

Trabalho de AG - Algoritmos Genéticos

Sarajane Marques Peres

14 de março de 2016

Disciplina: Inteligência Artificial
Bacharelado em Sistemas de Informação
Universidade de São Paulo
www.each.usp.br/si

Do objetivo e formação de grupos

Objetivo

Experimentação com Algoritmos Genéticos. A experimentação dar-se-á por meio de otimização de um conjunto de funções (parte I) e por meio da resolução da tarefa de roteamento de veículos (parte II).

- Este trabalho deve ser desenvolvido em grupos de, no máximo, **quatro** alunos. Grupos menores podem ser formados, mas a avaliação não considerará o número de integrantes do grupo, isto é, a avaliação seguirá os mesmos critérios quantitativos e qualitativos para todos os trabalhos, desenvolvidos em grupos de quatro alunos, em grupos menores ou mesmo individualmente.
- Todos os artefatos produzidos neste trabalho deverão ser postados no e-Tidia, na área de um dos integrantes do grupo. A necessidade da entregas na forma impressa será especificada neste documento, junto da especificação do artefato.

Das datas

Data de Entrega

A data máxima para entrega da primeira parte do trabalho é **2 de maio**.

A data máxima para entrega da segunda parte do trabalho é **20 de junho**.

Os grupos podem postar os artefatos no e-Tidia alguns dias antes da data máxima, se assim preferirem. O tempo dado para a realização do trabalho é suficiente para que os grupos não deixem as postagens para serem feitas na última hora e também para que não haja tentativas de entrega após a data.

Entregas após a data não serão aceitas e o grupo que deixar de entregar o trabalho na data correta ou antes dela, receberá **nota zero** no trabalho.

Dos artefatos

Artefatos a serem entregues

Os artefatos produzidos no trabalho deverão ser entregues via e-Tidia:

- artefatos de implementação (códigos): referentes ao Algoritmo Genético e outras funções e métodos que foram necessários para implementação dos problemas sob otimização e tratamento de instâncias de testes;
- instâncias de testes: organização, via código ou arquivos (.dat, .txt, outros) criados pelo grupo para aplicação do Algoritmo Genético;
- relatórios: arquivos .pdf nos quais o grupo explicará as tomadas de decisões realizadas durante o trabalho (parte I e parte II), os resultados obtidos, bem como as análises sobre os resultados obtidos (conforme solicitado mais à frente nesta especificação);
- vídeos de apresentação: vídeos nos quais o grupo apresenta seu trabalho em termos de codificação (conforme solicitado mais à frente nesta especificação).

Os relatórios também deverão ser entregues na forma impressa, no horário da aula dos dias de entrega da cada uma das partes do trabalho.

Da Avaliação

Processo de avaliação

A avaliação será realizada sobre todos os artefatos: relatório, vídeos e códigos, nesta ordem de importância. Eventualmente, os alunos do grupo podem ser chamados para prestar algum esclarecimento sobre o trabalho.

Do plágio

Se o plágio, de qualquer natureza, for constatado, o grupo não terá o seu trabalho avaliado.

Referências

Material de terceiros, disponíveis publicamente da Internet, provenientes de **homepages acadêmicas ou sites especializados no assunto**, poderão ser usados DESDE QUE esteja CLARAMENTE referenciados no relatório do grupo. Se corretamente referenciados, seu uso não se constitui em plágio.

Da codificação

Codificação

A codificação das rotinas necessárias no trabalho é de responsabilidade de cada grupo. O código deve ser comentado pelo grupo de maneira que seja simples encontrar, dentro do código, os trechos que implementam as principais estratégias codificadas. **Não economizem comentários – sejam claros e detalhistas.**

Dos comentários

O código deverá ser comentado de forma que seja possível identificar as funções/métodos/linhas de código que implementam as características do Algoritmo Genético. Cada uma das características da evolução da população, dos operadores, de estratégias de detecção/correção de infactibilidade, e uso de estratégias diferentes daquelas vistas em aula devem estar explicitamente indicadas no código.

Da codificação

Estratégias adicionais

Estratégias não requisitadas no trabalho podem ser apresentadas, desde que sejam ADICIONAIS e estejam completamente entendidas por todos os membros dos grupos. A apresentação de estratégias adicionais agregam valor ao trabalho e podem ajudar a melhorar a avaliação recebida pelo grupo, desde que elas não sejam entendidas como substituições ao que está aqui solicitado.

