, lots occupation, crops growing period and planning time.

In this model, it is considered a generic planting area with irregular lots and already planted crops in neighboring farms.

In the model's solutions, Genetic Algorithm and Simulated Annealing Metaheuristics strategies were used, as well as a Genetic Algorithm jointly with Simulated Annealing hybrid approach in order to obtain good estimates for the optimized planning of agricultural activities.

A practical application for the two instances was performed and the best feasible solutions found by the algorithms are presented as well as a comparison and discussion of the results and performance of computational methods.

The computational results indicate that the metaheuristic algorithms found feasible solutions with good quality in a reduced computational time, and in particular, the proposed hybrid strategies have proven to be excellent tools to help planning the planting.

1 INTRODUÇÃO

Com terras férteis, extensas e clima propício para a agricultura, o Brasil é um dos principais produtores e fornecedores mundiais de alimentos. Grande produção agrícola em atividade torna o país o cenário favorável ao surgimento das pragas. Pragas são organismos que causam danos econômicos ao agricultor, e o meio mais utilizado pelo homem em seu combate são os agrotóxicos (Martins Jr., 2002).

Os agrotóxicos tornaram-se uma preocupação brasileira devido ao seu uso indiscriminado em lavouras. Em 2010 foram consumidos no Brasil cerca de 790 mil toneladas de agrotóxicos, conforme pesquisa publicada no boletim *Em Questão* (edição 1460, de 31 de janeiro de 2012) da Secretaria de Comunicação da Presidência da República, o que coloca o país entre os maiores consumidores de agrotóxicos do mundo.

O controle de pragas com agrotóxicos tem se tornado cada vez mais ineficiente devido à resistência que as pragas adquirem com o tempo, exigindo dosagens cada vez mais elevadas e aplicações mais frequentes, onerando o custo de produção, aumentando os riscos de contaminação do meio ambiente e prejudicando a saúde humana e animal (Martins Jr., 2002).

Atualmente, a principal alternativa à utilização de agrotóxicos é o controle biológico, que tem se mostrado eficiente no combate as pragas das principais culturas produzidas no Brasil. Estudos realizados por Parra et al. (2002) e Venzon et al. (2008) utilizando controle biológico demonstram casos de sucesso e o potencial de redução na intensidade de infestação em culturas como citrus, cana-de açúcar, milho, sorgo, tomate, entre outros. Contudo, tanto o controle biológico quanto o químico, em geral, são ações não preventivas de combate às pragas, ou seja, são

utilizados quando existem focos de pragas ou estas já estão instaladas na lavoura.

Neste cenário, técnicas preventivas e alternativas sustentáveis de combate às pragas têm sido desenvolvidas, e dentre elas destaca-se o planejamento das atividades de plantio em sistemas agrícolas. A ação de planejar visa fornecer ao agricultor, diante de certas limitações de recursos, a melhor forma de ocupação de suas terras com as culturas, com objetivos específicos, como por exemplo, aumentar o lucro ou diminuir os custos.

Assim, neste trabalho é proposto um modelo matemático para auxiliar o planejamento de culturas em sistemas agrícolas, cujo objetivo é investigar, através de estratégias metaheurísticas de otimização, a melhor forma de ocupação de uma área disponível para plantio, subdividida em lotes, de modo que a disposição dessas culturas e/ou variedades em talhões favoreça o controle eficiente de pragas.

Aplicações práticas foram realizadas e as melhores soluções factíveis encontradas pelos algoritmos estudados são apresentadas, bem como a comparação e discussão dos resultados e desempenho computacional dos métodos.

A ação de planejar as atividades agrícolas pode contribuir para a minimização da proliferação de pragas entre as culturas e apresenta ganhos sociais, econômicos e ambientais, tais como, atender a demanda social de produtos livres de agrotóxicos, reduzir gastos do agricultor com produtos de combate às pragas, reduzir os riscos de contaminação ambiental com produtos químicos, entre outros.

2 REVISÃO DE LITERATURA

2.1 Modelagem Matemática

Nos últimos anos o Brasil tem ocupado posição de destaque no cenário mundial de produção e exportação de alimentos, e para atender tamanha demanda, grandes investimentos econômicos, tecnológicos e logísticos têm sido realizados. Com este avanço, os problemas nas agroindústrias se tornaram bastante complexos, fazendo com que os gestores lancem mão de todas as ferramentas possíveis de apoio, como exemplo, as computacionais e matemáticas. Assim, destacam-se as importantes contribuições da *modelagem matemática* para as pesquisas nesta área.

Por sua característica interdisciplinar, a modelagem matemática tem contribuído de forma significativa para a expansão das atividades agrícolas no país e para a redução dos problemas econômicos e ambientais decorrentes deste crescimento.

Para Biembengut & Hein (2000), a modelagem matemática constitui um ramo próprio da Matemática que tenta traduzir situações reais para uma linguagem matemática, para que por meio dela se possa melhor compreender, prever e simular ou, ainda, mudar determinadas vias de acontecimentos, com estratégias de ação, nas mais variadas áreas do conhecimento.

Bassanezi (2002) define modelagem matemática como a arte de transformar problemas da realidade em problemas matemáticos e resolvê-los interpretando suas soluções na linguagem do mundo real.

Numa perspectiva agrícola, a modelagem matemática visa transformar um problema econômico ou biológico real em objeto de estudo, possibilitando a previsão do comportamento do sistema de produção e a compreensão da relação e

influência entre os seus diversos componentes.

Contudo, trabalhar com modelagem matemática não é uma tarefa simples. Em geral, é preciso sair de uma área dominada para uma área em que os outros dominam e, para isso, é necessário bases e fundamentos.

Construir um modelo matemático, segundo Biembengut & Hein (2000), significa determinar um conjunto de símbolos e relações matemáticas que traduz, de alguma forma, o fenômeno em questão ou problema de situação real. Contudo, é importante ressaltar que a realidade é tão complexa que modelos mais rebuscados e que incorporam grande parte das variáveis reais podem se tornar difíceis de resolver, enquanto um modelo muito simples pode não representar bem o problema real.

Portanto, o que se observa na maioria das vezes, é a proposta de um modelo simples que permite explorar bons aspectos do fenômeno observado e a partir daí, incorporar mais variáveis e equações, a fim de aprimorar o modelo.

Existem vários trabalhos na literatura que foram desenvolvidos utilizando modelagem matemática com foco em planejamento de produção, em especial em produção agrícola. Em geral, estes trabalhos buscam otimizar áreas de produção, aproveitamento de matérias-primas, reduzir custos, aumentar lucro, reduzir produção de resíduos, entre outros.

Kantorovich (1960) foi pioneiro em propor modelos matemáticos para melhorias na organização e planejamento de produção. Em seu trabalho, o autor propõe modelos para determinar desde a distribuição ideal de terras cultiváveis, do trabalho de máquinas e mecanismos, até modelos para redução da produção de resíduos obtidos em cortes de matérias-primas pela indústria.

Desde então, o interesse por pesquisas de melhorias na área agrícola tem crescido, e modelos matemáticos têm sido utilizados para determinar, segundo Bernardon & Calgaro (2007), a alocação ótima de culturas em área de plantio, produtividade de culturas e de riscos de produção em sistemas de cultivo, crescimento de colheitas individuais ou a gerência de sistema de cultivos, a melhoria dos genótipos

e dos cultivares, o manejo da água de irrigação e a avaliação de risco da colheita e a segurança do alimento.

Com o objetivo de planejar o plantio de culturas em sistemas agrícolas, destacam-se trabalhos recentes com elaboração de modelos para a programação de rotação de culturas.

Fey et al. (2000) utilizaram um modelo de programação linear com restrições de terra, rotação de culturas, financeiras e de maquinaria, para planejar o plantio de culturas em uma propriedade agrícola real, visando aumentar a sua lucratividade.

Santos et al. (2007) propuseram um modelo de programação linear para a programação de rotação de culturas em uma área de plantio, cujo objetivo era maximizar a ocupação das áreas de plantio sujeito a restrições de vizinhança, sucessão para culturas de mesma família botânica, adubação verde e pousio.

Santos (2009) e Aliano Filho (2012) propuseram um modelo de programação linear para maximizar o lucro da rotação de culturas, sendo o último com acréscimo de restrições de demanda, e utilizaram estratégias metaheurísticas para resolução dos modelos. Também Gomes & Arenales (2010) estenderam as ideias de Santos (2009) para problemas com restrições de áreas mínimas para lotes e redução do número total de lotes usados.

Como visto, existe um amplo uso de modelos matemáticos na representação, análise e obtenção de estimativas de parâmetros para problemas reais na agricultura. O crescimento do número de pesquisas nesta área é consequência da importância que a agricultura tem representado para a economia brasileira, especialmente no que tange à redução de problemas ambientais.

Nas seções seguintes serão abordadas as técnicas necessárias para a re-solução do modelo objeto deste trabalho.

2.2 Otimização

A palavra “ótimo” vem do Latim *optimus* e significa “o melhor”. Otimizar refere-se, portanto, à tentativa de trazer o que estamos lidando em direção ao seu estado final, isto é, para o seu melhor (Andréasson et al., 2005).

Segundo Beightler, Phillips e Wilde, citado por Goldberg (1989), a teoria da otimização estuda como descrever e alcançar o que é melhor, uma vez que se sabe como medir e modificar o que é bom ou ruim. Ela abrange o estudo quantitativo do ótimo e métodos para encontrá-lo.

A Otimização é, portanto, uma área de conhecimento dentro da Matemática que fornece as ferramentas apropriadas para que melhorias desejadas em um determinado processo sejam alcançadas. E ainda, as técnicas de otimização visam, segundo Brandão & Saramago (2011), determinar a melhor solução para um problema sem ter que testar todas as possibilidades envolvidas.

Problemas de otimização podem ser encontrados nas mais diversas áreas das ciências exatas, biológicas e tecnológicas, e sua utilização está fundamentada na capacidade de auxiliar a tomada de decisões de forma eficiente.

Inicia-se o processo de modelagem identificando as variáveis do problema. Isso significa reconhecer no problema quais são as principais variáveis envolvidas no processo a ser otimizado, chamadas de *variáveis de decisão*. Em seguida, define-se o objetivo do problema, que pode ser de minimização ou maximização de uma função das variáveis de decisão, chamado de *função objetivo*. As variáveis de decisão muitas vezes devem obedecer à condições limitantes inerentes ao problema, que são chamadas de *restrições*. As equações que descrevem as restrições podem ser de igualdade ou desigualdade. A solução a ser encontrada pertence a um conjunto viável, denominado *espaço factível*. A Figura (1) apresenta a idéia de espaço de busca por soluções factíveis.

O procedimento descrito acima, de identificação do objetivo, variáveis e restrições de um dado problema é chamado de *modelagem* (Wright & Nocedal, 1999).

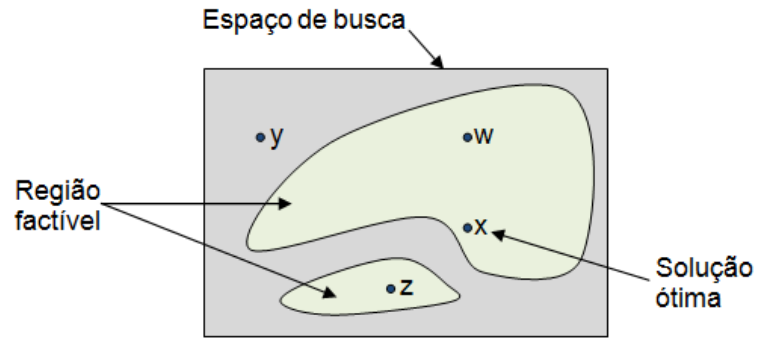


Figura 1 - Espaço de busca

2.2.1 Formulação geral de um problema de Otimização

Em geral, um problema de otimização é escrito como:

$$\begin{aligned}
 &\text{Minimizar} && f(\mathbf{x}) \\
 &\text{Sujeito a:} && h_i(\mathbf{x}) = 0, \quad i \in I \\
 &&& g_j(\mathbf{x}) \leq 0, \quad j \in J \\
 &&& \mathbf{x} \in D.
 \end{aligned}$$

Nesta formulação, $\mathbf{x}$ é um vetor n -dimensional das variáveis x_i , $\mathbf{x} = (x_1, x_2, \dots, x_n)$, e $f(\mathbf{x})$, $h_i(\mathbf{x})$ e $g_j(\mathbf{x})$ são funções reais das variáveis x_i . I e J representam os conjuntos de índices das restrições de igualdade e desigualdade, $I = 1, 2, \dots, m$ e $J = 1, 2, \dots, n$. O conjunto D é um subconjunto do espaço n -dimensional, $D \subset R^n$. A função f é a função objetivo do problema, definida em $f : R^n \rightarrow R$ e as equações de igualdade $h_i = 0$ e desigualdade $g_j \leq 0$ são as suas restrições, em que $h_i : R^n \rightarrow R$ e $g_j : R^n \rightarrow R$ (Luenberger & Ye, 2008).

A função objetivo e as funções de restrições podem ser lineares ou não lineares em relação às variáveis de decisão. Esta característica classifica o problema como sendo pertencente à Programação Linear ou Programação Não-Linear.

De acordo com Bregalda et al. (1988), em matemática, problemas de Programação Linear são problemas de otimização nos quais a função objetivo e as

restrições são todas lineares. Em contrapartida, um modelo de otimização constitui um problema de Programação Não-Linear se exibir qualquer tipo de não-linearidade, seja na função objetivo ou em qualquer de suas restrições (Goldbarg & Luna, 2005).

Além disso, algumas características importantes a serem observadas no problema a ser otimizado são: o tipo das variáveis de decisão, que podem ser discretas ou contínuas, e se possui ou não restrições, denominado problema restrito ou irrestrito.

Determinados problemas de otimização só tem sentido quando as respostas obtidas através das variáveis de decisão forem expressas por números inteiros, $x_i \in Z$, em que Z é o conjunto dos números inteiros, ou por números binários, $x_i \in \{0, 1\}$. Nestes casos, denominamos problemas de otimização discreta.

Por exemplo, se o objetivo é determinar o número de funcionários para realizar determinadas tarefas, como nos clássicos problemas de designação, utilizam-se variáveis inteiras. Ou ainda, se o objetivo é decidir se coloca ou não um determinado item na mochila, como nos clássicos problemas da mochila, é satisfatório utilizar variáveis binárias para expressar as respostas sim ou não, verdadeiro ou falso.

Caso contrário, quando as variáveis de decisão puderem assumir quaisquer valores reais, $x_i \in R$, denomina-se problemas de otimização contínua. Nestes problemas, em geral, deseja-se determinar as melhores medidas, os melhores tempos, dentre outras variáveis de natureza contínua.

Os problemas de otimização irrestritos são aqueles em que os conjuntos de restrições I e D são vazios, ou seja, $I = D = \emptyset$. Porém, a maioria dos problemas encontrados na natureza não variam livremente, ou seja, possuem alguma restrição com relação a quantidades disponíveis, tempo, capacidade, entre outros. Isso caracteriza os problema de otimização restritos, em que $I \neq \emptyset$ e/ou $D \neq \emptyset$.

Após a formulação de um modelo representativo para o problema real e determinadas as suas principais características, é possível aplicar as técnicas de otimização baseadas em algoritmos para determinar a melhor solução possível para o problema, chamada de *solução ótima*. Contudo, não existe um algoritmo universal

para a resolução dos modelos, mas um conjunto de métodos, entre os quais alguns são mais apropriados para determinadas aplicações (Wright & Nocedal, 1999), e a escolha da técnica de otimização irá depender, sobretudo, da qualidade e precisão da solução que se deseja determinar.

2.2.2 Otimização global e local

Um ponto x é dito *solução ótima global* da função f se $x \in D(f)$ e $f(x) \geq f(y)$ para todos os valores de y no domínio de f . Assim, a solução ótima global é o ponto cuja função objetivo atinge o menor ou maior valor entre todos os pontos da região factível, dependendo do objetivo de minimização ou maximização, respectivamente.

Um ponto x é dito *solução ótima local* da função f se $x \in D(f)$ e $f(x) \leq f(y)$ para todos os valores de y no domínio de f . Assim, a solução ótima local é o ponto cuja função objetivo é menor ou maior que todos os pontos vizinhos da região factível, dependendo do objetivo de minimização ou maximização, respectivamente.

Em muitos problemas reais a solução ótima global é desejada, mas, por serem estes problemas complexos, envolverem grande número de variáveis e instâncias de grande porte, tal solução é difícil de ser encontrada. Estes problemas são conhecidos como problemas combinatoriais, e suas principais idéias são expostas na seção a seguir.

2.2.3 Otimização Combinatória

Existe uma classe de problemas especialmente interessantes dentro da Otimização que são chamados de *Otimização Combinatória*. A principal característica destes problemas está relacionada ao espaço finito e discreto de busca de soluções, o que dificulta o tratamento até a otimalidade devido ao esforço computacional exigido em espaços suficientemente grandes.

Em muitos casos, a solução ótima destes problemas ainda está limitada

somente a pequenas instâncias, e não há metodologia comprovadamente eficaz para resolvê-los até a solução ótima global. Por isso, o grande desafio da Otimização Combinatória é produzir, em tempo computacional competitivo, soluções tão próximas quanto possíveis da solução ótima.

Muitos problemas reais se caracterizam por serem altamente combinatórios, e são neles que se concentram grande parte das pesquisas de avaliação de desempenho de algoritmos sobre instâncias de grande porte.

O termo instância refere-se a uma especificação de valores dados aos valores de entrada num determinado momento, satisfazendo às condições ou restrições próprias do problema (Goldbarg & Luna, 2005). O tamanho de uma instância está relacionado, portanto, a quantidade de variáveis de entrada do programa, independentemente dos valores associados aos parâmetros que o alimentarão.

Para Hochbaum (1996), quando a solução ideal se torna inatingível, então vale a pena sacrificar a otimalidade e se contentar com uma solução factível, que pode ser calculada de forma eficiente. Daí surge a necessidade de desenvolvimento de métodos para tratar problemas de natureza combinatorial a partir da exploração mais ampla do espaço de busca, procurando soluções, quando não ótimas, ao menos de boa qualidade, medida por alguma função pré-definida.

Santos (2009) classifica o problema de planejamento agrícola, objeto de interesse neste trabalho, como um problema de otimização combinatória complexo, pois envolve, em geral, um grande número de culturas com limitações específicas quanto à época de plantio e com períodos de cultivo e produtividade muito variáveis. Tal problema pode ser resolvido em pequenas instâncias com métodos exatos, mas para um número maior de instâncias, o recurso é fazer uso de métodos heurísticos. Na seção a seguir serão discutidas as vantagens e desvantagens de tais técnicas.

2.3 Exato *versus* Heurístico

Existem duas abordagens para resolver problemas de otimização: os métodos *Exatos* e os métodos *Heurísticos*.

Os métodos exatos são, em geral, baseados em gradientes e possuem condições de otimalidade que garantem a solução ótima. Já os métodos heurísticos são métodos aproximativos, que não garantem a convergência, mas proporcionam boas soluções em tempo computacional razoável, quando comparados com os métodos exatos.

Tradicionalmente, os métodos exatos se caracterizam pela rigidez de seus modelos matemáticos representados através de seus Teoremas, dificultando a representação de situações reais cada vez mais complexas e dinâmicas (Ochi, 1998). Nestes casos, o modelo é conhecido e sua solução depende do conhecimento das derivadas da função objetivo. Por isso, os melhores resultados destes métodos são para funções contínuas, convexas e semi-modais (Brandão & Saramago, 2011).

Muitas vezes, no mundo real, os problemas são complexos, não-lineares, de difícil representação e descritos por funções nem sempre diferenciáveis, necessitando de métodos numéricos para a sua solução (Ochi, 1998). Daí surgiu a necessidade de desenvolver técnicas capazes de proporcionar soluções ótimas ou próximas da ótima, mesmo que a qualidade não seja comprovadamente ótima.

Estes métodos se tornaram conhecidos como *heurísticas*, e trouxeram flexibilidade ao algoritmo de busca ao associar técnicas de otimização com ferramentas da Inteligência Artificial.

Por isso, a escolha do método depende das características do problema a ser otimizado, principalmente do comportamento da função que o representa (Goldbarg & Luna, 2005).

Os métodos exatos têm como vantagem a garantia da solução ótima, baixo número de avaliações da função objetivo, e em geral, convergência em tempo aceitável, dependendo do problema específico. Por outro lado, têm como desvantagem a modelagem mais complexa, alto custo computacional em problemas de grande instância, não são muito eficientes em tratar problemas em que o espaço de busca é grande e discreto e, muitos algoritmos dependem da existência de derivadas.

Também destacam-se as vantagens e desvantagens dos métodos

heurísticos. São vantagens a fácil implementação, flexibilidade do algoritmo com boa solução para a maioria dos problemas, baixo custo computacional, capacidade de encontrar ótimos globais de funções de alta complexidade e, não exigência do cálculo de derivadas. Por outro lado, são desvantagens destes métodos a não garantia da otimalidade da solução obtida, um algoritmo eficiente para um problema pode não ser eficiente para outro e, exigência de um elevado número de avaliações da função objetivo.

O cerne da dificuldade da abordagem exata está relacionada aos problemas de otimização combinatória, que representam grande parte dos problemas reais. A partir deste cenário e do avanço dos recursos computacionais, as heurísticas se tornaram uma ferramenta eficiente (rápida) para a solução de problemas de natureza combinatorial, e para comparação de solução com os métodos clássicos de otimização. A seguir, serão conceituados os elementos das técnicas heurísticas.

