METAHEURÍSTICAS EM UM PROBLEMA DE ROTAÇÃO DE CULTURAS

Angelo Aliano Filho

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

BOTUCATU
São Paulo - Brasil
Fevereiro - 2012

METAHEURÍSTICAS EM UM PROBLEMA DE ROTAÇÃO DE CULTURAS

Angelo Aliano Filho

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

Co-orientadora: Profª. Drª. **Margarida Vaz Pato**

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

BOTUCATU
São Paulo - Brasil
Fevereiro - 2012

Ficha Catalográfica

Dedicatória

Aos meu queridos pais,

Angelo e Loide Aliano.

Dedico.

Agradecimentos

A Deus pelo dom da vida eterna, saúde concedida, disposição, força e fé necessários para concluir mais uma jornada na vida.

Aos meus queridos pais Angelo e Loide Aliano, pelo apoio, educação, carinho, estímulo à vida acadêmica, pois sem eles não seria possível iniciar tampouco concluir este trabalho.

À minha querida orientadora, a profª. Helenice, pela paciência em sua orientação, competência, incentivo, valiosos conselhos, sugestões e pela confiança dispensada, que foram fundamentais para a elaboração deste trabalho.

Especialmente agradeço à querida profª. Margarida Pato por ter me aceitado com aluno estagiário no Instituto Superior de Economia e Gestão (ISEG), pela orientação nos dois meses que estagiei em Portugal, e que seus conselhos, sugestões e dicas foram cruciais para o aperfeiçoamento deste trabalho. Agradeço também ao ISEG e à Universidade Técnica de Lisboa (UTL) por terem me amparado neste período.

Aos professores e funcionários do depto. de Bioestatística. Em destaque ao profº. Paulo Manceira: pelas brincadeiras, incalculáveis ajudas, dicas e opiniões recebidas deste memorável professor.

Sou grato também ao profº. Arenales pelas dicas e sugestões dispensadas neste trabalho.

Ao profº. Balbo da Unesp/Bauru pelo incentivo à Pós-graduação.

Agradecimento especial ao meu querido Irmão na fé, o ancião Sebastião Soares de Oliveira, de Bernardino de Campos, que desde criança me apoiou e deu conselhos práticos valiosíssimos em minha vida, por ter fornecido gentilmente os

dados pertinentes para as simulações realizadas nesta dissertação. Agradeço também pelas minhas irmãs na fé, à *Graciele* e à *Kellyane*: pelo imenso carinho, força e alegria que elas me proporcionaram.

À Reitoria de Unesp pelo auxílio financeiro dispensado para realização do meu estágio em Lisboa, Portugal.

À CAPES pelo apoio financeiro concedido.

À todas as pessoas que, direta ou indiretamente contribuíram para a realização deste trabalho.

"Não entesourais para vós tesouros na terra, onde a traça e a ferrugem consomem e onde os ladrões minam e roubam. Mas entesourai para vós tesouros nos céus, onde nem a traça nem a ferrugem consomem e onde os ladrões não minam nem roubam. Porque onde estiver o teu tesouro, ali estará também o teu coração."

Mateus 6:19-21

Sumário

Lista de Figuras

Página

Lista de Tabelas

Página

METAHEURÍSTICAS EM UM PROBLEMA DE ROTAÇÃO DE CULTURAS

Autor: ANGELO ALIANO FILHO
Orientadora: Profª Drª HELENICE DE OLIVEIRA FLORENTINO SILVA
Co-orientadora: Profª Drª MARGARIDA VAZ PATO

RESUMO

Um dos focos centrais na produção vegetal, discutidos ultimamente, é a utilização de medidas que visam um planejamento sustentável e ecológico, tendo em vista a degradação ambiental ocorrida nos últimos anos.

Por este motivo, a Rotação de Culturas tem ganhado destaque na literatura, pois é um meio de produção cujos princípios práticos viabilizam uma agricultura ecológica e produtiva. Esta prática, uma vez bem conduzida pelos agricultores rurais, traz inúmeros benefícios, visto que o controle de pragas, patógenos e de plantas daninhas é realizado biologicamente, diminuindo a ação de pesticidas prejudiciais ao meio ambiente e medidas de recuperação do solo, tornando-o sempre fértil.

Nesta dissertação, é apresentado um modelo de otimização 0-1 para o problema de Rotação de Culturas, cujo objetivo foi encontrar uma programação de

plantio de hortaliças que maximize a lucro da produção, levando-se em consideração restrições de época de semeadura para cada cultura considerada, o não cultivo de plantas de mesma família em lotes vizinhos, proibição de plantio consecutivo de plantas de mesma família botânica em um mesmo lote, a necessidade de adubação verde, período de descanso do solo e de demanda. Nesta modelagem, foi considerada uma área de plantio genérica, cujos lotes são irregularmente distribuídos e de diferentes tamanhos.

Para resolução do problema, foram desenvolvidas e implementadas a seguintes metaheurísticas: (a) Algoritmo Genético, (b) *Simulated Annealing*, e as abordagens mistas (c) Algoritmo Genético com *Simulated Annealing* e (d) Algoritmo Genético com Busca Local (Memético). Para avaliar os comportamentos computacionais das heurísticas, considerou-se instâncias de diferentes formas com variações nas geometrias e área de plantio. Adicionalmente, uma aplicação destes métodos para um conjunto de dados reais, também foi realizada.

Os resultados computacionais ilustraram que os algoritmos, especialmente as versões híbridas, encontraram soluções factíveis num reduzido tempo computacional, mostrando serem excelentes ferramentas para as tomadas de decisões nesta área.

Palavras-chave: Otimização, Metaheurísticas, Rotação de Culturas.

METAHEURISTICS IN A CROP ROTATION PROBLEM

Author: ANGELO ALIANO FILHO
Adviser: Prof$^{\underline{a}}$ Dr$^{\underline{a}}$ HELENICE DE OLIVEIRA FLORENTINO SILVA
Co-adviser: Prof$^{\underline{a}}$ Dr$^{\underline{a}}$ MARGARIDA VAZ PATO

SUMMARY

The environmental degradation that has occurred throughout the world claims for sustainable and ecological plant production. In this context, agricultural planning based on crop rotation has been addressed in many studies. Once appropriately applied by farmers, this practice brings many benefits. In fact, it enables biological control of pests, pathogens and weeds, thus reducing the action of pesticides which are harmful to the environment. Planting according to crop rotation also restores the soil, making it always fertile.

This thesis presents a binary linear optimization model to the problem of crop rotation aiming to find a planting schedule for vegetables that maximizes the profits of production. The problem constraints include a specific period for planting each crop, a predefined demand per crop, the need for green manure and rest period. Other restraints prevent planting of vegetables of the same family consecutively in

the same lot, as well as in neighboring lots. A general planting area with irregularly distributed and different sized lots is considered.

Four metaheuristics were specifically developed for the above crop rotation problem and the respective algorithms were implemented: (a) a Genetic Algorithm, (b) a Simulated Annealing, and two hybrid approaches - (c) a Genetic Algorithm with Simulated Annealing and (d) a Genetic Algorithm with Local Search, that is, a Memetic algorithm.

To evaluate the computing behavior of these algorithms, we considered a crop rotation instance from literature and also an instance built with real data from a Brazilian agricultural company.

The computational results showed that the algorithms, specially the hybrid versions determined feasible solutions in a reasonable computational time, thus showing to be powerful tools for decision making in this area.

Keywords: Otimization, Metaheuristics, Crop Rotation.

1 INTRODUÇÃO

O Brasil é um país de tradição agrícola e os mais diferentes cereais e hortaliças são cultivados em larga escala. As condições naturais como o clima, os solos férteis e o relevo propiciam o desenvolvimento deste amplo setor econômico.

Desde o Brasil colonial, a produção agrícola convencional se baseia principalmente na monocultura, motivada pelo baixo custo operacional e facilidade de implementação. Mas isto acarreta uma série de prejuízos, tais como o uso maciço de capitais e pesticidas tóxicos, que por sua vez, são altamente prejudiciais ao meio ambiente, facilitando a ação e prevalência de pragas e patógenos.

A monocultura, a longo prazo, também causa a exaustão do solo, implicando na utilização de doses cada vez mais altas de adubos químicos. Pontos negativos de ordem econômica também são originários desta política, como o empobrecimento de pequenos agricultores e a redução da produção de alguns produtos de origem agrícola (Altieri, 2002; Gliessman, 2000).

Para evitar estes problemas, é interessante que haja uma expansão da policultura, mas de uma forma sustentável, levando em consideração, ao mesmo tempo, a preservação ambiental das áreas de plantio, o controle biológico de pragas (e, como consequência, o uso menos intensivo dos agrotóxicos) e os ganhos econômicos.

É neste sentido que o tema Rotação de Culturas tem sido objeto de estudo das áreas agronômicas e ecológicas. A ideia básica consiste em construir um calendário de plantio de várias culturas cultivadas em uma mesma região particionada em lotes, alternando, anualmente, as diferentes famílias botânicas de vegetais. As espécies escolhidas devem ter propósitos comerciais e de recuperação do solo (Santos et al., 2011).

As vantagens da Rotação de Culturas são inúmeras. Arf et al. (1999); Altieri (2002); Gliessman (2000) descrevem que, além de proporcionar a produção diversificada de alimentos e outros produtos agrícolas, se adotada e conduzida de modo adequado e por um período suficientemente longo, essa prática melhora as características físicas, químicas e biológicas do solo; auxilia no controle de doenças, pragas e plantas daninhas; repõe matéria orgânica no solo protegendo-o da ação dos agentes climáticos e da erosão.

Em uma Rotação de Culturas alguns princípios de cultivo, sugeridos por Arf et al. (1999); Altieri (2002); Gliessman (2000), devem ser seguidos, entre eles:

1. separar cultivos de mesma família botânica;
2. não permitir cultivo consecutivo de plantas de mesma família botânica em uma mesma área;
3. plantio de adubação verde;
4. introdução do pousio (descanso do solo);
5. respeitar os ciclos e períodos de plantio de cada cultura.

A estratégia (1) tem a finalidade de dificultar a ação das pragas. Geralmente, as culturas pertencentes à mesma família são suscetíveis aos mesmos agentes patológicos. Assim, em um modelo sustentável de produção, a diversificação de culturas no tempo e no espaço, baseando-se na combinação de culturas de diferentes famílias botânicas fará com que a produção sofra menos ataque das pestes e uma maior estabilidade frente às pressões ambientais (Altieri, 2002).

A restrição agronômica (2) visa proteger e conseguir um melhor aproveitamento dos recursos disponibilizados pelo solo. Observa-se que o plantio sequenciado e imediato de uma mesma cultura em um mesmo espaço geográfico acarreta o seu empobrecimento, esgotando rapidamente seus recursos minerais, mesmo utilizando altas doses de adubos inorgânicos. A ideia básica consiste em alternar

diferentes espécies botância na mesma área, afim de aproveitar de forma ótima os recursos naturais da terra, mantendo-a rica em microorganismos, sais minerais e húmus, reduzindo as taxas de produtos inorgânicos e tóxicos utilizadas.

O uso da adubação verde (3), ou das leguminosas, têm inúmeros benefícios. Entre as suas principais utilidades, destaca-se a introdução de nitrogêneo na terra naturalmente pelos nódulos que estas plantas têm em suas raizes. As leguminosas, uma vez cultivadas, deixam enorme quantidade de matéria orgânica no solo. Esta cobertura vegetal deixa-o fértil, o mantém protegido da intensa luz solar, conserva a umidade, controle da variação térmica e evita o processo erosivo. No pousio ou descanso do solo (4), a vegetação espontânea é deixada crescer livremente por um período definido, para a recuperação da estrutura e fauna do solo.

Como a maioria das plantas possui um desempenho sazonal, tendo o ciclo produtivo e evolutivo dependente das condições climáticas, como luz solar, umidade do ar, temperatura e precipitação, o modelo de Rotação de Culturas programa um calendário de cultivo visando respeitar a época de semeadura de cada cultura (5).

Trabalhos como o de Santos et al. (2011) e Lemalade et al. (2011), descrevem a dificuldade dos problemas de Rotação de Culturas, tendo em vista a natureza das variáveis envolvida e o elevado número de restrições a ser respeitado.

Assim, tendo as restrições de plantio (1)-(5) a serem atendidas, este trabalho apresenta um modelo matemático 0-1 para uma programação de plantio baseado-se no trabalho de Santos et al. (2011). Afim de aproximar o modelo à realidade, inseriu-se restrições de demanda no modelo apresentado.

Este trabalho visa apresentar uma modelagem matemática de otimização 0-1 para o problema de rotação de culturas e propor métodos heurísticos para a sua resolução. Para isto, são apresentados (a) Algoritmo Genético, (b) *Simulated Annealing*, as abordagens mistas (c) Algoritmo Genético com *Simulated Annealing* e (d) Algoritmo Genético com Busca Local (Memético). Uma comparação entre estes foi realizada, com a finalidade de verificar qual melhor se adptou neste modelo, de

forma a obter uma programação de plantio sustentável e auxiliar os produtores rurais nas tomadas de decisões.

2 REVISÃO DE LITERATURA

2.1 Otimização Combinatória

Informalmente, problemas de Otimização Combinatória são problemas para os quais o espaço de soluções possíveis (viáveis, candidatas ou factíveis) é finito e discreto. A área da Pesquisa Operacional (PO) de estudar e como buscar a solução ótima para tais problemas denomina-se *Otimização Combinatória* (Reeves & Beasley, 1993).

Três exemplos de problemas combinatoriais clássicos, citados por Reeves & Beasley (1993) e Lawler et al. (1985), são apresentados a seguir, evidenciando como modelam inúmeras situações práticas:

Problema da mochila 0-1:*um conjunto de n ítens é avaliavel para ser acondicionado dentro de uma mochila com capacidade de C unidades. O ítem i tem valor* v_i *e usa* c_i *da capacidade. Determinar o subconjunto I dos ítens que deveriam ser acondicionados para*

Maximizar

$$\sum_{i \in I} v_i$$

Sujeito a

$$\sum_{i \in I} c_i \leq C.$$

Aqui, a solução é representada pelo subconjunto $I \subseteq \{1, ..., n\}$.

Problema de cobertura de conjuntos: *uma família de m subconjuntos coletivamente contém n ítens tais que o subconjunto S_i contém $n_i \leq n$ itens. Seleciona-se $k \leq n$ subconjuntos $S_{i_1}, .., S_{i_k}$ tais que $\left|\bigcup_{j=1}^{k} S_{i_j}\right| = n$. Deve-se minimizar*

$$\sum_{j=1}^{k} c_{i_j}$$

em que c_i é o custo do subconjunto S_i. Neste caso, a família de soluções é representada pela família de subconjuntos $\{S_{i_1}, .., S_{i_k}\}$.

Problema de roteamento de veículos: *um depósito tem m veículos avaliáveis para fazer entregas para n consumidores. A capacidade do veículo k é C_k unidades, enquanto o consumidor i requer c_i unidades. A distância entre os consumidosres i e j é $d_{i,j}$. Nenhum veículo pode viajar mais que D unidades. Alocar os consumidores para os veículos e achar a ordem em que cada veículo visita seus consumidores assim, objetiva-se*

Minimizar

$$\sum_{k=1}^{m} \sum_{i=0}^{n_k} d_{\pi_{i,k},\pi_{i+1,k}}$$

Sujeito a

$$\sum_{i=1}^{n_k} c_{\pi_{i,k}} \leq C_k; \qquad k = 1..m$$

$$\sum_{i=0}^{n_k} d_{\pi_{i,k},\pi_{i+1,k}} \leq D; \quad k = 1..m$$

$$\sum_{k=1}^{m} n_k = n.$$

Aqui, o veículo k visita n_k consumidores, é a solução e representada pela permutação $\{\pi_{1,1}..\pi_{n_1,1}..\pi_{1,m}..\pi_{n_m,m}\}$ dos números $\{1..n\}$, que é particionado pelos números n_k. Isto também é entendido que o depósito é representado pelos "consumidores" $\pi_{0,k}$ e $\pi_{n_k+1,k}$ para cada k.

Nestes exemplos, pode-se fazer uma relação com a PL na busca por uma estratégia mais fácil de solucioná-los introduzindo variáveis decisórias 0-1, repre-

sentando a tomada ou não de uma decisão. Se no problema da mochila for utilizado

$$x_i = \begin{cases} 1 & \text{se o ítem } i \text{ é acondicionado} \\ 0 & \text{caso contrário} \end{cases}$$

então o problema seria transformado em um modelo de PLI a seguir

$$\begin{aligned} &\text{Maximizar} && \sum_{i=1}^{n} x_i v_i \\ &\text{Sujeito a} && \sum_{i=1}^{n} x_i c_i \leq C. \\ & && x_i \in \{0,1\},\ i = 1..n \end{aligned}$$

Diante desta modelagem, se há n ítens à disposição, então existem 2^n possibilidades de alocá-los na mochila. Para um n pequeno o problema é fácil de ser solucionado. Um método mais intuitivo, a "força bruta" seria testar as 2^n possibilidades e escolher a melhor. Mas para um n grande isto poderia ser incalculável. Considerando-se um computador que faz 10^8 cálculos por segundo e $n = 50$, por "força bruta", este problema seria resolvido em 130 dias!

Segundo Reeves & Beasley (1993), esta estratégia gera uma formulação simples do problema, entretanto precisa de uma certa engenhosidade do formulador. Além disto, esta codificação em problemas de elevadas dimensões, pode levar à um número muito elevado de variáveis e restrições, dificultando e fazendo os algoritmos computacionais terem um lento desempenho. Somado-se a isto, um problema com variáveis inteiras é muito mais complicado de ser solucionado do que um contínuo. Esta é a razão de muitos pesquisadores se preocuparem com problemas desta natureza aliado ao fato de serem muito comuns na prática.

O método enumerativo *Branch-and-Bound* foi apresentado por Land & Doig (1960) e é a técnica exata mais difundida para resolver problemas inteiros. Entretanto, a principal desvantagem técnica consiste no esforço computacional exi-

gido. O tempo de computação necessário para a determinação de uma solução pode atingir níveis impraticáveis, tornando-os muitas vezes inviáveis em abordagens reais.

A idéia central deste algoritmo é enumerar todos os pontos do espaço de definição do problema considerado, dado que este é limitado, descartando infactibilidades e pontos não promissores para a otimalidade e procurando a solução factível que dê o melhor valor para a função objetivo. Para mais detalhes deste método cita-se Nemhauser & Wolsey (1999).

Em Otimização Combinatória, um conceito muito utilizado pelos métodos de resolução não exatos é o de *vizinhança*, afim de escapar das armadilhas dos ótimos locais.

De acordo com Reeves & Beasley (1993), a vizinhança de uma solução $\mathbf{x}$, denotada por V($\mathbf{x}$,σ) é um conjunto de soluções que pode ser pesquisado, "movendo-se" de $\mathbf{x}$ por uma simples operação σ. Tal operação, por exemplo, pode remover ou adicionar um objeto da solução em curso. A troca de dois ou mais objetos da solução é um outro exemplo de uma vizinhança. Os mesmos autores ressaltam que, para cada problema, uma estrutura de vizinhança (na verdade a operação σ) deve ser cuidadosamente elaborada para garantir uma boa eficácia destas metodologias.

2.2 Técnicas heurísticas de otimização

Uma noção de complexidade computacional de algoritmos é apresentada a seguir.

Para medir o custo de execução de um algoritmo é comum definir uma função de custo ou função de complexidade f. Desta forma, $f(n)$ é a medida do tempo necessário para executar um algoritmo para um problema de tamanho n.

Um algoritmo para um problema de dimensão n tem complexidade $O(f(n))$, se existe uma constante positiva K tal que o tempo de execução desse algoritmo é majorado por $K(f(n))$, para todas as instâncias do referido problema com dimensão n. Há duas classificações para os algoritmos, a saber:

- *polinomial*: quando f é um polinômio ou inferior, por exemplo, a $O(n^2)$ ou $O(n \ln n)$;

- *exponencial*: quando f é superior a um polinômio ou, por exemplo, da ordem $O(n^{\ln n})$ ou $O(2^n)$.

Um problema pertence à classe P quando existe um algoritmo polinomial que o resolve, determinando a solução ótima. Por outro lado, denomina-se NP-difícil se ainda não foi encontrado um algoritmo polinomial para o solucionar. Algoritmos desta ordem de complexidade não são úteis sob o ponto de vista prático, pois o tempo computacional consumido aumenta muito rapidamente com a instância do problema. O problema de programação linear, por exemplo, pertence à classe P, enquanto o problema da mochila e do caixeiro viajante são classificados como NP-difíceis[1].

Freitas (2009) afirma que a grande maioria dos problemas de natureza combinatórial são NP-difíceis. Portanto, a resolução por métodos exatos é uma tarefa impraticável. Para esta classe de problemas, a solução somente pode ser encontrada após um considerável esforço computacional.

No entanto, é possível dar uma certa inteligência a estes métodos em problemas combinatórios, utilizando o *Branch-and-Bound* reduzindo o número de soluções a analisar no espaço de soluções; entretanto, dada a natureza complexa, pode ser que, no pior caso, todas as soluções tenham que ser analisadas. Este fato impede o uso exclusivo destes métodos, dito exatos, dado o tempo proibitivo de se encontrar a solução ótima. Isto faz com que a aplicação desta e outras metodologias determinísticas seja restrita (Freitas, 2009; Neto & Becceneri, 2009).

Em muitas situações práticas, em geral, é suficiente encontrar uma "boa" solução para um problema ao invés da melhor. Por este motivo, muitos pesquisadores têm concentrado esforços em métodos de otimização não determinísticos, denominado *heurísticas*, para solucionar problemas deste nível de complexidade. Seu

[1] Ver uma apresentação rigorosa destes conceitos ver em (Lawler et al., 1985)

desenvolvimento está intimamente ligado ao desenvolvimento da computação, pois foi somente a partir da década de 1980 que estas metodologias sugiram fortemente na literatura como uma alternativa viável na resolução de problemas de Otimização Combinatória (Duarte & Coelho, 2002; Buzzo & Moccellin, 2000).

A palavra heurística vem do grego *heuristike*, que significa descobrir, encontrar. Este termo remonta ao episódio de Arquimedes, vislumbrado quando descobriu como calcular o volume de um objeto irregular submegindo-o na água. Com esta decoberta, este matemático grego ficou tão feliz que saiu pelas ruas gritando: *eureka! eureka!* (descobri! descobri!).

Segundo Reeves & Beasley (1993) uma técnica heurística de otimização é aquela que é inspirada em processos intuitivos que procuram boas (isto é, quase-ótimas) soluções à um custo computacional aceitável, sem estar comprometida em garantir a otimalidade, ou mesmo em muitos casos, o quão perto da ótima está a solução encontrada.

Como retrata Neto & Becceneri (2009) e Freitas (2009), o desafio consiste em produzir, num tempo reduzido, soluções tão próximas quanto possível da solução ótima. Muitos esforços têm sido feitos nesta direção e heurísticas eficientes foram desenvolvidas para diversos problemas. Uma característica desta metodologia é ser muito específica para um problema particular: uma mesma heurística pode ser eficiente para um determinado problema e ineficiente para outro.

Bernardino (2008) descreve que as heurísticas são algoritmos baseados em duas etapas distintas: construção e refinamento (algoritmos de melhoria) agrupando-as como:

- **construção**: uma ou mais soluções são construídas, elemento a elemento, seguindo algum critério de otimização. O algoritmo é executado até que se tenha uma solução viável para o problema;

- **busca em vizinhança (refinamento)**: parte-se de uma solução inicial e movimenta-se para uma solução vizinha se a mesma for promissora. Este pro-

cesso é realizado até que um critério seja satisfeito;

- **híbridas**: combinações de duas ou mais heurísticas para favorecer a busca. A seção 2.5 retrata especificamente os algoritmos híbridos.

Uma heurística construtiva objetiva construir uma solução, elemento por elemento. A forma de escolha de cada elemento a ser inserido a cada passo varia de acordo com a função de avaliação adotada, a qual, por sua vez, depende do problema abordado. Nas heurísticas clássicas, os elementos candidatos são geralmente ordenados segundo uma função denominada gulosa, a qual estima o benefício da inserção de cada elemento, e somente o mais promissor é inserido a cada passo.

A Figura (1) a seguir ilustra um pseudocódigo de uma heurística que constrói uma solução s, utilizando uma função gulosa $g(.)$. A variável t_{melhor} indica o melhor elemento da lista de candidatos C com o valor mais favorável da função de avaliação g.

Algoritmo 1 *ConstruçãoGulosa(g(.),s)*

1 $s \leftarrow \varnothing$;
2 Inicialize o conjunto C dos elementos candidatos;
3 enquanto $(C \neq \varnothing)$ faça
4 $\quad g(t_{melhor}) = melhor\{g(t); t \in C\}$;
5 $\quad s \leftarrow s \cup \{t_{melhor}\}$;
6 Atualize a lista C dos elementos candidatos;
7 fim-enquanto
8 Retorne s;
9 **fim** *ConstruçãoGulosa*

Figura 1 - Pseudocódigo de construção gulosa (Freitas, 2009)

Existem outras formas de se construirem soluções para problemas de otimização. Em muitos casos, afim de obter um ganho no tempo computacional, opta-se em fazer uma construção aleatória, isto é, a escolha de cada elemento s é

feita de maneira arbitrária. Entretanto, há uma desvantagem com esta abordagem, pois podem fornecer soluções finais de baixa qualidade, requerendo um esforço computacional muito alto na fase de refinamento (Neto & Becceneri, 2009). A Figura (2) mostra um pseudocódigo de uma heurística construtiva aleatória.

Algoritmo 2 *ConstruçãoAleatória(g(.),s)*

```
1 s ← ∅;
2 Inicialize o conjunto C dos elementos candidatos;
3 enquanto (C ≠ ∅) faça
4    Escolha aleatoriamente t_escolhido ∈ C;
5    s ← s ∪ {t_escolhido};
6 Atualize a lista C dos elementos candidatos;
7 fim-enquanto
8 Retorne s;
9 fim ConstruçãoAleatória
```

Figura 2 - Pseudocódigo de construção aleatória (Freitas, 2009)

A fase de refinamento consiste em encontrar, a cada iteração, uma solução que melhore a solução atual, através de uma busca realizada em sua vizinhança. Essas heurísticas partem de uma solução inicial qualquer, que pode ser construída por um método guloso ou aleatório, tentando melhorar essa solução através de operações de troca, remoção ou inserção, até que não seja mais possível melhorar a solução ou algum outro critério de parada seja satisfeito. O método de Busca Local (BL) é o mais clássico e será discutido com mais detalhes na seção 2.4.

2.2.1 Metaheurísticas

Embora dispensem um conhecimento aprofundado do problema, as heurísticas podem ter seu desempenho melhorado ao incorporar algumas características do problema na sua implementação (Freitas, 2009; Bernardino, 2008).

Isto sugere o termo *metaheurística*, cujo nome vem de heurística combi-

nado com *meta*, que do grego, exprime a ideia de nível superior, maior generalidade. Assim, a metaheurística reúne conceitos das áreas de Otimização e Inteligência Artificial, viabilizando a construção das chamadas melhores estratégias ou dos métodos "inteligentemente flexíveis". Esse termo pode ser considerado como associado a um conhecimento circunstancial, não verificável matematicamente (Blum & Roli, 2003).

Neto & Becceneri (2009) afirmam que metaheurísticas são mecanismos de alto nível para explorar espaços de busca, cada uma usando um determinado tipo de estratégia, com o aspecto comum de sempre escapar dos ótimos locais.

Esses métodos, teoricamente ainda foram pouco explorados pela literatura, e possuem como característica básica estruturas com uma menor rigidez que as encontradas nos métodos clássicos de otimização sem, contudo, emergir em uma flexibilidade caótica. A desafio é evitar a convergência prematura para um ótimo local, mas também não impedir que o algoritmo divirja, fazendo uma busca aleatória (Neto & Becceneri, 2009; Coelho, 2003). Dois conceitos devem ser balanceados na busca:

- **intensificação**: tentar melhorar a solução corrente realizando pequenas mudanças que conduzam a novas soluções próximas à solução corrente em uma região;
- **diversificação**: encontrar novas regiões do espaço de busca que ainda não tenham sido investigadas.

Segundo Blum & Roli (2003) as heurísticas e metaheurísticas para serem desejáveis devem conter algumas propriedades tais como:

- **simples**: deve ser baseada em um princípio claro e aplicável em geral;
- **coerente**: traduzir a ideia da metaheurística nos algoritmos;
- **eficiente**: deve fornecer a solução em um tempo computacional razoável;
- **efetiva**: encontrar as soluções ótimas para a maioria dos problemas, para os quais se conhece a solução;

- **robusta**: deve ser consistente, em uma ampla variedade de problemas testes;
- **amigável**: fácil de entender e implementar, usando a menor quantidade de parâmetros possíveis;
- **inovação**: permitir o uso em novos tipos de aplicação.

Em geral, a maioria das metaheurísticas cumprem com somente algumas das propriedades acima descritas. Por outro lado, não existe uma metaheurística que possa ser considerada melhor com relação a outra, já que, em geral, dependem da natureza dos problemas e dos objetivos buscados.

