



Universidade de Pernambuco
Departamento de Sistemas e Computação

Engenharia da Computação

**Uma Aplicação do Algoritmo
Genético Multiobjetivo NSGA II
Para Seleção de Imagens de Satélite
de Trechos de Mata Atlântica**

Davi Marinho

Trabalho de Graduação

Recife
22 de junho de 2009

Universidade de Pernambuco
Departamento de Sistemas e Computação

Davi Marinho

**Uma Aplicação do Algoritmo Genético Multiobjetivo
NSGA II Para Seleção de Imagens de Satélite de
Trechos de Mata Atlântica**

*Trabalho apresentado ao Programa de Engenharia da
Computação do Departamento de Sistemas e Compu-
tação da Universidade de Pernambuco como requisito
parcial para obtenção do grau de Bacharel em Engen-
haria da Computação.*

Orientador: *Prof. Carmelo J. A. Bastos Filho*
Co-orientador: *Augusto Santiago (Dep. Botânica, UFPE)*

Recife
22 de junho de 2009

Deleita-te também no SENHOR, e te concederá os desejos do teu coração. Entrega o teu caminho ao SENHOR; confia nele, e ele o fará.

—SALMOS 37:4-5 (Bíblia Sagrada)

*Aos meus pais,
Antônio Falcão e
Miriam Falcão.*

Resumo

A devastação ambiental da Mata Atlântica Brasileira é um problema de motivos históricos e que vem preocupando e mobilizando ambientalistas, pesquisadores e setores do meio político. O uso de fotografias dessa floresta obtidas de satélite é um exemplo de que a tecnologia pode ser usada como forma de obtenção e tratamento dos objetos, que serão usados como fontes para o estudo do problema. Através da segmentação dessas imagens, trechos de florestas são detectados. Entretanto, surge a necessidade de selecionar fragmentos de forma a atender critérios de similaridade. Como vários critérios devem ser atendidos, a otimização multiobjetiva utilizando um Algoritmo Genético chamado NSGA II pode ser indicado como possível forma de solucionar o problema. As simulações e as análises dos Pareto através das métricas: Hypervolume, Maximum Spread e Spacing, mostraram que o NSGA II foi capaz de maximizar eficientemente os três objetivos de similaridade apresentados no problema e dessa forma demonstrando ser um bom algoritmo de busca para automatizar o processo de seleção desses trechos de imagens de satélite.

Palavras-chave: devastação ambiental, Mata Atlântica Brasileira, tecnologia, segmentação, critério de similaridade, otimização multiobjetiva, Algoritmo Genético, NSGA II, Pareto.

Abstract

The environmental devastation of the Brazilian Atlantic Forest is a problem with historical reasons. This is worrying and mobilizing environmentalists, researchers and political sectors. The use of photos of forests obtained by satellites is an example of how the technology can be used to acquire and process data. These photos can be used to propose solutions to the problem. By using image segmentation, portions of forests can be localized. However, it is necessary to select fragments in order to achieve the criterion of similarity. As many criteria must be considered, the multiobjective optimization using a genetic algorithm called NSGA II can be indicated as a possible solution to the problem. The simulations and analysis of Pareto through the metrics Hypervolume, Maximum Spread and Spacing, showed that the NSGA II was able to efficiently maximize the similarity of the three objectives presented in the problem and thus proving to be a good search algorithm to automate the process selection of these pieces of satellite images.

Keywords: environmental devastation, Brazilian Atlantic Forest, technology, segmentation, the similarity criterion, Multiobjective optimization, Genetic Algorithm, NSGA II, Pareto.

Sumário

1	Introdução ao problema ambiental	1
1.1	A Mata Atlântica Brasileira e o problema da devastação ambiental.....	1
1.2	O uso de tecnologia no controle da devastação ambiental.....	2
1.3	Estrutura da Monografia.....	4
2	Fotografias Obtidas de Satélite e Segmentação de Imagens	5
2.1	Fotografias Obtidas de Satélite e Segmentação de Imagens.....	5
2.2	Objetivos para Otimização.....	8
3	NSGA II - Non-dominated Sorting Genetic Algorithm II	10
3.1	Conceito de Dominância.....	10
3.2	Algoritmos Genéticos.....	11
3.2.1	Geração de população inicial.....	12
3.2.2	Cruzamento.....	13
3.2.3	Mutação.....	14
3.2.4	Seleção.....	15
3.2.5	Convergência.....	15
3.3	NSGA II.....	16
3.3.1	Fast Non-Dominated Sorting.....	18
3.3.2	Crowding distance.....	20
4	Ferramenta para Seleção de Imagens de Trechos de Mata Atlântica ..	23
4.1	Técnicas Utilizadas.....	23
4.2	Uma Ferramenta inteligente usando NSGA II.....	25
4.3	Métricas para Avaliação de Pareto.....	29
4.3.1	HyperVolume.....	29
4.3.2	Maximum Spread.....	29
4.3.3	Spacing.....	30
5	Resultados	31
6	Conclusão	36
	Referências Bibliográficas	37

Lista de Figuras

2.1	<i>Exemplo de fotografia de trecho de Mata Atlântica obtida por satélite.</i>	6
2.2	<i>Fotografia segmentada de trecho de Mata Atlântica.</i>	6
2.3	<i>A figura demonstra como a conectividade é calculada.</i>	7
3.1	<i>Diagrama de Blocos do Algoritmo Genético clássico.</i>	12
3.2	<i>Diagrama de Blocos do Algoritmo NSGA II, descrevendo o funcionamento da etapa de seleção.</i>	17
3.3	<i>A Figura mostra qual o conceito do cálculo da distância feito pelo Operador de Diversidade.</i>	21
4.1	<i>A Figura mostra a interface da ferramenta gráfica Gnuplot.</i>	25
4.2	<i>A Figura mostra um resultado gráfico obtido pela ferramenta Gnuplot.</i>	25
4.3	<i>Interface gráfica da ferramenta inteligente que implementa o algoritmo NSGA II.</i>	26
4.4	<i>Ferramenta Inteligente simulando 1000 gerações e com uma população inicial de 100 indivíduos.</i>	28
4.5	<i>Resultados gráficos que serão usados para análise das soluções obtidas pelo algoritmo. Através dos gráficos pode-se selecionar qual o ponto ou conjunto de pontos que melhor soluciona o problema em questão.</i>	28
5.1	<i>Simulação com 10 gerações e com população inicial de 100 indivíduos.</i>	33
5.2	<i>Simulação com 50 gerações e com população inicial de 100 indivíduos.</i>	33
5.3	<i>Simulação com 100 gerações e com população inicial de 100 indivíduos.</i>	34
5.4	<i>Simulação com 10000 gerações e com população inicial de 100 indivíduos.</i>	34
5.5	<i>Simulação com 10, 50, 100 e 10000 gerações e com população inicial de 100 indivíduos.</i>	35

Lista de Tabelas

3.1	<i>Exemplo de indivíduo formado de nove imagens representadas por valores inteiros, que codificam 9 imagens de satélite.</i>	13
3.2	<i>Exemplificação da etapa de cruzamento.</i>	13
3.3	<i>Exemplificação da etapa de mutação.</i>	14
3.4	<i>Pseudo-código Processo 1 do fast non-dominated sorting.</i>	19
3.5	<i>Pseudo-código Processo 2 do fast non-dominated sorting.</i>	20
3.6	<i>Pseudo-código do algoritmo Crowding Distance.</i>	22
5.1	<i>Resultados das métricas usadas para avaliação do Pareto.</i>	32

Introdução ao problema

Este trabalho tem como principal finalidade demonstrar uma aplicação de Inteligência Computacional como forma de auxiliar nas pesquisas sobre a devastação ambiental da Mata Atlântica brasileira. Também visa descrever com detalhes o algoritmo utilizado, e através da análise dos resultados poder concluir algo a respeito da eficiência do algoritmo.

Este capítulo contém algumas informações na seção 1.1 a respeito da Mata Atlântica brasileira, sua existência ao longo da história do Brasil e o grave problema de devastação ambiental que vem sofrendo desde a colonização até os dias atuais. Será feita uma breve citação sobre como o grupo de pesquisa em Computação Inteligente da UPE, através deste trabalho, auxiliou na atividade de pesquisa do Departamento de Botânica da UFPE (Universidade Federal de Pernambuco) envolvendo estudos a respeito imagens obtidas por meio de satélite. Na seção 1.2 será descrito sobre o uso de tecnologia no controle da devastação ambiental e na seção 1.3 conterà um resumo sobre a estrutura da monografia em termos de conteúdo.

1.1 A Mata Atlântica Brasileira e o problema da devastação ambiental

A Mata Atlântica originalmente percorria toda a costa brasileira. Tratava-se da segunda maior floresta tropical úmida do Brasil, só comparável à Floresta Amazônica. Um dos seus destaques era o pau-brasil, árvore que deu origem ao nome do país. No entanto, hoje é quase uma relíquia, existindo apenas alguns exemplares no Sul da Bahia.

Entretanto, ainda restam trechos da floresta na Serra do Mar e na Serra da Manti-

queira, no sudeste do Brasil. Seu clima é equatorial ao norte e quente temperado sempre úmida ao sul. Tem temperaturas médias elevadas durante o ano todo e não apenas no verão. A alta pluviosidade nessa região deve-se à barreira que a serra constitui para os ventos que sopram do mar. Seu solo é pobre e a topografia é bastante acidentada. No interior da mata, devido a densidade da vegetação, a luz é reduzida.

As condições físicas na floresta atlântica variam muito, dependendo do local estudado. Assim, apesar de a região estar submetida a um clima geral, há microclimas muitos diversos e que variam de cima para baixo nos diversos extratos. Os teores de oxigênio, luz, umidade e temperatura são bem diferentes dependendo da camada considerada [5].

Atualmente, restam apenas cerca de 5% da extensão original da segunda maior floresta brasileira. Hoje a maioria da área litorânea que era coberta pela Mata Atlântica é ocupada por grandes cidades, pastos e agricultura. Para tentar reverter esse quadro, ONGs (organizações não-governamentais), órgãos do governo, instituições e personalidades isoladamente têm buscado soluções visando a preservação e, se possível, a reconstituição do bioma [7].

”Nós estamos em um momento extremamente crítico, porque ou a gente estabelece realmente uma política de longo prazo de proteção da Mata Atlântica, com projetos completos - criação de novas unidades de conservação, adoção de diversas formas de intervenção -, ou, realmente, nós vamos perder o controle”, avalia Luiz Paulo Pinto, diretor para a Mata Atlântica da ONG Conservação Internacional [6].

