

5

Estudo de Casos

Para o estudo de caso foi necessário implementar uma ferramenta capaz de simular o modelo proposto no capítulo 4. Para os testes e análises da ferramenta e do modelo, foram escolhidos três estudos de caso em níveis diferentes de complexidade. O sistema foi testado, inicialmente, para síntese de modelagens triviais, em seguida, foram testadas modelagens mais complexas com a finalidade de facilitar a análise dos resultados.

Todas as informações necessárias para o funcionamento do sistema são fornecidas através de um arquivo XML e uma base de dados com a *especificação inicial* da forma já descrita no capítulo 4. Este arquivo é dividido em seções e um exemplo de um arquivo de configuração é mostrado a seguir:

```
<config>
  <global>
    <params>
      <maxClass>10</maxClass>
      <wProbCont>1</wProbCont>
      <experiments>20</experiments>
      <generations>150</generations>
      <popsiz>300</popsiz>
      <metrics>
        <reusability>true</reusability>
        <flexibility>true</flexibility>
        <understandability>true</understandability>
        <functionality>true</functionality>
        <extendibility>true</extendibility>
        <effectiveness>true</effectiveness>
      </metrics>
    </params>
  </global>
  <class_specie>
    <params>
      <mutation>
        <mSwap>5</mSwap>
        <mPI>5</mPI>
        <mRotateLeft>5</mRotateLeft>
        <mRotateRight>5</mRotateRight>
      </mutation>
      <mask_mutation>
        <mAdd>2</mAdd>
        <mSub>2</mSub>
        <mFlipBits>5</mFlipBits>
        <mAdd2d>5</mAdd2d>
        <mSub2d>5</mSub2d>
        <mFlipBits2d>5</mFlipBits2d>
      </mask_mutation>
      <crossover>
        <cPMX>14</cPMX>
      </crossover>
    </params>
  </class_specie>
</config>
```

```

        <cOX>15</cOX>
        <cCX>15</cCX>
    </crossover>
    <ga>
        <mutation_prob>0,5</mutation_prob>
        <crossover_prob>0,9</crossover_prob>
        <steadyState>0,3</steadyState>
        <classIni>0</classIni>
    </ga>
</params>
</class_specie>
<dependency_specie>
    <params>
        <mutation>
            <mFlipBits>9</mFlipBits>
        </mutation>
        <crossover>
            <cOnePoint>5</cOnePoint>
            <cTwoPoints>4</cTwoPoints>
        </crossover>
        <ga>
            <mutation_prob>0,5</mutation_prob>
            <crossover_prob>0,9</crossover_prob>
            <steadyState>0,3</steadyState>
            <dependencyIni>0</dependencyIni>
        </ga>
    </params>
</dependency_specie>
<method_specie>
    <params>
        <mutation>
            <mSwap>6</mSwap>
            <mPI>6</mPI>
            <mRotateLeft>7</mRotateLeft>
            <mRotateRight>7</mRotateRight>
        </mutation>
        <crossover>
            <cPMX>10</cPMX>
            <cOX>9</cOX>
            <cCX>9</cCX>
        </crossover>
        <ga>
            <mutation_prob>0,65</mutation_prob>
            <crossover_prob>1</crossover_prob>
            <steadyState>0,3</steadyState>
        </ga>
    </params>
</method_specie>
<attribute_specie>
    <params>
        <mutation>
            <mSwap>6</mSwap>
            <mPI>6</mPI>
            <mRotateLeft>7</mRotateLeft>
            <mRotateRight>7</mRotateRight>
        </mutation>
        <crossover>
            <cPMX>10</cPMX>
            <cOX>9</cOX>
            <cCX>9</cCX>
        </crossover>

```

```
        <ga>
            <mutation_prob>0,65</mutation_prob>
            <crossover_prob>1</crossover_prob>
            <steadyState>0,3</steadyState>
        </ga>
    </params>
</attribute_specie>
</config>
```

Neste arquivo de configuração é possível modificar os parâmetros gerais da evolução como tamanho da população, número de gerações e número de experimentos. É também nele que o usuário escolhe o número máximo de classes da modelagem a ser sintetizada (*MaxClasses*) e os atributos de qualidade que serão otimizados. Após essa configuração, deve-se configurar os parâmetros para cada espécie do modelo.

Caso as taxas de *crossover* e *mutação* sejam iguais a zero, o sistema irá executar uma busca aleatória. Esta configuração é permitida para efeitos de comparação do algoritmo genético com a busca aleatória.

Depois de diversos testes com as taxas de operadores das quatro espécies, os valores exibidos no arquivo de configuração acima foram considerados satisfatórios e utilizados para todos os estudos de casos. Nas próximas seções uma análise dos testes será feita, considerando o desempenho do AG e a modelagem sintetizada.

5.1 Estudo de Caso 1: Modelagens simples

O primeiro estudo de caso consistiu na síntese de diversas modelagens triviais de modo a avaliar a funcionalidade do sistema. Na próxima seção serão mostradas as modelagens sintetizadas a partir de especificações iniciais e a consideração de cada objetivo (atributo de qualidade) separadamente.

5.1.1 Resultados

A primeira modelagem sintetizada partiu da especificação inicial abaixo:

Métodos

met1, met2

Atributos

atr1

Relações

met1 **usa** atr1

met2 **usa** met1

No primeiro teste, todos os atributos de qualidade foram considerados e a modelagem sintetizada pode ser vista na Figura 18. A seta cheia representa uma relação de herança.

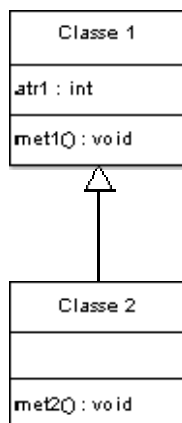


Figura 18 – Modelagem sintetizada

Analisando o resultado obtido, pode-se perceber que a *Classe 2* criada é uma especialização da *Classe 1*, herdando suas funcionalidades. A seta que aparece no diagrama indica a relação de herança, portanto a *Classe 2* possui o atr1, met1 e met2. Pode-se perceber também que o met1 aparece junto ao atr1, e o met2 junto ao met1 na *Classe 2*, devido as relações estabelecidas na especificação inicial, essa configuração diminui o acoplamento na modelagem.

Tornando o exemplo um pouco mais complexo, pode-se partir da especificação inicial abaixo:

Métodos

met1, met2, met3, met4

Atributos

atr1, atr2, atr3

Relações

met1 **usa** atr1

met2 **usa** met1

met2 **usa** atr3

met3 **usa** met4

met4 **usa** atr2

Otimizando todos os atributos de qualidade juntos, o AG chegou a seguinte modelagem:

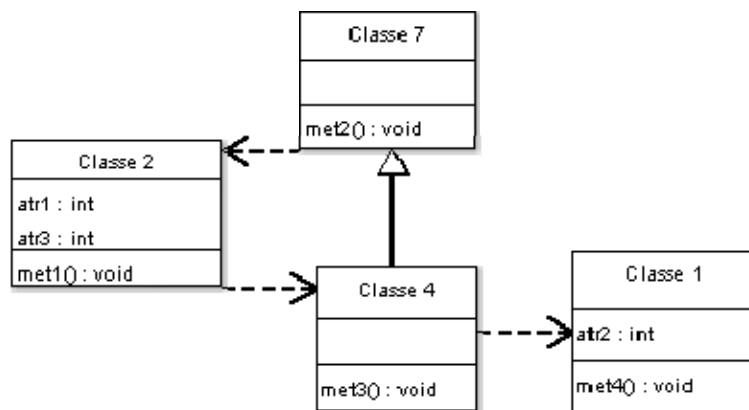


Figura 19 – Modelagem sintetizada

Na Figura, as setas cheias representam relações de herança, e as setas tracejadas representam relações de dependências. Neste caso, percebe-se que o AG criou algumas dependências entre as classes (setas tracejadas) para respeitar as relações entre métodos e atributos da especificação inicial. Novamente, foi criada relações de herança entre classes, de modo a proporcionar maior generalização-especialização.

Os valores atingidos para cada atributo de qualidade foram:

- Reutilização: 0,4762
- Flexibilidade: 0,7917
- Inteligibilidade: -0,1469
- Funcionalidade: 0,5269
- Extensibilidade: 0,7167
- Efetividade: 0,5533

Para uma melhor análise das características de cada atributo de qualidade, foram realizados testes favorecendo cada objetivo separadamente. As Figuras de 20 a 25 mostram as respectivas modelagens.

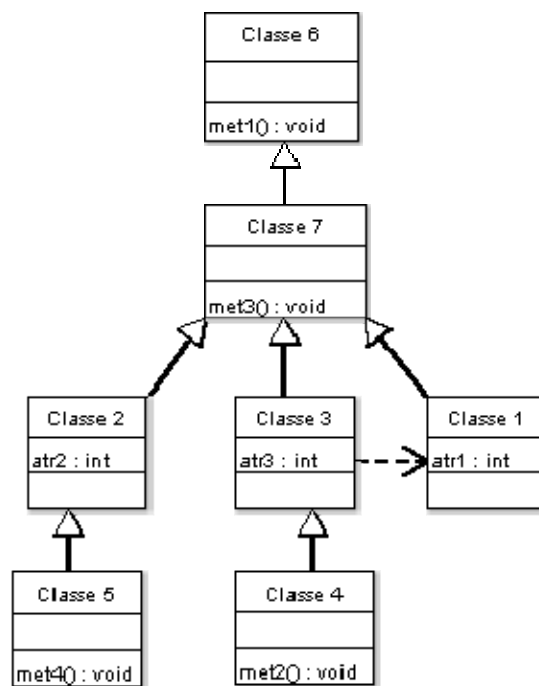


Figura 20 – Modelagem sintetizada otimizando atributo de *Reutilização*

- Reutilização: 0,9107

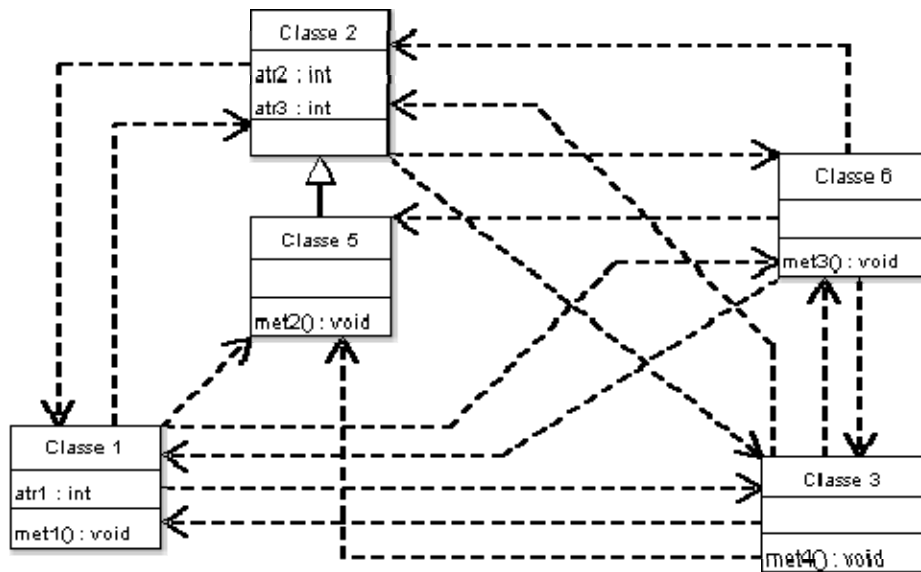


Figura 21 – Modelagem sintetizada otimizando atributo de *Flexibilidade*

- Flexibilidade: 0,9

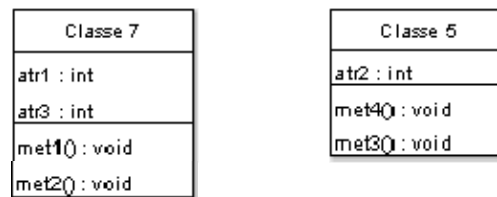


Figura 22 – Modelagem sintetizada otimizando atributo de *Inteligibilidade*

- Inteligibilidade: -0,0707

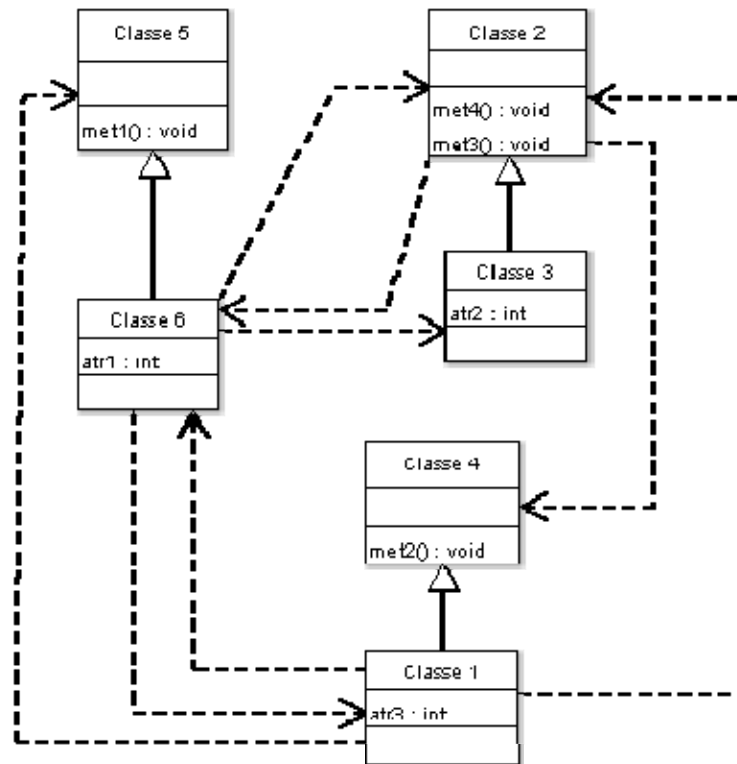


Figura 23 – Modelagem sintetizada otimizando atributo de *Funcionalidade*

- Funcionalidade: 0,7943

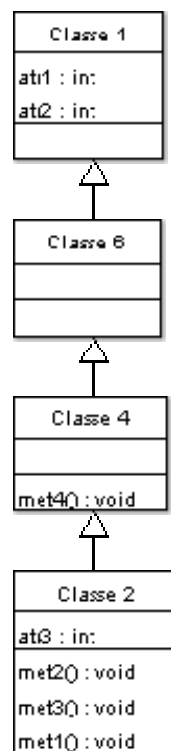


Figura 24 – Modelagem sintetizada otimizando atributo de *Extensibilidade*

- Extensibilidade: 0,9042

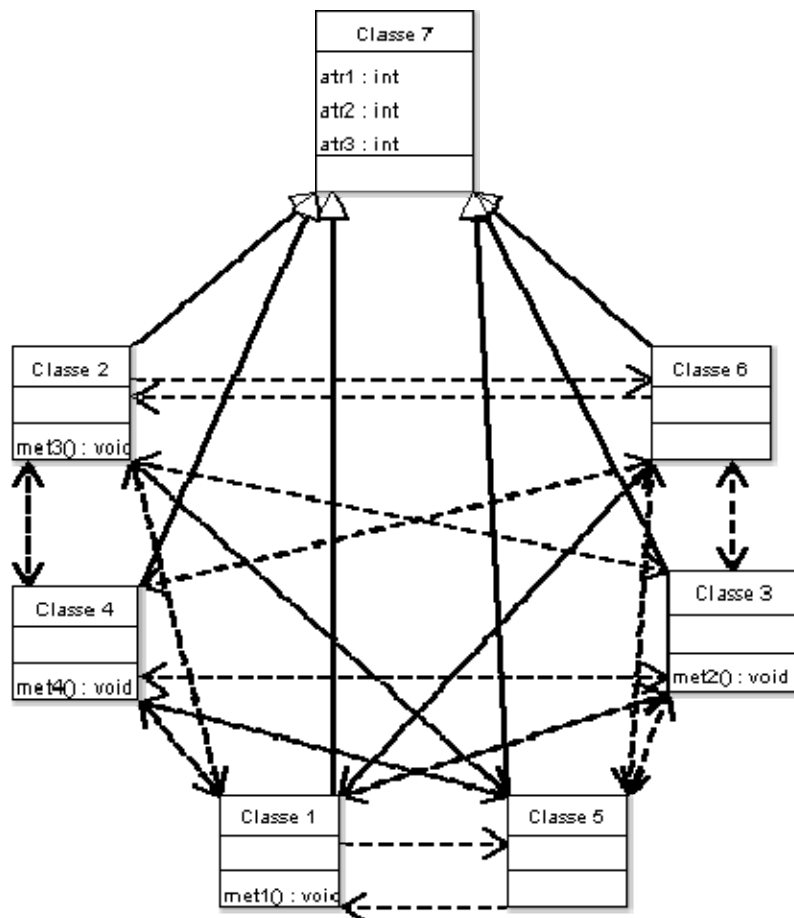


Figura 25 – Modelagem sintetizada otimizando atributo de *Efetividade*

- Efetividade: 0,7

Através da análise das modelagens geradas, pode-se determinar o que cada atributo de qualidade privilegia. Percebe-se que o valor do atributo de qualidade é muito maior quando se considera ele separadamente do que quando se considera todos juntos, o que era esperado, devido ao fato de que os atributos acabam sendo contraditórios, pois suas equações contêm as mesmas métricas.

Quando se otimiza a *Reutilização* algumas características podem ser percebidas, como o alto uso da herança e grande distribuição de métodos e atributos pelas classes. Existem também poucas dependências entre as classes e o diagrama fica bem parecido com uma árvore, sendo a *classe 6* a classe mais genérica, servindo de base para as outras, e as *classes 4* e *5* as classes mais especializadas. Outra característica a ser percebida é o grande número de classes criadas, o AG criou o número máximo de classes especificado (7). Pela definição

de *Reutilização*, a herança é uma das características que melhor proporcionam o reuso, dentro do paradigma de orientação a objetos.

No caso da *Flexibilidade*, pode-se ver um grande número de dependências entre as classes, isso ocorre para respeitar as relações entre métodos e atributos dadas na especificação inicial. Nesta otimização um grande número de classes também é criado, de modo que os métodos e atributos ficam bem distribuídos, e isso pode facilitar a modificação da modelagem de modo a prover as suas funcionalidades.

Um dos casos mais interessantes é quando se otimiza a *Inteligibilidade*. Naturalmente, a modelagem é a mais simples de todas, e provavelmente a mais trivial, o que concorda plenamente com o objetivo. Apenas duas classes são criadas, e todos os métodos e atributos que possuem relação estão juntos numa classe. Esse é um dos atributos de qualidade mais importante, pois é um dos que se aproxima mais das modelagens que uma pessoa com algum conhecimento na área criaria. A *Inteligibilidade* como dito na sua definição no capítulo 2, está fortemente ligada à complexidade.

O caso da otimização da *Funcionalidade* possui uma característica interessante, três hierarquias de classes foram criadas, de acordo com as relações especificadas. Pode-se identificar claramente três grupos de duas classes, uma mais genérica e a outra sendo uma especialização. Cada grupo de classes seria responsável por certa funcionalidade do sistema, mantendo em separado cada módulo funcional, e assim favorecendo o objetivo escolhido.

A *Extensibilidade* mede se a modelagem possui grande facilidade de se incorporar novas funcionalidades através da extensão da mesma. Na otimização foi criada uma única hierarquia de classes, possibilitando essa extensão em diversos níveis, variando a generalização.

A otimização da *Efetividade* privilegiou o grande uso da herança de um único nível, podendo ser notado pelo fato de todas as classes herdarem da *Classe 7*. Também teve um grande número de dependências, todas as classes estão dependendo de todas, com exceção da *Classe 7* que seria a classe *base*. Assim como na *Reutilização* o número total de classes foi criado. Uma característica importante de ser notada é o fato de todos os atributos pertencerem à classe *base*, e os métodos, que detêm as funcionalidades do sistema, estão espalhados pelas

classes herdeiras. Isso, de acordo com a definição de *Efetividade* seria de certa forma esperado.

Esse estudo de caso provou a diversificação e melhor caracterização dos atributos de qualidade considerados pelo modelo. Isso sugere que, de acordo com o objetivo do projetista, pode-se escolher a priorização de um ou outro objetivo, e até a combinação de dois ou mais. Mas sempre deve-se ter em vista que os objetivos são muitas vezes contraditórios, e isso diminui seu limite de otimização quando considerados juntos, pois o modelo procura um equilíbrio entre os mesmos.

Nos próximos estudos de caso, uma análise diferente será feita, considerando o desempenho do AG, as relações entre os atributos de qualidade e as métricas utilizadas e a validação das modelagens geradas através de comparações com modelagens feitas por especialistas.

5.2

Estudo de Caso 2: Controle de Elevador

Esse estudo de caso procurou usar uma modelagem um pouco mais complexa que as modelagens anteriores e que tivesse agora um sentido mais prático, de modo que possa ser comparado com uma modelagem gerada por um especialista. A modelagem escolhida é apresentada como um exemplo no livro “*Object-Oriented and Classical Software Engineering*” de Stephen R. Schach, pág. 407-415 (Schach, 2002). A modelagem consiste num sistema para o controle de um elevador.

5.2.1

Design

No livro, Schach apresenta as diversas etapas e diagramas da modelagem do problema para controlar o elevador. Para aplicar o modelo proposto, foram analisadas as características do problema proposto, através do livro e extraídas informações para formar a *especificação inicial*. O diagrama de classes simplificado, idealizado por Schach, pode ser visto na Figura 26.

