

## 3

# Otimização Evolucionária de Problemas com Restrição

### 3.1.

#### Introdução

Este capítulo resume os principais conceitos sobre os algoritmos evolucionários empregados nesta dissertação. Primeiramente, se fornece uma breve explicação sobre o princípio de funcionamento dos Algoritmos Genéticos (AG), descrevendo suas partes principais e seus parâmetros de evolução. Em seguida é apresentado o algoritmo de tratamento de restrições GACOM com Restrições.

### 3.2.

#### Algoritmos Genéticos

Na literatura especializada, é possível encontrar uma grande quantidade de métodos que podem ser empregados na resolução dos mais variados tipos de problemas de otimização. De maneira geral, os métodos de otimização podem ser separados em dois grupos distintos: métodos locais e métodos diretos. Nos métodos locais, também conhecidos como métodos baseados em gradiente, a cada passo do processo de otimização, os parâmetros envolvidos são atualizados com base no cálculo do gradiente da função objetivo (métrica de erro) em relação a cada parâmetro. Métodos dessa natureza apresentam uma tendência natural de convergência para ótimos locais, o que limita a sua capacidade de explorar o espaço de soluções. Devido a essas características, os métodos locais funcionam melhor como um mecanismo de ajuste fino de uma solução previamente encontrada. Alguns dos principais métodos de otimização baseados em gradiente são: Gauss-Newton e Levenberg-Marquardt (NOCEDAL 1999).

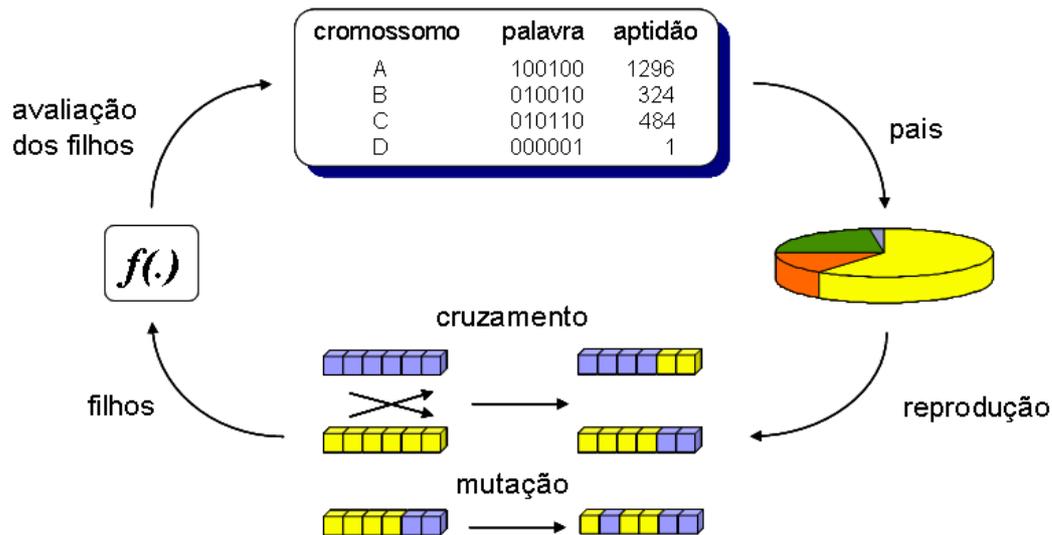
Diferentemente dos métodos locais, durante o processo de otimização, os métodos diretos levam em consideração apenas o valor da função objetivo e não necessitam do cálculo dos gradientes internos. No grupo dos métodos diretos, estão os métodos evolutivos, que são baseados em populações de potenciais soluções de um problema. Os métodos evolutivos se mostram interessantes na

resolução de problemas complexos de otimização porque conseguem um equilíbrio entre a capacidade de exploração do espaço de soluções e também de aproveitamento das melhores soluções ao longo da evolução. Isso os torna menos suscetíveis ao aprisionamento em ótimos locais, porém o tempo de convergência costuma ser bem mais elevado que o tempo gasto por métodos locais. Devido a essas características, os métodos evolutivos são mais adequados para a busca de uma solução aceitável, que não necessariamente seja uma solução ótima, para um problema de otimização.