1.2 O uso de tecnologia no controle da devastação ambiental

A tecnologia tem sido bastante utilizada no sentido de oferecer informações cada vez mais precisas para que se possa haver estudos mais completos sobre as questões ambientais de um modo geral. São vários os exemplos de aplicações tecnológicas com essa finalidade. Existem softwares que utilizam modelos matemáticos robustos que conseguem relacionar uma série de dados e, a partir dessas combinações, simular situações, devolvendo como

resultado dados estatísticos. Isso é bastante útil quando se pretende simular os impactos de uma determinada ação sobre o meio ambiente.

O monitoramento do grau de devastação ambiental por meio de imagens tiradas de satélite, o uso de técnicas de Processamento de Imagens, Inteligência Computacional e técnicas multiobjetivas também são exemplos bastante expressivos do uso de tecnologia no processo de controle da devastação ambiental. Pode-se ver que são várias abordagens existentes que podem ser utilizadas para solucionar ou pelo menos amenizar os problemas ambientais.

O departamento de Botânica da Universidade Federal de Pernambuco dispõe, para serem analisadas, de cerca de 1217 imagens de trechos de Mata Atlântica, oriundas de um pré-processamento de uma fotografia tirada por meio de satélite. A seleção dos melhores trechos devia ser feita de forma visual, contudo com o intuito de facilitar o procedimento, procurou-se uma forma de automatizar essa seleção de imagens utilizando uma técnica computacional de busca multiobjetiva, ou seja, onde mais de um aspecto é considerado importante durante a busca das soluções. É importante ressaltar que, à medida que o número de imagens aumenta, aumenta também a complexidade de seleção de um subconjunto de imagens.

O mecanismo proposto foi o Algoritmo Genético (*GA - Genetic Algorithm*) multiobjetivo NSGA II (*Nondominated sorting genetic algorithm*) [8]. A necessidade deste algoritmo decorre do fato de que os três parâmetros indicadores, utilizados nesse trabalho, são igualmente importantes durante o processo de escolha dos melhores trechos de imagens, não permitindo descartar algum ou supervalorizar outro. A visão multi-objetiva aliada ao caráter evolutivo dos algoritmos genéticos podem permitir uma seleção equilibrada sobre os três objetivos.

1.3 Estrutura da Monografia

Os próximos capítulos deverão obedecer a seguinte sequência de informações: o Capítulo 2 continuará apresentando o problema ambiental, mas enfocando sobre a aplicação de imagens de satélite para auxiliar no processo de controle do grau de devastação ambiental e dará também uma breve introdução sobre o algoritmo usado no processo de seleção de imagens.

No Capítulo 3 está descrito de forma detalhada o algoritmo adotado para solução do problema, suas etapas e principais características. No Capítulo 4 será apresentada a metodologia aplicada durante o projeto, e serão mencionadas as ferramentas utilizadas e suas características que as fizeram ser adotadas no projeto e sobre as métricas que serão adotadas no projeto para estudar o comportamento das soluções encontradas pelo algoritmo. O Capítulo 5 mostrará os resultados obtidos do Algoritmo Genético escolhido.

Fotografias Obtidas de Satélite e Segmentação de Imagens

Este capítulo irá explicar com mais detalhes o problema em questão, mostrar na seção 2.1 sobre a importância dos dados que foram extraídos das imagens de satélite para o estudo do problema ambiental, quais são esses dados e como são obtidos. Irá também contextualizar o Algoritmo Genético multiobjetivo NSGA II (*Non-dominated Sorting Genetic Algorithm II*) como mecanismo proposto para selecionar trechos de Mata Atlântica, levando em consideração os dados extraídos das imagens na seção 2.2.

2.1 Fotografias Obtidas de Satélite e Segmentação de Imagens

Como já foi mencionado no Capítulo 1, fotografias de satélites são bastante utilizadas como fonte de informação sobre o nível de devastação ambiental que um determinado ecossistema está sujeito. No entanto essas imagens precisam ser previamente processadas via software, tanto para que possíveis distorções possam ser corrigidas, como também para que características não perceptíveis ao campo visual humano possam ser reveladas.

Dessa forma, é possível disponibilizar ainda mais material de estudo sobre o problema da devastação ambiental, que além de ser um dos principais causadores do desaparecimento de espécies raras da natureza, também tem influenciado no problema do Aquecimento Global, visto que as florestas atuam como agentes reguladores do clima.

O Departamento de Botânica da Universidade Federal de Pernambuco (UFPE) vem desenvolvendo estudos através de fotografias tiradas por meio de satélite sobre a Mata Atlântica (ver Figura 2.1). Essas fotografias passaram pelo processo de segmentação

(técnica de Processamento de Imagens)[12] durante o qual cada pixel (a representação do menor elemento da imagem) é verificado a fim de que regiões com características distintas sejam localizadas e destacadas das demais (ver Figura 2.2).

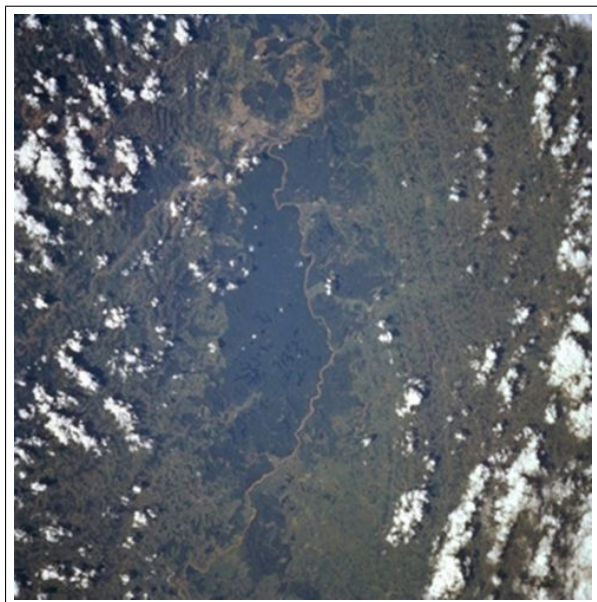


Figura 2.1 Exemplo de fotografia de trecho de Mata Atlântica obtida por satélite.

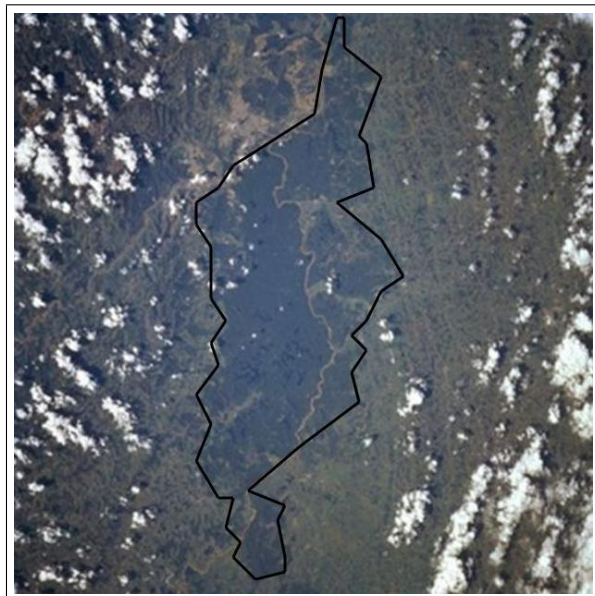


Figura 2.2 Fotografia segmentada de trecho de Mata Atlântica.

Pode-se ver que o procedimento da segmentação da imagem resultou na detecção de uma região de borda que delimita uma área chamada de fragmento da imagem (por se tratar de somente uma porção dela). E sabe-se que dentro do contexto da imagem inteira corresponde a um padrão que se destaca dentre as demais, separando a região preservada (área interna à borda) da região desmatada (área externa à borda).

Os fragmentos representam padrões, e pelo fato de existir grande similaridade entre seus pontos, será detectada pelo algoritmo de segmentação uma região de borda em que cada região dessa possui várias características que são aspectos importantes para a representação daquele fragmento.

Dentre os vários aspectos, os escolhidos para representação de cada um dos fragmentos foram a Área, a Forma e a Conectividade. A forma como se obtém os três parâmetros (Área, Forma e Conectividade) é bastante simples e será descrito a seguir:

- Primeiramente a área é obtida após a segmentação da imagem em polígonos e a área de cada um desses será calculada pela ferramenta ArcView 3.2[1] bastante utilizada em Geoprocessamento.
- O Índice de Forma é calculado como mostra a referência (2.1):

$$Forma = \frac{Perimetro}{200 \sqrt{(\pi Area)}}. \quad (2.1)$$

- A Conectividade é a porcentagem de cobertura dos outros fragmentos. É criada uma área de buffer, que é uma região que circunda o fragmento. No atual projeto, decidiu-se usar uma área de buffer com raio igual a 500 metros. A área de buffer engloba parte de outros fragmentos e é com essa área englobada que é calculada a porcentagem de conectividade do fragmento em relação aos outros ao seu redor, como pode-se visualizar na Figura 2.3.

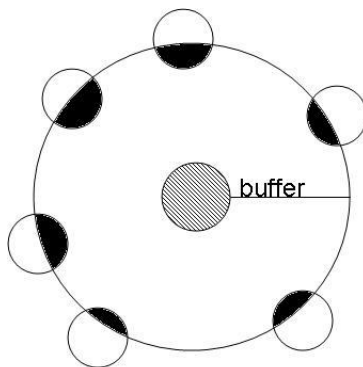


Figura 2.3 A figura demonstra como a conectividade é calculada.

2.2 Objetivos para Otimização

Durante esse trabalho será utilizado um algoritmo de busca, para procurar grupos de vinte fragmentos que se assemelham em relação a esses três parâmetros, ou seja, o objetivo é que a diferença entre eles chegue o mais próximo possível ao zero (caso que ocorreria com imagens iguais).

Quanto menor for a diferença entre Área (representada pelo símbolo Δ_{Area}), Forma (representada pelo símbolo Δ_{Forma}) e Conectividade (representada pelo símbolo $\Delta_{Conectividade}$) entre as imagens selecionadas, maior será a similaridade existente entre elas. Através de busca por similaridade, é que esses grupos de imagens devem ser selecionados.