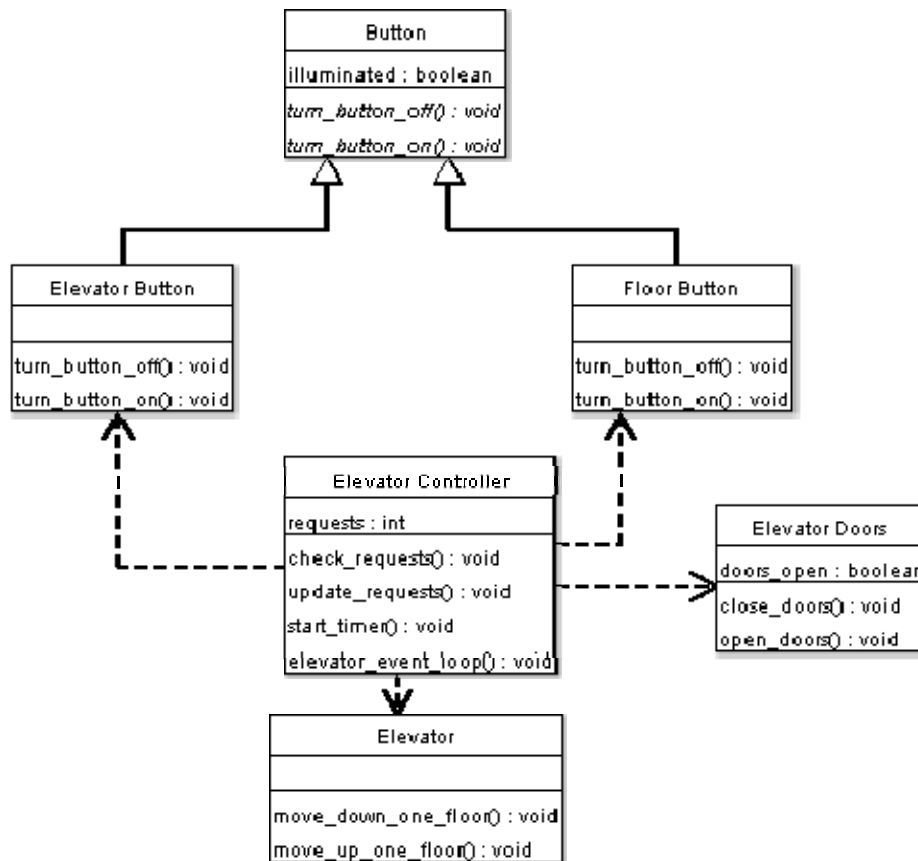


Figura 26 – Modelagem elaborada por um especialista do problema do elevador

A especificação inicial deduzida para o modelo ficou a seguinte:

Métodos

turn_button_on, turn_button_off, check_requests, update_requests, start_timer,
close_doors, open_doors, move_down_one_floor, move_up_one_floor,
elevator_event_loop

Atributos

illuminated, requests, doors_open

Relações

turn_button_off **usa** illuminated

turn_button_on **usa** illuminated

check_requests **usa** requests

update_requests **usa** requests

close_doors **usa** doors_open

open_doors **usa** doors_open

elevator_event_loop **usa** check_requests

elevator_event_loop **usa** turn_button_on

elevator_event_loop **usa** move_up_one_floor
 elevator_event_loop **usa** open_doors
 elevator_event_loop **usa** turn_button_off
 elevator_event_loop **usa** start_timer
 elevator_event_loop **usa** move_down_one_floor
 elevator_event_loop **usa** close_doors
 elevator_event_loop **usa** update_requests

A modelagem vista na Figura 26 foi colocada no modelo e avaliada. Os valores obtidos para os atributos de qualidade foram:

- Reutilização: 0,2571
- Flexibilidade: 0,7470
- Inteligibilidade: -0,0891
- Funcionalidade: 0,3796
- Extensibilidade: 0,6274
- Efetividade: 0,4819

5.2.2 Resultados

A análise dos resultados foi realizada sobre vários aspectos. Primeiramente, pode-se analisar o desempenho do AG através dos gráficos de evolução de cada objetivo. As Figuras seguintes mostram esses gráficos, em comparação com o algoritmo de busca aleatória. A linha mais escura representa o AG e a linha tracejada representa o algoritmo de busca aleatória. O gráfico mostra a média dos melhores em 150 gerações.

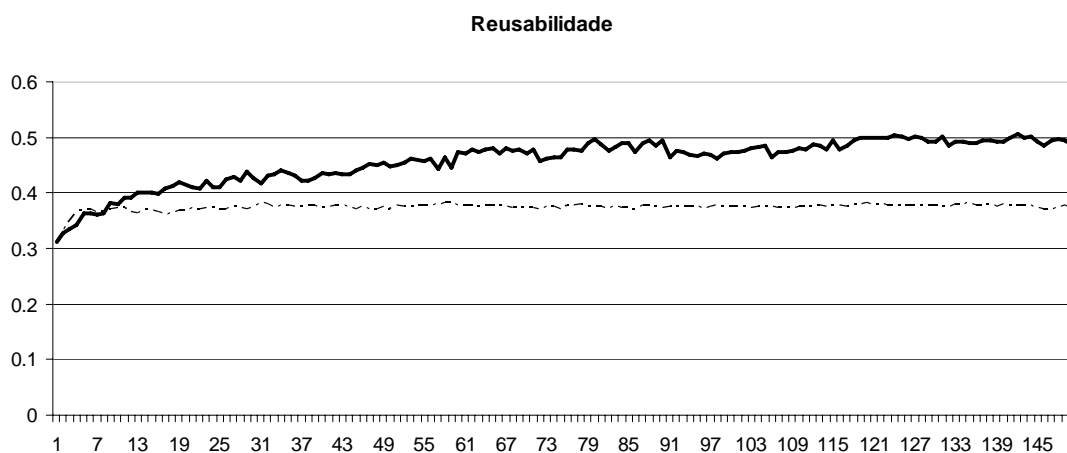


Figura 27 – Gráfico de desempenho do AG para o objetivo Reutilização

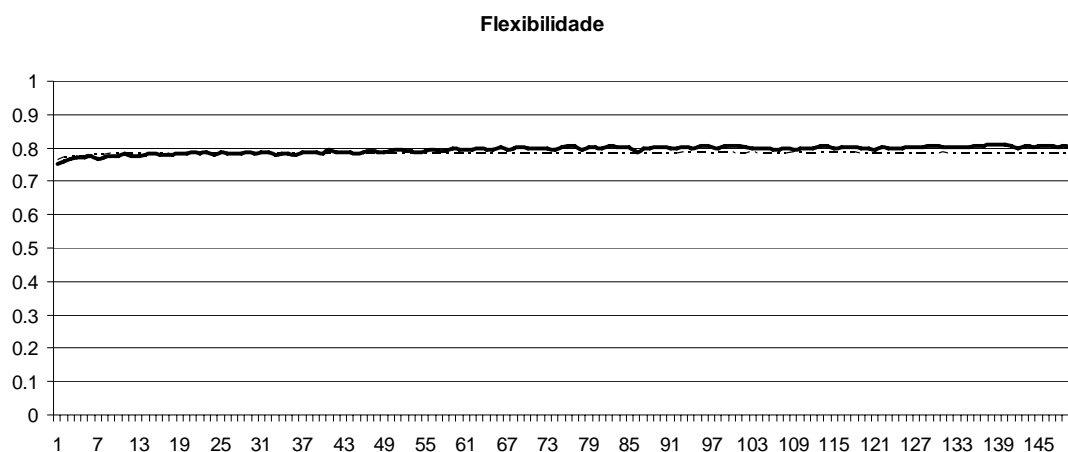


Figura 28 – Gráfico de desempenho do AG para o objetivo *Flexibilidade*

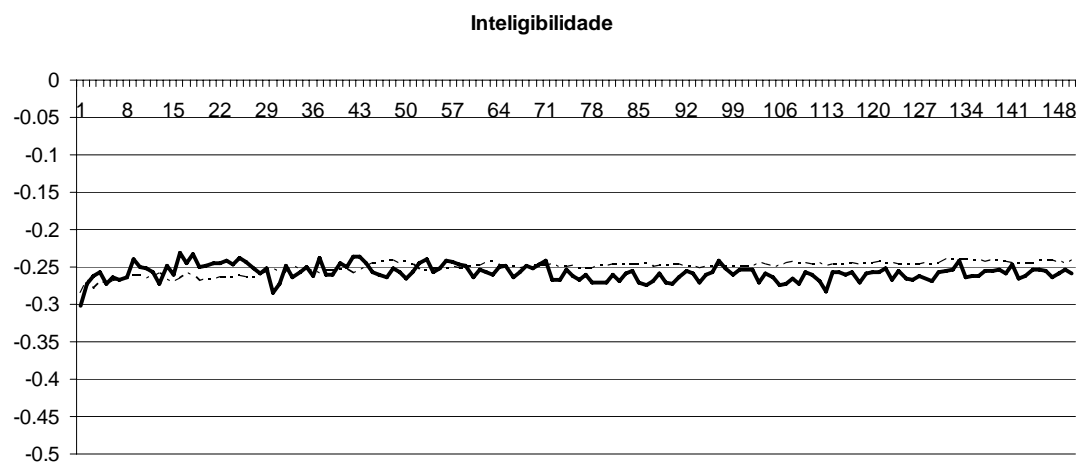


Figura 29 – Gráfico de desempenho do AG para o objetivo *Inteligibilidade*

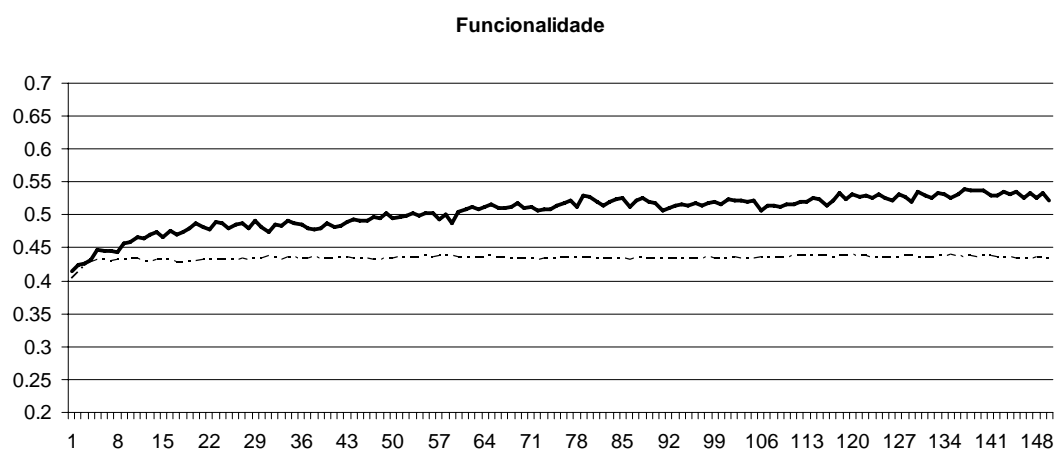


Figura 30 – Gráfico de desempenho do AG para o objetivo *Funcionalidade*

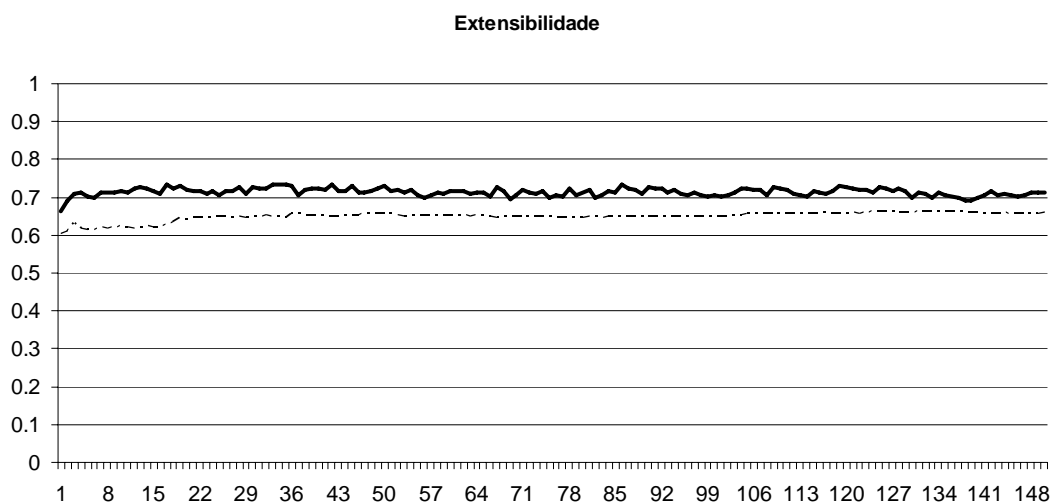


Figura 31 – Gráfico de desempenho do AG para o objetivo *Extensibilidade*

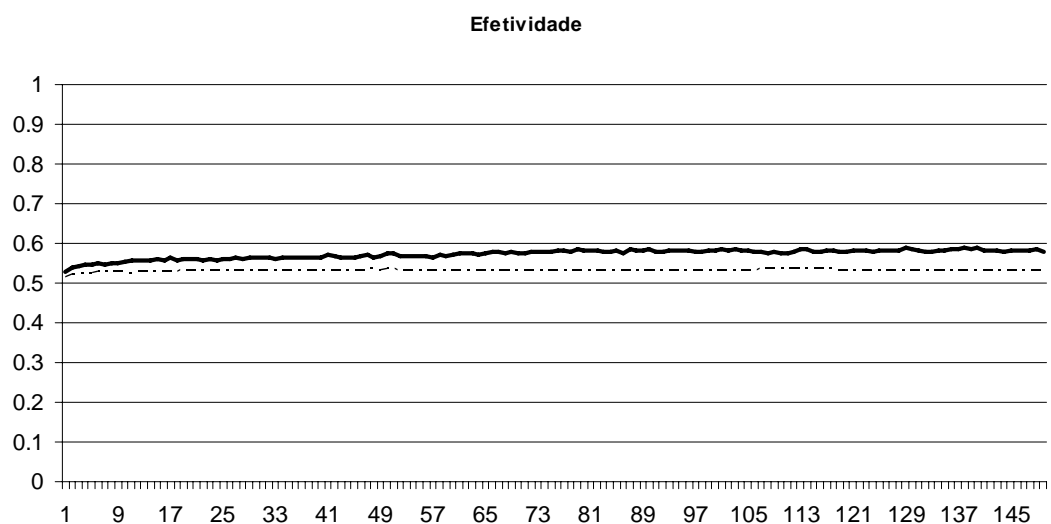


Figura 32 – Gráfico de desempenho do AG para o objetivo *Efetividade*

Esses gráficos foram gerados pela média dos melhores indivíduos de 15 experimentos com 150 gerações cada. Pode-se perceber que nos atributos *Reutilização* e *Funcionalidade*, o AG tem um desempenho bem superior à busca aleatória e nesses gráficos também percebe-se uma melhor evolução. Nos outros objetivos, a evolução praticamente fica constante, e o desempenho do AG não fica muito diferente do algoritmo de busca aleatória. Alguns fatores influenciam a ocorrência desse fato. A consideração de todos os objetivos juntos acaba prejudicando a evolução dos mesmos, já que alguns são considerados

contraditórios. Os gráficos gerados para cada objetivo separadamente serão mostrados mais adiante e isso poderá ser melhor percebido.

Outro fator a ser considerado é a evolução das métricas que formam as equações dos atributos. Os gráficos das métricas para a execução acima estão nas figuras abaixo.

