# ALGORITMOS GENÉTICOS NA OTIMIZAÇÃO DA LAPIDAÇÃO DE GEMAS CORADAS

#### GENETIC ALGORITHMS IN GEMSTONE CUTTING OPTIMIZATION

### Maurício Berthier Fiorest 1, Marcos José Brusso 2

Instituto de Ciências Exatas e Geociências – Curso de Ciência da Computação Universidade de Passo Fundo – UPF, RS E-mail: ¹deatlock@gmail.com; ²brusso@upf.br

#### **RESUMO**

Este artigo descreve o uso de algoritmos genéticos na otimização da lapidação de gemas, aprimorando o projeto 3D Gemas existente ao reduzir a interação humana. Será descrito como o algoritmo será usado para evoluir soluções formadas por conjuntos de pontos na gema digitalizada buscando alcançar o modelo com melhor aproveitamento do volume da gema.

Palavras Chave: Inteligência Artificial, Algoritmos Genéticos, Otimização, Lapidação, GALIB

#### **ABSTRACT**

This paper describes the use of genetic algorithms in the optimization of gemstone cutting, improving the existing 3D Gemas Project by reducing human intervention. It will be described how the algorithm will be used to evolve solutions formed by groups of points in the digitalized gem trying to reach a cutting model with the best gemstone volume usage.

Keywords: Artificial Intelligence, Genetic Algorithms, Optimization, Gemstone Cutting, GALIB

# 1. INTRODUÇÃO

A lapidação de gemas, processo necessário para realçar os atributos visuais e consequentemente o valor agregado da mesma com finalidade de uso em joalheria, no Brasil ainda segue padrões e métodos antiquados trabalhando praticamente de forma manual. Isso acarreta em um aproveitamento baixo da gema com risco de deixar imperfeições que resultam em perda de seu valor. Em contraste com métodos mais avançados empregados por empresas estrangeiras representa uma grande desvantagem ao mercado de joias do pais (Aspaham, 2006, p. 38).

O projeto 3D Gemas fornece um conjunto de ferramentas para auxiliar empresas do setor através do uso de tecnologias para geração de modelos de lapidação otimizados dando a essas empresas um diferencial competitivo.

[...] a Tecnologia 3D Gemas é um conjunto de técnicas e ferramentas desenvolvidas com o objetivo de auxiliar o processo de lapidação facetada de gemas coradas, visando encontrar para cada gema o(s) projeto(s) de lapidação que resultem em maior valor agregado, considerando o aproveitamento do volume da gema (peso) e brilho (lapidação) da pedra lapidada resultante. (Brusso et all, 2010 42)

O software, ainda sendo aperfeiçoado, já permite observar as vantagens do uso da tecnologia nessa área e o uso de inteligência artificial permitirá reduzir a interação humana, garantindo assim uma maior automação de todo o processo.

doi: 10.5335/ciatec.v3i2.1904 44

O projeto 3D gemas possui dois programas que funcionam em conjunto o *Otimizador 3D Gemas* e o *Visualizador 3D Gemas*. O otimizador é o foco desse artigo e será mais profundamente descrito a seguir. O visualizador exibe os dados calculados pelo otimizador mostrando para cada modelo o melhor aproveitamento encontrado, possibilitando a empresa responsável optar por um modelo e imprimir o diagrama de lapidação que será entregue ao lapidário.

#### 2. OTIMIZADOR 3D GEMAS

O otimizador 3D gemas é o programa que analisa virtualmente a gema digitalizada testando a inserção de modelos de lapidação em busca de um melhor aproveitamento do seu volume. O otimizador, em sua primeira versão, segue o algoritmo abaixo:

```
1. gema = lerArquivoSTL()
2. pontosCriticos = lerArquivoPontosCriticos()
3. modelosDisponiveis = [Redondo8, Oval8, ...]
5. Para Cada Combinação de 3 pontos distintos {i, j, k} em pontosCriticos
      p = \text{criarPlano}(i, j, k)
7.
      c = \text{extrairContorno}(gema, p)
      Para Cada Modelo m em modelosDisponiveis
9.
          maior = encontrarMaiorCinturaInscrita(m, c)
10.
           Inserir p em m.planos ordenado por maior
       Fim Para
11.
12. Fim Para
13.
14. Para Cada Modelo m em modelosDisponiveis
      Para Cada Plano p em m.planos_{0...10}
15.
         c = \text{extrairContorno}(gema, p)
16.
          {centro3D, raio} = encontrarMaiorCinturaInscrita(m, c)
gemaAux = transladar3D(gema, centro3D)
17.
18.
19.
          rotacionar3D(gemaAux, p)
20.
          escalar(m, raio)
          {volume, escala} = encontrarMaiorInscricaoValida(m, gemaAux)
21.
22.
          Se volume > m.modeloInscrito.volume
23.
              m.modeloInscrito = {volume, escala, p, centro3D}
24.
          Fim Se
25.
      Fim Para
26. Fim Para
27. inserirNoArquivoXML(gema)
28. Para Cada Modelo m em modelosDisponiveis
29.
          inserirNoArquivoXML(m)
30. Fim Para
```