Segundo ainda estes autores, uma característica que pode ser usada para a classificação de uma metaheurística é o número de soluções usadas ao mesmo tempo (população ou única solução). Existem algoritmos que trabalham com uma única solução, ou algoritmos não-populacionais, sendo métodos que descrevem uma trajetória no espaço de busca, durante sua execução. Assim, a exploração do espaço de soluções é feita por meio de movimentos, os quais são aplicados a cada passo sobre a solução corrente, gerando outra solução promissora em sua vizinhança.

Já as metaheurísticas baseadas em população, executam um processo de busca que descrevem a evolução de um conjunto de soluções no espaço de busca. Os mantendo um conjunto de boas soluções e combiná-las de forma a tentar produzir soluções ainda melhores.

Dentre as várias metaheurística algumas delas têm analogia com a natureza como os Algoritmos Genéticos (AG), Algoritmos Meméticos (AM), Colônia de Formigas (CF), *Simulated Annealing* (SA), Busca Tabu (BT), Grasp, etc. As três primeiras utilizam a natureza como fonte de inspiração e são populacionais. Em particular, os AGs e AMs inspiram-se no processo de evolução natural das espécies enquanto o SA metaforiza um processo físico, mais precisamente com a termodinâmica. A BT faz uso de uma memória flexível para efetuar uma busca mais eficaz.

As seções que seguem descrevem, de maneira mais detalhada, o AG, SA, as abordagens híbridas Algoritmo Genético com *Simulated Annealing* (AG+SA)

e AM, metaheurísticas estas que foram utilizadas neste trabalho.

2.3 O Algoritmo Genético

Os métodos de otimização não determinísticos baseados nos princípios da evolução biológica natural das espécies viventes têm recebido crescente interesse nas últimas décadas, devido a sua versatilidade e adaptabilidade em problemas de natureza complexa e combinatorial (Chipperfield & Fleming, 1996).

Dentre estes métodos, o principal é Algoritmo Genético (AG). A ideia do AG pode ser ententida como uma "técnica de exploração pseudo-aleatória do espaço de pesquisa, mas de maneira inteligente".

O nome *Algoritmo Genético* é originário da analogia entre a representação de uma complexa estrutura por meio de um vetor de componentes, constituindo um cromossomo artificial. No processo de seleção das espécies vivas, a descendência procura preservar e levar para as próximas gerações aquelas que têm características mais desejáveis - determinadas e levadas às gerações subsequentes pela combinação dos cromossomos pais (Reeves & Beasley, 1993).

O AG teve como fonte de inspiração o princípio Darwiniano da evolução das espécies e na genética, que apregoa a sobrevivência do mais apto. Foi declarado em 1859 pelo naturalista e fisiologista inglês Charles Darwin. De acordo com ele, "Quanto mais um indivíduo se adaptar no meio onde vive, maior será sua chance de sobreviver e gerar descendentes". O AG busca simular exatamente este processo seletivo que ocorre na natureza, como a reprodução, diversificação e sobrevivência do mais apto.

O AG utiliza um vocabulário emprestado da genética natural descrito a seguir (Goldberg, 1989):

- **cromossomo**: cadeia de caracteres representando informações relativas às variáveis do problema;
- **indivíduo**: solução reprentada por cromossomos;

- **gene**: unidade básica do cromossomo;
- **população**: conjunto de cromossomos ou indivíduos;
- **geração**: iteração em que o AG está sendo executado;
- **operações gênicas**: operações que o algoritmo realiza nos cromossomos;
- **aptidão ou *fitness***: infomação numérica de cada cromossomo na população.

Este método de otimização se desenvolveu, primeiramente, nos trabalhos de John Holland e seus associados, na Universidade de Michigan na década de 70. Em 1975, no primeiro tratamento sistemático, publicou o livro *"Adaptation in Natural and Artificial Systems"*. Após isto, seu aluno Goldberg, continuou trabalhando com estes métodos, publicando o livro "*Genetic Algorithms and Machine Learning*" inserindo fortemente esta área no campo acadêmico (Mitchell, 1997).

Coelho (2003), afirma que Holland percebeu que o "mecanismo biológico permitiu a adaptação do sistema natural biológico e que isto poderia ser expresso matematicamente e simulado computacionalmente". Desta abstração nasceu o AG.

Pacheco (1994), afirma que estes principios básicos de evolução são imitados na construção de algoritmos computacionais que buscam uma melhor solução, para um determinado problema matemático, através da evolução de soluções codificadas por estruturas artificiais.

Esta ideia é transportada para a matemática: para obter uma solução para um problema complexo, a grosso modo, combina-se as partes cromossômicas de soluções obtendo outras soluções (filhas). Embora esta metodologia seja não-exata, foi suficientemente persuasiva a Holland.

Segundo Coelho (2003), estas estruturas de codificação permitem contornar incovenientes de muitos problemas na prática, como a descontinuidade das funções envolvidas e a natureza dos mesmos. Desse modo, o AG não requer o conhecimento das características do problema de otimização e não depende de certas pro-

priedades da função objetivo e restrições, tais como convexidade e diferenciabilidade. A busca pela solução é apenas guiada por sucessivas avaliações dos cromossomos na função de aptidão.

Os problemas de natureza combinatorial são dos mais difíceis de serem resolvidos, e muitos autores têm dedicado esforços para solucioná-los. O AG, segundo Goldberg (1989), é uma técnica flexivel para esta classe de problemas, e fornece uma ampla gama de ferramentas para solucioná-los.

Especificamente, em muitas situações, os problemas são modelados por variáveis binárias, do tipo 0-1, o que não reduz o seu nível de dificuldade, mas aumenta. Goldberg (1989) sugere que há significativa vantagem de se utilizar o AG para este tipo de modelagem, onde os métodos clássicos de otimização, como por exemplo o *Branch-and-Bound*, têm pouca eficiência e demandam um tempo computacional inaceitável, dependendo da instância analisada.

Segundo este autor, de maneira simplista, os AGs são procedimentos de busca pseudo-probabilística projetados especialmente para trabalhar em grandes espaços de soluções. Estes métodos utilizam um conjunto de amostras, e a partir destas, constroem novas estruturas mais evoluídas e adaptadas, prevalecendo somente as mais fortes.

Uma importante característica desta abordagem consiste em mesclar entre determinística e probabilística a busca pela solução. A população inicial do método (ponto de partida) seja construtiva, ou seja aleatória, juntamente com as regras estocáticas dos operadores, fazem com que o algoritmo explore diferentes regiões do conjunto de busca. Se, por um lado, for apenas aleatória (caminho aleatório ou *random walk*), a busca é realizada em uma pequena parte do conjunto admissível, resultando em uma exploração pouco efetiva.

Por outro lado, uma busca puramente determinística leva a uma convergência prematura do método. Este tipo de busca apenas retêm e favorecem os melhores indivíduos na população, fazendo-a perder a diversidade, característica crucial no processo evolutivo. Sendo assim, o espaço de soluções é explorado de forma

parcial e facilmente há convergência em ótimos locais (também denominados picos), podendo ser soluções muito distantes da resposta ótima (Goldberg, 1989). Como enfatiza Michalewicz & Fogel (2000), "esta busca requer um equilíbrio entre dois objetivos aparentemente conflitantes: o aproveitamento das melhores soluções (determinística) e a exploração do espaço de busca (estocática)".

Sendo assim, o AG constitui uma método de busca de propósito geral que apresenta um balanço notável entre aproveitamento de melhores soluções encontradas e exploração do espaço de busca.

Este método é dividivo em várias etapas: primeiro impõe uma amostragem aleatória do espaço de busca, tentando simular a diversidade gênica. A etapa seguinte é controlada pela seleção, onde esta variedade passa a ser mais controlada pelo fatores externos ao meio. Por meio desta seleção, os cromossomos que possuem as características mais promissoras são conservados, e por meio da reprodução tais características são transmitidas para as próximas gerações. Assim sendo, os menos promissores tendem a se reproduzir menos e consequentemente, serem extintos.

Goldberg (1989) enumera algumas peculiaridades deste método e cita, por exemplo, a busca pela solução ótima com um conjunto de cromossomos, em contraste com as tradicionais metodologias, onde apenas uma solução é utilizada para realizá-la. O mesmo autor caracteriza um AG por:

- trabalha com múltiplas soluções concomitantemente;
- não utiliza informações matemáticas do problema, como derivadas, continuidade e convexidade;
- usa regras de transição probabilísticas e determinísticas;
- precisa apenas do valor da função objetivo nas soluções para cada indivíduo;
- apresenta simplicidade conceitual, baseado em processos intuitivos.

Entretanto, Freitas (2009) e Coelho (2003) ressaltam algumas questões que podem dificultar o uso deste algoritmo:

- a não garantia da otimalidade;
- a necessidade de um conhecimento muito específico do problema;
- dependendo do problema, elevado consumo de tempo computacional;
- em condições usuais, dificuldade de se obter o ótimo global;
- a fácil convergência prematura (ótimo local).

Goldberg (1989) divide o algoritmo, de forma geral, em inicialização, avaliação, seleção, cruzamento, mutação, atualização e finalização. Assim, inicia-se o processo de busca com uma população P de cromossomos, sendo possíveis respostas ao problema. Esta população é submetida ao processo de evolução, dividido nas etapas descritas a seguir:

- **avaliação**: considerando-se uma população com P cromossomos, cada um é avaliado pela função custo (objetivo) e são classificadas quanto à sua adaptabilidade ao meio. Em outras palavras, é estabelecida uma hierarquia dos cromossomos que melhor atendem ao problema a ser resolvido;
- **elitismo**: visa preservar o melhor indivíduo em cada geração;
- **seleção**: alguns indivíduos devem ser escolhidos para reproduzirem e deixarem descendentes;
- ***crossover***: cria novos indivíduos para a população através da recombinação de partes diferentes de cromossomos-pais pelo método anterior;
- **mutação**: o operador de mutação nos AGs é introduzido para prevenir a convergência prematura para ótimos locais através da amostragem de novos pontos no espaço de busca;
- **migração**: este operador nos AGs funciona para substituir um ou mais indivíduos na população por outros cromossomos, gerados aleatoriamente ou por uma heurística construtiva, afim de favorecer a diversidade cromossômica;

- **reavaliação**: os indivíduos criados na geração corrente pelos operadores de *crossover*, mutação e migração são inseridos na população e avaliados pela função objetivo do problema;

- **atualização**: separam-se os P melhores cromossomos e tem-se uma nova geração.

Sendo assim, o funcionamento básico de um AG (sem elitismo) se resume na inicialização e em sucessivas etapas de evolução, como descritas acima, repetindo-se até que um critério de parada seja satisfeito.

Basicamente, este método tem o funcionamento esquematizado como o fluxograma da Figura (3).

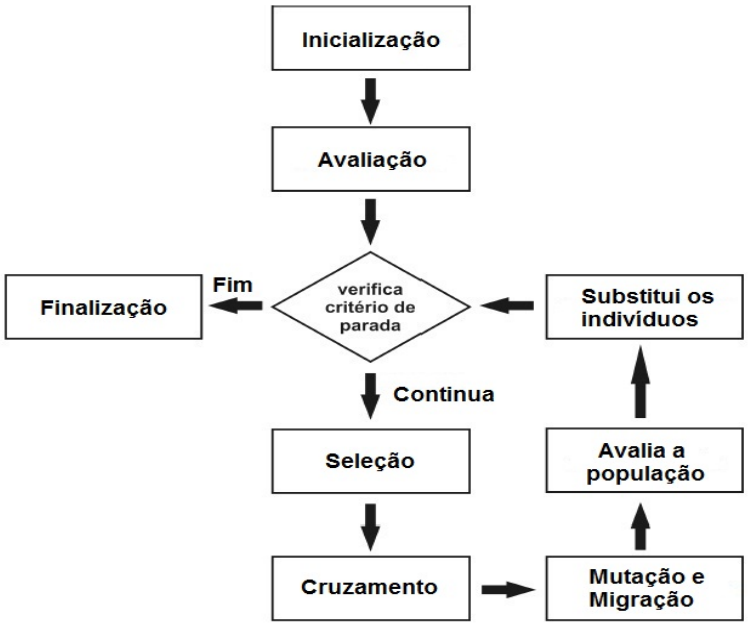

Figura 3 - Estrutura de um AG simples (Bernardino, 2008)

Um esquema simples para um AG genérico é escrito a seguir, com $Pop(t)$ indicando a população no tempo $t \geq 0$.

Nas seções seguintes, uma breve descrição de cada operador gênico é realizada.

Algoritmo 3 AG

```
1   t ← 0;
2   gere a população inicial Pop(t);
3   avalie Pop(t);
4   separe a elite;
5   enquanto (os critérios de parada não estiverem satisfeitos) faça
6           t ← t + 1;
7           gere Pop(t) a partir de Pop(t − 1);
8           avalie Pop(t);
9           defina a população sobrevivente;
10  fim-enquanto
11  x* ← {elite} {Solução do AG = x*};
12  fim AG
```

Figura 4 - Pseudocódigo do AG (Freitas, 2009)

2.3.1 Inicialização

Tipicamente, faz-se o uso de funções aleatórias para gerar uma população inicial. Este recurso é o mais simplista, do ponto de vista computacional, e visa garantir a *biodiversidade genética*, fator crucial para uma boa abrangência do espaço de soluções.

Entretanto, em alguns problemas, principalmente aqueles dotados de muitas restrições, a inicialização do método pode ser dado por uma *heurística construtiva*, visando uma melhora em sua performance. Isto faz com que o AG receba indivíduos com características desejadas, fazendo com que a busca seja mais incisiva em uma região do espaço de pesquisa onde se tenha mais certeza de infactibilidade. Goldberg (1989), enumera quatro métodos de inicialização:

- **inicialização randômica uniforme**: cada gene do cromossomo reberá, aleatoriamente, um dos possíveis valores do conjunto de alelos[2];

[2]Conjunto de valores que um gene pode assumir

- **inicialização randômica não uniforme**: determinados valores para alguns genes tendem a serem escolhidos com uma maior frequência;

- **inicialização randômica com "dope"**: quando se insere um ou mais indivíduos otimizados. Isso representa um sério risco ao método, pois pode ocasionar a geração do *super-indivíduo*, facilitando a convergência prematura do método;

- **inicialização parcialmente enumerativa**: insere-se na população alguns elementos, de forma a fazer o "processo evolutivo possuir todos os esquemas possíveis de uma determinada ordem".

O tamanho da população é um importante parâmetro a ser decidido e que afetam o desempenho global do método. Lacerda & Carvalho (1999) descrevem que uma população com poucos elementos estabiliza-se muito facilmente perdendo a diversidade necessária para convergir para o ótimo global. Por outro lado, populações com muitos cromossomos, o AG perderá sua eficiência computacional consideravelmente.

2.3.2 Avaliação

Nesta etapa do método, cada elemento da população é avaliado para que possa ser medido o seu grau de adaptabilidade no meio. A forma mais usual é avaliar cada cromossomo na função objetivo, denotada por $f_O(x)$. Em problemas de muitas restrições, funções baseadas em penalidades são mais comuns. Alguns exemplos de como a função objetivo pode ser construída são exibidas a seguir.

Por exemplo, para o Problema do Caixeiro Viajante tem-se

$$f_O(x) = \frac{1}{\sum_{i=1}^{l_G-1} \text{distância}(g_i, g_{i+1})}$$

em que g_i representa o *i-ésimo* gene do genótipo G e l_G o seu comprimento.

No caso de um problema de escalonamento, cujo objetivo principal é obter soluções sem violações, uma alternativa para medir o grau de adaptablidade de um indivíduo seria adotar

$$f_O(x) = \frac{1}{\sum_i penalidade_{ij}}$$

em que o índice i indica a penalização *i-ésimo* indivíduo e j a penalização por ele infringida.

Goldberg (1989) salienta o extremo cuidado que se deve ter ao escolher $f_O(x)$, pois uma vez que esta é massivamente testada, sendo a única guia determinística para o mesmo, se for complexa o custo computacional pode ser prejudicado.

2.3.3 Elitismo

Este operador tem uma enorme importância no AG, tendo em vista que impede o método em perder as melhores soluções até então encontradas, melhorando consideravelmente sua eficácia. Assim, o elitismo clássico guarda o melhor cromossomo em uma variável, impedindo-a de participar dos operadores seleção, *crossover*, mutação e migração, inserindo-a ao final de cada geração. Se uma outra solução, mais apta que esta, for obtida nas próximas gerações, então esta variável vai sendo atualizada. Deste modo, garante-se o não decrescimento do algoritmo com o passar das gerações, como pode ser visto na Figura (5). Foi proposto por DeJong (1975), um dos pioneiros trabalhos sobre AGs.

2.3.4 Seleção

O operador de *seleção* emprega o princípio de sobrevivência dos mais aptos. Para metaforizar o que ocorre na natureza, dá-se a preferência aos indivíduos mais adaptados ao meio para realizar o *Crossover*, visando manter suas características para as próximas gerações. A única forma do método fazer esta identificação é pelo valor de sua aptidão (*fitness*). Um indivíduo é apto se ele tem uma

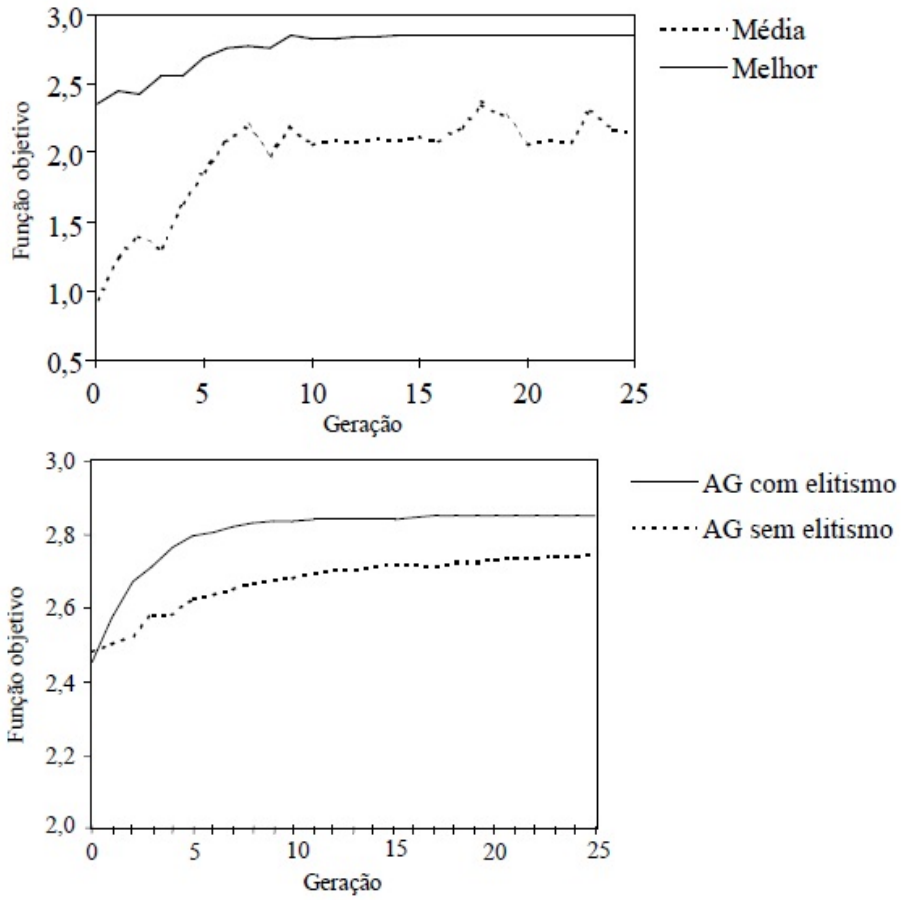

Figura 5 - Efeito de um AG com e sem o elitismo (Lacerda & Carvalho, 2000)

alta probabilidade de ser selecionado.

Este operador não cria novos cromossomos. Por um lado, não se pode priorizar apenas os melhores indivíduos para realizar a reprodução, pois isto poderia acarretar a convergência prematura do AG, pela escassez de diversidade genética. Por outro lado, não se pode simplesmente fazer uma escolha *ao acaso* das soluções que irão participar da próxima etapa e consequentemente levar suas informações para as gerações seguintes.

Deb & Goldberg (1991) incluem no operador *seleção* o conceito de *Pressão de Seletiva*. Tal conceito distingue um tipo de seleção que pioriza somente os melhores cromossomos, denominando-se *alta pressão*, e outra que busca uma maior

variabilidade entre os que vão ser escolhidos (*menor pressão*).

Estes autores descrevem que uma forte pressão ajuda o método a convergir rapidamente, perdendo o poder de busca, atingindo um ótimo local. Um operador com menos pressão ajuda que este processo não ocorra tão cedo, permitindo um tempo necessário a atingir pontos mais promissores. Contudo, uma pressão muito baixa pode dar ao AG uma busca completamente aleatória (*random walk*), dificultando a convergência e demandando um alto tempo de processamento.

Na literatura há vários métodos de seleção presentes, porém não existe um senso comum sobre qual metodologia é mais adequada para um problema em específico (Mitchell, 1997).

As noções de alguns operadores seletivos mais utilizados são descritas a seguir.

1. **Seleção por roleta viciada (*roulette wheel*)**: este mecanismo utiliza como critério seletivo uma probabilidade de seleção para cada indivíduo. Como já descrito anteriormente, cromossomos com alto grau de adaptabilidade possuem grandes chances de serem escolhidos, em contraste com os de baixa adaptabilidade. A probabilidade p_i do *i-ésimo* indivíduo ser selecionado é dada por:

$$p_i = \frac{f_O(i)}{\sum_{i=1}^{P} f_O(i)}$$

em que P é a cardinalidade da população.

Após calcular cada um destes valores, a probabilidade acumulada c_j para cada indivíduo é formada do seguinte modo

$$c_j = \sum_{k=1}^{j} p_k,\ j \in \{1..P\}.$$

O ato de girar a roleta para escolher uma solução, consiste em obter um número aleatório r uniformemente distribuído no intervalo [0,1]. Se $c_{j-1} < r < c_j$ indica que o elemento j foi selecionado. A sequência das probabilidades acumuladas

por ser vista como uma roleta: os cromossomos que têm uma aptidão têm uma maior parcela nesta e consequentemente, terão uma maior chance de serem escolhidos, em confronto com os de menor *fitness*.

Quando a pressão é baixa, segundo o *fitness* da população é igual para todos e o método da roleta, passa ser aleatório praticamente. Por outro lado, se uma classe de indivíduos tem um *fitness* muito acima da média pode ocorrer o inverso: a geração dos *super-indivíduos*, ocasionando a convergência prematura do AG, pela falta de diversidade.

Autores como Michalewicz & Fogel (2000) afirmam que no início das gerações, a pressão seletiva no AG deve ser baixa, favorecendo cromossomos com baixa aptidão a fazerem o *crossover*, afim de explorarem o espaço de soluções de forma global. Já nas gerações finais, a pressão deve ser maior, em razão dos indivíduos terem *fitness* muito próximo e consequentemente nortear a busca para um ótimo global. O método de seleção por roleta criam pressões seletivas de maneira inversa, prejudicando fortemente o desempenho do AG.

1.1. **Seleção por Nivelamento Linear**: Baker (1989) propõe uma alternativa para contornar este problema: recalcular a aptidão dos cromossomos, fazendo uma transformação linear do *fitness* e procedendo-se como anteriormente. Primeiro, classificam-se as soluções em ordem decrescente, com base na aptidão (se o problema for de minimização). Seja x_k uma solução e o índice k denotando a sua classificação na população ordenada. Então a função de avaliação é da forma:

$$f_O(x_k) = min + (Max - min)\frac{P-k}{P-1} \tag{1}$$

em que os parâmetros Max e min devem satisfazer $Max + min = 2$, sendo Max o parâmetro que mede a pressão da seleção, $1 \leq Max \leq 2$. A partir desta transformação linear, $min \leq f_O(x_k) \leq Max$, $\forall\ k \in \{1, P\}$. Para valores de Max próximos de 1, indicam menos pressão seletiva e mais aleatoridade nas

escolhas, ao passo que valores próximo de 2 indicam uma escolha priorizando os melhores cromossomos. De acordo com estes autores, nem uma nem outra situação deve ser fortemente utilizada. Para isto, recomendam que o parâmetro $1,2 \leq Max \leq 1,5$.

1.2. **Seleção por Nivelamento Exponencial**: Baker (1989) , alternativamente, ao invés de uma transformação linear como em (1), propôe um escalonamento exponencial. O cálculo para $f_O(x_k)$ é feito do seguinte modo:

$$f_O(x_k) = q(1-q)^k \tag{2}$$

em que k a classificação decrescente do indivíduo x_k na população pelo *fitness* e $q \in [0,1]$ um parâmetro de controle da transformação. Para valores de q próximos de 1, há mais diversificação na seleção (*menos pressão*) enquanto valores próximos de 0 há mais intensificação (*maior pressão*).

2. **Seleção elitista truncada**: Este método de seleção escolhe somente os melhores cromossomos da população (pelo valor da função objetivo), para realizarem o *crossover*, deterministicamente. Deb & Goldberg (1991) afirmam que este método faz o algoritmo convergir ou se estabilizar precocemente.

3. **Seleção por torneio (*tournament method*)**: Apresentada pela primeira vez por Deb & Goldberg (1991), este tipo de seleção é feita escolhendo-se arbitrariamente τ indivíduos da população; o melhor dentre estes é escolhido. Este procedimento é repetido até que se tenha a quantidade de indivíduos suficientes para realizar o *crossover*. O valor de τ é conhecido como o tamanho do torneio. Tipicamente, este valor é dois, representando um torneio binário. Também é utilizado para contornar os efeitos de uma possível seleção "quase-aleatória" que a roleta viciada pode ter.

Os operadores genéticos são os que geram novos indivíduos a partir da população inicial. A cada geração, novos cromossomos vão emergindo, fazendo a

população evoluir e se diversificar, abrindo caminho para encontrar e explorar pontos do conjunto solução. Dentre os principais operadores de recombinação, destacam-se o *crossover*, a *mutação* e a *migração*. Uma breve descrição destes é realizada a seguir.

2.3.5 *Crossover* (cruzamento)

Especificamente para este operador é necessário uma metodologia para parear os pais que originarão os seus descendentes. Geyer & Schultz (1997) citam algumas das principais formas de alocá-los, chamada de *mating*:

- **escolha aleatória**: os pares cromossômicos reprodutores são pareados arbitrariamente;
- ***line breeding***: um indivíduo de alta aptidão é colocado para cruzar com indivíduos de uma subpopulação;
- ***self-fertilization***: o indivíduo é cruzado consigo mesmo;
- ***positive assortive mating***: indivíduos de diferentes estruturas cromossômicas são combinados.

Feito o pareamento, aplica-se o *crossover* o qual consiste na troca (evento aleatório) de material genético entre cromossomos pais (Goldbarg & Luna, 2000). A idéia intuitiva por trás do operador é a mistura da informação entre os pais para formar uma outra diferente, porém com as informações oriundas dos progenitores. O resultado desta operação é um indivíduo que potencialmente combine as melhores características dos indivíduos usados como base.

O crossover é aplicado com uma dada probabilidade a cada grupo de pais, denominada *taxa de crossover* (c), geralmente entre 60 a 90% (Lacerda & Carvalho, 1999). Não ocorrendo este operador, os filhos serão idênticos aos pais, permitindo que algumas soluções sejam preservadas.

Diferentes tipos de *crossover* tem sido apresentados na literatura, dentre os quais destacam-se:

1. ***crossover* 1-ponto**: o modo mais comum presente nos algoritmos, por sua simplicidade. Para aplicá-lo, dois pais são pareados utilizando uma metodologia *mating* e dois filhos são obtidos. Cada um dos cromossomos pais tem sua cadeia de bits cortada em uma posição o aleatória, produzindo duas "cabeças" e duas "caudas". As caudas são trocadas, gerando dois filhos. Considere, na Figura (6), os pais P_1 e P_2 como sendo estruturas binárias, e os dois filhos F_1 e F_2 obtidos por um ponto de corte efetuado aleatoriamente.

   Michalewicz (1996) afirma que, dentre os operadores de cruzamento tradicionais este é o que normalmente apresenta o pior desempenho para problemas de natureza complexa, tendo em vista que a mistura do material genético não é tão efetiva.

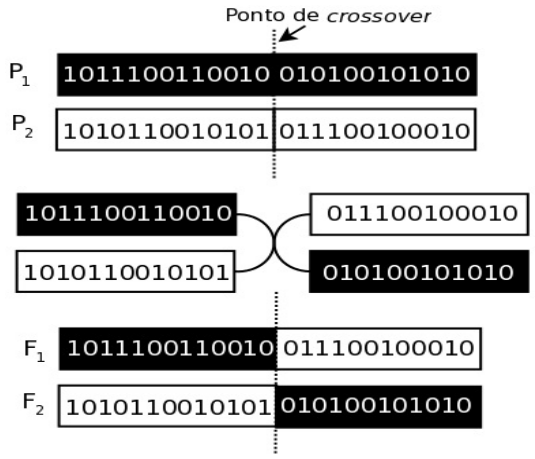

Figura 6 - *crossover* 1-ponto (Lacerda & Carvalho, 1999)

2. ***crossover* n-pontos**: esta abordagem é uma extensão do *crossover* 1-ponto (dois pais originam dois filhos), na qual são sorteados n pontos de cortes nos cromossomos progenitores e os materiais genéticos resultantes são trocados, obtendo-se os filhos. Na Figura (7) exemplifica-se o operador para $n = 2$.