A computação evolutiva é uma abordagem para resolução de problemas inspirada na Teoria da Evolução de Darwin, principalmente no princípio da sobrevivência dos mais aptos. Segundo Darwin (DARWIN 1859):

“Como muito mais indivíduos de cada espécie são gerados do que os que teriam possibilidade de sobreviver; e como, conseqüentemente, há um esforço freqüentemente recorrente para a existência: qualquer ser vivo, se variar, mesmo que ligeiramente, de qualquer maneira lucrativa a si próprio, sob as complexas e, às vezes, variáveis condições de vida, terá uma possibilidade melhor de sobreviver, e assim de ser selecionado naturalmente. Do princípio forte da herança, toda a variedade selecionada tenderá a propagar sua nova e modificada forma.” (DARWIN 1859)

Os Algoritmos Genéticos, um dos principais modelos pertencentes à computação evolutiva, são utilizados tipicamente para resolver problemas na forma  $f : S \rightarrow \mathfrak{R}$ , onde  $S$  é um espaço de busca constituído por todas as possíveis soluções para um problema particular. Dependendo das peculiaridades do problema, as soluções podem ser representadas por vetores  $n$ -dimensionais de números binários, inteiros, reais, ou estruturas mais complexas. Para todas as soluções existentes no domínio de  $S$ , um número real é associado, medindo quão adequada é a solução para resolver o problema em questão (BACK 1996), (MITCHELL 1994), (DAVIS 1991). A tarefa principal de um algoritmo genético é buscar, de forma eficiente, em amostras do espaço de busca  $S$ , soluções que estejam de acordo com o objetivo do problema. É importante mencionar que essas soluções não precisam ser necessariamente ótimas, mas sim satisfatórias. Ao lidar com espaços de busca grandes e complexos, o ótimo pode ser difícil de ser atingido e, neste caso, pode-se apenas esperar achar uma solução satisfatória. A Figura 21 ilustra o funcionamento de um algoritmo genético.



**Figura 21. Fluxo básico de um algoritmo genético.**

Dado um problema particular de busca, uma representação adequada deve ser selecionada para codificar possíveis soluções em estruturas de dados do tipo definido no conjunto  $S$ . Após a representação ser escolhida, um número  $n$  de potenciais soluções, também chamadas de indivíduos, é gerado aleatoriamente. Esses indivíduos passam então por duas etapas básicas: avaliação e operações genéticas. Durante a avaliação, um número real, também chamado de aptidão, é associado a cada indivíduo. A aptidão do indivíduo mede o quanto ele é adequado para satisfazer à especificação de um problema particular. Após serem avaliados, as seguintes operações, ou operadores genéticos, são utilizados nos indivíduos: seleção, cruzamento e mutação. Depois de passarem por tais operadores, uma nova população é criada, formando assim a próxima geração. A avaliação e os operadores genéticos são aplicados na próxima geração e o processo continua, até que um critério de parada seja satisfeito. Esse critério de parada pode ser um número máximo de gerações ou o alcance de uma solução adequada para o problema (ZEBULUM 2001). A seguir são apresentados, com maiores detalhes, os conceitos por trás dos Algoritmos Genéticos.

### 3.2.1. Representação

A representação se refere à forma como as soluções de um determinado problema são codificadas em uma estrutura de dados que possa ser processada em um computador digital. Uma solução para um dado problema pode ser: um

número, representando o valor ótimo de uma função matemática; um vetor de reais, representando o valor ótimo de uma função com múltiplas variáveis; uma lista de eventos, representando uma ordem ótima de eventos para se realizar uma determinada tarefa; uma estrutura, simbolizando algum modelo de engenharia, um circuito elétrico, uma reação química ou qualquer outro sistema.

A solução de um problema pode ser representada por um conjunto de parâmetros, cada um desses parâmetros é representado por um gene. Esse gene pode ser um número real, inteiro ou binário. Um conjunto de genes de mesma representação forma um segmento e um conjunto de segmentos forma um indivíduo. Um conjunto de indivíduos com mesma estrutura forma uma população. Uma solução é formada pela interpretação de um indivíduo da população, em alguns casos, pode-se utilizar mais de um indivíduo para representar uma solução, no entanto, cada um dos indivíduos, que compõem a solução, deve pertencer a uma população diferente.

A decodificação do cromossomo consiste basicamente na construção da solução real do problema, fenótipo, a partir do cromossomo, genótipo. O processo de decodificação constrói a solução para que esta seja avaliada pelo problema. É importante que um cromossomo tenha apenas uma decodificação, enquanto que não é tão importante que uma solução tenha apenas uma representação no cromossomo.