O otimizador recebe como entrada dois arquivos, um contendo a representação virtual 3D da gema, com seus pontos e faces e outro contendo os pontos críticos.

A representação virtual da gema é obtida através de um scanner 3D que resulta em um arquivo de texto contendo os pontos da malha. Para uso no otimizador essa malha deve ser simplificada reduzindo o tempo computacional necessário para os cálculos da otimização. A simplificação reduz o número de pontos e faces do modelo tridimensional mantendo as características topológicas e volumétricas da gema sem perdas significativas (Brusso *et all*, 2010, 47).

A seleção dos pontos críticos é realizada por uma pessoa qualificada através do uso do software Meshlab. No Meshlab é feita uma seleção, de forma visual, de pontos considerados pelo usuário como principais pontos de inflexão da superfície da gema. (Figura 1)

O algoritmo, como descrito acima, gera todos os planos possíveis pela combinação dos pontos críticos, três a três. Para cada um desses planos é extraído o contorno da gema encontrando os pontos

## Revista CIATEC – UPF, vol.3 (2), p.p.44-55, 2011

da gema no plano. Esse contorno é então usado para a inscrição da cintura do modelo de lapidação, para cada um dos modelos disponíveis. Esse processo trabalha apenas em 2D e seleciona os planos que possibilitam a inscrição da maior cintura, os quais são armazenados em um vetor.

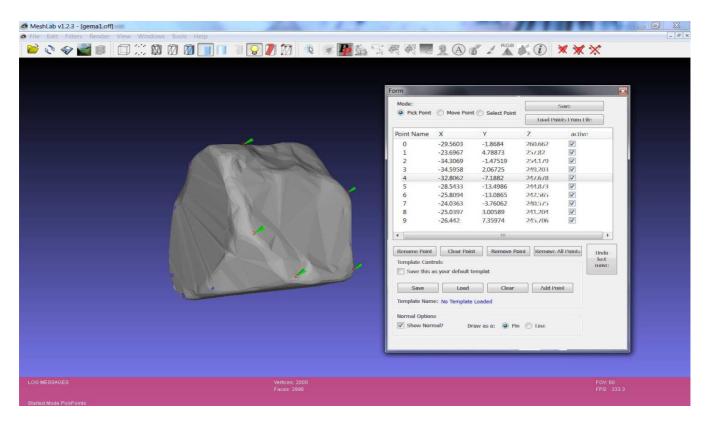


Figura 1. Seleção de pontos críticos no programa Meshlab

Os planos resultantes da etapa anterior são validados com o modelo de lapidação em 3D. A validação da inscrição consiste em inserir o modelo de lapidação na gema com a cintura no plano em questão e iterativamente ajustar o modelo e verificar se o mesmo está inteiramente inserido dentro da gema. Os ajustes feitos ao modelo de inscrição irão rotacionar o modelo, transladá-lo e reduzir sua escala conforme seja necessário. Para cada um dos modelos disponíveis é selecionado nesse processo o modelo inscrito com maior volume.

O resultado final é escrito em um arquivo XML para posterior visualização. Esse arquivo contém todos os dados para se gerar o diagrama de lapidação para qualquer um dos modelos disponíveis.