3. ***crossover* uniforme**: para cada gene a ser preenchido nos cromossomos filhos, o operador de cruzamento uniforme sorteia de qual dos pais este deve ser

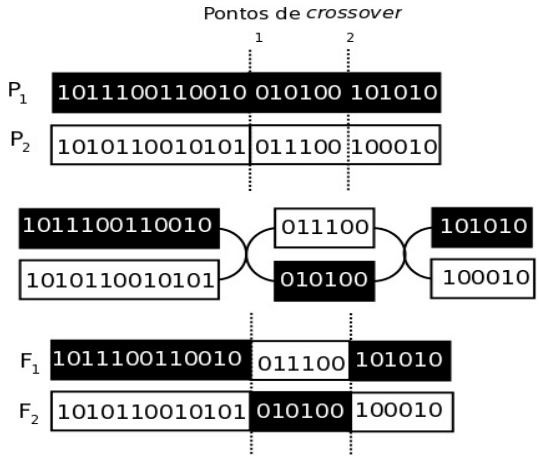

Figura 7 - *crossover* 2-pontos (Lacerda & Carvalho, 1999)

gerado. A máscara de cruzamento de tal operador é uma sequência qualquer de 0's e 1's gerada aleatoriamente. Assim, este procedimento troca genes, ao invés de segmentos de genes, como na Figura (8).

Figura 8 - *crossover* uniforme (Lacerda & Carvalho, 1999)

4. ***crossover* segmentado**: o cruzamento segmentado funciona de maneira semelhante ao de n-pontos, com a exceção de que sorteia o número de pontos de corte toda vez que é executado.

5. ***crossover* por combinação parcial**: sorteia dois pontos de corte no cromossomo e faz com que os filhos recebam, na íntegra os genes do pai ou da mãe (mãe para um e pai para outro) situados entre os pontos de corte. Após isto, preenche os genes restantes com os valores considerados mais adequados para cada filho.

De acordo com Mitchell (1997), decidir qual operador *crossover* utilizar depende da função *fitness*, codificação utilizada e outros detalhes adicionais do problema. Cada operador é particularmente eficiente para uma determinada classe de problemas e extremamente ineficiente para outras. Em geral, a decisão é tomada por experimentação.

2.3.6 Mutação

Após os cromossomos passarem pelo processo de reprodução, a população passará pelo operador mutação, modificando aleatoriamente um ou mais genes dos cromossomos.

Este operador é responsável pela exploração global do espaço de busca, introduzindo novo material genético em indivíduos já existentes. A importância deste operador é, uma vez bem escolhido seu modo de atuar, garantir diversas alternativas de exploração, mantendo assim um nível mínimo de abrangência na busca e impedir a convergência prematura do método, principalmente quando a população se estabiliza devido à baixa diversidade e à pressão seletiva (Goldberg, 1989).

Um cuidado adotado para a mutação é aplicá-la à uma baixa probabilidade, normalmente entre 1 a 5%, pois caso contrário pode estragar toda a boa informação genética da população até então construída, levando a perdas irreparáveis e uma exploração aleatória (*random walk*).

No entanto autores como Constantino et al. (2003) propõem o uso de uma taxa de mutação variável que aumenta de acordo com a convergência da população. Deste modo, com o passar das gerações, a ocorrência da mutação é maior.

Considerando codificação binária, o operador de mutação padrão simplesmente seleciona um cromossomo da população e alguns genes; se estes têm valor um, passará a ser zero após a aplicação e vice-e-versa (Goldberg, 1989), como na Figura (9) . Porém, a taxa de mutação, quantidade de genes e cromossomos modificados e a metodologia podem variar para cada problema.

Figura 9 - mutação padrão (Lacerda & Carvalho, 1999)

Mitchell (1997) segere algumas formas distintas de fazer uma mutação:

1. **mutação *flip***: o gene a ser mutado é substituído por outro diferente;
2. **mutação por troca (*swap mutation*)**: n pares de genes são sorteados e os valores são permutados entre si;
3. **mutação *creep***: um valor aleatório é somado ou subtraído do valor do gene.

2.3.7 Migração

Um dos grandes problemas do AG é a convergência prematura, onde os genes de alguns indivíduos relativamente bem adaptados, contudo não ótimos, podem rapidamente dominar a população causando a convergência à um ótimo local.

Afim de evitar que isto aconteça, alguns autores utilizam um outro operador para diversificar a busca pela solução, impedindo que método estacione em picos. Standerski (2003), por exemplo, utiliza em seu trabalho, uma migração de uma percentagem de indivíduos, e no lugar destes, substitui por novos elementos, retardando a convergência e fazendo com que diminua as chances de cair nestas armadilhas.

Outro mecanismo migratório, denominado *crowding* (exclusão), foi proposto por DeJong (1975) para manter diversidade na população. Neste esquema, a substituição de indivíduos na população é realizada com a intenção de substituir os cromossomos similares e os de menor aptidão.

2.3.8 Reavaliação e Atualização

Os indivíduos resultantes dos processos de cruzamento, mutação e migração serão inseridos na nova geração segundo o critério mais comum na literatura, empregado por Goldberg e Holland: conservar o tamanho da população, mantendo-a em um tamanho fixo. Neste ponto, todos os indivíduos serão novamente reavaliados e os P melhores cromossomos serão preservados e levados para a próxima geração, que por sua vez, será a população inicial para a próxima geração.

Esta metodologia de atualização é denominada por *steady-state* (estado fixo) e é a mais comum na prática. Um AG deste tipo mantém o tamanho populacional constante por todo o processo de evolução. A cada geração, são criados indivíduos que serão ou não inseridos na nova geração, dependendo exclusivamente do seu grau de aptidão medido por $f_O(x)$. Uma vez feito isto, os menos adaptados são "exterminados" de maneira a fazer com que esta retorne a seu tamanho original, eliminando sempre os menos aptos e sobrevivendo os mais evoluídos. É neste procedimento que se faz alusão ao modelo de "seleção do mais forte" de Charles Darwin (Mitchell, 1997).

2.3.9 Finalização e Critério de parada

A finalização não envolve o uso de nenhum operador genético: ela simplesmente é composta de um teste que dá fim ao processo de evolução caso o AG tenha chegado a algum ponto pré-estabelecido de parada. Os critérios de parada podem ser vários. O mais utilizado é o número de gerações previamente estabelecido. Porém há metodologias baseadas no próprio acompanhamento do processo evolutivo, isto é, enquanto não houver melhoria na média da população depois de um número pré definido de gerações, o processo evolutivo finaliza (Koza, 1992).

2.3.10 Parâmetros

Lacerda & Carvalho (1999) descrevem que em um AG, há vários tipos de parâmetros que controlam o processo evolutivo e que foram avaliados e discutidos

neste trabalho. Estes podem ser qualitativos ou quantitativos. A seguir, estes autores enumeram e descrevem a natureza de cada um:

- Quantitativos:
  - **tamanho populacional** P: é a cardinalidade do conjunto populacional;
  - **taxas de cruzamento** (c)**, mutação** (mt) **e migração** (mg): é a probabilidade destes operadores atuarem na população, respectivamente;
  - **itervalo do *ranking* linear** (Max) **e exponencial** (q): caso no emprego do método de seleção por escalonamento linear ou exponencial, estes parâmetros quantificam a pressão da seleção.
- Qualitativos:
  - **tipo de *matting***: escolhe-se a maneira de parear os cromossomos;
  - **tipo de cruzamento**: tipo de reprodução que será utilizado;
  - **tipo de seleção**: define-se o tipo de seleção a ser empregado;
  - **tipo de mutação e migração**: metodologia de efetuar a migração e a mutação.

Muitos outros parâmetros podem surgir para serem analisados e testados, como a quantidade de indivíduos que irão participar da mutação e migração, o número de genes transformados, entre outros, e como dito anteriormente, não há um metodologia específica para afiná-los e predizer qual a configuração melhor se reproduzirá. Goldberg (1989); Michalewicz & Fogel (2000) afirmam que a escolha adequada para estes parâmetros é por experimentação e depende de cada problema em específico, como seu nível de complexidade e dimensão.

2.4 A Busca Local

Um problema de Otimização Combinatória, em teoria, poderia ser solucionado por uma pesquisa exaustiva no conjunto S de soluções. Na prática, en-

tretanto, o espaço de soluções é extremamente grande, tornando este procedimento inviável. A otimização local supera este inconveniente pesquisando somente um pequeno subconjunto do espaço de soluções. Isto é feito definindo-se uma estrutura de vizinhança e uma forma de se "movimentar" dentro e fora deste conjunto. O método mais clássico reportado na literatura para este tipo de pesquisa é a Busca Local (BL).

Um pseudocódigo de uma BL é ilustrado na Figura (10), usando a seguinte terminologia: $Nmax$ é a quantidade máxima de iterações sem melhora, $V(s)$ a vizinhança da solução em s atual e f a função que se deseja minimizar.

Algoritmo 4 *BuscaLocal(f(.),V(.),Nmax,s)*

```
1   iter ← 0 {Contador de iterações sem melhora};
2   enquanto (iter < Nmax) faça
3         iter ← iter + 1;
4         selecione aleatoriamente v ∈ V(s);
5         se (f(v) < f(s)) então;
6            iter ← 0;
7            s ← v;
8         fim-se;
9    fim-enquanto;
10  retorne s;
11  fim BuscaLocal
```

Figura 10 - Pseudocódigo do algoritmo BL (Freitas, 2009; Dowsland, 1993)

Um subconjunto de soluções factíveis (vizinhos) é explorada repetidamente movendo da solução corrente s para um vizinho v promissor. Sempre o movimento, ou troca de soluções correntes, é em direção à uma melhora na função objetivo. O procedimento é interrompido após um número fixo de iterações sem melhora no valor da melhor solução obtida até então (Dowsland, 1993).

No entanto, esta estratégia frequentemente converge à um ótimo local podendo estar distante do ótimo global. Bona & Algeri (2001) definem um ótimo local e o principal problema dos métodos de decida, da seguinte maneira:

"O método de busca local é ineficiente quanto à armadilha do ótimo local, fazendo deste método uma heurística pobre para muitos problemas de otimização combinatorial. Uma propriedade desejável de qualquer algoritmo é a habilidade de encontrar uma boa solução, independente do ponto de partida. Um ótimo local se caracteriza quando o algoritmo atinge uma região correspondente ao fundo de um vale, em se tratando de um problema de minimização, que não contém a solução ótima e dele não consegue sair, uma vez que todas as soluções naquela vizinhança possuem valores maiores do que a solução corrente."

Uma estratégia para escapar da armadilha do ótimo local é executar diversas vezes o algoritmo com diferentes soluções iniciais, sendo adotado como solução ótima a melhor solução encontrada. Entretanto, esse procedimento conduz a um novo problema que é o de determinar quando parar o algoritmo, além de poder ser inviável em se tratando de grandes problemas (Araujo, 2001).

Outras metodologias para melhorar esta técnica são discutidas, entre elas a de aumentar a quantidade e a complexidade da vizinhança, porém nenhuma destas variantes têm demonstrado potencial de convergência satisfatório, em razão da extrema dependência deste método com a solução inicial empregada (Dowsland, 1993; Freitas, 2009; Saramago, 2003).

A Figura (11) ilustra o tipo de situação comum desta ingênua heurística de busca: convergência aos picos um s, s' ou s'' tendo em vista as soluções iniciais s_0, s_1 ou s_2, respectivamente.

2.5 O *Simulated Annealing*

O uso do *Simulated Annealing* (SA) como uma técnica de otimização discreta, iniciou-se na década de 80. As primeiras e remotas ideias deste método foram publicadas nos trabalhos de Metropolis et al. (1953), em um algoritmo para simular o processo de congelamento (recozimento) de um metal, denominado *anne-*

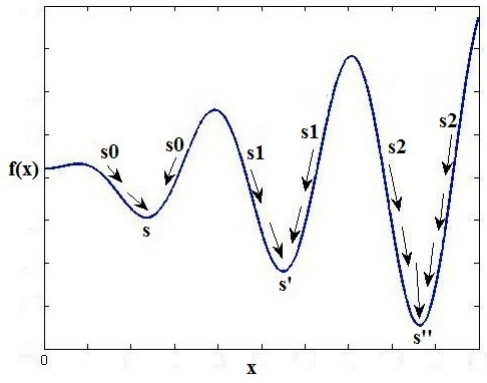

Figura 11 - Convergência para ótimos locais na BL

aling. Este pioneiro trabalho simulava a mudança do estado de energia do sistema quando sujeito ao processo de resfriamento, até convergir ao estado de congelamento (*frosen state*). Trinta anos mais tarde, Kirkpatrick et al. (1983) sugeriu que este tipo de simulação poderia ser usado para descobrir soluções factíveis em problemas de Otimização Combinatória, cujo objetivo era convergir para soluções ótimas (Dowsland, 1993; Saramago, 2003).

Assim, o SA nasceu na analogia entre o processo físico do resfriamento de um metal em estado de fusão e o problema de otimização. Inicialmente, para obter uma liga metálica, esta é fundida a elevadíssimas temperaturas e depois resfriada. Uma questão crucial para obter uma estrutura rígida é um recozimento cuidadoso, ou seja, numa redução bem lenta da temperatura desde o estado de fusão do material até o seu congelamento. Isto irá resultar em uma estrutura cristalina pura. Se não for feito assim os cristais resultantes terão muitos defeitos (amorfos) ou a substância pode se transformar em um vidro, que é uma estrutura apenas ótima localmente. Fundamentado nesta ideia de resfriamento até atingir um nível de equilíbrio térmico, simula-se a obtenção de uma solução para um problema de otimização (Neto & Becceneri, 2009).

De acorco com Kirkpatrick et al. (1983), o SA empresta um vocabulário da termodinâmica para a otimização e que é mostrado na Tabela (1) a seguir.

Tabela 1 - Vocabulário do SA

Termodinâmica	**Otimização Combinatória**
Átomo	Solução qualquer
Estado do sistema	Solução factível
Energia	Custo
Mudança de estado	Movimento
Temperatura	Parâmetro de controle
Congelamento	Solução heurística
Estado final	Parada

2.5.1 O Algoritmo SA

Considerando um problema de minimização, tendo um espaço de soluções S e o conjunto de vizinhança para cada solução corrente definido por V, cuja função objetivo é denotada por f, um SA genérico é escrito como segue.

Primeiramente, um elemento de s_0 de S é escolhido para ser a solução inicial. Esta passa a ser a solução corrente s na primeira iteração e a melhor resposta s^* até então encontrada.

O procedimento principal consiste em gerar o conjunto de vizinhaça $V(s)$ em torno da solução corrente, tomar aleatoriamente[3] um único elemento de V, digamos v, e calcular a variação de energia $\Delta = f(v) - f(s)$. O método aceita o movimento da solução corrente de s para seu vizinho v se $\Delta < 0$. Caso contrário, se $\Delta \geq 0$, a solução candidata vizinha poderá ser aceita, mesmo causando um acréscimo em f, mas com a probabilidade

$$P(\Delta, T) = e^{-\Delta/(k_B T)} \tag{3}$$

em que T é um parâmetro de controle do método, chamado de *temperatura* cuja

[3]todos os elementos $v \in V(s)$ têm a mesma probabilidade de serem acessados, independentemente da distância geométrica entre s e v

finalidade é de regular o aceite de soluções de pior custo e que vai sendo reduzido progressivamente. O outro valor, k_B, é denominado constante Boltzmann. Em muitos casos, este parâmetro passa a ser um simples fator de escala normalmente igualado à unidade ou, em outras situações, equilibrar as dimensões de Δ e T (Kirkpatrick et al., 1983).

Um pseudocódigo do SA genérico é apresentado na Figura (12).

Algoritmo 5 $SA(f(.), V(.), \alpha, SAmax, T_0, T_f, s)$

```
1   s* ← s {Melhor solução até então obtida};
2   iterT ← 0 {Contador de iterações numa isoterma T};
3   T ← T_0 {Temperatura corrente};
4   enquanto (T > T_f) faça;
5     enquanto (iterT < SAmax) faça;
6           iterT ← iterT + 1;
7           gere uma vizinhança em torno de V(s);
8           escolha aleatoriamente v ∈ V(s);
9           Δ = f(v) − f(s);
10           se (Δ < 0) então
11             s ← v;
12             se (f(v) < f(s*)) então s* ← v;
13           senão;
14             tome r ∈ [0, 1];
15             se r < e^{−Δ/T} então s ← v;
16           fim-se;
17    fim-enquanto;
18    T = α(T) {A temperatura é atualizada segundo uma regra α};
19    iterT ← 0;
20  fim-enquanto;
21  s ← s*;
22  retorne s;
fim SA
```

Figura 12 - Pseudocódigo do algoritmo SA (Kirkpatrick et al., 1983; Freitas, 2009)

O processo anterior, para um mesmo T, é repetido $SAmax$ vezes, isto é, em uma isoterma, a simulação deve ser executada num número tal de vezes que o

estado de equilíbrio seja atingido. Após isto, o valor de T é atualizado (diminuído) e o ciclo é novamente executado enquanto a temperatura corrente não atingir um valor próximo de zero, T_f, definido pelo usuário.

O algoritmo apresentado na Figura (12) é geral, mas modificações podem ser realizadas para resolver um dado problema. Uma modificação A primeira consiste em afinar os parâmetros de controle, como $T_0, T_f, \alpha(T)$ e $SAmax$. E outra envolve a escolha do espaço de soluções admissíveis, a forma da função de avaliação e principalmente, a topologia da estrutura de vizinhança a ser empregada. Estes dois tipos de decisões serão cruciais para um bom aproveitamento do método, devendo ser feitas com cuidado (Dowsland, 1993).

A solução inicial do SA pode ser feita de forma aleatória ou por construção, normalmente com base na experiência (Neto & Becceneri, 2009). O parâmetro T_0 deve ser um valor alto o suficiente para explorar diferentes regiões em S. Evidentemente, esse valor depende do tipo do problema e da instância analisada.

Na literatura, existem algumas propostas para o cálculo da temperatura inicial. Aarts & Korst (1989), por exemplo, propõem duas metodologias. Na primeira

$$T_0 = \ln(f(s_0)) \tag{4}$$

em que $f(s_0)$ é o valor da função objetivo da solução inicial.

Alternativamente

$$T_0 = \Delta E_{max} \tag{5}$$

em que ΔE_{max} é a diferença máxima de custo entre duas soluções vizinhas. No entanto esta estimativa pode ter um custo computacional alto.

O valor T_f (critério de parada) deve ser zero, em teoria, entretanto, na prática é suficiente chegar a uma temperatura muito próxima deste valor, desde que um elevado esforço computacional seja desprendido (Torreao, 2004). Segundo Dowsland (1993), o número de iterações em uma isoterma, $SAmax$, tem íntima

ligação com o tamanho da instância do problema analisado. Frequentemente este parâmetro é a cardinalidade de V, mas em alguns casos este valor pode ser outro.

Um procedimento eficiente e que pode economizar passos desnecessários consiste em menos iterações nas altas temperaturas e mais iterações nas baixas temparaturas, aumentando-se o valor de $SAmax$ gradativamente com a dimuição de T (Kirkpatrick et al., 1983; Dowsland, 1993; Neto & Becceneri, 2009).

Após o ciclo em uma mesma temperatura terminar, o valor T deve ser atualizado. Kirkpatrick et al. (1983) e Dowsland (1993) propõem um resfriamento geométrico, isto é

$$\alpha(T) = \beta \times T \tag{6}$$

em que β é denominada a taxa de resfriamento ou arrefecimento, $\beta \in [0, 1]$.

Outros autores como Torreão (1980) utilizam um resfriamento baseado na seguinte lei:

$$\alpha(T) = \frac{T}{1 + \gamma\sqrt{T}} \tag{7}$$

onde $0 < \gamma < 1$ é um parâmetro indicador da velocidade do resfriamento: valores próximos de 0 indicam um resfriamento muito lento.

Para obtenção de uma solução de boa qualidade e a não convergência para um ótimo local, é vital que a velocidade de resfriamento seja lenta. Em um resfriamento for geométrico, um elevado número de trabalhos demonstraram que $\beta \in [0,8; 0,99]$ são valores ideais.

Observa-se que a equação (3) depende da magnitude Δ e do estágio de resfriamento que o processo se encontra. Esta probabilidade aumenta à medida que a magnitude Δ diminui e também quando a temperatura é alta. Desse modo, no início, o SA tem uma grosseira estimativa do espaço S. Assim, o que difere o SA do método clássico de descida é exatamente esta tentativa probabilística de escapar dos picos, promovendo seguidos e cuidadosos passos de subida, explorando não só os "caminhos de descida" mas também regiões onde possa estar localizado o ótimo

global.

Por outro lado, $\lim_{T \to 0} P(\Delta, T) = 0$, isto é, o SA se comporta basicamente como um método de descida (apenas aceitando soluções promissoras) à medida que o processo se aproxima do congelamento, focalizando a área onde o mínimo global deve estar localizado.

A Figura (13) mostra a influência da variação da temperatura na função de probabilidade, considerando-se que a variação de energia Δ é a mesma durante toda a busca, fixada em uma unidade.

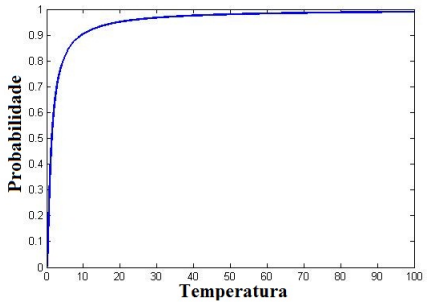

Figura 13 - Probabilidade de movimentos SA

O SA pode visto como uma variante da técnica heurística de pesquisa local, combinando as ideias deste procedimento com um cuidadoso passo de aceite de soluções não promissoras, difundido por Metropolis et al. (1953). Este aspecto peculiar do SA visa escapar dos ótimos locais e aumenta a possibilidade de se atingir um ótimo global, mesmo visitando soluções que pioram a função custo. Este cuidado é medido por uma probabilidade de aceitação destes movimentos (Dowsland, 1993; Neto & Becceneri, 2009; Saramago, 2003).

A Figura (14) ilustra geometricamente esta ideia e como pode ser benéfica, no sentido de uma otimização global. Inicialmente o método parte da solução s_0. Pelo método de descida simples, naturamente a solução de convergência seria s. Como o método aceita movimentos não promissores, mais intensamente no início da busca, foi possível obter a solução a, que a priori, teve um efeito negativo tendo em vista $f(s_0) < f(a)$. Ressalta-se, no entretanto, que o SA tem um

mecanismo de armazenar a melhor solução até então encontrada, evitando assim, a divergência.

Assim, por este movimento, foi possivel descobrir uma nova vizinhança onde foi possivel obter sucessivas melhoras na função objetivo, obtendo-se o ponto b resultando $f(b) < f(s)$. Novamente, o SA permite novas "subidas", às vezes por longas iterações, mas que permitirá obter novas regiões ainda mais promissores, como foi o caso de encontar a solução final d.

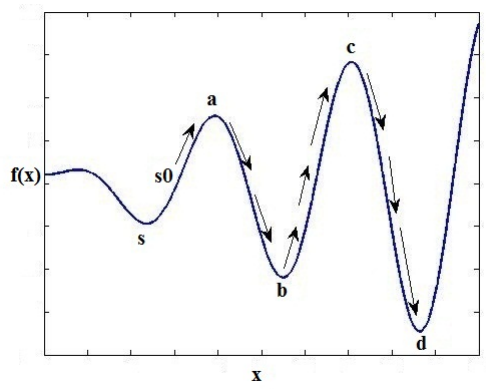

Figura 14 - Pesquisa de soluções no SA

Em relação à escolha dos parâmetros a serem utilizados, decidir quais as melhores configurações, estruturas de vizinhanças empregadas não é uma tarefa fácil e isto é conseguido depois de muitas experimentações (Kirkpatrick et al., 1983; Dowsland, 1993; Neto & Becceneri, 2009).

2.6 Algoritmos Híbridos

Há autores que defendem que alguns AGs tradicionais não são eficientes para solucionar certos problemas. Davis (1991) afirma que os AGs clássicos, embora robustos, geralmente não são os algoritmos de otimização mais eficientes em certos domínios.

Este autor ressalta que, apesar da eficácia das metaheurísticas em muitos problemas de complexidade elevada, estas podem não ter desempenho satisfatório

se usadas isoladamente ou não combinadas com outras técnicas. Isto se deve, principalmente, à dificuldade em afinar os parâmetros, sejam eles qualitativos ou quantitativos.

Dowsland (1993), de maneira similar, explica que a metaheurística SA mostrou ser uma útil técnica para uma variedade de problemas, em algumas instâncias sua performance pode ser melhorada consideravelmente combinando esta técnica com outros métodos heurísticos.

Os algoritmos híbridos consistem em combinar dois ou mais procedimentos e formar um novo método com o intuito de obter melhores resultados para os problemas propostos, diminuindo assim, a responsabilidade de uma afinação precisa dos parâmetros do "algoritmo puro" (Caseau & Laburthe, 1999). Não existem regras para estas combinações e segundo Coelho (2003), a dificuldade desta estratégia consiste em quando parar um algoritmo para começar o outro.

Embora exista uma vasta quantidade de combinações que podem ser realizadas, um elevado número de trabalhos como o de Buzzo & Moccellin (2000), Coelho (2003) e Caseau & Laburthe (1999) mostraram a eficácia das metaheurísticas hibridizadas. Este último mostrou uma eficiência especial quando se combinam metaheurísticas bio-inspiradas.

Yen (1998) em relação aos métodos híbridos, descreve que eles podem ser de quatro tipos:

- **híbridos *pipelining***: uma metaheurística é sequencialmente executada com outra técnica. Um Algoritmo Populacional Inspirado na Natureza (APIN) é processado primeiramente, e em seguida, uma busca local é realizada na população final obtida;

- **híbridos assíncronos**: esses métodos mantém uma população compartilhada em um método e outra técnica de busca. Ambos procedimentos que compõem o algoritmo híbrido de busca trabalham cooperativamente, modificando as soluções da população em comum;

- **híbridos hierárquicos**: um APIN e outro procedimento atuam em diferentes níveis do problema;

- **híbrido com uso de operadores adicionais**: podem ser construídos por qualquer procedimento de busca com um operador de movimento adicional que melhora os indivíduos pela sua aplicação. A população gerada pelo processo é inserida juntamente com o conjunto de filhos gerados pelos operadores de movimentos padrões do algoritmo.

A seguir são apresentados dois métodos híbridos que foram utilizados neste trabalho.

2.6.1 Híbrido: Algoritmo Genético & *Simulated Annealing* (AG+SA)

Segundo Goldberg (1989), um caminho eficiente para melhorar consideravelmente o desempenho de um AG é incorporá-lo com uma pesquisa com vizinhança local ou variante, citando por exemplo o SA. A vantagem desta metodologia híbrida está na melhoria da velocidade de convergência pela avaliação da função objetivo. Deste modo, o valor final obtido pelo método de busca local será mais preciso que o obtido pelo AG "puro".

De forma mais precisa, um AG é executado e a solução final deste servirá como a solução inicial para o SA. Isto permite ao segundo método iniciar sua busca em uma região mais específica, onde provavelmente possa estar localizado o ótimo global. Um considerável ganho computacional, com esta metodologia pode ser obtido, pois segundo Dowsland (1993), a temperatura inicial do SA não precisa ser relativamente alta quando a solução inicial for de boa qualidade[4].

Desta forma, o AG contribui em diminuir o processamento do SA, este é beneficiado por iniciar a sua busca com uma solução mais promissora, comparada com processos aleatórios.

[4]este termo, dependendo das circunstâncias do problema, pode se referir à uma solução factível ou com poucas infactibilidades.

Apresenta-se na Figura (15) o algoritmo AG+SA.

Algoritmo 6 $AG + SA(f(.), V(.), \alpha, SAmax, T_0, T_f)$

```
1   t ← 0;
2   gere a população inicial Pop(t);
3   avalie Pop(t);
4   separe a elite;
5   enquanto (os critérios de parada não estiverem satisfeitos) faça
6         t ← t + 1;
7         gere Pop(t) a partir de Pop(t − 1);
8         avalie Pop(t);
9         defina a população sobrevivente;
10  fim-enquanto
11  ag ← {elite} {ag é a solução do AG};
12   s ← ag {solução corrente do SA};
13   s* ← ag {melhor solução do SA} ;
14   iterT ← 0 {contador de iterações numa isoterma T};
15   T ← T_0 {temperatura corrente};
16   enquanto (T > T_f) faça:
17      enquanto (iterT < SAmax) faça:
18            iterT ← iterT + 1;
19            gere uma vizinhança em torno de V(s);
20            escolha aleatoriamente v ∈ V(s);
21            Δ = f(v) − f(s);
22            Se (Δ < 0) então
23              s ← v;
24              Se (f(v) < f(s*)) então s* ← v;
25            senão
26              tome r ∈ [0, 1];
27              Se r < e^{−Δ/T} então s ← v;
28            fim-se;
29      fim-enquanto;
30      T = α(T) {a temperatura é atualizada segundo uma regra α};
31      iterT ← 0;
32   fim-enquanto;
33   s ← s*;
34   retorne s;
fim AG+SA
```

Figura 15 - Pseudocódigo do algoritmo AG+SA (Kirkpatrick et al., 1983; Freitas, 2009)

2.6.2 Híbrido: Algoritmo Genético & Busca Local - Memético

Coelho (2003) observa que os AGs convencionais geralmente são eficientes para uma busca global, mas são relativamente lentos em sintonia local, principalmente em problemas de elevadas dimensões.