Os valores que expressam similaridade que serão utilizados como objetivos no algoritmo de busca são: diferença de Forma, diferença de Área e diferença de Conectividade e são obtidas segundo as seguintes fórmulas (2.2), (2.3) e (2.4) nessa ordem:

$$\Delta_{Forma} = \sum_{i=0}^n \sum_{j=i+1}^n |Forma_{(i)} - Forma_{(j)}|, \quad (2.2)$$

$$\Delta_{Area} = \sum_{i=0}^n \sum_{j=i+1}^n |Area_{(i)} - Area_{(j)}|, \quad (2.3)$$

$$\Delta_{Conectividade} = \sum_{i=0}^n \sum_{j=i+1}^n |Conectividade_{(i)} - Conectividade_{(j)}|. \quad (2.4)$$

No qual n representa o número de imagens agrupadas e que constituem o indivíduo (no projeto atual n é igual a 20). Ou seja, cada ponto do indivíduo e que representa uma imagem será comparado um a um em relação aos seus aspectos considerados importantes e que os caracterizam.

O resultado dessas comparações serão armazenadas e serão úteis como objetivos pelo

Algoritmo Genético, guiando todo o processo de seleção. É utilizando esses três objetivos, que são obtidos a cada grupo de vinte fragmentos, que o Algoritmo Genético multiobjetivo NSGA II irá realizar suas buscas a fim de encontrar os melhores agrupamentos que minimizam cada um desses objetivos. O funcionamento detalhado do algoritmo irá ser demonstrado no próximo capítulo.

NSGA II - Non-dominated Sorting Genetic Algorithm II

Este capítulo visa apresentar o NSGA II. A primeira seção (seção 3.1) irá apresentar o Conceito de Dominância. Posteriormente, na seção 3.2, será realizada uma descrição sobre o funcionamento de um Algoritmo Genético clássico, sua utilização como algoritmo de busca, seus aspectos evolutivos que permitem a melhora dos resultados ao longo das gerações e suas principais etapas. Após isso, na seção 3.3, o algoritmo NSGA II, que usa em sua implementação o Critério de Dominância a fim de estabelecer o conceito de qualidade no seu espaço de soluções, será apresentado. Por fim, nas subseções 3.3.1 e 3.3.2 respectivamente, será descrito o funcionamento de dois processos importantes que o NSGA II utiliza para fazer a seleção das soluções, primeiramente o *Fast Non-Dominated Sorting* e depois o *Crowding Distance* (que é um tipo de Operador de Diversidade).

3.1 Dominância

Para proporcionar uma comparação mais justa entre os vários objetivos, foi implementado o conceito de Dominância [9] que é uma característica dos Algoritmos multi-objetivos. Essa classe de algoritmos que leva em consideração a otimização de mais de um objetivo (vários objetivos são minimizados ou maximizados dependendo do tipo de problema), sendo estes normalmente conflitantes.

O critério de dominância segue a seguinte regra :

Dado dois indivíduos p e q pertencentes a uma mesma população P , então:

- Um indivíduo p domina um indivíduo q , se no mínimo o valor em um dos objetivos

de p é melhor que o mesmo objetivo em q e o restante dos valores dos objetivos de p não podem ser piores que o restante dos mesmos valores nos objetivos em q . Ou seja, isso significa dizer que p não pode possuir nenhum objetivo com menos qualidade do que q .

Ao final de cada análise um determinado grupo de indivíduos são classificados como pertencentes a uma categoria específica denominada *front* e ao ser concluído o processo classificatório todos os indivíduos estarão inseridos em um dos n *fronts*.

O *front 1* é constituído de todas as soluções não-dominadas. O *front 2* pode ser conseguido considerando todas as soluções não-dominadas excluídas as soluções do *front 1*. Para determinação do *front 3*, exclui-se as soluções previamente classificadas no *front 1* e *2*, e assim por diante até que todos os indivíduos tenham sido classificados em algum *front*. A seguir, é apresentada uma descrição completa sobre o Algoritmo Genético clássico.

3.2 Algoritmos Genéticos

Algoritmos Genéticos foram introduzidos por J.H. Holland [13] e são baseados no mecanismo de seleção natural proposto por Charles Darwin.

Algoritmos Genéticos permitem que, a partir de um conjunto de soluções iniciais, com o passar das gerações (número de iterações do Algoritmo Genético), este possa atingir resultados ainda melhores do que os das gerações anteriores.

O fluxograma de um Algoritmo Genético Clássico nas simulações está apresentado na Figura 3.1.

Um Algoritmo Genético convencional, é constituído por quatro etapas: Geração de população inicial, Cruzamento, Mutação e Seleção. Através dos operadores de cruzamento, mutação e seleção, é possível melhorar os resultados entre uma geração e outra. São eles que atribuem ao algoritmo a capacidade de evoluir. Nas próximas subseções serão apresentados cada um desses operadores, como também sobre os dois métodos de convergência adotados no atual projeto.

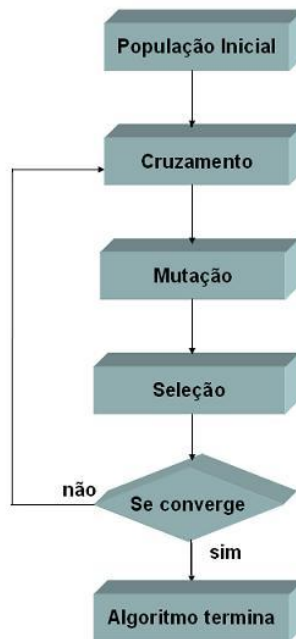


Figura 3.1 Diagrama de Blocos do Algoritmo Genético clássico.

3.2.1 Geração de população inicial

Diferentemente das outras etapas, a Geração de população inicial é realizada somente no começo do algoritmo e não se repete ao longo das próximas gerações. Indivíduos (ver exemplo de formato de Indivíduo com nove posições de imagens na Tabela 3.1) que codificam um grupo de n imagens distintas (isto significa que, não pode haver imagens repetidas em um mesmo indivíduo) são aleatoriamente gerados dentro do algoritmo e armazenados em uma estrutura de dados.

Após a geração de cada indivíduo, é verificado se aquele já existe na população. Isso serve para evitar a ocorrência de indivíduos repetidos, pois é importante para o bom funcionamento do algoritmo que haja diversidade de soluções. Isto permite uma maior variabilidade genética durante as gerações e pode contribuir para que soluções melhores sejam encontradas.

Se um indivíduo atualmente gerado já existe na população, ele será ignorado e um novo indivíduo será criado a fim de se completar o número de indivíduos da população

inicial previamente especificado. Foi adotado o formato inteiro para representar cada elemento do indivíduo.

Tabela 3.1 *Exemplo de indivíduo formado de nove imagens representadas por valores inteiros, que codificam 9 imagens de satélite.*

indivíduo	57 335 416 446 525 550 583 602 674
-----------	------------------------------------

3.2.2 Operador de Cruzamento

Os indivíduos existentes na população, respeitando uma probabilidade inicialmente imposta, irão trocar sequências, aleatoriamente escolhidas, de suas informações gerando novos indivíduos que herdaram características dos indivíduos anteriores. Bastante semelhante à união cromossômica para formar novas combinações de genes.

É importante ressaltar que indivíduos repetidos não serão tolerados dentro da População, ou seja, caso o indivíduo resultante do Operador de Cruzamento já existir ele será descartado e não será inserido entre os indivíduos válidos. Esta restrição é importante, pois ajuda a manter a diversidade das soluções.

Também não será permitido que indivíduos contenham imagens repetidas dentro de sua representação de números inteiros, dessa forma um indivíduo gerado através do Operador de Cruzamento deverá também ser representado por uma sequência de números inteiros distintos. Um simples exemplo de cruzamento entre indivíduos com 9 posições cromossômicas é demonstrado na Tabela 3.2. Este tipo de cruzamento é chamado de cruzamento de 1 ponto.

Tabela 3.2 *Exemplificação da etapa de cruzamento.*

indivíduo 1	<u>57 335 416 446</u> 594 597 750 880 883
indivíduo 2	7 145 251 335 <u>525 550 583 602 674</u>
novo indivíduo	57 335 416 446 525 550 583 602 674

Perceba que o novo indivíduo é a combinação das quatro primeiras imagens do indivi-

duo 1 com as cinco ultimas imagens do indivíduo 2. A junção das duas sequências darão origem a um novo indivíduo, o qual herdará características dos seus pais (os indivíduos 1 e 2).

3.2.3 Operador de Mutação

A operação de mutação não agrega novos indivíduos à população, ou seja, não aumenta numericamente a população como ocorre por intermédio do operador de cruzamento, mas ele apenas modifica esses indivíduos que já existem transformando-os em outros indivíduos diferentes. Cada indivíduo, obedecendo a uma probabilidade inicialmente imposta, irão em alguns pontos de sua sequência mudar de valor.

Mais uma vez, é importante ressaltar que indivíduos repetidos não serão tolerados dentro da População, ou seja, caso o indivíduo resultante do Operador de Mutação já exista ele será descartado e não será inserido entre os indivíduos válidos, restrição importante pois ajuda a manter a diversidade das soluções.

Também não será permitido que indivíduos contenham imagens repetidas dentro de sua representação de números inteiros, dessa forma um indivíduo gerado através do Operador de Mutação deverá também ser representado por uma sequência de números inteiros distintos. No caso em que cada indivíduo está sendo tratado como uma sequência de inteiros, poderá ocorrer uma mudança aleatória sobre algumas posições como visto no exemplo da Tabela 3.3.

Tabela 3.3 *Exemplificação da etapa de mutação.*

indivíduo original	57 335 <u>416</u> 446 525 <u>550</u> 583 <u>602</u> 674
indivíduo alterado	57 335 348 446 525 535 583 600 674

Isso fará com que o algoritmo consiga uma maior diversidade genética que a função de cruzamento sozinha não seria capaz de fornecer. O operador de mutação permite uma variabilidade genética maior e uma maior exploração do universo de possibilidades de configurações. Além do que evita que o algoritmo fique estacionado em mínimos locais.

Evitar mínimos locais seria a maior vantagem do operador de mutação, pois ele permite uma variação brusca das características que o indivíduo possui através da alteração de partes aleatoriamente selecionadas dos seus genes.

3.2.4 Operador de Seleção

Esta é a etapa mais importante de um GA, pois será nessa fase que serão utilizada uma função objetiva que irá dar ao algoritmo padrões para selecionar os melhores indivíduos. A função objetiva é essencial para propagar as gerações futuras os melhores resultados de cada geração. Neste caso, somente as melhores soluções continuarão a existir na próxima geração. Nesse trabalho será considerado o processo de seleção usado no NSGA II [14].