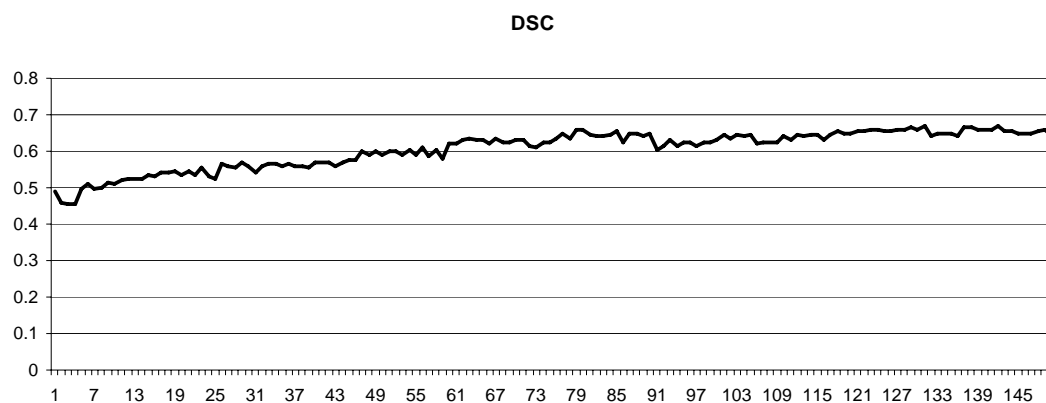


Figura 33 – Gráfico de evolução da métrica DSC

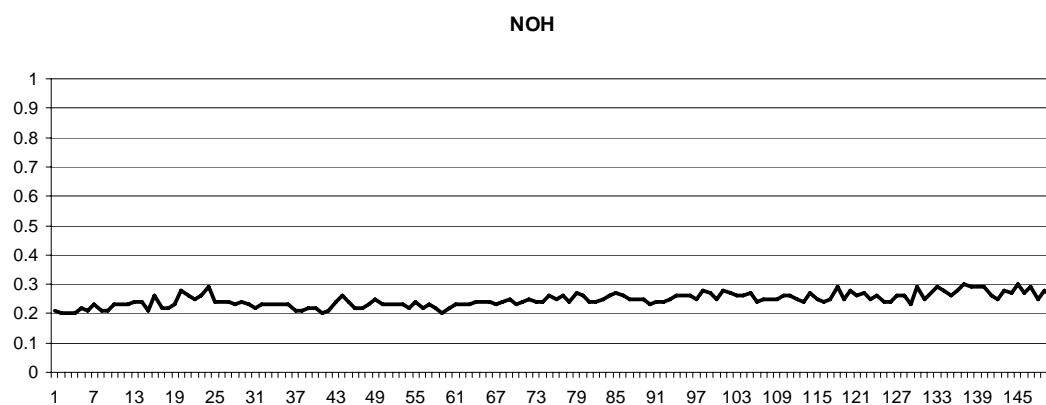


Figura 34 – Gráfico de evolução da métrica NOH

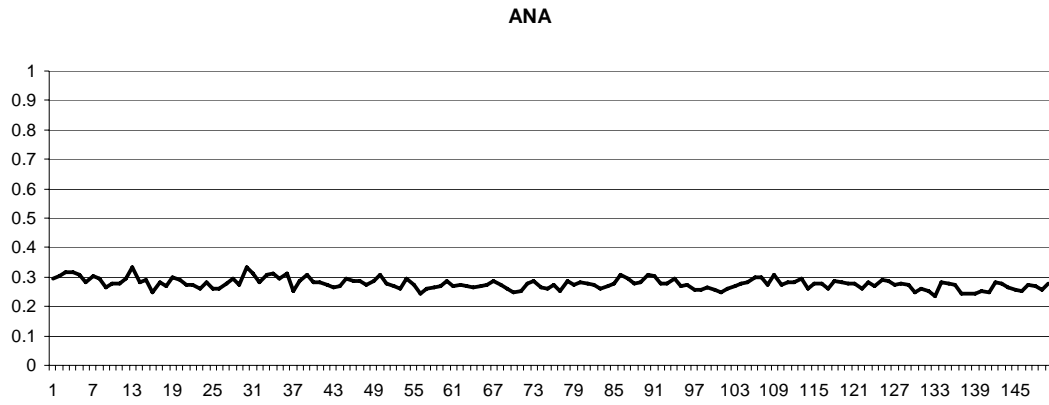


Figura 35 – Gráfico de evolução da métrica ANA

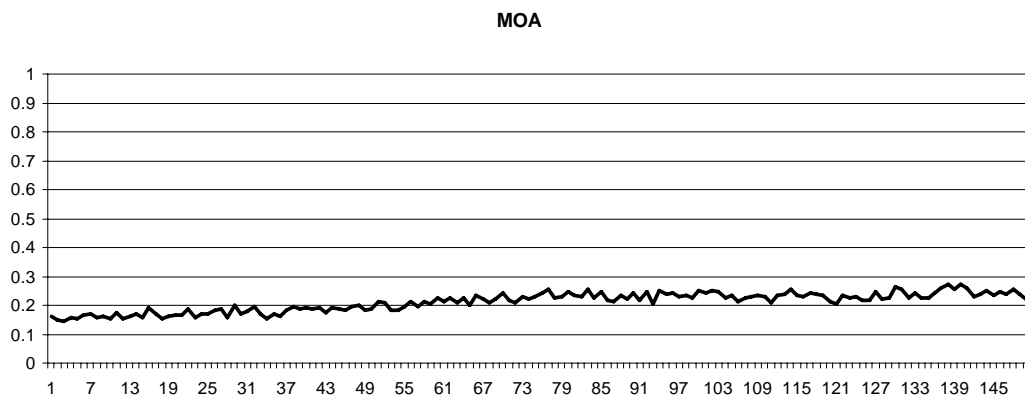


Figura 36 – Gráfico de evolução da métrica MOA

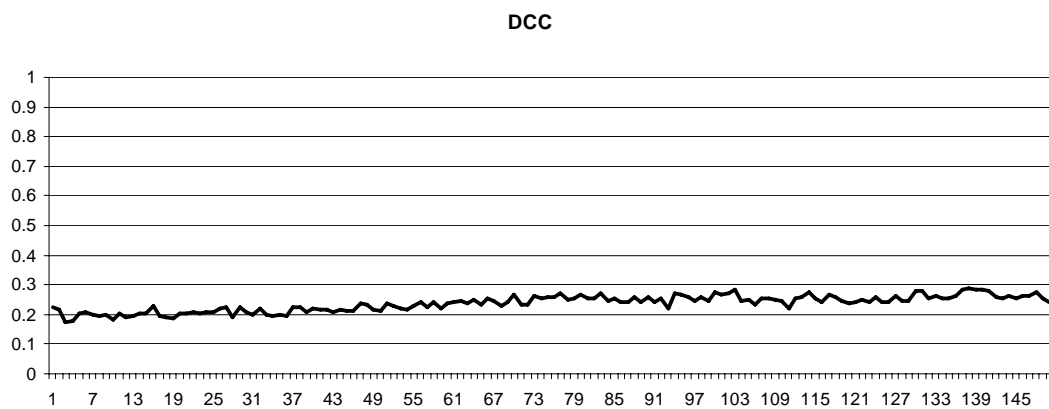


Figura 37 – Gráfico de evolução da métrica DCC

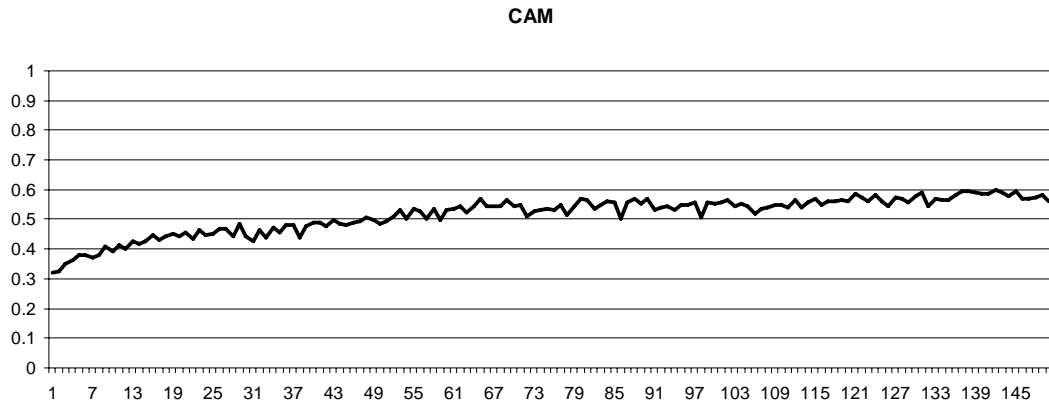


Figura 38 – Gráfico de evolução da métrica CAM

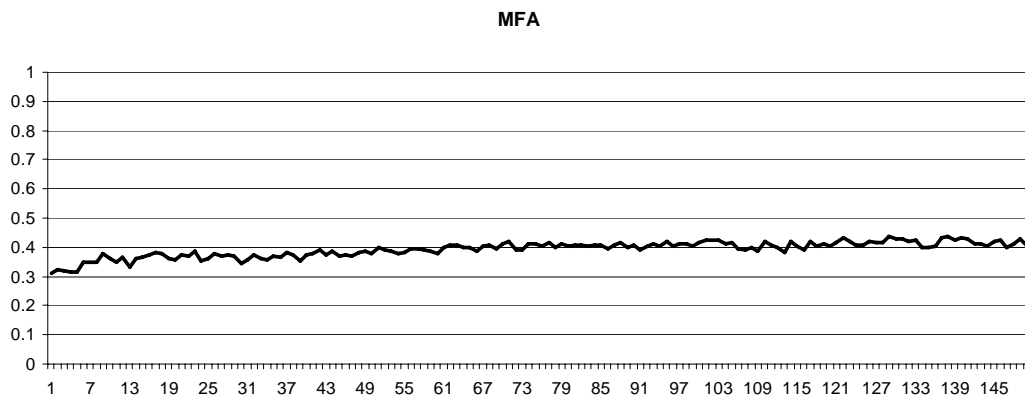


Figura 39 – Gráfico de evolução da métrica MFA

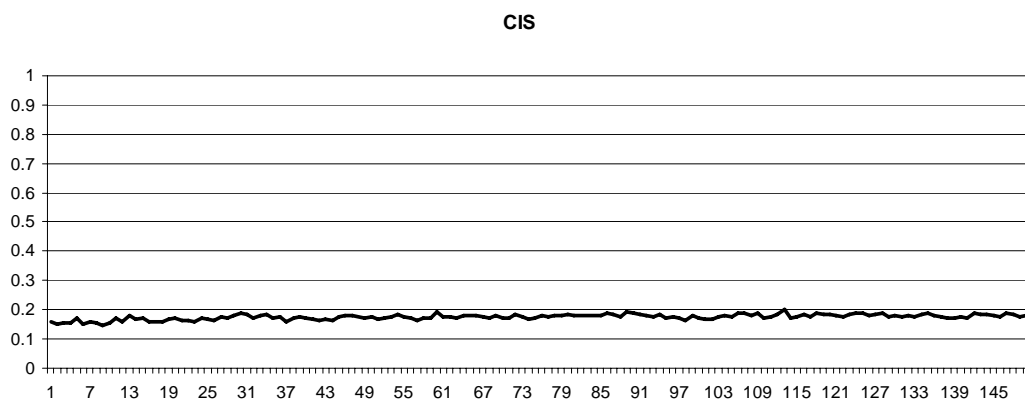


Figura 40 – Gráfico de evolução da métrica CIS

Analisando as equações que formam os atributos de qualidade abaixo, pode-se facilmente perceber o motivo do melhor desempenho dos atributos

Reutilização e Funcionalidade. As métricas *DSC* e *CAM* são as que possuem melhor evolução nos gráficos acima, e ambas estão presentes nas equações dos atributos de *Reutilização e Funcionalidade*.

Analisando o gráfico da *Flexibilidade* e sua equação, percebe-se que o motivo para a mínima evolução no gráfico se deve ao fato de, além de possuir duas métricas que têm valores constantes (*DAM* e *NOP*), como explicado no capítulo 4, as métricas *DCC* e *MOA* não possuem desempenho significativo, e possuem influência inversa na equação. As equações podem ser vistas abaixo:

$$\mathbf{Reutilização} = -0.25 * DCC + 0.25 * CAM + 0.5 * CIS + 0.5 * DSC$$

$$\mathbf{Flexibilidade} = 0.25 * DAM - 0.25 * DCC + 0.5 * MOA + 0.5 * NOP$$

$$\mathbf{Inteligibilidade} = -0.33 * ANA + 0.33 * DAM - 0.33 * DCC + 0.33 * CAM - 0.33 * NOP - 0.33 * CIS - 0.33 * DSC$$

$$\mathbf{Funcionalidade} = 0.12 * CAM + 0.22 * NOP + 0.22 * CIS + 0.22 * DSC + 0.22 * NOH$$

$$\mathbf{Extensibilidade} = 0.5 * ANA - 0.5 * DCC + 0.5 * MFA + 0.5 * NOP$$

$$\mathbf{Efetividade} = 0.2 * ANA + 0.2 * DAM + 0.2 * MOA + 0.2 * MFA + 0.2 * NOP$$

Essas métricas são medidas em características da modelagem sintetizada, sendo que algumas delas dependem dos mesmos componentes da modelagem, como visto no capítulo 4, e isso provoca contradições entre elas também. Por isso, quando todos os objetivos são considerados, todas as métricas acabam influenciando no resultado e, assim, as métricas contraditórias acabam encontrando um ponto equilíbrio para oscilar. Isso pode ser visto em casos em que somente é considerado um objetivo, portanto, as métricas que influenciam no resultado são as que fazem parte da equação do atributo sendo otimizado, permitindo maior liberdade para as métricas evoluírem.

A próxima análise consistiu na otimização de cada objetivo separadamente, de modo que cada um possa evoluir sem influências. Os gráficos destes testes podem ser vistos abaixo.

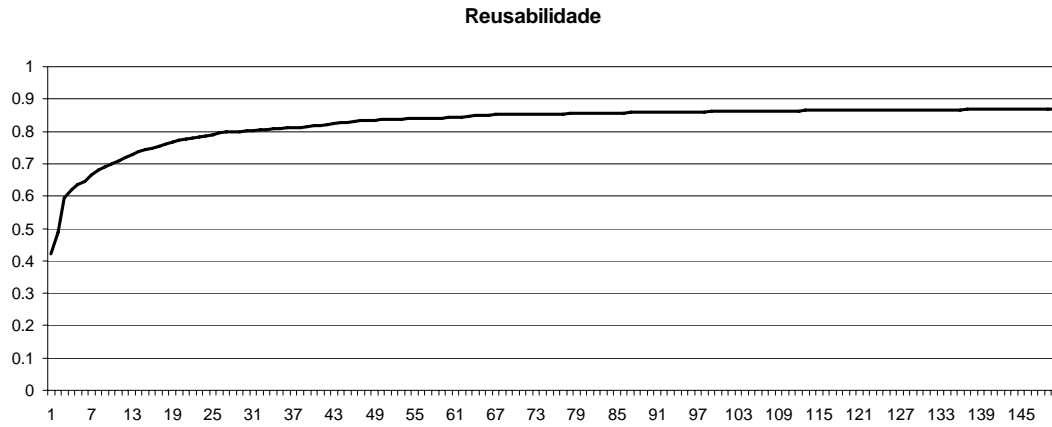


Figura 41 – Gráfico da otimização independente do atributo *Reutilização*

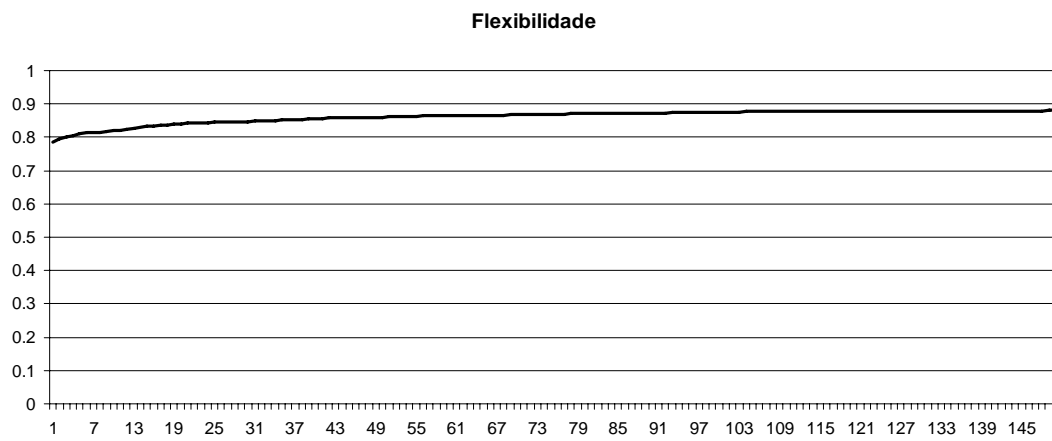


Figura 42 – Gráfico da otimização independente do atributo *Flexibilidade*

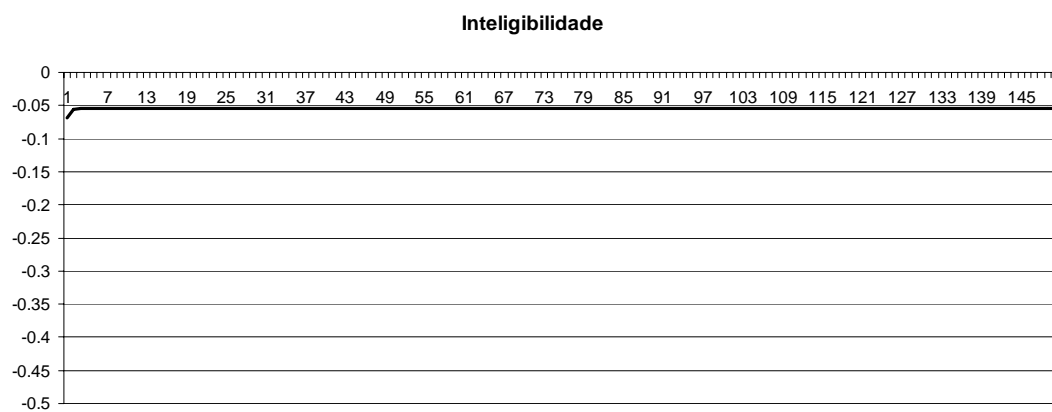


Figura 43 – Gráfico da otimização independente do atributo *Inteligibilidade*

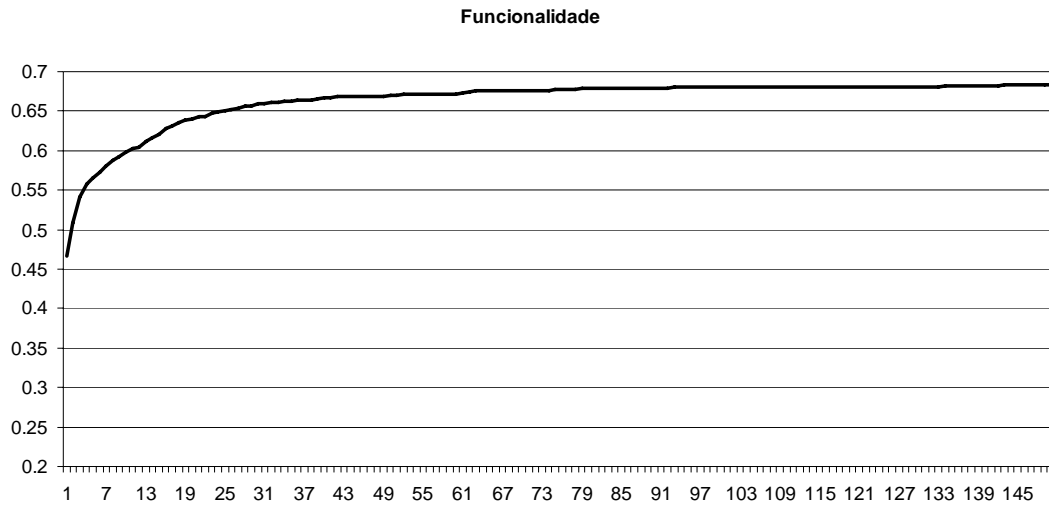


Figura 44 – Gráfico da otimização independente do atributo *Funcionalidade*

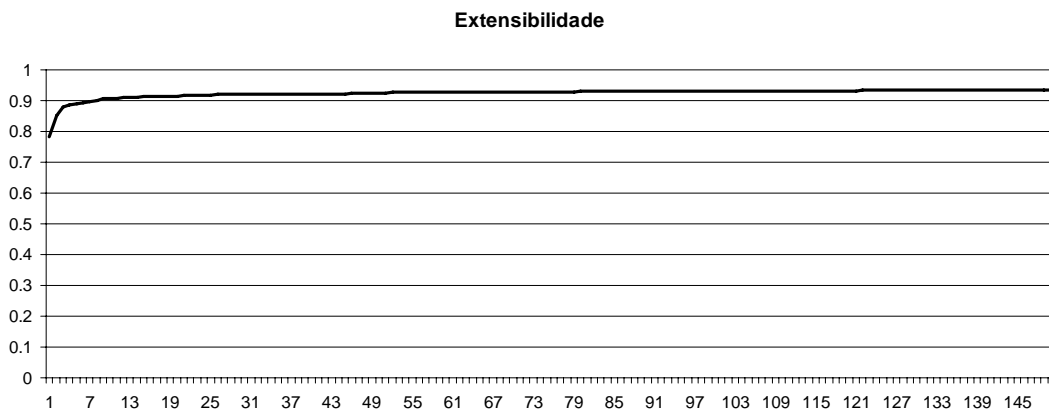


Figura 45 – Gráfico da otimização independente do atributo *Extensibilidade*

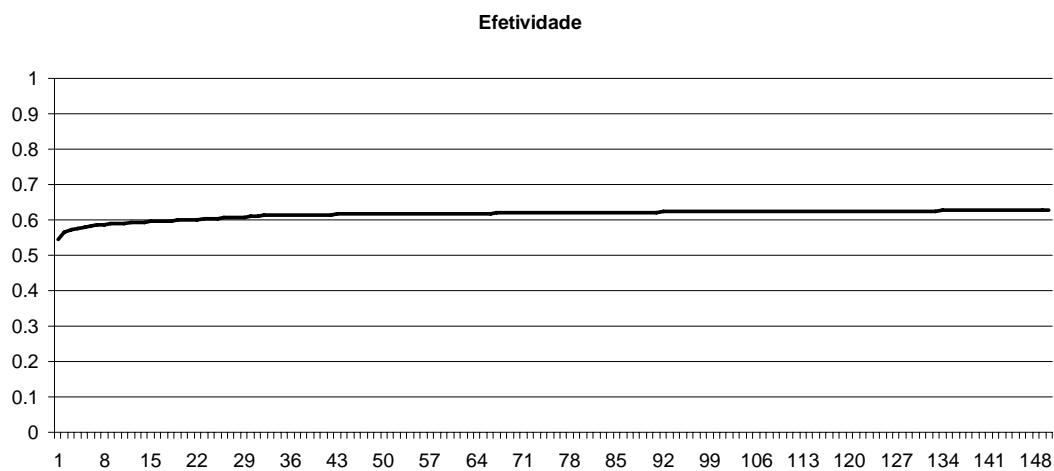


Figura 46 – Gráfico da otimização independente do atributo *Efetividade*

Os objetivos, quando são otimizados separadamente, atingem valores muito mais altos, e alguns chegam à convergência muito rapidamente. Isso ocorre porque as métricas que formam esses atributos são de certa forma rapidamente otimizadas.

O atributo *Efetividade* não teve muita mudança em relação à evolução em conjunto com os outros objetivos. Pode-se concluir portanto que as métricas que formam sua equação não permitem muita liberdade de evolução, além de ser influenciado por métricas constantes no modelo, a *DAM* e *NOP*.

A seguir, a análise focará nas modelagens sintetizadas nos diversos testes realizados. Primeiramente, todos os objetivos foram considerados, e a modelagem gerada pode ser vista na Figura 47. Os valores atingidos para os atributos estão na Tabela 19, onde são comparados com os medidos para a modelagem feita pelo especialista.

Atributo	Modelo	Especialista
Reutilização	0,4669	0,2571
Flexibilidade	0,8055	0,7470
Inteligibilidade	-0,2559	-0,0891
Funcionalidade	0,4994	0,3796
Extensibilidade	0,7555	0,6274
Efetividade	0,5911	0,4819

Tabela 19 – Comparação dos atributos de qualidade das modelagens

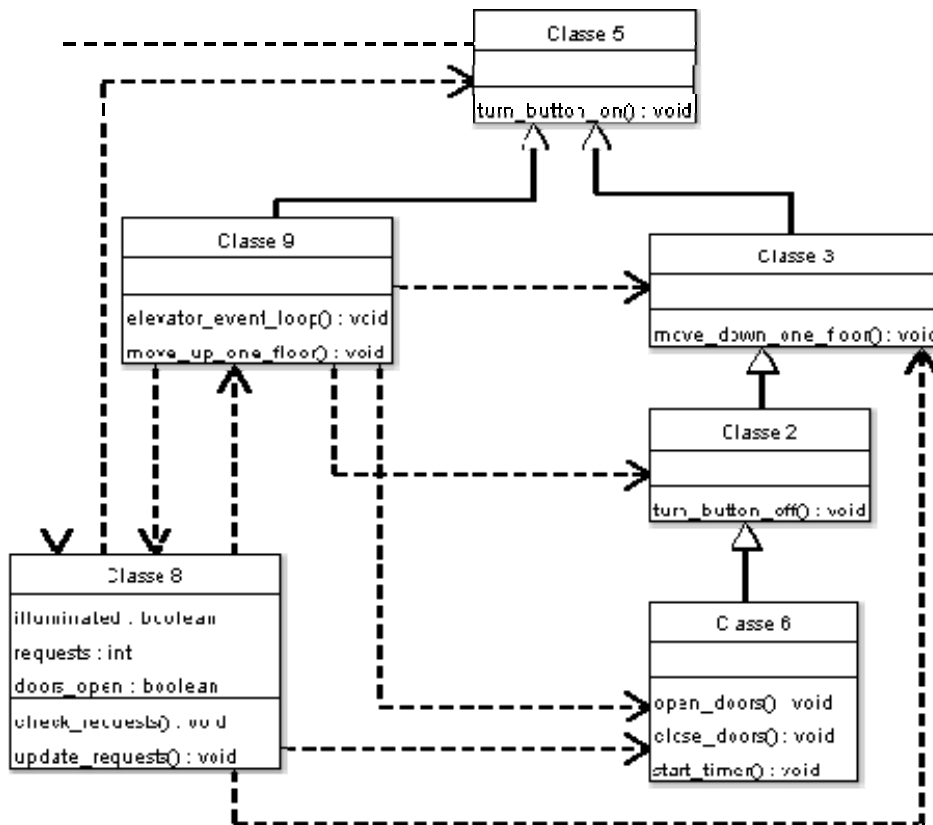


Figura 47 – Modelagem sintetizada na otimização de todos os atributos de qualidade

Analisando a Tabela 19, conclui-se que a modelagem sintetizada tem melhores valores para todos os atributos de qualidade, exceto a *Inteligibilidade*, onde a modelagem criada por um especialista possui um valor 65% melhor. Isso também pode ser notado na comparação das Figuras 26 e 47. Essa característica ocorreu na maioria dos testes, indicando que para melhorar os cinco outros atributos de qualidade a *Inteligibilidade* acaba sendo prejudicada. Por isso, analisando a modelagem da Figura 47, percebe-se muitas coisas sem sentido e que uma projetista não faria. Isso mostra que quando um especialista cria uma modelagem ele acaba priorizando a *Inteligibilidade*, como também pôde ser percebido no primeiro estudo de caso.

Um caso interessante de análise é a síntese da modelagem priorizando somente o objetivo *Inteligibilidade*. A Figura 48 mostra o resultado.

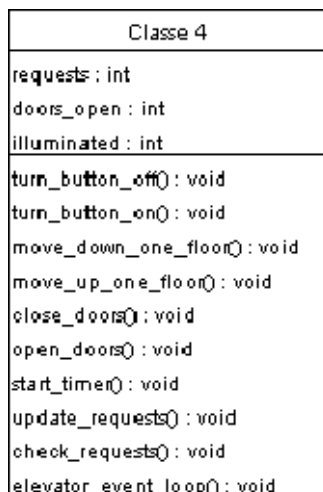


Figura 48 – Modelagem sintetizada na otimização da *Inteligibilidade*

Somente uma classe foi criada na otimização, e o valor atingido para o atributo *Inteligibilidade* foi -0,055, apenas 3,8% melhor do que a modelagem do livro neste atributo. Isso sugere que a melhor modelagem talvez seja obtida através de combinações entre os objetivos, sendo um deles a *Inteligibilidade*, pois é o que mais se aproxima de uma modelagem clara e objetiva.

Considerando a técnica de avaliação estudada, a solução criada por um especialista seria uma solução pertencente ao conjunto Pareto-Ótimo, e poderia perpetuar na evolução. Por isso, um novo teste realizado foi o de inserir a solução do especialista como uma semente na primeira população. Neste teste, o AG aproveitou a solução gerada no começo, e várias combinações de objetivos foram testadas, a melhor foi utilizando os atributos *Inteligibilidade* e *Flexibilidade*. A modelagem gerada pode ser vista na Figura 49.