Com a primeira versão do algoritmo foi possível perceber alguns problemas resultantes desta abordagem baseada na exploração apenas dos planos obtidos a partir dos pontos críticos: por causa da forma irregular da gema não é garantido que os pontos extremos resultarão no maior aproveitamento, pois o plano obtido a partir dos pontos críticos não necessariamente possui a forma adequada para inserção da cintura do modelo de lapidação. Logo, na etapa de validação da inscrição, devido aos ajustes feitos à escala do modelo, esse acaba tendo reduzido o seu tamanho e conseqüente o aproveitamento da gema. Junta se a isso a necessidade de intervenção humana qualificada para a seleção dos pontos críticos.

# 3. ALGORITMOS GENÉTICOS

Com o uso de inteligência artificial nesse sistema é possível diminuir a necessidade de intervenção humana no processo de otimização. A seleção dos pontos críticos, feita manualmente pode ser substituída por uma técnica de IA que torna possível em tempo de execução testar um escopo mais abrangente, não limitado aos pontos que um usuário, mesmo que experiente, selecionou visualmente como sendo os pontos extremos da gema.

Embora não exista uma definição exata, é consenso para pesquisadores que algoritmos genéticos tem grande chance de serem competitivos ou melhores que outros métodos de IA similares quando o domínio de busca é grande, não é unimodal, ou não é totalmente conhecido, ou a função de avaliação é incerta, e se a tarefa não exige que a resposta encontrada seja a melhor resposta possível (Mitchell,1999, p.116).

Em algoritmos genéticos cada solução possível do problema é vista como um indivíduo e através de mudanças nessa solução o algoritmo tenta encontrar soluções melhores, indivíduos mais aptos em uma apologia a evolução das espécies proposta por Darwin. Com essa base na seleção natural, da sobrevivência do mais apto, o algoritmo genético permite chegar a uma solução sabendo apenas como reconhece-la.

Existem várias formas de se representar um indivíduo usando números, letras, e códigos, dentre todas a mais simples e mais utilizada é a *string* de bits com tamanho fixo. A representação da solução é um passo importante na construção de um algoritmo genético e deve-se procurar a representação que melhor se adapte ao problema em questão. A representação precisa ser simples mas completamente expressiva(Wall,1996).

Um algoritmo genético em sua estrutura básica é composto de uma população de indivíduos que é sucessivamente substituída por uma nova população até que se alcance um critério de parada como ilustrado na figura 2.

O algoritmo genético simples descrito por David Goldberg (Goldberg 1989) utiliza os operadores de crossover e mutação. Para que a nova população seja gerada, uma seleção é feita mantendo, ou não, alguns dos membros da população atual e gerando novos através de cruzamento (*crossover*) e mutação (*mutation*).

As populações são normalmente iniciadas com indivíduos aleatórios e podem ter tamanho fixo ou não de acordo com o resultado esperado, assim também o número de indivíduos preservados a cada geração e a seleção para cruzamento e mutação variam de acordo com o problema em questão. Algoritmos que mantém os melhores indivíduos de uma geração para a próxima são chamados elitistas.

O crossover é uma operação que tenta imitar a reprodução sexuada criando novos indivíduos a partir de dois indivíduos pais que tem suas características misturadas. É a operação mais representativa de um algoritmo genético (Holland, 1979).

Existem tipos diferentes de crossover em relação ao material genético trocado, sendo que os mais comuns são o crossover de um ponto e o crossover de dois pontos. Como o nome já diz, no crossover de um pondo cada cromossomo pai é dividido em dois em um ponto aleátorio e os descendentes pegam uma parte de cada pai, e no de dois pontos o cromossomo é dividido em três e os descendentes recebem duas partes de um pai e uma de outro, sempre mantendo as posições das partes.

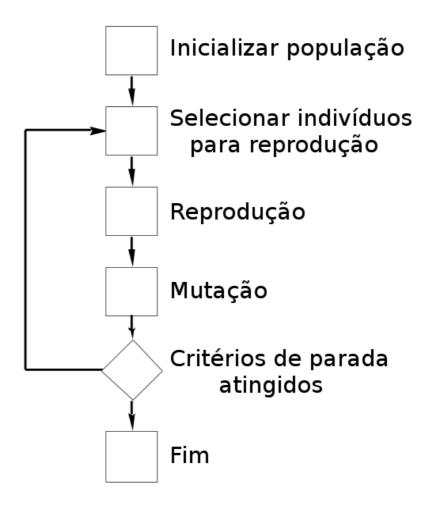


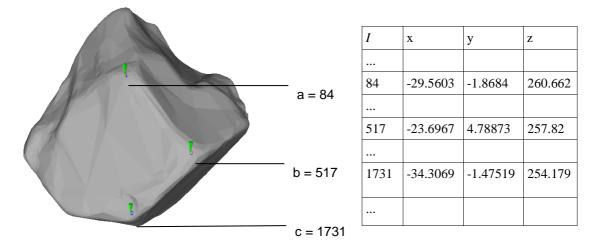
Figura 2. Estrutura de um algoritmo genético