Linguagem de programação

Qualquer linguagem de programação ou ambiente de programação poderá ser usado pelo grupo, DESDE QUE a professora da disciplina tenha acesso (em sua máquina) ou que o grupo organize o acesso em máquinas do laboratório (ou particulares). Matlab e R são as possibilidades de acesso via máquina da professora. Outras linguagens ou ambientes a serem usados devem ser informados à professora ANTES de iniciar a confecção do trabalho.

Da codificação

O grupo deverá preparar o seu código de forma que seja possível executá-lo passando como parâmetro para o algoritmo, pelo menos, as seguintes informações:

- função a ser otimizada;
- porcentagens para execução de parâmetros
- número de gerações
- número de cromossomos na população

A entrega dos códigos deve ser acompanhada de uma arquivo readme.txt que explique como executá-lo.

Como resultado, a implementação deverá:

- mostrar na tela a progressão do fitness (total da população, médio da população, máximo e mínimo) dentro de um intervalo de gerações;
- gravar em arquivo texto a progressão do fitness (total da população, médio da população, máximo e mínimo) geração a geração;
- o arquivo texto deverá ainda indicar quais os valores dos parâmetros usados na execução do algoritmo que gerou aquele comportamento de fitness;

Operadores x Estratégias - Parte I

Devem ser implementados pelo menos os seguintes operadores e estratégias:

- uso de dois diferentes tamanhos de população, de grandezas bem diferentes;
- dois critérios de parada;
- cromossomo com codificação binária;
- um operador de seleção (preferencialmente roleta);
- dois operadores de crossover (crossover de um ponto e outro à escolha do grupo);
- dois operadores de mutação (mutação simples e outro à escolha do grupo);
- dois critérios de troca de população;
- evolução sem elitismo e com elitismo.

O grupo precisa estabelecer uma sistemática de combinação destes operadores e características nas execuções do seu Algoritmo Genético.

Instâncias de teste - Parte I

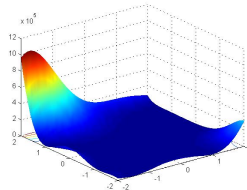
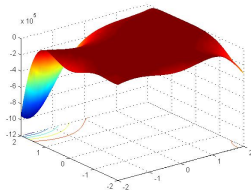
- para a otimização de funções (parte I), o teste deve ser realizado considerando um intervalo de otimização, conforme especificado em cada uma das funções;

Das funções a serem otimizadas - Parte I

Função Gold (corrigindo o erro na variável a

- função para minimização
- variação do x e do y no gráfico: $[-2, 2]$
- intervalo para minimização: $[-2, 2]$
- considere a função z e a função $-z$;

```
a = 1+(x+y+1)^2*(19-14*x+3*x^2-14*y+6*x*y+3*y^2);  
b = 30+(2*x-3*y)^2*(18-32*x+12*x^2+48*y-36*x*y+27*y^2);  
z = a*b;  
z = -z;
```



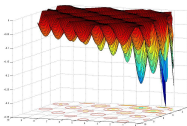
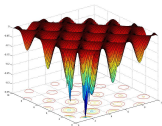
Das funções a serem otimizadas - Parte I

Essa função pode ser substituída pela Função Bump 2.

Função Bump 1

- função para minimização
- variação do x e do y no gráfico: $[0, 10]$
- intervalo para minimização: $[0, 10]$
- considere a função z ;

```
for i = 1 : n
    for j = 1 : n
        if (X(i,j) * Y(i,j)) < 0.75
            z(i,j) = NaN;
        elseif (X(i,j) + Y(i,j)) > 7.5 * 2
            z(i,j) = NaN;
        else
            temp0 = cos(X(i,j))^4 + cos(Y(i,j))^4;
            temp1 = 2* (cos(X(i,j))^2) * (cos(Y(i,j))^2);
            temp2 = sqrt(X(i,j)^2 + 2*Y(i,j)^2);
            z(i,j) = - abs((temp0-temp1)/temp2);
        end
    end
end
```