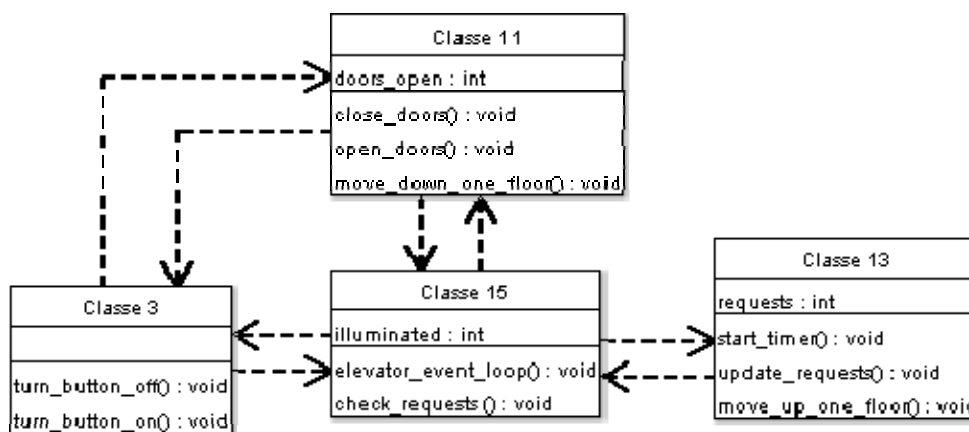


Figura 49 – Otimização da *Inteligibilidade* e *Flexibilidade* usando semente inicial

Neste caso, percebe-se várias características da semente inicial na solução. Os valores atingidos para os atributos de qualidade foram:

- Flexibilidade: 0,7857
- Inteligibilidade: -0,0827

O valor do atributo *Inteligibilidade* está bem próximo do valor encontrado para a modelagem do especialista. Outra característica a ser percebida é o agrupamento dos métodos e atributos, que foi praticamente mantido parecido com a semente inicial. Isso tudo indica que a solução dada pelo especialista realmente é uma boa solução, já que características dela foram aproveitadas na evolução.

No próximo estudo de caso, uma modelagem mais complexa é sintetizada e analisada. A modelagem obtida foi comparada a uma modelagem gerada manualmente para representar uma parte de um *framework* de Algoritmos Genéticos.

5.3 Estudo de Caso 3: GACOM

Esse estudo de caso consistiu num teste mais complexo que também pode ser comparado a uma modelagem desenvolvida manualmente por especialistas. O exemplo escolhido utiliza uma modelagem já desenvolvida e implementada, relacionado com Inteligência Computacional.

O GACOM (Genetic Algorithms Components) pode ser visto como um *framework* para o desenvolvimento de sistemas que utilizam a técnica inteligente de Algoritmos Genéticos. Portanto, o objetivo desse teste foi utilizar uma fração deste *framework* para ser sintetizado e avaliado qualitativamente.

5.3.1 Design

O GACOM é composto de quatro módulos principais: o módulo responsável pelo processo de evolução, o módulo responsável pelo processo de avaliação, o módulo das estruturas principais do AG e o módulo que contém todas

as interfaces do GACOM. No estudo de caso, somente foi utilizado o módulo do processo de evolução. Um diagrama deste módulo pode ser visto na Figura 50.

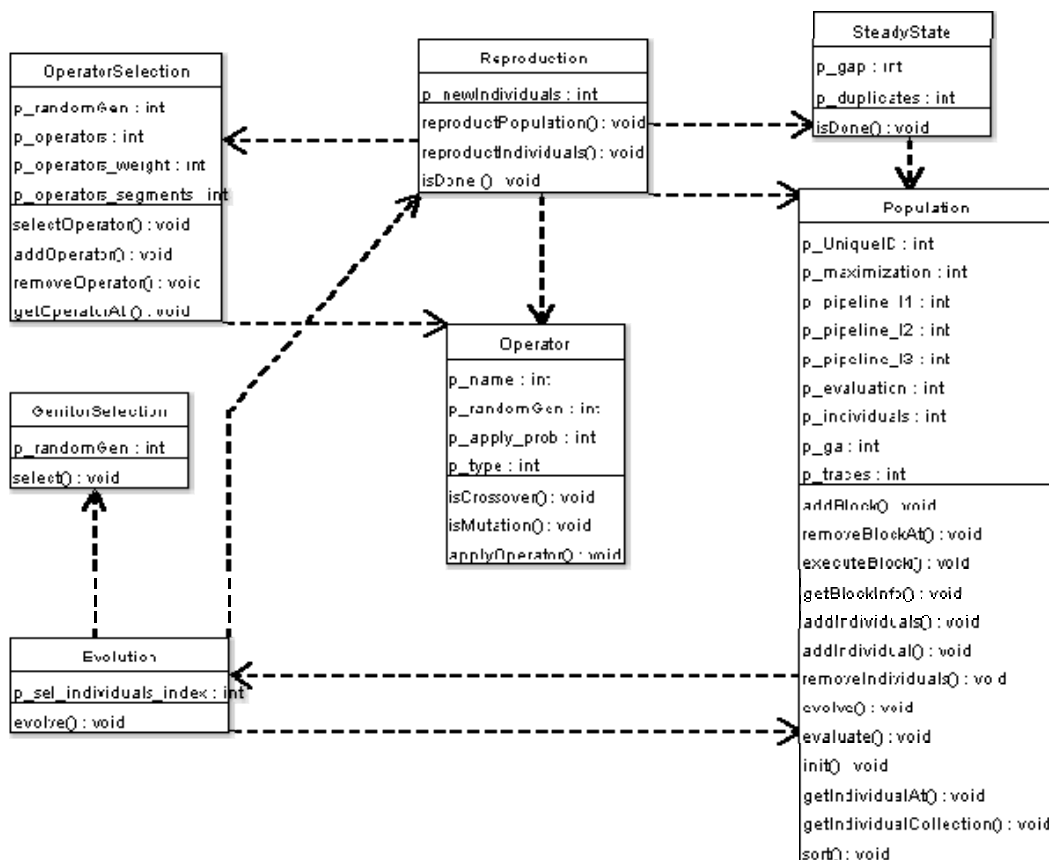


Figura 50 – Módulo do processo de evolução do GACOM

A especificação inicial que serviu de entrada para o modelo foi a seguinte:

Métodos

isDone, select, selectOperator, addOperator, getOperatorAt, removeOperator, isMutation, isCrossover, applyOperator, reproductIndividuals, isDone, reproductPopulation, evolve, addBlock, removeBlockAt, getBlockInfo, addIndividuals, addIndividual, removeIndividuals, evolve, evaluate, init, getIndividualAt, getIndividualCollection, sort, executeBlock

Atributos

p_gap, p_duplicates, p_randomGen, p_operators, p_operators_weight, p_operators_segments, p_randomGen, p_type, p_apply_prob, p_name, p_randomGen, p_newIndividuals, p_sel_individuals_index, p_UniqueID, p_maximization, p_pipeline_11, p_pipeline_12, p_pipeline_13, p_evaluation, p_individuals, p_ga, p_traces

Relações

isDone **usa** p_gap

isDone **usa** p_duplicates
isDone **usa** p_newIndividuals
select **usa** p_randomGen
select **usa** p_individuals
selectOperator **usa** p_randomGen
selectOperator **usa** p_operators
selectOperator **usa** p_operators_weight
selectOperator **usa** p_operators_segments
addOperator **usa** p_operators
addOperator **usa** p_operators_weight
addOperator **usa** p_operators_segments
getOperatorAt **usa** p_operators
removeOperator **usa** p_operators
removeOperator **usa** p_operators_weight
removeOperator **usa** p_operators_segments
isMutation **usa** p_apply_prob
isCrossover **usa** p_apply_prob
applyOperator **usa** p_apply_prob
applyOperator **usa** p_randomGen
reproductIndividuals **usa** p_individuals
reproductIndividuals **usa** p_sel_individuals_index
reproductPopulation **usa** p_newIndividuals
evolve **usa** p_gap
evolve **usa** p_UniqueID
evolve **usa** p_sel_individuals_index
addBlock **usa** p_pipeline_11
addBlock **usa** p_pipeline_12
addBlock **usa** p_pipeline_13
removeBlockAt **usa** p_pipeline_11
removeBlockAt **usa** p_pipeline_12
removeBlockAt **usa** p_pipeline_13
getBlockInfo **usa** p_pipeline_11
getBlockInfo **usa** p_pipeline_12
getBlockInfo **usa** p_pipeline_13
addIndividuals **usa** p_individuals
addIndividual **usa** p_individuals
removeIndividuals **usa** p_individuals
evolve **usa** p_individuals
evolve **usa** p_traces
evolve **usa** p_ga

```

evolve usa p_UniqueID
evaluate usa p_ga
evaluate usa p_evaluation
evaluate usa p_UniqueID
init usa p_traces
init usa p_individuals
getIndividualAt usa p_individuals
getIndividualCollection usa p_individuals
sort usa p_individuals
sort usa p_maximization
executeBlock usa p_pipeline_11
executeBlock usa p_pipeline_12
executeBlock usa p_pipeline_13
reproductIndividuals usa selectOperator
reproductIndividuals usa isMutation
reproductIndividuals usa isCrossover
reproductIndividuals usa applyOperator
isDone usa isDone
reproductPopulation usa sort
reproductPopulation usa getIndividualCollection
evolve usa getIndividualCollection
evolve usa select
evolve usa isDone
evolve usa reproductIndividuals
evolve usa reproductPopulation
evolve usa sort
evolve usa evolve
evolve usa executeBlock
evaluate usa executeBlock

```

Novamente, a modelagem da Figura 49 foi colocada no modelo e avaliada. Os valores obtidos para os atributos de qualidade foram:

- Reutilização: 0,280
- Flexibilidade: 0,7678
- Inteligibilidade: -0,1579
- Funcionalidade: 0,3598
- Extensibilidade: 0,6670
- Efetividade: 0,5137

5.2.2 Resultados

Os testes realizados neste estudo de caso consistiram em comparações de desempenho, evolução separada dos objetivos e avaliação das modelagens sintetizadas, em comparação com a gerada manualmente.

Os gráficos abaixo mostram a comparação da evolução do AG com o algoritmo de busca aleatória. Todos os gráficos foram gerados a partir da média dos melhores indivíduos em 30 experimentos com 300 gerações cada. A linha tracejada representa a busca aleatória.

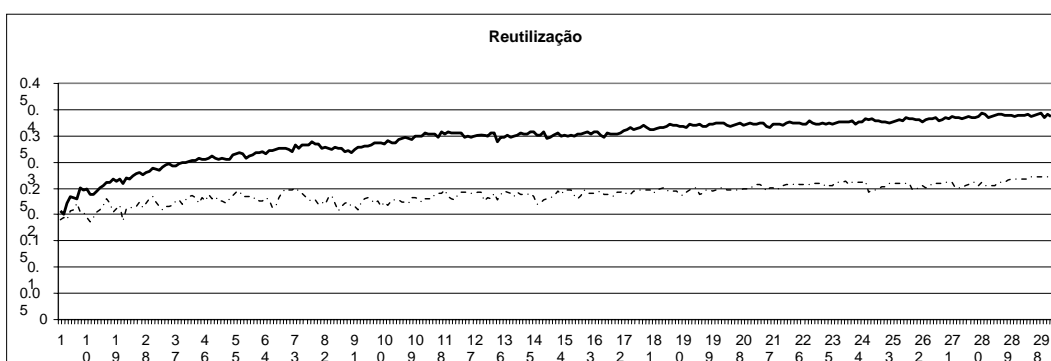


Figura 51 – Comparativo da evolução do AG X Busca aleatória. AG 42,6% melhor

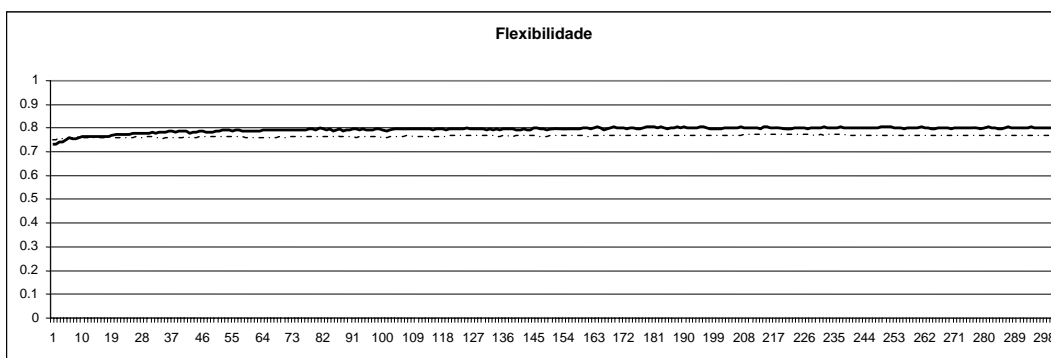


Figura 52 – Comparativo da evolução do AG X Busca aleatória. AG 3,9% melhor

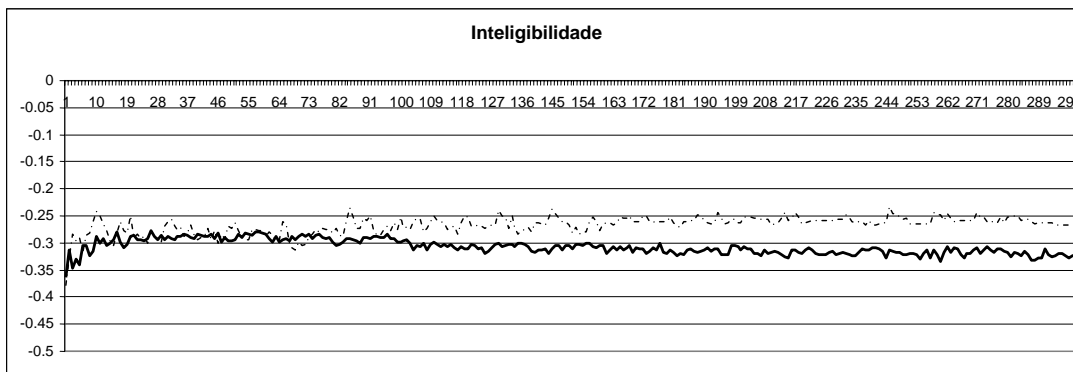


Figura 53 – Comparativo da evolução do AG X Busca aleatória. AG 21% pior

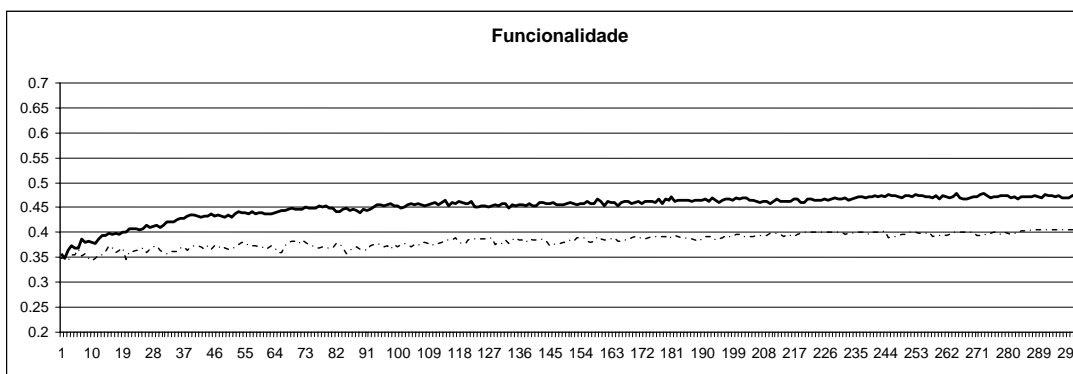


Figura 54 – Comparativo da evolução do AG X Busca aleatória. AG 19,25% melhor

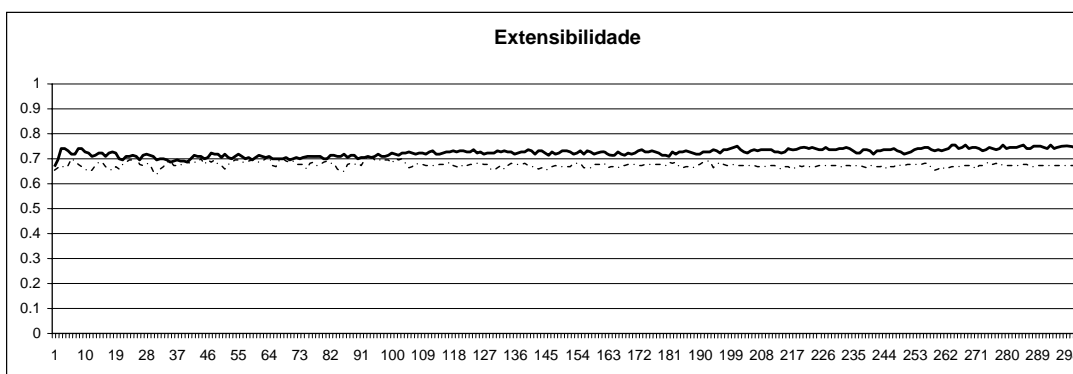


Figura 55 – Comparativo da evolução do AG X Busca aleatória. AG 10,7% melhor

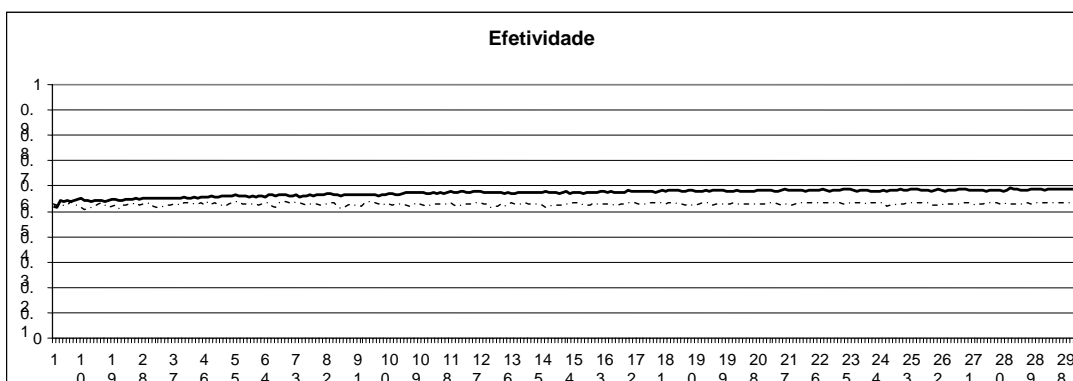


Figura 56 – Comparativo da evolução do AG X Busca aleatória. AG 9,9% melhor

Modelagem sintetizada	Modelagem manual
Class13	SteadyState
Class14 e Class20	Operator
Class7 e Class3	OperatorSelection
Class11	GenitorSelection
Class19	Evolution
Class4 e Class2	Reproduction
Class8 e Class12	Population

Tabela 20 – Comparação das modelagens

Testes de como as métricas influenciam em cada objetivo foram realizados. Para isso, o modelo foi executado separadamente para cada objetivo, e as métricas foram monitoradas durante a evolução. Foi gerado no mesmo gráfico, a evolução com todos os objetivos juntos para a comparação. A linha mais escura se refere a evolução do objetivo separadamente. Abaixo seguem os gráficos gerados para cada objetivo.