A mutação é uma operação que altera aleatoriamente uma parte do cromossomo para manter a diversidade genética da população e permitir que todas as diferentes soluções tenham possibilidade de serem avaliadas. A mutação ocorre normalmente em uma taxa pequena.

A aptidão (*fitness*) é um valor calculado sobre cada indivíduo por uma função de avaliação e representa a qualidade de uma solução. A aptidão determina a capacidade de um indivíduo de sobreviver e se reproduzir. A função de avaliação é de extrema importância, pois são os resultados da avaliação que vão guiar a evolução para resultados melhores.

# 4. O ALGORITMO GENÉTICO NO 3D GEMAS

No 3D Gemas cada indivíduo é formado por um conjunto de três valores decimais, cada um desses três valores é um índice em um vetor contendo todos os pontos da malha obtida pela digitalização 3D de uma gema. No algoritmo isso é representado com a concatenação dos três índices (a, b e c) em sua forma binária resultando em uma cadeia de valores binários, ou seja, uma *string* binária. (Figura 3)



Cromossomo = 00001010100 01000000101 11011000011

Figura 3. Representação de um indivíduo no 3D Gemas

Para a implementação do algoritmo genético no 3D gemas utilizou-se a biblioteca de algoritmos genéticos para C++ GALib escolhida por sua versatilidade e extensa documentação.

Foi utilizado um algoritmo bastante similar ao algoritmo genético simples, previamente descrito, com elitismo mantendo o melhor indivíduo a cada geração. A população inicial é formada por um conjunto aleatório de indivíduos.

O método de seleção dos indivíduos para reprodução é o de roleta russa, onde a probabilidade de ser selecionado é dada aos indivíduos de acordo com a relação entre sua aptidão e a aptidão da população, favorecendo a reprodução de indivíduos mais aptos. O critério de parada foi definido como sendo um número determinado de gerações.

A função de avaliação é uma das partes mais importantes do algoritmo genético, pois é essa função que permite quantificar aptidão de cada indivíduo e fornece a base para a evolução. A função objetivo é específica para cada problema.

Inicialmente o algoritmo genético seguiria a estrutura da versão anterior substituindo a etapa que escolhia os planos com a maior cintura inscrita dentre todos os planos formados com os pontos críticos. Dessa forma a função de avaliação do algoritmo genético seria responsável por extrair o contorno no plano e inserir a cintura nesse contorno. A aptidão seria, então, a área da cintura.

No planejamento da implementação descobriu-se que essa abordagem não era viável. Para os pontos críticos, um vetor contendo os planos com a maior cintura era armazenado para posterior validação, isso permitia mais opções caso a validação da inscrição acabasse por reduzir demais a escala do modelo de lapidação. No algoritmo genético, caso o melhor indivíduo encontrado resultasse em um modelo inscrito com pouco aproveitamento, os demais indivíduos resultantes do AG, devido a convergência decorrente da evolução, teriam aproveitamento muito similar e o resultado final acabaria sendo ruim.

Para garantir um resultado satisfatório no aproveitamento da gema a validação da inscrição passou a ser parte da função de avaliação. O valor da aptidão de cada indivíduo passou então a ser a escala do modelo de lapidação inscrito, em relação ao seu tamanho original.

Como o 3D Gemas trabalha com mais de um modelo de lapidação, atualmente oval e redondo, e os pontos nos quais é inscrito um modelo com bom aproveitamento podem não ter bom aproveitamento para outro modelo, é feita uma rodada completa e independente do algoritmo genético para cada modelo de lapidação. O algoritmo genético é o mesmo para cada uma dessas rodadas, com uma única mudança na função de avaliação, pois a validação é diferente para cada modelo.