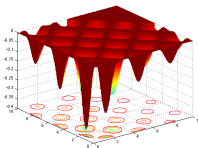


Das funções a serem otimizadas - Parte I

Função Bump 2

- função para minimização
- variação do x e do y no gráfico: $[0, 10]$
- intervalo para minimização: $[0, 10]$
- considere a função z ;

```
for i=1:n
    for j=1:m
        if (x(i)*y(j))<0.75
            z(i,j)=0;
        elseif (x(i)+y(j))>7.5*2
            z(i,j)=0;
        else
            temp0=cos(x(i))^4+cos(y(j))^4;
            temp1=2*(cos(x(i))^2)*(cos(y(j))^2);
            temp2=sqrt(x(i)^2+2*y(j)^2);
            z(i,j)=abs((temp0-temp1)/temp2);
        end
    end
end
```



Das funções a serem otimizadas - Parte I

Bump 2

Os laços FOR dos slides anteriores são necessários para percorrer o domínio de x e y para criação dos gráficos. Para a otimização não é necessário usá-lo. Veja abaixo:

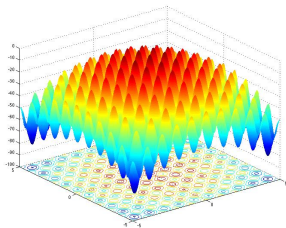
```
temp0=cos(x)^4+cos(y)^4;  
temp1=2*(cos(x)^2)*(cos(y)^2);  
temp2=sqrt(x^2+2*y^2);  
z =abs((temp0-temp1)/temp2);  
z = -z;
```

Das funções a serem otimizadas - Parte I

Função Rastrigin

- função para maximização;
- variação do x e do y no gráfico: $[-5, 5]$
- intervalo para minimização: $[-5, 5]$
- considere a função z ;

```
zx = (x^2-10*cos(2*pi*x)+10);  
zy = (y^2-10*cos(2*pi*y)+10);  
z = zx+zy;  
z = -z;
```



Conteúdo do relatório - Parte I

Relatório

Importante entender que se trata realmente de um relatório. Ou seja, é esperado que o aluno mostre como se deu o processo de estudo do algoritmo diante dos problemas apresentados.

Não é esperado qualquer menção às seções como introdução, motivação, justificativa, objetivos, fundamentação teórica ou trabalhos correlatos.

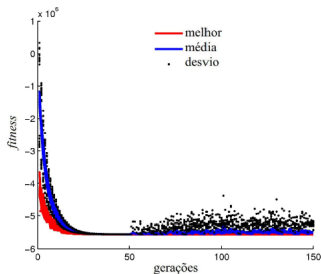
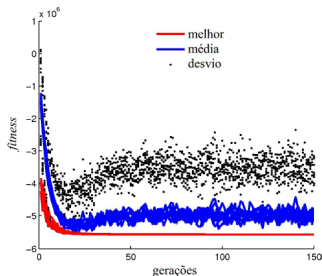
Para a parte I do trabalho, o relatório deve conter:

- representação para o cromossomo (se representações diferentes foram usadas em funções diferentes, cada uma delas deve estar apresentada);
- método utilizado para estudo dos parâmetros do algoritmo;
- explicação para as tomadas de decisão sobre a parametrização;
- descrição do processo de parametrização e observação de resultados para cada uma das funções a serem otimizadas;
- análise dos resultados de maneira comparativa, considerando cada uma das funções otimizadas;
- critérios de parada.

Gráficos

Ajudando a entender a evolução do algoritmo

A observação do gráfico de evolução do fitness pode ajudar a entender o comportamento do algoritmo mediante as diferentes parametrizações usadas.



Vídeo

O objetivo do vídeo é tão somente apresentar características da codificação construída. A duração do vídeo deverá ser de 3 a 10 minutos.

O vídeo deverá conter explicações, ilustradas a partir da codificação, sobre:

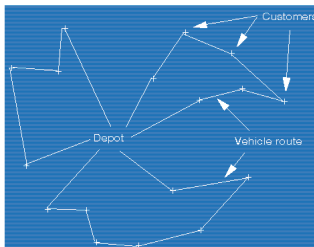
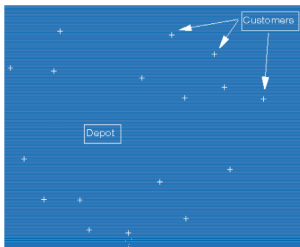
- estrutura de dados usada para o cromossomo;
- cálculo do fitness;
- lógica geral do Algoritmo Genético;
- lógica usada para implementação de cada um dos operadores (seleção, crossover, reprodução);
- implementação da troca da população e uso do elitismo;
- critérios de parada;
- outras estratégias que foram implementadas além do minimamente exigido.