$$\text{Reutilização} = -0.25 *DCC + 0.25 *CAM + 0.5 *CIS + 0.5 *DSC$$

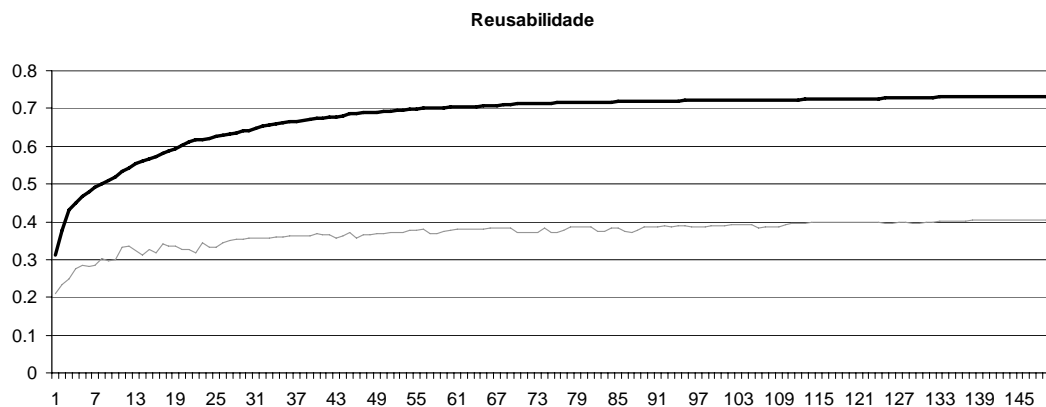


Figura 58 – Gráfico de desempenho para o atributo *Reutilização*

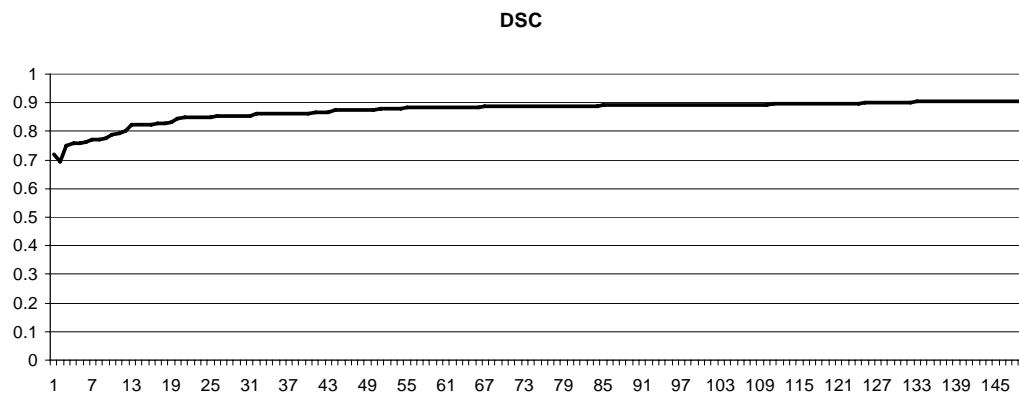


Figura 59 – Gráfico de monitoramento da métrica DSC durante a evolução

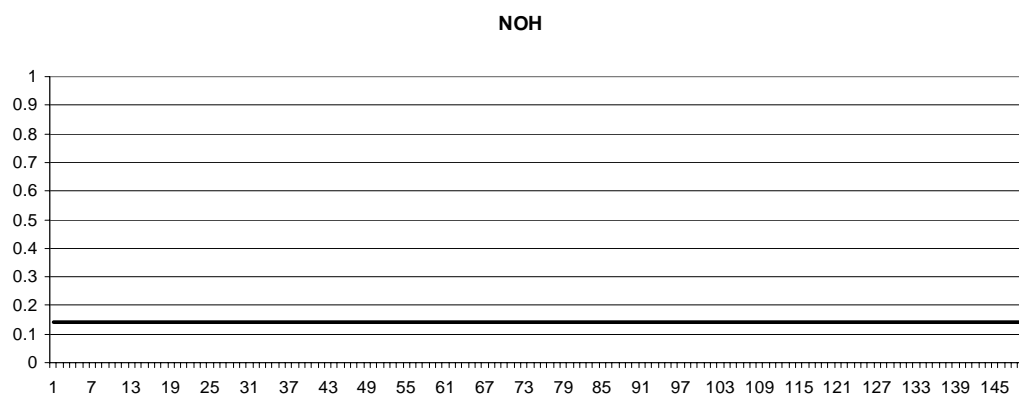


Figura 60 – Gráfico de monitoramento da métrica NOH durante a evolução

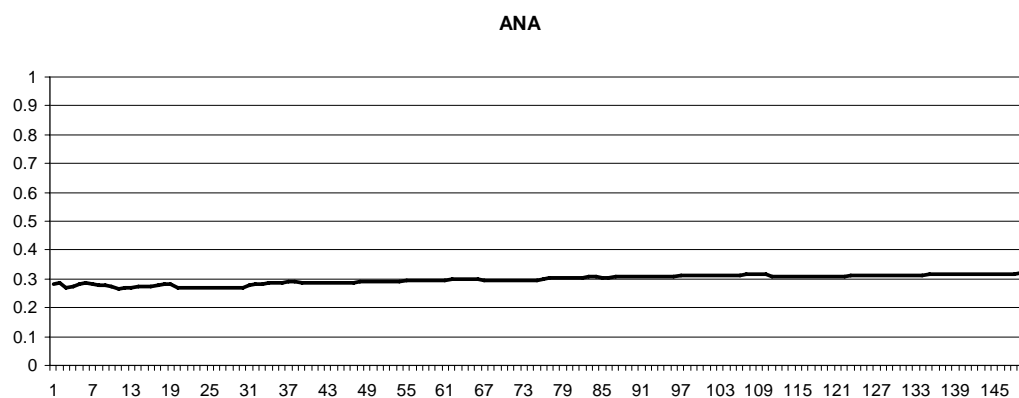


Figura 61 – Gráfico de monitoramento da métrica ANA durante a evolução

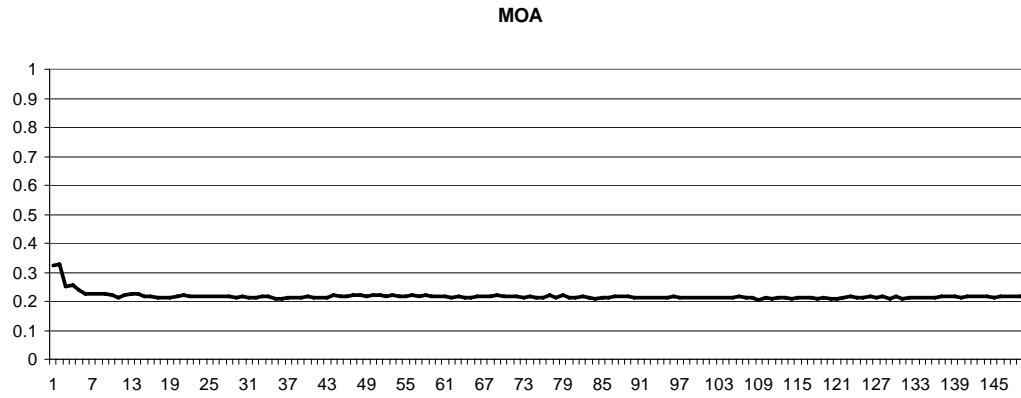


Figura 62 – Gráfico de monitoramento da métrica MOA durante a evolução

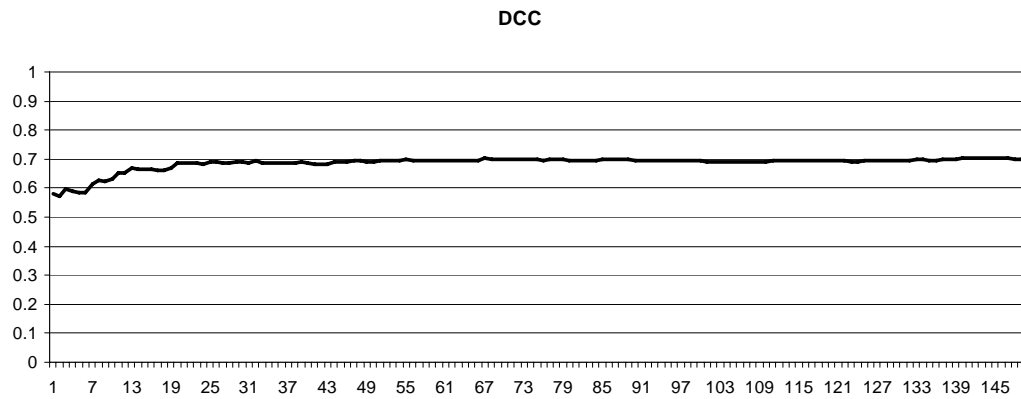


Figura 63 – Gráfico de monitoramento da métrica DCC durante a evolução

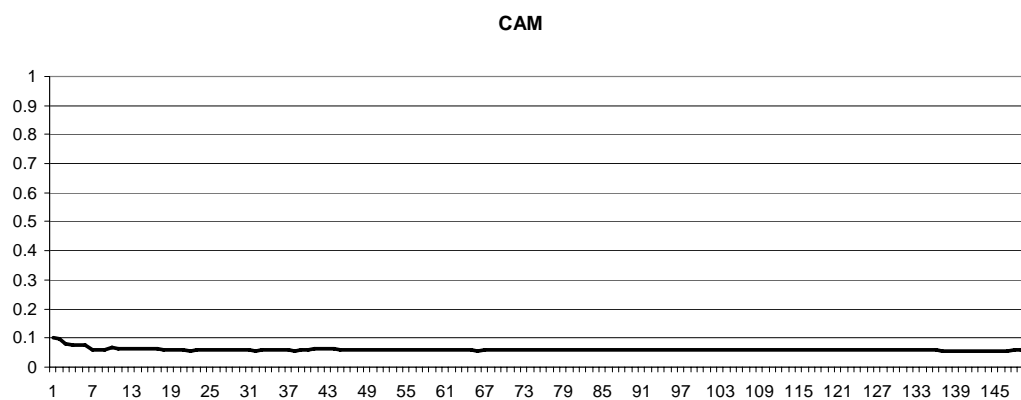


Figura 64 – Gráfico de monitoramento da métrica CAM durante a evolução

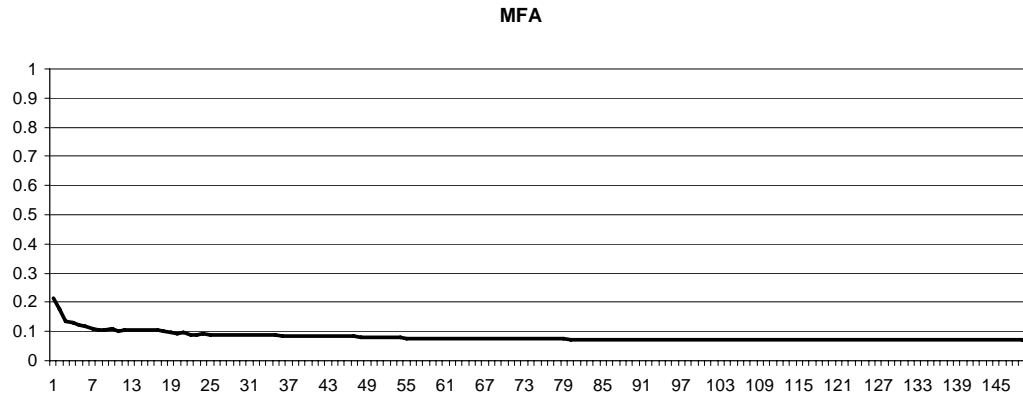


Figura 65 – Gráfico de monitoramento da métrica MFA durante a evolução

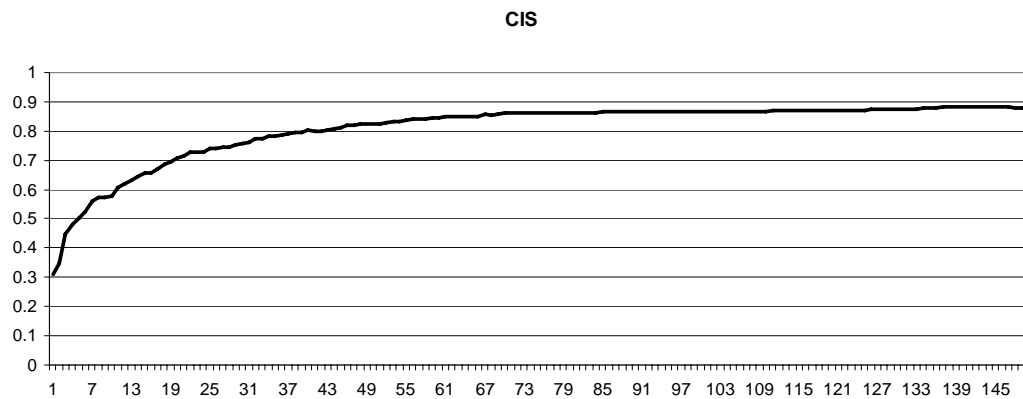


Figura 66 – Gráfico de monitoramento da métrica CIS durante a evolução

Nota-se que a métrica CIS evolui bastante, enquanto a NOH permanece constante, já que não influencia no atributo de *Reutilização*. As outras chegam a um equilíbrio rapidamente.

$$Flexibilidade = 0.25 *DAM - 0.25 *DCC + 0.5 * MOA + 0.5 *NOP$$

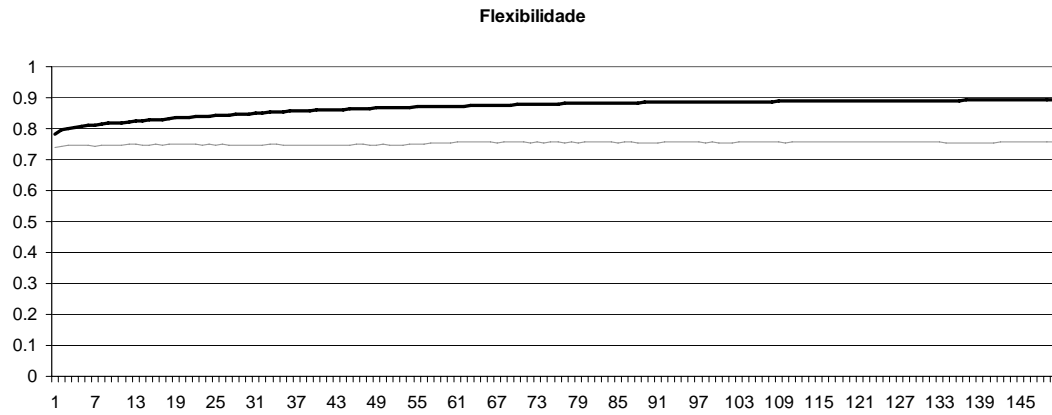


Figura 67 – Gráfico de desempenho para o atributo *Flexibilidade*

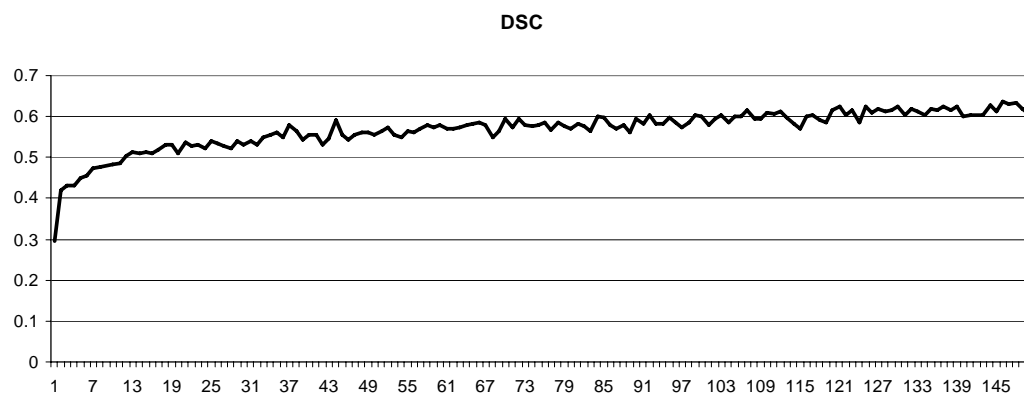


Figura 68 – Gráfico de monitoramento da métrica DSC durante a evolução

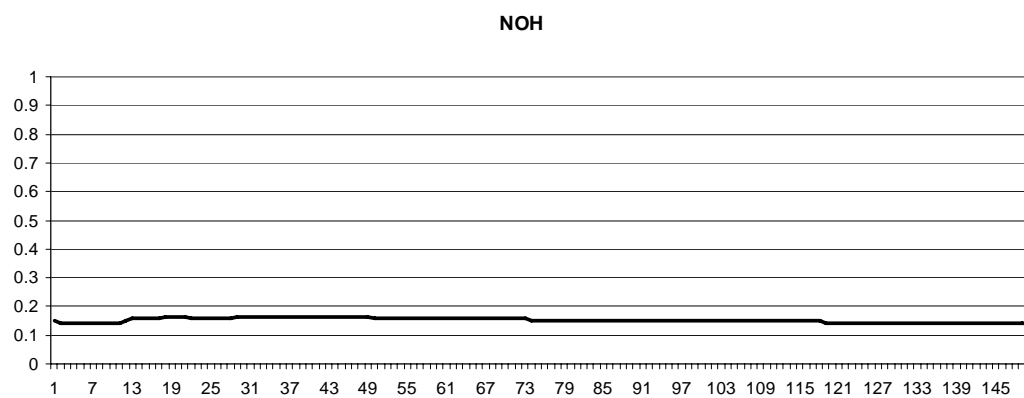


Figura 69 – Gráfico de monitoramento da métrica NOH durante a evolução

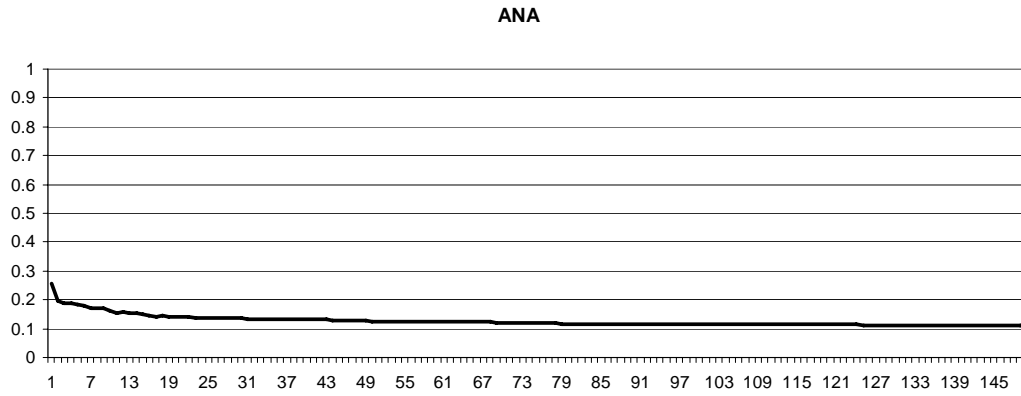


Figura 70 – Gráfico de monitoramento da métrica ANA durante a evolução

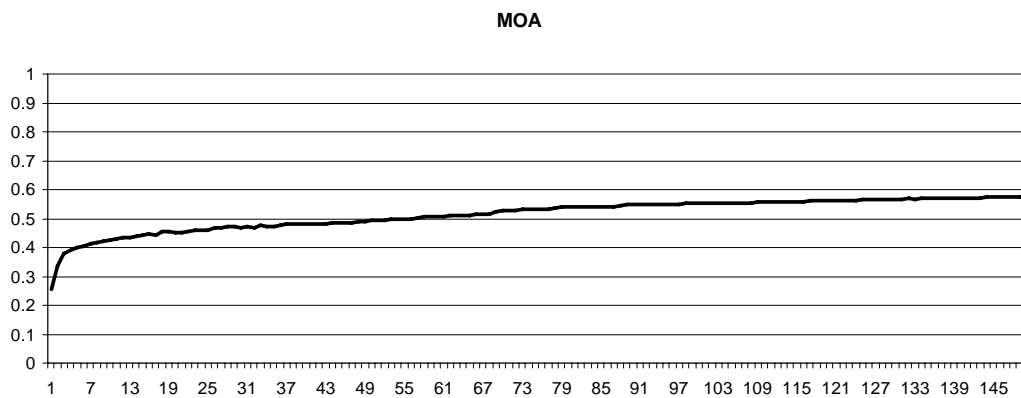


Figura 71 – Gráfico de monitoramento da métrica MOA durante a evolução

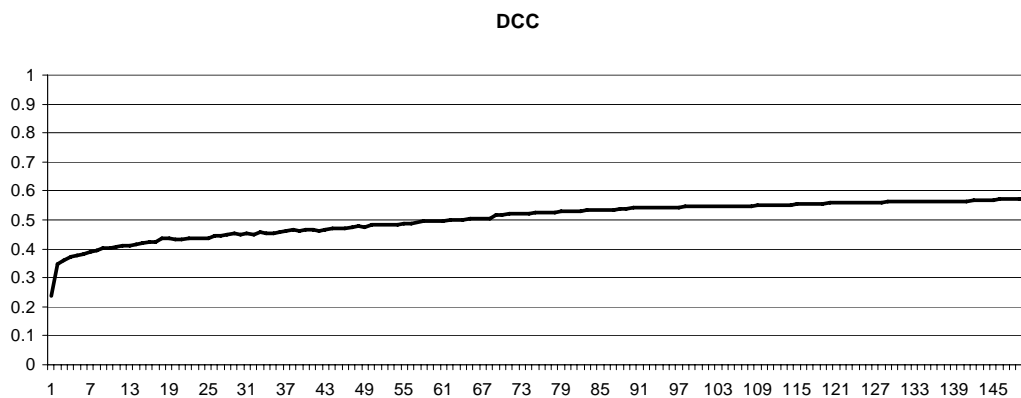


Figura 72 – Gráfico de monitoramento da métrica DCC durante a evolução

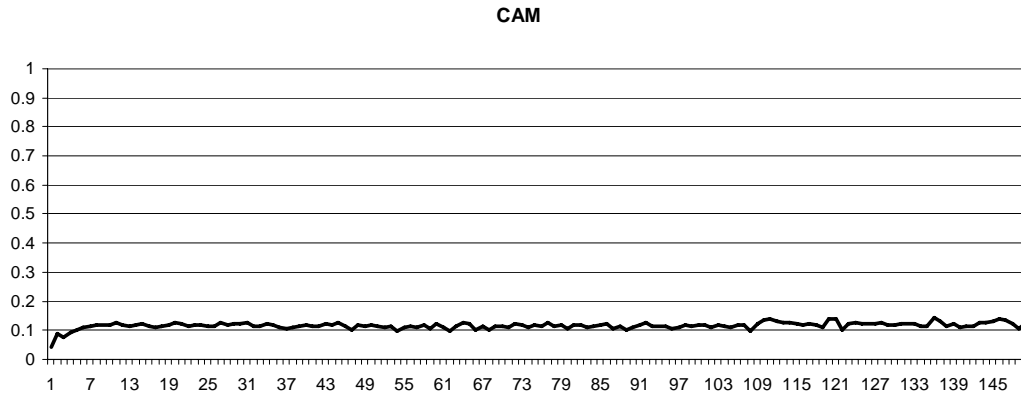


Figura 73 – Gráfico de monitoramento da métrica CAM durante a evolução

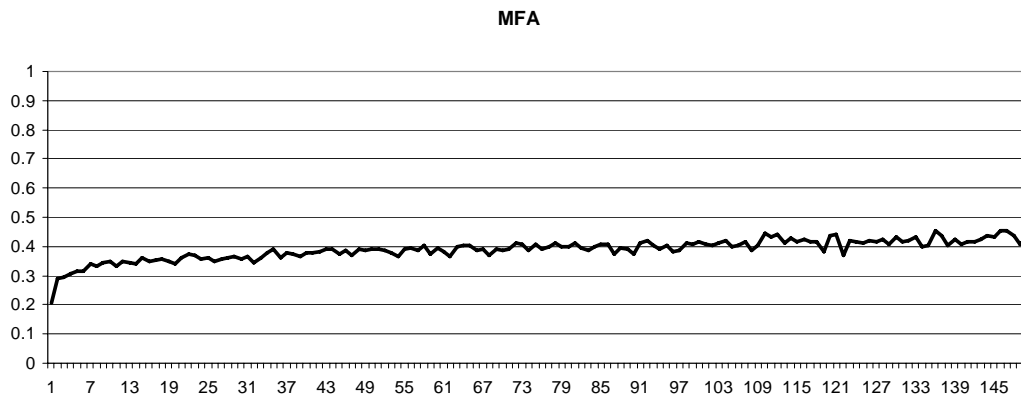


Figura 74 – Gráfico de monitoramento da métrica MFA durante a evolução

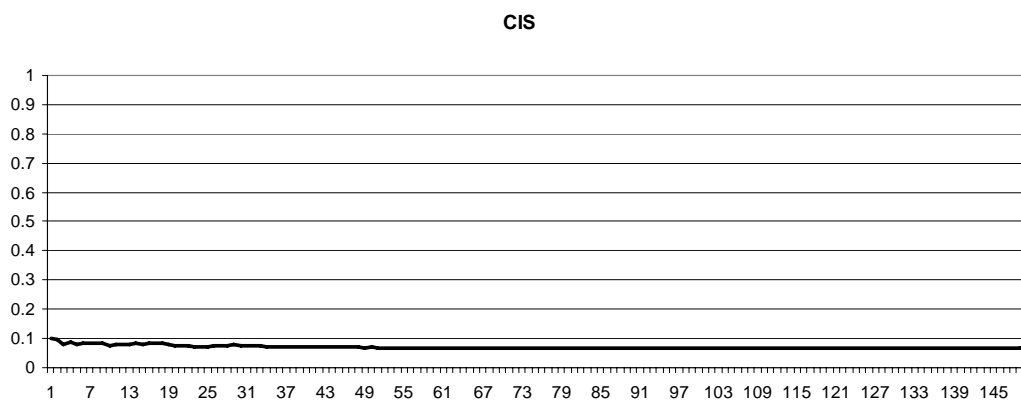


Figura 75 – Gráfico de monitoramento da métrica CIS durante a evolução

As diferenças entre os atributos *Reutilização* e *Flexibilidade* ficam claras quando comparam-se os gráficos das métricas durante a evolução, o AG procura

evoluir principalmente as métricas que formam a equação do atributo de qualidade, embora algumas entrem em conflito com outras e não evoluam bem.