## Revista CIATEC - UPF, vol.3 (2), p.p.44-55, 2011

A função de avaliação segue o algoritmo abaixo, operando sobre os três pontos a,b e c obtidos do cromossomo:

```
    p = criarPlano(a, b, c)
    contorno = extrairContorno(gema, p)
    {centro3D, raio} = encontrarMaiorCinturaInscrita(m, contorno)
    gemaAux = transladar3D(gema, centro3D)
    rotacionar3D(gemaAux, p)
    escalar(m, raio)
    {volume, escala} = encontrarMaiorInscricaoValida(m, gemaAux)
    retornar(escala)
```

Inicialmente a função de avaliação ira extrair o contorno formado pelos pontos do indivíduo. Nesse contorno será encontrada a maior cintura inscrita possível para o modelo.

Utilizando uma cópia da gema, para que não se modifique a original, são feitos os ajustes, rotação, translação e escala, para que o modelo de inscrição esteja totalmente inserido na gema. Após a validação a aptidão do indivíduo fica definida como sendo a escala obtida.

## 4.1. Definição de Parâmetros

Como os valores referentes ao tamanho da população, número de gerações, taxa de crossover e taxa de mutação são diferentes para cada aplicação e não seguem nenhum critério exato de definição uma série de testes foi realizada para se estabelecer os mesmos.

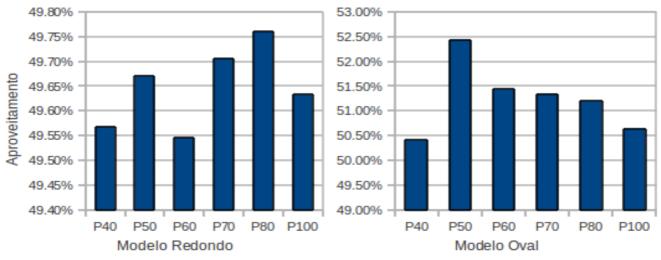


Figura 4. Testes de tamanho da população

Para definir o tamanho da população foram realizados testes com 100 gerações e tamanho da população 40, 50, 60, 70, 80 e 100. Esses testes foram feitos com a gema *SenaiL1-G2*<sup>1</sup> sobre os dois modelos de lapidação disponíveis, o gráfico contendo os resultados obtidos pode ser visto na figura 4. Como na realização destes testes ainda não haviam valores determinados para crossover e mutação, arbitrariamente foi definido crossover=0.75 e mutação=0.1.

Nesses testes foi utilizado um mesmo número (seed) no gerador de números aleatórios para analisar o impacto do tamanho da população sobre um mesmo conjunto de dados. Esses testes foram

Esta gema foi selecionada para os testes por ser a que resultou em maior aproveitamento na versão anterior do otimizador.

rodados em um notebook dual core 1.6 ghz com 2 gigabytes de memória ram em um sistema operacional linux ubuntu 9.04.

Embora o melhor resultado para o modelo redondo tenha sido obtido com uma população grande, o mesmo não foi verdadeiro para o modelo oval e a diferença nos resultados não compensou o custo computacional visto que para o primeiro caso, com 100 gerações de população 40, a execução levou aproximadamente três horas e vinte minutos enquanto que no último caso, com 100 gerações de população 100, a execução levou aproximadamente oito horas e vinte minutos. O tamanho da população foi definido então como sendo cinqüenta em virtude do bom resultado visto nos testes sem ter um custo computacional muito alto.

Com o mesmo conjunto de resultados dos testes anteriores foi possível observar o aproveitamento em razão do número de gerações.

Independente do tamanho da população, nos testes realizados, para ambos os modelos o aproveitamento converge rapidamente nas primeiras gerações e sofre poucas mudanças em gerações mais avançada, como exemplificado no gráfico da figura 5 que mostra o melhor aproveitamento a cada geração para uma população de tamanho cinqüenta. Essa convergência constituí uma relação de velocidade contra desempenho, sendo que para se obter melhores resultados o algoritmo deve ter uma convergência permitindo explorar mais opções.

O número de gerações foi definido como cinqüenta levando em conta que, nos testes realizados, as maiores mudanças no aproveitamento ocorriam até essa geração e um maior número de gerações gastaria muitos recursos por pouca diferença nos resultados.

Para definir os parâmetros de crossover e mutação utilizou-se os valores definidos anteriormente de população e número de gerações cinqüenta. Os testes foram feitos com a gema *SenaiL1-G2* sobre os dois modelos de lapidação disponíveis. Os resultados abaixo são valores médios entre três execuções, cada teste levou em torno de uma hora e quarenta minutos para ser realizado.