2.4 Estratégias Heurísticas

Etimologicamente, a palavra “heurística” vem da palavra grega *Heuriskein*, que significa “descobrir”, “encontrar”. Seu desenvolvimento se deu a partir de problemas de otimização específicos, em que as heurísticas eram desenhadas com o propósito de cada problema, e não eram, via de regra, passíveis de serem utilizadas em outros problemas. Sua principal limitação está relacionada a dificuldade de escapar de ótimos locais, por serem algoritmos de busca intensiva.

Para Melián et al. (2003), heurística é o termo apropriado para os procedimentos que, empregando conhecimento acerca de um problema e das técnicas aplicáveis, tentam encontrar soluções (ou aproximar-se delas) usando uma quantidade de recursos (geralmente tempo) razoável.

Goldbarg & Luna (2005) definem uma heurística como uma técnica que busca alcançar uma boa solução utilizando um esforço computacional considerado razoável, sendo capaz de garantir a viabilidade da solução encontrada ou, ainda, em alguns casos, a otimalidade, especialmente nas ocasiões em que essa busca partir de

uma solução viável próxima ao ótimo.

Os métodos heurísticos podem ser classificados em:

- **heurísticas de construção:** as soluções são construídas elemento a elemento, seguindo algum critério heurístico de otimização, até que se tenha uma solução viável. Ex: o método guloso;
- **heurísticas de busca em vizinhança:** a partir de uma solução inicial viável, realizam-se operações de troca na vizinhança da solução corrente, a fim de melhorar esta solução até que não seja mais possível a melhoria ou algum outro critério de parada seja satisfeito. Ex: busca local;
- **metaheurísticas:** são heurísticas genéricas mais sofisticadas, em que uma heurística mais simples é gerenciada por um procedimento que visa explorar inteligentemente a instância do problema e o seu espaço de soluções;
- **heurísticas híbridas:** é a combinação de duas ou mais heurísticas, visando melhorias na *performance* do algoritmo.

Dentre as heurísticas, as metaheurísticas são estratégias mais genéricas que, controladas por critérios probabilísticos, possibilitam escapar de ótimos locais e, portanto, explorar regiões mais promissoras.

2.4.1 Metaheurísticas

O termo “metaheurística” se obtém de antepor a palavra heurística ao prefixo meta, que significa “mais além” ou “a um nível mais elevado” (Melián et al., 2003). Elas surgiram como forma de superar as heurísticas convencionais, pois possuem estratégias para tentar escapar das armadilhas dos ótimos locais e a facilidade para se aplicar a um extenso conjunto de problemas.

Para Melián et al. (2003), as metaheurísticas podem ser concebidas como estratégias inteligentes para a concepção de procedimentos heurísticos mais generalistas e com alto desempenho.

Glover & Kochenberger (2003) definem as metaheurísticas como métodos de solução que orquestram uma interação entre os processos de melhoria e estratégias locais de nível superior para criar um processo capaz de escapar de ótimos locais e realizar uma pesquisa robusta de um espaço de solução.

A estratégia inteligente a qual as metaheurísticas são pautadas consiste basicamente em melhorar a solução ao longo do desenvolvimento do algoritmo por meio de ações de *intensificação* e *diversificação*.

Na intensificação, ao encontrar uma boa solução para o problema, realiza-se uma busca próxima à sua região, introduzindo pequenas mudanças, a fim de se obter alguma melhora. E, para não permitir convergência para ótimos locais, utiliza-se a estratégia de diversificação para encontrar outras regiões do espaço de busca que ainda não tenham sido exploradas, e que podem conter melhores soluções.

Outra característica importante a ser observada nas diferentes metaheurísticas é com relação ao número de soluções com que trabalham, que pode ser única ou uma população. As estratégias de solução única partem de uma solução inicial e descrevem movimentos no espaço de busca que consistem em sair da solução corrente para uma melhor solução ou região mais promissora. Já as estratégias baseadas em população partem de um conjunto de soluções iniciais e, por meio de procedimentos de busca definidos, descrevem a evolução da população no espaço de busca, com o objetivo de combiná-las e produzir melhores soluções.

A seguir destacam-se as principais metaheurísticas com relação ao número de soluções com que trabalham:

- **Solução única:** *Simulated Annealing*, Busca Tabu, GRASP, VNS, entre outros.
- **População:** Algoritmos Genéticos, *Scatter Search*, Colônia de Formigas, entre outros.

Em geral, as metaheurísticas apresentam as seguintes vantagens (Melián et al., 2003):

- **Simplicidade:** são baseadas em um princípio simples e claro;
- **Adaptabilidade:** são aplicáveis a qualquer tipo de problema, através de pequenas adaptações;
- **Robustez:** apresenta bons resultados a uma grande variedade de problemas.

A reunião de atributos aqui destacados, que compõem as meta-heurísticas, caracterizam-na como um bom método de busca por soluções em problemas de otimização combinatória e, por isso, o número de pesquisas utilizando estas estratégias tem crescido e se sofisticado cada vez mais, aplicando-se a espaços de busca de soluções cada vez mais complexos.

As seções que seguem descrevem os Algoritmos Genéticos e *Simulated Annealing*, estratégias que serão utilizadas neste trabalho.

2.5 Algoritmo Genético (AG)

A perspectiva evolutiva registrada por Darwin em 1859 em sua obra *A Origem das Espécies* (Darwin, 1859) revolucionou a ciência com idéias inovadoras sobre evolução e seleção natural. Gould (1977) resume a teoria darwiniana em 3 componentes, sendo estes dois fatos inegáveis e uma conclusão inevitável:

1. **Produção excessiva de descendentes:** Os organismos produzem mais descendentes do que podem sobreviver.
2. **Hereditariedade:** organismos variam, e estas variações são herdadas, pelo menos em parte, por sua descendência.
3. **Seleção Natural:** Em média, indivíduos mais adaptados são favorecidos pelo ambiente e tendem a reproduzir mais.

A condição (1) gera competição por recursos entre organismos para a sua sobrevivência e reprodução, e está condicionada a capacidade suporte do ambiente. A condição (2) está relacionada a observação de Darwin de que os organismos

variam dentro das populações e que os filhos tendem a parecer-se com os seus pais. Estas condições levam a conclusão (3), que os organismos com características mais vantajosas que os seus competidores têm mais chances de reprodução e geram maior número de descendentes, enquanto que características que não conferem nenhuma vantagem não são passadas para a geração seguinte.

Para Futuyma (2002), a evolução biológica consiste na mudança das características hereditárias de grupos de organismos, denominados populações e espécies, ao longo das gerações. Numa perspectiva de longo prazo, a Evolução é a descendência, com modificações, de diferentes linhagens a partir de ancestrais comuns.

Desde então, diversas áreas do conhecimento têm sofrido fortes influências advindas dos pensamentos Darwinianos, em especial a Matemática e a Computação, que vem registrando numerosos estudos na área de Computação Evolutiva e Inteligência Artificial.

De acordo com Brandão & Saramago (2011), estudos utilizando estratégias evolutivas em computação iniciaram-se em 1964, com os pesquisadores alemães Ingo Rechenberg e Hans-Paul Schwefel, mas foi em 1975 que as principais idéias de computação evolutiva foram solidificadas pelo cientista americano John Henry Holland, em seu livro *Adaptation in Natural and Artificial Systems* (Holland, 1975), com a criação dos Algoritmos Genéticos como estratégia para a solução de problemas numéricos.

De acordo com Mitchell (1992) e Goldberg (1989), o objetivo original de Holland não era projetar algoritmos para resolver problemas específicos, mas sim estudar formalmente o fenômeno da adaptação como ocorre na natureza e desenvolver formas em que os mecanismos de adaptação natural pudessem ser importados para os sistemas de computador.

Foi em 1989 que o cientista americano David E. Goldberg descreveu o Algoritmo Genético como um procedimento de busca baseado nos mecanismos de seleção natural e genética (Goldberg, 1989).

Desde a sua criação e formalização, os Algoritmos Genéticos têm sido muito utilizados para uma grande variedade de problemas de busca e otimização, por serem algoritmos computacionalmente simples, porém poderosos na sua busca por melhoras (Goldberg, 1989).

O Algoritmo Genético (AG) é, portanto, um procedimento de busca que utiliza escolhas aleatórias como ferramenta para guiar a busca altamente exploratória por meio da codificação de um espaço de parâmetros em direção a regiões do espaço de busca com provável melhora (Goldberg, 1989).

O AG baseia-se na representação de um problema por meio de um conjunto de indivíduos que são soluções potenciais para o problema em questão, chamada de *população*. Cada indivíduo da população é denominado *cromossomo* e possui uma *aptidão*, que é a sua capacidade de sobreviver e reproduzir no ambiente.

Em sua abordagem mais geral, inicia-se o algoritmo construindo uma *população inicial*, que pode ser gerada de forma aleatória ou por uma heurística construtiva. Em seguida, todos os indivíduos da população têm o seu desempenho avaliado por uma *função de avaliação*. Esta população passará então por processos consecutivos de reprodução, por meio de operações de *seleção*, *crossover* e *mutação*. Cada ciclo de reprodução determina uma nova população de indivíduos, chamada de *geração*, e o processo continua até que um *critério de parada* seja satisfeito.

O processo ilustrado na Figura (2) busca uma melhor solução para um determinado problema por meio da evolução das populações. Após um certo número de gerações, espera-se convergir para uma geração de elite, que corresponde a uma solução ótima ou quase ótima para o problema.

Goldbarg & Luna (2005) resumem os procedimentos básicos de um AG em:

1. Representação das soluções na estrutura de cromossomos;
2. Construção de uma população inicial de cromossomos;
3. Definição de uma função de avaliação dos cromossomos segundo suas aptidões;

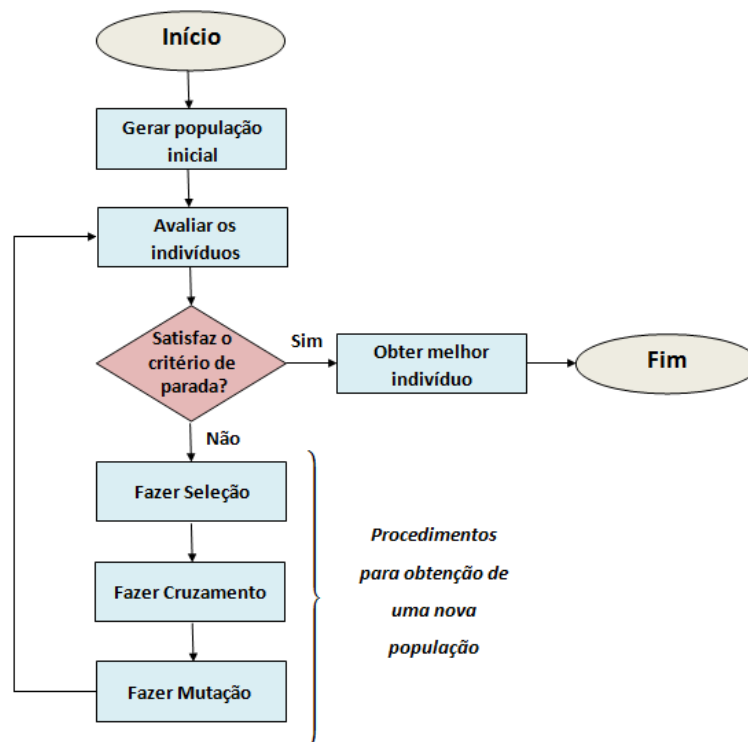


Figura 2 - Fluxograma do processo de busca do AG em uma abordagem geral

4. Definição dos operadores genéticos que vão permitir a produção de novos indivíduos;
5. Definição dos parâmetros do AG, tais como critério de parada, tamanho da população, entre outros.

2.5.1 Representação de um cromossomo

Para trabalhar com o Algoritmo Genético é preciso primeiramente codificar as informações do problema em uma estrutura denominada *cromossomo*, de modo que o método consiga trabalhar com os dados que lhes são fornecidos (Brandão & Saramago, 2011). Esta codificação é, na verdade, a representação de uma solução.

Para Koza (1992), a representação é um passo fundamental no trabalho com o AG, pois ele manipula diretamente a representação codificada do problema, e o esquema de representação pode limitar severamente a janela pela qual o sistema

observa o mundo.

Na literatura, a codificação das variáveis para o AG é usualmente feita por meio de uma representação binária (zero-um) em que cada valor associado representa a presença (valor 1) ou ausência (valor 0) de certa característica no indivíduo.

Embora esta representação tenha se mostrado eficiente para vários problemas, observou-se, à medida que foram crescendo as aplicações dos AGs em problemas com um elevado número de restrições, que esta representação pode não ser a mais adequada, surgindo daí alternativas como a representação por números reais ou inteiros, em que um cromossomo é descrito por um vetor de números reais ou inteiros, dependendo do problema (Koza, 1992).

A título de ilustração, vamos considerar um exemplo simples:

Problema do Agricultor: Suponha que um agricultor dispõe de cinco tipos de plantas e apenas três canteiros disponíveis para plantio. Deseja-se saber qual a melhor combinação de plantas nos canteiros, a fim de minimizar os custos do agricultor com o plantio.

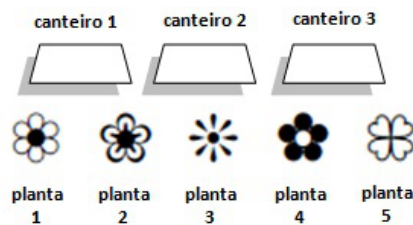


Figura 3 - Itens que compõem o problema do agricultor

Um cromossomo com representação binária para o problema do agricultor pode ser definido como mostra a Figura (4).

Na Figura (4), a combinação de 15 bits representa um cromossomo para o problema do agricultor, ou seja, uma possível solução. Cada gene X_i , $i = 1, 2, 3$, representa um canteiro e cada alelo de X_i representa uma planta. Desta forma, o primeiro alelo de X_1 pode assumir o valor 1 ou 0, que representará se a planta 1

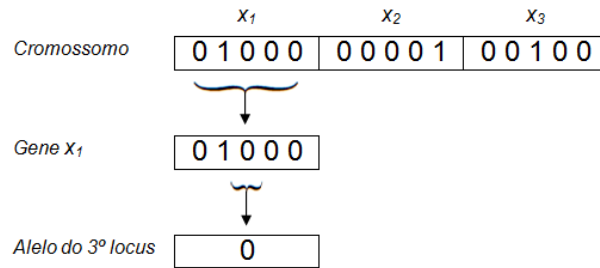


Figura 4 - Cromossomo com representação binária

será cultivada ou não naquele canteiro, respectivamente. Por isso, a posição do alelo dentro do gene é fundamental para a interpretação da solução.

Observe ainda que, para a solução ser factível, cada gene X_i deve conter em sua configuração apenas um bit de valor 1, pois ele associa ao canteiro uma única planta a ser cultivada. Do contrário, implicaria que mais de uma planta pode ser plantada em um determinado canteiro, o que não é permitido no problema. Portanto, na configuração ilustrada na Figura (4), o agricultor deverá plantar nos canteiros 1, 2, e 3 as plantas 2, 5 e 3, respectivamente.

A Figura (5) utiliza a representação com números inteiros sugerida em Koza (1992) para a mesma configuração do problema do agricultor.

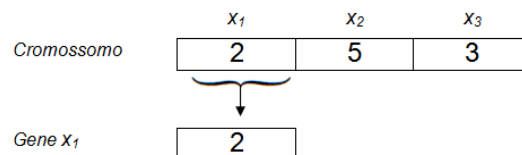


Figura 5 - Cromossomo com representação inteira

Nota-se que representação inteira para o problema do agricultor torna-se mais simplificada do que a binária, pois não necessita da posição do alelo indicando a planta que deve ser cultivada, e cada gene X_i é composto por apenas um valor, que especifica a cultura que deve ser plantada naquele canteiro.

Michalewicz (1994) defende o uso de estruturas adequadas de dados (possivelmente complexos) para a representação do cromossomo, e que este não pre-

cisa necessariamente ser representado por uma cadeia binária. Desta forma, a representação deve ser aquela que melhor represente o problema em questão e deve estar corretamente associada a soluções válidas do problema analisado.

2.5.2 População

Com a representação de um indivíduo definida, o primeiro passo para inicialização do AG é criar uma população inicial. É um processo, na maioria das vezes simples, que implica em criar Q indivíduos que irão compor a primeira geração ($g = 1$) do AG. Após a criação da população inicial, a população estará em constante fluxo nas próximas gerações devido a reprodução e morte dos indivíduos que a compõem.

Comumente, a população inicial é gerada de forma aleatória, ou seja, cada alelo pode conter um dos possíveis valores do conjunto adotado (binário, inteiro, real, ...), com probabilidade uniforme. Para o problema do agricultor, gerou-se uma população inicial com Q indivíduos na estrutura de cromossomos com representação inteira em que cada indivíduo é expresso por um cromossomo de 3 genes, e cada gene contém um número inteiro escolhido aleatoriamente entre 1 e 5, que é a identificação das plantas disponíveis. Suponha $Q = 5$, obteve-se aleatoriamente para o problema do agricultor os cromossomos apresentados na Figura (6).

s_1	2	5	3
s_2	1	1	4
s_3	5	3	5
s_4	3	4	2
s_5	5	2	1

Figura 6 - População inicial gerada aleatoriamente para o problema do agricultor

Estudos utilizando técnicas heurísticas para construção da população inicial têm constatado que a inicialização não aleatória desta população pode acelerar

a convergência do AG (Moujahid et al., 2007). Assim, para problemas em que a geração de uma população inicial não é tão imediata e o espaço de soluções factíveis é muito restrito, utilizam-se heurísticas simples e rápidas para esta tarefa.

Além da população inicial, o Algoritmo Genético trabalha com um população intermediária (*mating pool*), que é obtida a partir da população inicial e por meio da operação de seleção.

O número de indivíduos Y na população intermediária é determinado pela taxa de seleção p_s definida pelo programador, de modo que $1 \leq Y \leq Q$, em que Q é o número de indivíduos na população inicial. Os Y indivíduos são escolhidos utilizando-se algum método de seleção e participarão do processo de reprodução da população.

A ideia da população intermediária é permitir que uma parte dos indivíduos da geração atual, aqueles que não foram selecionados para a reprodução, sejam mantidos na próxima geração, preservando assim características da população atual na nova população após o processo de reprodução.

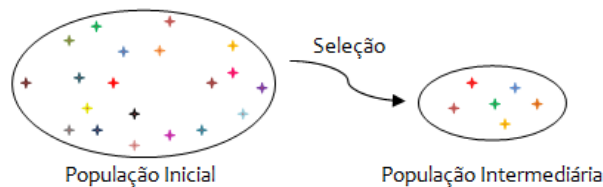


Figura 7 - Representação esquemática da população inicial e população intermediária

Observe que quando $Y = 1$ ocorre reprodução assexuada, ou seja, há apenas um indivíduo na população intermediária para fazer cruzamentos. Neste caso, não haverá variação genética. Para $Y = Q$ tem-se que a população intermediária é a própria população inicial. Neste caso, não há garantias de manter indivíduos da geração atual na próxima geração, pois todos os indivíduos estão susceptíveis a participar do cruzamento e serem substituídos por indivíduos mais aptos.

2.5.3 Aptidão

Aptidão é a essência da seleção natural de Darwin e igualmente dos Algoritmos Genéticos. Na natureza, a aptidão de um indivíduo é a sua probabilidade de sobreviver até a idade de reprodução, e reproduzir-se. Esta medida pode ser obtida considerando-se, por exemplo, o número de filhos.

No mundo artificial dos algoritmos matemáticos, em geral, mede-se a aptidão dos indivíduos utilizando alguma função representativa do problema e, em seguida, utiliza-se essa medida para controlar a aplicação das operações que modificam as estruturas da população artificial (Koza, 1992). A forma mais comum de medir esta aptidão nos AGs é construindo uma função de avaliação que atribui um valor de aptidão a cada indivíduo da população. Essa função deve ser definida para cada problema de maneira específica e será denotada por $eval(S_i)$. Assim, dado um cromossomo S_i , a função de avaliação $eval(S_i)$ lhe designa um número real que supostamente reflete o grau de adaptação desse indivíduo ao problema (Moujahid et al., 2007).

Para o problema do agricultor, a aptidão de um dado cromossomo poderá ser, por exemplo, o custo total do cultivo de determinadas plantas nos canteiros. Conhecido os custos de cultivar cada planta em cada canteiro, é possível obter valores para a função de avaliação. A Tabela (1) mostra, dentre as soluções correntes, que S_1 é a solução viável para o problema, pois apresenta menor custo.

Tabela 1: Aptidão das soluções geradas para o problema do agricultor

Cromossomo	Solução	Aptidão
S_1	3 – 5 – 1	3
S_2	5 – 2 – 4	16
S_3	2 – 1 – 2	6
S_4	5 – 3 – 4	13
S_5	4 – 4 – 5	10

A aptidão de um indivíduo está fortemente relacionada com a sua in-

fluência sobre o desenvolvimento da população. Quando muitos descendentes de um único indivíduo sobrevivem e reproduzem-se, então muitos membros da população resultante, a chamada “próxima geração”, levarão alelos deste indivíduo (Holland, 1975).

2.5.4 Penalização

Observou-se na Seção 2.5.3 que uma boa função de avaliação é aquela que reflete uma aptidão real para cada indivíduo. Contudo, em muitos problemas de otimização combinatória, boa parte das soluções representadas na população do AG são não factíveis, ou seja, não satisfazem todas as restrições do problema. Estas soluções podem muitas vezes apresentar melhores valores de avaliação do que soluções factíveis, e daí surge a necessidade de penalizá-las, de modo que o algoritmo não convirja para uma solução que não satisfaça as exigências do problema.