3.2.5 Convergência

Foram dois métodos adotados no atual projeto, o primeiro deles e o mais simples utilizava quantidade de gerações como condição de parada. Dessa forma, quando o algoritmo alcança determinado número de gerações, o mesmo finalizava a sua execução.

Já o segundo método de convergência, verificava se houve uma alteração significativa em cima dos indivíduos da população entre n gerações. Se entre as gerações não for detectada uma variação de qualidade significativa, então existe a indicação de convergência, e o algoritmo é finalizado. Caso contrário (se a taxa de variação calculada pelo algoritmo supera a taxa de variação percentual inicialmente pré-definida), o algoritmo segue sua execução normal em direção a uma nova geração. O método de convergência será verificado novamente durante as próximas gerações até que o algoritmo demonstre que verdadeiramente está convergindo sobre uma determinada solução.

3.3 NSGA II

O algoritmo escolhido foi o NSGA II (*Non-dominated Sorting Genetic Algorithm II*) que é um algoritmo multiobjetivo, baseado em Algoritmos Genéticos [15] e que implementa o conceito de Dominância, ou seja, classificar a População Total em *fronts* de acordo com o grau de dominância.

Segundo o NSGA II, os indivíduos que estão localizados no primeiro *front* são considerados as melhores soluções daquela geração, enquanto que no último *front* encontram-se as piores. Usando esse conceito, pode-se encontrar resultados mais consistentes (pontos mais próximos da região de Pareto) e que se adaptam melhor ao tipo do problema. Alguns aspectos são importantes na solução de problemas multiobjetivos:

- Dividir a população em diferentes níveis (*fronts*), utilizando critério de Dominância.
- Indivíduos do *front* n são melhores do que indivíduos do *front* $n+1$.

Através do critério de dominância, o algoritmo agrega o conceito de Elitismo que classifica a população total em diferentes categorias de qualidade ao invés de tratá-las como pertencentes a um único grupo. Isso permite ao algoritmo a priorização daqueles que foram melhores classificados.

O funcionamento do NSGA II se destaca por possuir dois mecanismos importantes no processo de Seleção, são eles: o *Fast Non-Dominated Sorting* e o *Crowding Distance*. O fluxograma da Figura 3.2 descreve o funcionamento da seleção do algoritmo NSGA II.

O que existe inicialmente é uma População Inteira ainda não classificada, que irá passar por um processo em que serão atribuídos a cada indivíduo um grau de dominância em relação a todos os outros indivíduos da População Inteira. Isso é obtido comparando uns com os outros e assim classificando-os de acordo com o Critério de Dominância descrito anteriormente.

Após os valores de dominância terem sido atribuídos a todos da população, esses indivíduos serão em seguida classificados em *fronts* de acordo com os seus valores de dominância. Os melhores indivíduos são classificados no primeiro *front*, e os piores são

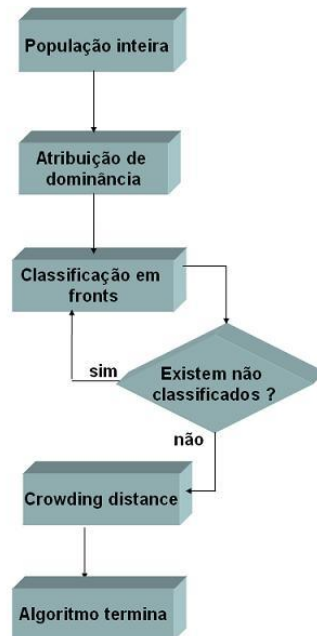


Figura 3.2 Diagrama de Blocos do Algoritmo NSGA II, descrevendo o funcionamento da etapa de seleção.

classificados como pertencentes ao último *front*. Essa etapa segue até que não haja mais indivíduos a serem classificados.

Após todos os indivíduos estarem classificados dentro de um *front*, eles irão ser classificados pelo Operador de Diversidade usado neste trabalho, ou seja o Crowding Distance. Esse operador irá ordenar cada indivíduo de acordo com a sua distância em relação aos pontos vizinhos no mesmo *front*(em relação a cada objetivo).

Quanto mais distantes desse ponto central, maior a probabilidade de serem selecionados. Esse operador permite que haja um melhor espalhamento dos resultados, evitando-se assim aglomerações de soluções sobre um mesmo ponto.

Nas próximas subseções (3.3.1 e 3.3.2) serão essas etapas serão descritas com mais detalhes.

3.3.1 *Fast Non-Dominated Sorting:*

O algoritmo de seleção *Fast Non-Dominated Sorting* é executado em duas etapas, a primeira delas será referenciada neste trabalho como sendo o Processo 1 (*Pseudo-código do Processo 1 na Tabela 3.4*) e a segunda delas como o Processo 2 (*Pseudo-código do Processo 2 na Tabela 3.5*).

Em geral, o Processo 1 na Tabela 3.4 irá analisar todos os indivíduos da População total P , comparando-os uns com os outros para classificá-los de acordo com o grau de dominância np (número de indivíduos que dominam p , em que p é um indivíduo da população P). Dessa forma se um indivíduo p é dominado por um número x de indivíduos da população total P , o seu valor correspondente de np é igual a x .

Se ao final do Processo 1, o indivíduo possuir o valor de np igual a 0, significa dizer que esse indivíduo não é dominado por ninguém dentro da População Total P e que tais indivíduos farão parte do primeiro *front*, no qual estão os melhores indivíduos de toda a população atual.

O Processo 2, irá separar cada indivíduo em diferentes categorias (*os fronts*) de acordo com os seus valores de dominância, indicados pelos seus respectivos valores de np . Cada indivíduo incluído em um dos fronts são retirados totalmente do contexto do sistema, decrementando os valores de np de cada indivíduo dominado por esses. Isso repete-se até que não sobrem mais indivíduos na população restante.

Observando passo-a-passo o funcionamento do pseudo-código do Processo 1 na Tabela 3.4, é selecionado um indivíduo p da população P (linha 1) e para cada indivíduo q restante na população P (linha 2) é verificado se o indivíduo p domina o indivíduo q (linha 3), caso seja verdade o indivíduo q será armazenado em Sp (na lista dos indivíduos dominados por p) (linha 4). Caso contrário, se q dominar p (linha 5) o valor de np é incrementado (pois np é um contador de quantos indivíduos dominam p) (linha 6).

Logo em seguida é testado se o np é igual a 0 (linha 7), se for verdade isso significará que p não foi dominado por ninguém da população o que quer dizer que ele irá fazer parte do primeiro *front* (linha 8).

Tabela 3.4 Pseudo-código Processo 1 do fast non-dominated sorting.

1.	para cada p (pertencente a) P
2.	para cada q (pertencente a) P
3.	se (p domina q) então
4.	Armazena q em S_p ($S_p \leftarrow q$)
5.	senão se (q domina p) então
6.	Incrementa np ($np \leftarrow np+1$)
7.	se np (é igual a) 0 então
8.	Armazena p em $F1$ ($F1 \leftarrow p$)

Como foi anteriormente descrito, o *fast-nondominated-sorting* irá comparar indivíduo a indivíduo para poder determinar o grau de dominância np de cada componente da população. Ou seja, quanto menores os valores de np encontrados, melhores soluções irão representar.

Analisando-se agora o funcionamento do Processo 2 na Tabela 3.5, é observada uma variável i (que funcionará como um contador para o número de *fronts*) a qual é inicializada com o valor unitário (linha 1), enquanto houver novos *fronts* sendo gerados (ou seja se o tamanho de F_i (o atual *front*) for diferente de 0) o Processo 2 será executado (linha 2). A variável H é inicializada (que será um armazenador temporário dos próximos *fronts*) com o valor 0 (linha 3).

Nas linhas 4 e 5 um indivíduo p do *front* F_i é selecionado, e uma verificação é realizada a fim de se achar quais os indivíduos em q dominados por p (que se encontram na lista S_p) e os valores dos nq desses indivíduos são decrementados com o objetivo de se tirar do contexto os indivíduos que já foram classificados no *front* anterior ao atual.

O decréscimo dos valores de nq tem também como objetivo de que esses se aproximem cada vez mais do valor 0, pois tornar esses valores iguais a 0 significará que em uma próxima iteração tais indivíduos com nq igual a 0 farão parte do novo *front* (linhas 6 à 7).

Posteriormente na linha 8 o valor de i será incrementado e os valores armazenados em H serão copiados para o novo F_i (que representa o novo *front*) (linha 9).

Tabela 3.5 Pseudo-código Processo 2 do fast non-dominated sorting.

1.	Inicializa uma variável i com o valor 1 ($i < -1$)
2.	enquanto F_i (for diferente de) 0
3.	$H = 0$
4.	para cada p (pertencente a) F_i
5.	para cada q (pertencente a) S_p
6.	decrementa o valor de nq ($nq < - nq-1$)
7.	se nq (é igual a) 0 então $H < - q$ (insere em H o valor de q)
8.	Incrementa o valor de i em uma unidade ($i < - i+1$)
9.	Armazena os valores de H no F_i (que será o <i>front</i> atual no próximo loop)

3.3.2 Crowding distance

O *crowding distance* é o Operador de Diversidade usado no NSGA II, a fim de se garantir um maior espalhamento dos resultados ao longo da linha de pareto. Evita-se assim a concentração de soluções em cima de um mesmo ponto ou região, como também é utilizado como método de ordenação dos indivíduos dentro de um mesmo *front*. O *Crowding Distance* utiliza como métrica a distância de cada indivíduo aos indivíduos mais próximos.

O algoritmo de *Crowding Distance* calcula a distância média entre um ponto central i selecionado dentro da população e dois pontos localizados nas extremidades do ponto central ($i-1$) e ($i+1$). A idéia é que a partir de um ponto central o Operador de Diversidade possa encontrar pontos extremos e priorizar os pontos mais distantes durante o processo de seleção a fim de espalhar os resultados ao longo do pareto. A disposição dos pontos extremos formam um cubóide em relação ao ponto central (Ver Figura 3.3).

A Tabela 3.6 mostra o pseudo-código do algoritmo Crowding Distance. Antes de tudo (linha 3) é criada uma variável l que servirá para armazenar a quantidade de valores existentes no conjunto de entrada S . Posteriormente, são inicializados todos os valores do conjunto $S[i]_{distancia}$ (que é o vetor que irá conter todas as distâncias de cada indivíduo do conjunto S) com o valor 0 (linha 4). O Operador de Diversidade irá levar em consideração os m objetivos em questão, ordenando em relação a cada um deles (linha 6).