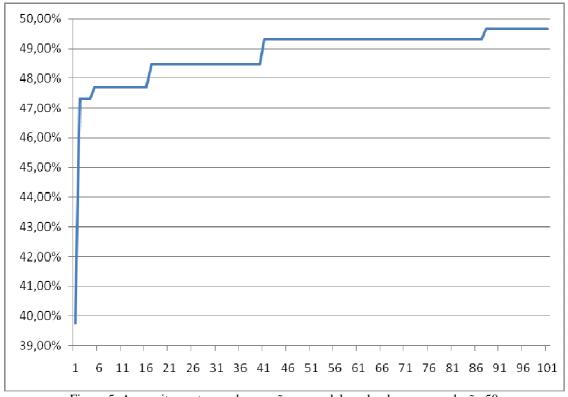


Figura 5. Aproveitamento a cada geração no modelo redondo com população 50

Os resultados obtidos nesses testes mostram um bom desempenho para todas as situações, mesmo no pior caso, crossover 0.8 e mutação 0.2 no modelo redondo, o aproveitamento foi superior a 49% um resultado superior ao anteriormente obtido com a versão que utiliza os pontos críticos, para a mesma gema

Modelo Redondo

C 0.75 M 0.01

C 0.75 M 0.05

C 0.85 M 0.1

C 0.75 M 0.1

C 0.8 M 0.2

C 0.8 M 0.1

C 0.7 M 0.1

A8.80% 48.90% 49.00% 49.10% 49.20% 49.30% 49.40% 49.50% 49.60% 49.70%

Aproveitamento

Figura 6. Testes de crossover e mutação no modelo redondo

Como os testes anteriores com relação ao número de gerações apresentaram grande convergência, optou-se por realizar testes com taxa de mutação relativamente alta, numa tentativa de aumentar a diversidade em gerações mais avançadas. Infelizmente essa hipótese não obteve o resultado esperado. Embora o teste com crossover 0.8 mutação 0.3 tenha igualado o melhor aproveitamento para o modelo redondo (Figura 6), a melhor opção encontrada foi a com mutação mais baixa com crossover 0.75 e mutação 0.01 que obteve o melhor desempenho nos dois modelos.(Figura 7)

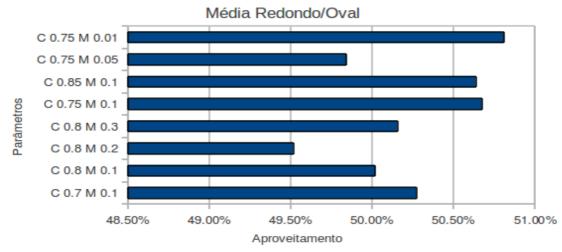


Figura 7. Testes de crossover e mutação média dos modelos

Optou-se por utilizar os mesmos parâmetros para ambos os modelos disponíveis pois a estrutura do algoritmo genético é a mesma para os dois e os testes realizados com os mesmos conjuntos de parâmetros mostraram bom desempenho nos dois modelos.

52

# 5. COMPARAÇÃO

Com os parâmetros já definidos, uma nova série de testes foi realizada para analisar o desempenho do algoritmo genético implementado em relação à versão anterior. Do conjunto de dados obtido na versão anterior com o aproveitamento de setenta e duas gemas, foram escolhidas nove gemas para a comparação de acordo com seu aproveitamento, três com bom aproveitamento, três intermediárias e três com baixo aproveitamento.

Os resultados obtidos mostram um excelente desempenho do algoritmo genético em gemas que obtiveram baixo aproveitamento na versão anterior com os picked points, enquanto que nos casos em que a versão anterior tinha alcançado aproveitamentos altos o algoritmo genético conseguiu resultados próximos, mas, para esse conjunto de testes, não conseguiu igualar ou superar os resultados anteriores. (figuras 8 e 9)