Assim, para obter benefícios com abordagens híbridas, em termos computacionais e em relação a qualidade da solução encontrada, utiliza-se um algoritmo de Busca Local após o término do AG. Esta metodologia, que sucede o AG, comumente é um método de Descida Randômica, descrito em detalhes na seção 2.4, cujo objetivo é explorar a vizinhança da solução obtida e caminhar em busca do ótimo global.

Desta forma, nas aborgagens mistas, o objetivo é de combinar as técnicas visando obter um novo método de otimização mais eficiente do que qualquer um dos componentes tomados isoladamente (Buzzo & Moccellin, 2000).

Um pseudocódigo deste método é dado na Figura (16).

Algoritmo 7 *Memetico(f(.),V(.),Nmax)*

```
1   t ← 0;
2   gere a população inicial Pop(t);
3   avalie Pop(t);
4   separe a elite;
5   enquanto (os critérios de parada não estiverem satisfeitos) faça
6         t ← t + 1;
7         gere Pop(t) a partir de Pop(t − 1);
8         avalie Pop(t);
9         defina a população sobrevivente;
10  fim-enquanto
11  ag ← {elite} {ag é a solução do AG};
12   s ← ag {Solução corrente do BL};
13   iter ← 0 {Contador de iterações sem melhora};
14   enquanto (iter < Nmax) faça
15         iter ← iter + 1;
16         selecione aleatoriamente v ∈ V(s);
17         se (f(v) < f(s)) então;
18            iter ← 0;
19            s ← v;
20          fim-se;
21    fim-enquanto;
22  retorne s;
23 fim Memetico
```

Figura 16 - Pseudocódigo do algoritmo Memético (Kirkpatrick et al., 1983; Freitas, 2009)

2.7 Rotação de Culturas

A experiência em muitos trabalhos práticos, como em Aita (1994), Schick (2000) e Muzilli (1983), mostrou que a repetição, ano após ano, de uma mesma cultura na mesma área, contribui para a diminuição da biodiversidade e, consequentemente, estimula os desequilíbrios físicos, químicos e biológicos do solo. Um planejamento cultural estratégico na propriedade proporciona o equilíbrio necessário

quanto aos aspectos ecológicos e econômicos dos sistemas de produção. Dentre estas estratégias, pontua-se a rotação de culturas, foco deste trabalho.

Entende-se como rotação de culturas a alternância regular e ordenada no cultivo de diferentes espécies vegetais em sequência temporal numa determinada área, utilizando plantas de cobertura que servem para formação da palhada na superfície do solo, culminando na redução de gastos com fertilizantes nitrogenados e herbicidas (Duarte & Coelho, 2002).

Segundo Altieri (2002), este modo de produção vegetal é considerado benéfico sob vários aspectos, dentre eles:

- aumenta a produção das plantas;
- melhora a fertilidade do solo;
- atua nas propriedades físicas do solo, controlando a erosão e protegendo contra a ação dos agentes climáticos;
- reduz a presença de nematóides, insetos, ácaros, ervas adventícias, vermes e fitotoxinas e plantas daninhas.

Este mesmo autor salienta que, embora muitas rotações sejam aceitáveis, seqüências bem-sucedidas devem obedecer aos seguintes princípios:

1. separação de culturas com suscetibilidade semelhante às mesmas pragas e doenças;
2. fertilização equilibrada, utilizando seqüências distintas de cultivos em períodos consecutivos;
3. inclusão de um período de cultivo de leguminosas, no mínimo;
4. uso de práticas que aumentam a matéria orgânica do solo;
5. respeito às épocas de semeadura de cada plantação.

O não cultivo adjacente de plantas de mesma família, como citado no ítem (1), tem razões ecológicas. Produtores rurais estão acostumados com o plantio adjacente de plantas de mesma natureza em regiões homogêneas, facilitando a ação de pragas e patógenos. Uma área com diferentes plantações alocadas em áreas vizinhas, ajuda na quebra do ciclo reprodutivo de muitas pragas, dificultando a sua ação.

O solo possui um fundamental papel na produtividade de uma plantação. Uma estragégia não pensada por muitos agricultores é o uso maciço de adubos químicos e em quantidades cada vez mais crecentes. Esta estratégia é quando pelo plantio é feito por longos períodos e com plantas de mesma família na mesma área; esta prática, uma vez realizada, promove a exaustão do solo, tendo em vista que plantas de mesma família esgotam ou utilizam os mesmos recursos minerais da terra. O princípio abordado em (2) visa alterar esta situação, alternando plantas que se nutrem de diferentes recursos, fazendo com que o solo se recupere para a cultura seguinte.

As leguminosas (plantas da família *leguminosae*), são fixadoras de nitrogênio com sistema radicular profundo ou abundante e também são chamadas por muitos autores de adubação verde (3). Estas são capazes de aproveitar os fertilizantes residuais das culturas comerciais. O manejo da matéria orgânica mediante rotação de culturas, adubação verde e consorciação de culturas pode proporcionar melhor aproveitamento de adubos químicos e possibilitar redução nos custos com adubação nitrogenada mineral, uma vez que propicia aumento da atividade biológica do solo (Altieri, 2002; Arf et al., 1999; Gliessman, 2000).

Há muitos benefícios com a utilização da adubação verde. Segundo Myiazaka & Camargo (1984), as plantas leguminosas, deixam no solo, quando decompostas, cerca de duas vezes mais nitrogênio em comparação com a família das gramíneas. Adicionalmente, em condições favoráveis, estas apresentam nas raízes, nódulos onde se encontram bactérias que vivem associadas com essas plantas. As bactérias vivem à custa das leguminosas, mas, ao mesmo tempo, promovem a fixação do nitrogênio do ar, enriquecendo a terra com esse elemento, suprindo a parte ne-

cessária para sua própria sobrevivência.

Além disto, suas raízes alcançam altas profundidades; este sistema radicular permite extrair elementos menos solúveis e mobiliza nutrientes das camadas do solo mais profundas, aproventando-os eficientemente para a nutrição da planta.

Arf et al. (1999) constatatou que a manutenção da matéria orgânica do solo é muito importante para o controle de nematóides e patógenos. Verificou também que todos os adubos verdes são altamente eficientes na redução da população ativa de nematóides fitoparasitas e saprófagos; assim, a adubação verde e um dos métodos mais baratos de controle de nematóides e pragas. Estes motivos justificam o papel vital da adubação verde em uma programação de platio, tando do ponto de vista ecológico e econômico.

Altieri (2002); Arf et al. (1999); Gliessman (2000) recomendam também, em conjunto com a adubação verde, que haja na área rotacionada um período de descanso da terra, chamado de *pousio* (4). Este, por sua vez, tem a finalidade de deixar a vegetação espontânea crescer por um tempo definido, contribuindo para o controle biológico de nematóides e para o solo recuperar a sua estrutura química, física e biológica. Após o crescimento desta vegetação, a mesma é dessecada; a cobertura vegetal seca formada tem muitas utilidades, entre elas:

- evita a erosão do solo, distribuindo e tornando mais lento o movimento da água da superfície, reduzindo o escoamento e mantendo-o no lugar;
- melhora a fertilidade do solo acrescentando-lhe material orgânico durante a decomposição, e tornam mais disponíveis seus nutrientes através da fixação do nitrogênio;
- controla a poeira, mantendo o solo preso aos sistemas radiculares;
- ajuda no controle de insetos nocivos, acolhendo predadores benéficos de insetos e parasitas;
- modifica o microclima e a temperatura, reduzindo o reflexo da luz e do calor e

aumentando a umidade do solo no verão;

- minimizam a competição entre a lavoura principal e as plantas daninhas;
- reduz a temperatura do solo.

Altieri (2002) defendo a importância da cobertura vegetal no solo, afirma "*A flora do cultivo de cobertura funciona como um 'conversor ecológico' importante, que ativa e influencia processos e componentes chave do agroecossistema, como o complexo de fauna benéfica, a biologia do solo e o ciclo do nitrogênio.*"

Finalmente, o ítem (5) anterior aborda a época de plantio de cada plantação, tendo em vista que o fator clima, como temperatura, precipitação, incidência de luz solar têm um papel vital na colheita. Mesmo em regiões tropicais nas quais é possível cultivar o ano inteiro (desde que haja irrigação disponível), existem culturas cujo desempenho é extremamente sazonal, com grandes restrições de época de cultivo.

A seção seguinte apresentará um modelo matemático que visará elaborar um calendário de plantio que maximize o lucro e atender todas as restrições ecológicas anteriormente enunciadas.

2.7.1 Modelagem Matemática para um Problema de Rotação de Culturas com restrições de adjacências

Define-se uma *Programação de Rotação de Culturas* de tamanho T como a elaboração de um calendário de plantio, na unidade temporal adotada (semana, mês, bimestre), decidindo-se quais culturas deverão ser plantadas, em que período da rotação e os lotes onde deverão ser cultivadas.

Como a àrea de cultivo considerada numa rotação é dividida em vários lotes, pode-se associar um grafo planar conexo $G(V, B)$, em que V é conjunto dos lotes e B é o conjunto das arestas tal que $(u, v) \in B$, se, e somente se, os lotes u e v são adjacentes. Considera-se que dois lotes são adjacentes quando eles compartilham um número não discreto de pontos em sua fronteira.

O modelo matemático apresentado neste trabalho para o Problema de Rotação de Culturas com restrições de adjacências (PRC-A), baseou-se no artigo de Santos et al. (2011), cujo o objetivo foi maximizar o tempo ocupação do terreno cultivado, enquanto neste trabalho, o objetivo foi maximizar a lucratividade da produção. Adicionalmente, inseriu-se uma restrição de demanda, que do ponto de vista prático, é relevante (mais detalhes na próxima seção). As restrições ecológicas e técnicas seguidas naquele e neste trabalho para o horizonte de planejamento foram as mesmas, a seguir:

a) *Época de semeadura* - necessita-se respeitar rigorosamente a época de plantio e o ciclo de vida de cada cultura.

b) *Áreas vizinhas* - plantas da mesma família botânica não podem ser plantadas em lotes adjacentes no mesmo período.

c) *Plantio consecutivo* - uma mesma família botânica de plantas não pode ser plantada em um mesmo lote consecutivamente.

d) *Adubação verde* - todos os lotes devem contemplar, no mínimo, uma cultura leguminosa que faz adubação verde e se sujeita às condições a), b) e c).

e) *Pousio* - em cada lote, é necessário programar um período de descanso do solo.

O PRC-A neste trabalho visa determinar uma programação de rotação de culturas, de mesma duração para cada um dos lotes, objetivando-se maximizar o lucro total da plantação realizada. O horizonte de planejamento foi dividido em M períodos de mesma duração, com um conjunto de N culturas pertencentes à N_f famílias de plantas botânicas e a área de plantio dividida L lotes. Os outros parâmetros considerados são:

- C - conjunto de culturas para fins comerciais;
- A - conjunto de culturas para adubação verde (leguminosas);

- F_p - conjunto de plantas da família $p,\ p = 1..N_f$;
- t_i - ciclo de vida da *i-ésima* plantação, incluindo tempo de preparação da terra e colheita;
- l_i - lucratividade (R\$. ha^{-1}) da cultura i nos t_i períodos;
- $I_i = [C_i, T_i]$ - intervalo de semeadura da *i-ésima* plantação, em que C_i é o período mais cedo e T_i o período mais tarde;
- S_k - o conjunto dos lotes vizinhos do lote k;
- $area_k$ - área do lote k em hectares (ha);
- T - duração da programação, igual para todos os lotes.

O PRC-A foi modelado como uma otimização linear binária descrito a seguir. Por convenção e simplicidade, a cultura indicada por $n = N + 1$ será o pousio. A variável decisória será codificada do seguinte modo:

$$x_{ijk} = \begin{cases} 1 & \text{se a cultura } i \text{ tiver seu plantio iniciado no período } j \text{ no lote } k; \\ 0 & \text{caso contrário} \end{cases}$$

$$\text{Maximize } z = \sum_{i \in C} \sum_{j \in I_i} \sum_{k=1}^{L} area_k \, l_i \, x_{ijk} \tag{8}$$

Sujeito a

$$\sum_{i \in F_p} \sum_{r=0}^{t_i - 1} \sum_{v \in S_k} x_{i(j-r)v} \leq L \left(1 - \sum_{i \in F_p} x_{ijk} \right), p = 1..N_f, j = 1..M, k = 1..L \tag{9}$$

$$\sum_{i \in F_p} \sum_{r=0}^{t_i} x_{i(j-r)k} \leq 1, p = 1..N_f, j = 1..M, k = 1..L \tag{10}$$

$$\sum_{i=1}^{N+1} \sum_{r=0}^{t_i - 1} x_{i(j-r)k} \leq 1,\ j = 1..M,\ k = 1..L \tag{11}$$

$$\sum_{i \in A} \sum_{j=1}^{M} x_{ijk} \geq 1,\ k = 1..L \tag{12}$$

$$\sum_{j=1}^{M} x_{njk} \geq 1,\ k = 1..L \tag{13}$$

$$x_{ijk} \in \{0,1\},\ i = 1..N+1,\ j \in I_i,\ k = 1..L. \tag{14}$$

Observação: se $j - r < 0$, substitui-se $j - r$ por $j - r + M$.

A função objetivo (8) visa maximizar a lucratividade da rotação realizada nos L lotes, nos M períodos. Entram para o cômputo da lucratividade somente as culturas que pertencem ao conjunto C comerciais, isto é, aquelas onde $l_i \neq 0$. As plantas com $l_i = 0$, são de utilidade ecológica, inclusive o pousio.

As restrições (9) impedem que plantas da mesma família botânica sejam plantadas em lotes adjacentes nos $t_i - 1$ períodos anteriores ao período corrente. As restrições (10) evitam que uma planta de mesma família botânica seja plantada em períodos consecutivos no mesmo lote.

As restrições indicadas em (11) impedem que duas plantas ocupem o mesmo lote no mesmo intervalo de tempo. Em outras palavras, se uma cultura i for plantada no período j, no lote k, é necessário que passem t_i períodos até que uma nova plantação ocupe o mesmo espaço.

Finalmente, (12) e (13) garantem que em cada lote, haja pelo menos uma adubação verde e um pousio, respectivamente. Para o pousio, não se aplicam as restrições de vizinhança e plantio consecutivo.

Para exemplificar o modelo apresentado, considera-se uma área de plantio como na Figura (17-a), ilustrando-se uma área de plantio dividida em cinco lotes não paralelos e Figura (17-b) seu grafo associado.

Suponha que existam as culturas $i = \{1..10\}$ sendo $F_1 = \{1,2,3,4\}$, $F_2 = \{5,6,7,8\}$ e $F_3 = \{9,10\}$ os conjuntos de plantas de mesma família botânica. Considera-se todos os períodos de plantio iguais a 2, ou seja, $t_i = 2, i \in \{1..10\}$. Pela Figura, sabe-se que $S_1 = \{2,3,4,5\}, S_2 = S_4 = \{1,3,5\}$ e $S_3 = S_5 = \{1,2,4\}$ o conjunto dos vizinhos dos lotes 1-5, respectivamente.

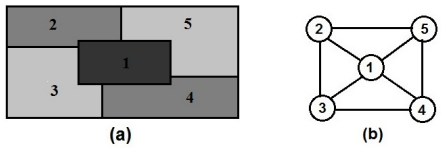

Figura 17 - Uma particular área de cinco lotes Santos et al. (2011)

Admita que uma cultura qualquer da família F_1, digamos $i = 1$, seja plantada no lote $k = 1$ no período $j = 12$ então $x_{1,12,1} = 1$. Deve-se obedecer às seguintes restrições:

- **plantio não sobreposto** - uma vez que a terra está ocupada no período 12 com a cultura 1, nenhuma outra cultura pode ocupá-lo nos períodos 11 e 12. Assim:

$$\sum_{i=1}^{10}\sum_{r=0}^{1} x_{i,(12-r),1} =$$
$$= \underbrace{(\overbrace{x_{1,12,1}}^{1} + x_{2,12,1} + x_{3,12,1} + x_{4,12,1} + x_{5,12,1} + x_{6,12,1} + x_{7,12,1} + x_{8,12,1} + x_{9,12,1} + x_{10,12,1})}_{\text{período 12}} +$$
$$+ \underbrace{(x_{1,11,1} + x_{2,11,1} + x_{3,11,1} + x_{4,11,1} + x_{5,11,1} + x_{6,11,1} + x_{7,11,1} + x_{8,11,1} + x_{9,11,1} + x_{10,11,1}+)}_{\text{período 11}} \leq 1$$

e generalizando esta ideia para todas as culturas, lotes e períodos, tem-se a restrição (11)

$$\sum_{i=1}^{N+1}\sum_{r=0}^{t_i-1} x_{i(j-r)k} \leq 1,\ j = 1..M,\ k = 1..L.$$

- **plantio consecutivo e simultâneo** - pela restrição de plantio consecutivo, as plantas $\in F_1$ não podem ser plantadas nos períodos 12, 11 e 10 no lote 1 (com exceção a que fora plantada). Então

$$\sum_{i\in F_1}\sum_{r=0}^{2} x_{i,(12-r),1} =$$
$$= \underbrace{(x_{1,10,1} + x_{2,10,1} + x_{3,10,1} + x_{4,10,1})}_{\in F_1 \text{ e período 10}} + \underbrace{(x_{1,11,1} + x_{2,11,1} + x_{3,11,1} + x_{4,11,1})}_{\in F_1 \text{ e período 11}} +$$
$$+ \underbrace{(\overbrace{x_{1,12,1}}^{1} + x_{2,12,1} + x_{3,12,1} + x_{4,12,1})}_{\in F_1 \text{ e } j=12} \leq 1$$

isto é generalizando para culturas pertencentes à mesma família botânica, em todos lotes e períodos, obtendo-se a restrição (10)

$$\sum_{i\in F_p}\sum_{r=0}^{t_i} x_{i(j-r)k} \leq 1,\ k = 1..L, j = 1..M, p = 1..N_f$$

são as restrições de plantio consecutivo.

- **vizinhança** - uma vez que o lote 1 está sendo ocupado com uma cultura pertencente à F_1 no período 12, para garantir a factibilidade, é necessário impor a condição do não plantio das culturas 1, 2, 3 e 4 no conjunto S_1 no período 11 e 12. Em outras palavras, $x_{i,j,v} = 0, i = 1, 2, 3, 4, j = 11, 12$ e $v = 2, 3, 4, 5$. Deste modo, tem-se:

$$\begin{aligned}
&\sum_{i\in F_1}\sum_{r=0}^{1}\sum_{v\in S_1} x_{i(j-r)v} = \\
&= \underbrace{(x_{1,12,2} + x_{2,12,2} + x_{3,12,2} + x_{4,12,2})}_{v=2} + \underbrace{(x_{1,12,3} + x_{2,12,3} + x_{3,12,3} + x_{4,12,3})}_{v=3} + \\
&+ \underbrace{(x_{1,12,4} + x_{2,12,4} + x_{3,12,4} + x_{4,12,4})}_{v=4} + \underbrace{(x_{1,12,5} + x_{2,12,5} + x_{3,12,5} + x_{4,12,5})}_{v=5} + \\
&+ \underbrace{(x_{1,11,2} + x_{2,11,2} + x_{3,11,2} + x_{4,11,2})}_{v=2} + \underbrace{(x_{1,11,3} + x_{2,11,3} + x_{3,11,3} + x_{4,11,3})}_{v=3} + \\
&+ \underbrace{(x_{1,11,4} + x_{2,11,4} + x_{3,11,4} + x_{4,11,4})}_{v=4} + \underbrace{(x_{1,11,5} + x_{2,11,5} + x_{3,11,5} + x_{4,11,5})}_{v=5} = 0.
\end{aligned}$$

Desta forma, a soma

$$\Phi = \sum_{i\in F_p}\sum_{r=0}^{t_i-1}\sum_{v\in S_k} x_{i(j-r)v},\ p = 1..N_f, j = 1..M, k = 1..L$$

deverá assumir valor 0 quando

$$\sum_{i\in F_1} x_{i,12,1} = \overbrace{x_{1,12,1}}^{1} + x_{2,12,1} + x_{3,12,1} + x_{4,12,1} = 1$$

ou de forma geral

$$\Psi = \sum_{i\in F_p} x_{ijk} = 1,\ p = 1..N_f,\ j = 1..M, k = 1..L.$$

Com isto, se uma planta da família p for semeada no período j em algum lote, então todos os vizinhos deste lote, $t_i - 1$ períodos anteriores à j, não poderão receber qualquer planta desta família.

Por outro lado, se em $j = 12$, no lote 1, nenhuma cultura pertencente à F_1 for plantada (implicando $\Psi = 0$), pode-se ter uma alocação factível na região S_1, nos períodos 11 e 12, fazendo com que o valor de Φ exceda 0. A alocação deve ser feita, deixando as culturas da família 1 separadas. Suponha, que a cultura 1 esteja nos lotes 2 e 4 (lotes distantes). Desta forma, os lotes 3 e 5 não poderão abrigar esta família nos períodos considerados, tendo em vista a vizinhança com os lotes já programados. Assim, a soma Φ para a família botânica 1 será:

$$\Phi = \sum_{i \in F_p} \sum_{r=0}^{t_i - 1} \sum_{v \in S_k} x_{i(j-r)v} =$$
$$+ \underbrace{(x_{1,12,2} + x_{2,12,2} + x_{3,12,2} + x_{4,12,2})}_{v=2} + \underbrace{(x_{1,12,3} + x_{2,12,3} + x_{3,12,3} + x_{4,12,3})}_{v=3} +$$
$$+ \underbrace{(x_{1,12,4} + x_{2,12,4} + x_{3,12,4} + x_{4,12,4})}_{v=4} + \underbrace{(x_{1,12,5} + x_{2,12,5} + x_{3,12,5} + x_{4,12,5})}_{v=5} +$$
$$+ \underbrace{(\overbrace{x_{1,11,2}}^{1} + x_{2,11,2} + x_{3,11,2} + x_{4,11,2})}_{v=2} + \underbrace{(x_{1,11,3} + x_{2,11,3} + x_{3,11,3} + x_{4,11,3})}_{v=3} +$$
$$+ \underbrace{(\overbrace{x_{1,11,4}}^{1} + x_{2,11,4} + x_{3,11,4} + x_{4,11,4})}_{v=4} + \underbrace{(x_{1,11,5} + x_{2,11,5} + x_{3,11,5} + x_{4,11,5})}_{v=5} = 2.$$

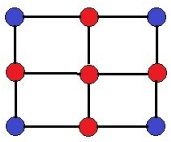

Figura 18 - Um grafo com seus conjuntos independentes

Um conjunto independente de um grafo $G(V, A)$ é um subconjunto de vértices no qual não existem dois vértices adjacentes. Em outras palavras, se u e v são vértices quaisquer de um conjunto independente, não há aresta entre u e v. Assim, na Figura (18), os dois conjuntos de vértices coloridos com mesma cor são independentes.

No exemplo dado anteriormente, a quantidade de lotes que poderiam receber a cultura pertencente à F_1, nesse caso, é igual ao número de lotes diferentes do lote 1 e que são vizinhos, nesse caso em 2 lotes possíveis (nos lotes 2 e 4 ou nos lotes 3 e 5). Assim, o valor Φ associado ao lote k ($\Phi(k)$) é obtido tomando-se o subgrafo da vizinhança do lote k e calculando-se a quantidade dos conjuntos independentes possuídos.

Segundo Diestel (2000), $\Phi(k)$ é o ótimo do problema a seguir:

Maximizar

$$\Phi(k) = \sum_{(u,v)\in A} y_u \tag{15}$$

Sujeito a

$$y_u + y_v \leq 1,\ (u,v) \in A \tag{16}$$

$$y_k = 0;\ y_u, y_v \in \{0,1\} \tag{17}$$

Desta forma, a quantidade de lotes L da área de plantio é um limitante superior para $\Phi(k)$, valor que foi utilizado na restrição de vizinhança (9), generalizada para todas as famílias, períodos e lotes:

$$\sum_{i\in F_p}\sum_{r=0}^{t_i-1}\sum_{v\in S_k} x_{i(j-r)v} \leq \Phi(k)\left(1 - \sum_{i\in F_p} x_{ijk}\right), p = 1..N_f, j = 1..M, k = 1..L.$$

Segundo Hell et al. (2002), o problema (15-17) é NP-difícil, pelo que a utilização do parâmetro $\Phi(k)$ na modelagem das restrições (9) poderia trazer problemas de eficiência computacional.

2.7.2 Modelagem Matemática para um Problema de Rotação de Culturas com restrições de adjacências e demanda

Afim de aproximar o modelo PRC-A com a realidade, inseriu-se mais restrição de demanda, denominando-o de PRC-A-D. Este novo problema, por sua vez, faz com que a produção de todas as culturas pertencentes ao conjunto C (comerciais)

atendam uma determinada demanda, num intervalo I_i^D em que esta demanda é solicitada. Assim, a nova restrição

$$\sum_{j \in I_i^D} \sum_{k=1}^{L} area_k \, p_{ij} \, x_{ijk} \geq D_i, \; i \in C \tag{18}$$

obriga a cultura i, plantada no período j, num lote k de área $area_k$, produzir no mínimo a sua demanda D_i solicitada no intervalo $I_i^D = [C_i^D, T_i^D]$, onde C_i^D e T_i^D é o período mais cedo e mais tarde da demanda, respectivamente. Assim, entram para o cômputo da produção somente as culturas que forem colhidas neste intervalo.

Como as culturas, na prática, têm suas produtividades (p) e lucratividades (l) afetadas com a época em que são colhidas, considerou-se neste modelo, estes parâmetros variantes com a época j, isto é, p_{ij} e l_{ij} denotam, respectivamente, a produtividade e a lucratividade da cultura i no período j, por hectare.

Consequentemente, o modelo PRC-A-D terá como função objetivo:

$$\text{maximize } z = \sum_{i \in C} \sum_{j \in I_i} \sum_{k=1}^{L} area_k \, l_{ij} \, x_{ijk} \tag{19}$$

sujeito a mesmas restições do PRC-A de vizinhança (9), de plantio consecutivo (10), da espera de plantio (11), de adubação verde (12), de pousio (13) e a nova restrição de demanda (18).

3 MATERIAL E MÉTODOS

Todos os experimentos computacionais foram realizados no software Matlab versão 7.4.0 R2007a, em micro-computadores Core 2 Quad com 2GB de memória de 250 GB de memória de disco rígido, no Laboratório Científico de Informática (LCI) do Departamento de Bioestatística do IB/UNESP de Botucatu/SP. Para cada uma das quatro metaheurísticas, foram propostas algumas variações dos parâmetros, afim de investigar a influência dos mesmos e determinar qual configuração se reproduziu melhor. As melhores combinações foram comparadas, em termos do tempo computacional, qualidade das soluções encontradas, percentagem de convergência à soluções factíveis, número de penalizações na solução final e finalmente determinar a metaheurística que possuiu o melhor desempenho respectivamente com os seus parâmetros, tanto para o Problema de Rotação de Culturas com restrições de Adjacências e Demanda. Cada algoritmo proposto foi simulado 100 vezes.

Para aplicar e testar os métodos heurísticos à um PRC, duas fases foram realizadas. Em ambas, utilizou-se os mesmos os dados intríncecos às culturas, disponibilizados na Tabela (27), retirados de Santos (2005); Filgueira (2003) e Souza & Resende (2006). Trabalhou-se com $N = 29$ culturas disponíveis, de $N_f = 10$ famílias botânicas de plantas distintas, sendo 5 delas para adubação verde. Nestes dados estão contidas a época de cultivo, a duração do ciclo de vida, a família que pertencem, a quantidade de períodos t_i de cada planta. Na primeira fase das simulações, o PRC com restrições de adjacências foi testado para três instâncias, com L igual à 10, 15 e 20, cuja estrutura geográfica foi hipotética, esquematizada na Figura (29). Os valores l_i das áreas de cada lote bem como as lucratividades destas

culturas também foram dados hipotéticos, todos exibidos no Apêndice B nas Tabelas (28) e (29) respectivamente.