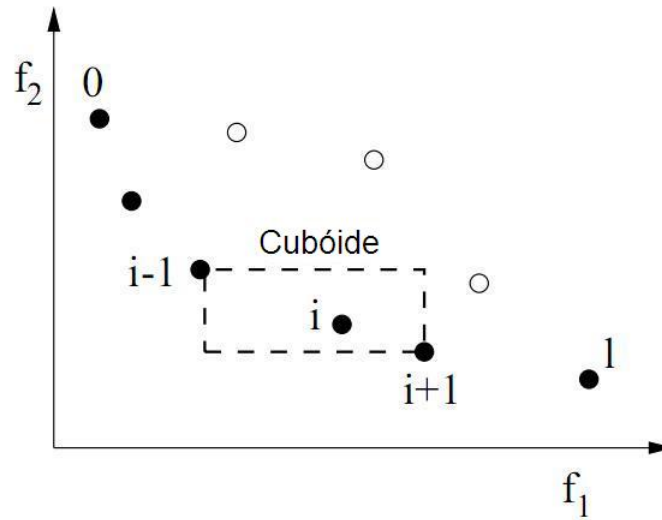


Figura 3.3 A Figura mostra qual o conceito do cálculo da distância feito pelo Operador de Diversidade.

Depois são inicializadas as posições $S[0]_{distancia}$ e a ultima posição $S[l]_{distancia}$ do vetor distância com um valor muito alto (linha 7), isso é uma estratégia para que esses indivíduos que se situam nas extremidades sejam sempre selecionados (visto que o *crowding distance* escolhe sempre aqueles que possuem os maiores valores de distância). A partir da linha 8 existe um laço que irá chamar o cálculo do valor médio dos pontos em relação ao ponto central i .

Segundo a fórmula da linha 9, $S[i+1].m$ e $S[i-1].m$ significa respectivamente a posição $i+1$ e $i-1$ do vetor S em relação ao objetivo m . No caso do atual projeto são os três objetivos minimizar (ou seja $m = 3$) diferenças de Área, Forma e Conectividade.

Ao final do *Crowding Distance*, os resultados serão ordenados dando preferência às soluções com distâncias médias maiores. O algoritmo visa encontrar soluções diversificadas e que não se encontrem concentradas em uma região limitada do gráfico, e que permita ao tomador de decisão ter uma ampla visão sobre as melhores soluções que ele poderá aplicar para gerar soluções.

A Fórmula (3.1) é usada para calcular a distância de cada um dos indivíduos em relação ao ponto médio (isso levando-se em consideração cada um dos m objetivos no caso: Similaridade de Forma, Similaridade de Área e Similaridade de Conectividade).

Tabela 3.6 Pseudo-código do algoritmo *Crowding Distance*.

1.	distância de cada ponto no conjunto S :
2.	atribuição do crowding-distance(S)
3.	$l= S $ ($ S $ representa o número de elementos do conjunto S e que será atribuído à l)
4.	para cada i , do conjunto S [i] $_{distancia} = 0$
5.	para cada objetivo m :
6.	ordenar(S, m) (ordena cada indivíduo do grupo S em relação a cada objetivo m)
7.	$S[0]_{distancia}$ e $S[l]_{distancia} <-$ valor alto
8.	para $i = 1$ até $(l-1)$ (para todos os outros pontos)
9.	$S[i]_{distancia} = S[i]_{distancia} + (S[i+1].m - S[i-1].m)$

A distância atual $d_{I_j^m}$, que significa a distância do j -ésimo elemento do conjunto I em relação ao objetivo m . A distância será somada com a razão da diferença dos valores consecutivos dos objetivos, correspondentes aos elementos das posições $j+1$ e $j-1$ do conjunto I , que são valores oriundos do m -ésimo objetivo, e representados por $f_m^{(I_{j+1}^m)}$ e $f_m^{(I_{j-1}^m)}$. E esse valor dividido pela diferença entre seus objetivos máximo e mínimo também pertencentes ao m -ésimo objetivo, ou seja, $f_m^{max} - f_m^{min}$. A fórmula descrita está abaixo representada.

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{(I_{j+1}^m)} - f_m^{(I_{j-1}^m)}}{f_m^{max} - f_m^{min}}. \quad (3.1)$$

Esse procedimento será repetido dependendo do número de objetivos existentes no problema, ou seja m vezes, isso com a finalidade de contemplar todos os aspectos considerados importantes durante a busca das soluções.

Ferramenta para Seleção de Imagens de Trechos de Mata Atlântica

Este capítulo irá descrever, na seção 4.1, sobre as técnicas utilizadas durante a estruturação e implementação do projeto, as ferramentas adotadas e suas características. Logo após, na seção 4.2, falará especificamente sobre a ferramenta que implementa o algoritmo NSGA II, suas características de como ela funciona, seus parâmetros de entrada e como são exibidos os resultados na saída. Finalmente, na seção 4.3, serão descritas as métricas de avaliação do Pareto, as quais serão importantes para analisar o comportamento evolutivo do algoritmo ao longo das gerações.

4.1 Técnicas Utilizadas

Primeiramente foi feita uma análise a respeito do problema em questão e conhecer qual a motivação que torna o tema algo a ser considerado importante. Para isso foram marcadas reuniões com membros do Departamento de Botânica da UFPE (Universidade Federal de Pernambuco), nas quais informações úteis para a implementação do projeto puderam ser compartilhadas. Informações como características das imagens fotografadas pelo satélite e quais os dados relevantes no processo de seleção das imagens (Área, Forma e Conectividade), o que cada um representa e como são obtidos.

Dentre os Algoritmos Genéticos, o escolhido foi o NSGA II (*Nondominated Sorting Genetic Algorithm*) por apresentar caráter Elitista e Multiobjetivo, ou seja, considera mais de um aspecto na busca das melhores soluções, sendo eles considerados igualmente importantes. O que se enquadra dentro da realidade do projeto em questão, pois trata-se

de um problema com três objetivos.

A linguagem de programação adotada na implementação da ferramenta foi Java [10], pois como ela incorpora a idéia da Orientação a Objetos permitiu que o código fosse melhor modularizado e organizado, como também pelo fato de Java possuir estruturas já disponíveis (mecanismos de inserção, remoção e consultas a listas, vetores e etc) que puderam agilizar a implementação dos conceitos. Como plataforma de desenvolvimento foi adotada a ferramenta Eclipse [2]. O Eclipse foi escolhido como IDE (*Integrated Development Environment*) por ser gratuito, compatível com várias tecnologias e exaustivamente documentado.

Por se tratar de um problema com três objetivos, uma ferramenta gráfica foi necessária para representar as soluções de forma tridimensional. Foi realizada uma pesquisa sobre qual seria a ferramenta mais adequada ao projeto. Uma das opções foi o JfreeChart [4] que apesar de ser bastante adaptável ao Java, não foi compatível a representação em mais de duas dimensões.

A segunda opção foi a ferramenta gráfica Gnuplot (ver Figura 4.1) [3], que é uma ferramenta de código aberto e que permite que por meio de scripts e linhas de comando, a ferramenta possa se comunicar com a interface gráfica do Gnuplot. Os dados foram salvos em um arquivo do tipo *.txt* (texto) e um script foi criado a fim de que após a última iteração do algoritmo o script fosse finalmente executado e os gráficos 3D exibidos na tela, como mostra a Figura 4.2.

Tanto os gráficos quanto os relatórios são importantes no projeto para serem feitas as análises dos paretos, e assim observar se o aplicativo está funcionando ou não da forma esperada. Em ambos deve ficar visível que está havendo uma melhora das soluções entre uma geração e outra, isso significa que os resultados estejam caminhando em direção a região denominada de pareto, onde localizam-se as soluções não-dominadas (as melhores soluções).

Pode-se ver como exemplo a Figura 4.3 que mostra uma representação gráfica de um conjunto de soluções encontradas pela ferramenta, mostrando em suas coordenadas os três objetivos usados durante o projeto e que já foram descritos no Capítulo 2.

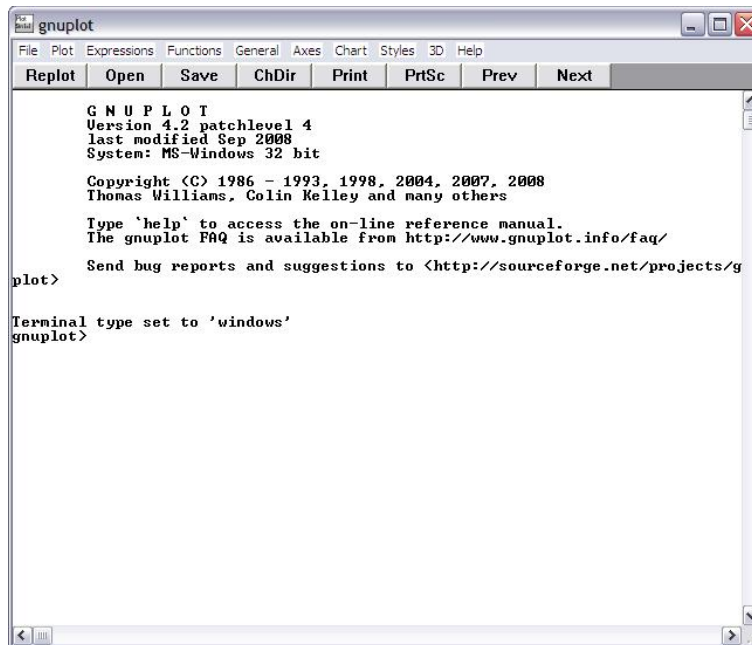


Figura 4.1 A Figura mostra a interface da ferramenta gráfica Gnuplot.