Nestes casos, os indivíduos estão sujeitos a um conjunto de restrições, tornando necessária a utilização de estratégias para melhorar o desempenho do Algoritmo Genético, e dentre elas destacam-se a *absolutista* e a *penalização*.

Na abordagem absolutista os indivíduos que não satisfazem as restrições não são considerados soluções para o problema, mas seguem fazendo cruzamentos e mutações para a obtenção de indivíduos factíveis. Na abordagem de penalização, atribui-se um valor de penalização para cada indivíduo que infringe restrições. A idéia geral da penalização é diminuir a aptidão do indivíduo (ou aumentar, dependendo do objetivo do problema) em um valor que diz respeito às restrições que o indivíduo viola.

Desta forma, a função objetivo de um problema com penalização pode ser escrita como segue:

$$f(S_i) = \begin{cases} eval(S_i), & \text{se } S_i \text{ é factível} \\ eval(S_i) \pm penalidade(S_i), & \text{caso contrário} \end{cases},$$

em que S_i é o indivíduo, $i = 1, \dots, Q$, *eval* é a função de avaliação e *penalidade* é a função de penalização, que vale zero se nenhuma violação ocorre, e é um valor positivo, caso contrário.

Linden (2008) destaca três formas de penalização: *estática*, *dinâmica* e *adaptativa*. A penalização estática é definida a priori e não se altera com a execução do algoritmo. Na penalização dinâmica, define-se uma função para a evolução da punição com o tempo, aumentado-a com o decorrer do algoritmo. Já a adaptativa, ao invés de usar uma função para a penalização, utiliza a avaliação da população corrente como um *feedback* para guiar a modificação da penalização.

A estratégia de penalização, de modo geral, deve garantir que o pior indivíduo que respeita as restrições tenha avaliação superior àquela do melhor indivíduo não factível. Espera-se que, com a penalização das soluções não factíveis, o algoritmo se aproxime cada vez mais da região viável do problema restrito, haja visto que as soluções fora do espaço viável apresentarão valores de aptidão muito ruins para a otimização do problema e serão facilmente desconsideradas pelo algoritmo.

2.5.5 Elitismo

O elitismo é um operador que força o Algoritmo Genético a preservar um número fixo de melhores indivíduos factíveis na população em cada geração. A ideia é que os melhores indivíduos da população não sejam perdidos durante o processo de reprodução, e que a diversificação gerada pelo algoritmo aproveite informações destas soluções.

Um Algoritmo Genético com Elitismo copia os melhores indivíduos da população em toda geração, assegurando que durante a reprodução estes indivíduos não sejam perdidos, e ao final do processo, os devolvem à população. A Figura (8) mostra o funcionamento do AG com Elitismo.

Na literatura, o elitismo tem mostrado melhoras significantes no desempenho do AG (Mitchell, 1992). Contudo, há críticas em sua utilização por causar decréscimo na diversidade da população por meio do aumento da pressão seletiva

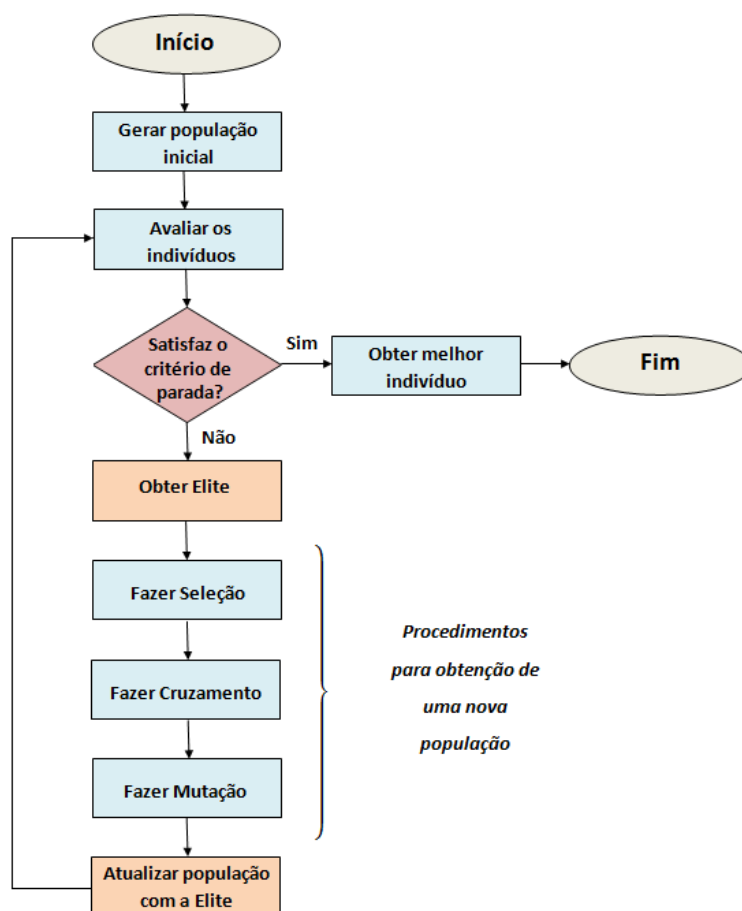


Figura 8 - Fluxograma do Algoritmo Genético com elitismo

(Ishibuchi et al., 2007).

2.5.6 Seleção

Antes de operar geneticamente para obter a reprodução da população, deve-se selecionar os indivíduos que participarão deste processo, e isto é feito por meio do operador de *seleção*. Seleção é, portanto, um método que copia cromossomos da população corrente para a população intermediária, a fim de se reproduzirem. Em geral, a seleção privilegia os indivíduos mais aptos, ou seja, quanto maior for a aptidão do indivíduo, maior a sua chance de ser selecionado para a população intermediária.

Na literatura existem 4 principais formas de seleção: *Proporcional*,

Torneio, Classificação e Truncada. A escolha do método pode influenciar diretamente o desempenho do algoritmo, pois produzem mudanças no comportamento das soluções, induzindo uma pressão de seleção diferente na população (De Jong & Sarma, 1995).

Dentre os métodos citados, a Seleção Proporcional (*Fitness-Proportional Selection*) foi pioneira e é a que dá maior chance de indivíduos mais aptos serem selecionados. Contudo, em casos de existência de super-indivíduos na população (indivíduos com aptidão muito superior a dos demais), pode-se obter convergência prematura para um ótimo local, já que a reprodução descomedida deste indivíduo ao longo das gerações pode acabar com a diversidade da população.

Pensando nisto, métodos de seleção que diminuam a dominância do super-indivíduo e solucionem o problema da convergência prematura foram propostos. Estes métodos visam diminuir a pressão seletiva na população, dando mais chances para indivíduos menos aptos serem selecionados para a reprodução. Por um lado, nestes métodos, se ganha com diversidade da população, mas por outro lado, faz com que o algoritmo chegue mais lentamente à solução desejada.

Linden (2008) destaca alguns pontos em que o método de seleção utilizado pode influenciar no desempenho do algoritmo:

1. Pode-se acelerar ou retardar a ocorrência da convergência genética;
2. Fica mais ou menos agressivo no aproveitamento das melhores soluções;
3. Ao utilizar apenas indivíduos com excelentes avaliações pode-se estar jogando fora bons esquemas presentes nos indivíduos com avaliações ruins;
4. Ao permitir muito que indivíduos com avaliações ruins participem do processo reprodutivo, os esquemas que os tornam ruins não desaparecerão da população.

2.5.6.1 Proporcional

A implementação da Seleção Proporcional é usualmente feita utilizando um dos métodos: *Roleta Viciada* ou *Amostragem Estocástica Uniforme*.

A Roleta Viciada (*Roulette Wheel*) é a mais popular forma de implementação da Seleção Proporcional à aptidão do indivíduo. Neste método, a chance de um indivíduo ser selecionado está proporcionalmente ligada à sua aptidão, e em geral, os indivíduos são selecionados aleatoriamente, dando maior chance de reprodução àqueles mais aptos. Koza (1992) descreve o procedimento usual da Roleta Viciada como segue.

Se $f(S_i)$ é a aptidão do indivíduo S_i da geração corrente, então, neste método de seleção, a chance do indivíduo S_i ser copiado para a população intermediária é dada pela Equação (1).

$$p_i = \frac{f(S_i)}{\sum_{j=1}^Q f(S_j)}, \quad i = 1, \dots, Q. \quad (1)$$

Tipicamente, p_i é a aptidão normalizada do indivíduo S_i . Define-se uma função q_i que acumula o valor das aptidões de todos os indivíduos da geração corrente, dada pela Equação (2).

$$q_i = \sum_{j=1}^Q p_j, \quad i = 1, \dots, Q, \quad (2)$$

em que Q é o número de indivíduos da população e $0 \leq q_i \leq 1$, com $i = 1, \dots, Q$.

A seleção de um indivíduo por Roleta Viciada consiste em gerar um número aleatório r uniformemente distribuído no intervalo $[0, 1]$ e compará-lo com a função q_i . Quando $p_{i-1} \leq r \leq p_i$, indica que o indivíduo S_i foi selecionado. Repete-se o processo por Y vezes, o número de indivíduos da população intermediária determinada pela taxa p_s .

Para a população inicial criada no exemplo do agricultor, disponível na Tabela (1), suponha que deseja-se determinar uma população intermediária de $Y = 3$ indivíduos utilizando o método da Roleta Viciada, como mostra a Tabela (2).

Tabela 2: Valores de aptidão normalizada e acumulada para os indivíduos gerados no problema do agricultor

Indivíduo	p_i	q_i
S_1	0,0625	0,0625
S_2	0,3333	0,3958
S_3	0,1250	0,5208
S_4	0,2708	0,7917
S_5	0,2083	1

A partir das informações da Tabela (2), pode-se construir a Roleta Viciada com aptidão normalizada, como mostra a Figura (9), e prosseguir com o método, descrito a seguir.

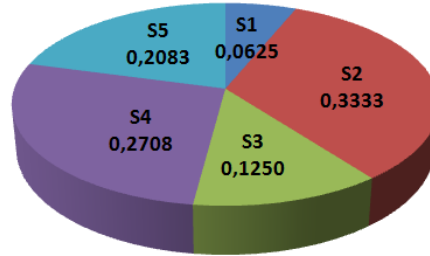


Figura 9 - Roleta com aptidão normalizada para a população inicial gerada no problema do agricultor

Sorteia-se um valor para r pertencente ao intervalo $[0, 1]$, e verifica-se, por meio da função acumulada, qual indivíduo foi selecionado pelo método para compor a população intermediária. Por exemplo, se $0 \leq r \leq 0,0625$, o indivíduo S_1 será selecionado, ou se $0,0625 < r \leq 0,3958$, o indivíduo S_2 será selecionado, e assim por diante. O indivíduo selecionado é então retirado da Roleta Viciada e uma nova

roleta é construída com os indivíduos restantes. Este processo é repetido Y vezes, que no exemplo ilustrado equivale a três vezes.

O método da Roleta Viciada tem sido muito utilizado na literatura, porém, como destacado, pode apresentar problemas de rápida convergência devido a existência de super-indivíduos na população, que muitas vezes representa um ótimo local, e não global, como é de interesse.

Outro método de seleção que utiliza a idéia de seleção Proporcional à aptidão é a Amostragem Estocástica Uniforme (*Stochastic Universal Sampling*). Neste método, todos os indivíduos são mapeados para segmentos contíguos de uma linha, sendo que o tamanho de cada segmento é proporcional ao valor da aptidão do indivíduo (Linden, 2008).

O primeiro passo é fazer uma normalização dos tamanhos dos segmentos, como no método da Roleta Viciada, expressa na Equação (1). Em seguida, para selecionar os Y indivíduos que irão compor a população intermediária, sorteia-se um número r pertencente ao intervalo $[0, \frac{1}{Y}]$ e constrói-se Y ponteiros, como mostra a Equação (3).

$$r, r + \frac{1}{Y}, r + \frac{2}{Y}, \dots, r + \frac{Y-1}{Y} \quad (3)$$

Estes ponteiros apontam os segmentos de reta normalizados selecionados pelo método e indicam que os indivíduos “donos” destes segmentos serão os escolhidos para a população intermediária.

Suponha que deseja-se selecionar $Y = 3$ indivíduos para compor a população intermediária no problema do agricultor, e que sorteu-se aleatoriamente r igual a 0,2, sabendo que $r \in [0, \frac{1}{3}]$. Então, os indivíduos selecionados pelos ponteiros é mostrado na Figura (10).

Observe na Figura (10) que os ponteiros construídos pelo método indicam que S_2 , S_4 e S_5 foram os indivíduos selecionados.

Linden (2008) destaca ainda que, assim como no método da Roleta Viciada, a Amostragem Estocástica Uniforme ainda não resolve o problema da con-

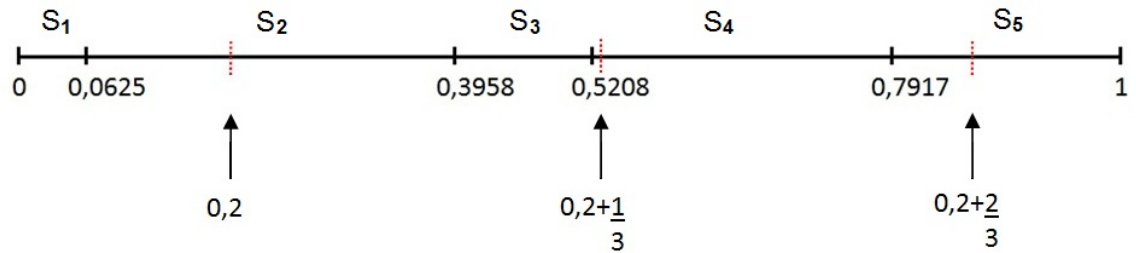


Figura 10 - Ponteiros construídos pelo método de Amostragem Estocástica Uniforme para o problema do agricultor

vergência para super-indivíduos.

2.5.6.2 Torneio

Torneio (*tournament selection*) é um método de seleção em que k indivíduos são escolhidos aleatoriamente da população e apenas um será selecionado para a população intermediária, aquele mais apto entre os k .

Desta forma, um número aleatório r é escolhido uniformemente no intervalo $[0, 1]$. Se $r < b$ (em que b é um parâmetro, por exemplo, $0,9$), o indivíduo mais apto entre os k será selecionado para ser um dos pais, caso contrário, o menos apto será selecionado. Os k indivíduos são então devolvidos na população original e podem ser selecionados novamente (Mitchell, 1992).

A forma mais usual de implementação da seleção por torneio é o chamado *torneio binário*, em que dois indivíduos concorrem na seleção. Neste caso, $k = 2$ e não é utilizado o parâmetro b para testar o vencedor do torneio, e isso implica em colocar os indivíduos em competição direta e o mais apto vence sempre.

A implementação da seleção por torneio binário é simples e é descrita a seguir:

1. Seleciona-se 2 indivíduos da população aleatoriamente para participar do torneio;

2. Os 2 indivíduos são colocados em competição direta pelo direito de ser pai, usando como arma a sua aptidão;
3. O indivíduo mais apto entre os 2 é copiado para a população intermediária, ficando disponível para a reprodução.

Observe que o torneio de tamanho k implica em quantos competidores serão selecionados aleatoriamente dentro da população para participar do torneio e é definido pelo programador. Assim, k deve ser no mínimo igual a 2 para poder haver competição entre indivíduos e no máximo igual ao tamanho da população Q , sendo que, neste segundo caso, o vencedor será sempre o mesmo, o indivíduo mais apto.

Vale ainda notar que, quando o parâmetro b não é utilizado, o pior indivíduo nunca participará da reprodução, pois ele nunca será vencedor de um torneio, exceto em raros casos de concorrer ele com ele mesmo.

2.5.6.3 Classificação

A implementação da Seleção por Classificação, assim como a Seleção Proporcional, é feita utilizando um dos métodos: *Linear* ou *Exponencial*.

No método de seleção por Classificação (*rank selection*) os indivíduos da população não serão selecionados de acordo com sua aptidão absoluta, como na seleção Proporcional, e sim *classificados* de acordo com sua aptidão (Mitchell, 1992). A partir de sua classificação, terão os seus valores de desempenho recalculado por meio de uma transformação, que pode ser linear ou exponencial.

Este método foi proposto por Baker em 1985, como forma de superar a convergência prematura observada nos métodos de Seleção Proporcional (Mitchell, 1992), e é descrito como segue.

Primeiramente, ordena-se todos os indivíduos da população de acordo com a sua aptidão, estabelecendo assim uma classificação dos indivíduos em ordem crescente (o pior terá classificação igual a 1, o segundo pior 2, até o último que terá a melhor classificação igual a Q , que é número de indivíduos na população). Em

seguida, atribui-se a cada indivíduo um valor de adequação determinado por sua posição na classificação, que pode ser obtida pela Equação (4).

$$E(i, t) = Min + (Max - Min) \frac{(rank(i, t) - 1)}{Q - 1}, \quad (4)$$

em que,

- $E(i, t)$ é o valor de adequação que deseja-se calcular para o indivíduo i da geração t ;
- Min é o valor da avaliação que será atribuído ao indivíduo pior classificado;
- Max é o valor da avaliação que será atribuído ao indivíduo melhor classificado;
- Q é o número de indivíduos na população;
- $rank(i, t)$ é a classificação atribuída ao indivíduo i na geração t .

Esta forma de classificação é chamada de *classificação linear*. Com o valor E de adequação de cada indivíduo estabelecido pela Equação (4), pode-se utilizar um dos métodos descritos anteriormente (Roleta Viciada ou Amostragem Estocástica Uniforme) para determinação da população intermediária.

Linden (2008) propõe ainda uma maneira de manter a pressão seletiva em um nível mais alto, utilizando uma função de classificação exponencial, denominada *classificação exponencial*.

$$E(i) = \frac{1 - e^{-i}}{c}, \quad (5)$$

em que i é a posição do indivíduo (do pior para o melhor) e c é uma constante.

O uso da função exponencial aumenta a diferença entre a avaliação do melhor indivíduo e a do pior, aumentando a pressão seletiva e, por conseguinte, diminuindo a diversidade da população. Assim, o problema do super-indivíduo fica imediatamente resolvido e não se tem tanta homogeneidade nas avaliações quanto no caso de uso de uma função linear (Linden, 2008).

Koza (1992) observa as vantagens e desvantagens do método de seleção por classificação:

Seleção por classificação reduz potencialmente os efeitos dominantes dos super-indivíduos na população por distribuir melhor a chance de sorteio entre os indivíduos, limitando assim a pressão de seleção. Ao mesmo tempo, este método de seleção exagera na aproximação dos valores de adequação entre indivíduos e super-indivíduos, de modo que os melhores cromossomos não se distinguem muito dos demais.

2.5.6.4 Truncada

No método de Seleção Truncada (*truncation selection*) apenas os x melhores indivíduos da população serão escolhidos para a reprodução. O parâmetro x pertence ao intervalo $[1, Q]$, em que Q é o número de indivíduos da população. Para $x = 1$, a reprodução torna-se monótona, ou seja, apenas um indivíduo participará da reprodução, e isso implica que todos os filhos serão iguais ao pai. Para $x = Q$, a versão truncada passa a se comportar como a não truncada, e todos os indivíduos da população estão disponíveis para a reprodução. Neste último caso, a população intermediária é a própria população.

Para a implementação do método, os indivíduos são ordenados de forma decrescente (de Q a 1) de acordo com sua avaliação, e somente aqueles cujas posições estiverem entre 1 e a posição de corte x poderão participar da seleção. Fica claro então que este método de seleção favorece a reprodução dos indivíduos mais aptos, causando maior pressão seletiva do que todos os outros métodos.

Linden (2008) explica que “a seleção truncada permite que o AG se concentre somente nas melhores soluções da população, permitindo que se chegue a uma boa solução de forma mais rápida”. Contudo, por causar maior perda da diversidade, faz com que o AG apresente o pior desempenho e sugere que seja utilizada em uma estratégia híbrida, no começo da evolução do AG, quando a variedade de indivíduos ainda é grande.

2.5.7 Operadores Genéticos

Os operadores genéticos representam o núcleo de um AG. O objetivo básico de um operador genético é produzir cromossomos que possuam propriedades genéticas superiores às encontradas nos cromossomos anteriores (Ochi, 1998).

No Algoritmo Genético, os operadores genéticos convencionais são a *mutação* e o *crossover*. Contudo, existem outras formas não convencionais de obter variação genética, tais como: inversão, migração e epidemia. Enquanto os operadores genéticos convencionais são utilizados em todas as gerações do AG, os operadores não convencionais podem ser acionados ocasionalmente, como artifício para melhorar a diversidade genética da população em uma geração específica ou por um determinado número de gerações.

Em geral, quando os indivíduos de uma população passam a ter cromossomos muito semelhantes, e isso pode ser medido comparando-se as suas aptidões, possivelmente o algoritmo estará convergindo para um ótimo local, e então a utilização de operadores não convencionais pode ser interessante.

2.5.7.1 *Crossover*

O operador genético *crossover* (ou cruzamento) é a fonte de quase toda a variação genética do AG. Ele permite que novos indivíduos sejam criados e, portanto, que novos pontos no espaço de busca sejam testados (Koza, 1992).

O *crossover* inicia-se escolhendo aleatoriamente dois pais da população intermediária. Em seguida, escolhe-se também aleatoriamente uma posição de corte k nos cromossomos pais e troca-se as subsequências antes e depois deste local, criando-se dois filhos. Esta posição de corte k determina a partir de qual *locus gênico*¹ o *crossover* será efetuado. No exemplo do agricultor com representação inteira, suponha alguns pais selecionados para o cruzamento, como mostra a Figura (11).