A segunda fase foi uma aplicação do modelo PRC com restrições de demanda para dados reais das lucratividades (l_{ij}), produtividade (p_{ij}), demanda (D_i), intervalos das demandas (I_i^D), para uma geometria e área de plantio como na Figura (30), dividida em 16 lotes em forma de "tabuleiro". Os valores das áreas para estes lotes encontram-se na Tabela (30). Estes dados foram gentilmente fornecidos pelo Sr. Sebastião Soares de Oliveira, proprietário de um centro de produção de hortaliças localizado no município de Santa Cruz do Rio Pardo/SP, tendo em vista que esta propriedade serve o mercado atacadista da região. Os períodos de demanda I_i^D e quantidade demandada D_i, lucratividade l_{ij} e produtividade p_{ij} reais estão, respectivamente, nas Tabelas (31), (32) e (33) no Apêndice C.

Cada período foi fixado em um mês, o tamanho de rotação em um ano (M=12) para a primeira fase das simulações e de dois anos ($M = 24$) na segunda. Estabeleceu-se um período de pousio de um mês para todos os lotes. As plantas 1-25 serão cultivadas para serem comercializadas (conjunto C). Já as culturas 23-29 são as selecionadas para adubação verde (conjunto A), desempenhando funções ecológicas de recuperação do solo. O pousio é representado pela cultura $n = N + 1 = 30$, e pode ser feito em qualquer época do ano.

3.1 Estratégias de resolução

3.1.1 Mudança nas variáveis

Como foi descrito na seções 2.7.1 e 2.7.2, a modelagem matemática desenvolvida para os PRCs baseia-se em variáveis binárias. Desta forma, a quantidade de variáveis decisórias x_{ijk}, neste caso, será igual à $2^{(N+1)\times M\times L}$. Para o conjunto de dados reais, adotados nas simulações, esta quantidade é elevada para o emprego dos métodos heurísticos como a BL, tendo em vista que trabalham com estruturas vizinhanças de soluções (métodos SA e AG+SA) e com cromossomos (métodos AG,

AG+SA e Memético).

Assim, para um uso mais correto e eficaz destes métodos, adota-se uma estratégia de resolução, onde as variáveis decisórias passam a ser inteiras, com intervalo de variação [1,N+1], estruturadas em uma matriz solução $\mathbf{S}_{L\times M}$, em que cada elemento $s(k,j) \in [1, N+1]$, $k \in \{1..L\}$ e $j \in \{1..M\}$, representa que a cultura $s(k,j)$ está sendo plantada no lote k e no período j. Em outras palavras, cada linha da matriz **S** repesenta uma programação de plantio para o respectivo lote nos seus M períodos.

3.1.2 Heurística construtiva

A heurística construtiva para os problemas PRC-A e PRC-A-D, mais precisamente a construção aleatória descrita na seção 2.2, foi igualmente utilizada pelos quatro métodos, afim de gerar soluções nas quais a busca começará a ser efetuada. No caso do AG, o AG+SA e o Memético, um número P de soluções é construído ao passo que no SA é apenas uma.

Esta construção de soluções permitiu elaborar soluções cujas culturas são plantadas no período correto e sempre atenderem à restrição (10) de plantio consecutivo do modelo, tendo em vista que estas restrições são as mais difíceis de serem atendidas. Desta forma, os métodos heurístico procuram eliminar nas soluções correntes aquelas que não atendem as restrições de vizinhança, de adubação verde, de pousio e de demanda. Nota-se que com esta representação de soluções, as restrições da espera de plantio (11) para os modelos já estão satisfeitas com a recodificação realizada.

Esta heurística de construção tem o seu funcionamento descrito como segue. Uma permutação aleatória (cíclica) das $(N+1)$ culturas é selecionada. Para construir uma rotação em um lote, toma-se, na permutação, a primeira cultura que puder ser plantada no primeiro período. Uma vez que esta é encontrada, insere-se na programação e é escrita na matriz tantas vezes quanto for o seu ciclo de vida. Assim, a *q-ésima* cultura entrará na programação deste lote, se puder ser plantada

no período

$$1 + \sum_{i=1}^{q-1} t_i$$

e não for da mesma família botânica que a cultura $q-1$, com excessão da cultura $N+1$, que pôde ser "cultivada" consecutivamente (pousio). Este processo se repete até que os M períodos sejam preenchidos. Finalmente, uma solução consistirá em repetir este processo L vezes. A Figura (19) exibe o seu pseudocódigo, sendo **S** a programação de plantio nos L lotes e $s(k,j)$ o elemento genérico desta matriz.

Algoritmo 7 *ConstruçãoPRC*$(N,\ M,\ L,\ t_i,\ I_i,\ F_p)$

1 $\mathbf{S} = \mathbf{0}$ {inicialmente **S** tem todos os elementos nulos};
2 para k de 1 a L faça:
3 $\quad dif = M$;
4 $\quad$ escolher uma permutação Ω das $(N+1)$ culturas;
5 $\quad$ procure a cultura i_1 em Ω, de índice c, que pode ser plantada no período 1;
6 $\quad s(k,1) = s(k,2) = \cdots = s(k,t_{i_1}) = i_1$;
7 $\quad dif = M - ti_1$;
8 $\quad$ enquanto $(dif > 0)$ faça:
9 $\quad\quad$ para q de $c+1$ a $(N+1)$ faça:
10 $\quad\quad\quad$ se $(t_q \leq dif)$ faça:
11 $\quad\quad\quad\quad$ se $\left(\left(r = 1 + \sum_{k=1}^{q-1} t_k\right) \in I_q\right)$;
12 $\quad\quad\quad\quad\quad$ se $(F_{q-1} \neq F_q)$ ou $(q = N+1)$;
13 $\quad\quad\quad\quad\quad\quad s(k,r) = s(k,r+1) = \cdots = s(k,r+t_q-1) = i_q$;
14 $\quad\quad\quad\quad\quad$ fim-se
15 $\quad\quad\quad\quad\quad dif = dif - (\sum_{k=1}^{q} t_k)$;
16 $\quad\quad\quad\quad$ fim-se
17 $\quad\quad\quad$ fim-se
18 $\quad\quad$ fim-para
19 $\quad$ fim-enquanto
20 fim-para
21 retorne **S**;
fim *ConstruçãoPRC*

Figura 19 - Heurística de construtiva para o PRC

A próxima subseção descreve como as soluções que violam as restrições de vizinhança, adubação verde, pousio e demanda foram tratadas pelos métodos.

3.1.3 Penalização das soluções

O valor $z = f(\mathbf{S})$ associado a uma solução $\mathbf{S}$ foi definido como a soma das lucratividades l_i das culturas em todos os lotes. De modo a fazer os métodos convergirem para soluções factíveis, adotou-se uma metodologia de penalização de soluções infactíveis. Esta penalização, por sua vez, depende da quantidade de restrições que a programação em análise possui. Pretende-se fazer com que as soluções com poucas infactibilidades tenham seu lucro menos reduzido em comparação com uma que possui um elevado número de violações.

Para cada período da rotação, deve-se verificar se nos lotes adjacentes há plantio de culturas da mesma família botânica. Assumindo que esta coincidência ocorre w vezes em todos os M períodos, a quantidade de penalizações de vizinhos (P.V) será de $w - 1$. Uma solução que possui r penalizações de adubação verde (P.A.V) é aquela que não contemplou adubação verde em r lotes durante todo o período considerado. De forma similar, aplica-se este conceito para penalizações de pousio (P.P). As penalizações de Demanda (D) são computadas determinando-se a quantidade de culturas que não atenderam à demanda solicitada.

Um método de penalização exponencial para penalizá-las foi utilizado. Desta forma, todas as penalizações são adicionadas. Seja $p(\mathbf{S})$ este valor. Para cada solução

$$p(\mathbf{S}) \leftarrow [p(\mathbf{S})]\, e^{-\frac{p(\mathbf{S})}{K}} \tag{20}$$

em que K é uma constante a ser definida para medir a "força" desta penalização. É claro que uma progamação factível não sofrerá nenhuma redução em sua lucratividade por este método.

Por exemplo, dada a área de plantio como na Figura (18-a), as culturas disponíveis, família a que pertencem e duração do ciclo de vida iguais as exibidas

na seção 2.7.1, considera-se ainda número de períodos igual a 10 e o pousio sendo a cultura indicada pelo número 11, cuja duração foi fixada em 1 período. Por simplicidade, admite-se que todas as culturas podem ser cultivadas em qualquer época do calendário considerado. A adubação verde é a família F_3 (culturas 9 e 10).

Então uma matriz solução $\mathbf{S}_{5\times10}$, construída pela heurística explicitada anteriormente, poderia ser

$$\mathbf{S} = \begin{pmatrix} 1 & 1 & 5 & 5 & 9 & 9 & 2 & 2 & 11 & 11 \\ 6 & 6 & 11 & 7 & 7 & 3 & 3 & 11 & 8 & 8 \\ 9 & 9 & 2 & 2 & 11 & 10 & 10 & 7 & 7 & 11 \\ 8 & 8 & 3 & 3 & 7 & 7 & 4 & 4 & 7 & 7 \\ 4 & 4 & 10 & 10 & 6 & 6 & 3 & 3 & 9 & 9 \end{pmatrix}$$

Considerando-se a terminologia $PV_{i,j}$ sendo o número de violações de vizinhanças (em vermelho) entre os lotes i e j, tem-se nesta solução:

1. P.V $= PV_{1,2} + PV_{1,3} + PV_{1,4} + PV_{1,5} + PV_{2,3} + PV_{2,5} + PV_{3,4} + PV_{4,5} = 2 + 1 + 2 + 4 + 1 + 2 + 3 + 4 = 19$;

2. P.A.V $= 2$, pois os lotes 2 e 4 não tiveram adubação verde;

3. P.P $= 2$, pois os lotes 4 e 5 não tiveram o pousio.

Desta forma $p(\mathbf{S}) = 19 + 2 + 2 = 23$ penalizações no total. Se $K = 10$, $F(\mathbf{S})$ será multiplicada por $e^{-23/10} \simeq 0,1$. É claro que se pode dar uma penalização mais "branda" ou "rigorosa" às soluções diminuindo-se ou aumentando-se o valor do parâmetro de contole K.

A penalização exponencial se justifica pois uma das principais dificuldades deste problema é o elevado número de restrições. Nesse sentido, uma penalização "severa" para as programações foi proposta. Assim, mesmo uma rotação com p pequeno teria sua lucratividade mais afetada se a metodologia de penalidade fosse proporcional à p. Em todas as simulações, $K = 10$.

Nas quatro subseções que seguem, apresenta-se as metodologias desenvolvidas para os quatro algoritmos, pontuando os parâmetros qualitativos e quanti-

tativos utilizados.

3.2 Algoritmo Genético

3.2.1 População inicial, Elitismo e Seleção

O AG para o este problema teve como **População inicial** P matrizes construídas pela heurística da seção 3.1.2, respeitando as restrições de plantio consecutivo e de época de plantio.

As P matrizes foram avaliadas e penalizadas utilizando o processo descrito na seção anterior. Uma vez obtendo a aptidão de cada cromossomo, o melhor cromossomo (aquele que faz a função objetivo atingir o maior valor) será separado, colocando-o em uma varíável denominada *elite*, caracterizando o operador **Elitismo**. Este elemento, por sua vez, não passará nos operadores de seleção, *crossover*, mutação e migração, de modo a impedir que o método destrua a melhor solução até então encontrada.

Após isto, $c\,P$ cromossomos foram selecionados para realizar o crossover, em que c é a taxa de *crossover*. Utilizou-se quatro métodos distintos de seleção:

(a) **Roleta viciada;**

(b) **Torneio;**

(c) **Elitista truncada.**

Os AGs com estes processos seletivos, serão denotados respectivamente por: R, T e E. Na **Roleta viciada**, houve a necessidade de realizar um escalonamento linear, com parâmetro Max, em razão de haver um elevado número de matrizes com aptidão semelhante, procurando equilibrar os processos de diversificação e não aleatoriedade do método.

3.2.2 *Crossover*

Utilizou-se 3 métodos de *crossover*, a saber:

(a) ***crossover* 2-pontos;**

(b) ***crossover* 3-pontos;**

(c) ***crossover* uniforme.**

e serão denotados respectivamente por $2P$, $3P$ e U.

Em todos estes métodos de realizar o *crossover*, dois pais $\mathbf{P}_1$ e $\mathbf{P}_2$ são pareados de forma aleatória, consistindo em um *mating* aleatório, gerando-se dois filhos $\mathbf{F}_1$ e $\mathbf{F}_2$.

Tendo em vista a codificação e as características particulares deste problema, os "cortes" aleatórios nas matrizes foram realizados horizontalmente. Deste modo, os ciclos de vida das culturas não serão interrompidos, as restrições de plantio consecutivo de culturas de mesma família serão sempre satisfeitas e as épocas de plantio serão respeitadas, asseguradas na construção das programações.

Para ilustrar o *crossover* uniforme, sejam os cromossomos pais:

$$\mathbf{P}_1 = \begin{pmatrix} 2 & 2 & 6 & 6 & 2 & 2 & 11 & 7 & 3 & 3 \\ 7 & 7 & 3 & 3 & 5 & 5 & 9 & 9 & 2 & 2 \\ 4 & 4 & 6 & 6 & 1 & 1 & 5 & 5 & 3 & 11 \\ 1 & 1 & 10 & 10 & 3 & 3 & 6 & 6 & 11 & 11 \\ 9 & 9 & 5 & 5 & 11 & 4 & 4 & 9 & 9 & 9 \end{pmatrix}$$

$$\mathbf{P}_2 = \begin{pmatrix} 5 & 5 & 1 & 1 & 5 & 5 & 3 & 3 & 5 & 5 \\ 3 & 3 & 6 & 6 & 2 & 2 & 5 & 5 & 2 & 2 \\ 10 & 10 & 2 & 2 & 10 & 10 & 1 & 1 & 9 & 9 \\ 8 & 8 & 3 & 3 & 7 & 7 & 9 & 9 & 11 & 11 \\ 1 & 1 & 9 & 9 & 4 & 4 & 6 & 6 & 2 & 2 \end{pmatrix}$$

e a máscara aleatória de tamanho $L = 5$:

$$m = (1\ 0\ 1\ 0\ 0)^t.$$

O filho $\mathbf{F}_1$ será obtido tomando-se as linhas do pai $\mathbf{P}_1$ onde a máscara indicar 1 e as linhas do pai $\mathbf{P}_2$ onde o valor da máscara for 0. Neste caso, este filho herdará do primeiro pai os genes 1 e 3 e do segundo pai os genes 2, 4 e 5.

$$\mathbf{F}_1 = \begin{pmatrix} 2 & 2 & 6 & 6 & 2 & 2 & 11 & 7 & 3 & 3 \\ 3 & 3 & 6 & 6 & 2 & 2 & 5 & 5 & 2 & 2 \\ 4 & 4 & 6 & 6 & 1 & 1 & 5 & 5 & 3 & 11 \\ 8 & 8 & 3 & 3 & 7 & 7 & 9 & 9 & 11 & 11 \\ 1 & 1 & 9 & 9 & 4 & 4 & 6 & 6 & 2 & 2 \end{pmatrix}.$$

De forma similar, o filho $\mathbf{F}_2$ será formado tomando-se os genes dos pais de maneira inversa: se a linha k da máscara indicar 1, a informação vem da k-ésima linha do pai 2 e se for 0 do pai 1, assim:

$$\mathbf{F}_2 = \begin{pmatrix} 5 & 5 & 1 & 1 & 5 & 5 & 3 & 3 & 5 & 5 \\ 7 & 7 & 3 & 3 & 5 & 5 & 9 & 9 & 2 & 2 \\ 10 & 10 & 2 & 2 & 10 & 10 & 1 & 1 & 9 & 9 \\ 1 & 1 & 10 & 10 & 3 & 3 & 6 & 6 & 11 & 11 \\ 9 & 9 & 5 & 5 & 11 & 4 & 4 & 9 & 9 & 9 \end{pmatrix}.$$

Para ilustrar o *crossover* de 3-pontos, considera-se os mesmos cromossomos pais e os dois números aleatórios, 2 e 4, indicando o local dos cortes horizontais realizados. O primeiro descendente será obtido, nesta sequência, tomando-se as duas primeira linhas do pai 1, as linhas 3 e 4 do pai 2 e finalmente a linha 5 do pai 1:

$$\mathbf{F}_1 = \begin{pmatrix} 2 & 2 & 6 & 6 & 2 & 2 & 11 & 7 & 3 & 3 \\ 7 & 7 & 3 & 3 & 5 & 5 & 9 & 9 & 2 & 2 \\ 10 & 10 & 2 & 2 & 10 & 10 & 1 & 1 & 9 & 9 \\ 8 & 8 & 3 & 3 & 7 & 7 & 9 & 9 & 11 & 11 \\ 9 & 9 & 5 & 5 & 11 & 4 & 4 & 9 & 9 & 9 \end{pmatrix}.$$

Similarmente a construção do filho 2: toma-se, nesta ordem, as linhas 1 e 2 do pai 2, as linhas 3 e 4 do pai 1 e a linha 5 do pai 2:

$$\mathbf{F}_2 = \begin{pmatrix} 5 & 5 & 1 & 1 & 5 & 5 & 3 & 3 & 5 & 5 \\ 3 & 3 & 6 & 6 & 2 & 2 & 5 & 5 & 2 & 2 \\ 4 & 4 & 6 & 6 & 1 & 1 & 5 & 5 & 3 & 11 \\ 1 & 1 & 10 & 10 & 3 & 3 & 6 & 6 & 11 & 11 \\ 1 & 1 & 9 & 9 & 4 & 4 & 6 & 6 & 2 & 2 \end{pmatrix}.$$

Similarmente fez-se o *crossover* 2-pontos: se o corte for realizado na linha $k \leq L$ então o filho 1 herdará as mesmas informações do pai 1 até a *k-ésima* linha e do pai 2 da $(k+1)$-*ésima* linha até L. Já o filho 2, tomará do pai 2 as linhas $\{1, 2..k\}$ e do pai 1 as linhas $\{k+1, k+2..L\}$, sempre preservando a alocação original dos genes.

3.2.3 Mutação e Migração

Após a população passar à fase de *crossover*, tem-se $(c+1)\,P$ cromossomos[5] (pais mais os filhos). Ordenando-a decrescentemente, pelo valor da aptidão,

[5]Considera-se que o número P seja divisível por três, ou alternativamente, toma-se a parte inteira da divisão de P por 3.

e dividindo-a em 4 classes distintas, nomeou-se:

(a) **classe I** - do 2° ao $\left(\frac{P}{3}\right)$-*ésimo* indivíduo;

(b) **classe II** - do $\left(\frac{P}{3}+1\right)$-*ésimo* ao $\left(2\frac{P}{3}\right)$-*ésimo* indivíduo;

(c) **classe III** - do $\left(2\frac{P}{3}+1\right)$-*ésimo* ao (P)-*ésimo* indivíduo;

(d) **classe IV** - do $(P+1)$-*ésimo* ao $((c+1)\,P)$-*ésimo* indivíduo.

Esta divisão foi importante, pois permitiu que os operadores de **mutação** e **migração** atuassem em setores distintos da população, impedindo, por exemplo, que um mesmo cromossomo sofresse ações de dois operadores concomitantemente. O operador **mutação** atuou somente em uma fração θ_2 da classe II.

Como descrita na subseção 2.3.6, trabalhos como o de Constantino et al. (2003) propõem uma taxa de mutação variável conforme a população vai se desenvolvendo. Assim, esta probabilidade foi feita do seguinte modo: fixa-se um parâmetro θ_1 como sendo a probabilidade de haver mutação na geração $g = 1$ do algoritmo. Sendo g corrente, então a probabilidade de mutação $mt(g)$, em função de g, será dada pela função logística:

$$mt(g) = \frac{\theta_1}{\theta_1 + e^{-g/10}} \tag{21}$$

isto é, no início do método, em razão de haver muita diversidade de cromossomos a mutação ocorre em baixa probabilidade. Por outro lado, nas gerações finais, onde a quantidade de cromossomos com as mesmas características genéticas é alta, há um esforço em introduzir novos cromossomos através deste operador, com o intuito de impedir a convergência prematura e a formação dos "super-indivíduos". A Figura (20) ilustra o comportamento desta função para 100 gerações e $\theta_1 = 0,01$.

Para cada um dos $\frac{P\theta_2}{3}$ indivíduos candidatos a se transfomarem, cada gene seu poderá ser trocado com uma probabilidade θ_3. A formação de novos genes (que corresponde à uma linha da matriz solução) é realizada segundo a heurística da

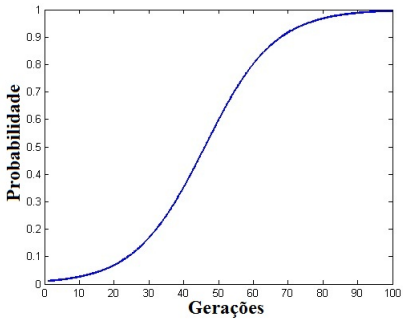

Figura 20 - Probabilidade de mutação

seção 3.1.2 explicitada anteriormente. Desta forma, a mutação pode alterar parcialmente ou totalmente a estrutura genética de um ou mais cromossomos. Este tipo de mutação se justifica em uma estratégia de diversificação para explorar o espaço de busca e uma tentativa de se escapar de soluções infactíveis.

Similarmente, o operador **migração** opera somente na classe I da população com probabilidade mg dada por

$$mg(g) = \frac{\eta_1}{\eta_1 + e^{-g/10}} \tag{22}$$

em que η_1 indica a probabilidade de se ocorrer **migração** na primeira geração.

Este operador gênico opera selecionando uma fração η_2 da classe I. Estes indivíduos serão exterminados e no lugar serão geradas $\eta_2\,(P)$ soluções usando o algoritmo construtivo anterior.

Em razão do elevado número de restrições que este problema possui e da dificuldade de encontrar uma solução factível, quando uma solução com esta característica é encontrada, a sua aptidão é disparadamente superior às demais, que faz o método "valorizá-la". Com isto, o método passa a gerar muitas cópias idênticas à esta, estagnando-se rapidamente.

Afim de impedir este frequente comportamento, propõe-se uma segunda migração, a **migração em massa**, utilizando o mesmo princípio da original: na geração $\gamma\,G$, em que G é a quantidade de gerações proposta, a classe III é dizi-

mada e em seu lugar uma sub-população de mesmo tamanho é gerada utilizando a heurística construtiva. Desta forma, preserva-se os $\frac{2}{3}$ dos melhores cromossomos e coloca-se $\frac{1}{3}$ de um material genético completamente novo, permitindo que o método possa se iteragir com estes novos indivíduos por mais algumas gerações e fazendo a solução atingir níveis mais promissores.

Finalmente, a população após passar por todos estes operadores, será reduzida, selecionando-se os P mais aptos. O processo voltará a se repetir até completar as G gerações.

Os valores dos parâmetros utilizados neste método para o PRC-A estão exibidos na Tabela (2) e foram obtidos por exaustivas simulações. Já para o PRC-A-D, em se tratando de uma aplicação com dados reais e por possuir um grande número de restrições, utilizou-se um AG_{R}_{U}_{*120;601*} para solucioná-lo.

Para diferenciar os AGs simulados, utilizu-se a terminologia AG_{*tipo_seleção*}_{*tipo_crossover*}_{*parâmetros_quantitativos*} que representa um AG com um tipo de seleção e *crossover* abordados nas seções 3.2.1 e 3.2.2, respectivamente. O conjunto {*parâmetros_quantitativos*} terá os dois parâmetros {G,P}, pois os demais foram fixos.

3.3 *Simulated Annealing*

Um SA tanto para o Problema de Rotação de Culturas foi desenvolvido neste trabalho. A solução inicial para este procedimento foi obtida construtivamente, utilizando a heurística descrita na seção 3.1.2 Neste método, fez-se necessária a definição de uma estrutura de vizinhança eficaz que leve em consideração dois aspectos: a simetria e a mesma probabilidade de acesso a qualquer vizinho definido como afirmam os autores Kirkpatrick et al. (1983) e Dowsland (1993).

Desta forma, a vizinhança empregada neste trabalho foi a seguinte: dada uma solução qualquer $\mathbf{S}$ (com L linhas), um vizinho $\mathbf{S}'$ será uma matriz que possui $L-1$ linhas de $\mathbf{S}$. A diferença entre elas está somente na linha faltante.

Pensando que o nível de dificuldade e a quantidade de soluções in-

Tabela 2 - Parâmetros utilizados no AG para solucionar o PRC-A

Operador	Parâmetro	Valor
Seleção	Max	1,3
	τ	2
Crossover	c	0,8
Mutação	θ_1	0,01
	θ_2	0,1
	θ_3	0,5
Migração	η_1	0,01
	η_2	0,1
	γ	0,7
Elementos na população	P	{121,241}
Gerações	G	{100,150}

factíveis aumenta à medida que a dimensão problema cresce, é plausível definir a cardinalidade conjunto das vizinhanças de $\mathbf{S}$ ser L, isto é, quanto maior é a instância do problema, maior será a busca nas vizinhança em torno de uma dada solução[6].

Seja uma solução $\mathbf{S}_{5\times10}$

$$\mathbf{S} = \begin{pmatrix} 2 & 2 & 5 & 5 & 9 & 9 & 8 & 8 & 3 & 3 \\ 5 & 5 & 1 & 1 & 10 & 10 & 11 & 11 & 5 & 5 \\ 4 & 4 & 9 & 9 & 3 & 3 & 10 & 10 & 2 & 2 \\ 1 & 1 & 8 & 8 & 4 & 4 & 9 & 9 & 11 & 11 \\ 6 & 6 & 2 & 2 & 9 & 9 & 7 & 7 & 10 & 10 \end{pmatrix}$$

os 5 possíveis vizinhos $\mathbf{V}_i$ $i \in \{1,..,5\}$ de $\mathbf{S}$ serão

$$v_1 = \begin{pmatrix} 6 & 6 & 1 & 1 & 10 & 10 & 5 & 5 & 2 & 2 \\ 5 & 5 & 1 & 1 & 10 & 10 & 11 & 11 & 5 & 5 \\ 4 & 4 & 9 & 9 & 3 & 3 & 10 & 10 & 2 & 2 \\ 1 & 1 & 8 & 8 & 4 & 4 & 9 & 9 & 11 & 11 \\ 6 & 6 & 2 & 2 & 9 & 9 & 7 & 7 & 10 & 10 \end{pmatrix} \quad v_2 = \begin{pmatrix} 2 & 2 & 5 & 5 & 9 & 9 & 8 & 8 & 3 & 3 \\ 3 & 3 & 10 & 10 & 8 & 8 & 5 & 5 & 1 & 1 \\ 4 & 4 & 9 & 9 & 3 & 3 & 10 & 10 & 2 & 2 \\ 1 & 1 & 8 & 8 & 4 & 4 & 9 & 9 & 11 & 11 \\ 6 & 6 & 2 & 2 & 9 & 9 & 7 & 7 & 10 & 10 \end{pmatrix}$$

[6]Embora $\mathbf{S}$, a partir deste definição, possa ter uma quantidade de vizinhos muito maior, o conjunto vizinhança teve poucos elementos para que algoritmo pudesse ter um tempo computacional exiquível.

$$v_3 = \begin{pmatrix} 2 & 2 & 5 & 5 & 9 & 9 & 8 & 8 & 3 & 3 \\ 5 & 5 & 1 & 1 & 10 & 10 & 11 & 11 & 5 & 5 \\ 2 & 2 & 8 & 10 & 11 & 5 & 5 & 11 & 7 & 7 \\ 1 & 1 & 8 & 8 & 4 & 4 & 9 & 9 & 11 & 11 \\ 6 & 6 & 2 & 2 & 9 & 9 & 7 & 7 & 10 & 10 \end{pmatrix} \quad v_4 = \begin{pmatrix} 2 & 2 & 5 & 5 & 9 & 9 & 8 & 8 & 3 & 3 \\ 5 & 5 & 1 & 1 & 10 & 10 & 11 & 11 & 5 & 5 \\ 4 & 4 & 9 & 9 & 3 & 3 & 10 & 10 & 2 & 2 \\ 1 & 1 & 6 & 6 & 5 & 5 & 2 & 2 & 6 & 6 \\ 6 & 6 & 2 & 2 & 9 & 9 & 7 & 7 & 10 & 10 \end{pmatrix}$$

$$v_5 = \begin{pmatrix} 2 & 2 & 5 & 5 & 9 & 9 & 8 & 8 & 3 & 3 \\ 5 & 5 & 1 & 1 & 10 & 10 & 11 & 11 & 5 & 5 \\ 4 & 4 & 9 & 9 & 3 & 3 & 10 & 10 & 2 & 2 \\ 1 & 1 & 8 & 8 & 4 & 4 & 9 & 9 & 11 & 11 \\ 10 & 10 & 5 & 5 & 2 & 2 & 7 & 7 & 3 & 3 \end{pmatrix}$$

Após definir e gerar os vizinhos da solução corrente, define-se o parâmetro $SAmax$ que mede a quantidade máxima de movimentos possíveis em uma isoterma: L. Sendo assim, no pior das hipóteses, dá-se a possibilidade do algoritmo percorrer toda a sua vizinhança antes de atualizar a temperatura, que por sua vez, é feita por geometricamente na razão β (taxa de arrefecimento) e temperatura inicial T_0. O critério de parada do método foi definido como sendo a temperatura de congelamento T_f.