### **3.2.2. Avaliação**

A avaliação é a ligação entre o Algoritmo Genético e o problema a ser solucionado. A avaliação é o processo de associar um valor de aptidão a cada indivíduo, selecionado pelo AG. Na natureza, a aptidão de um indivíduo mede o quão bem adaptado ele está a um determinado ambiente. Da mesma forma, no caso dos AG, a aptidão mede o desempenho de um indivíduo de acordo com a especificação de um problema. Essa medida é, normalmente, um valor escalar, inteiro ou real. No caso da definição padrão de problemas de busca, pode-se observar que um número real (o valor de aptidão) é associado com cada ponto do espaço de busca  $S$ . Quando há somente um objetivo a ser satisfeito no problema em questão, a função de avaliação dos indivíduos é normalmente encontrada de forma bem direta. Entretanto, quando um ou mais objetivos devem ser

perseguidos pelo algoritmo deve-se ponderar o peso de cada objetivo e a avaliação dos indivíduos deve ser encontrada em forma de uma função.

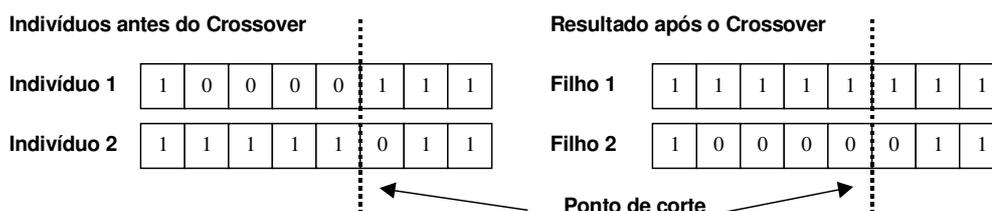
### 3.2.3. Operadores

Existem três mecanismos naturais nos quais os algoritmos genéticos são baseados: seleção natural, recombinação e mutação. Estes são os principais operadores genéticos utilizados:

*Seleção:* O operador de seleção é um componente essencial de um Algoritmo Genético. Esse operador, baseado no valor da aptidão dos indivíduos, seleciona aqueles que farão parte da próxima geração. Com isto os cromossomos mais aptos (valor de aptidão maior), têm maior probabilidade de contribuir para a formação de um ou mais indivíduos da população seguinte. Existem basicamente os seguintes métodos: troca de toda população, troca de toda população com elitismo, onde todos os cromossomos são substituídos sendo o cromossomo mais apto da população corrente copiado para população seguinte, troca parcial da população (*steady state*), onde os M melhores indivíduos da população corrente são copiados para população seguinte (KOZA 1992), (GOLDBERG 1989) e (MICHALEWICZ 1996).

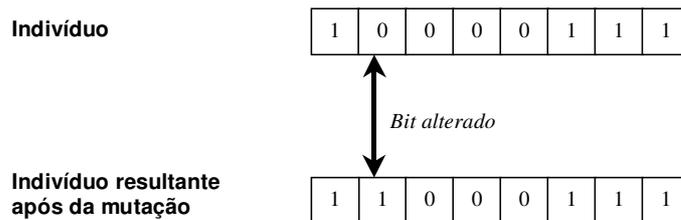
*Cruzamento:* Este operador é inspirado na idéia da recombinação de material genético entre indivíduos. Como mostrado na Figura 21, o cruzamento ocorre após a seleção. Esse operador é aplicado de forma probabilística nos indivíduos. Dois indivíduos são selecionados aleatoriamente e, de acordo com uma probabilidade pré-definida (DAVIS 1989), seu material genético é recombinado ou não. Se isso ocorrer, dois novos indivíduos com material de ambos os progenitores são gerados; caso contrário, os dois indivíduos permanecem inalterados.

Este processo pode ser observado no exemplo da Figura 22 a seguir, onde a solução está codificada com alfabeto binário.



**Figura 22. Cruzamento de um Ponto.**

*Mutação*: Este operador confere a troca aleatória do valor contido nos genes de um cromossomo por outro valor válido. No caso de se utilizar um gene binário, troca-se seu valor de 0 para 1 e vice-versa. Utiliza-se uma taxa de mutação que, para cada gene do segmento, sorteia-se se ocorrerá ou não a mutação; no caso de ocorrência, o gene será trocado por outro valor válido aleatório. Este processo pode ser observado no exemplo na Figura 23 a seguir, onde a solução está codificada com alfabeto binário.