$$\text{Inteligibilidade} = -0.33 * \text{ANA} + 0.33 * \text{DAM} - 0.33 * \text{DCC} + 0.33 * \text{CAM} - 0.33 * \text{NOP} - 0.33 * \text{CIS} - 0.33 * \text{DSC}$$

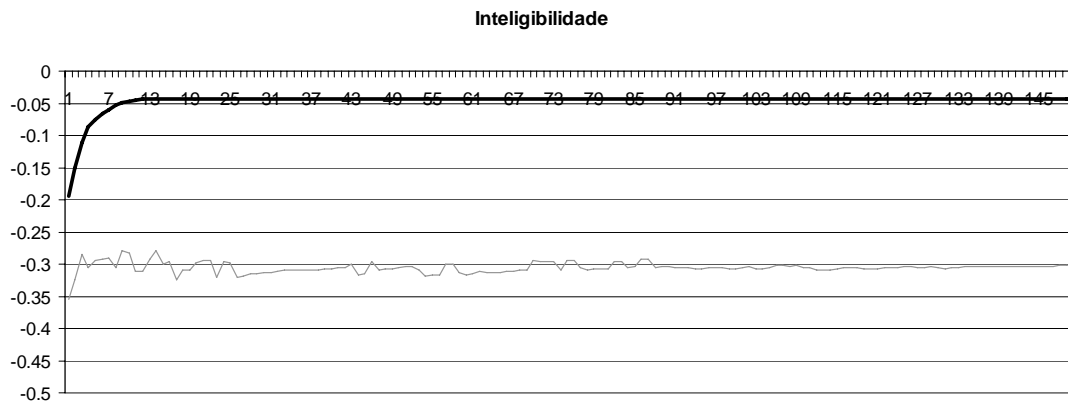


Figura 76 – Gráfico de desempenho para o atributo *Inteligibilidade*

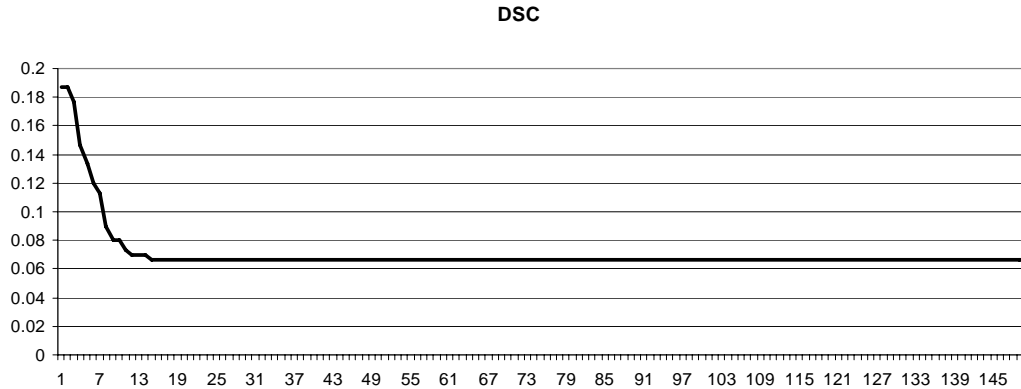


Figura 77 – Gráfico de monitoramento da métrica DSC durante a evolução

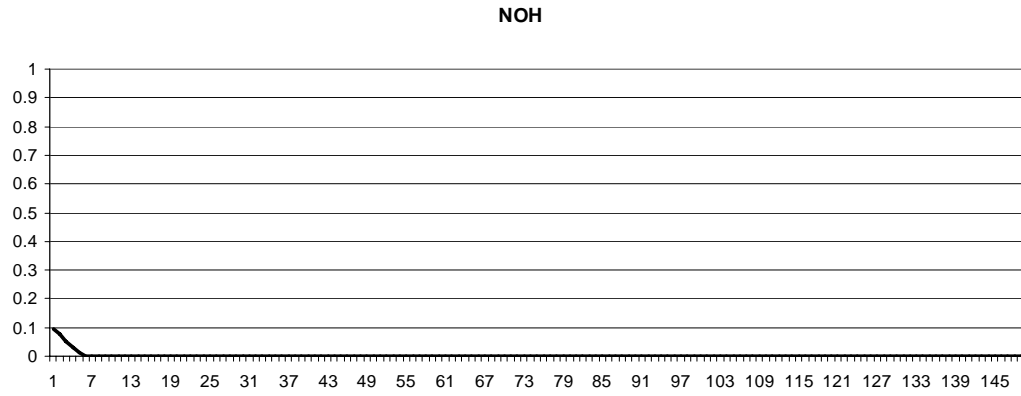


Figura 78 – Gráfico de monitoramento da métrica NOH durante a evolução

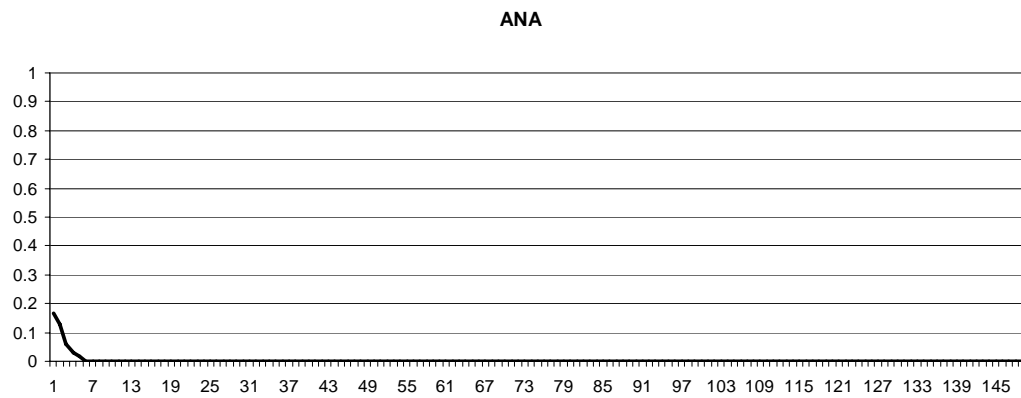


Figura 79 – Gráfico de monitoramento da métrica ANA durante a evolução

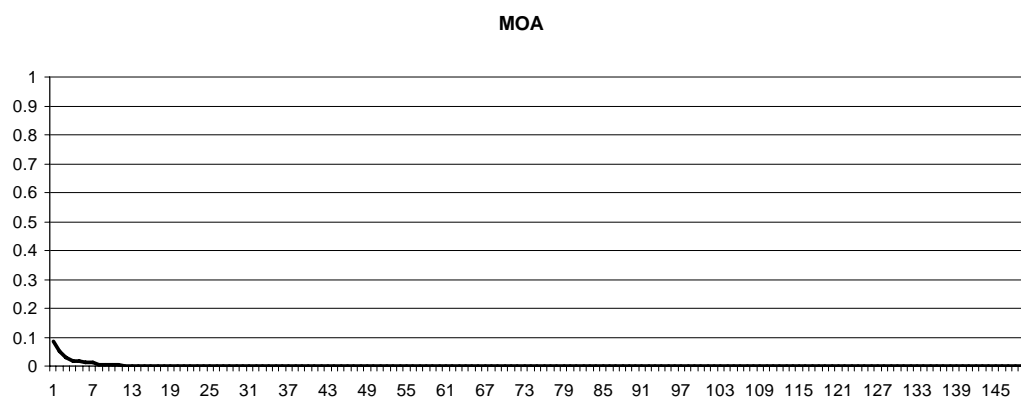


Figura 80 – Gráfico de monitoramento da métrica MOA durante a evolução

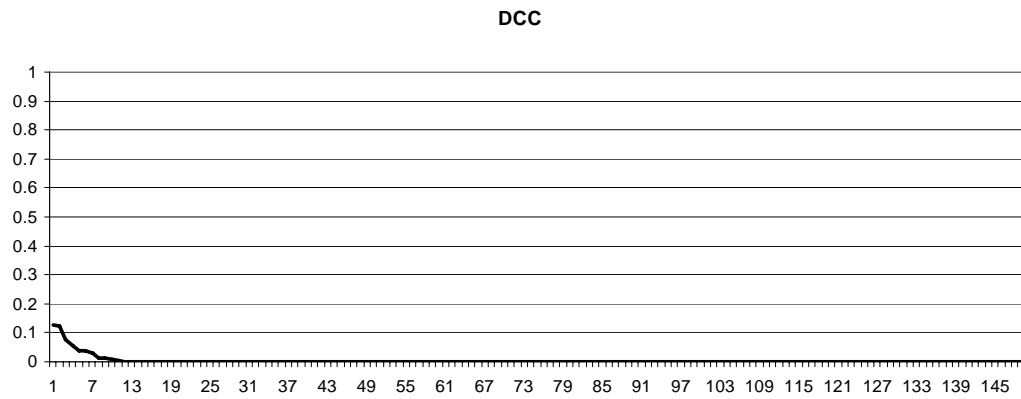


Figura 81 – Gráfico de monitoramento da métrica DCC durante a evolução

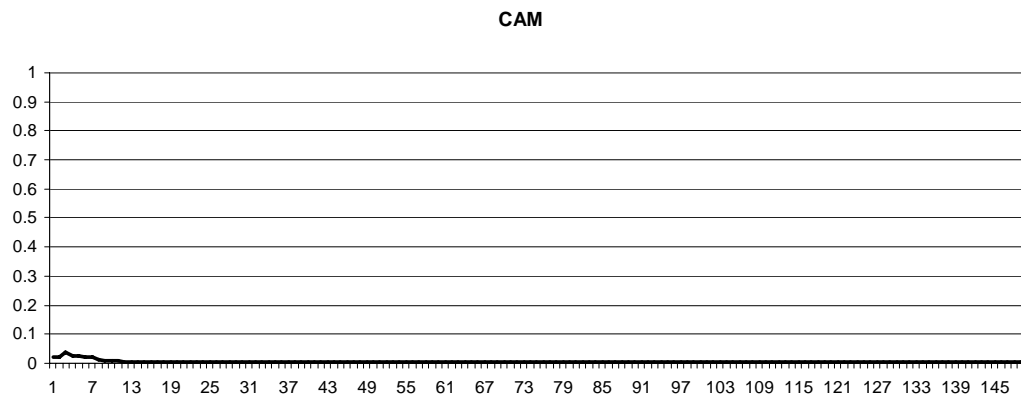


Figura 82 – Gráfico de monitoramento da métrica CAM durante a evolução

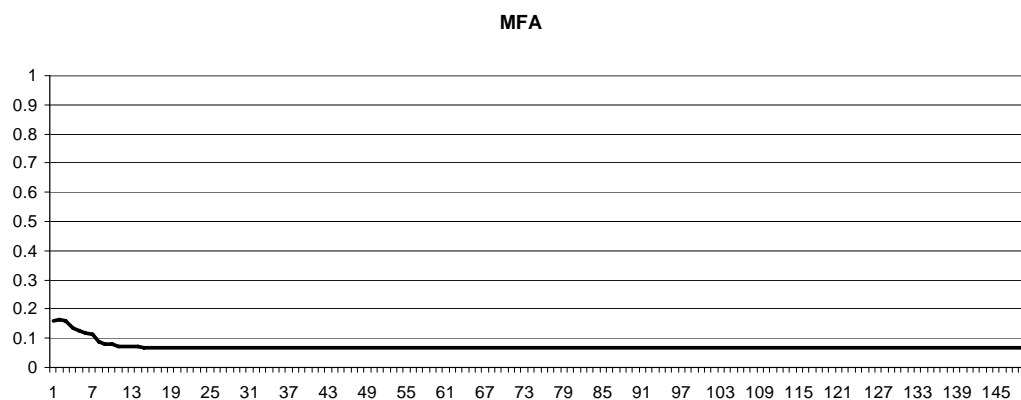


Figura 83 – Gráfico de monitoramento da métrica MFA durante a evolução

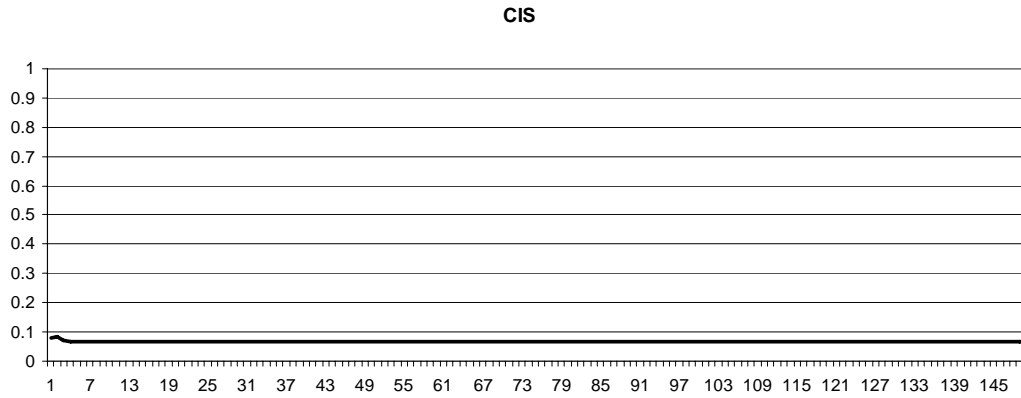


Figura 84 – Gráfico de monitoramento da métrica CIS durante a evolução

É interessante o resultado obtido para o atributo *Inteligibilidade*, onde a maioria das métricas são levadas a zero. Pode-se verificar uma grande redução no número de classes, a partir da métrica *DSC*.

$$\text{Funcionalidade} = 0.12 * \text{CAM} + 0.22 * \text{NOP} + 0.22 * \text{CIS} + 0.22 * \text{DSC} + 0.22 * \text{NOH}$$