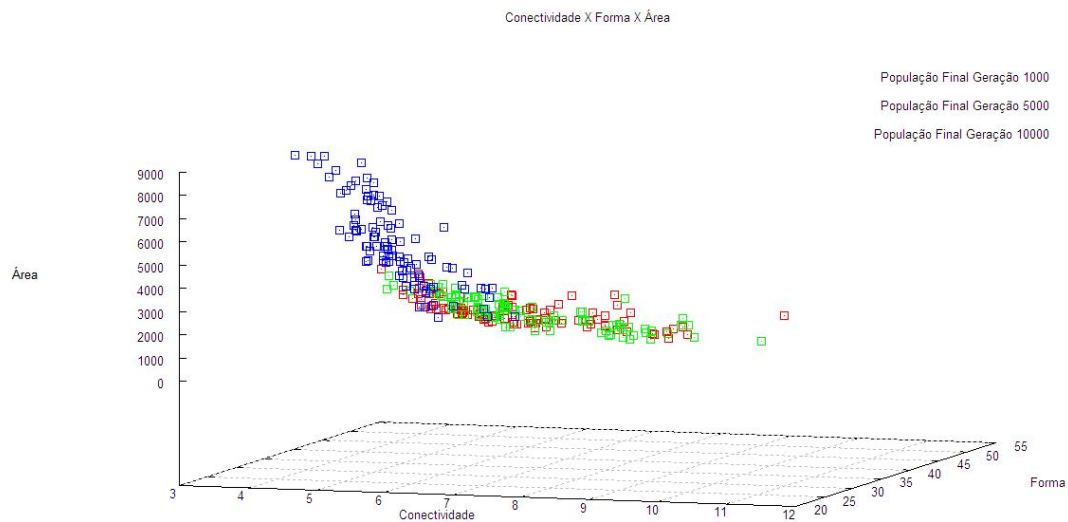


Figura 4.2 A Figura mostra um resultado gráfico obtido pela ferramenta Gnuplot.

4.2 Uma Ferramenta inteligente usando NSGA II

Foi criada uma ferramenta que recebe como parâmetros de entrada o conjunto dos dados sobre cada imagem contendo os valores de Conectividade, Área e forma, esses va-

lores irão alimentar a ferramenta que usará essas informações para selecionar as melhores imagens utilizando como critério a similaridade existente entre elas.

Observando a Figura 5.1 pode-se ver a interface gráfica da ferramenta inteligente.

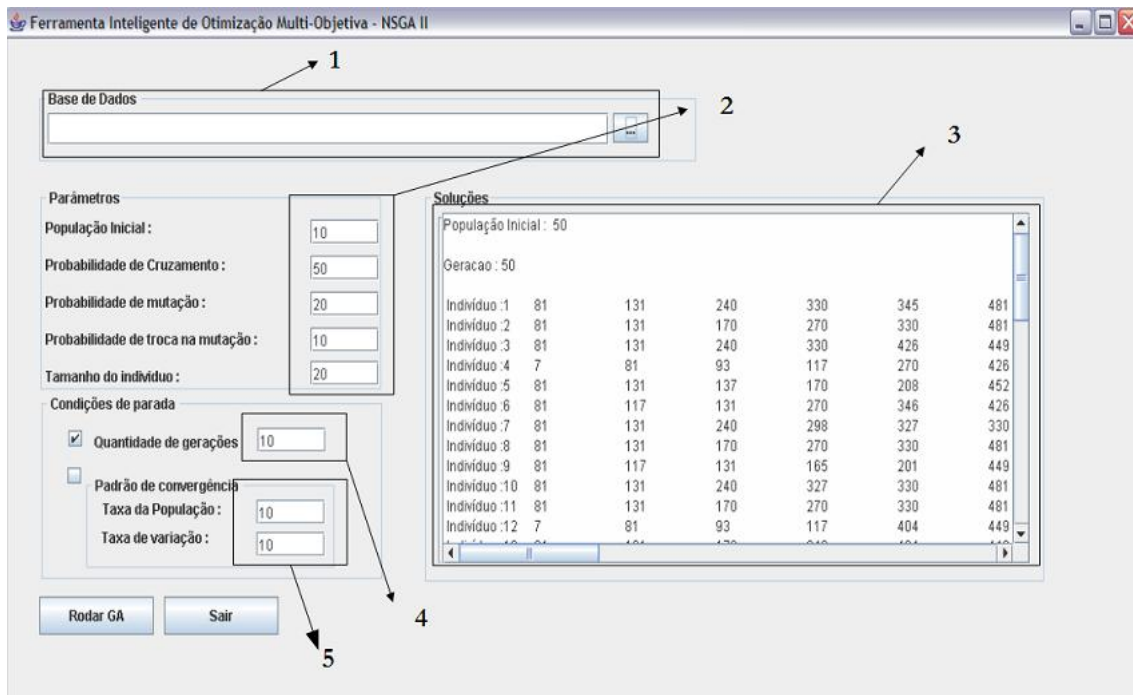


Figura 4.3 Interface gráfica da ferramenta inteligente que implementa o algoritmo NSGA II.

1. Entrada de Dados:

É na entrada de Dados que serão colocados os dados oriundos da segmentação das imagens dos trechos de Mata Atlântico contendo todos os dados que compõem o espaço total busca.

2. Parâmetros de Entrada do algoritmo de busca

- População Inicial;
- Probabilidade de Cruzamento;
- Probabilidade de Mutação;
- Probabilidade de troca na mutação;
- Tamanho do Indivíduo;

3. Tela de soluções:

Nessa área estarão todos os melhores resultados ao fim de cada simulação.

4. Quantidade de Gerações: Nessa área, é definida a quantidade de gerações do algoritmo, ou seja a quantidade de iterações que o algoritmo irá passar antes de terminar. Irá funcionar como condição de parada, caso o padrão de convergência não for acionado.

5. Critério de convergência:

Caso não seja especificado uma certa quantidade de gerações, um mecanismo de convergência está presente para testar se houve alguma mudança significativa da qualidade da população. Se não ocorrer esta mudança significativa o algoritmo termina, ou seja, converge. No entanto, se houver uma alteração significativa da população, o algoritmo roda mais uma geração e a convergência é dessa forma testada sempre entre uma geração e outra.

A Figura 4.4 mostra a interface da Ferramenta Inteligente executando um exemplo de simulação com 1000 gerações e com uma população inicial de 100 indivíduos.

Nesse exemplo a probabilidade de cruzamento é de 50%, probabilidade de mutação de 20%, probabilidade de troca na mutação de 10% e tamanho do indivíduo composto de 20 fragmentos de imagens de satélite, ou seja cada indivíduo é constituído de uma sequência de n imagens que deve ser especificada antes da simulação na própria interface da ferramenta.

Ao final da simulação o gráfico contendo as melhores soluções será exibido para ser analisado, e a partir das análises poder ser escolhido o melhor ponto (que representa uma sequência de vinte imagens de trechos de Mata Atlântica). (Ver Figura 4.5)

Uma característica importante dos Algoritmos Genéticos é que a medida que se aumenta a quantidade de gerações e a População Inicial, maior a probabilidade de se alcançar soluções cada vez mais próximas ao pareto, ou seja região onde estão localizadas as soluções não-dominadas e que representam os melhores resultados. Se for aumentada a

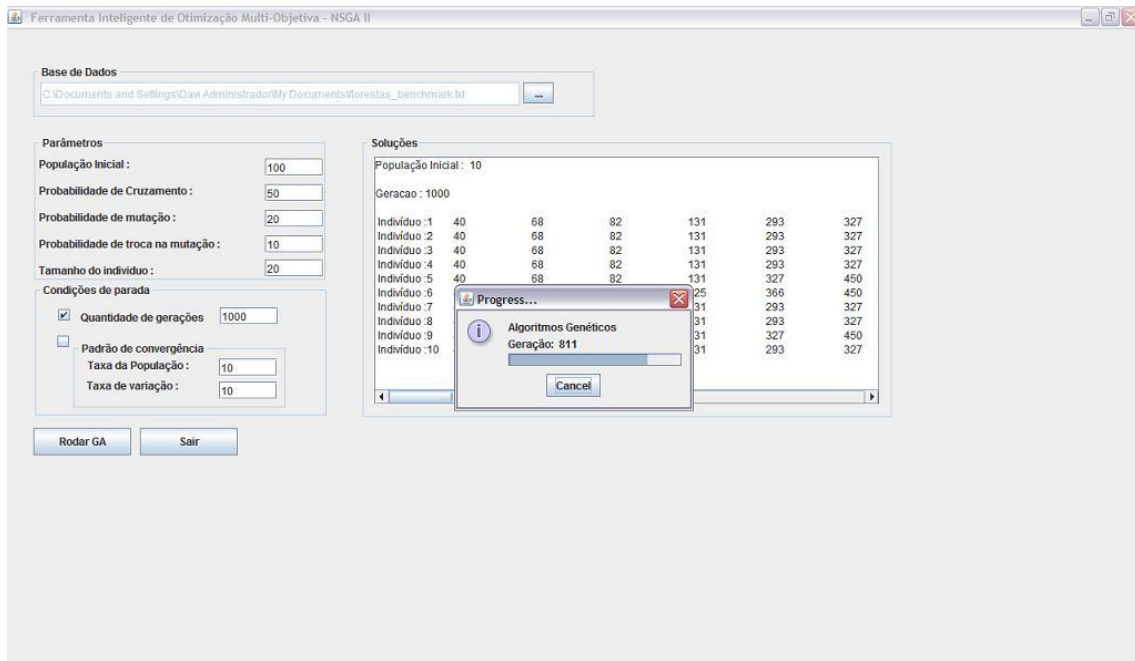


Figura 4.4 Ferramenta Inteligente simulando 1000 gerações e com uma população inicial de 100 indivíduos.

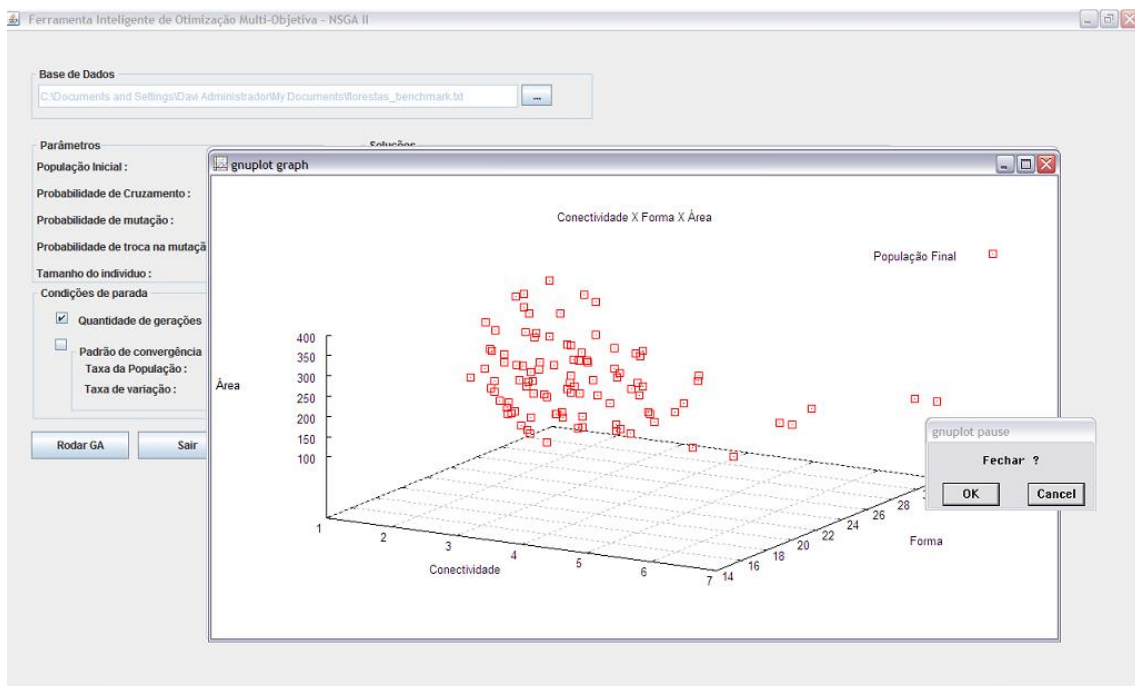


Figura 4.5 Resultados gráficos que serão usados para análise das soluções obtidas pelo algoritmo. Através dos gráficos pode-se selecionar qual o ponto ou conjunto de pontos que melhor soluciona o problema em questão.

quantidade de indivíduos da População Inicial, aumenta-se também as possíveis combinações para formação de novos indivíduos durante a ação do operador de Cruzamento. Se for aumentada a quantidade de gerações, o algoritmo terá mais oportunidades de modificar os membros de sua População e de selecionar indivíduos melhores do que os indivíduos da geração anterior.