**Figura 23. Mutação.**

A mutação garante a diversidade das características dos indivíduos da população e permite que sejam introduzidas informações que não estiveram presentes em nenhum dos indivíduos. Além disto, proporciona uma busca aleatória (*exploration*) no AG, oferecendo oportunidade para sair de ótimos locais e permitindo que mais pontos do espaço de busca sejam avaliados.

Do ponto de vista de otimização, uma das principais vantagens das técnicas de computação evolutiva é que estas não impõem muitos requisitos matemáticos sobre o problema a ser otimizado. Só necessitam da avaliação da função objetivo e, com isso, podem tratar os mais variados tipos de problemas, sejam eles definidos num espaço discreto, contínuo ou misto, com ou sem restrições (MICHALEWICZ 1996).

### 3.2.4. Parâmetros da Evolução

Ao se implementar um sistema que utilize Algoritmos Genéticos existem uma série de parâmetros e taxas que devem ser configuradas para o correto funcionamento do sistema e a convergência ao ótimo global no menor tempo possível. Os parâmetros que mais influenciam o desempenho do AG são:

*Tamanho da População*: o tamanho da população afeta o desempenho global e a eficiência dos AG's. Uma população muito pequena oferece uma pequena cobertura do espaço de busca, causando uma queda no desempenho.

Uma população suficientemente grande fornece uma melhor cobertura do domínio do problema e previne a convergência prematura para soluções locais. Entretanto, com uma grande população tornam-se necessários recursos computacionais maiores, ou um tempo maior de processamento do problema. Logo, deve-se buscar um ponto de equilíbrio no que diz respeito ao tamanho escolhido para a população.

*Taxa de Cruzamento:* probabilidade de um indivíduo ser recombinado com outro. Quanto maior for esta taxa, mais rapidamente novas estruturas serão introduzidas na população. Entretanto, isto pode gerar um efeito indesejável, pois a maior parte da população será substituída, ocorrendo assim perda de variedade genética, podendo ocorrer perda de estruturas de alta aptidão e convergência a uma população com indivíduos extremamente parecidos, indivíduos estes de solução boa ou não. Com um valor baixo, o algoritmo pode tornar-se muito lento para oferecer uma resposta aceitável.

*Taxa de Mutação:* probabilidade do conteúdo de um gene do cromossomo ser alterado. A taxa de mutação previne que uma dada população fique estagnada em um valor, além de possibilitar que se chegue a qualquer ponto do espaço de busca. Porém deve-se evitar uma taxa de mutação muito alta uma vez que pode tornar a busca essencialmente aleatória, prejudicando fortemente a convergência para uma solução ótima.

*Intervalo de Geração:* controla a porcentagem da população que será substituída durante a próxima geração (substituição total, substituição com elitismo, substituição dos piores indivíduos da população atual, substituição parcial da população sem duplicatas). Esse número de indivíduos substituídos também é conhecido como GAP.

*Número de gerações:* representa o número total de ciclos de evolução de um Algoritmo Genético, sendo este um dos critérios de parada do AG. Um número de gerações muito pequeno causa uma queda no desempenho, pois não consegue cobrir todo o espaço de busca. Um valor grande faz necessário um tempo maior de processamento, mas fornece uma melhor cobertura do domínio do problema evitando a convergência para soluções locais.

*Número de rodadas:* representa o número total de vezes que um AG será executado, sendo este um dos critérios de parada do AG. Sendo o AG uma técnica probabilística, é interessante que se execute mais de uma vez o AG para obter

uma melhor solução. Utilizar mais que uma rodada para resolver um problema é vantajoso, pois uma nova rodada não está presa aos ótimos locais encontrados por uma rodada anterior. Para um mesmo número de avaliações (mesmo tempo de processamento) o número de rodadas deve ser pequeno suficiente para permitir que o algoritmo aproveite (*exploitation*) a informação genética dos indivíduos da população e grande o suficiente para que uma nova rodada com novos indivíduos seja iniciada quando a rodada anterior convergir para um ótimo local e fornece uma melhor cobertura do domínio.