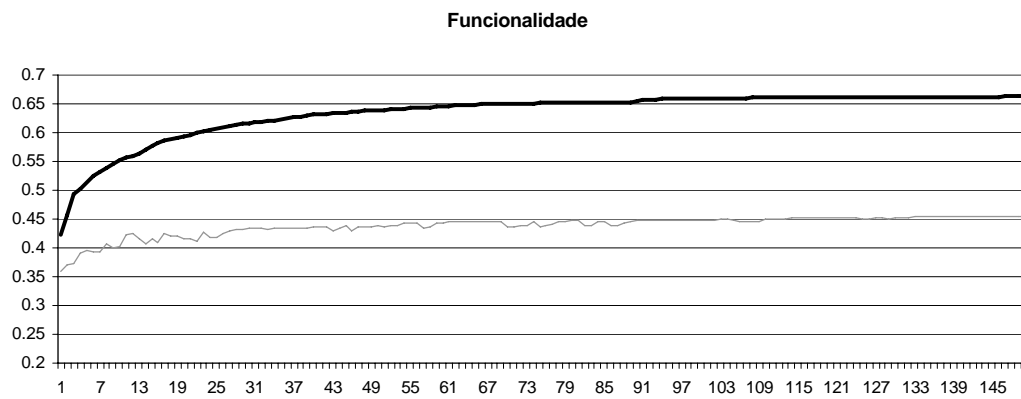


Figura 85 – Gráfico de desempenho para o atributo *Funcionalidade*

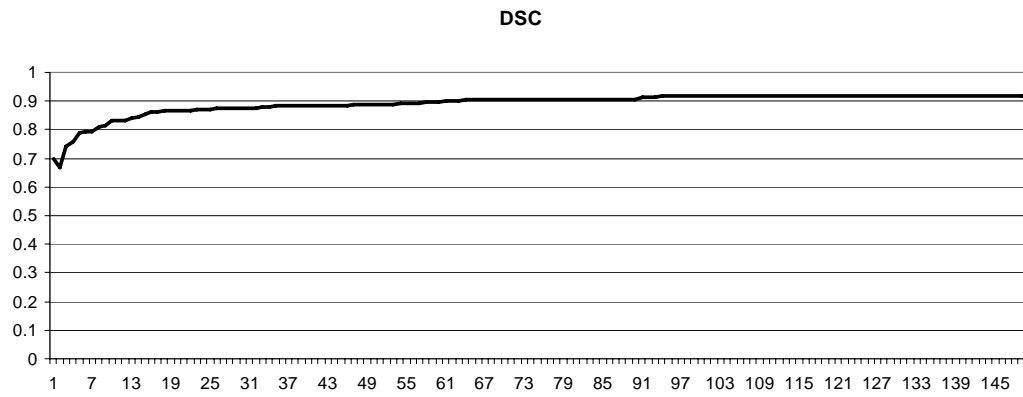


Figura 86 – Gráfico de monitoramento da métrica DSC durante a evolução

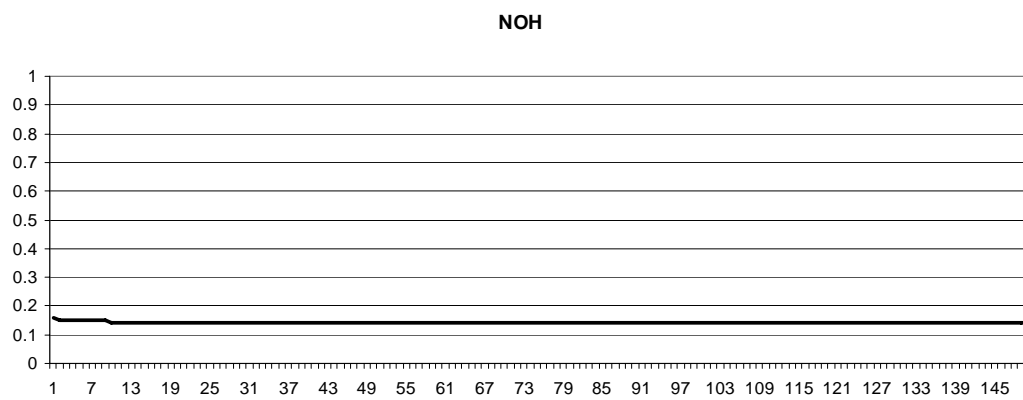


Figura 87 – Gráfico de monitoramento da métrica NOH durante a evolução

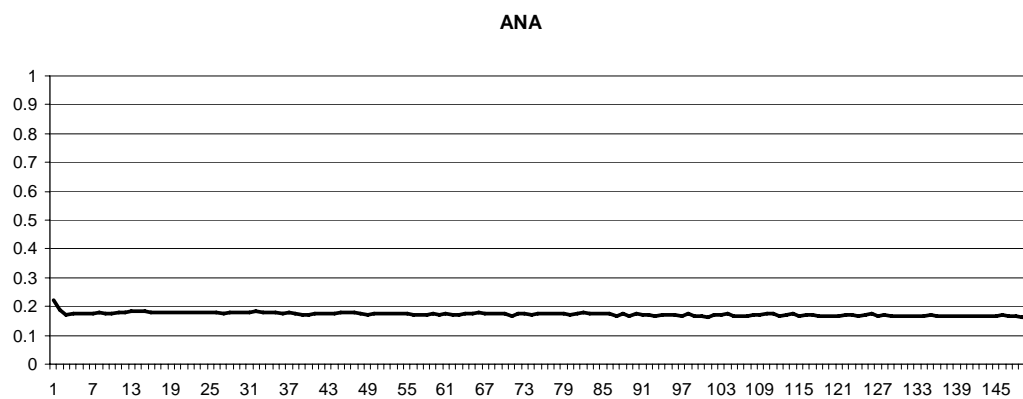


Figura 88 – Gráfico de monitoramento da métrica ANA durante a evolução

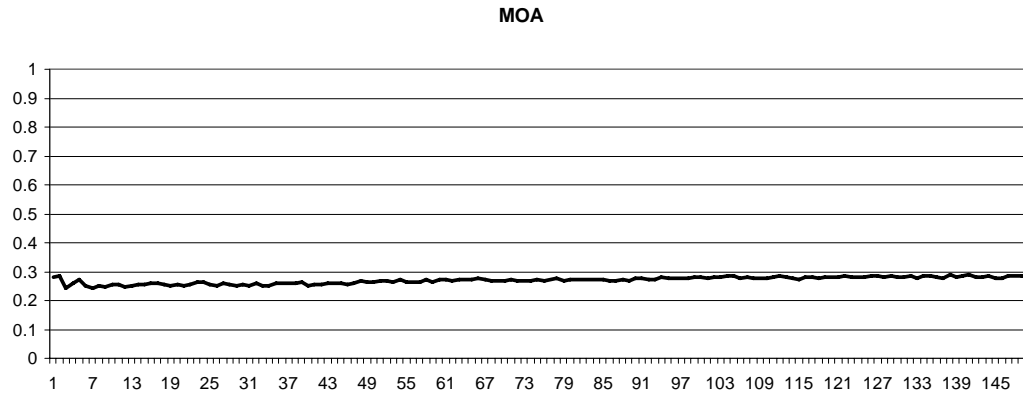


Figura 89 – Gráfico de monitoramento da métrica MOA durante a evolução

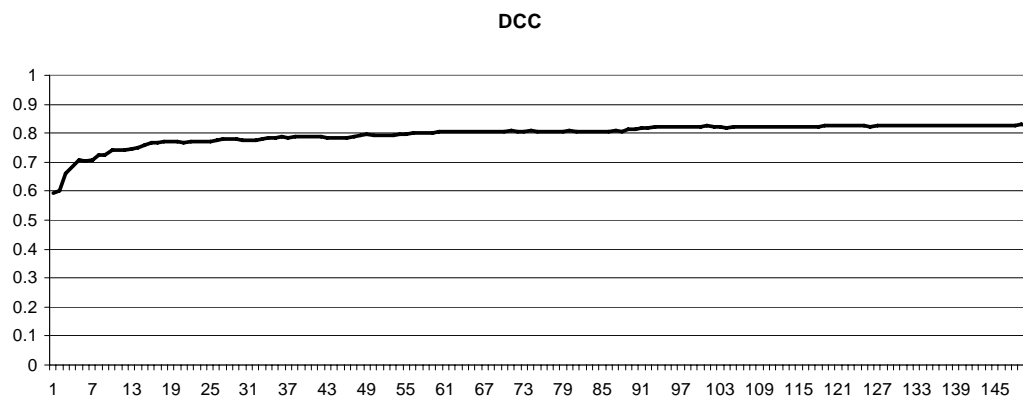


Figura 90 – Gráfico de monitoramento da métrica DCC durante a evolução

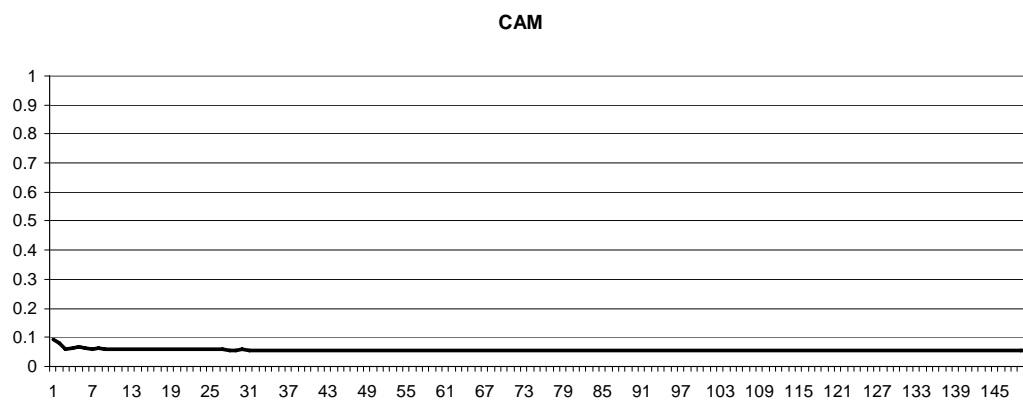


Figura 91 – Gráfico de monitoramento da métrica CAM durante a evolução

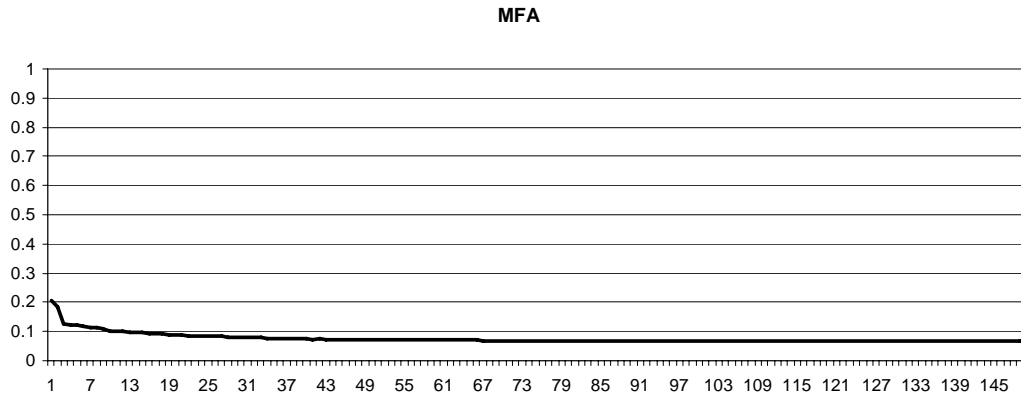


Figura 92 – Gráfico de monitoramento da métrica MFA durante a evolução

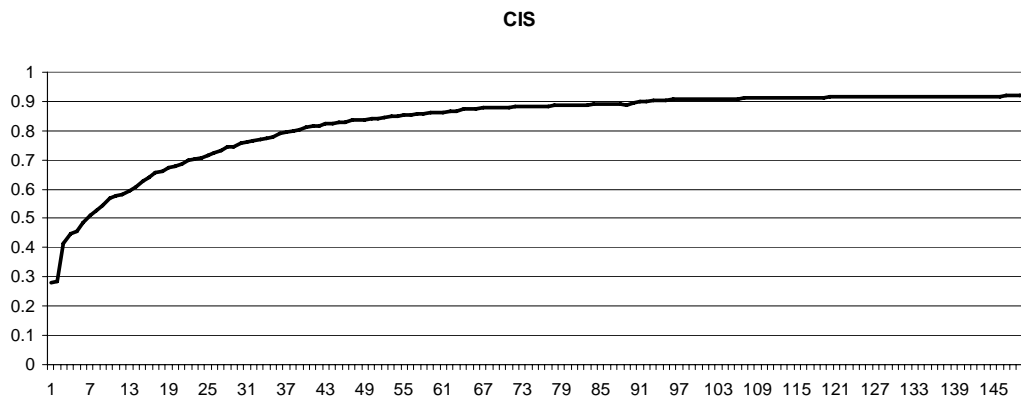


Figura 93 – Gráfico de monitoramento da métrica CIS durante a evolução

Assim como a *Reutilização*, o atributo de *Funcionalidade* provoca uma grande evolução na métrica *CIS*. Enquanto que na evolução da *Inteligibilidade* essa métrica permanece constante e baixa, isso prova o conflito existente entre os atributos de qualidade.

$$\textit{Extensibilidade} = 0.5 * \textit{ANA} - 0.5 * \textit{DCC} + 0.5 * \textit{MFA} + 0.5 * \textit{NOP}$$

Extensibilidade

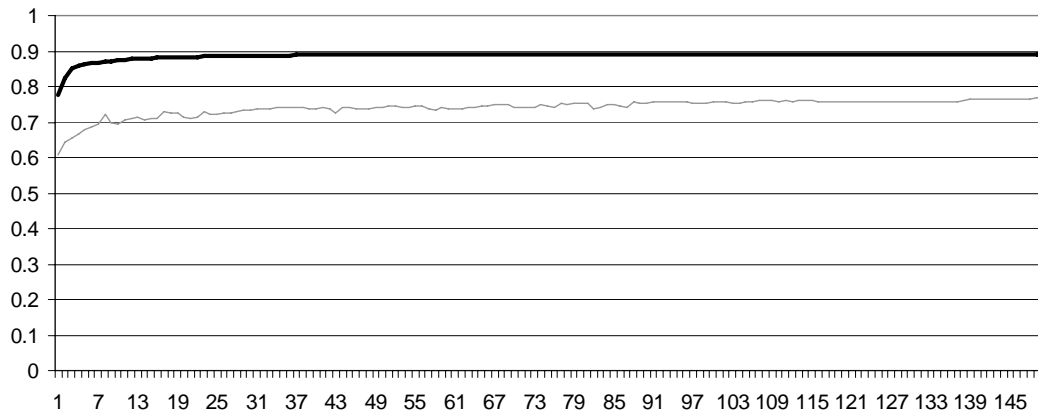


Figura 94 – Gráfico de desempenho para o atributo *Extensibilidade*

DSC

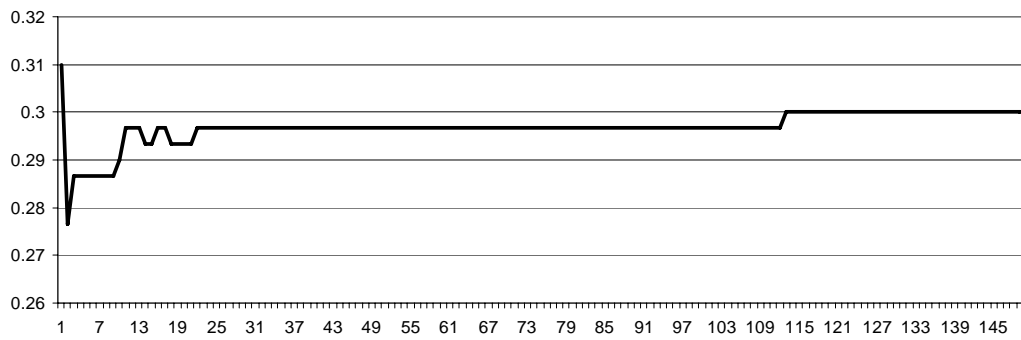


Figura 95 – Gráfico de monitoramento da métrica DSC durante a evolução

NOH

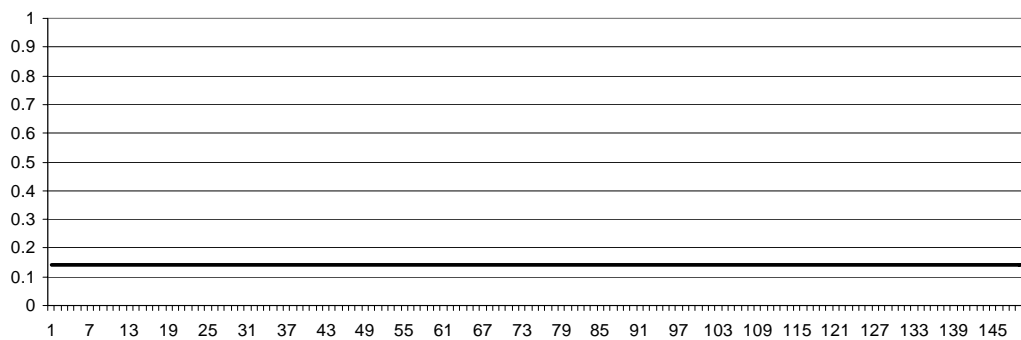


Figura 96 – Gráfico de monitoramento da métrica NOH durante a evolução

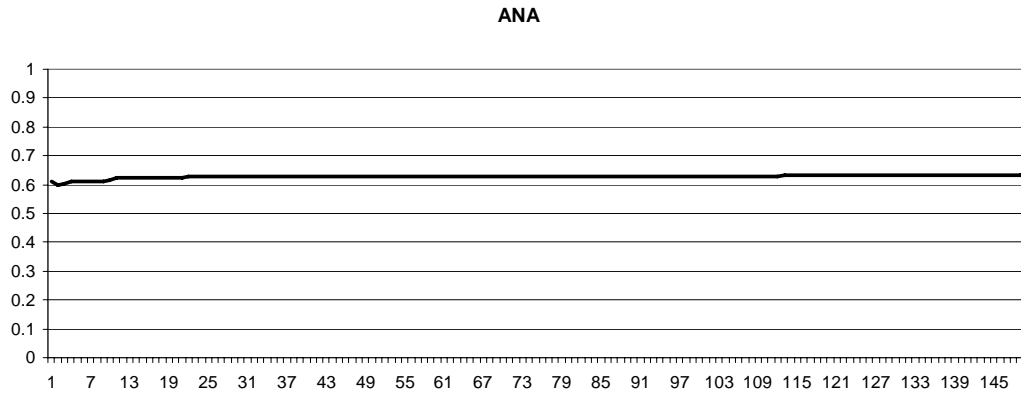


Figura 97 – Gráfico de monitoramento da métrica ANA durante a evolução

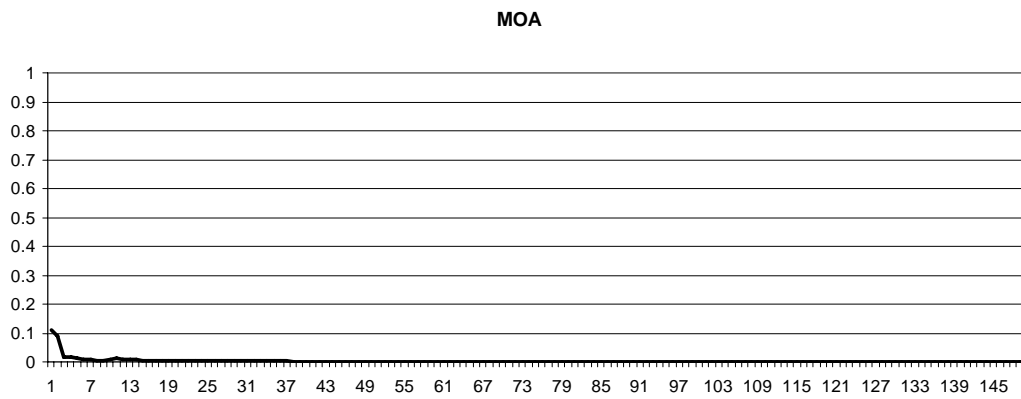


Figura 98 – Gráfico de monitoramento da métrica MOA durante a evolução

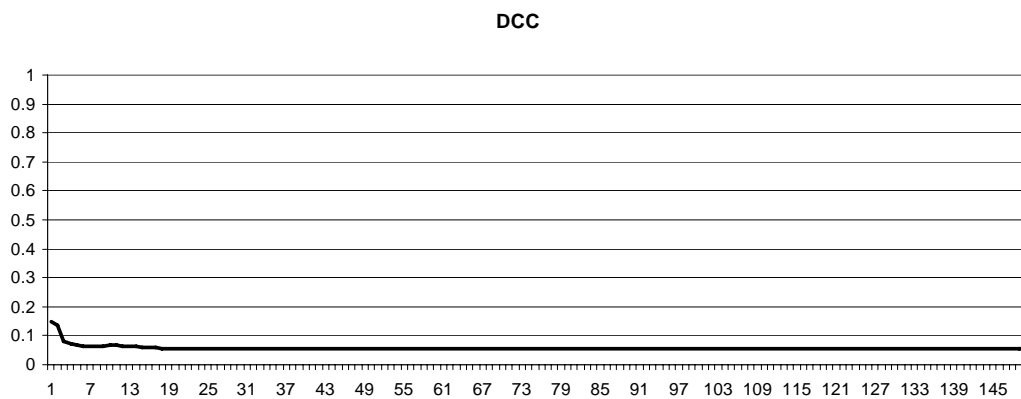


Figura 99 – Gráfico de monitoramento da métrica DCC durante a evolução

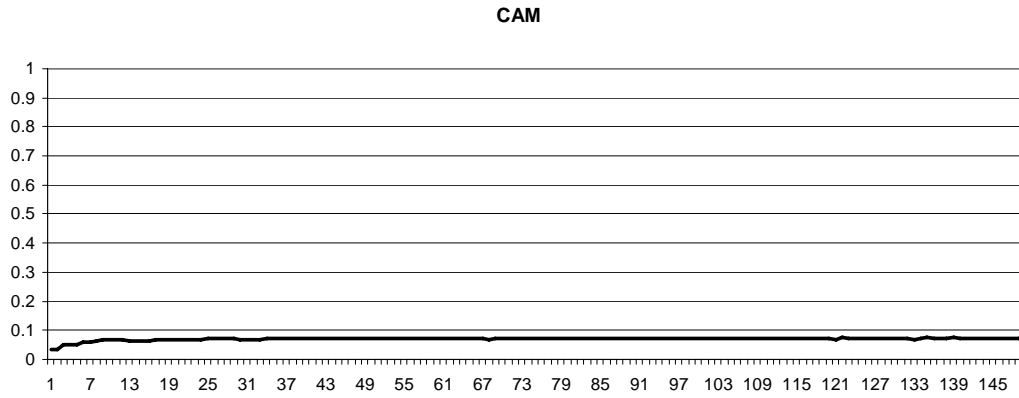


Figura 100 – Gráfico de monitoramento da métrica CAM durante a evolução

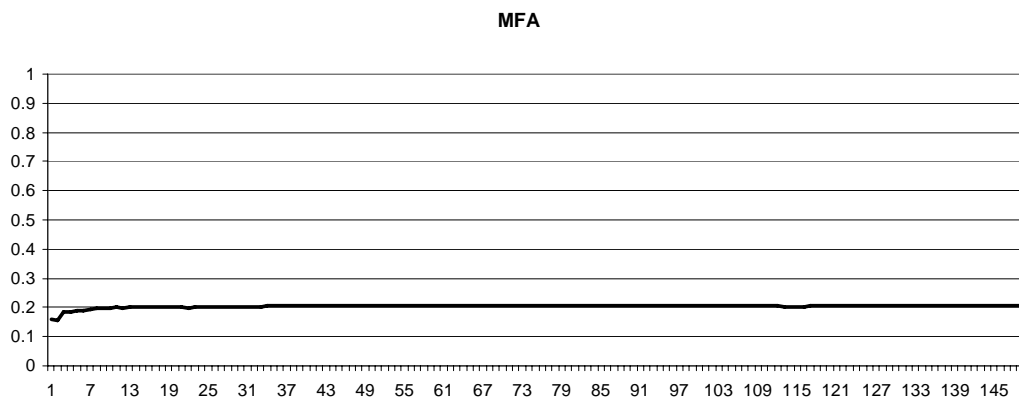


Figura 101 – Gráfico de monitoramento da métrica MFA durante a evolução

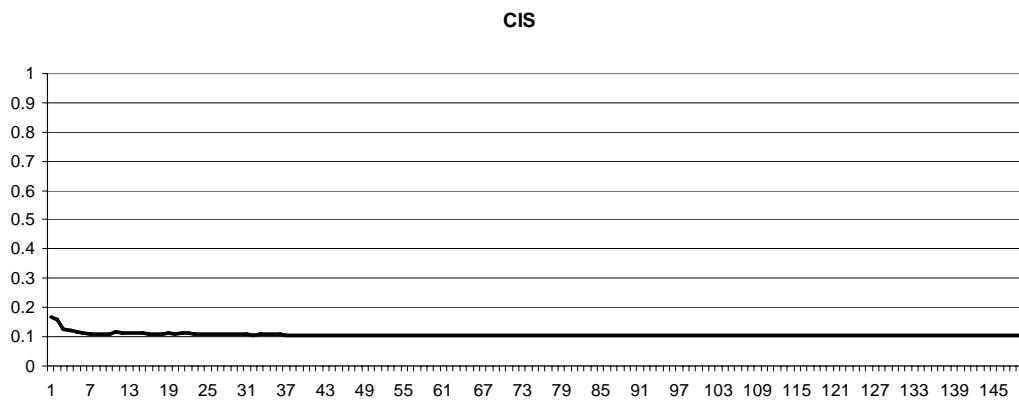


Figura 102 – Gráfico de monitoramento da métrica CIS durante a evolução

Esse atributo se caracterizou por uma grande oscilação na métrica *DSC*, ou seja, ocorreram grandes variações no número de classes na evolução. O alto valor para a métrica *ANA* representa um grande uso de herança, o que já tinha sido

percebido nos exemplos realizados no primeiro estudo de caso. As outras métricas também convergem para valores baixos.