4.3 Métricas para Avaliação de Pareto

4.3.1 HyperVolume

Métrica proposta por Zitzler e Thiele [16]. O hypervolume calcula a área (para o caso de problemas com dois objetivos) ou o volume (para o caso de problemas com três ou mais objetivos). Para problemas com dois objetivos será calculada uma área retangular limitada por uma origem e pelos objetivos $(f_1(\vec{x}); f_2(\vec{x}))$ pertencentes ao vetor de soluções.

4.3.2 Maximum Spread

Maximum Spread: Métrica proposta pelo Zitzler et. al. [11] apresenta um valor que representa a máxima extensão entre as soluções mais distantes no conjunto não-dominado encontrados. Em um problema com dois objetivos, o valor será a distância euclidiana entre as duas maiores soluções.

$$MS = \sqrt{\sum_{m=1}^M (\max_{i=1}^n f_i^n - \min_{i=1}^m f_i^m)^2} \quad (4.1)$$

Lembrando que n é o número de soluções no conjunto não-dominado, e M é o número de objetivos em um dado problema. Nesta métrica um maior valor indica uma melhor performance.

4.3.3 Spacing

Técnica usada para medir a propagação (distribuição) dos valores ao longo de todos os valores não-dominados encontrados até o momento. Uma vez que o "início" e o "final" do atual Pareto front encontrado são conhecidos, uma métrica indica o quanto as soluções se encontram bem distribuídas em tal front. Foi proposta uma tal medida do alcance (distância) de variância das soluções vizinhas entre as soluções não-dominadas encontradas até o momento. Esta métrica é definida como:

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2}. \quad (4.2)$$

Em que $d_i = \min_j (|f_1^i - f_1^j| + |f_2^i - f_2^j|)$, como também $i, j=1, \dots, n$, e que \bar{d} é a média de todos os d_i , e n é o número nondominated de soluções encontrados até o momento. Um valor de zero para esta métrica indica todos os membros do Pareto front atualmente disponível estão equidistantemente espaçados.

CAPÍTULO 5

Resultados

Este capítulo irá mostrar os resultados obtidos através de métricas para avaliação do Pareto, o que é muito importante para certificar matematicamente de que o algoritmo está funcionando ou não de forma correta. As métricas já foram apresentadas anteriormente no final do Capítulo 4.

Após rodar as simulações é necessário fazer uma análise dos paretos para certificar que as soluções do algoritmo genético tendem com o passar das gerações se aproximar mais da Região de Pareto onde ficam as soluções não-dominadas, ou seja as melhores soluções.

Para assegurar uma análise mais adequada das soluções, foram feitas 30 simulações de 20000 gerações cada uma, com uma população inicial de 100 indivíduos, probabilidade de Cruzamento de 50%, probabilidade de Mutação de 20% e probabilidade de troca na Mutação de 10% (parâmetros que já foram previamente explicados anteriormente), foram coletados os paretos obtidos das gerações 10, 100, 200, 250, 5000, 10000, 15000 e 30000.

A partir desses paretos foram calculadas as métricas de *Hypervolume*, *Maximum Spread* e *Spacing* cujos valores de média e desvio padrão, entre parêntesis, estão a mostra na Tabela 5.1. O objetivo é mostrar que os dados variam entre uma geração e outra e daí através das variações dessas métricas se poderá observar a evolução dos dados com o passar das gerações.

Pode-se ver na Figura 5.5 que verdadeiramente existe essa tendência de melhora das soluções ao longo das gerações. Essa é a principal característica de um Algoritmo Genético, melhorar a cada geração, isso significa que normalmente as soluções da geração $n+1$ seriam melhores do que as soluções da geração n . Contudo, pode ocorrer de que entre uma geração e outra os resultados possam piorar. Isso deve-se ao Operador de Mutação que pode transformar bons indivíduos em indivíduos de qualidade inferior aos da geração

Tabela 5.1 *Resultados das métricas usadas para avaliação do Pareto.*

Geração	Hypervolume	Spacing	Maximum Spreading
10	0.49 (0.02)	241.14 (133.13)	10046.69 (1747.18)
50	0.77 (0.02)	125.41 (94.34)	7391.23 (1875.83)
100	0.81 (0.02)	70.17 (49.77)	6231.73 (1781.21)
250	0.82 (0.02)	66.51 (68.4)	4671.11 (1438.31)
5000	0.83 (0.02)	61.99 (52.58)	6410.58 (1794.59)
10000	0.84 (0.02)	47.58 (24.56)	5976.86 (1311.22)
15000	0.84 (0.01)	80.48 (69.2)	6752.34 (1678.06)
20000	0.83 (0.02)	60.74 (30.81)	6650.24 (1116.93)

anterior.

Como o problema é de minimização as melhores soluções devem se encontrar próximas a origem dos eixos X, Y e Z , que representam diferença de Forma, diferença de Conectividade e diferença de Área.

Nota-se visualizando os gráficos das Figuras 5.1, 5.2, 5.3 e 5.4 uma melhora em relação a uma geração e outra, ou seja, quanto maior a geração mais próximas serão as soluções do Pareto front.

Percebe-se na Figura 5.5 que com 10000 gerações obtém-se resultados melhores do que com 10, 50 e 100 gerações, isso deve-se ao caráter evolutivo do Algoritmo Genético NSGA II. Essa tendência de melhora das soluções durante as simulações permanece até o algoritmo convergir sobre um determinado pareto, chamado de pareto ótimo. Ou seja, vai haver a partir de uma determinada geração uma acomodação desses resultados sobre um determinado espaço de soluções, não importa quantas gerações ainda estejam por vir. Isso porque as soluções do pareto ótimo são as melhores que podem ser alcançadas dentro do problema. Deve-se também tomar cuidado pois o motivo de tal acomodação

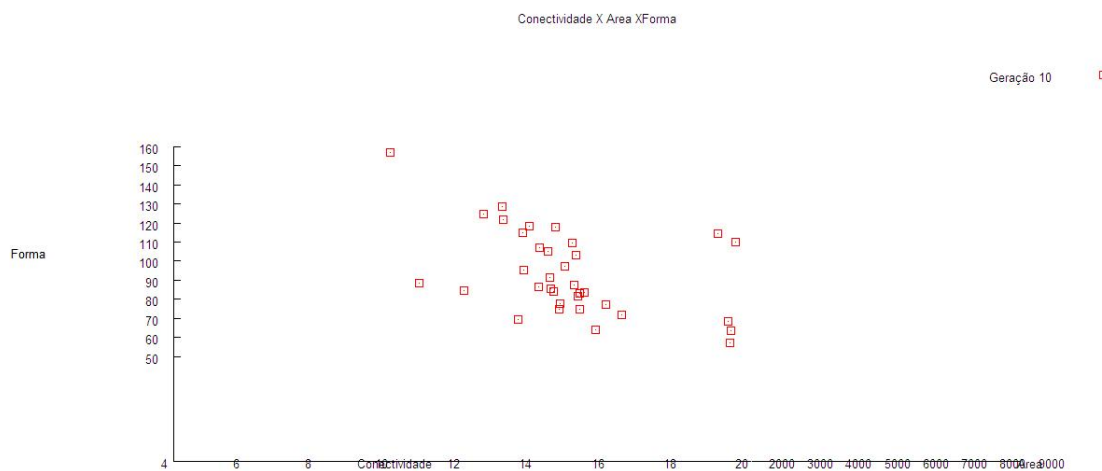


Figura 5.1 Simulação com 10 gerações e com população inicial de 100 indivíduos.

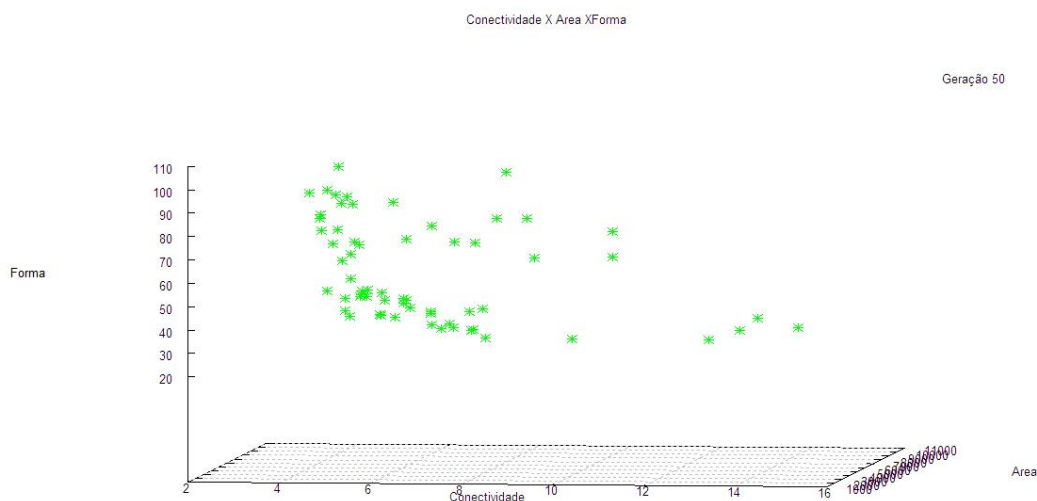


Figura 5.2 Simulação com 50 gerações e com população inicial de 100 indivíduos.