*Taxa de sementeira*: representa a porcentagem dos melhores indivíduos de uma rodada que serão utilizados como semente inicial na rodada seguinte. Uma taxa de sementeira muito pequena faz com que a rodada seguinte tire pouco proveito dos resultados obtidos na rodada anterior. Um valor grande faz com que a rodada corrente não consiga sair do ótimo local que a rodada anterior possa ter convergido.

### **3.2.5. Tratamento de Restrições**

#### **3.2.5.1. Introdução**

Em problemas complexos de otimização, a presença de restrições impostas ao espaço de busca é praticamente inevitável e o tratamento adequado dispensado a estas restrições tem influência direta na eficiência do processo de otimização. A maneira mais trivial de se lidar com restrições é aplicar penalidades às soluções que violem uma ou mais restrições impostas pelo problema. Entretanto, para a penalização das soluções inválidas é necessário avaliá-las, o que pode ser computacionalmente custoso. Outra opção é corrigir o que está tornando a solução inválida, entretanto nem sempre é possível fazer essa correção de modo direto e sem prejudicar a evolução de um AG por modificar um indivíduo sem ser pelos operadores. Uma última opção é remover os indivíduos não válidos e fazer com que um novo indivíduo seja gerado em seu lugar, no entanto, dependendo da dificuldade de se gerar um indivíduo válido essa opção pode se tornar altamente custosa.

### 3.2.5.2.GACOM com Restrições

Com o intuito de evitar avaliações desnecessárias e melhorar a eficiência do processo de otimização, foi desenvolvida, juntamente ao Laboratório de Inteligência Computacional Aplicada (ICA) da PUC-Rio, uma técnica de tratamento de restrições baseado no GENOCOP III (*Genetic algorithm Numerical Optimization for Constrained Problems*) (MICHALEWICZ 1996) chamada GACOM com Restrições. Utilizado em conjunto com o Algoritmo Genético, o GACOM com Restrições garante a geração apenas de segmentos válidos ao longo de todo o processo de otimização do segmento associado.

O GACOM com Restrições incorpora duas populações independentes, uma de busca e uma de referência, onde o desenvolvimento dos indivíduos em uma população influencia as avaliações dos indivíduos na outra população. A população inicial de busca é formada por indivíduos aleatórios que satisfazem apenas as restrições lineares, enquanto a população inicial de referência é formada por indivíduos válidos, que cumprem todas as restrições, tanto as lineares quanto as não lineares.

Ao longo das gerações do Algoritmo Genético, a evolução acontece sempre na população de busca, que no início, conforme mencionado anteriormente é formada por indivíduos que respeitam apenas as restrições lineares. No início da execução do Algoritmo Genético, tanto a população de busca quanto a de referência são simuladas. A partir daí, dado um indivíduo  $S$  da população de busca, para que este se torne válido, é realizada uma seqüência de cruzamentos aritméticos com um indivíduo  $R$  da população de referência, até que um indivíduo válido  $Z$  seja encontrado. Esse indivíduo  $Z$  é simulado e, caso o seu valor de avaliação seja maior que a avaliação do indivíduo  $R$ , o indivíduo  $R$  é substituído pelo indivíduo  $Z$ . Além disso, a cada operação, existe uma probabilidade  $P$  de o indivíduo  $S$  ser substituído pelo indivíduo  $Z$ . Caso a substituição não ocorra, o valor de avaliação do indivíduo  $Z$  é atribuído ao indivíduo  $S$  da população de busca. Isto acontece porque, uma vez que o indivíduo de busca não é avaliado, ao ser utilizado para a obtenção de um indivíduo válido, é necessário atribuir a ele um valor de avaliação de forma que seja possível selecioná-lo em gerações futuras do processo de evolução.