Observe na Figura (11) que existem $(lg - 1)$ posições de corte possíveis

¹Lócus gênico representa o local ocupado pelo gene no cromossomo.

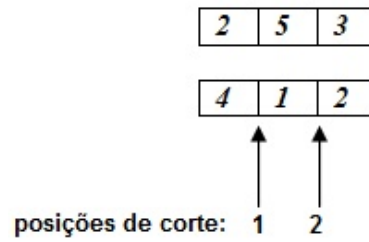


Figura 11 - Posições de corte possíveis no cromossomo com representação inteira

para se fazer o cruzamento entre os pais selecionados, em que lg é o número de locos gênico que compõem o cromossomo. Se o ponto de corte selecionado for o ponto 1, o cruzamento produzirá os filhos mostrados na Figura (12).

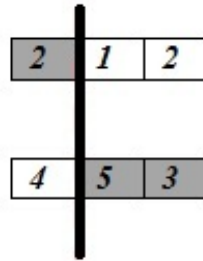


Figura 12 - Troca de genes entre cromossomos pais realizados a partir do ponto de corte selecionado

A operação de cruzamento produzirá, portanto, dois filhos. Estes filhos, em geral, são diferentes de seus pais e diferentes entre si. Contudo, cada filho contém algum material genético herdado de seus genitores.

A quantidade de cromossomos a serem submetidos ao processo de cruzamento é definida pela taxa de crossover p_c , especificada pelo usuário e será discutido na Seção 2.5.8.

Podemos explorar algumas variações do operador *crossover*, tais como:

- Monogamia ou Poligamia;
- Sexuado ou Assexuado;

- Um-ponto, n -pontos e uniforme.

Quando estabelecemos apenas uma posição de corte nos cromossomos pais, tem-se a estratégia denominada *crossover 1-ponto* (*single-point crossover*). Pode-se também escolher mais de uma posição de corte, e neste caso, tem-se a estratégia denominada *crossover n -pontos* (*multi-point crossover*). No *crossover n -pontos* são escolhidas n posições de corte aleatoriamente ($1 < n < gl$, em que gl é o número de loci nos genes), e em seguida realizam-se trocas alternadas de sequências de genes determinadas pelas posições de cortes. A Figura (13) ilustra um *crossover 2-pontos*.

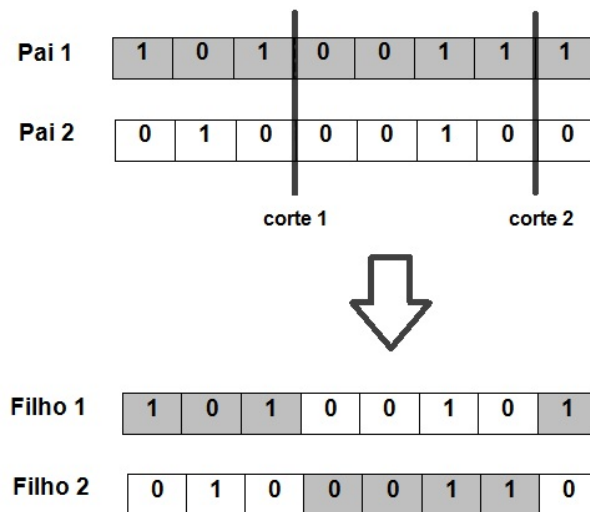


Figura 13 - Troca de genes entre cromossomos pais realizados com *crossover 2-pontos*

Outra possibilidade é, ao invés de trocar segmentos determinados pelas posições de cortes, criar uma máscara de *crossover*, operação chamada de *crossover uniforme*. Desta forma, para cada gene a ser preenchido nos cromossomos filhos, o operador de cruzamento uniforme sorteia de qual dos pais este deve ser copiado. A máscara de cruzamento é uma sequência binária de tamanho G , em que cada valor associado representa a troca de genes (valor 1) ou não (valor 0). A Figura (14) ilustra um exemplo de *crossover* uniforme.

Na monogamia, dois pais são escolhidos da população intermediária

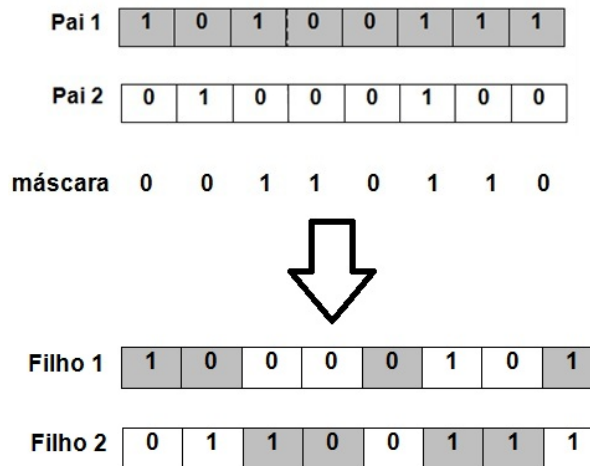


Figura 14 - Troca de genes entre cromossomos pais realizados com *crossover* uniforme

para a reprodução apenas uma vez. Isso significa que, após o cruzamento, são retirados da população intermediária e não podem ser sorteados novamente. Nesta forma de reprodução, a população intermediária deve conter obrigatoriamente um número par de indivíduos e no mínimo $2 \times p_c \times Q$ indivíduos (em que $p_c \times Q$ é o número de cruzamentos a serem realizados).

No caso da poligamia, o número de indivíduos na população intermediária é independente de ser par e da taxa de cruzamento p_c . A cada cruzamento entre dois indivíduos, eles retornam à população intermediária, podendo ser selecionados novamente para cruzar com outros cromossomos.

O *crossover* sexuado é aquele que força o algoritmo a escolher 2 pais distintos para a reprodução. Do contrário, quando permitimos o cruzamento utilizando o mesmo pai, estamos permitindo o *crossover* assexuado, e que produzirá 2 filhos idênticos ao pai. Esta segunda forma de *crossover* não traz variação genética para a nova população, apenas conduz o algoritmo genético a agir de forma diferente da primeira. Isto pode ser definido na implementação do algoritmo.

Assim, estas características podem ser combinadas na implementação do algoritmo, como por exemplo: *Crossover 1-ponto* e com Poligamia, ou *Crossover*

3-pontos e com Monogamia.

2.5.7.2 Mutação

Como visto, o operador *crossover* causa variação nos indivíduos da população fazendo recombinação dos genes dos cromossomos pais. Contudo, se não houver diversificação de cromossomos além daqueles já criados e testados, o algoritmo genético pode convergir para um ótimo local e dali não conseguir escapar. Por isso, o operador de mutação é fundamental para um AG. É ele que garante a continuidade da existência de diversidade genética na população, enquanto o operador de *crossover* contribui fortemente para a igualdade entre os indivíduos (Linden, 2008).

O operador de mutação força o algoritmo a explorar outras áreas da região de busca por introduzir mudanças aleatórias, ou mutações, nos cromossomos (Brandão & Saramago, 2011).

De modo geral, a mutação altera um ou mais genes de um cromossomo de acordo com uma função de probabilidade, e a frequência de aplicação da operação de mutação é controlada por um parâmetro chamado *probabilidade de mutação*, denotado por p_m (Koza, 1992).

Para todo indivíduo da população intermediária, sorteia-se aleatoriamente um valor r uniformemente distribuído no intervalo $[0, 1]$ e verifica-se se este sofrerá mutação comparando r com p_m . Para os indivíduos que sofrerão mutação (quando $r < p_m$), escolhe-se aleatoriamente um locus do cromossomo (dentre os L existentes) como posição de mutação. O alelo contido neste locus será modificado. Se a representação é binária, o alelo é apenas complementado (Koza, 1992).

Em casos em que a representação do cromossomo é inteira, deve-se escolher aleatoriamente um novo alelo, diferente daquele existente, para ocupar o locus selecionado.

Na Figura (15), selecionou-se aleatoriamente o locus 2 para a mutação. Como a representação utilizada é binária, inverte-se o bit 0 para 1, originando um

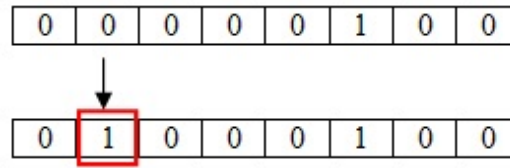


Figura 15 - Mutação aleatória em cromossomo com representação binária

novo indivíduo.

Diferentemente do *crossover*, a mutação é uma operação assexuada e que opera em apenas um indivíduo. Além disso, o operador mutação pode agir sobre uma solução factível e torná-la não factível, ou vice e versa, por isso, é utilizado com muita moderação nos AGs.

2.5.7.3 Epidemia

Epidemia, também chamada de Eliminação, é uma operação que implica em, numa determinada geração, eliminar uma porcentagem X de indivíduos menos aptos da população e gerar novos indivíduos para ocupar seus lugares na população.

A gravidade da epidemia é quem determina a porcentagem X de indivíduos que serão eliminados, podendo chegar até a 100%, e neste caso, toda a população atual será eliminada e substituída por uma nova.

Observe que a epidemia, em geral, só terá efeito se utilizada junto com o Elitismo, em especial quando $X = 100\%$, pois acionar uma epidemia desta magnitude na g -ésima geração implica em dar início a um novo AG de apenas $(G - g)$ gerações (em que G é o número total de gerações e $1 \leq g \leq G$).

2.5.8 Parâmetros

Outra decisão importante a ser feita na implementação de um AG é com relação a definição dos valores para os principais parâmetros. São eles: tamanho

da população Q , taxa de seleção p_s , taxa de crossover p_c , probabilidade de mutação p_m e número de gerações G .

O tamanho da população Q determina o número de pontos no espaço de busca que será explorado por geração no AG. Pode ser um número fixo ou variável, controlado por alguma função. Reduzir o tamanho da população pode ocasionar convergência para um ótimo local ou a convergência mais lenta para um ótimo global. Por outro lado, maior população implica em maior custo computacional. Estas ideias também se estendem ao parâmetro G , que determina o número de gerações do AG.

A taxa de *crossover* p_c é um parâmetro definido pelo usuário que determina, para todas as gerações, o número de cruzamentos entre indivíduos que serão realizados. Pode ser um valor fixo ou variável, ao longo das gerações, determinado por uma função. Por exemplo, se a taxa de *crossover* é definida em 60% sobre uma população de 100 indivíduos, então, a cada geração, ocorrerá 60 cruzamentos. A taxa de *crossover* não pode ser muito baixa, pois como visto, é a grande responsável pela diversificação dos indivíduos na população.

A taxa de mutação p_m está associada a chance de um indivíduo sofrer alteração das suas informações genéticas. Pode ser um valor fixo ou variável, determinado por uma função. Se a taxa p_m for baixa demais (próxima de 0), a mutação agirá de forma extremamente parcimoniosa e a população terá pouca diversificação nos indivíduos ao longo das gerações (Linden, 2008). Contudo, se a taxa de mutação for muito alta (próxima de 1), então o AG passará a ter um comportamento parecido com um algoritmo aleatório e perderá suas características interessantes.

De modo geral, os parâmetros p_c e p_m são responsáveis por controlar a frequência de reprodução na população, e a determinação de seus valores pode interferir na qualidade da solução obtida pelo algoritmo.

Há uma grande discussão na definição destes parâmetros na literatura. Mitchell (1992) faz um levantamento dos parâmetros testados por diversos autores da área e dentre eles, destacam-se experimentos de De Jong, que indicaram como melhores valores: tamanho da população entre 50 – 100 indivíduos; *crossover* 1-ponto

a uma taxa de 60%; e probabilidade de mutação de 1%.

Estas configurações tornaram-se amplamente utilizadas nos AGs pela comunidade. Contudo, estes valores não são comprovadamente os mais eficientes, e sugere-se ainda mais pesquisas para determinar valores nos quais o algoritmo trabalha melhor para problemas específicos, visto que o controle sobre os parâmetros é de fundamental importância para uma convergência rápida e global.

2.5.9 Critério de parada

No Algoritmo Genético o processo de reprodução acontece em cada geração, de forma iterativa, diversificando os indivíduos da população, até que algum critério de parada seja satisfeito.

O critério de parada é, em geral, expresso em termos do número de gerações a serem executadas. Neste caso, determina-se um número fixo G de gerações para a evolução do algoritmo, cujo tempo computacional seja aceitável. Obtém-se ao final das gerações a melhor solução possível para o problema no tempo estabelecido. Em problemas em que uma solução ótima pode ser reconhecida, o algoritmo pode terminar quando tal indivíduo for encontrado (Koza, 1992).

Pode-se ainda estabelecer um número w de gerações em que não se observa melhoras significativas na solução ótima encontrada até o momento. Neste caso, é estabelecida uma precisão p no início do algoritmo que verificará, ao longo de w gerações, se as soluções encontradas são melhoradas em ao menos um valor p .

2.6 *Simulated Annealing* (SA)

O método *Simulated Annealing* (Recozimento Simulado) tem sua fundamentação baseada no processo físico de recozimento, no qual substâncias como metais são fundidos a altas temperaturas e em seguida resfriados lentamente, até atingirem o estado de congelamento.

Sob altas temperaturas, o material possui uma configuração de alta energia, em que suas partículas se arranjam de forma aleatória. Mas, durante o

Tabela 3: Vocabulários utilizados no processo de recozimento físico e simulado

Física	Otimização
Configuração	Solução
Configuração de energia mínima	Solução ótima
Nível de energia	Valor da função objetivo
Temperatura	Parâmetro de controle
Congelamento	Critério de parada

processo de resfriamento, o material passará por diversas configurações de energia mais baixa, até atingir uma configuração de energia mínima, formando um material sólido, livre de imperfeições.

Em 1983, Kirkpatrick et al. (1983) propuseram esta idéia como estratégia de otimização, e seu desenvolvimento só foi possível a partir de estudos realizados por Metropolis et al. (1953), que resultou no mais conhecido método de Monte Carlo, o algoritmo de Metropolis.

O algoritmo de Metropolis é utilizado para simular a mudança do estado de energia de um sistema quando sujeito ao processo de resfriamento, até convergir ao estado de congelamento. Já o algoritmo *Simulated Annealing*, ao inserir o fator temperatura no algoritmo de Metropolis, emprega uma sequência de temperaturas decrescentes, imitando o processo de recozimento e gerando soluções para um problema de otimização.

Para se alcançar o estado de energia mínimo desejado, deve-se partir de uma temperatura inicial suficientemente alta e o processo de resfriamento deve ser suficientemente lento. Isto possibilita explorar melhor o espaço de soluções, e consequentemente, encontrar a melhor solução que minimiza o problema em questão. A analogia entre o processo físico e o problema de otimização podem ser resumidos na Tabela (3).

Na implementação do SA, os principais parâmetros a serem considerados são: temperatura inicial ($T_{inicial}$), temperatura final (T_{final}), número de iterações

em cada temperatura (L), e taxa de resfriamento (α).

A partir de uma solução inicial s , construída de forma aleatória ou por uma heurística construtiva, o processo de recozimento simulado inicia-se com $T_{inicial}$ elevada, e a cada temperatura corrente, geram-se soluções até que o equilíbrio a esta temperatura seja alcançado. Este equilíbrio é dado pelo número de iterações L na temperatura corrente.

A cada iteração do método, um novo estado de energia é gerado a partir do estado corrente, por uma modificação aleatória neste. Isso significa que, a partir da solução corrente s , determina-se uma vizinhança $V(s)$, escolhe-se aleatoriamente uma solução $s' \in V(s)$, e avalia-se a função objetivo neste novo ponto.

A variação de energia no sistema obtida do movimento de s para s' é expressa em Δ e pela Equação (6).

$$\Delta = f(s') - f(s) \geq 0 \quad (6)$$

Assim, se a nova configuração s' é de energia menor que a configuração corrente s , ou seja, $\Delta < 0$, a nova configuração passa a ser a configuração corrente, ou seja, $s \leftarrow s'$. Caso contrário, sorteia-se um número aleatório $r \in [0, 1]$ e verifica-se a probabilidade de aceitar um movimento que não é de melhora, dada pela Equação (7).

$$P(\text{aceitação}) = e^{-\Delta/(kT)}, \quad (7)$$

em que k é a constante de Boltzmann e T é a temperatura corrente.

O comportamento da função de probabilidade apresentada na Equação (7) pode ser observado na Figura (16). Note que em baixas temperaturas movimentos que não são de melhora são aceitos com probabilidade baixa, enquanto em altas temperaturas, a chance de aceitação aumenta, a fim de evitar que o algoritmo convirja para um ótimo local. Na Figura (16) fixou-se Δ igual a 1, e temperaturas variando de 1 a 100, com o objetivo de verificar o comportamento da função de um modo mais geral.

Após L iterações, a temperatura corrente é então rebaixada a uma taxa

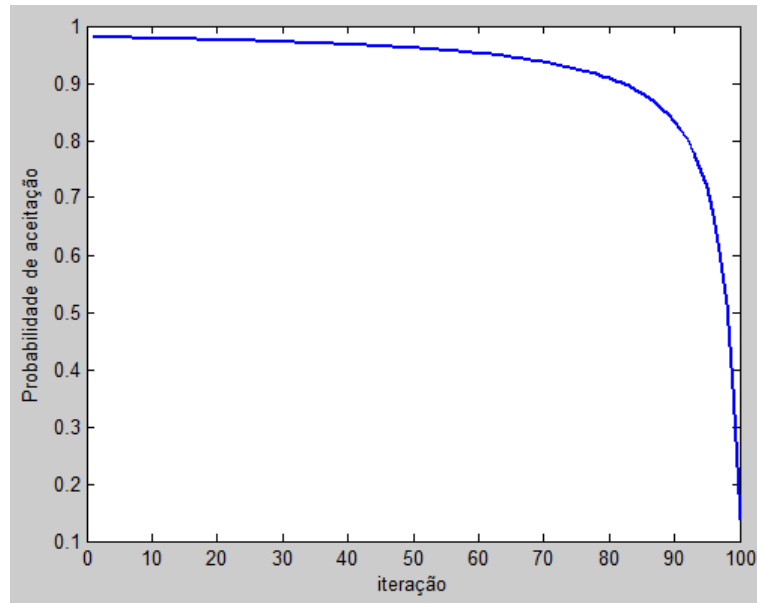


Figura 16 - Probabilidade de aceitação de movimentos que não são de melhora pelo algoritmo SA

α , $T = \alpha T$ e o processo prossegue até o congelamento, ou seja, até atingir a mais baixa temperatura definida em T_{final} , que deve ser suficientemente próxima de zero ($T_{final} \approx 0$).

A sequência de temperaturas empregadas, determinada pela taxa de resfriamento α ($0 < \alpha < 1$), juntamente com o número de iterações L a cada temperatura ($L > 0$), são parâmetros definidos pelo usuário.

Em geral, como o objetivo é obter um resfriamento suficientemente lento, adota-se α o mais próximo de 1 possível, por exemplo, $\alpha = 0,95$. Isso implica em aumentar o número de temperaturas testadas pelo método, e conseqüentemente, haverá mais iterações. Por isso, o número de iterações L deve ser equilibrada ao número de iterações definidas por α , a fim de se obter um algoritmo com tempo computacional de execução aceitável.

Este procedimento é repetido até se atingir o estado de congelamento. Para melhor compreensão do processo, um pseudocódigo para o algoritmo do *Simulated Annealing* é apresentado na Figura (17).

```

s=solução inicial;
s*=s;
T=Tinicial;
enquanto (T > Tfinal)
  para (i = 1:L)
    s' = uma solução aleatória de V(s);
    Δ = f(s') - f(s);
    se (Δ < 0) então
      s = s';
      se (f(s') < f(s*)) então
        s* = s';
      fim_se
    senão
      r = uniforme(0,1);
      se (r < e(-Δ/T)) então
        s = s';
      fim_se
    fim_se
  fim_para
  T = αT;
fim_enquanto

```

Figura 17 - Pseudocódigo do algoritmo *Simulated Annealing*

Como estratégia de otimização, o SA é um método de busca local que permite movimentos que não são de melhora, de acordo com uma probabilidade de aceitação, possibilitando assim escapar de mínimos locais.

2.7 Algoritmos Híbridos

Algoritmos híbridos consistem na utilização conjunta de métodos de otimização exatos e/ou heurísticos, combinando suas vantagens a fim de obter melhores soluções. Estes métodos, em geral, visam reunir boas características de dois ou mais métodos em um único algoritmo, com o objetivo de melhorar a convergência e obter melhores soluções para o problema.

A partir da constatação de que nenhum método de otimização é perfeito, vários sistemas híbridos têm sido propostos. Goldberg & Luna (2005) fazem um levantamento de trabalhos que propõem Algoritmos Genéticos combinados com

estratégias exatas ou heurísticas para resolução de problemas de otimização. Dentre estes métodos, destacam-se a combinação do AG com *Simulated Annealing*, *Hill Climbing*, Colônia de Formigas e Busca Tabu.

Para Davis (1991), o resultado da hibridização costuma ser melhor que o obtido com qualquer uma das duas técnicas isoladamente. Porém, não existe regra para a combinação dos métodos, depende da complexidade do problema e das estratégias de otimização conhecidas pelo programador.

Quando utilizamos o Algoritmo Genético como estratégia de otimização, Ochi (1998) sugere a combinação com alguma estratégia de busca local ao redor das soluções de elite, fazendo assim um refinamento a partir das melhores soluções encontradas. Para Goldberg (1989), um caminho eficiente para melhorar consideravelmente o desempenho de um AG é fazer este refinamento da solução através do SA.

A partir desta idéia, utilizou-se a estratégia híbrida AG+SA que consiste em pegar a melhor solução gerada pelo Algoritmo Genético ao final das gerações e utilizá-la como solução inicial do *Simulated Annealing*, que normalmente utilizaria uma solução inicial construída aleatoriamente.