Como descrito na seção 2.5.1, a equação (3) refere-se à probabilidade de mudança à uma solução de pior lucratividade. Afim de igualar a dimensão dos dois valores, T e Δ, considerou-se a diferença relativa entre os lucros das duas soluções comparadas, ou seja

$$\Delta = \frac{f(\mathbf{V}) - f(\mathbf{S})}{f(\mathbf{V})}. \tag{23}$$

A notação SA_$\{T_0, T_f, \beta\}$ evidencia um SA com os parâmetros já definidos nesta seção. Para o PRC-A, os valores dos parâmetros utilizados no algoritmo encontram-se na Tabela (3). Para o PRC-A-D usou-se um SA_$\{100; 10^{-8}; 0,99\}$.

Tabela 3 - Parâmetros utilizados no SA para solucionar o PRC-A

Parâmetro	**Valor**
T_0	{1;10;100}
T_f	$\{10^{-4}; 10^{-6}; 10^{-8}\}$
β	{0,95;0,99}
$SAmax$	L

3.4 Algoritmo Genético-*Simulated Annealing*

O método híbrido AG+SA foia terceira metaheurística utilizada. A idéia central deste procedimento consiste em encontrar, através do AG, uma "boa" solucão inicial para a segunda metaheurística efetuar a sua busca, o SA.

Desta forma, o AG_$\{R\}$_$\{U\}$_$\{G, P\}$ (ou simplesmente AG__$\{G, P\}$) gerou uma solução inicial para o SA_$\{T_0, T_f, \beta\}$, cujos parâmetros utilizados encontram-se na Tabela (4) seguinte.

Tabela 4 - Parâmetros utilizados no AG+SA para solucionar o PRC-A

Parâmetro	**Valor**
G	{50;80}
P	{61;91}
T_0	{1;10}
T_f	$\{10^{-4}; 10^{-6}\}$
β	{0,90;0,95}

Como se nota, os parâmetros do procedimento híbrido fazem das componentes isoladas deste método mais "fracas", em comparação com as abordagens puras realizadas nas seções 3.2 e 3.3. Com isto, deseja-se verificar se a junção dos

dois algoritmos mais “fracos” (método híbrido) é capaz de superar um único método de “maior força”, como definido nas seções anteriores.

O PRC-A-D foi solucionado pelo método híbrido AG_{100;301}+SA_{$100; 10^{-5}; 0,95$}.

3.5 Memético

Finalmente, o quarto método proposto foi o algoritmo Memético, que começa por um AG_{R}_{U}_{G, P} (ou simplesmente AG__{G, P}) cuja solução, será aplicada uma BL. Esta busca se baseia em um esquema de vizinhanças definido igualmente como no SA. A única diferença é a quantidade de vizinhos gerados para cada solução em curso. Neste caso, em se tratando de um método de refinamento de uma solução, o número de vizinho construídos foi de $(a^2\, L^2)$, com a o único parâmetro desta heurística, que agora, é denotada por BL_{a}. O critério de parada foi estabelecido como sendo $a^2\, L^2$ iterações ou comparações sem melhora na função objetivo z. Deste modo, se a solução em curso for a mais promissora em toda a sua vizinhança restrita, a BL é encerrada.

A Tabela (5) exibe os parâmetros utilizados neste método para o PRC-A, enquanto o Memético AG_{100;301}+BL_{32} foi utilizado para resolver o PRC-A-D.

Tabela 5 - Parâmetros utilizados no Memético para solucionar o PRC-A

Parâmetro	**Valor**
G	{50;80}
P	{61;91}
a	{2;5;10}

Observação: tanto no AG+SA quanto no Memético, os demais parâmetros concernentes ao AG, foram iguais aos definidos na Tabela (2).

4 RESULTADOS E DISCUSSÃO

Este capítulo apresenta os resultados computacionais obtidos a partir da resolução do PRC. Foram aplicadas as quatro metaheurísticas propostas, munidas das metodologias descritas capítulo 3. As subseções seguintes apresentam as avaliações dos desempenhos dos métodos e comparações.

Esta seção encontra-se dividiva em duas partes. Na primeira, com o intuito de avaliar o comportamento dos algoritmos implementados e desenvolvidos, estão apresentados os resultados para computacionais para o PRC-A, onde foram feitas simulações utilizando-se três instâncias com 10, 15 e 20 lotes e com dados hipotéticos. Na subseção seguinte, afim de aproximar o modelo com uma situação mais real, envolvendo-se restrições de demanda, ilustram-se os resultados para uma aplicação do PRC-A-D com dados reais.

Em todas as simulações e aplicações, os algoritmos foram testados 100 vezes. Deste modo, os resultados apresentados nas tabelas são valores médios.

4.1 Resultados para o PRC-A

4.1.1 Resultados obtidos pelo AG

Lembrando-se que AG_$\{R\}$_$\{2P\}$_$\{150; 121\}$, por exemplo, significa um AG com método de seleção por roleta, *crossover* 2-pontos com 150 gerações e população com 121 indivíduos. Nesta seção simulou-se para os seguintes números de lotes $L = 10$, $L = 15$ e $L = 20$ os seguintes AGs:

1) AG_$\{R\}$_$\{2P\}$_$\{100; 121\}$;

2) AG_{R}_{$2P$}_{100; 241};

3) AG_{R}_{$3P$}_{100; 121};

4) AG_{R}_{$3P$}_{100; 241};

5) AG_{R}_{U}_{100; 121};

6) AG_{R}_{U}_{100; 241};

7) AG_{R}_{U}_{150; 121};

8) AG_{R}_{U}_{150; 241};

9) AG_{T}_{U}_{150; 241};

10) AG_{E}_{U}_{150; 241}.

A Tabela (6) apresenta os resultados médios de tempo computacional para estes métodos.

Tabela 6 - Tempos computacionais médios (s) para cada instância utilizando os 10 AGs

AG	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Número de Lotes (L)										
10	7,1	17,2	7,4	17,5	7,9	19,5	12,5	30,1	27,5	28,1
15	11,2	31,2	12,1	31,2	13,1	36,2	20,4	55,7	52,4	54,5
20	17,1	49,4	17,1	49,6	19,5	56,3	30,0	89,5	86,9	86,1

Comparando os AGs 1, 3 e 5, que utilizam o mesmo processo de seleção (Roleta) com 100 gerações, 121 indivíduos e variam as metodologias de realizem o *crossover*, observa-se que houve pouca variação no desempenho computacional entre os métodos. A mesma observação pode ser estendida aos métodos 2, 4 e 6. Isto mostra que o tipo de cruzamento não afeta o custo computacional do método, mas o número de lotes alterou significativamente o tempo.

Por outro lado, confrontando-se AGs com apenas diferenças na quantidade de indivíduos na população, percebe-se uma variação muito grande em termos

de tempo computacional, como exemplo o tempo do AG-6 em relação ao AG-5 foi quase três vezes maior.

Comparando-se os AGs 8, 9 e 10 com diferença apenas no tipo de seleção, percebeu-se que as seleções por torneio e elitista apresentaram tempos computacionais similares, enquanto a seleção por roleta viciada consumiu três segundos a mais em relação às demais.

Na busca de solução para o PRC-A, cujo modelo é proposto na seção 2.7.1, foram investigadas quais das metodologias utilizadas encontraram mais soluções admissíveis e que forneceu a melhor solução para o problema. .

A Tabela (7) apresenta o número de vezes que o AG forneceu solução factível ao final da execução, lembrando-se que foram feitas 100 simulações para cada AG.

Tabela 7 - Número de soluções factíveis encontradas pelo AG

AG	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Número de Lotes										
10	52	100	100	92	86	98	88	99	100	100
15	6	40	20	22	32	92	32	89	78	92
20	1	4	1	22	8	72	6	73	32	62

Observando a Tabela (7) nota-se que para $L = 10$, encontra-se mais soluções admissíveis em comparação com os resultados obtidos para $L = 15$ e 20. Para $L = 15$, observa-se que os AGs 6, 8, 9 e 10 com $P = 241$ apresentaram bons resultados na busca de soluções factíveis, enquanto os demais apresentaram pouca eficiência nesta busca, menos de 40%. Para $L = 20$, os AGs 6, 8 e 10 apresentaram uma percentagem variando de 62% a 72% de vezes que encontraram soluções factíveis. Os demais apresentaram no máximo 32% de vezes.

Comparando-se os AGs 6 e 8 (e também os AGs 5 e 7), observou-se que o número de gerações a partir de 100 pouco favoreceu a melhoria na busca de soluções factíveis.

Pode-se notar que o AG-8, que usa o método da roleta com $P = 241$, $G = 150$ e *crossover* uniforme, apresentou bons resultados na busca de soluções factíveis, visto que com o aumento da quantidade de lotes, a percentagem de soluções factíveis encontradas foi caiu lentamente com o aumento do valor de L, o que não aconteceu com os demais AGs.

A Tabela (8) mostra os valores médios encontrados para os lucros das rotações encontradas. Com o intuito de obter uma solução factível para o PRC-A com o maior lucro possível, investigou-se a qualidade das programações factíveis fornecidas pelos métodos, medindo a diferença relativa (Δ) entre a lucro médio obtido pela solução final e o obtido pela primeira solução admissível encontrada. Em que esta diferença relativa (em percentagem) foi calculada pela fórmula:

$$\Delta = 100\frac{((lucro\ final) - (lucro\ inicial))}{(lucro\ final)}.$$

Tais valores estão na Tabela (9).

Tabela 8 - Lucratividade média (R$)

AG	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Número de Lotes										
10	43.255	54.540	51.413	51.263	49.137	55.022	54.780	55.296	52.092	49.842
15	38.875	58.397	49.019	53.142	58.928	74.910	58.532	76.590	71.185	74.411
20	19.901	41.809	29.146	53.142	55.111	90.590	54.750	93.834	75.171	90.905

Tabela 9 - Valores médios de Δ obtidos pelo AG

AG	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Número de Lotes										
10	34,18	45,08	45,45	36,45	34,74	48,11	42,71	47,22	41,21	54,02
15	29,36	32,74	35,34	36,40	31,60	43,72	28,67	44,74	36,39	42,37
20	22,50	25,12	21,70	36,40	62,43	38,46	28,60	38,53	33,46	39,90

Notou-se que o AG com seleção por roleta, *crossover* uniforme, $P = 241$ e $G = 150$ (AG-8) foi o que proporcionou o maior lucro dentre as dez abordagens

simuladas, nas três instâncias. Isto revela a eficiência destes operadores quando atuam juntos.

Quanto à Tabela (9), para $L = 10$ nota-se uma maior diferença entre os lucros obtidos pelas rotações factíveis final e inicial, em comparação com as demais instâncias, isto pode ser devido ao aumento do número de lotes, que proporciona um considerável aumento no número de restrições do problema.

Os AGs 6 e 8 difenciam-se pelo número de gerações, respectivamente de 100 e 150. Evidencia-se que a variação no lucro, para as três instâncias simuladas foi despresível, demostrando uma exploração pouco eficiente após a centésima geração.

Para $L = 20$, o tempo computacional do AG-8 foi cerca de 60% maior em relação ao AG-6 e obteve praticamente a mesma variação no lucro a partir da primeira até a solução final. Mas, por outro lado o ganho no lucro do AG-8 comparando com o AG-6 foi de apenas 1,4% e a convergência para soluções factíveis foi de apenas 3,58% melhor que o AG-6. Desta forma, foi mais conveniente utilizar o AG-6 do que o AG-8 pois o benefício do AG-8 foi muito pequeno comparado com o seu alto custo computacional.

4.1.2 Resultados obtidos pelo SA

Nesta seção, na tentativa de resolução do PRC-A, foram abordados os seguintes SAs, para $L = 10$, 15 e 20:

1) SA_$\{1; 10^{-4}; 0,95\}$;

2) SA_$\{1; 10^{-4}; 0,99\}$;

3) SA_$\{10; 10^{-6}; 0,95\}$;

4) SA_$\{10; 10^{-6}; 0,99\}$;

5) SA_$\{100; 10^{-6}; 0,95\}$;

6) SA_$\{100; 10^{-6}; 0,99\}$;

7) SA_$\{100; 10^{-8}; 0,95\}$;

8) SA_$\{100; 10^{-8}; 0,99\}$.

A Tabela (10) ilustra os tempos computacionais para as oito configurações do SA simuladas. Nota-se que houve um aumento de aproximadamente 14% no tempo computacional, quando a temperatura inicial T_0 passa de 10 para 100, nas três instâncias.

Tabela 10 - Tempos computacionais médios (s) para cada instância utilizando os 8 SAs

SA	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Número de Lotes								
10	4,11	21,25	6,80	36,84	7,75	42,52	9,65	49,34
15	8,25	42,17	14,57	74,44	16,31	84,31	20,35	104,26
20	14,98	75,35	24,86	126,80	28,15	143,90	35,38	181,16

A temperatura final T_f do SA-7 foi 100 vezes menor que a do SA-5, entretanto, isto refletiu em um aumento no tempo de processamento do SA-7 de aproximadamente 25% em relação ao SA-5, nas três dimensões.

De acordo com as experimentações realizadas nestas simulações, o parâmetro que causou o maior impacto no tempo de processamento do SA foi a velocidade de resfriamento β. Isto pode ser visto na Tabela (10) onde o tempo do SA-2 com $\beta = 0,99$ foi cerca de cinco vezes maior que o SA-1 com $\beta = 0,95$, para $L = 10$, 15 e 20. De maneira similar, esta conclusão pode ser feita comparando SA-5 com SA-6.

A Tabela (11) mostra o número de soluções factíveis encontradas para cada instância, nas 100 simulações. A Tabela (12) mostra o lucro obtido pelas soluções dos SAs. Conclui-se que os métodos SAs utilizando $L = 10$ e $L = 15$ acharam soluções factíveis em no mínimo 92% das vezes que o método foi executado.

Para $L = 20$ e $\beta = 0,99$ o SA conseguiu encontrar programações factíveis para as culturas, em pelo menos, 96% das simulações, ilustrando a importância de utilizar uma taxa de arrefecimento mais próxima possível de 1 (o que

Tabela 11 - Número de soluções factíveis encontradas pelo SA

SA	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Número de Lotes								
10	98	100	100	100	100	100	100	100
15	90	100	92	100	98	100	98	100
20	48	96	94	100	92	100	94	100

Tabela 12 - Lucratividade média (R$)

SA	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Número de Lotes								
10	54.264	64.144	59.147	66.428	58.602	68.624	59.840	68.304
15	76.555	94.830	84.170	99.808	86.428	100.590	88.444	101.200
20	86.083	117.010	105.220	125.010	105.660	126.420	109.140	128.890

equivale reduzir a temperatura de maneira mais lenta). Estes SAs forneceram, nas três instâncias, as rotações com os maiores lucros.

A temperatura inicial foi um parâmetro de pouco efeito neste algoritmo. Usando o SA com $T_0 = 10$ e $T_0 = 100$, o lucro e o número de encontro de soluções admissíveis obtidos desenharam valores bastante próximos.

Objetivando testar a interferência do parâmetro T_f, foi simulado o SA com $T_f = 10^{-6}$ e $T_f = 10^{-8}$ e os resultados ilustraram que o número de vezes que foi encontrado soluções admissíveis foi praticamente o mesmo, com um ligeiro acréscimo no lucro, aproximadamente 3% maior quando utilizado $T_f = 10^{-8}$.

A Tabela (13) mostra os valores de Δ para todas as instâncias. Nota-se que problemas com dimensões menores apresentam diferenças relativas maiores, tendo em vista o menor número de restrições a ser respeitado e a menor quantidade de soluções a serem investigadas.

Com relação ao β, os SAs 2, 4, 6 e 8 em comparação com 1, 3, 5 e 7, respectivamente, conseguiram Δ maior, devido a taxa de arrefecimento ser mais próxima de 1. Isto demonstrou uma maior capacidade do algoritmo em encontrar

Tabela 13 - Valores médios de Δ obtidos pelo SA

SA	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Número de Lotes								
10	44,41	72,57	58,57	83,70	60,65	91,03	66,72	95,82
15	42,88	63,66	44,89	71,89	61,52	89,42	58,47	96,60
20	38,24	53,34	42,92	60,54	41,01	85,85	47,99	91,48

soluções factíveis de maior lucro, mostrando que o valor β foi o que mais interviu no desempenho computacional e no custo das soluções deste algoritmo.

De maneira geral comparando-se os SAs 5 e 8 pode-se inferir que embora o tempo computacional do SA-8 seja 5 vezes maior do que no SA-5, o SA-8 apresenta um lucro médio 22% maior que o SA-5 e encontrou soluções factíveis em todas as simulações. Isto demonstrou, do ponto de vista de custo/benefício, uma maior eficiência do algoritmo SA-5.

4.1.3 Resultados obtidos pelo AG+SA

Na tentativa de melhorar os desempenhos dos métodos AG e SA para resolução do PRC-A, abordou-se nesta seção o algoritmo híbrido AG+SA, descrito na seção 2.6.1. Conforme as simulações foram feitas em três instâncias com diferentes parâmetros conforme listados a seguir:

1) AG_{50;61}+SA_$\{1; 10^{-4}; 0,90\}$;

2) AG_{50;61}+SA_$\{10; 10^{-6}; 0,95\}$;

3) AG_{80;91}+SA_$\{1; 10^{-4}; 0,90\}$;

4) AG_{50;91}+SA_$\{1; 10^{-4}; 0,90\}$;

5) AG_{50;91}+SA_$\{1; 10^{-4}; 0,95\}$;

6) AG_{80;91}+SA_$\{10; 10^{-6}; 0,95\}$.

As Tabelas (14), (15) e (16) ilustram, nesta ordem, o tempo computacional médio por cada simulação, o número de soluções admissíveis encontradas pelo método e o lucro médio obtido com as 100 simulações.

Tabela 14 - Tempos computacionais médios (s) para cada instância utilizando os 6 AG+SAs

AG+SA	**1**	**2**	**3**	**4**	**5**	**6**
Número de Lotes						
10	4,00	8,05	7,07	5,36	6,38	13,67
15	7,72	14,05	11,16	9,89	10,65	24,45
20	12,92	27,06	22,75	16,09	13,67	34,01

Tabela 15 - Número de soluções factíveis encontradas pelo AG+SA

AG+SA	**1**	**2**	**3**	**4**	**5**	**6**
Número de Lotes						
10	82	100	96	98	96	98
15	80	100	88	84	94	98
20	42	96	56	46	98	97

Tabela 16 - Lucratividade média (R$)

AG+SA	**1**	**2**	**3**	**4**	**5**	**6**
Número de Lotes						
10	56.698	62.511	58.712	59.330	66.985	66.414
15	81.073	66.161	85.868	84.347	91.796	97.487
20	92.767	122.710	98.118	94.709	136.708	126.100

No híbrido AG+SA-2, concentrou-se a maior parte do tempo de sua execução no SA, enquanto na abordagem AG+SA-3 o maior tempo ocorreu no AG. Os resultados, para as três instâncias, apresentados na Tabela (14), mostraram que o tempo de processamento AG+SA-2 foi menor do que na abordagem AG+SA-3 (em média 22%). Um outro fato observado foi que a convergência para soluções admissíveis do híbrido AG+SA-2, não foi afetada pelo aumento do número de lotes utilizados. Esta característica não foi contemplada no AG+SA-3.

Desta forma, para melhorar o desempenho do híbrido AG+SA, foi necessário fornecer uma solução inicial de boa qualidade para o SA, esta solução inicial foi fornecida pelo AG com maior número de gerações e indivíduos (denominado AG_forte). Este procedimento foi mais vantajoso do que efetuar uma busca mais refinada pelo SA partindo de uma solução inicial fornecida por um AG com menor número de gerações e indivíduos (denominado AG_fraco).

A Tabela (17) mostra os valores de Δ para as três instâncias. O AG+SA-3 confrontado com o AG+SA-4, mostra que o número de gerações no AG não favoreceu uma melhoria na busca de melhores soluções para o problema pois as diferenças nos valores de Δ foram pequenas.

Tabela 17 - Valores médios de Δ obtidos pelo AG+SA

AG+SA	**1**	**2**	**3**	**4**	**5**	**6**
Número de Lotes						
10	36,64	62,22	47,28	49,55	61,54	57,46
15	27,48	60,13	35,77	32,38	43,95	55,65
20	11,55	40,82	15,90	13,08	57,46	37,98

Nota-se que o valor de Δ decresce à medida que o número de lotes aumenta e que os maiores valores para a taxa de resfrimento β foi fundamental para conseguir os maiores Δs.

Os AG+SA-5 e AG+SA-6 demonstraram grande eficiência, pois as rotações obtidas por eles foram as de maior lucro e num elevado número de vezes encontraram soluções admissíveis, nas três instâncias. No entanto, o AG+SA-6, que utiliza $G = 80$ gerações, apresentou um tempo computacional cerca 2,5 vezes maior que o AG+SA-5 ($G = 50$ gerações) e forneceu, pelas rotações, um lucro 8% menor em relação ao AG+SA-5. Evidentemente a melhor performance foi conseguida pelo AG+SA-5.

4.1.4 Resultados obtidos pelo Memético

Nesta seção foi abordado o algoritmo Memético para resolução do PRC-A, conforme descrito na seção 2.6.2. As simulações foram feitas em três instâncias com diferentes parâmetros conforme listados a seguir:

1) AG_{50;61}+BL_{2};

2) AG_{50;91}+BL_{2};

3) AG_{50;61}+BL_{5};

4) AG_{50;61}+BL_{10};

5) AG_{50;91}+BL_{10};

6) AG_{80;91}+BL_{2};

7) AG_{80;61}+BL_{5};

8) AG_{80;91}+BL_{10}.

Apresenta-se nas Tabelas (18), (19) e (20) a seguir os tempos computacionais médios por simulação, o número de soluções admissíveis obtidas e os lucros médios das rotações encontradas.

Tabela 18 - Tempos computacionais médios (s) para cada instância utilizando os 8 Meméticos

Memético	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Número de Lotes								
10	2,12	2,70	2,02	3,98	4,13	4,44	5,96	5,90
15	2,97	4,73	4,09	10,22	9,71	7,47	10,81	12,66
20	4,65	7,12	7,49	20,76	22,62	11,11	14,18	31,31

Foram utilizados os algoritmos Memético-1 com $P = 91$ e Memético-2 com $P = 61$ para ilustrar a influência dos tamanhos populacionais do AG nos resultados finais. O tempo de processamento foi de aproximadamente 60% maior

Tabela 19 - Número de soluções factíveis encontradas pelo Memético

Memético	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Número de Lotes								
10	60	78	78	90	96	86	84	98
15	32	56	80	92	96	48	82	96
20	22	28	64	95	98	34	78	98

Tabela 20 - Lucratividade média (R$)

Memético	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Número de Lotes								
10	45.363	49.341	49.493	54.508	55.369	50.876	52.489	56.987
15	62.752	67.527	76.607	84.851	86.916	65.429	80.000	86.625
20	75.044	79.390	100.310	116.710	119.140	82.635	108.370	119.670

no Memético-2 quando comparado com Memético-1; por outro lado o lucro ficou em média apenas 8% maior Memético-2 para todas as três instâncias.

Os Meméticos 3 e 2, tiveram performances computacionais parecidas. O Memético-3 utiliza 91 indivíduos em seu AG e um tamanho de vizinhança de 2^2L^2 na BL, o Memético-2 tem uma população de 61 indivíduos e a cardinalidade da vizinhança da BL igual a 5^2L^2. O Memético-3 nas três instâncias encontrou soluções com maior lucro, em que para $L = 20$ a diferença foi de 27% e obteve 64% das soluções factíveis, ao passo que Memético-2 obteve 28%.

Para investigar a influência do tamanho da vizinhança na BL comparou-se os Meméticos 6 ($a = 2$) e 8 ($a = 10$). O híbrido que incorpora mais vizinhos na BL, Memético-8, apresenta um lucro maior comparado com um de menos vizinhos Memético-6, nas três instâncias. Para a situação apresentada, ficou evidente que o melhor desempenho, em termos de lucro, foi obtido pelo Memético-8. A convergência para soluções admissíveis foi outro fator que ressalta a melhor eficácia da metodologia 8, pois com o aumento do número de lotes, o número de soluções admissíveis ficou praticamente estabilizado em 1, característica não contemplada no método 6. Desta forma, a eficiência do híbrido apresenta indícios de dependencia do

parâmetro a.

As abordagens Memético-4 e Memético-5 usam $a = 10$ e mudam apenas em relação ao tamanho da população, 61 e 91 respectivamente. Mas os resultados quanto ao tempo, número de soluções factíveis encontradas e lucro demonstraram que há pouca influência do parâmetro P neste híbrido visando a busca de soluções factíveis de boa qualidade.

Conclusões semelhantes podem ser tiradas para o número de gerações utilizado no AG. Os métodos Memético-5 (com $G = 50$) e Memético-8 (com $G = 80$) apresentaram performance muito parecida, embora o tempo computacional para 80 gerações seja 55% maior no Memético-8.

A seguir, a Tabela (21) mostra os valores de Δ para todas as instâncias referidas. Os resultados mostraram uma queda no valor de Δ quando o número lotes aumentou. Isto indica que a BL reduz a sua eficiência quando a instância do problema aumenta, pois a quantidade de vizinhos gerados foi proporcional à L^2.

Tabela 21 - Valores médios de Δ obtidos pelo Memético

Memético	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Número de Lotes								
10	37,48	38,57	39,60	41,92	45,24	36,96	42,65	50,28
15	27,87	40,32	31,76	40,24	43,47	31,93	38,98	42,71
20	34,54	34,20	33,31	39,86	45,82	29,38	38,20	43,22

Baseado nas conclusões anteriores, o híbrido Memético com melhor custo/benefício pode ser atribuído àquele que tem 50 gerações, 91 indivíduos e $100L^2$ vizinhos pois consumiu um menor tempo computacional em relação ao Memético-8, e não diferenciam-se em termo de lucro das rotações e de Δ.

4.1.5 Comparação dos Métodos

Os algoritmos com os melhores custos/benefícios obtidos nas quatro seções anteriores estão listados abaixo:

- AG_{R}_{U}_{100; 241};
- SA_{100; 10^{-8}; 0, 99};
- AG_{50;91}+SA_{1; 10^{-4}; 0, 95};
- AG_{50;91}+BL_{10}.

Estas quatro metodologias de otimização, utilizadas para resolver o PRC-A, apresentaram distintas performances, do ponto de vista do tempo computacional, na obtenção de soluções admissíveis, na qualidade das rotações de plantio e na exploração do espaço de busca. Para afirmar quais algoritmos obtiveram um melhor custo/benefício neste problema, analisou-se os métodos separadamente em cada instância, utilizando como critérios de avaliação, nesta ordem: (a) a quantidade de soluções admissíveis encontradas (α), (b) lucro das rotações, (c) os valores Δ e (d) o tempo de processamento. Como os tempos computacionais consumidos pelos algoritmos, em todas as instâncias, foram considerados baixos (no máximo 3 minutos por simulação), deu-se um menor enfoque à este quesito. Assim, a comparação destas metaheurísticas baseou-se em seu benefício promovido, no sentido da qualidade das soluções encontradas.

A Tabela (22) mostra o desempenho dos algoritmos na instância $L = 10$. Notou-se que nesta dimensão o SA foi o que forneceu a solução com maior lucro, em todas as simulações forneceu soluções factíveis e o que mais explorou o espaço de busca pelo valor de Δ mais elevado. Embora o AG+SA forneceu 4% de soluções infactíveis (2% a mais que o AG), o lucro das rotações e o valor de Δ foi, nesta ordem, 22% e 28% superior em relação ao AG. Assim, atribui-se ao AG+SA o segundo melhor desempenho para $L = 10$.

Comparando-se o AG com o Memético, considera-se muito pequena a diferença entre os lucros das rotações por eles fornecidas (apenas 0,6%) e o valor de Δ (6%). Embora o Memético seja o mais rápido, do ponto de vista computacional, o AG em apenas 2% das simulações forneceu soluções infactíveis enquanto no Memético

Tabela 22 - Resultados computacionais médios dos quatro melhores algoritmos para 10 lotes

Método	**Tempo (s)**	α	**Lucro**	Δ
AG	19,5	98	55.022	48,11
SA	49,34	100	68.304	95,82
AG+SA	6,38	96	66.985	61,54
Memético	4,13	96	55.369	45,24

este índice foi o dobro. Sendo assim, nesta instância o Memético possuiu o menor benefício dentre os quatro analisados.

A Tabela (23) mostra os resultados para das quatro metaheurísticas para $L = 15$. O único método capaz de garantir soluções factíveis nas 100 simulações novamente foi o SA. Adicionalmente, o lucro das programações por ele fornecidas e Δ foram os maiores valores, desempenhando o melhor benefício nesta instância. Percebeu-se uma similaridade de desempenhos entre o AG+SA e Memético; embora o lucro das rotações do AG+SA foi 5,6% maior, comparado com o Memético, caracteriza o híbrido com uma BL o segundo melhor benefício, tendo em vista o seu número de soluções factíveis fornecido.