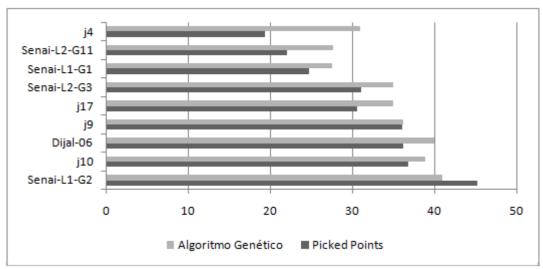


Figura 8. Comparação entre as versões no modelo de lapidação oval

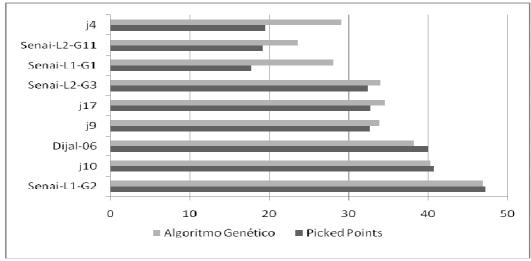


Figura 9. Comparação entre as versões no modelo de lapidação redondo

O algoritmo genético obteve, no melhor caso, 11,64% a mais de aproveitamento em relação à versão anterior e, no pior caso, 4,32% a menos de aproveitamento. Esses resultados mostram como a

limitação imposta pelos picked points pode prejudicar o aproveitamento das gemas e, ao mesmo tempo, mostram que a visão de um especialista não pode ser facilmente superada.

Um último teste realizado sobre a gema *Senai L2-G1*, que em testes com a versão anterior tinha obtido o menor aproveitamento de todos com 8,66% no modelo redondo e 5,28% no modelo oval obteve resultados surpreendentes com o algoritmo genético com aproveitamento 34,88% no modelo redondo e 30,47% no modelo oval. Um ganho de 22% no melhor caso.

## 6. CONCLUSÕES E TRABALHOS FUTUROS

O algoritmo genético implementado foi capaz de cumprir os objetivos propostos, obtendo resultados superiores ou muito próximos aos obtidos com o uso dos pontos críticos sem necessidade de intervenção humana no processo de otimização.

Com o volume de testes realizado foi possível determinar bons valores para os parâmetros do algoritmo genético assim como validar sua execução. Futuramente um maior número de testes pode alcançar melhores resultados.

A convergência prematura detectada nos testes pode ser melhor tratada com uso de técnicas como:

- escalonamento sigma (*sigma scaling*), onde a aptidão de um indivíduo leva em conta a aptidão do resto da população e o desvio padrão da população;
- crowding, em que indivíduos muito similares são substituídos por novos indivíduos;
- sharing, onde indivíduos muito similares tem aptidão reduzida;
- *Decimation-and-hot-boot* (DHB), em que, se o melhor resultado se manter por um número *x* de gerações, o melhor indivíduo da população é mantido e substitui-se uma parte da população restante por uma nova população (Benítez 2010).

Trabalhos futuros podem avaliar a utilização de outras técnicas em adição as do algoritmo genético atual, como por exemplo, o uso de populações evoluindo paralelamente com migração de indivíduos entre elas.

# 7. REFERÊNCIAS BIBLIOGRÁFICAS

ASPAHAN, S. Manual de Lapidação, Como Lapidar em Máquinas Lapidart. Minas Gerais: Lapidart Ltda, 2006.

BENÍTEZ, C.M., LOPES, H.S. Protein structure prediction with the 3D-HP side-chain model using a master-slave parallel genetic algorithm. Journal of the Brazilian Computer Society, v. 16, n. 1, p. 69-78, 2010.

BRUSSO M. J. *et all*. "Tecnologia 3D Gemas: otimização do aproveitamento de gemas coradas digitalizadas tridimensionalmente". In: HARTMANN, L. A.; DA SILVA, TONEZER, J. "Tecnologias para o setor de gemas, joias e mineração". Porto Alegre, Editora UFRGS. 2010 (*no prelo*).

GOLDBERG, D. E. Genetic Algorithms in Search, Optimization, and Machine. Earning, Addison-Wesley Publishing Company, 1989. (Algoritmo Genético e otimização).

HOLLAND, J. H. 1975. Adaptation in Natural and Artificial Systems. 2. ed. University of Michigan Press: MIT Press, 1992.

# Revista CIATEC – UPF, vol.3 (2), p.p.44-55, 2011

MITCHELL, M.. An Introduction to Genetic Algorithms. 5. ed. Londres: The MIT Press, 1999.

WALL, M. 1996. GALIB: A C++ Library of Genetic Algorithm Components disponível em http://lancet.mit.edu/ga/dist/galibdoc.pdf, acessado em Novembro de 2010