Problema de otimização - Parte II

Problema de Roteamento de Veículos (Vehicle Routing Problem - VRP¹)

VRP é um nome genérico para uma classe de problemas na qual um conjunto de rotas para uma frota de veículos, baseados em um ou mais depósitos, deve ser determinada considerando um conjunto de cidades ou clientes distribuídos geograficamente.

O objetivo do VRP é atender a um conjunto de clientes que possuem demandas conhecidas, a partir do estabelecimento de rotas de custo mínimo para veículos que iniciam e terminam sua rota em um depósito.



¹Conteúdo baseado em www.bernabe.dorronsoro.es/vrp.

Problema de otimização - Parte II

Sugestão de representação para o cromossomo

Cada indivíduo representa uma solução para o problema. Ele é composto por uma sequência de inteiros que representam clientes e veículos. Para um problema com 10 clientes e 4 veículos, um cromossomo pode ser:

4 - 5 - 2 - 11 - 10 - 3 - 1 - 13 - 7 - 8 - 9 - 12 - 6

sendo que números de 1 a 10 indicam clientes e números de 11 a 14 indicam veículos. E, na solução apresentada, quatro rotas foram compostas, e cada uma é da seguinte forma:

- rota do veículo 11: depósito - cliente 4 - cliente 5 - cliente 2 - depósito
- rota do veículo 12: depósito - cliente 7 - cliente 8 - cliente 9 - depósito
- rota do veículo 13: depósito - cliente 10 - cliente 3 - cliente 1 - depósito
- rota do veículo 14: depósito - cliente 6 - depósito

Alternativamente você pode usar um caracter (*) para representar a troca de veículo, sem indicar qual veículo está associado a qual rota. Porém, isso só é possível quando assume-se que os veículos têm igual capacidade.

Problema de otimização - Parte II

Considerando capacidade infinita

Neste caso considera-se que os veículos tem capacidade infinita, e uma boa solução para o problema é aquela que minimiza a soma das distâncias das rotas planejadas

$$fitness(x) = \sum_{i=1}^m C(R_i)$$

em que:

- x : é uma solução (um conjunto de rotas)
- i : é um veículo
- R_i : é a rota do veículo i
- m : é o número de veículos disponíveis
- $C(R_i)$: é a soma dos custos de ir de um cliente a outro, seguindo a ordem da rota

Problema de otimização - Parte II

VRP - como um problema real

Em aplicações do mundo real, o problema VRP precisa assumir restrições. Existem várias restrições:

- CVRP ou Capacitated VRP: todo veículo tem uma capacidade limitada;
- VRPTW ou VRP with time windows: cada cliente tem que ser atendido dentro de uma determinada janela de tempo;
- MDVRP ou Multiple Depot VRP: o vendedor (prestador do serviço) usa muitos depósitos para atender os clientes;
- VRPPD ou VRP with Pick-up and Delivering: clientes podem devolver alguns produtos para o depósito;
- SDVRP ou Split Delivery VRP: os clientes podem ser servidos por diferentes veículos (mais de um atende a um cliente)
- SVRP ou Stochastic VRP: alguns valores são randômicos (número de clientes, suas demandas, tempo para o serviço, tempo de viagem);
- PVRP ou Periodic VRP: as entregas podem ser feitas em alguns dias.

Problema de otimização - Parte II

CVRP ou Capacitated VRP: todo veículo tem uma capacidade limitada

Nesta variação do problem CVRP uma frota fixa de veículos de capacidade uniforme é estabelecida para atender demandas conhecidas referentes a um único produto proveniente de um depósito comum, procurando atingir um custo mínimo de trânsito (viagem).

- **Objetivo:** o objetivo é minimizar a frota de veículos e o custo de viagem, sendo que a demanda total de cada rota não pode exceder a capacidade do veículo que serve aquela rota.
- **Factibilidade:** uma solução é factível se a quantidade total associada a cada rota não excede a capacidade do veículo que serve a rota
- **Formulação:** Seja Q a capacidade de um veículo. Matematicamente, uma solução para o CVRP é a mesma do problema VRP, mas com a restrição adicional que a demanda total de todos os clientes atendidos na rota R_i não excede a capacidade do veículo Q : $\sum_{i=1}^c d_i \leq Q$, em que d é a demanda do cliente e c é o último cliente atendido na rota.