Tabela 23 - Resultados computacionais médios dos quatro melhores algoritmos para 15 lotes

Método	**Tempo (s)**	α	**Lucro**	Δ
AG	36,2	92	74.910	43,72
SA	104,26	100	101.200	96,60
AG+SA	10,65	94	91.796	43,95
Memético	9,71	96	86.916	43,47

O AG teve o desempenho mais pobre, pois a aptidão de sua solução ficou abaixo da média e em 8% das vezes encontrou rotações infactíveis.

A próxima Tabela (24) ilustra os resultados das quatro melhores metaheurísticas para a instância $L = 20$.

Como nas duas instâncias anteriores, o SA foi o que proporcionou o melhor benefício (embora o lucro das soluções fora 8% menor em relação ao AG+SA e o seu tempo de processamento o mais elevado), tendo em vista a sua alta capa-

Tabela 24 - Resultados computacionais médios dos quatro melhores algoritmos para 20 lotes

Método	**Tempo (s)**	α	**Lucro**	Δ
AG	56,3	72	90.590	38,46
SA	181,16	100	128.890	91,48
AG+SA	13,67	98	136.708	57,46
Memético	22,62	98	119.140	45,82

cidade de encontrar soluções factíveis e no alto índice Δ fornecido, demonstrando, mesmo para problemas de elevado número de restrições e variáveis, um alto poder de busca. Os híbridos também proporcionaram altos índices de obtenção de soluções admissíveis (98%). O AG+SA forneceu a rotação com o maior lucro e o segundo maior valor de Δ, incumbindo elegê-lo como a segundo melhor algoritmo nesta instância.

Notou-se que o AG à medida que a instância foi aumentada, foi gradativamente perdendo a sua eficácia em relação aos outros. Nesta instância apresentou o menor número de soluções admissíveis, a menor lucratividade das rotações e o menor valor para Δ.

De forma geral, para as três instâncias investigadas, notou-se um ótimo desempenho do algoritmo SA e uma boa performance das abordagens híbridas em problemas combinatorias e de elevado número de restrições. Ao AG, pode-se afimar que a sua eficiência foi seriamente comprometida à medida que o número de restrições aumentou. O principal motivo para esta queda de rendimento foi o seu sistema de cruzamento, onde filhos infactíveis são obtidos de pais factíveis. Este aspecto peculiar não foi evidencidado nos algoritmos que usam em sua busca, a noção de vizinhança, tendo em vista que o vizinho de uma solução factível tenha uma grande chance de ser factível. Isto justifica a boa eficiência dos métodos SA, AG+SA e Memético.

As melhores programações de plantio para o PRC-A, nas três dimensões, encontram-se no Apêndice D.

4.2 Resultados para o PRC-A-D

Baseada nas experiências das simulações apresentadas na seção anterior, como a influência de cada parâmetro nestes algoritmos, foram propostos os quatro métodos e seus respectivos parâmetros para solucionar o PRC-A-D:

- AG_{R}_{U}_{120;601};
- SA_{$100; 10^{-8}; 0,99$};
- AG_{100;301}+SA_{$100; 10^{-5}; 0,95$};
- AG_{100;301}+BL_{32}.

Como se observa, o AG sugerido apresentou um elevado número de indivíduos com comparação com a abordagem da seção anterior; o SA proposto com índice de resfriamento muito próximo de 1; e as abordagens híbridas contendo grande quantidade de indivíduos na parte do AG. Estas considerações e propostas são para fornecer soluções da melhor qualidade possível nesta aplicação.

Os resultados computacionais destes quatro métodos como os tempos t de execução médios (em segundos), o número de vezes que os algoritmos atingiram soluções admissíveis (α), o lucro médio obtido pelas rotações finais obtidas ($\overline{z}$), o coeficiente de variação dos lucros obtidos[7] (cv) e Δ encontram-se na Tabela (25) a seguir. O lucro médio exibido refere-se às rotações admissíveis encontradas.

Nos métodos híbridos, ilustra-se t_{AG}, t_{SA} e t_{BL} como sendo os tempos (em segundos) utilizados pelo AG, SA e BL, isto é, os tempos das componentes isoladas dos híbridos. Semelhantemente, $\overline{z_{AG}}$ é a média das dos lucros das soluções dos AGs dos métodos híbridos.

Os métodos híbridos foram as metodologias que demonstraram as melhores eficiências nesta aplicação, pois forneceram as programações de plantio com os maiores lucros, em média, e os maiores índices de obtenção de soluções admissíveis α, ambas iguais à 99%. Mais precisamente, o AG+SA apresentou um desempenho

[7] O coeficiente de variação cv é definido como a razão entre o desvio padrão σ e a média μ.

Tabela 25 - Resultados computacionais médios dos quatro algoritmos para o PRC-A-D

Método	t_{AG} (s)	t_{SA} (s)	t_{BL} (s)	t (s)	$\overline{z_{AG}}$ (R$)	$\overline{z}$ (R$)	α	cv	Δ (R$)
AG	1.017,1	-	-	1.017,1	-	1.487.720,00	72	0,11	11,2
SA	-	158,4	-	158,4	-	1.896.105,00	98	0,07	24,2
AG+SA	169,5	20,1	-	189,6	1.450.210,00	2.204.890,00	99	0,08	25,3
Memético	172,0	-	426,2	568,2	1.487.200,00	1.987.600,00	99	0,07	21,2

superior em relação ao Memético, pois o lucro médio fornecido foi quase 11% maior e consumiu a terça parte do tempo computacional.

O tempo computacional do AG+SA foi de aproximadamente 190 segundos, sendo cerca de 170 segundos para o AG e 20 segundos para o SA. A aptidão da solução do AG deste método foi R$ 1.450.210,00, em média. Assim, o SA deste algoritmo aumentou em 52% o lucro desta solução em menos de 11% do tempo computacional total do AG+SA.

A mesma análise pode ser feita no Memético: o AG forneceu uma solução cuja aptidão foi de R$ 1.487.200,00 em 172 segundos, ao passo que a BL demorou 426,2 segundos (75% do tempo total do método), cujo lucro da solução final foi 33,4% maior. Desta a forma, o SA foi uma estratégia de melhora mais eficiente que a BL para o AG.

O SA nesta aplicação desempenhou ótimos resultados, tendo em vista a qualidade das soluções encontradas, a quantidade de soluções admissíveis fornecidas, o aumento Δ promovido nelas e principalmente ao reduzido (o menor) tempo computacional utilizado. Por outro lado, o AG possuiu uma performance razoável, pois obteve em 72% das simulações soluções factíveis e usou o maior tempo computacional dentre as quatro métodologias.

Embora estes tempos de execução por simulação foram considerados baixos (no máximo 17 minutos) foi notável a discrepância do tempo usado pelo AG, devido principalmente, ao elevado número de indivíduos utilizado em sua população (601 indivíduos). Caso fosse utilizado uma população menor, o valor de α seria inferior pela experiência adquirida nas simulações.

A Tabela (26) a seguir ilustra os maiores valores de lucros (z_{max}) encontrados pelos algoritmos nas 100 simulações, a iteração média $\overline{g}$ em que a solução admissível foi encontrada, os valores médios das penalizações das soluções iniciais de vizinhança, adubação verde, pousio e demanda, denotadas respectivamente por $\overline{pv_i}$, $\overline{pav_i}$, $\overline{pp_i}$ e $\overline{pd_i}$. Semelhantemente observou-se os valores médios das penalizações de vizinhança, adubação verde, pousio e demanda das soluções finais denotadas, nesta ordem, por $\overline{pv_f}$, $\overline{pav_f}$, $\overline{pp_f}$ e $\overline{pd_f}$.

Tabela 26 - Média das penalizações da solução inicial e final, lucro da melhor solução encontrada e iteração na qual uma solução admissível foi encontrada

Método	z_{max} (R$)	$\overline{g}$	$\overline{pv_i}$	$\overline{pav_i}$	$\overline{pp_i}$	$\overline{pd_i}$	Σ	$\overline{pv_f}$	$\overline{pav_f}$	$\overline{pp_f}$	$\overline{pd_f}$	Σ
AG	1.916.700,00	107	33,2	2,8	2,8	5,3	44,1	0,08	0	0	0,2	0,28
SA	2.080.100,00	504	39,3	4,2	3,8	4,7	52,0	0,01	0	0	0,01	0,02
AG+SA	2.550.930,00	151	43,5	4,5	4	5	57,0	0	0	0	0,01	0,01
Memético	2.245.300,00	428	32,4	2,6	3,4	5,9	44,3	0	0	0	0,01	0,01

Observando a Tabela (26) pode-se notar que o AG encontrou soluções admissíveis apenas em média na geração 107, devido às elevadas taxas de mutação, migração e após a migração em massa, pois nesta geração houve uma alta probabilidade de inserção de novos cromossomos e estruturas, o que promoveu uma busca mais global e eficaz.

Como a temperatura no SA abaixa baseada numa sequência geométria[8] com razão β, de termo inicial T_0 e final T_f, então o número n_1 de iterações até o congelamento é

$$n_1 = \frac{\ln(T_f/T_0)}{\ln(\beta)} + 1$$

e pelos parâmetros utilizados, $n_1 = 2.292$ iterações. Pode-se afirmar que a solução admissível foi precocemente encontrada (em média, na iteração 504 $\overline{g} = 504$), ou seja, em 22% de n_1. Assim, o método passou 1.788 iterações refinando as soluções admissíveis encontradas e isto refletiu no alto índice Δ encontrado (24,2%).

[8]Em uma Progressão Geométrica, o termo geral a_n pode ser escrito em função do termo inicial a_1, do número de termos n e da razão q como: $a_n = a_1 q^{n-1}$.

No caso dos métodos AG+SA e Memético, possuíram respectivamente n_2 e n_3 iterações em que

$$n_2 = G + \frac{\ln(T_f/T_0)}{\ln(\beta)} + 1$$

e

$$n_3 = G + aL$$

e pelos parâmetros utilizados, $n_2 = 415$ e $n_3 = 612$ iterações. Assim, pode-se afirmar que estes métodos conseguiram obter as soluções admissíveis após a busca do AG ter sido realizada. No AG+SA, o SA necessitou de apenas 16% de suas iterações para encontrá-las enquanto na BL, 64%. Estes números afirmaram a melhor capacidade do método *annealing* em pesquisar novas e melhores soluções para o problema, comparado com o método clássico de descida, utilizado pelo Memético. Assim, verificou-se uma maior "rapidez" para encontrar soluções admissíveis no SA do AG+SA comparado ao SA puro. Este algoritmo, teve uma melhora em seu desempenho ao iniciá-lo com uma solução de "boa" qualidade.

Quanto ao número de penalizações, pode-se afirmar que as soluções iniciais tiveram um elevado número de penalizações, principalmente as de vizinhança. De maneira geral todos os algoritmos desenvolvidos conseguiram eliminá-las quase completamente. A quantidade de penalizações finais mostrou esta eficácia: no AG, por exemplo, das 28 soluções inadimissíveis encontradas, 20 delas possuíram apenas uma penalização de vizinhança e 8 soluções uma penalização de demanda. Deste modo, as soluções do AG inadmissíveis foram muito próximas de serem admissíveis.

O mesmo comportamento aconteceu com os demais métodos, mas em menor quantidade: no SA apenas duas soluções foram infactíveis onde uma teve uma penalização de vizinhança e a outra de demanda. Da mesma forma no AG+SA e Memético: uma só solução infactível com apenas uma penalização de demanda.

Com o intuito de investigar o comportamento das soluções destas metaheurísticas sob um mesmo número de iterações, fez-se uma breve simulação, uma única vez, destas abordagens. Deste modo simulou-se um AG_{R}_{U}_{500;601}, SA_{$100; 10^{-5}$;0,9682},

AG_{100;301}+SA_{100; 10^{-5}; 0, 95} e o AG_{100;301}+BL_{25}. Como se observa, todos estes algoritmos possuiram 500 iterações. Nesta pequena simulação, foram omitidos os tempos computacionais utilizados pois priorizou-se o comportamento geométrico das soluções.

A Figura (21) à esquerda ilustra o comportamento geométrico da solução do AG e SA e à direita dos híbridos, num mesmo plano coordenado. Nota-se o AG manteve sua evolução, concorrente com o SA até iteração 120 aproximadamente. Após esta fase de alto crescimento, o AG estaciona-se e o SA continua sua evolução.

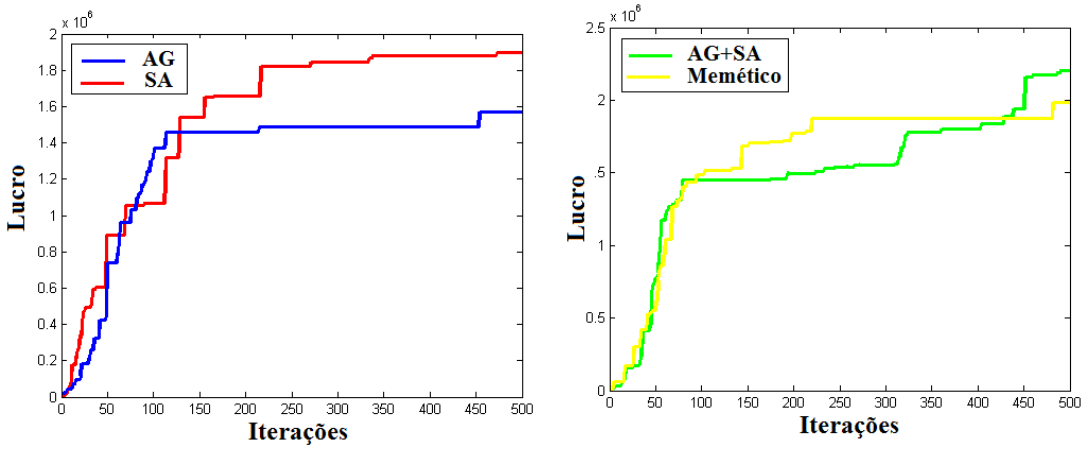

Figura 21 - Evolução da solução no AG e SA (à esq.) e no AG+SA e Memético (à dir.)

De forma semelhante, o híbrido Memético estacionou a sua busca praticamente na iteração 220, ficando preso num ótimo local por quase 250 iterações. No entanto sua convergência foi mais tardia em relação ao AG puro. Em relação ao AG+SA, pode-se afirmar que sua busca por novas programações de plantio permaneceu ativa até completar o seu ciclo.

A Figura (22) ilustra a evolução dos métodos híbridos. À esquerda mostra-se AG+SA e à direita o Memético. Em destaque, aponta-se o momento em que o AG encerra sua busca (na iteração 100) e os novos algoritmos começam a

efetuar sua pesquisa. É notável que ambos tem a sua solução melhorada após a centésima geração com a mudança do método.

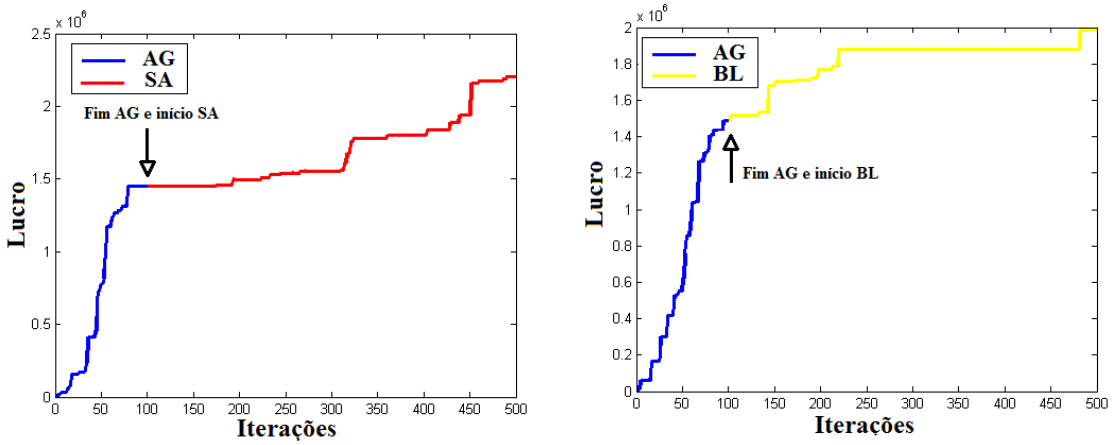

Figura 22 - Evolução da solução no AG+SA (à esq.) e no Memético (à dir.)

A Figura (23) coloca num mesmo plano a evolução de um AG_{R}_{U}_{500;301} (que é o AG dos híbridos, apenas diferente no parâmetro G) e os métodos híbridos da Figura (22).

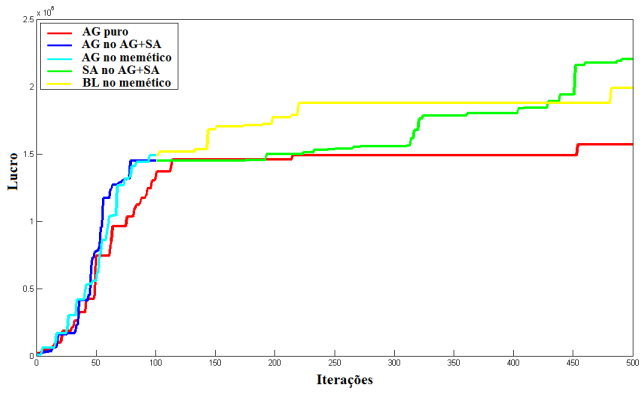

Figura 23 - Comparação e a evolução das soluções de um AG_{500,301}, o AG+SA e Memético com 500 iterações

Assim, nota-se que este AG com 500 iterações e 301 indivíduos convergiu prematuramente na geração 100, estacionu-se num ótimo local por cerca de 350 gerações. Isto se deve principalmente ao reduzido número populacional utilizado por este AG.

Nos métodos híbridos, o SA e a BL cumpriram a intenção proposta: impedir a convergência prematura, explorando novas soluções e economizando em seu tempo de execução.

Finalmente, a Figura (24) exibe concomitantemente a evolução das soluções das quatro metaheurísticas propostas. Nesta Figura, o AG possui 601 indivíduos.

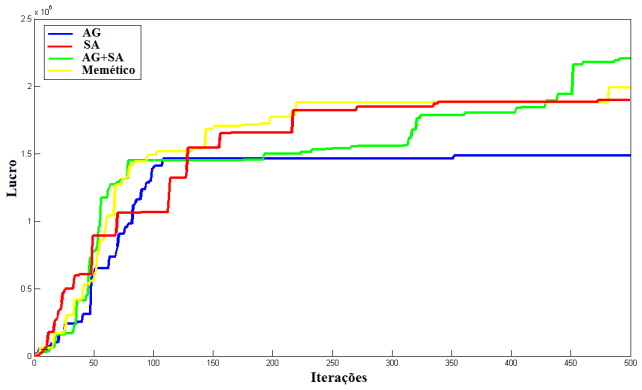

Figura 24 - Comparação e a evolução das soluções dos quatro métodos, todos com 500 iterações

Observa-se que antes da centésima iteração todos têm uma evolução similar, diferenciando-se após esta fase. Adicionalmente, notou-se uma ótima performance do SA puro, equiparando-se ao Memético, demostrando uma alta capacidade na pesquisa de soluções e na não convergência prematura.

Nesta simulação com todos os métodos possuindo 500 iterações, o AG+SA obteve a solução com o maior lucro, seguido do Memético, SA e AG, confirmando os resultados exibidos na Tabela (26).

As Figuras (25), (26), (27) e (28) mostram as melhores programações de plantio obtidas nas 100 simulações pelo AG, SA, AG+SA e Memético para o

PRC-A-D e resultaram em R$ 1.916.700,00, R$ 2.080.100,00, R$ 2.550.930,00 e R$2.245.300,00 de lucro, nesta ordem.

Uma característica observada nestas rotações é o período de pousio concentrado no mês de julho. Tal observação está relacionado ao conjunto de dados utilizados, mais precisamente ao período de plantio, famílias botânicas e duração do ciclo de vida de cada cultura considerada (Tabela (27) do apêndice). Do ponto de vista prático é plausível esta alocação, tendo em vista que o mês de julho ser um mês de prolongadas estiagens e de mudanças climáticas no Brasil. Como há poucas culturas que podem ser inseridas ou plantadas neste mês, os algoritmos concentram o pousio nesta época (pois pode ser "cultivado" durante todo o ano) e alocam nos demais meses as culturas específicas de verão e de inverno.

Os números nos mosaicos a seguir, estão associados na Tabela (27), e indicam a culturas a serem plantadas nos 16 lotes (horizontal) e nos 24 períodos (vertical).

	Ano 1												Ano 2											
	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1	28	28	7	7	1	1	30	25	25	25	26	26	26	2	2	2	2	30	21	21	21	21	28	28
2	20	1	1	16	16	16	30	9	9	9	9	25	25	25	3	3	3	14	14	14	14	30	20	20
3	26	18	18	18	18	18	30	23	23	23	23	11	11	11	11	11	17	17	17	17	17	17	26	26
4	28	28	28	28	8	8	30	30	2	2	2	2	24	24	24	4	4	4	4	4	4	4	1	1
5	19	19	19	19	8	8	30	2	2	2	2	10	10	10	10	10	6	6	6	6	26	26	26	19
6	28	28	14	14	14	14	30	23	23	23	23	22	22	22	22	22	22	8	8	12	12	12	28	28
7	2	2	2	8	8	30	30	12	12	12	1	1	15	15	15	15	15	15	15	23	23	23	23	2
8	16	16	24	24	24	1	1	23	23	23	23	11	11	11	11	11	8	8	2	2	2	2	30	16
9	10	2	2	2	2	30	30	15	15	15	15	15	15	15	23	23	23	23	8	8	10	10	10	10
10	23	23	23	30	7	7	30	9	9	9	9	19	19	19	19	19	29	29	29	29	16	16	16	23
11	26	26	19	19	19	19	19	30	20	20	20	16	16	16	5	5	5	5	11	11	11	11	11	26
12	22	22	22	5	5	5	5	12	12	12	30	28	28	28	28	13	13	13	13	13	13	22	22	22
13	22	22	22	22	3	3	3	30	26	26	26	2	2	2	2	6	6	6	6	16	16	16	22	22
14	28	28	28	14	14	14	14	2	2	2	2	30	10	10	10	10	10	8	8	9	9	9	9	28
15	5	5	5	5	8	8	30	12	12	12	23	23	23	23	29	29	29	29	1	1	25	25	25	30
16	18	18	18	18	18	30	30	15	15	15	15	15	15	15	24	24	24	5	5	5	5	12	12	12

Figura 25 - Melhor programação de plantio obtida para o PRC-A-D pelo AG

	Ano 1												Ano 2											
	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1	27	27	8	8	30	1	1	2	2	2	2	19	19	19	19	19	4	4	4	4	4	4	4	27
2	28	28	28	28	3	3	3	11	11	11	11	11	1	1	14	14	14	14	18	18	18	18	18	30
3	23	23	7	7	1	1	30	25	25	25	19	19	19	19	19	29	29	29	29	12	12	12	23	23
4	19	19	19	3	3	3	30	12	12	12	28	28	28	28	21	21	21	21	8	8	1	1	19	19
5	1	1	3	3	3	30	30	12	12	12	28	28	28	28	13	13	13	13	13	13	23	23	23	23
6	16	16	24	24	24	30	30	23	23	23	23	18	18	18	18	18	7	7	1	1	12	12	12	16
7	18	18	6	6	6	6	30	9	9	9	9	1	1	23	23	23	23	21	21	21	21	18	18	18
8	27	9	9	9	9	30	30	19	19	19	19	19	15	15	15	15	15	15	15	12	12	12	27	27
9	16	19	19	19	19	19	30	23	23	23	23	9	9	9	9	5	5	5	5	12	12	12	16	16
10	30	11	11	11	11	11	30	9	9	9	9	27	27	27	15	15	15	15	15	15	15	20	20	20
11	23	23	23	7	7	1	1	18	18	18	18	18	16	16	16	29	29	29	29	12	12	12	30	23
12	20	5	5	5	5	30	30	12	12	12	26	26	26	22	22	22	22	22	22	25	25	25	20	20
13	22	22	11	11	11	11	11	12	12	12	30	18	18	18	18	18	29	29	29	29	22	22	22	22
14	27	27	29	29	29	29	30	2	2	2	2	10	10	10	10	10	8	8	11	11	11	11	11	27
15	16	16	16	30	3	3	3	12	12	12	27	27	27	22	22	22	22	22	22	19	19	19	19	19
16	9	9	9	9	30	8	8	19	19	19	19	19	16	16	16	23	23	23	23	12	12	12	1	1

Figura 26 - Melhor programação de plantio obtida para o PRC-A-D pelo SA

	Ano 1												Ano 2											
	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1	22	22	22	22	22	22	30	12	12	12	20	20	20	8	8	29	29	29	29	11	11	11	11	11
2	2	2	5	5	5	5	30	25	25	25	1	1	13	13	13	13	13	13	21	21	21	21	2	2
3	16	16	16	24	24	24	30	20	20	20	9	9	9	9	7	7	1	1	30	19	19	19	19	19
4	25	8	8	21	21	21	21	12	12	12	2	2	2	2	4	4	4	4	4	4	4	30	25	25
5	26	26	16	16	16	30	30	23	23	23	23	22	22	22	22	22	22	6	6	6	6	1	1	26
6	10	10	10	10	10	30	30	20	20	20	27	27	27	15	15	15	15	15	15	15	23	23	23	23
7	22	22	22	22	1	1	30	12	12	12	30	25	25	25	16	16	16	11	11	11	11	11	22	22
8	10	10	10	10	30	8	8	25	25	25	12	12	12	22	22	22	22	22	22	2	2	2	2	10
9	28	28	28	28	7	7	30	12	12	12	27	27	27	5	5	5	5	1	1	11	11	11	11	11
10	15	15	15	15	15	15	30	16	16	16	12	12	12	30	23	23	23	23	19	19	19	19	19	15
11	19	19	19	19	3	3	3	18	18	18	18	18	30	22	22	22	22	22	22	23	23	23	23	19
12	26	26	29	29	29	29	30	12	12	12	1	1	4	4	4	4	4	4	4	9	9	9	9	26
13	11	11	11	11	11	1	1	19	19	19	19	19	10	10	10	10	10	30	8	8	23	23	23	23
14	26	30	7	7	3	3	3	12	12	12	15	15	15	15	15	15	15	11	11	11	11	11	26	26
15	22	22	9	9	9	9	30	25	25	25	16	16	16	18	18	18	18	18	8	8	22	22	22	22
16	25	30	8	8	1	1	30	16	16	16	22	22	22	22	22	22	14	14	14	14	1	1	25	25

Figura 27 - Melhor programação de plantio obtida para o PRC-A-D pelo AG+SA

	Ano 1												Ano 2											
	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1	24	24	24	8	8	30	30	9	9	9	9	18	18	18	18	18	11	11	11	11	11	20	20	20
2	2	2	5	5	5	5	30	25	25	25	1	1	13	13	13	13	13	13	21	21	21	21	2	2
3	13	13	13	13	13	13	30	1	1	12	12	12	18	18	18	18	18	6	6	6	6	25	25	25
4	22	22	22	22	1	1	30	23	23	23	23	15	15	15	15	15	15	15	30	12	12	12	22	22
5	22	22	9	9	9	9	30	16	16	16	25	25	25	1	1	5	5	5	5	30	22	22	22	22
6	19	19	19	19	19	30	30	20	20	20	12	12	12	15	15	15	15	15	15	15	23	23	23	23
7	10	8	8	1	1	30	30	16	16	16	27	27	27	24	24	24	14	14	14	14	10	10	10	10
8	1	24	24	24	30	8	8	12	12	12	16	16	16	9	9	9	9	7	7	23	23	23	23	1
9	23	23	23	30	3	3	3	9	9	9	9	1	1	11	11	11	11	11	18	18	18	18	18	23
10	9	9	9	9	8	8	30	30	23	23	23	23	18	18	18	18	18	2	2	2	2	16	16	16
11	13	13	13	13	13	13	30	9	9	9	9	1	1	30	15	15	15	15	15	15	15	27	27	27
12	15	15	15	15	15	15	30	2	2	2	2	26	26	26	5	5	5	5	11	11	11	11	11	15
13	12	16	16	16	8	8	30	2	2	2	2	23	23	23	23	4	4	4	4	4	4	4	12	12
14	22	22	2	2	2	2	30	12	12	12	26	26	26	8	8	11	11	11	11	11	22	22	22	22
15	10	16	16	16	3	3	3	1	1	30	23	23	23	23	22	22	22	22	22	22	10	10	10	10
16	22	22	22	22	8	8	30	12	12	12	19	19	19	19	19	7	7	29	29	29	29	30	22	22

Figura 28 - Melhor programação de plantio obtida para o PRC-A-D pelo Memético

A Tabela (34), no Apêndice C, ilustra a produção de cada cultura para estas quatro programações de plantio obtidas. O cômputo desta produção levou em cosideração somente as culturas que produziram em sua respectiva época de demanda.

5 CONCLUSÕES

Neste trabalho, apresentaram-se dois modelos matemáticos para um Problema de Rotação de Culturas. Na primeira abordagem, apresentou-se um modelo para o problema elaborado por Santos et al. (2011) com restrições para lotes adjacentes (PRC-A) e cujo objetivo era maximizar o tempo de ocupação do terreno cultivado. O presente trabalho, no entanto, visou maximizar a lucratividade da rotação realizada.