3 MATERIAL E MÉTODOS

Neste capítulo é proposto o estudo e a formulação de um modelo matemático para o planejamento de plantio em lotes visando reduzir a proliferação de pragas entre culturas, bem como a resolução do modelo a partir de estratégias metaheurísticas de otimização. O estudo e a implementação das metaheurísticas Algoritmo Genético (AG), *Simulated Annealing* (SA) e híbrido AG+SA foram realizados.

Para o desenvolvimento das simulações computacionais foi utilizado o software Matlab R2010a, versão 7.10.0.499, em microcomputadores com processador Intel Core i3 e 2 GB de memória RAM, em sistema operacional Windows 7. Neste ambiente foram implementadas as estratégias metaheurísticas AG, SA e híbrido *AG+SA*, e simuladas para duas instâncias do problema, variando-se os parâmetros de entrada, a fim de se observar a influência destes no desempenho dos algoritmos.

3.1 Modelagem Matemática do Problema Biológico

Nesta seção é apresentado um modelo de Programação Não-Linear para auxílio no planejamento otimizado do plantio e cultivo de culturas em determinados lotes durante um período de tempo, de modo que a disposição dessas culturas e/ou variedades nos lotes favoreça o controle eficiente de pragas. Isto implica em determinar um modelo matemático para o planejamento de plantio e cultivo que minimize a probabilidade de proliferação de pragas entre as culturas, levando em consideração restrições de demanda, período de cultivo de culturas, ocupação de lotes e tempo de

planejamento.

Para a construção do modelo matemático para o problema apresentado serão considerados os seguintes parâmetros:

- N : número de culturas disponíveis para o plantio;
- K : número de lotes disponíveis para o plantio;
- M : número de lotes/culturas circunvizinhos à fazenda de interesse;
- i e $\bar{i}$: índices relacionados às culturas, $i = 1, \dots, (N + M)$ e $\bar{i} = 1, \dots, (N + M)$;
- j e $\bar{j}$: índices relacionados aos lotes, $j = 1, \dots, (K + M)$ e $\bar{j} = 1, \dots, (K + M)$;
- A : duração do planejamento, igual para todos os lotes;
- T : quantidade de subdivisões do período A ;
- t : índice relacionado ao período do planejamento, $t = 1, \dots, T$.

Seja um planejamento de culturas em um período A e uma unidade temporal que divide A em T subperíodos. Construir um planejamento de culturas de duração A implica em decidir qual cultura deverá ser plantada em cada lote na unidade de tempo adotada (dia, semana, mês, bimestre, etc). Considera-se que a área de plantio está subdividida em K lotes e que possui M fazendas circunvizinhas com M culturas já plantadas.

Para ilustração do problema proposto, considere um planejamento de culturas no horizonte de 2 anos ($A = 2$), dividido em 24 meses ($T = 24$), em que os subperíodos considerados no modelo, neste caso, variam em $t = 1, \dots, 24$. Assim, $t = 3$ refere-se ao 3º mês do planejamento.

Suponha a região ilustrada na Figura (18) em que deseja-se realizar o plantio de culturas. Os lotes $M1$, $M2$, $M3$ e $M4$ representam as fazendas circunvizinhas à área de interesse de plantio. Já os lotes $K1$, $K2$, $K3$, $K4$, $K5$ e $K6$, representam as subdivisões em lotes da área disponível para plantio. Podemos observar as seguintes características da região:

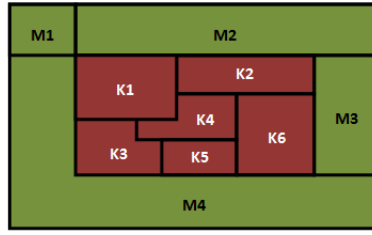


Figura 18 - Área hipotética dividida em lotes para planejamento de plantio

- Área disponível para plantio subdividida em $K = 6$ lotes ($K1, \dots, K6$);
- Área circunvizinha à região de interesse com $M = 4$ fazendas e culturas já fixadas ($M1, \dots, M4$).

O relacionamento de vizinhança entre os lotes da Figura (18) pode ser representado por um grafo planar, não direcionado, como mostra a Figura (19).

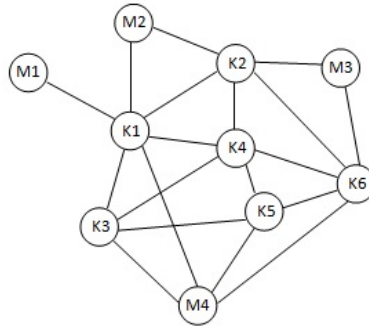


Figura 19 - Representação em grafo da região ilustrada na Figura (18)

No grafo, cada lote é representado por um vértice e cada aresta expressa a relação de adjacência entre dois lotes. A interpretação do grafo é importante pois auxiliará na construção da matriz de adjacências, a seguir.

Seja V uma matriz de dimensão $(K + M) \times (K + M)$, simétrica, em que seus elementos $v_{j\bar{j}}$ expressam o relacionamento de vizinhança entre os lotes j e $\bar{j}$. $V = [v_{j\bar{j}}]$, com $j = 1, \dots, (K + M)$ e $\bar{j} = 1, \dots, (K + M)$.

Para $j = \bar{j}$ define-se $v_{j\bar{j}} = +\infty$. Esta estratégia impede que se plante ao mesmo tempo culturas distintas no mesmo lote, o que não é permitido no modelo.

Fazer $v_{j\bar{j}} = +\infty$ torna a decisão de plantar duas culturas no mesmo lote muito cara, o que inviabiliza o objetivo do problema que é a minimização da probabilidade de transmissão de pragas. Define-se $v_{j\bar{j}} = +\infty$ para $j \geq (K + 1)$ e $\bar{j} \geq (K + 1)$, pois trata-se do relacionamento de vizinhança entre as fazendas da circunvizinhança, e isso não é de interesse do modelo.

Seja $d_{j\bar{j}}$ a menor distância entre os pontos dos lotes j e $\bar{j}$, com $d_{j\bar{j}} \geq 0$, define-se:

$$v_{j\bar{j}} = \frac{1}{d_{j\bar{j}} + 1},$$

para $j < (K + 1)$ ou $\bar{j} < (K + 1)$, $j \neq \bar{j}$, em que $v_{j\bar{j}}$ mede o grau de vizinhança entre os lotes j e $\bar{j}$, e $d_{j\bar{j}}$ é a menor distância entre os pontos dos lotes j e $\bar{j}$.

Se os lotes j e $\bar{j}$ são vizinhos, eles possuem uma aresta em comum, e isso implica que a menor distância $d_{j\bar{j}}$ entre eles é zero e $v_{j\bar{j}}$ assume o valor 1, ou seja, um alto grau de vizinhança entre os lotes. Nos demais casos em que os lotes j e $\bar{j}$ não são vizinhos, a relação de vizinhança $v_{j\bar{j}}$ é inversamente proporcional a distância entre eles, ou seja, quanto maior a distância $d_{j\bar{j}}$ entre os lotes j e $\bar{j}$, menor é o valor de $v_{j\bar{j}}$. Assim,

$$\lim_{d_{j\bar{j}} \rightarrow +\infty} v_{j\bar{j}} = \lim_{d_{j\bar{j}} \rightarrow +\infty} \frac{1}{d_{j\bar{j}} + 1} = 0.$$

Isso significa que plantar em lotes mais distantes diminui o valor de $v_{j\bar{j}}$. Do contrário, plantar em lotes mais próximos, implica no aumento de $v_{j\bar{j}}$, ou seja,

$$\lim_{d_{j\bar{j}} \rightarrow 0} v_{j\bar{j}} = \lim_{d_{j\bar{j}} \rightarrow 0} \frac{1}{d_{j\bar{j}} + 1} = 1.$$

Assim, a matriz V será escrita como mostra a Figura (20).

Outra matriz de parâmetros a ser definida para uso no modelo é a matriz de probabilidade de transmissão de praga, denotada por P .

Seja P uma matriz de dimensão $(M + N) \times (M + N)$, não simétrica, em que seus elementos $p_{i\bar{i}}$ expressam a probabilidade de transmissão da praga da cultura i para a cultura $\bar{i}$. $P = [p_{i\bar{i}}]$, com $i = 1, \dots, (N + M)$ e $\bar{i} = 1, \dots, (N + M)$.

$$V = \begin{pmatrix} \infty & v_{12} & \dots & v_{1k} & v_{1k+1} & \dots & v_{1k+m} \\ v_{21} & \infty & \dots & v_{2k} & v_{2k+1} & \dots & v_{2k+m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ v_{k1} & v_{k2} & \dots & \infty & v_{kk+1} & \dots & v_{kk+m} \\ v_{k+11} & v_{k+12} & \dots & v_{k+1k} & \boxed{\infty} & \dots & v_{k+1k+m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ v_{k+m1} & v_{k+m2} & \dots & v_{k+m k} & \vdots & \dots & v_{k+m k+m} \end{pmatrix}$$

Figura 20 - Matriz de relacionamento de vizinhança entre lotes

Define-se $p_{i\bar{i}} = 1$ para $i = \bar{i}$, pois trata-se da transmissão da praga entre a mesma cultura. Para $i \geq (N + 1)$ e $\bar{i} \geq (N + 1)$ define-se que $p_{i\bar{i}} = 1$, pois trata-se da transmissão da praga entre as culturas das fazendas circunvizinhas, e isso não é de interesse do modelo. Já para $i < (N + 1)$ ou $\bar{i} < (N + 1)$, $p_{i\bar{i}}$ recebe um valor $\in [0, 1]$ que representa a probabilidade da praga migrar da cultura i para a cultura $\bar{i}$. Valores de $p_{i\bar{i}}$ próximos de 1 representam alta probabilidade de transmissão da praga entre as culturas e, da mesma forma, valores de $p_{i\bar{i}}$ próximos de 0 representam baixa probabilidade.

Definidos os parâmetros, a matriz P terá a seguinte forma:

$$P = \begin{pmatrix} 1 & p_{12} & \dots & p_{1n} & p_{1n+1} & \dots & p_{1n+m} \\ p_{21} & 1 & \dots & p_{2n} & p_{2n+1} & \dots & p_{2n+m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \dots & 1 & p_{nn+1} & \dots & p_{nn+m} \\ p_{n+11} & p_{n+12} & \dots & p_{n+1n} & \boxed{1} & \dots & p_{n+1n+m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ p_{n+m1} & p_{n+m2} & \dots & p_{n+m n} & \vdots & \dots & p_{n+m n+m} \end{pmatrix}$$

Figura 21 - Matriz de probabilidade de proliferação de praga entre culturas

A seguir é formulado um modelo de planejamento, cujo objetivo é minimizar a probabilidade de proliferação de praga entre as N culturas a serem plantadas em uma área dividida em K lotes, sujeito as restrições: as culturas não podem ocupar o mesmo lote no mesmo período de tempo; deve-se atender a demanda

de cada cultura no período definido; deve-se respeitar o período de plantio e colheita de cada cultura; culturas de mesma família botânica não podem ser plantadas em períodos consecutivos; e deve-se respeitar o tempo previsto no planejamento.

Sejam X_{ijt} as variáveis de decisão do modelo, em que $X_{ijt} = 1$ se a cultura i for plantada no lote j no período t e $X_{ijt} = 0$ em caso contrário. Segue o modelo:

$$\min \sum_{i=1}^{N+M} \sum_{\bar{i}=1}^{N+M} \sum_{j=1}^{K+M-1} \sum_{\bar{j}=j+1}^{K+M} \sum_{t=1}^T V_{j\bar{j}} P_{\bar{i}} X_{ijt} X_{\bar{i}\bar{j}t} \quad (8)$$

Sujeito a:

$$\sum_{i=1}^N X_{ijt} = 1, \quad j = 1, \dots, K \text{ e } t = 1, \dots, T \quad (9)$$

$$\sum_{j=1}^K R_{it} \text{Area}_j X_{ijt} \geq D_{it}, \quad i = 1, \dots, N \text{ e } t = 1, \dots, T \quad (10)$$

$$\sum_{i=1}^N \sum_{r=0}^{C_i-1} X_{ij(t+r)} \leq 1, \quad j = 1, \dots, K \text{ e } t = 1, \dots, T \quad (11)$$

$$\sum_{i \in F_p} \sum_{r=0}^{C_i} X_{ij(t+r)} \leq 1, \quad j = 1, \dots, K, \quad t = 1, \dots, T \text{ e } p = 1, \dots, N_f \quad (12)$$

$$\sum_{t=1}^T \sum_{i=1}^N X_{ijt} \quad C_i = T, \quad j = 1, \dots, K \quad (13)$$

$$X_{ijt} = \{0, 1\}, \quad i = 1, \dots, N, \quad j = 1, \dots, K \text{ e } t = 1, \dots, T$$

O objetivo do problema é minimizar a probabilidade de proliferação da praga entre culturas de uma área de plantio dividida em lotes, definido pela função (8). As restrições (9) impedem que mais de uma cultura ocupe o mesmo lote j no mesmo período de tempo t ; (10) respeitam a demanda da cultura i no período t ; (11) respeitam o período de cultivo da cultura i ; (12) impedem que culturas de mesma família botânica sejam plantadas em períodos consecutivos; e (13) respeitam o tempo T de planejamento. E ainda,

- R_{it} é a estimativa de produção associada a cultura i no tempo t , plantada no lote j ;
- $Area_j$ é a área de cada lote j , medida em hectares (ha);
- C_i é o ciclo de vida da i -ésima cultura, incluindo tempo de preparação da terra e colheita;
- D_{it} é a estimativa de demanda associada a cultura i no período t , dada em kilogramas (kg);
- N_f é o número de famílias botânicas disponíveis para o plantio;
- F_p é o conjunto de culturas pertencentes a família p , $p = 1, \dots, N_f$.

No ambiente agronômico, produtividade é definida como a quantidade de produção por unidade de área. Informações sobre produtividade e demanda das culturas utilizadas nas simulações foram retiradas e adaptadas de Aliano Filho (2012) e podem ser vistas no Apêndice.

3.2 Heurística Construtiva

Nesta seção será apresentada uma heurística construtiva para criar soluções que atendam às restrições (11), (12) e (13), de período de cultivo das culturas, plantio consecutivo de culturas de mesma família botânica em um lote e tempo de planejamento, respectivamente.

A heurística construtiva inicia-se selecionando aleatoriamente uma cultura, dentre as N disponíveis, e plantando-a no primeiro lote, no primeiro período de tempo. Até que o tempo de planejamento T seja satisfeito para cada lote e os K lotes sejam preenchidos, a heurística prossegue sorteando culturas e preenchendo a matriz de solução, respeitando o período de cultivo da cultura sorteada, e descartando culturas que são da mesma família botânica da cultura anteriormente plantada no lote, ou que não respeitam o tempo de planejamento, ultrapassando-o.

Ainda, se em um número de vezes limitado ℓ o algoritmo não conseguir, por meio do sorteio aleatório, plantar uma cultura no lote, este ficará vazio por uma unidade de tempo, tornando esta solução infactível. O número de tentativas ℓ foi definido como o número de culturas disponíveis, ou seja, $\ell = N = 24$.

3.3 Penalização

Neste trabalho será utilizada uma função de penalização exponencial, sugerida por Aliano Filho (2012), para penalizar as soluções que não atendem as restrições (9) e (10), de ocupação de lotes e demanda, respectivamente. Nesta proposta, a função penalidade é denotada por p . Primeiro, $p(S)$ soma o número de restrições não satisfeitas pela solução S , e em seguida, lhe é atribuído um novo valor, dado pela Equação (14).

$$p(S) \leftarrow p(S) \exp \left(\frac{-p(S)}{k} \right), \quad (14)$$

em que k é uma constante a ser definida pelo programador para medir a força de penalização. A Figura (22) mostra o comportamento da função de penalização definida pela Equação (14).

Desta forma, uma solução S factível não sofrerá nenhuma redução em sua avaliação e pode-se escrever a função de avaliação como na Equação (15).

$$eval(S) = f(S) \pm p(S) \quad (15)$$

Na Equação (15), soma-se $p(S)$ se o objetivo for de minimização e subtrai-se este valor se o objetivo for de maximização. Com o objetivo de minimização, utilizou-se neste trabalho $eval(S) = f(S) + p(S)$, em que $f(S)$ é o valor da função objetivo para a solução S e $eval(S)$ é a avaliação final da solução, já penalizada.

A penalização exponencial utilizada se justifica por relacionar o número de restrições violadas pelo indivíduo, interferindo de forma mais rígida no valor de sua aptidão conforme mais restrições são violadas.

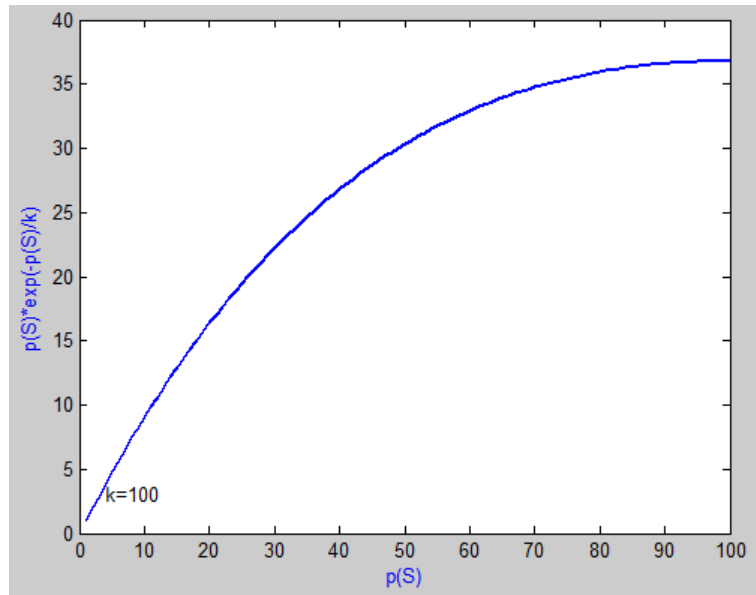


Figura 22: Comportamento da função de penalização exponencial com $k = 100$.

3.4 Algoritmo Genético

A principal característica do Algoritmo Genético (AG) é que, diferente dos métodos tradicionais de busca e otimização, estes trabalham com uma população de soluções e não apenas uma única. Portanto, na implementação do AG, inicialmente foi gerada uma população de Q indivíduos utilizando a heurística construtiva sugerida na Seção 3.2.

A representação utilizada para uma solução foi uma matriz S de dimensão $K \times T$, em que K é o número de lotes disponíveis para plantio e T é o tempo de planejamento. Assim, cada matriz representa um planejamento de plantio para os lotes no tempo determinado, e cada elemento da matriz representa a cultura que será plantada naquele lote, naquele tempo.

Ilustrou-se, na Figura (23), uma matriz solução S para um planejamento de plantio em 5 lotes no período de 6 meses.

Na Figura (23), as culturas 1, 2, 3 e 4 estão disponíveis para plantio com os períodos de cultivo 1, 4, 3 e 2 meses, respectivamente. A matriz solução S tem dimensão 5×6 , ou seja, $K = 5$ lotes e $T = 6$ meses.

	Tempo (mês)					
	Jan	Fev	Mar	Abr	Mai	Jun
Lote 1	3	3	3	1	4	4
Lote 2	4	4	1	3	3	3
Lote 3	3	3	3	1	4	4
Lote 4	4	4	3	3	3	1
Lote 5	1	2	2	2	2	1

Figura 23 - Exemplo de matriz solução para o problema abordado

Durante o processo evolutivo, todo indivíduo é avaliado por meio da função objetivo descrita na Equação (8), que reflete a probabilidade total de haver proliferação da praga entre as culturas plantadas naquela configuração, e o valor final de sua aptidão é dado pela soma do valor da função objetivo e a penalização, proposta na Equação (14).

Utilizou-se o AG com elitismo, a fim de não perder bons indivíduos após o processo de reprodução. Para tanto, um número fixo Q_{elite} de indivíduos mais aptos e factíveis são copiados da população inicial para o grupo de elite em cada geração do AG. Após a reprodução, estes Q_{elite} indivíduos são devolvidos para a população inicial em substituição aos indivíduos menos aptos desta geração. O número de indivíduos utilizados no elitismo foi de até $Q_{elite} = 2$, dependendo da factibilidade dos indivíduos na população.

O próximo passo é definir o número de indivíduos da população intermediária. Estabeleceu-se a taxa de seleção $p_s = 80\%$ de indivíduos da população inicial que serão selecionados pelos métodos de Roleta Viciada ou Torneio Binário para comporem a população intermediária e participarem da reprodução. Os indivíduos selecionados podem sofrer modificações em suas características fundamentais através de cruzamentos e mutações, gerando descendentes para a próxima geração.

Todas as simulações foram realizadas utilizando o *crossover* 1-ponto, com taxa de cruzamentos p_c de 60% sobre o número de indivíduos da população

inicial. Entre os indivíduos da população intermediária foram realizados cruzamentos com poligamia.

A probabilidade de mutação foi estabelecida por meio da função logística descrita pela Equação (16).

$$p_m(g) = \frac{\theta}{\theta + e^{(1-g)/10}}, \quad (16)$$

em que o parâmetro θ representa a probabilidade de haver mutação na geração $g = 1$, e θ foi estabelecido em 1%, ou seja, $\theta = 0,01$.