pode ser pelo fato de haver mínimos locais, daí deve-se variar os parâmetros de entrada de Probabilidade de Cruzamento e principalmente os parâmetros de Probabilidade de

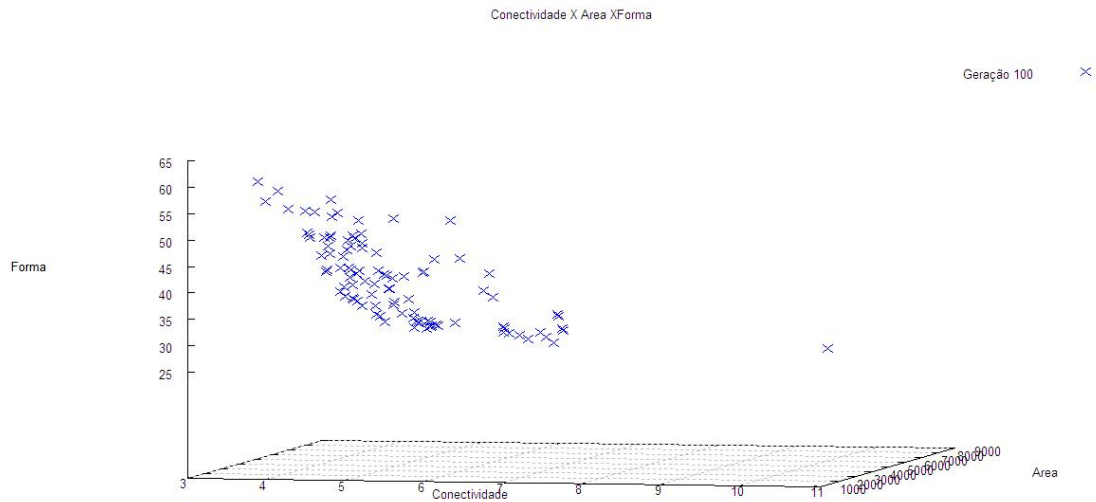


Figura 5.3 Simulação com 100 gerações e com população inicial de 100 indivíduos.

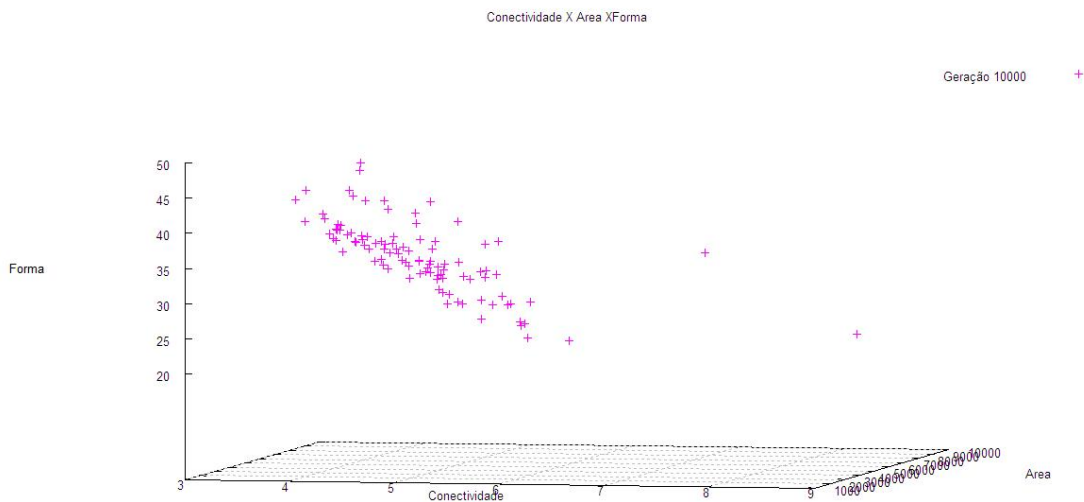


Figura 5.4 Simulação com 10000 gerações e com população inicial de 100 indivíduos.

Mutação.

A média do tempo de simulação foi de 431.93 segundos com desvio padrão de 16.59 para se fazer uma simulação de 20000 gerações.

Isso demonstra um considerável ganho de desempenho em relação a selecção manual

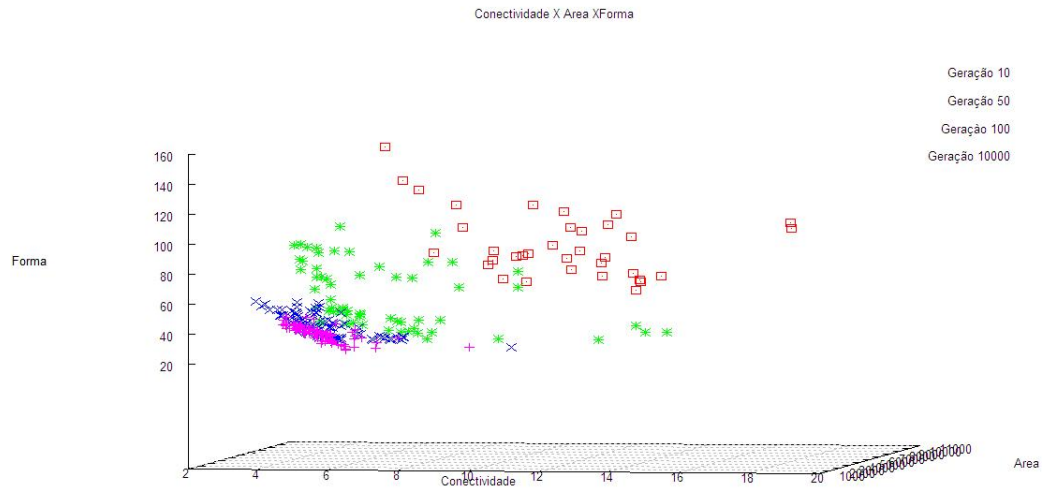


Figura 5.5 Simulação com 10, 50, 100 e 10000 gerações e com população inicial de 100 indivíduos.

dos fragmentos, o que poderia levar semanas através do trabalho de uma equipe. O Algoritmo Genético NSGA II mostra-se como uma ferramenta eficiente no processo de seleção de trechos de imagens de satélite da Mata Atlântica.

Demonstra também ser uma boa técnica computacional a ser usada para substituir o esforço humano que ainda é bastante empregado nessa área. Dessa forma, além de melhorar o desempenho em relação ao tempo de simulação o algoritmo traz uma maior confiabilidade em relação às informações obtidas.

CAPÍTULO 6

Conclusão

Através desse projeto foi possível concluir que o Algoritmo NSGA II funcionou de forma eficiente na seleção dos 20 trechos de Mata Atlântica, minimizando para isso apenas três objetivos que são : diferença de Forma, diferença de Área e diferença de Conectividade. Tornando o resultado desse estudo em uma importante contribuição da computação na pesquisa em preservação ambiental da Mata Atlântica do Departamento de Botânica da UFPE.

Quanto menores forem as diferenças de Área, Forma e Conectividade, maior será a similaridade existente entre os fragmentos, ou seja é segundo esse princípio que o Algoritmo Genético estabelecerá o seu padrão de busca, procurando por aqueles indivíduos que sejam constituídos por fragmentos que possuem maior grau de similaridade entre eles.

Através dos resultados obtidos pela ferramenta, pode-se ver claramente a utilidade desse software no suporte a tomada de decisão pela equipe de pesquisadores da UFPE, o que seria mais uma grande contribuição da Computação Inteligente na área de Otimização Multiobjetiva.

Vendo desta forma entende-se que essa área de estudo pode crescer ainda mais e ser usado em outros projetos correlatos. Trata-se de um estudo que certamente irá trazer impactos econômicos, mas principalmente impactos sociais e ambientais. Pois os estudos dos fragmentos permitirá que decisões possam ser tomadas a fim de se diminuir o impacto da degradação ambiental que ocorre sobre os trechos de Mata Atlântica que hoje sobrou por volta de apenas 7% de sua área original.

Referências Bibliográficas

- [1] Arkview - desktop gis for mapping, data integration, and analysis. Disponível em : <<http://www.esri.com/software/arcgis/arcview/index.html>> Acesso em : 16 Abril. 2009. 2.1
- [2] Eclipse. Disponível em : <<http://www.eclipse.org/>> Acesso em : 22 Abril. 2009. 4.1
- [3] Gnuplot software. Disponível em : <<http://www.gnuplot.info/>> Acesso em : 30 Março. 2009. 4.1
- [4] Jfreechart software. Disponível em : <<http://www.jfree.org/jfreechart/>> Acesso em : 30 Março. 2009. 4.1
- [5] Mata atlântica. Disponível em : <<http://educar.sc.usp.br/licenciatura/trabalhos/mataatl.htm>> Acesso em : 25 Abril. 2009. 1.1
- [6] Ong conservação internacional. Disponível em : <<http://www.conservation.org/Pages/default.aspx>> Acesso em : 04 Junho. 2009. 1.1
- [7] Situação da mata atlântica. Disponível em : <<http://www.universia.com.br/materia/materia.jsp?materia=4351>> Acesso em : 25 Abril. 2009. 1.1
- [8] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for solving multi-objective problems*. GENETIC AND EVOLUTIONARY COMPUTATION. Springer, 2 edition, 2002. 1.2
- [9] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for solving multi-objective problems*, chapter 1, page 11. GENETIC AND EVOLUTIONARY COMPUTATION. Springer, 2 edition, 2002. 3.1
- [10] H. M. Deitel and P. J. Deitel. *Java Como Programar*. 2005. 4.1
- [11] K. Deb E. Zitzler and L. Thiele. *Comparison of multiobjective evolutionary algorithms: Empirical results*. Evolutionary computation. 2000. 4.3.2
- [12] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*, chapter 6.7, page 331. Tom Robbins, 2 edition. 2.1
- [13] J.H. Holland. *Adaption in natural and artificial systems*. MI: University of Michigan Press, 1975. 3.2

- [14] Amrit Pratap Kalyanmoy Deb, Samir Agrawal and T Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Kanpur Genetic Algorithms Laboratory (KanGAL)*. 3.2.4
- [15] Melanie Mitchell. *An Introduction to Genetic Algorithms*. A Bradford Book The MIT Press, 1998. 3.3
- [16] E. Zitzler. Evolutionary algorithms for multiobjective optimization: Methods and applications. 1999. 4.3.1