Na segunda abordagem, inseriu-se no PRC-A restrições de demanda (PRC-A-D), tornando o modelo mais próximo da realidade. Ambos problemas são de natureza combinatorial, com uma elevada quantidade de restrições, e portanto, de alta complexidade.

Isto motivou a aplicação métodos heurísticos de otimização para resolvê-los. Assim, desenvolveram-se e implementaram-se uma heurística construtiva e quatro metaheurísticas, que partem destas soluções construídas para buscar programações de plantio mais promissoras. Utilizou-se nestes dois problemas um Algoritmo Genético, um *Simulated Annealing* e duas abordagens híbridas, um Algoritmo Genético com *Simulated Annealing* e um Memético.

Afim de testar e validar a eficiência destes métodos de otimização, as simulações para o Problema de Rotação de Culturas basearam-se em duas fases. Na primeira, utilizou-se dados hipotéticos, utilizando-se três instâncias distintas. Na segunda fez-se uma aplicação para o PRC-A-D para um conjunto de dados real, com uma uma instância de 16 lotes.

Face à elevada complexidade destes problemas, ficou evidente que os métodos desenvolvidos e implementados foram técnicas promissoras e eficazes, visto

o reduzido tempo computacional utilizado em todas as simulações e aplicações, o elevado número de vezes que estes métodos forneceram soluções factíveis e o aumento Δ promovido no lucro das rotações factíveis encontradas. Isto demonstrou que as programações de plantio fornecidas foram de boa qualidade, especialmente pelos métodos que basearam a sua busca em vizinhanças.

Os resultados deste trabalho ainda demonstraram que o AG foi perdendo a sua eficiência à medida que a instância e o número de restrições do problema aumentou. Por outro lado, foi evidente a vantagem de se construir métodos heurísticos híbridos, principalmente em problemas mais complexos como foi o PRC-A-D. Adicionalmente, nestas experiências, foi destacável a eficiência do SA dentre as metaheurísticas não híbridas, tendo em vista a qualidade de suas soluções fornecidas e o reduzido tempo computacional usado.

Sendo assim, o modelo foi aplicável e resolvido na prática, elaborando-se um horizonte de planejamento para a produção de vegetais em consonância com a preocupação de produzi-los de maneira sustentável e lucrativa, auxiliando os agricultores em suas tomadas de decisões.

REFERÊNCIAS BIBLIOGRÁFICAS

AARTS, E.; KORST, J. Simulated Annealing and Boltzmann Machines. **John Wiley and Sons**, 1989.

AITA, C. Espécies de inverno como fonte de nitrogênio para o milho no sistema de cultivo mínimo e feijão em plantio direto. **Revista Brasileira da Ciência do Solo**, v.18, n.1, p.101–108, 1994.

ALTIERI, M. A. **Agroecologia: bases científicas para uma agricultura sustentável**. Porto Alegre: Agropecuária, 2002.

ARAUJO, H. A. Algoritmo Simulated Annealing: Uma nova abordagem. Florianópolis, 2001. Dissertação (Mestrado) - Universidade Federal de Santa Catarina.

ARF, O.; SILVA, L. S.; BUZETTI, S. Efeito da Rotação de Culturas, Adubação Verde e Nitrogenada sobre o Rendimento do Feijão. **Pesquisa agropecuária brasileira, Brasília**, v.34, n.11, p.2029–2036, 1999.

BAKER, J. E. An analysis of the the effects of selection in genetic algorithms. Nashiville, 1989. 198-220p. Phd thesis - Graduate Schol of Vanderbilt University.

BERNARDINO, H. S. Hibridização de Algoritmos Genéticos e Sistemas Imunológicos Artificiais para Problemas de Otimização com Restrições em Engenharia. Juiz de Fora, 2008. 40-43p. Dissertação (Mestrado) - Universidade Federal de Juiz de Fora.

BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. **ACM Computing Surveys**, v.35, n.3, p.268–308, 2003.

BONA, A. A.; ALGERI, A. Algoritmo de Otimização Combinatorial: Uma Proposta Híbrida Utilizando os Algoritmos Simulated Annealing e Genético em Ambiente Multiprocessado. **Otimização Combinatorial**, 2001.

BUZZO, W. R.; MOCCELLIN, J. V. Programação da produção em sistemas Flow Shop utilizando um método heurístico híbrido Algoritmo Genético e Simulated Annealing. **Gestão e Produção**, v.7, n.3, p.364–377, 2000.

CASEAU, Y.; LABURTHE, F. Heuristics for large constrained vehicle routing problems. **Journal of Heuristics**, v.5, p.281–303, 1999.

CHIPPERFIELD, A.; FLEMING, P. Proceedings of the 13th IFAC World Congress. In: 2, USA, 1996. ; resumos. San Francisco: CA, 1996. 181-186.

COELHO, L. S. Fundamentos Potencialidades e Aplicações de Algoritmos Evolutivos. In: 2, SBMAC, 2003. **Notas de Matemática Aplicada**; resumos. São Carlos: XXVI CNMAC, 2003. 1-111.

CONSTANTINO, A. A.; REIS, P. A.; MENDONÇA, C. F. X. N.; FIGUEIREDO, M. F. Aplicação de Algoritmo Genético ao Problema de Cobertura de Conjunto. **XXV SBPO**, p.1406–1414, 2003.

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

DEB, K.; GOLDBERG, D. E. A comparison of selection schemes used in genetic algorithms. Foundations of Genetic Algorithms. **Morgan Kaufmann: San Mateo**, p.69–93, 1991.

DEJONG, K. A. An analysis of the the effects of selection in genetic algorithms. Nashiville, 1975. Phd thesis - Graduate Schol of Vanderbilt University.

DIESTEL, R. **Graph Theory**. Graduate Text in Mathematics, 2000.

DOWSLAND, K. A. 4. In: SPRINGER (Ed.). **Modern Heuristic Techniques for Combinatorial Problems Advanced Topics in Computer Science Series**. London: Blackwell Scientific Publications, 1993. p.20-69.

DUARTE, J. B. J.; COELHO, F. C. Rotação de Culturas. **Manual técnico**, v.1, p.147–155, 2002.

FILGUEIRA, F. A. R. **Novo Manual de Olericultura**. Viçosa: UFV, 2003.

FREITAS, M. J. S. Inteligência Computacional para Otimização. In: UFOP, 2009. **Notas de aula**; resumos. Ouro Preto: , 2009. 1-59.

GEYER, A.; SCHULTZ, A. Fuzzy rule-based expert systems and genetic machine learning. **Heidelberg: Physica-Verlag**, 1997.

GLIESSMAN, S. R. **Agroecologia: Processos ecológicos em agricultura sustentável**. Porto Alegre: UFRGS, 2000.

GOLDBERG, D. E. Genetic Algorithms and Machine Learning. **Kluwer Academic Publishers**, v.3, 1989.

HELL, P.; KLEIN, S.; NOGUEIRA, L. T.; PROTTI, F. Particionamento de Grafos Cordais em Conjuntos Independentes e Cliques. **Tendências em Matemática Aplicada e Computacional**, v.1, p.147–148, 2002.

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

KOZA, J. R. Genetic Programing: on the programing of the computers by means of the natural selection. **Cambridge**, 1992.

LACERDA, E. G. M.; CARVALHO, A. C. P. 3. In: **Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais. Introdução aos Algoritmos Genéticos**. Porto Alegre - RS. Disponível em: http://www.dca.ufrn.br/ estefane/metaheuristicas/ag.pdf: , 1999. p.87-146.

LAWLER, E. L.; LENSTRA, J. K.; RINNOOY, A. H. G.; SHMOYS, D. B. The traveling salesman problem: A Guided Tour of Combinatorial Optimization. **New York: Wiley**, 1985.

LEMALADE, J. L.; NAGIH, A.; PLATEAU, G. A MIP flow model for crop-rotation planning in a context of forest sustainable development. In: , 2011. **Annals of Operations Research 190**; resumos. , 2011. 149-164.

METROPOLIS, N.; ROSENBLUTH, A. W.; ROSENBLUTH, M. N.; TELLER, A. H. Equation of State Colaculation by Fast Computing Machines. **Chen**, v.21, p.1087–1091, 1953.

MICHALEWICZ, Z.; FOGEL, B. D. 5-7. In: SPRINGER (Ed.). **How to solve it: Modern Heuristics**. USA: , 2000. p.115-183.

MITCHELL, M. An introduction to genetic algorithms. **Cambridge**, v.2, p.207–218, 1997.

MUZILLI, O. Adubação nitrogenada em milho no Paraná. III. Influência da recuperação do solo com adubação verde de inverno nas respostas a adubação nitrogenada. **Revista Brasileira da Ciência do Solo**, v.18, n.1, p.23–27, 1983.

MYIAZAKA, S.; CAMARGO, O. A. Adubação orgânica, adubação verde, e rotação de culturas no estado de São Paulo. **Fundação Cargill**, p.1–36, 1984.

NEMHAUSER, G. L.; WOLSEY, L. A. 1. In: PUBLICATION, W. I. (Ed.). **Integer and Combinatorial Otimization**. USA: , 1999. p.2-20.

NETO, A. J. S.; BECCENERI, J. C. Técnicas de Inteligência Computacional Inspiradas na Natureza: Aplicação em Problemas Inversos de Transferência Radioativa. In: 41, CNMAC, 2009. **Notas em Matemática Aplicada**; resumos. São José dos Campos: SBMAC, 2009. 1-122.

PACHECO, M. A. C. Algoritmos Genéticos: Princípios e Aplicações. **Laboratório de Inteligência Computacional Aplicada**, v.65, p.1–9, 1994.

REEVES, R. C.; BEASLEY, J. E. Combinatorial Problems. In: SPRINGER (Ed.). **Modern Heuristic Techniques for Combinatorial Problems, Advanced Topics in Computer Science Series**. London: Blackwell Scientific Publications, 1993. p.

SANTOS, L. M. R.; MICHELON, P. R. H.; ARENALES, M. N.; S, S. R. H. Crop rotation scheduling with adjacency constraint. In: , 2011. **Annals of Operations Research 190**; resumos. , 2011. 165-180.

SANTOS, R. H. S. Olericultura: teoria e prática. In: FONTES, P. C. R. (Ed.). **Olericultura orgânica**. Viçosa: Universidade Federal de Viçosa, 2005. p.249-276.

SARAMAGO, S. F. P. Métodos de Otimização Randômica: Algoritmos Genéticos e Simulated Annealing. In: 6, CNMAC, 2003. **Notas em Matemática Aplicada**; resumos. Uberlândia: SBMAC, 2003. 1-42.

SCHICK, J. Erosão hídrica em Cambissolo Húmico Alumínico submetido a diferentes sistemas de preparo e cultivo do solo: I. Perdas de solo e água. **Revista Brasileira da Ciência do Solo**, v.24, p.427–436, 2000.

SOUZA, J. L.; RESENDE, P. L. **Manual de Horticultura Orgânica**. Viçosa: Aprenda Fácil, 2006.

STANDERSKI, N. Aplicação de Algoritmo Genético Paralelos a Problemas de Grande Escala de VMI. **SBPO**, p.1183–1195, 2003.

TORREAO, J. R. A. Inteligência Computacional. In: Universidade FederalFluminense, 2004. **Notas de aula**; resumos. Niterói: , 2004.

YEN, J. A Hibrid approach to modeling metabolic systens using a genetic algorithm and simplex method. **IEEE Transactions on Systens Man and Cybernetics**, v.28, n.2, p.173–191, 1998.

APÊNDICES

Apêndice A

Em todos os experimentos computacionais para o PRC, foram utilizadas 29 culturas, sendo as plantas 1-25 comerciais e as remanescentes destinadas à adubação verde e o pousio. Os dados intríncecos às culturas, utilizados no PRC-A e PRC-A-D encontram-se na Tabela (27).

Tabela 27 - Dados das culturas: nome, família, época de plantio e duração do ciclo de vida

Cultura		**Família**		**Época de plantio**		**Ciclo**	
				Início	**Fim**	**dias**	t_i
1	Alface	1	*Compositaee*	ano todo		60	2
2	Almerão	1	*Compositae*	ano todo		120	4
3	Couve	2	*Brassicaceae*	março	junho	90	3
4	Brócolis	2	*Brassicaceae*	fevereiro	junho	210	7
5	Repolho	2	*Brassicaceae*	fevereiro	junho	120	4
6	Couve-Flor	2	*Brassicaceae*	março	junho	120	4
7	Beterraba	3	*Chenopodiaceae*	março	julho	60	2
8	Espinafre	3	*Chenopodiaceae*	fevereiro	julho	60	2
9	Abobrinha	4	*Cucurbitaceae*	agosto	março	120	4
10	Moranga	4	*Cucurbitaceae*	setembro	janeiro	150	5
11	Pepino	4	*Cucurbitaceae*	ano todo		120	4
12	Melancia	4	*Cucurbitaceae*	agosto	outubro	90	3
13	Alho	5	*Liliaceae*	março	abril	180	6
14	Cebola	5	*Liliaceae*	março	junho	120	4
15	Quiabo	6	*Malvaceae*	agosto	março	210	7
16	Milho	7	*Gramineae*	agosto	abril	90	3
17	Aveia	7	*Gramineae*	março	maio	180	6
18	Tomate	8	*Solanaceae*	ano todo		150	5
19	Pimentão	8	*Solanaceae*	ano todo		150	5
20	Batata	8	*Solanaceae*	agosto	outubro	90	3
21	Cenoura	9	*Umbelliferae*	março	julho	120	4
22	Salsinha	9	*Umbelliferae*	setembro	março	180	6
23	Feijão Vagem	10	*Leguminosae*	agosto	abril	120	4
24	Ervilha	10	*Leguminosae*	março	abril	90	3
25	Feijão	10	*Leguminosae*	agosto	setembro	90	3
26	Crotalália	10	*Leguminosae*	setembro	dezembro	90	3
27	Feijão-de-porco	10	*Leguminosae*	setembro	dezembro	90	3
28	Mucuna	10	*Leguminosae*	setembro	janeiro	90	3
29	Ervilha Peluda	10	*Leguminosae*	março	junho	120	4
30	Pousio	11	-	ano todo		30	1

Apêndice B

5.1 Dados para o PRC-A

Nos testes computacionais para o PRC-A, utilizou-se três dimensões de áreas de plantio: $L = 10, 15$ e 20 lotes, cujas geometrias encontram-se na Figura (29) (estes lotes são hipotéticos).

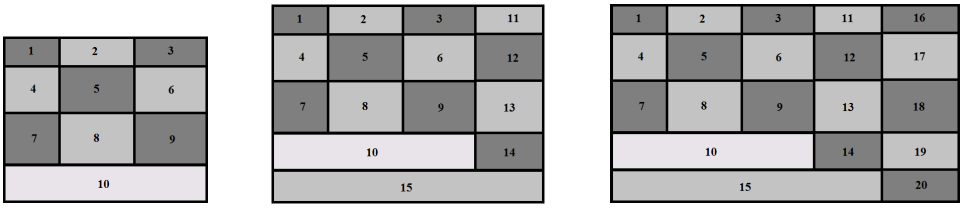

Figura 29 - 10, 15 e 20 lotes, respectivamente

Os valores das áreas (em ha.) de cada lote também foram valores hipotéticos. Seus respectivos valores encontram-se na Tabela (28) a seguir:

Tabela 28 - Área dos 20 lotes hipototéticos

Lote	Área	Lote	Área
1	1,50	11	2,00
2	2,00	12	3,00
3	2,00	13	4,00
4	2,25	14	3,00
5	3,00	15	5,50
6	3,00	16	2,50
7	3,00	17	3,75
8	4,00	18	5,00
9	4,00	18	3,75
10	8,25	20	2,50

A Tabela (29) se refere à lucratividade em R$ (dados hipotéticos) das culturas nas simulações para o PRC-A. Estes números referem-se ao lucro da cultura i em seus t_i períodos por hectare.

Tabela 29 - Lucratividade das culturas em PRC-A nos t_i períodos por hectare

Cultura	Lucratividade	Cultura	Lucratividade
Alface	300	Cebola	430
Almeirão	150	Quiabo	710
Couve	300	Milho	350
Brócolis	400	Aveia	350
Repolho	400	Tomate	810
Couve-Flor	900	Pimentão	550
Beterraba	810	Batata	240
Espinafre	600	Cenoura	620
Abobrinha	400	Salsinha	400
Moranga	200	Feijão-Vagem	750
Pepino	450	Ervilha	830
Melancia	900	Feijão	720
Alho	810	-	-

Apêndice C

5.2 Dados para o PRC-A-D

Já para a aplicação real para o PRC-A-D a estrutura geográfica da área compreendia 16 lotes em forma de tabuleiro como a seguir:

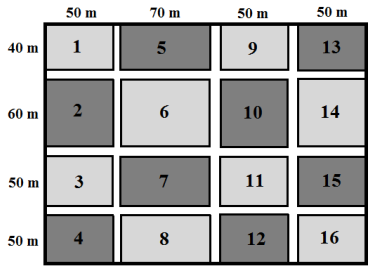

Figura 30 - Estrutura geográfica da área real aplicada

A Tabela (30) seguinte fornece o valor das áreas (ha.) destes lotes.

Tabela 30 - Área dos 16 lotes reais

Lote	Área	Lote	Área
1	0,2	11	0,20
2	0,3	12	0,30
3	0,25	13	0,25
4	0,25	14	0,25
5	0,28	15	0,20
6	0,42	16	0,30
7	0,35	17	0,25
8	0,35	18	0,25

A próxima Tabela (31) ilustra a demanda mensal das 24 culturas comerciais[9] no período em que são solicitadas e na unidade em que são comercializadas.

Tabela 31 - Produtividade, lucratividade e período de demanda mensal das culturas comerciais

Cultura	Época de demanda		Demanda
	Início	Fim	
Alface	Dezembro	Março	3.750 pés
Almerão	Novembro	Março	6.000 pés
Couve	Junho	Setembro	1.667 pés
Brócolis	Setembro	Dezembro	1.333 pés
Repolho	Agosto	Novembro	1.000 pés
Couve-Flor	Agosto	Outubro	1.250 pés
Beterraba	Maio	Setembro	3.000 kg
Espinafre	Julho	Setembro	5.000 pés
Abobrinha	Janeiro	Março	5.000 unidades
Moranga	Fevereiro	Junho	4.000 unidades
Pepino	Abril	Julho	5.000 unidades
Melancia	Novembro	Janeiro	1.875 unidades
Alho	Setembro	Outubro	100 kg
Cebola	Agosto	Outubro	875 kg
Quiabo	Agosto	Outubro	1.250 kg
Milho	Janeiro	Maio	200 kg
Aveia	-	-	0 kg
Tomate	Maio	Agosto	2.500 kg
Pimentão	Maio	Setembro	800 kg
Batata	Novembro	Janeiro	2.500 kg
Cenoura	Agosto	Novembro	1.000 kg
Salsinha	Abril	Agosto	1.000 kg
Feijão Vagem	Dezembro	Abril	400 kg
Ervilha	Junho	Julho	125 kg
Feijão	Novembro	Dezembro	100 kg

As Tabelas (32) e (33) denotam, respectivamente, a lucratividade e produtividade reias das culturas comerciais durante o ano. Estes valores se referem

[9] A cultura de Aveia não foi demandada.

ao mês em que a colheita foi efetuada. A lucratividade está na unidade R$ por m^2 e a produtividade na unidade em que são vendidas (maços, pés, kg, unidades) por m^2. Os campos em "-" indicam que a cultura não pôde ser colhida naquele mês por estar fora da época.

Tabela 32 - Lucratividade mensal (R$) por m^2 das culturas comerciais

Cultura	**Jan**	**Fev**	**Mar**	**Abr**	**Mai**	**Jun**	**Jul**	**Ago**	**Set**	**Out**	**Nov**	**Dez**
Alface	4	4	4	4	3	3	3	3	4	4	4	4
Almeirão	4	4	4	4	3	3	3	3	4	4	4	4
Couve	-	-	-	-	-	4	3	3	3	-	-	-
Brócolis	-	-	-	-	-	-	-	-	6	6	5	5
Repolho	-	-	-	-	-	6	5	5	5	6	6	-
Couve-Flor	-	-	-	-	-	-	-	5	5	6	-	-
Beterraba	-	-	-	-	0,9	0,9	0,9	0,8	0,8	-	-	-
Espinafre	-	-	-	10	10	10	6	6	6	-	-	-
Abobrinha	20	20	20	15	15	15	15	-	-	-	-	15
Moranga	15	20	20	20	20	20	-	-	-	-	-	-
Pepino	15	15	15	20	20	20	20	20	20	15	15	15
Melancia	50	-	-	-	-	-	-	-	-	-	30	30
Alho	-	-	-	-	-	-	-	-	4	4	-	-
Cebola	-	-	-	-	-	-	1	0,8	0,8	0,8	-	-
Quiabo	-	-	10	10	8	8	8	8	10	10	-	-
Milho	1	1	1	0,8	0,8	0,8	0,8	-	-	-	0,8	0,8
Aveia	-	-	-	-	-	-	-	-	0,8	0,8	1	-
Tomate	5	5	5	6	6	6	6	6	6	5	5	5
Pimentão	5	5	5	6	6	6	6	6	6	5	5	5
Batata	0,5	-	-	-	-	-	-	-	-	-	0,5	0,5
Cenoura	-	-	-	-	-	-	0,9	0,7	0,7	0,7	0,7	-
Salsinha	-	-	10	15	15	15	15	15	10	-	-	-
Feijão Vagem	0,2	0,2	0,2	0,2	0,15	0,15	0,15	0,2	-	-	-	0,2
Ervilha	-	-	-	-	-	0,15	0,15	-	-	-	-	-
Feijão	-	-	-	-	-	-	-	-	-	-	0,2	0,2

Tabela 33 - Produtividade mensal por m^2 das culturas comerciais

Cultura	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Alface (pés)	12	12	12	12	16	16	16	16	12	12	12	12
Almeirão (pés)	12	12	12	12	16	16	16	16	12	12	12	12
Couve (pés)	-	-	-	-	-	3	4	4	4	-	-	-
Brócolis (pés)	-	-	-	-	-	-	-	-	4	4	5	5
Repolho (pés)	-	-	-	-	-	3	4	4	4	3	3	-
Couve-Flor (pés)	-	-	-	-	-	-	-	4	4	3	-	-
Beterraba (kg)	-	-	-	-	5	5	5	8	8	-	-	-
Espinafre (pés)	-	-	-	8	8	8	10	10	10	-	-	-
Abobrinha (unidades)	12	12	12	20	20	20	20	-	-	-	-	20
Moranga (unidades)	30	15	15	15	15	30	-	-	-	-	-	-
Pepino (unidades)	20	20	20	15	12	15	15	15	15	20	20	20
Melancia (unidades)	5	-	-	-	-	-	-	-	-	-	10	10
Alho (kg)	-	-	-	-	-	-	-	-	1	1	-	-
Cebola (kg)	-	-	-	-	-	-	3	5	5	5	-	-
Quiabo (kg)	-	-	4	4	5	5	5	5	4	4	-	-
Milho (kg)	0,8	0,8	0,8	1	1	1	1	-	-	-	1	1
Aveia (kg)	-	-	-	-	-	-	-	-	1	1	0,8	-
Tomate (kg)	10	10	10	8	8	8	8	8	8	10	10	10
Pimentão (kg)	4	4	4	3	3	3	3	3	3	4	4	4
Batata (kg)	10	-	-	-	-	-	-	-	-	-	10	10
Cenoura (kg)	-	-	-	-	-	-	3	5	5	5	5	5
Salsinha (maços)	-	-	50	40	40	40	40	40	50	-	-	-
Feijão Vagem (kg)	0,4	0,4	0,4	0,4	0,5	0,5	0,5	0,4	-	-	-	0,4
Ervilha (kg)	-	-	-	-	-	0,4	0,4	-	-	-	-	-
Feijão (kg)	-	-	-	-	-	-	-	-	-	-	0,4	0,4

A Tabela (34) ilustra a produção de cada cultura nas rotações que forneceram os maiores lucros nas quatro metodologias propostas.

Tabela 34 - Produção por cultura obtida pelas melhores programações de plantio encontradas

Culturas	**AG**	**SA**	**AG+SA**	**Memético**
Alface (pés)	30.000	48.000	92.400	54.000
Almeirão (pés)	60.000	78.000	66.000	90.000
Couve (pés)	27.000	13.500	16.800	18.000
Brócolis (pés)	10.000	15.500	10.000	24.000
Repolho (pés)	12.000	21.600	8.000	26.500
Couve-Flor (pés)	15.000	17.500	12.600	10.000
Beterraba (kg)	53.000	33.600	32.500	66.500
Espinafre (pés)	33.600	44.000	60.000	68.000
Abobrinha (unidades)	70.000	80.400	72.000	60.000
Moranga (unidades)	168.000	157.500	120.000	105.000
Pepino (unidades)	176.000	96.000	110.000	110.000
Melancia (unidades)	42.500	79.000	128.500	136.500
Alho (kg)	2.240	2.400	6.400	4.400
Cebola (kg)	12.500	22.500	10.000	10.000
Quiabo (kg)	15.000	33.500	27.500	25.000
Milho (kg)	4.640	4.240	8.000	8.240
Aveia (kg)	0	2500	0	0
Tomate (kg)	45.000	55.000	103.000	87.000
Pimentão (kg)	34.000	12.000	14.000	10.000
Batata (kg)	25.000	54.400	25.000	25.000
Cenoura (kg)	30.000	32.500	15.000	17.400
Salsinha (maço)	100.000	225.000	440.000	275.000
Feijão-Vagem (kg)	41.00	4.900	55.40	4.390
Ervilha (kg)	3.350	1.600	2.260	2.430
Feijão (kg)	1.350	1.850	3.310	4.050

Apêndice D

As Figuras (31), (32) e (33) mostram as melhores programações de plantio para o PRC-A utilizando-se as três instâncias consideradas, $L = 10$, 15 e 20 respectivamente.

	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1	10	10	10	24	24	24	30	25	25	25	10	10
2	28	28	8	8	30	1	1	12	12	12	28	28
3	9	9	9	3	3	3	30	23	23	23	23	9
4	28	28	8	8	1	1	30	12	12	12	28	28
5	10	10	10	10	30	7	7	23	23	23	23	10
6	23	23	23	1	1	30	30	9	9	9	9	23
7	10	10	10	10	7	7	30	23	23	23	23	10
8	27	27	7	7	3	3	3	16	16	16	30	27
9	9	9	9	9	30	7	7	25	25	25	1	1
10	23	23	23	21	21	21	21	30	12	12	12	23

Figura 31 - Melhor programação factível encontrada para $L = 10$

	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1	22	22	22	22	7	7	30	25	25	25	22	22
2	23	23	9	9	9	9	30	12	12	12	23	23
3	18	18	18	18	7	7	30	25	25	25	30	18
4	23	23	9	9	9	9	30	12	12	12	23	23
5	15	15	29	29	29	29	30	15	15	15	15	15
6	9	9	9	23	23	23	23	12	12	12	30	9
7	9	9	24	24	24	30	30	20	20	20	9	9
8	19	19	19	3	3	3	30	25	25	25	19	19
9	28	28	14	14	14	14	30	16	16	16	28	28
10	23	23	21	21	21	21	30	12	12	12	23	23
11	23	23	23	3	3	3	30	9	9	9	9	23
12	27	16	16	16	7	7	30	25	25	25	27	27
13	30	8	8	24	24	24	30	12	12	12	1	1
14	11	11	11	30	3	3	3	25	25	25	11	11
15	28	28	28	28	30	7	7	1	1	16	16	16

Figura 32 - Melhor programação factível encontrada para $L = 15$

	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1	23	23	9	9	9	9	30	12	12	12	23	23
2	28	28	7	7	1	1	30	25	25	25	28	28
3	26	16	16	16	7	7	30	12	12	12	26	26
4	16	1	1	7	7	30	30	25	25	25	16	16
5	23	23	14	14	14	14	30	12	12	12	23	23
6	9	9	9	23	23	23	23	30	16	16	16	9
7	18	18	18	24	24	24	30	12	12	12	18	18
8	10	10	10	30	3	3	3	25	25	25	10	10
9	23	23	7	7	1	1	30	12	12	12	23	23
10	28	28	28	28	30	7	7	1	1	16	16	16
11	10	10	10	24	24	24	30	20	20	20	10	10
12	27	30	7	7	3	3	3	25	25	25	27	27
13	15	15	23	23	23	23	30	15	15	15	15	15
14	9	9	30	8	8	1	1	25	25	25	9	9
15	23	23	30	3	3	3	30	12	12	12	23	23
16	23	23	1	1	30	7	7	12	12	12	23	23
17	9	9	23	23	23	23	30	20	20	20	9	9
18	1	16	16	16	3	3	3	25	25	25	30	1
19	23	23	23	1	1	7	7	20	20	20	30	23
20	9	9	21	21	21	21	30	25	25	25	9	9

Figura 33 - Melhor programação factível encontrada para $L = 20$