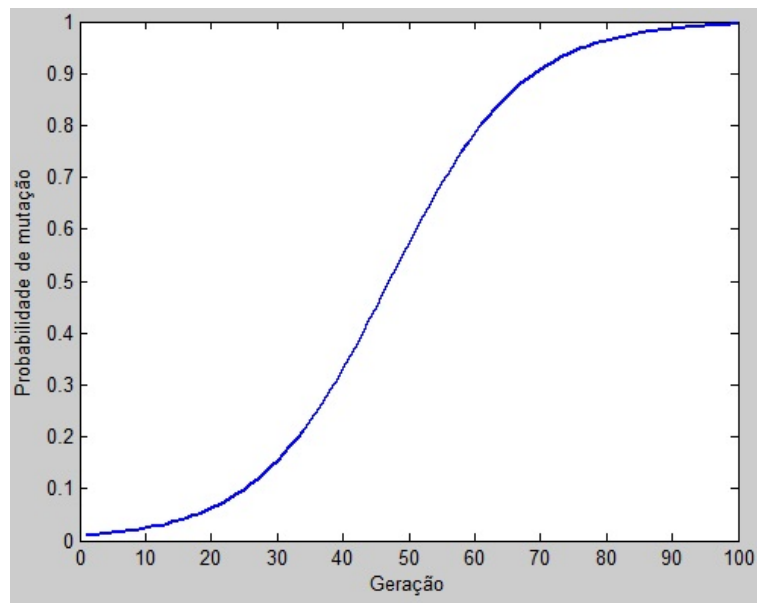


Figura 24 - Comportamento da função logística de probabilidade de mutação

Observe na Figura (24) que, no início do algoritmo, a probabilidade de mutação é baixa e, ao longo de 100 gerações, a fim de evitar convergência para um ótimo local, esta probabilidade $p_m(g)$ cresce, e consequentemente, cresce a diversidade nas soluções.

Este processo é repetido até que o critério de parada $G = 100$ número de gerações seja satisfeito e a melhor solução para o problema é encontrada.

3.5 *Simulated Annealing*

O esquema de representação da solução utilizado no *Simulated Annealing* é o mesmo utilizado no Algoritmo Genético, através de uma matriz S de dimensão $K \times T$.

No método *Simulated Annealing*, a principal definição é com relação a estrutura de vizinhança da solução corrente S . Uma vizinhança $V(S)$ de uma solução S foi definida trocando-se aleatoriamente uma linha da matriz solução por uma nova linha construída pela heurística construtiva, como mostra a Figura (25).

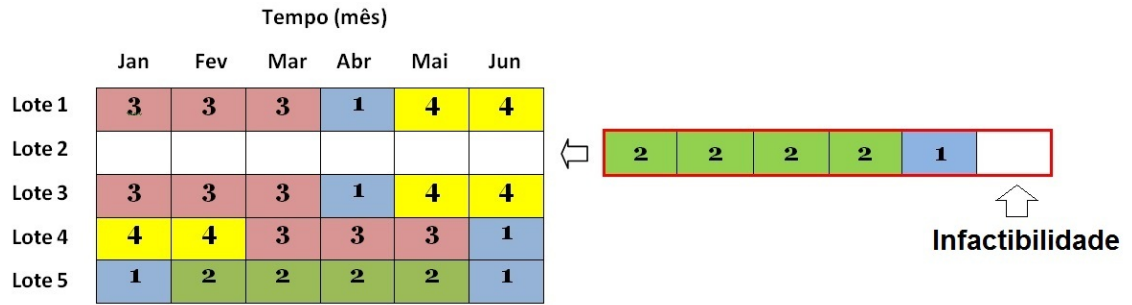


Figura 25 - Possível vizinho da solução corrente S apresentada na Figura (23)

Desta forma, é possível definir 5 vizinhanças para a solução corrente S observada na Figura (23), por meio da escolha aleatória de uma das linhas da matriz.

O número de iterações L em cada temperatura foi estabelecido de acordo com o número de culturas disponíveis para plantio, ou seja, fixa igual a 24.

A temperatura inicial $T_{inicial}$ foi definida em 100 graus, e o critério de parada do método é a temperatura de congelamento T_{final} , suficientemente próxima de zero, testadas em $T_{final} = 10^{-4}$ e $T_{final} = 10^{-6}$.

Os parâmetros fixados para o *Simulated Annealing* foram resumidos na Tabela (4), e a constante de Boltzmann, apresentada na Seção 2.5, foi fixada em uma unidade.

Tabela 4: Parâmetros utilizados no *Simulated Annealing*

Parâmetro	Valor
$T_{inicial}$	100
T_{final}	$\{10^{-4}; 10^{-6}\}$
L	24
α	0,95

3.6 Híbrido AG+SA

O método híbrido AG+SA foi a terceira metaheurística utilizada nas simulações. A idéia central deste procedimento consiste em encontrar, utilizando o método Algoritmo Genético, uma boa solução inicial para a segunda heurística que entrará em ação, o *Simulated Annealing*. Desta forma, o primeiro algoritmo híbrido testado utiliza o método de seleção Torneio Binário no Algoritmo Genético e o segundo híbrido utiliza o método de seleção Roleta Viciada.

Utilizou-se neste trabalho a combinação de métodos híbridos com os mesmos parâmetros utilizados nos métodos puros, e a implementação desta estratégia consiste em tomar a melhor solução gerada pelo Algoritmo Genético e utilizá-la como solução inicial do *Simulated Annealing*.

3.7 Instâncias simuladas

As estratégias propostas foram simuladas para as instâncias ilustradas na Figura (26). A instância A contém 10 lotes, sendo 06 disponíveis para o planejamento ($K1, \dots, K6$), e 04 propriedades vizinhas que circundam a área de interesse e podem interferir no planejamento. Do mesmo modo, a instância B contém 15 lotes, sendo 10 disponíveis para o planejamento e 05 propriedades vizinhas. A Tabela (5) resume as informações sobre os lotes utilizados nas simulações.

A probabilidade de uma praga migrar de uma cultura para outra é um dos principais parâmetros do modelo e está relacionada a preferência alimentar

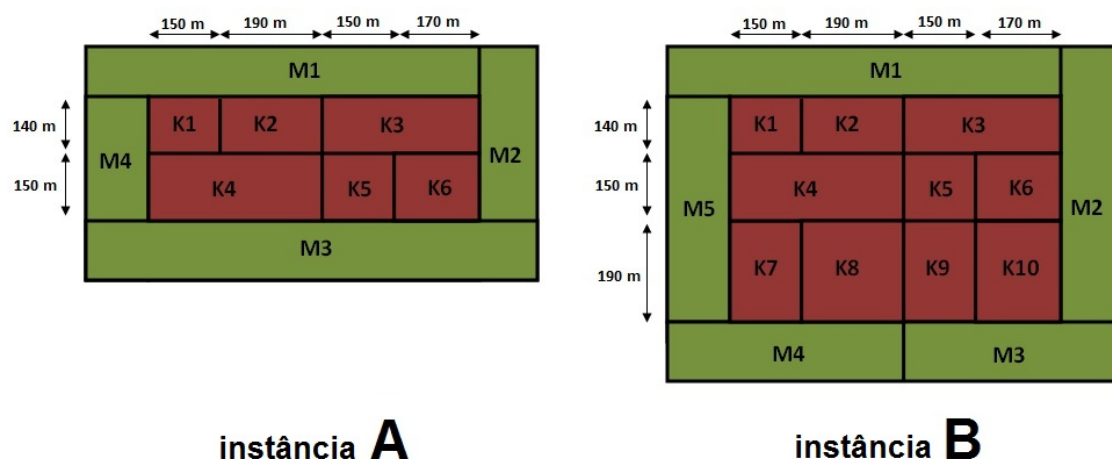


Figura 26 - Geometria e dimensão das áreas consideradas nas simulações

da praga pelas culturas. Nas simulações esta probabilidade foi estabelecida em alta (0,9), média (0,5) e baixa (0,1), e a preferência alimentar da praga pelas culturas é hipotética, como segue:

- Culturas de mesma família botânica tem probabilidade alta de proliferação;
- Entre as culturas de famílias botânicas pares², a probabilidade de proliferação é média;
- Entre as culturas de famílias botânicas ímpares, a probabilidade de proliferação é média;
- A probabilidade baixa de proliferação acontece entre culturas de famílias botânicas pares para as ímpares, e vice-versa;
- As propriedades vizinhas identificadas por um número par possuem culturas plantadas de famílias botânicas pares, e da mesma forma, as propriedades vizinhas ímpares possuem culturas de famílias botânicas ímpares.

²A identificação numérica das famílias botânicas pode ser consultada no Apêndice.

Tabela 5. Áreas dos lotes

Lote	Dimensão (m)	Área (ha)
K_1	140x150	2,10
K_2	140x190	2,66
K_3	140x320	4,48
K_4	150x340	5,10
K_5	150x150	2,25
K_6	150x170	2,55
K_7	190x150	2,85
K_8	190x190	3,61
K_9	190x150	2,85
K_{10}	190x170	3,23

Foi considerado nas simulações um tempo de planejamento de $A = 2$ anos, subdividido em períodos de 1 mês ($T = 24$ meses), e utilizou-se informações sobre 24 culturas retiradas de Santos (2009) (Apêndice).

3.8 Algoritmos testados

Nesta seção são apresentados os algoritmos testados com relação aos métodos e parâmetros estabelecidos.

Algoritmo 1

Algoritmo Genético com elitismo testado na instância A, com população inicial de $Q = 100$ indivíduos e elite com $Q_{elite} = 2$, taxa de seleção $p_s = 80\%$ com seleção por Torneio Binário, *crossover* 1-ponto com taxa $p_c = 60\%$, taxa de mutação p_m é uma função logística com $\theta = 0,01$ e critério de parada $G = 100$ gerações.

Notação: $AG_Torneio_A$.

Algoritmo 2

Algoritmo Genético com elitismo testado na instância A, com população inicial de $Q = 100$ indivíduos e elite com $Q_{elite} = 2$, taxa de seleção $p_s = 80\%$ com seleção por Roleta Viciada, *crossover* 1-ponto com taxa $p_c = 60\%$, taxa de mutação p_m é uma função logística com $\theta = 0,01$ e critério de parada $G = 100$ gerações.

Notação: *AG_Roleta_A*.

Algoritmo 3

Simulated Annealing testado na instância A, com temperatura inicial $T_{inicial} = 100$ graus, temperatura final $T_{final} = 10^{-4}$ graus, número de iterações por temperatura $L = 24$ e taxa de resfriamento $\alpha = 0,95$.

Notação: *SA_10⁻⁴_A*.

Algoritmo 4

Algoritmo híbrido de Genético com *Simulated Annealing* que utiliza a melhor solução gerada pelo Algoritmo 1 como solução inicial para refinamento através do Algoritmo 3.

Notação: *AG+SA_Torneio_A*.

Algoritmo 5

Algoritmo híbrido de Genético com *Simulated Annealing* que utiliza a melhor solução gerada pelo Algoritmo 2 como solução inicial para refinamento através do Algoritmo 3.

Notação: *AG+SA_Roleta_A*.

Algoritmo 6

Algoritmo Genético com elitismo testado na instância B, com população inicial de $Q = 100$ indivíduos e elite com $Q_{elite} = 2$, taxa de seleção $p_s = 80\%$

com seleção por Torneio Binário, *crossover* 1-ponto com taxa $p_c = 60\%$, taxa de mutação p_m é uma função logística com $\theta = 0,01$ e critério de parada $G = 100$ gerações.

Notação: *AG_Torneio_B*.

Algoritmo 7

Algoritmo Genético com elitismo testado na instância B, com população de $Q = 100$ indivíduos e elite com $Q_{elite} = 2$, taxa de seleção $p_s = 80\%$ com seleção por Roleta Viciada, *crossover* 1-ponto com taxa $p_c = 60\%$, taxa de mutação p_m é uma função logística com $\theta = 0,01$ e critério de parada $G = 100$ gerações.

Notação: *AG_Roleta_B*.

Algoritmo 8

Simulated Annealing testado na instância B, com temperatura inicial $T_{inicial} = 100$ graus, temperatura final $T_{final} = 10^{-4}$ graus, número de iterações por temperatura $L = 24$ e taxa de resfriamento $\alpha = 0,95$.

Notação: *SA_10⁻⁴_B*.

Algoritmo 9

Simulated Annealing testado na instância B, com temperatura inicial $T_{inicial} = 100$ graus, temperatura final $T_{final} = 10^{-6}$ graus, número de iterações por temperatura $L = 24$ e taxa de resfriamento $\alpha = 0,95$.

Notação: *SA_10⁻⁶_B*.

Algoritmo 10

Algoritmo híbrido de Genético com *Simulated Annealing* que utiliza a melhor solução gerada pelo Algoritmo 7 como solução inicial para refinamento através do Algoritmo 8.

Notação: *AG+SA_Roleta_B*.

Cada estratégia, em cada uma das instâncias propostas, foram simuladas 100 vezes e os resultados médios são apresentados no próximo Capítulo.

4 RESULTADOS E DISCUSSÕES

São apresentados neste Capítulo os resultados computacionais para o modelo proposto, obtidos por meio dos algoritmos destacados anteriormente. Os resultados foram obtidos da média de 100 simulações de cada algoritmo para as instâncias estabelecidas, e as melhores soluções foram comparadas. Dentre elas, uma é apresentada como solução ideal de planejamento.

As Tabelas (6) e (7) mostram os resultados médios obtidos pelos algoritmos nas instâncias A e B, respectivamente.

Tabela 6. Valor da função objetivo e tempo computacional (em segundos) médios para os algoritmos testados na instância A

Algoritmo	Função Objetivo	Tempo Computacional
<i>AG_Torneio_A</i>	33,2	25,84
<i>AG_Roleta_A</i>	32,4	28,30
<i>SA_{10⁻⁴}_A</i>	30,5	25,48
<i>AG+SA_Torneio_A</i>	29,3	38,17
<i>AG+SA_Roleta_A</i>	28,8	41,47

Para a instância A, observa-se na Tabela (6) que, dentre os Algoritmos Genéticos *AG_Torneio_A* e *AG_Roleta_A*, o segundo, que utiliza seleção por Roleta Viciada, apresentou em média melhores resultados para a função objetivo, com custo computacional cerca de 10% maior. Entre os métodos puros *AG_Torneio_A*, *AG_Roleta_A* e *SA_{10⁻⁴}_A*, o último apresentou melhores resultados para a função

Tabela 7. Valor da função objetivo e tempo computacional (em segundos) médios para os algoritmos testados na instância B

Algoritmo	Função Objetivo	Tempo Computacional
<i>AG_Torneio_B</i>	62,05	77,87
<i>AG_Roleta_B</i>	59,93	81,54
<i>SA_{10⁻⁴}_B</i>	71,81	62,03
<i>SA_{10⁻⁶}_B</i>	70,58	72,27
<i>AG+SA_Roleta_B</i>	59,09	104,07

objetivo e custo computacional.

Para os métodos híbridos testados na instância A (*AG+SA_Torneio_A* e *AG+SA_Roleta_A*), apesar de maior custo computacional (cerca de 10%), o segundo, que utiliza seleção por Roleta Viciada, apresentou em média melhores resultados para o valor da função objetivo.

Para a instância B, testou-se os Algoritmos Genéticos *AG_Torneio_B* e *AG_Roleta_B*, variando-se apenas os métodos de seleção. Dentre eles, aquele que utilizou seleção por Roleta Viciada apresentou melhores valores para a função objetivo do problema, como mostra a Tabela (7).

Nota-se que, para ambas instâncias testadas, o Algoritmo Genético com seleção por Roleta Viciada apresentou melhores resultados que o método com seleção por Torneio Binário. A partir desta informação, testou-se na instância B apenas a hibridização do Algoritmo Genético com *Simulated Annealing* utilizando a seleção por Roleta Viciada (*AG+SA_Roleta_B*). Comparando-se o AG puro de melhor resultado, ou seja, aquele com seleção por Roleta Viciada, e o algoritmo híbrido também com este método de seleção, verifica-se que, em média, o refinamento com o *Simulated Annealing* proporciona alguma melhora para os valores da função objetivo do problema.

Ainda na Tabela (7), observa-se que o algoritmo *SA_{10⁻⁴}_B* apresen-

tou os piores resultados para a função objetivo, quando comparado com os demais métodos testados na instância B. Contudo, o tempo computacional exigido por este algoritmo foi, em média, o mais baixo. Por este motivo, alterou-se o critério de parada do algoritmo, diminuindo a temperatura final de 10^{-6} para 10^{-6} , gerando um novo algoritmo, o $SA_{10^{-6}}B$. Com esta modificação, aumentou-se o número de iterações do algoritmo, o que proporcionou uma pequena melhora nos valores da função objetivo do problema, como mostra a Tabela (7).

De modo geral, os algoritmos híbridos apresentaram melhores valores médios para a função objetivo do problema avaliada nas instâncias A e B, em especial, os híbridos com seleção por Roleta Viciada.

Com relação à convergência dos métodos para soluções factíveis, os algoritmos testados na instância A obtiveram 100% de soluções factíveis para o problema. Já na instância B, de maior porte que a instância A, alguns métodos apresentaram infactibilidade nas soluções encontradas. A Tabela (8) mostra o número de soluções factíveis encontradas para esta instância em 100 simulações.

Tabela 8. Número de soluções factíveis encontradas pelos algoritmos testados na instância B

Algoritmo	Soluções factíveis
$AG_Torneio_B$	60
AG_Roleta_B	54
$SA_{10^{-4}}B$	100
$SA_{10^{-6}}B$	100
$AG+SA_Roleta_B$	--

Observa-se na Tabela (8) que o algoritmo *Simulated Annealing* manteve a factibilidade das soluções em 100% das simulações para a instância B. Já o Algoritmo Genético nem sempre apresentou resultados factíveis ao final de sua execução. Este fato impossibilitou a análise de factibilidade do algoritmo híbrido

$AG + SA_Roleta_B$, devido ao fato de que a sua execução completa só é possível se o AG resultar uma solução factível para que o SA possa entrar em ação, refinando a busca. Em geral, para o algoritmo híbrido, quando o AG resultou uma solução factível para o SA, este também gerou uma solução final factível.

As Tabela (9) e (10) mostram a homogeneidade das soluções e desempenho dos algoritmos avaliados nas instâncias A e B, respectivamente.

Tabela 9. Homogeneidade e desempenho dos algoritmos obtidos para a instância A com relação ao valor da função objetivo

Algoritmo	Média	DP	CV(%)	Máximo	Mínimo
<i>AG_Torneio_A</i>	33,2	1,93	5,82	40,18	28,87
<i>AG_Roleta_A</i>	32,4	2,43	7,49	45,49	28,46
<i>SA_10⁻⁴_A</i>	30,5	4,71	15,42	41,40	23,98
<i>AG+SA_Torneio_A</i>	29,3	3,10	10,57	36,35	23,67
<i>AG+SA_Roleta_A</i>	28,8	2,84	9,84	35,18	22,54

Tabela 10. Homogeneidade e desempenho dos algoritmos obtidos para a instância B com relação ao valor da função objetivo

Algoritmo	Média	DP	CV(%)	Máximo	Mínimo
<i>AG_Torneio_B</i>	62,05	6,28	10,13	84,65	52,21
<i>AG_Roleta_B</i>	59,93	5,01	8,37	73,55	52,24
<i>SA_10⁻⁴_B</i>	71,81	14,97	20,85	93,33	48,14
<i>SA_10⁻⁶_B</i>	72,27	14,63	20,24	101,09	47,66
<i>AG+SA_Roleta_B</i>	59,09	5,40	9,14	77,17	47,93

A homogeneidade analisada nos dados está relacionada ao Coeficiente de Variação (CV), dado em porcentagem, e o desempenho com as piores e melhores

soluções obtidas, denominadas Máximo e Mínimo, respectivamente, encontradas pelos algoritmos testados.

O Coeficiente de Variação fornece a variação dos dados obtidos em relação à média. Quanto menor for o seu valor, mais homogêneos são os dados. O cálculo do CV é dado pela Equação (17).

$$CV = 100 \frac{DP}{\bar{X}}, \quad (17)$$

sendo DP o desvio padrão e $\bar{X}$ a média dos dados.

Observa-se nas Tabelas (9) e (10) que o *Simulated Annealing* foi o método que apresentou maiores valores para o coeficiente de variação, e portanto, as soluções são menos homogêneas, para ambas as instâncias. Esta característica observada no SA pode ser atribuída ao fato de o método ser um algoritmo de busca local, que encontra boas soluções quando em regiões promissoras, mas que pode cair em ótimos locais e dali não conseguir escapar.

Em geral, os AGs apresentaram resultados mais homogêneos para a função objetivo. Por ser um método baseado em populações, que utiliza vários pontos de exploração no espaço de busca, diversificando as regiões de busca, mas sem a intensificação necessária para determinar melhores soluções.

Com relação ao desempenho dos métodos, a melhor solução que minimiza o problema para a instância A foi encontrada pelo algoritmo híbrido com seleção por Roleta Viciada (*AG+SA_Roleta_A*), enquanto a pior solução foi encontrada pelo AG com Roleta Viciada (*AG_Roleta_A*). Já para a instância B, o algoritmo *SA_10⁻⁶_B* foi o método que encontrou a melhor e a pior solução de minimização do problema.

Fez-se também a análise da homogeneidade e desempenho dos algoritmos para o tempo computacional, apresentadas nas Tabelas (11) e (12).

Em ambas instâncias, o *Simulated Annealing* foi o método que apresentou menor homogeneidade no tempo computacional demandado para a execução do algoritmo, seguido dos algoritmos híbridos, que utilizam o SA para refinamento da melhor solução gerada pelo AG.