$$Efetividade = 0.2 * ANA + 0.2 * DAM + 0.2 * MOA + 0.2 * MFA + 0.2 * NOP$$

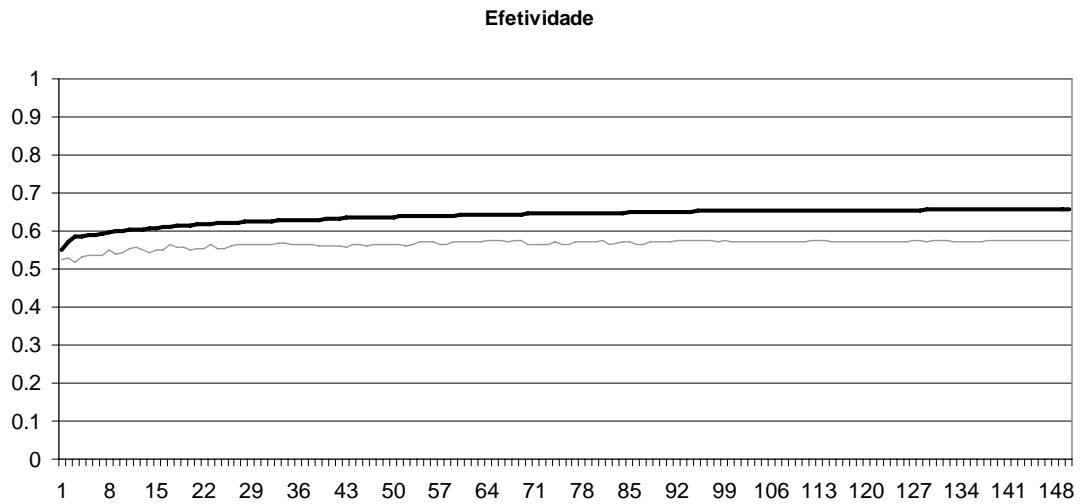


Figura 103 – Gráfico de desempenho para o atributo *Efetividade*

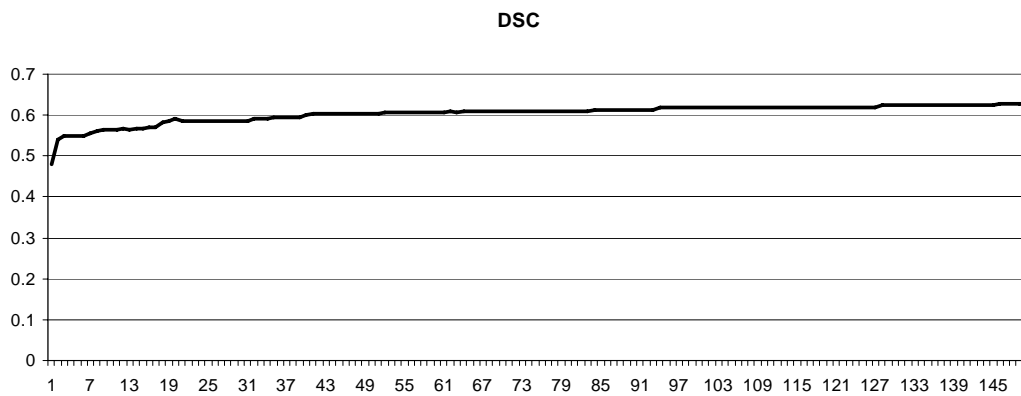


Figura 104 – Gráfico de monitoramento da métrica DSC durante a evolução

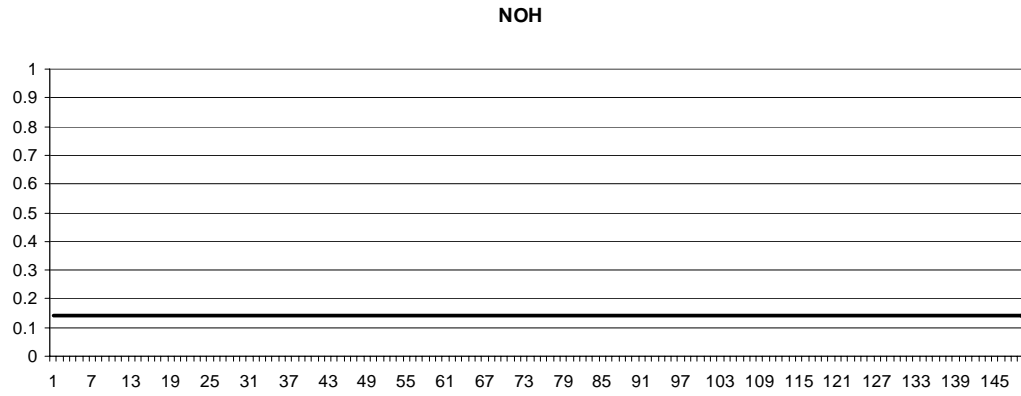


Figura 105 – Gráfico de monitoramento da métrica NOH durante a evolução

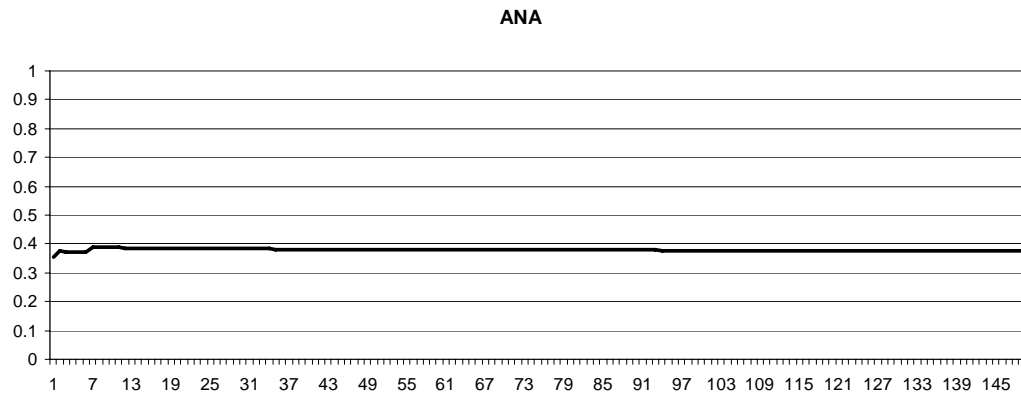


Figura 106 – Gráfico de monitoramento da métrica ANA durante a evolução

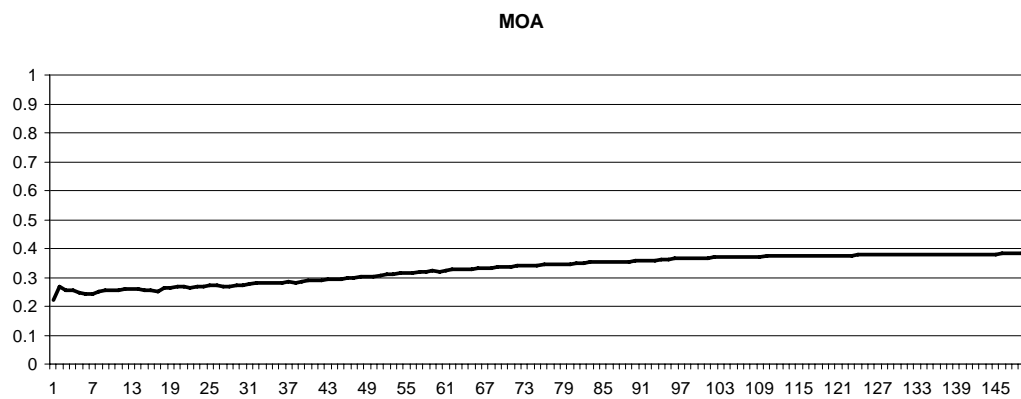


Figura 107 – Gráfico de monitoramento da métrica MOA durante a evolução

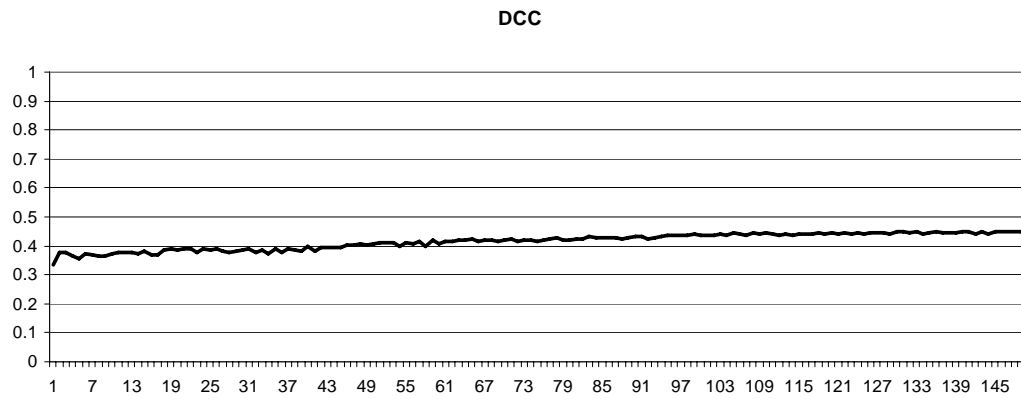


Figura 108 – Gráfico de monitoramento da métrica DCC durante a evolução

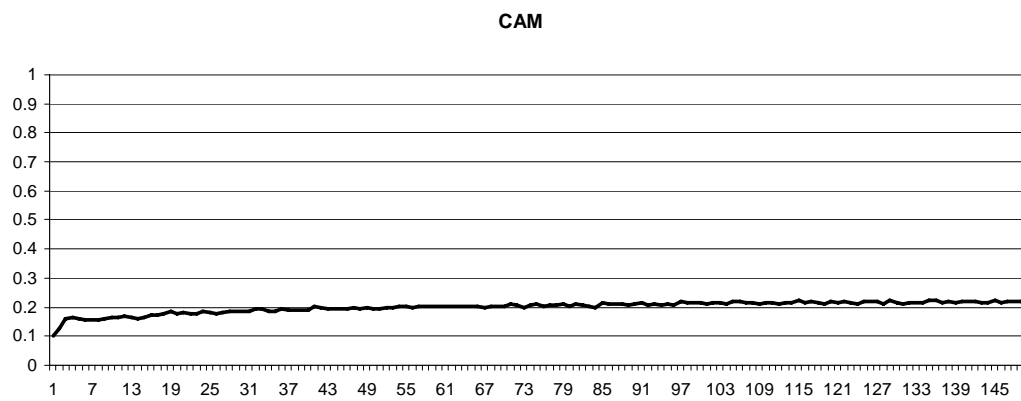


Figura 109 – Gráfico de monitoramento da métrica CAM durante a evolução

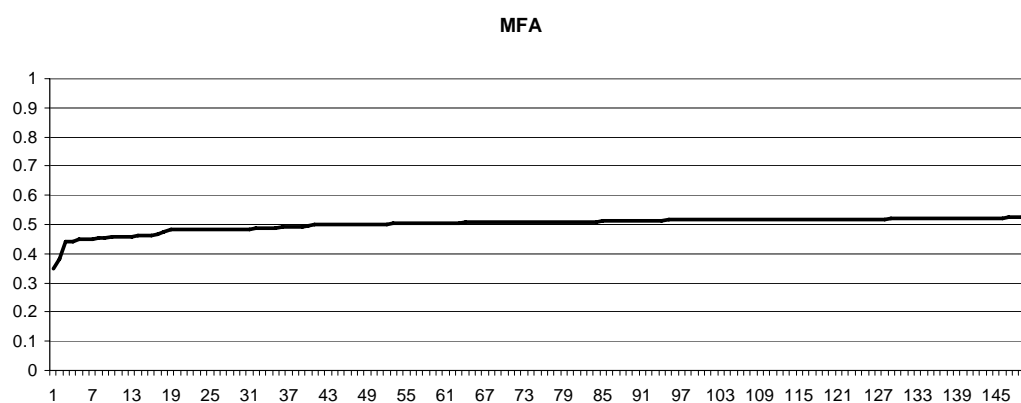


Figura 110 – Gráfico de monitoramento da métrica MFA durante a evolução

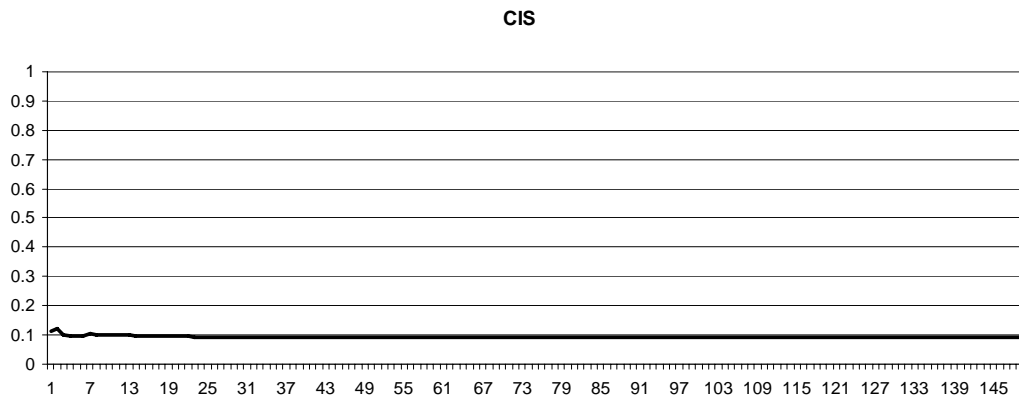


Figura 111 – Gráfico de monitoramento da métrica CIS durante a evolução

Novamente na *Efetividade* nota-se a evolução principalmente nas métricas que formam a equação do atributo.

Numa análise geral, percebe-se que a métrica NOH (*Number of Hierarchies*) sempre converge para valores baixos, e na maioria dos atributos ele permanece constante. Isso se deve ao fato dele ser muito prejudicial às outras métricas. Analisando as equações, ele somente é utilizado no cálculo da *Funcionalidade*. Num teste de evolução considerando somente a métrica NOH, obteve-se os gráficos abaixo para as métricas.

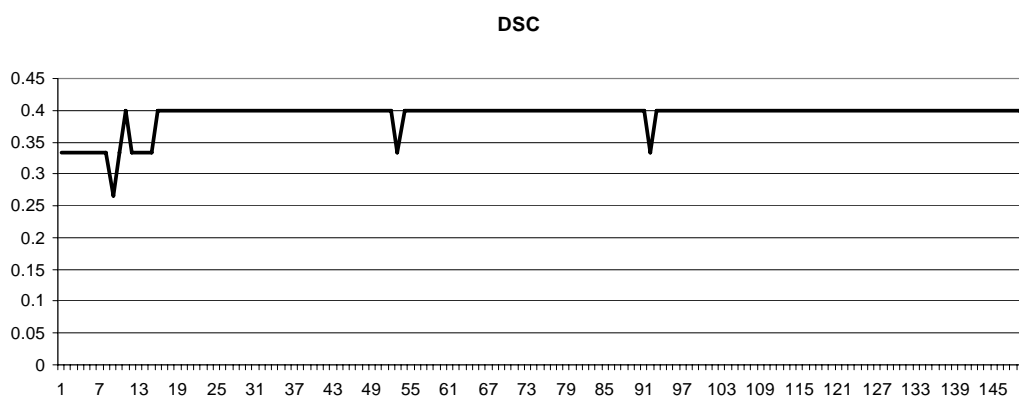


Figura 112 – Gráfico de evolução da métrica NOH separadamente. Métrica DSC

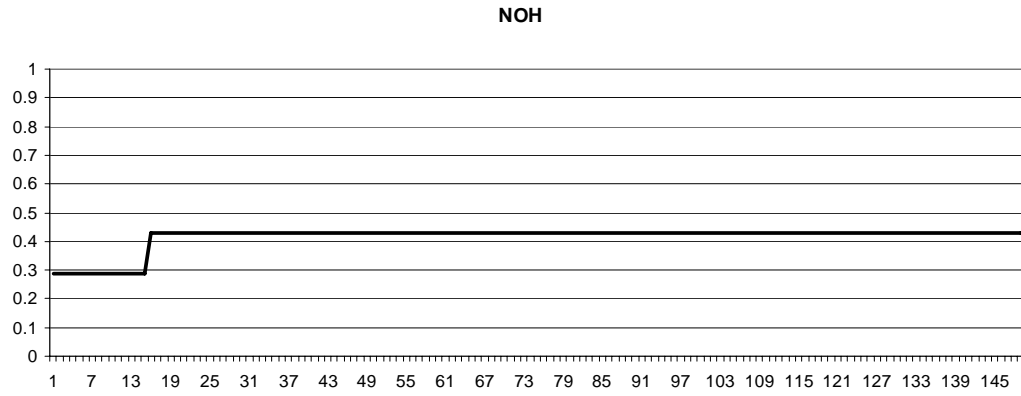


Figura 113 – Gráfico de evolução da métrica NOH separadamente. Métrica NOH

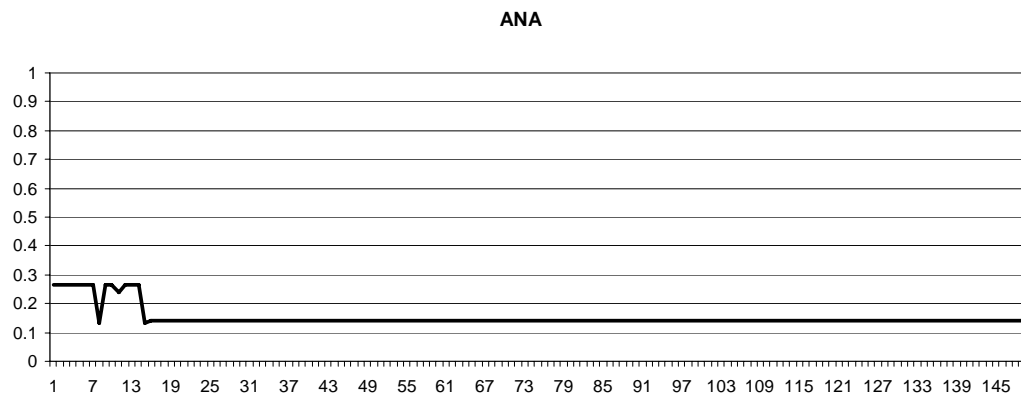


Figura 114 – Gráfico de evolução da métrica NOH separadamente. Métrica ANA

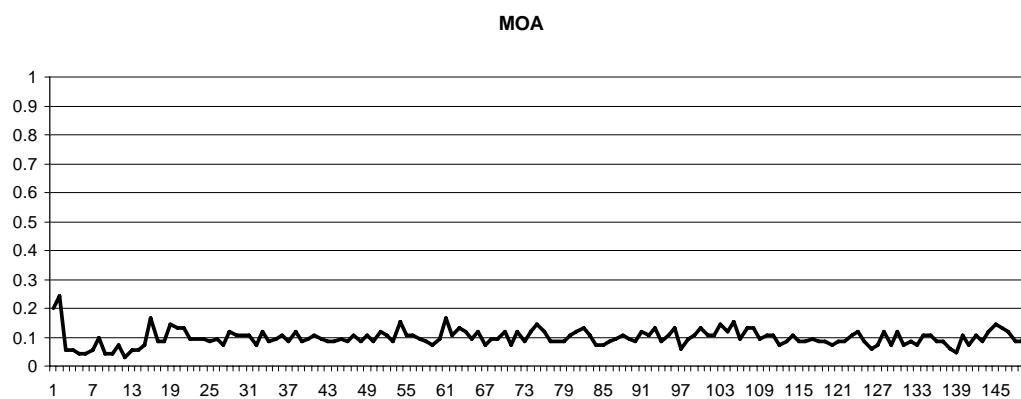


Figura 115 – Gráfico de evolução da métrica NOH separadamente. Métrica MOA

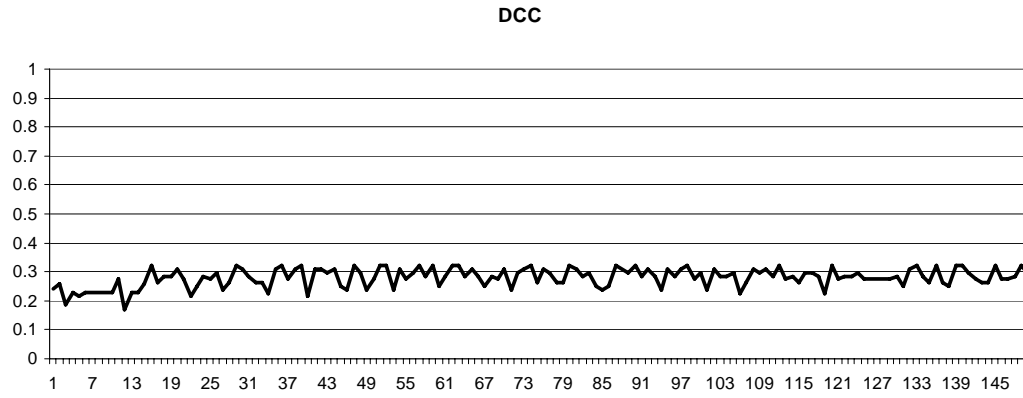


Figura 116 – Gráfico de evolução da métrica NOH separadamente. Métrica DCC

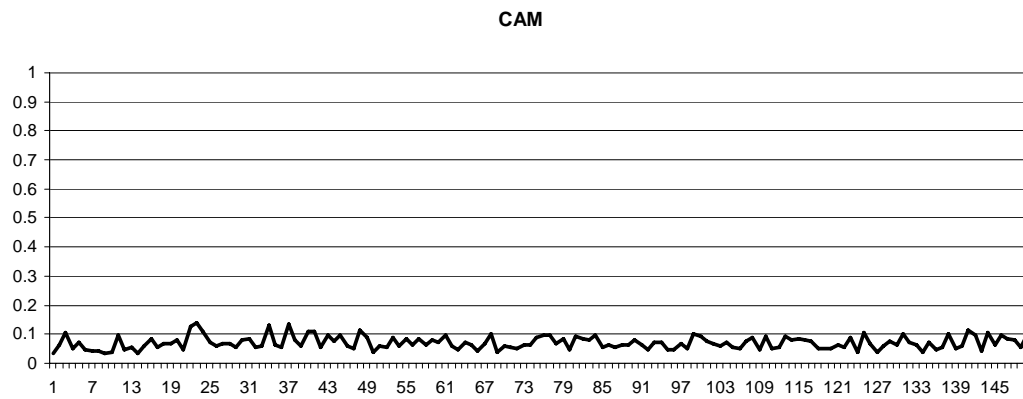


Figura 117 – Gráfico de evolução da métrica NOH separadamente. Métrica CAM

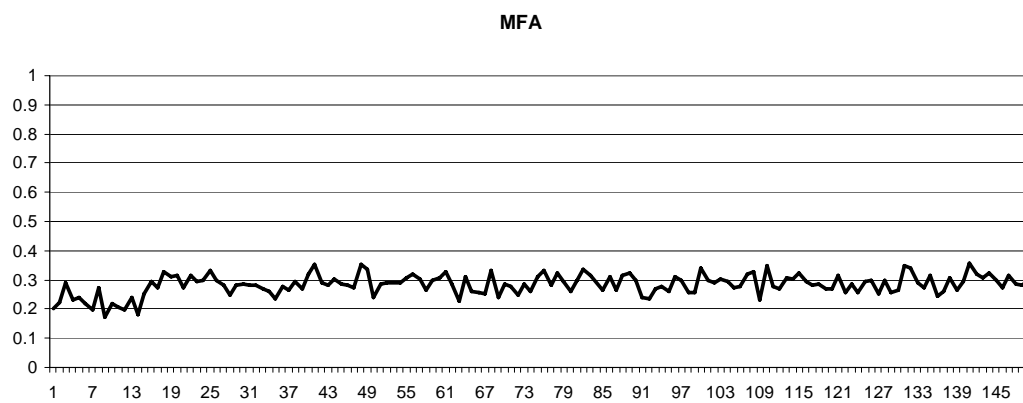


Figura 118 – Gráfico de evolução da métrica NOH separadamente. Métrica MFA

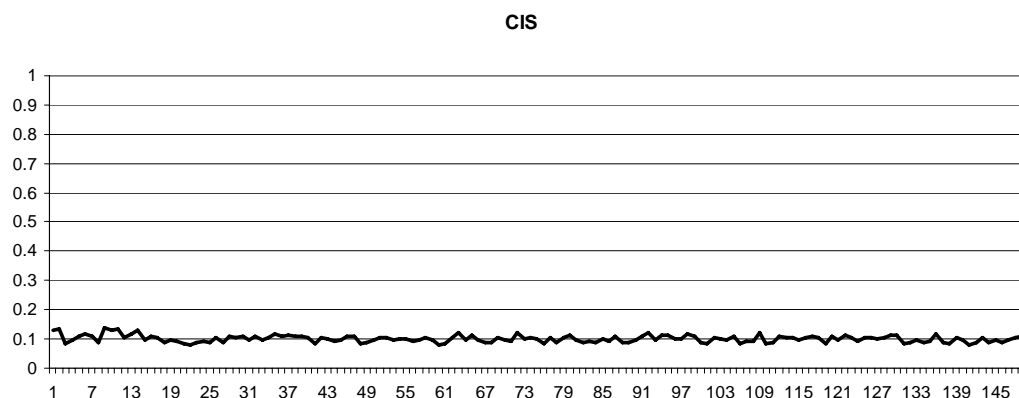


Figura 119 – Gráfico de evolução da métrica NOH separadamente. Métrica CIS

As Figuras acima mostram que com a evolução da métrica *NOH* as outras métricas permanecem oscilando entre valores baixos, com exceção do *DSC*, que reflete o número de classes no sistema, favorável ao *NOH*. Essa análise prova que a métrica *NOH* realmente não evolui junto com as outras devido aos conflitos entre elas, e assim, o AG a mantém constante e com um valor baixo para poder aumentar as outras.

Observa-se também uma substancial melhora da otimização quando o modelo é executado para otimizar cada objetivo separadamente. Porém, no caso de atributos como *Flexibilidade* e *Efetividade* a otimização não foi tão superior, melhorando 15% e 11% respectivamente, ao contrário da *Reutilização* e *Inteligibilidade*, por exemplo, que obtiveram uma melhora de 41% e 618% respectivamente.