Problema de otimização - Parte II

Considerando capacidade limitada

Neste caso considera-se que os veículos tem capacidade limitada uniforme (ou seja, igual para todos). Uma boa solução para o problema é aquela que minimiza a soma das distâncias das rotas planejadas e não excede a capacidade limitada dos veículos

$$fitness(x) = \sum_{i=1}^m C(R_i) + \gamma overcapacity(x)$$

em que:

- x : é uma solução (um conjunto de rotas)
- i : é um veículo
- R_i : é a rota do veículo i
- m : é o número de veículos disponíveis
- $C(R_i)$: é a soma dos custos de ir de um cliente a outro, seguindo a ordem da rota
- $overcapacity(x)$: é uma função que retorna o estouro de capacidade de veículos gerado em cada uma das rotas ($overcapacity(x) : \sum_{i=1}^m estouro_i$.)
- γ : é um peso para a penalização do estouro da capacidade

Tarefas - Parte II

Nesta parte do trabalho o objetivo é ilustrar a aplicação de um Algoritmo Genético a um problema real. No entanto, problemas reais possuem restrições específicas que devem ser atendidas na representação usada no cromossomo, na função fitness, em operadores reparadores ou em operadores orientados ao problema (porém ainda, e como sempre em Algoritmos Genéticos, operadores probabilísticos).

Então é esperado que o grupo, considerando o problema **VRP com capacidade limitada**:

- implemente a representação do problema na forma de um cromossomo (o grupo é livre para escolher a representação que quer usar)
- implemente a função fitness (o grupo é livre para escolher a função fitness que quer usar - na literatura há diferentes estratégias para medir o fitness nesse problema)
- aplique os operadores clássicos de Algoritmos Genéticos no problema e analise as limitações e problemas observados
- aplique um operador de correção e refaça os testes e análises
- implemente (pelo menos) um crossover e (pelo menos) uma mutação orientada ao problema e refaça os testes e análises

Obs.: Considerar o problema VRP com capacidade infinita pode ser interessante para conhecer melhor as nuances do problema

Operadores - Parte II

Para encontrar operadores orientados a problema consulte e funções fitness alternativas consulte:

- Bjarnadóttir, A. S. Solving the Vehicle Routing Problem with Genetic Algorithms. Master Thesis. Technical University of Denmark, 2004.
- Masum, A. K. M et al. Solving the Vehicle Routing Problem using Genetic Algorithm. Int. J. of Advanced Computer Science and Applications, vol.2, n.7, 2011, pp. 126–131.
- Kovács, Á. Solving the Vehicle Routing Problem with Genetic Algorithm and Simulated Annealing. Master Thesis, Högskolan Dalarna, Sweden, 2008.
- infinitas outras referências que você pode encontrar em sites acadêmicos.

Instâncias de testes - Parte II

Instâncias de testes podem ser acessadas em: <http://www.bernabe.dorronsoro.es/vrp/>. No link "Instances" você encontra várias instâncias de teste para o **CVRP**. Escolha instâncias que sejam interessantes para os testes de seu trabalho. Desafie o seu trabalho.

Como requisito mínimo você deve fazer teste em:

- uma instância de dimensão pequena de cada um dos grupos (A, B, C) de Augerat et al.
- uma instância de dimensão alta de um dos grupos de Augerat et al.

Você deve ler o arquivo de explicação sobre os conjuntos de teste (file format) disponível junto com os conjuntos.

Entrega - Parte II

- relatório: contendo as análises realizadas e decisões tomadas - nos mesmos moldes do relatório da parte I
- artefatos: códigos e outros arquivos gerados nos testes
- vídeo: um vídeo de 3 a 10 minutos de duração explicando APENAS a representação usada (se usou mais de uma, explique também), a função fitness (se usou mais de uma explique também) e os novos operadores

Trabalho de AG - Algoritmos Genéticos

● Sarajane Marques Peres - sarajane@usp.br

Disciplina de Inteligência Artificial
Graduação em Sistemas de Informação - SI
Escola de Artes, Ciências e Humanidades - EACH
Universidade de São Paulo - USP