Tabela 11. Homogeneidade e desempenho dos algoritmos obtidos para a instância A com relação ao tempo computacional

Algoritmo	Média	DP	CV (%)	Máximo	Mínimo
<i>AG_Torneio_A</i>	25,84	1,17	4,53	31,74	24,41
<i>AG_Roleta_A</i>	28,30	1,17	4,14	33,64	26,51
<i>SA_{10⁻⁴}_A</i>	25,48	1,55	6,09	32,45	22,37
<i>AG+SA_Torneio_A</i>	38,21	2,30	6,01	45,57	34,57
<i>AG+SA_Roleta_A</i>	41,47	2,05	4,94	51,38	37,83

Tabela 12. Homogeneidade e desempenho dos algoritmos obtidos para a instância B com relação ao tempo computacional

Algoritmo	Média	DP	CV (%)	Máximo	Mínimo
<i>AG_Torneio_B</i>	77,87	3,36	4,32	86,36	71,64
<i>AG_Roleta_B</i>	81,54	3,01	3,69	90,61	75,91
<i>SA_{10⁻⁴}_B</i>	62,02	11,20	18,05	89,19	43,84
<i>SA_{10⁻⁶}_B</i>	70,58	3,57	5,06	78,71	58,27
<i>AG+SA_Roleta_B</i>	104,07	3,04	2,99	115,41	97,31

Os métodos híbridos demandaram tempo de execução cerca de 50% superior com relação aos métodos puros, o que era esperado, visto que estes são combinações de métodos puros. Contudo, por apresentarem melhoras significativas na solução do problema, o tempo computacional acrescido em segundos permanece aceitável.

As Figuras (27) e (28) apresentam de forma gráfica o desempenho dos métodos apresentados nas Tabelas (9) e (11), para a instância A.

Observa-se na Figura (27) uma maior homogeneidade das soluções en-

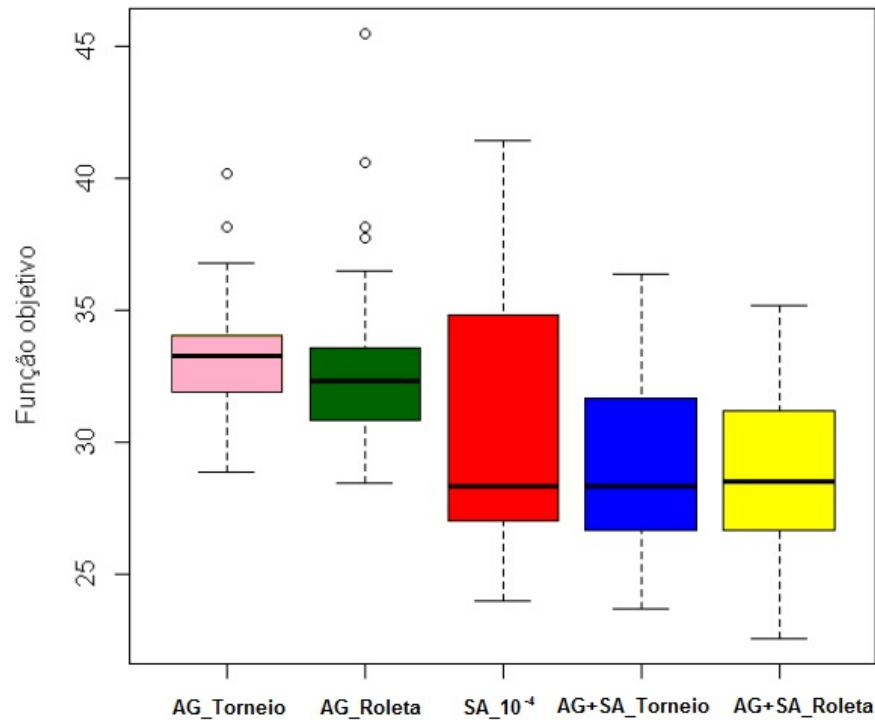


Figura 27 - Boxplot com resultados de homogeneidade e desempenho dos algoritmos testados na instância A para valores da função objetivo

contradas pelos Algoritmos Genéticos puros *AG_Torneio_A* e *AG_Roleta_A*, a menor homogeneidade nas soluções encontradas pelo algoritmo *SA_10⁻⁴_A* e a melhor solução apresentada pelo algoritmo *AG+SA_Roleta_A*. Já na Figura (28), nota-se que o algoritmo *SA_10⁻⁴_A* é o que demanda menor tempo de execução entre os métodos testados, em especial, dentre os métodos puros *AG_Torneio_A*, *AG_Roleta_A* e *SA_10⁻⁴_A*.

As Figuras (29) e (30) apresentam de forma gráfica o desempenho dos métodos apresentados nas Tabelas (9) e (11), para a instância B.

Na Figura (29) pode-se visualizar a menor homogeneidade apresentada pelos algoritmos *Simulated Annealing*, além do bom desempenho apresentado pelos Algoritmos Genéticos puro e híbrido que utilizam a seleção por Roleta Viciada para valores da função objetivo.

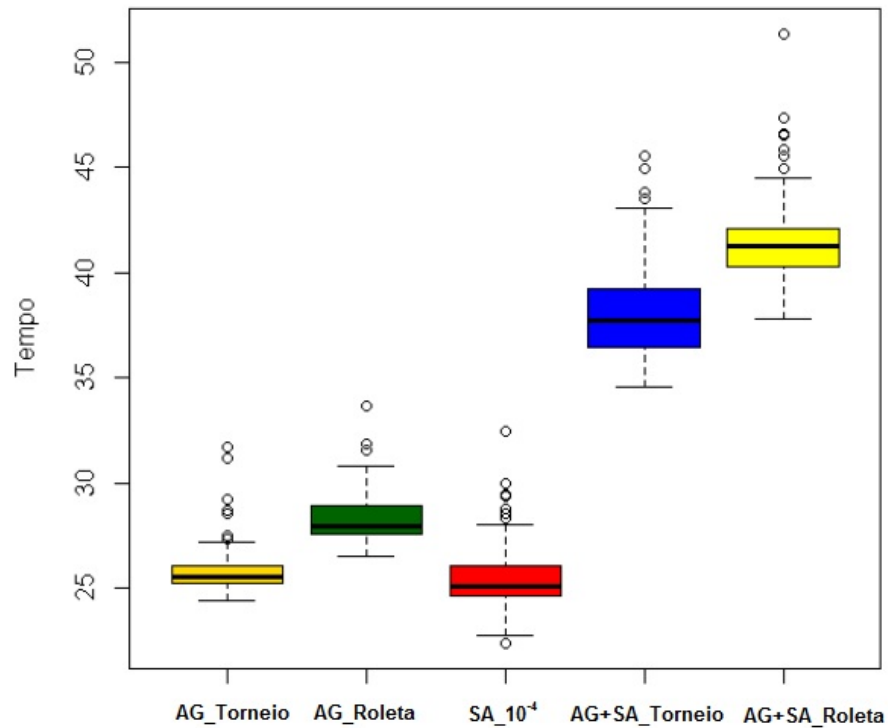


Figura 28 - Boxplot com resultados de homogeneidade e desempenho dos algoritmos testados na instância A para o tempo computacional

Observa-se na Figura (30) que o algoritmo *Simulated Annealing* com temperatura final de 10^{-4} foi o método que apresentou menor homogeneidade no tempo computacional demandado nas simulações da instância B. Além disso, os Algoritmos Genéticos se comportam de maneira bem próxima com relação ao tempo computacional, enquanto o algoritmo híbrido possui o tempo computacional médio superior aos demais métodos.

As melhores configurações de plantio apresentadas pelos algoritmos em 100 simulações são mostradas nas Figuras de (31) a (40).

A Figura (31) mostra a melhor configuração de plantio obtida pelo Algoritmo Genético com seleção por Torneio Binário para a instância A (*AG_Torneio_A*), com valor da função objetivo igual a 28,87 e tempo computacional de 24,41 segundos.

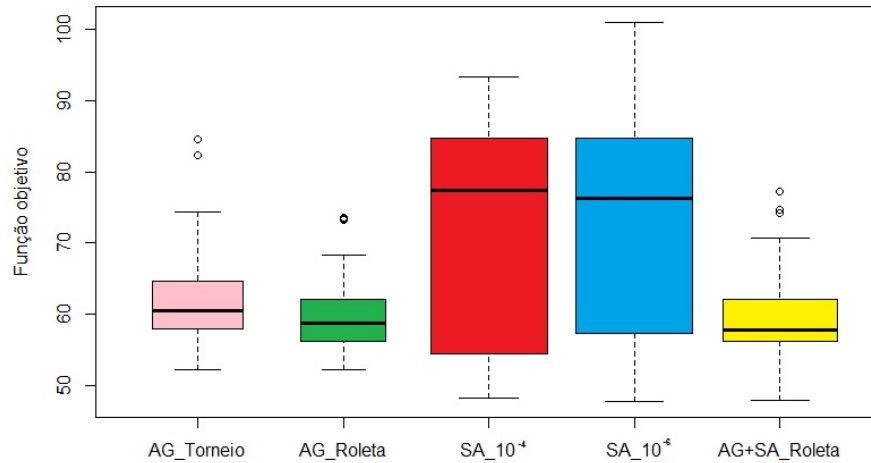


Figura 29 - Boxplot com resultados de homogeneidade e desempenho dos algoritmos testados na instância B para valores da função objetivo

A Figura (32) mostra a melhor configuração de plantio obtida pelo Algoritmo Genético com seleção por Roleta Viciada para a instância A (*AG_Roleta_A*), com valor da função objetivo igual a 28,46 e tempo computacional de 29,35 segundos.

A Figura (33) mostra a melhor configuração de plantio obtida pelo *Simulated Annealing* com temperatura final de 10^{-4} para a instância A (*SA_10⁻⁴_A*), com valor da função objetivo igual a 23,98 e tempo computacional de 26,22 segundos.

A Figura (34) mostra a melhor configuração de plantio obtida pelo algoritmo híbrido AG+SA com seleção por Torneio Binário para a instância A (*AG+SA_Torneio_A*), com valor da função objetivo igual a 23,67 e tempo computacional de 35,66 segundos.

A Figura (35) mostra a melhor configuração de plantio obtida pelo algoritmo híbrido AG+SA com seleção por Roleta Viciada para a instância A (*AG+SA_Roleta_A*), com valor da função objetivo igual a 22,54 e tempo computacional de 38,06 segundos.

A Figura (36) apresenta a melhor configuração de plantio obtida pelo Algoritmo Genético com seleção por Torneio Binário para a instância B

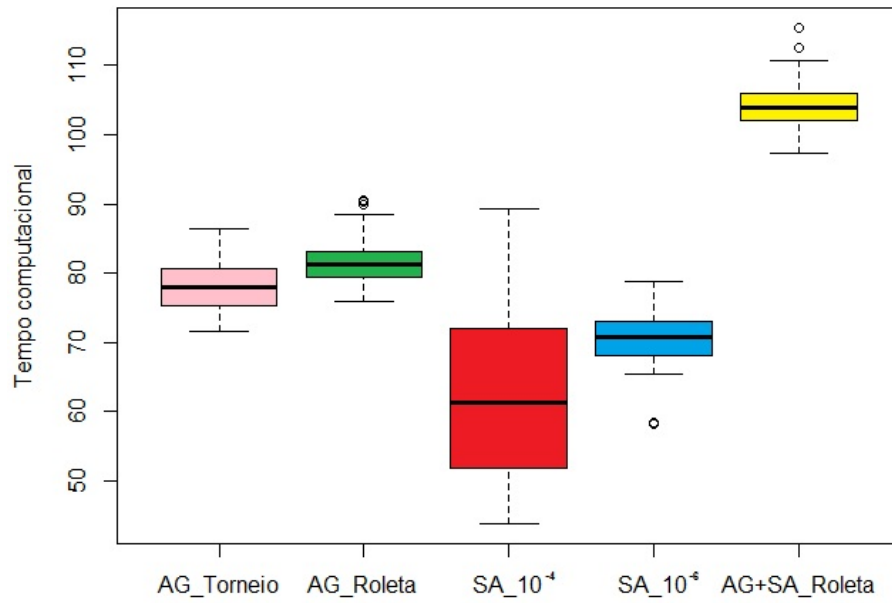


Figura 30 - Boxplot com resultados de homogeneidade e desempenho dos algoritmos testados na instância B para o tempo computacional

(*AG_Torneio_B*), com valor da função objetivo igual a 52,21 e tempo computacional de 73,50 segundos.

A Figura (37) apresenta a melhor configuração de plantio obtida pelo Algoritmo Genético com seleção por Roleta Viciada para a instância B (*AG_Roleta_B*), com valor da função objetivo igual a 52,24 e tempo computacional de 78,91 segundos.

A Figura (38) apresenta a melhor configuração de plantio obtida pelo *Simulated Annealing* com temperatura final de 10^{-4} para a instância B (*SA_10⁻⁴_B*), com valor da função objetivo igual a 48,14 e tempo computacional de 60,61 segundos.

A Figura (39) apresenta a melhor configuração de plantio obtida pelo *Simulated Annealing* com temperatura final de 10^{-6} para a instância B (*SA_10⁻⁶_B*), com valor da função objetivo igual a 47,66 e tempo computacional de 67,23 segundos.

A Figura (40) apresenta a melhor configuração de plantio obtida pelo algoritmo híbrido AG+SA com seleção por Roleta Viciada para a instância B

	Mês 1	Mês 2	Mês 3	Mês 4	Mês 5	Mês 6	Mês 7	Mês 8	Mês 9	Mês 10	Mês 11	Mês 12	Mês 13	Mês 14	Mês 15	Mês 16	Mês 17	Mês 18	Mês 19	Mês 20	Mês 21	Mês 22	Mês 23	Mês 24
Lote 1	14	14	14	14	14	14	14	14	20	20	20	20	20	16	16	16	16	16	16	11	11	11	11	11
Lote 2	14	14	14	14	14	14	14	14	4	4	4	4	4	4	4	4	18	18	18	18	18	24	24	24
Lote 3	16	16	16	16	16	16	21	21	21	21	21	21	16	16	16	16	16	16	12	12	12	12	12	12
Lote 4	15	15	15	20	20	20	20	20	24	24	24	11	11	11	11	11	14	14	14	14	14	14	14	14
Lote 5	19	19	19	19	19	19	16	16	16	16	16	16	13	13	13	13	18	18	18	18	18	15	15	15
Lote 6	12	12	12	12	12	12	18	18	18	18	18	10	10	10	10	10	4	4	4	4	4	4	4	4

Figura 31 - Melhor configuração de plantio apresentada pelo algoritmo *AG_Torneio_A*

	Mês 1	Mês 2	Mês 3	Mês 4	Mês 5	Mês 6	Mês 7	Mês 8	Mês 9	Mês 10	Mês 11	Mês 12	Mês 13	Mês 14	Mês 15	Mês 16	Mês 17	Mês 18	Mês 19	Mês 20	Mês 21	Mês 22	Mês 23	Mês 24
Lote 1	12	12	12	12	12	12	1	1	4	4	4	4	4	4	4	4	16	16	16	16	16	16	1	1
Lote 2	4	4	4	4	4	4	4	4	14	14	14	14	14	14	14	14	23	23	23	17	17	17	17	17
Lote 3	14	14	14	14	14	14	14	14	2	2	2	2	13	13	13	13	15	15	15	10	10	10	10	10
Lote 4	3	3	3	11	11	11	11	11	14	14	14	14	14	14	14	14	4	4	4	4	4	4	4	4
Lote 5	7	7	21	21	21	21	21	21	4	4	4	4	4	4	4	4	12	12	12	12	12	12	8	8
Lote 6	18	18	18	18	18	12	12	12	12	12	12	17	17	17	17	17	14	14	14	14	14	14	14	14

Figura 32 - Melhor configuração de plantio apresentada pelo algoritmo *AG_Roleta_A*

(*AG+SA_Roleta_B*), com valor da função objetivo igual a 47,93 e tempo computacional de 103,14 segundos.

Destacam-se nas Figuras (41), (42) e (43) as curvas de melhora obtidas pelos algoritmos híbridos *AG+SA_Torneio_A*, *AG+SA_Roleta_A* e *AG+SA_Roleta_B* no desenvolvimento das melhores soluções, respectivamente.

Como visto, o melhor resultado bruto foi obtido pelo algoritmo *AG+SA_Roleta_A* para a função objetivo dentre os métodos simulados para a instância A, que apresenta características de hibridização do Algoritmo Genético com *Simulated Annealing* e seleção por Roleta Viciada. Para a instância B, o melhor resultado bruto foi apresentado pelo método puro *Simulated Annealing* com temperatura final de 10^{-6} graus.

Os resultados computacionais obtidos condizem com os resultados da literatura que demonstram que o método da Roleta Viciada possui desempenho superior ao Torneio Binário, e que em média métodos híbridos apresentam melhores resultados em comparação aos métodos puros.

	Mês 1	Mês 2	Mês 3	Mês 4	Mês 5	Mês 6	Mês 7	Mês 8	Mês 9	Mês 10	Mês 11	Mês 12	Mês 13	Mês 14	Mês 15	Mês 16	Mês 17	Mês 18	Mês 19	Mês 20	Mês 21	Mês 22	Mês 23	Mês 24
Lote 1	12	12	12	12	12	12	16	16	16	16	16	16	24	24	24	13	13	13	13	20	20	20	20	20
Lote 2	23	23	23	10	10	10	10	10	4	4	4	4	4	4	4	4	14	14	14	14	14	14	14	14
Lote 3	21	21	21	21	21	21	7	7	16	16	16	16	16	16	2	2	2	2	21	21	21	21	21	21
Lote 4	4	4	4	4	4	4	4	4	14	14	14	14	14	14	14	14	4	4	4	4	4	4	4	4
Lote 5	12	12	12	12	12	12	21	21	21	21	21	21	12	12	12	12	12	12	12	12	12	12	12	12
Lote 6	6	6	6	6	6	14	14	14	14	14	14	14	14	11	11	11	11	11	19	19	19	19	19	19

Figura 33 - Melhor configuração de plantio apresentada pelo algoritmo $SA_{10^{-4}}_A$

	Mês 1	Mês 2	Mês 3	Mês 4	Mês 5	Mês 6	Mês 7	Mês 8	Mês 9	Mês 10	Mês 11	Mês 12	Mês 13	Mês 14	Mês 15	Mês 16	Mês 17	Mês 18	Mês 19	Mês 20	Mês 21	Mês 22	Mês 23	Mês 24
Lote 1	16	16	16	16	16	16	2	2	2	2	21	21	21	21	21	21	12	12	12	12	12	12	1	1
Lote 2	4	4	4	4	4	4	4	4	9	9	9	9	9	3	3	3	14	14	14	14	14	14	14	14
Lote 3	21	21	21	21	21	21	8	8	16	16	16	16	16	16	21	21	21	21	21	21	13	13	13	13
Lote 4	4	4	4	4	4	4	4	4	14	14	14	14	14	14	14	14	4	4	4	4	4	4	4	4
Lote 5	14	14	14	14	14	14	14	14	12	12	12	12	12	12	12	16	16	16	16	16	16	13	13	13
Lote 6	7	7	19	19	19	19	19	19	14	14	14	14	14	14	14	14	4	4	4	4	4	4	4	4

Figura 34 - Melhor configuração de plantio apresentada pelo algoritmo $AG+SA_{Torneio}_A$

	Mês 1	Mês 2	Mês 3	Mês 4	Mês 5	Mês 6	Mês 7	Mês 8	Mês 9	Mês 10	Mês 11	Mês 12	Mês 13	Mês 14	Mês 15	Mês 16	Mês 17	Mês 18	Mês 19	Mês 20	Mês 21	Mês 22	Mês 23	Mês 24
Lote 1	4	4	4	4	4	4	4	4	2	2	2	2	16	16	16	16	16	16	12	12	12	12	12	12
Lote 2	14	14	14	14	14	14	14	14	4	4	4	4	4	4	4	4	14	14	14	14	14	14	14	14
Lote 3	16	16	16	16	16	16	21	21	21	21	21	21	16	16	16	16	16	16	12	12	12	12	12	12
Lote 4	12	12	12	12	12	12	16	16	16	16	16	16	12	12	12	12	12	12	19	19	19	19	19	19
Lote 5	4	4	4	4	4	4	4	4	14	14	14	14	14	14	14	14	4	4	4	4	4	4	4	4
Lote 6	14	14	14	14	14	14	14	14	9	9	9	9	9	4	4	4	4	4	4	4	4	24	24	24

Figura 35 - Melhor configuração de plantio apresentada pelo algoritmo $AG+SA_{Roleta}_A$

lote K1	14	14	14	14	14	14	14	14	9	9	9	9	9	24	24	24	4	4	4	4	4	4	4	4
lote K2	14	14	14	14	14	14	14	14	4	4	4	4	4	4	4	4	14	14	14	14	14	14	14	14
lote K3	14	14	14	14	14	14	14	14	12	12	12	12	12	12	7	7	4	4	4	4	4	4	4	4
lote K4	6	6	6	6	6	21	21	21	21	21	21	21	17	17	17	17	7	7	16	16	16	16	16	16
lote K5	16	16	16	16	16	16	18	18	18	18	18	9	9	9	9	9	14	14	14	14	14	14	14	14
lote K6	14	14	14	14	14	14	14	14	4	4	4	4	4	4	4	4	1	1	12	12	12	12	12	12
lote K7	20	20	20	20	20	18	18	18	18	18	12	12	12	12	12	12	14	14	14	14	14	14	14	14
lote K8	17	17	17	17	17	14	14	14	14	14	14	14	23	23	23	4	4	4	4	4	4	4	4	4
lote K9	21	21	21	21	21	10	10	10	10	10	10	20	20	20	20	20	4	4	4	4	4	4	4	4
lote K10	19	19	19	19	19	19	20	20	20	20	20	15	15	15	19	19	19	19	19	19	2	2	2	2