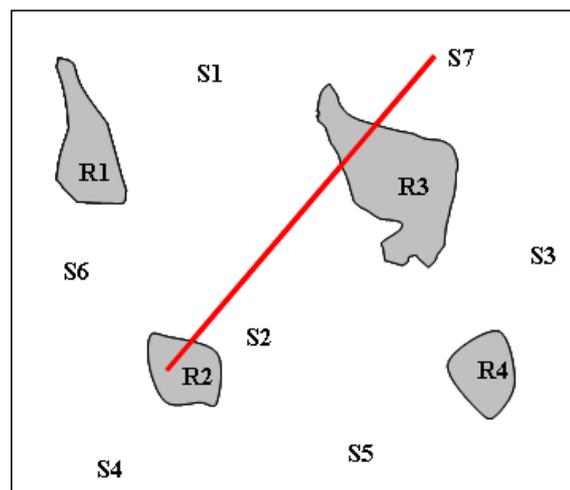
Devido às substituições que ocorrem nas populações de busca e de referência, à medida que transcorre o processo evolutivo, a população de referência ganha diversidade e a população de busca passa a ser povoada por indivíduos válidos. Assim, a evolução de um algoritmo genético com GACOM com Restrições tende a ser mais modesta nas primeiras gerações e torna-se mais atrativa nas gerações posteriores. O número de gerações necessárias, para que se comece efetivamente a evolução, está diretamente relacionado à facilidade de se encontrar indivíduos válidos para o problema em questão. O procedimento completo do GACOM com Restrições está ilustrado no algoritmo da Figura 24 e graficamente na Figura 25.

```

Início
  P = p; // probabilidade de substituição
  se válido(S) == falso
    Z = aS + (1 - a)R; // a ∈ [0, 1]
    enquanto válido(Z) == falso
      Z = aZ + (1 - a)R; // a ∈ [0, 1]
    fim enquanto
    se avaliação(Z) > avaliação(R)
      R = Z;
    fim se
    se rand() ≤ P
      S = Z; // S é substituído por Z com probabilidade P
    senão
      avaliação_S = avaliação(Z);
    fim se
  fim se
fim

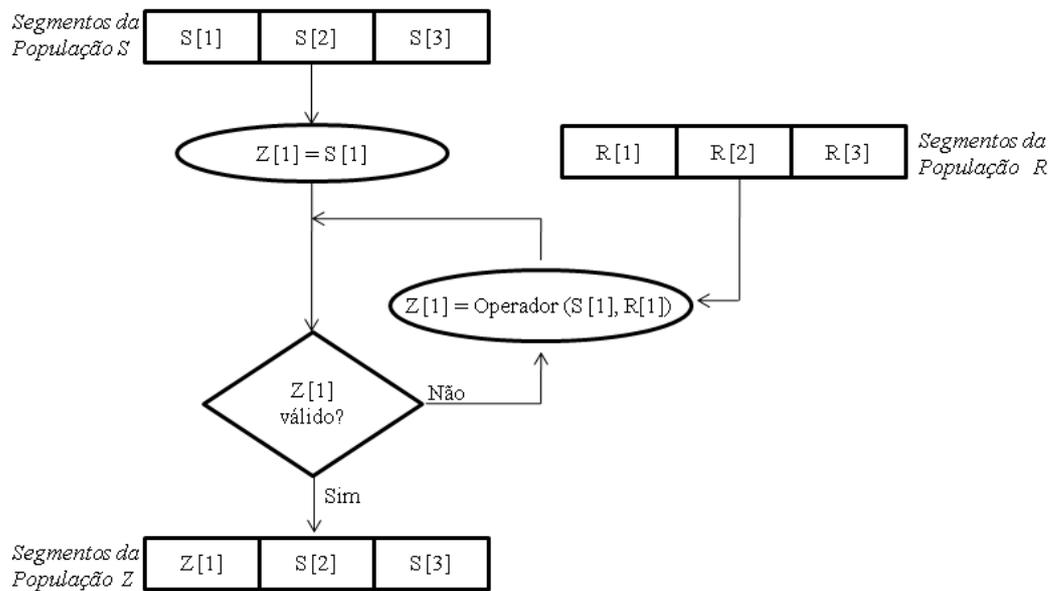
```

**Figura 24. Procedimento GACOM com Restrições – Construção de indivíduos válidos.**



**Figura 25. Procedimento GACOM com Restrições – Representação gráfica.**

Tanto os indivíduos da população de busca quanto os indivíduos da população de referência possuem todos os segmentos do indivíduo, no entanto, apenas o segmento associado ao tratamento de restrições é modificado na transformação de S para Z. Do ponto de vista dos outros segmentos a população que é avaliada é a população de busca. Os outros segmentos que compõem um indivíduo da população de referência serão utilizados somente durante a simulação inicial com o objetivo de avaliar o segmento que deve cumprir as restrições. Esse procedimento é ilustrado na Figura 26.



**Figura 26. GACOM com Restrições – Diagrama do procedimento com múltiplos segmentos**