Figura 36 - Melhor configuração de plantio apresentada pelo algoritmo $AG_{Torneio}_B$

lote K1	4	4	4	4	4	4	4	4	23	23	23	18	18	18	18	18	4	4	4	4	4	4	4	4	4
lote K2	8	8	14	14	14	14	14	14	14	14	16	16	16	16	16	16	14	14	14	14	14	14	14	14	14
lote K3	6	6	6	6	6	21	21	21	21	21	21	18	18	18	18	18	4	4	4	4	4	4	4	4	4
lote K4	12	12	12	12	12	12	2	2	2	2	21	21	21	21	21	21	7	7	2	2	2	2	7	7	7
lote K5	19	19	19	19	19	19	6	6	6	6	6	9	9	9	9	9	14	14	14	14	14	14	14	14	14
lote K6	16	16	16	16	16	16	12	12	12	12	12	12	21	21	21	21	21	12	12	12	12	12	12	12	12
lote K7	4	4	4	4	4	4	4	4	23	23	23	4	4	4	4	4	4	4	4	18	18	18	18	18	18
lote K8	21	21	21	21	21	21	13	13	13	13	16	16	16	16	16	16	14	14	14	14	14	14	14	14	14
lote K9	21	21	21	21	21	21	4	4	4	4	4	4	4	4	4	4	1	1	14	14	14	14	14	14	14
lote K10	12	12	12	12	12	12	14	14	14	14	14	14	14	14	14	4	4	4	4	4	4	4	4	4	7

Figura 37 - Melhor configuração de plantio apresentada pelo algoritmo *AG_Roleta_B*

lote K1	24	24	24	9	9	9	9	9	4	4	4	4	4	4	4	4	14	14	14	14	14	14	14	14	14
lote K2	12	12	12	12	12	12	18	18	18	18	18	9	9	9	9	9	4	4	4	4	4	4	4	4	4
lote K3	21	21	21	21	21	21	10	10	10	10	10	17	17	17	17	17	14	14	14	14	14	14	14	14	14
lote K4	14	14	14	14	14	14	14	14	2	2	2	2	16	16	16	16	16	16	12	12	12	12	12	12	12
lote K5	16	16	16	16	16	16	20	20	20	20	20	4	4	4	4	4	4	17	17	17	17	17	17	17	17
lote K6	4	4	4	4	4	4	4	4	16	16	16	16	16	16	16	13	13	13	13	21	21	21	21	21	21
lote K7	21	21	21	21	21	21	10	10	10	10	10	4	4	4	4	4	4	4	10	10	10	10	10	10	10
lote K8	19	19	19	19	19	19	16	16	16	16	16	16	21	21	21	21	21	21	12	12	12	12	12	12	12
lote K9	21	21	21	21	21	21	12	12	12	12	12	12	21	21	21	21	21	21	16	16	16	16	16	16	16
lote K10	14	14	14	14	14	14	14	14	4	4	4	4	4	4	4	4	14	14	14	14	14	14	14	14	14

Figura 38 - Melhor configuração de plantio apresentada pelo algoritmo *SA_{10⁻⁴}_B*

lote K1	4	4	4	4	4	4	4	4	10	10	10	10	10	23	23	23	14	14	14	14	14	14	14	14	14
lote K2	12	12	12	12	12	12	7	7	2	2	2	2	21	21	21	21	21	12	12	12	12	12	12	12	12
lote K3	4	4	4	4	4	4	4	4	14	14	14	14	14	14	14	14	4	4	4	4	4	4	4	4	4
lote K4	14	14	14	14	14	14	14	14	4	4	4	4	4	4	4	4	14	14	14	14	14	14	14	14	14
lote K5	21	21	21	21	21	21	12	12	12	12	12	12	12	16	16	16	16	16	16	19	19	19	19	19	19
lote K6	7	7	19	19	19	19	19	19	4	4	4	4	4	4	4	4	14	14	14	14	14	14	14	14	14
lote K7	1	1	12	12	12	12	12	12	16	16	16	16	16	16	2	2	2	2	19	19	19	19	19	19	19
lote K8	14	14	14	14	14	14	14	14	20	20	20	20	20	9	9	9	9	9	21	21	21	21	21	21	21
lote K9	4	4	4	4	4	4	4	4	10	10	10	10	10	21	21	21	21	21	21	6	6	6	6	6	6
lote K10	14	14	14	14	14	14	14	14	20	20	20	20	20	18	18	18	18	18	16	16	16	16	16	16	16

Figura 39 - Melhor configuração de plantio apresentada pelo algoritmo *SA_{10⁻⁶}_B*

lote K1	19	19	19	19	19	19	10	10	10	10	10	18	18	18	18	18	14	14	14	14	14	14	14	14	14
lote K2	12	12	12	12	12	12	21	21	21	21	21	21	2	2	2	2	4	4	4	4	4	4	4	4	4
lote K3	16	16	16	16	16	16	19	19	19	19	19	19	22	22	22	22	14	14	14	14	14	14	14	14	14
lote K4	19	19	19	19	19	19	16	16	16	16	16	16	21	21	21	21	21	21	2	2	2	2	7	7	7
lote K5	14	14	14	14	14	14	14	14	3	3	3	9	9	9	9	9	14	14	14	14	14	14	14	14	14
lote K6	21	21	21	21	21	21	2	2	2	2	2	12	12	12	12	12	12	12	16	16	16	16	16	16	1
lote K7	12	12	12	12	12	12	18	18	18	18	18	6	6	6	6	6	14	14	14	14	14	14	14	14	14
lote K8	19	19	19	19	19	19	2	2	2	2	7	7	16	16	16	16	16	16	21	21	21	21	21	21	21
lote K9	12	12	12	12	12	12	19	19	19	19	19	19	22	22	22	22	14	14	14	14	14	14	14	14	14
lote K10	19	19	19	19	19	19	21	21	21	21	21	21	22	22	22	22	21	21	21	21	21	21	21	8	8

Figura 40 - Melhor configuração de plantio apresentada pelo algoritmo *AG+SA_Roleta_B*

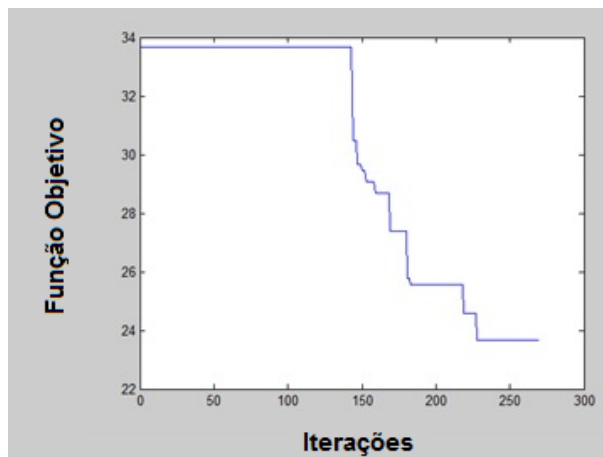


Figura 41 - Curva de melhora realizada pelo SA iniciado com a melhor solução factível do AG com seleção por Torneio Binário para a instância A

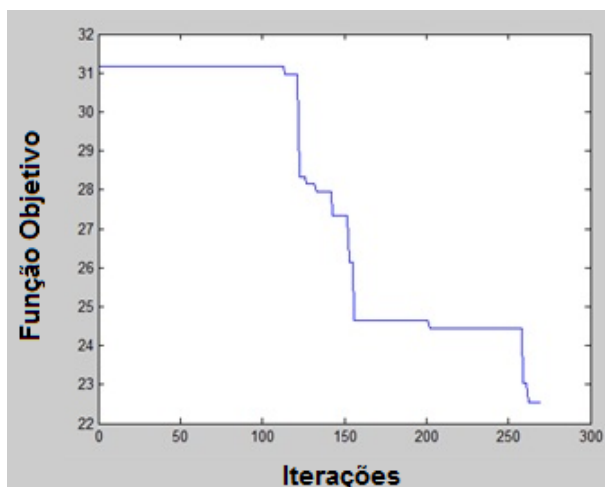


Figura 42 - Curva de melhora realizada pelo SA iniciado com a melhor solução factível do AG com seleção por Roleta Viciada para a instância A

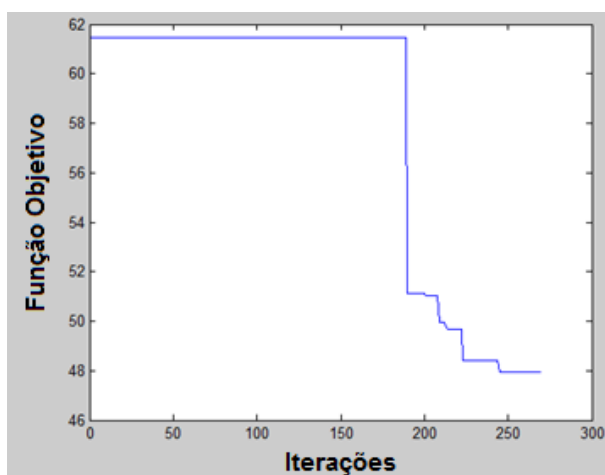


Figura 43 - Curva de melhora realizada pelo SA iniciado com a melhor solução factível do AG com seleção por Roleta Viciada para a instância B

5 CONCLUSÕES

Neste trabalho foi proposto um modelo matemático para o planejamento de plantio de culturas, visando reduzir a proliferação de pragas entre as culturas, respeitando as necessidades agrônômicas e de demanda sazonal de cada cultura.

O processo de Modelagem Matemática consistiu em definir o problema real encontrado na agricultura e recolher informações que pudessem ajudar a levantar hipóteses com o objetivo de elaborar um modelo que represente a situação da melhor forma possível. Compreendido o problema, o primeiro passo foi selecionar parâmetros e variáveis essenciais para formulação do modelo matemático (o objetivo e as restrições envolvidas no problema de otimização). Após a formulação do modelo, estudou-se as estratégias de resolução, sistematizou-se os conceitos envolvidos e resolveu-se o modelo, obtendo soluções e interpretações gráficas e analíticas. A modelagem matemática detalhada para o problema de planejamento do plantio de culturas com lotes de diferentes áreas e geometrias foi apresentado na Seção 3.1.

Tendo em vista que este é um problema de natureza combinatorial, foram desenvolvidas e implementadas metaheurísticas de otimização para resolução do modelo. Inicialmente desenvolveu-se uma heurística construtiva para construir soluções que respeitam as restrições mais difíceis de serem satisfeitas. Em seguida, implementou-se as estratégias metaheurísticas Algoritmo Genético (AG), *Simulated Annealing* (SA) e híbrida AG+AS.

Para as instâncias A e B, propostas na Seção 3.7, com 10 e 15 lotes, respectivamente, e informações sobre 24 culturas, disponíveis no Apêndice, todos os algoritmos apresentaram resultados capazes de auxiliar os agricultores na elaboração

de planejamentos de plantio das culturas no período estabelecido.

Bons resultados foram obtidos com a utilização de algoritmos híbridos e conclui-se que estes métodos são apropriados para a resolução de problemas de otimização complexos, que envolvem muitas variáveis e um espaço de busca de dimensão elevada. Destaca-se a hibridização dos algoritmos Genético e *Simulated Annealing*, métodos estes considerados robustos por abrangerem um grande número de aplicações.

A vantagem da modelagem matemática apresentada neste trabalho é que esta proporciona um planejamento das atividades de plantio que não requer gastos com recursos materiais e pode ser complementado com controle biológico e defensivos agrícolas, ajudando a reduzir custos para o agricultor e danos ambientais.

Por fim, por ser a natureza vulnerável à intervenção humana, deve-se ter cuidado com as ações sobre os agroecossistemas, especialmente com relação à utilização de agrotóxicos, e estudos de formas de prevenção, como o realizado neste trabalho, devem ser incentivados.

REFERÊNCIAS BIBLIOGRÁFICAS

ALIANO FILHO, A. Metaheurísticas em um Problema de Rotação de Culturas. Botucatu, 2012. Dissertação (Mestrado) - Universidade Estadual Paulista.

ANDRÉASSON, N.; EVGRAFOV, A.; PATRIKSSON, M. **An Introduction to Continuous Optimization: Foundations and Fundamental Algorithms**. Lund, Sweden: Studentlitteratur, 2005.

BASSANEZI, R. C. **Ensino-aprendizagem com modelagem matemática**. São Paulo: Contexto, 2002.

BERNARDON, T.; CALGARO, M. Modelo de simulação de crescimento de plantas. **Ambiência**, v.3, n.2, p.283–294, 2007.

BIEMBENGUT, M. S.; HEIN, N. **Modelagem matemática no ensino**. São Paulo: Contexto, 2000.

BRANDÃO, M. A. L.; SARAMAGO, S. F. P. **Métodos Estocásticos de Otimização: Algoritmos Genéticos e Evolução Diferencial**. São Carlos, SP: SBMAC, 2011. 55v.

BREGALDA, P. F.; OLIVEIRA, A. A. F.; BORNSTEIN, C. T. **Introdução à Programação Linear**. Rio de Janeiro: Campus, 1988. 329p.

DARWIN, C. **A Origem das Espécies e a seleção natural**. São Paulo, 2004: Madras, 1859.

DAVIS, L. **Handbook of Genetic Algorithms**. New York: Van Nostrand Reinhold, 1991.

- DE JONG, K.; SARMA, J. **On decentralizing selection algorithms**. San Francisco, CA, 1995.
- FEY, E.; SANTOS, S. R.; SANTA CATARINA, A.; PEREIRA, J. O. **Planejamento de um sistema agrícola utilizando Programação Linear**. Ponta Grossa, PR, 2000.
- FUTUYMA, D. J. **Evolução, ciência e sociedade**. São Paulo: SBG, 2002. 73p.
- GLOVER, F.; KOCHENBERGER, G. **Handbook of Metaheuristics**. Kluwer, 2003.
- GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos**. Rio de Janeiro: Elsevier, 2005.
- GOLDBERG, D. **Genetic Algorithms in Search, Optimization and Machine Learning**. Addison Wesley Longman, 1989.
- GOMES, R. M.; ARENALES, M. N. Otimização Linear Aplicada ao plantio sustentável de vegetais. In: , 2010. ; resumos. Bento Gonçalves, RS: , 2010.
- GOULD, S. J. **Ever Since Darwin**. New York: Norton, 1977. 285p.
- HOCHBAUM, D. **Approximation Algorithms for NP-Hard Problems**. Boston: PWS Publishing, 1996.
- HOLLAND, J. H. **Adaptation in natural and artificial systems**. Ann Arbor: University of Michigan Press, 1975.
- ISHIBUCHI, H.; OHARA, K.; NOJIMA, Y. **Implementation of elitism in cellular genetic algorithms**. Sokcho, Korea, 2007.
- KANTOROVICH, L. V. Mathematical Methods of Organizing and Planning Production. **Management Science**, v.6, n.4, p.366–422, 1960.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by Simulated Annealing. **Science**, v.220, n.4598, p.671–680, 1983.

KOZA, J. R. **Genetic Programming: On the Programming of Computers by Means of Natural Selection**. Cambridge: The MIT Press, 1992. 819p.

LINDEN, R. **Algoritmos Genéticos: Uma importante ferramenta da Inteligência Computacional**. Rio de Janeiro: Brasport, 2008.

LUENBERGER, D. G.; YE, Y. **Linear and nonlinear programming**. New York: Springer, 2008.

MARTINS JR., O. P. **A dinâmica dos agrotóxicos no meio ambiente**. Goiânia, GO, 2002.

MELIÁN, B.; PÉREZ, J. A. M.; VEGA, M. M. Metaheuristics: A Global View. **Revista Iberoamericana de Inteligencia Artificial**, v.2, n.19, p.7–28, 2003.

METROPOLIS, N.; ROSENBLUTH, A.; ROSENBLUTH, M.; TELLER, A.; TELLER, E. Equation of State Calculations by Fast Computing Machines. **Journal of Chemical Physics**, v.21, n.1087, 1953.

MICHALEWICZ, Z. **Genetic Algorithms+Data Structures=Evolution Programs**. New York: Springer-Verlag, 1994.

MITCHELL, M. **An Introduction to Genetic Algorithms**. Cambridge: The MIT Press, 1992. 205p.

MOUJAHD, A.; INZA, I.; LARRAÑAGA, P. Algoritmos Genéticos. **Intelligent Systems Group, University of the Basque Country, San Sebastián, España**, 2007.

OCHI, L. S. Algoritmos Genéticos: Origem e Evolução. **Sociedade Brasileira de Matemática Aplicada e Computacional**, 1998.

PARRA, J. R. P.; BOTELHO, P. S. M.; CORRÊA-FILHO, B. S.; BENTO, J. M. S. **Controle biológico no Brasil: parasitóides e predadores**. São Paulo: Manole, 2002.

SANTOS, L. M. R. Programação de rotação de culturas modelos e métodos de solução. São Carlos, 2009. 86p. Tese (Doutorado) - Universidade de São Paulo.

SANTOS, L. M. R.; SANTOS, R. H.; ARENALES, M. N.; RAGGI, L. A. Um modelo para a programação de rotações de culturas. **Pesquisa Operacional**, v.27, n.3, p.535–547, 2007.

VENZON, M.; PAULA JÚNIOR, T. J.; PALLINI, A. **Avanços no controle alternativo de pragas e doenças**. Viçosa: EPAMIG/CTZM, 2008.

WRIGHT, S. J.; NOCEDAL, J. **Numerical optimization**. New York: Springer-Verlag, 1999.

APÊNDICE

Em todos os experimentos computacionais para o modelo proposto foram utilizadas informações sobre 24 culturas, obtidas de Santos (2009), conforme a Tabela (13), e informações sobre produtividade e demanda destas culturas, obtidas de Aliano Filho (2012), conforme as Tabelas (14) e (15), respectivamente.

Tabela 13: Informações sobre culturas: nome, família, época de plantio e duração do ciclo de vida, adaptado de Santos (2009)

				Época de plantio		Ciclo	
				Início	Fim	dias	c_i
1	Alface	1	<i>Compositae</i>	ano todo		50	2
2	Almerão	1	<i>Compositae</i>	ano todo		120	4
3	Couve	2	<i>Brassicaceae</i>	março	junho	100	3
4	Brócolis	2	<i>Brassicaceae</i>	fevereiro	junho	230	8
5	Repolho	2	<i>Brassicaceae</i>	fevereiro	julho	130	4
6	Couve-Flor	2	<i>Brassicaceae</i>	março	junho	140	5
7	Beterraba	3	<i>Chenopodiaceae</i>	março	julho	70	2
8	Espinafre	3	<i>Chenopodiaceae</i>	fevereiro	julho	50	2
9	Abobrinha	4	<i>Cucurbitaceae</i>	agosto	março	140	5
10	Moranga	4	<i>Cucurbitaceae</i>	setembro	janeiro	150	5
11	Pepino	4	<i>Cucurbitaceae</i>	ano todo		140	5
12	Alho	5	<i>Liliaceae</i>	março	abril	180	6
13	Cebola	5	<i>Liliaceae</i>	março	junho	120	4
14	Quiabo	6	<i>Malvaceae</i>	agosto	março	230	8
15	Milho	7	<i>Gramineae</i>	agosto	abril	100	3
16	Aveia	7	<i>Gramineae</i>	março	maio	180	6
17	Tomate	8	<i>Solanaceae</i>	ano todo		150	5
18	Pimentão	8	<i>Solanaceae</i>	ano todo		160	5
19	Jiló	8	<i>Solanaceae</i>	setembro	fevereiro	190	6
20	Cenoura	9	<i>Umbelliferae</i>	março	julho	140	5
21	Salsinha	9	<i>Umbelliferae</i>	setembro	março	190	6
22	Feijão Vagem	10	<i>Leguminosae</i>	agosto	abril	110	4
23	Ervilha	10	<i>Leguminosae</i>	março	abril	90	3
24	Feijão	10	<i>Leguminosae</i>	agosto	setembro	90	3

Tabela 14: Produtividade mensal por m^2 das culturas, adaptado de Aliano Filho (2012)

[illegible]

Tabela 15: Demanda e períodos de demanda das culturas, adaptado de Aliano Filho (2012)

	Cultura	Época de demanda		Demanda
		Início	Fim	
1	Alface	Dezembro	Março	3.750 pés
2	Almerão	Novembro	Março	6.000 pés
3	Couve	Junho	Setembro	1.667 pés
4	Brócolis	Setembro	Dezembro	1.333 pés
5	Repolho	Agosto	Novembro	1.000 pés
6	Couve-Flor	Agosto	Outubro	1.250 pés
7	Beterraba	Maio	Setembro	3.000 kg
8	Espinafre	Julho	Setembro	5.000 pés
9	Abobrinha	Janeiro	Março	5.000 unidades
10	Moranga	Fevereiro	Junho	4.000 unidades
11	Pepino	Abril	Julho	5.000 unidades
12	Alho	Setembro	Outubro	100 kg
13	Cebola	Agosto	Outubro	875 kg
14	Quiabo	Agosto	Outubro	1.250 kg
15	Milho	Janeiro	Maio	200 kg
16	Aveia	-	-	0 kg
17	Tomate	Maio	Agosto	2.500 kg
18	Pimentão	Maio	Setembro	800 kg
19	Jiló	-	-	0 kg
20	Cenoura	Agosto	Novembro	1.000 kg
21	Salsinha	Abril	Agosto	1.000 maços
22	Feijão Vagem	Dezembro	Abril	400 kg
23	Ervilha	Junho	Julho	125 kg
24	Feijão	Novembro	Dezembro	100 kg