

Análise dos algoritmos ACO e PSO

Autor

André Furlan - ensismoebius@gmail.com

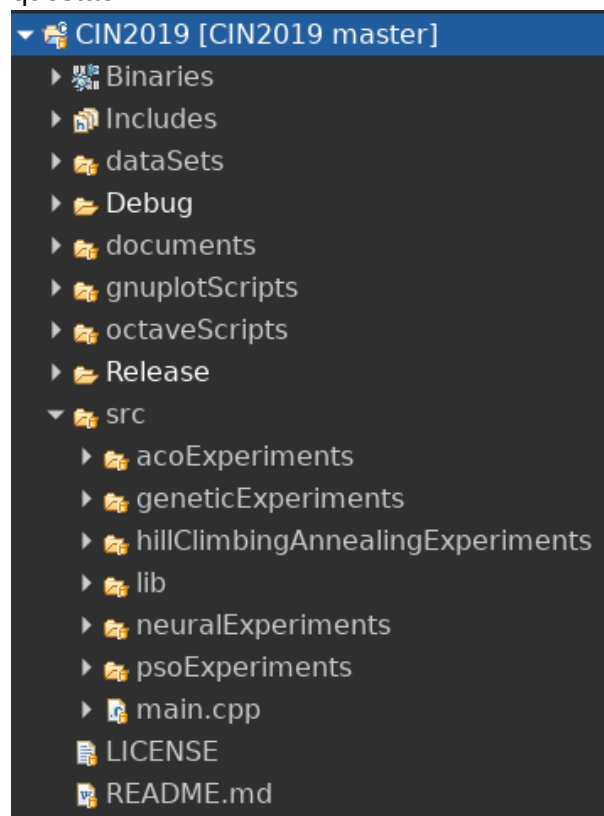
Onde encontrar o projeto

Este projeto está licenciado sob a licença GPL versão 3 e está armazenado em um repositório compartilhado no endereço:

<https://github.com/ensismoebius/computacaolInspiradaPelaNatureza>

A estrutura do projeto

Para melhor acompanhamento deste trabalho é importante entender a estrutura de arquivos do projeto em questão:



O **diretório principal** do projeto chama-se “src” é nele que todos os códigos residem.

Para o conteúdo deste documento o diretório de interesse é o “**acoExperiments**” e “**psoExperiments**” são neles que serão executados os testes.

No diretório “**lib**” estão todas as bibliotecas derivadas do desenvolvimento deste trabalho cada uma delas procura agrupar e organizar as lógicas e técnicas empregadas de forma que as mesmas possam ser usadas em qualquer outro sistema, ou seja, procurou-se diminuir ao máximo o acoplamento de código.

Por fim perto do final da figura se percebe um arquivo chamado “main.cpp” dentro do qual estão as chamadas para todos os experimentos contidos dentro do projeto.

Tecnologia necessárias

- Para que se possa reproduzir os experimentos aqui descritos antes de tudo é necessário ter instalado em seu computador os programas:
- Compilador C++
- GNU/Octave e seu respectivo pacote de estatísticas
- Gnuplot
- Recomenda-se também o computador com sistema operacional GNU/Linux não sendo garantida a execução dos programas em outro sistema operacional embora o mesmo seja totalmente possível.

As bibliotecas

As bibliotecas contendo os algoritmos em questão foram confeccionadas segundo o paradigma de orientação a objetos, as bibliotecas confeccionadas foram separadas em:

Para o ACO:

- **ACOAnt.cpp** - Que representa a formiga o sistema
- **ACOMap.cpp** - Que representa o mapa onde os pontos serão colocados
- **ACOPoint.cpp** - Representa os pontos no mapa

Para o PSO:

- **PSO.cpp** - Usada para encapsular toda a lógica do algoritmo
- **PSOLimits.cpp** - Define os limites do espaço de busca
- **PSONeighborhood.cpp** - Contém referências para as/os vizinhos do ponto
- **PSOParticle.cpp** - O ponto em si

Os experimentos

PSO

O experimento usando PSO tem como objetivo atender ao enunciado abaixo:

Utilize o PSO para minimizar a seguinte função no intervalo

$[-5, +5]$

$[-5, +5]$

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

Inclua em seu relatório um gráfico que mostre o valor mínimo e médio de $f(x, y)$ ao longo das iterações, além das configurações utilizadas e outros resultados que julgar interessantes.

Experimento

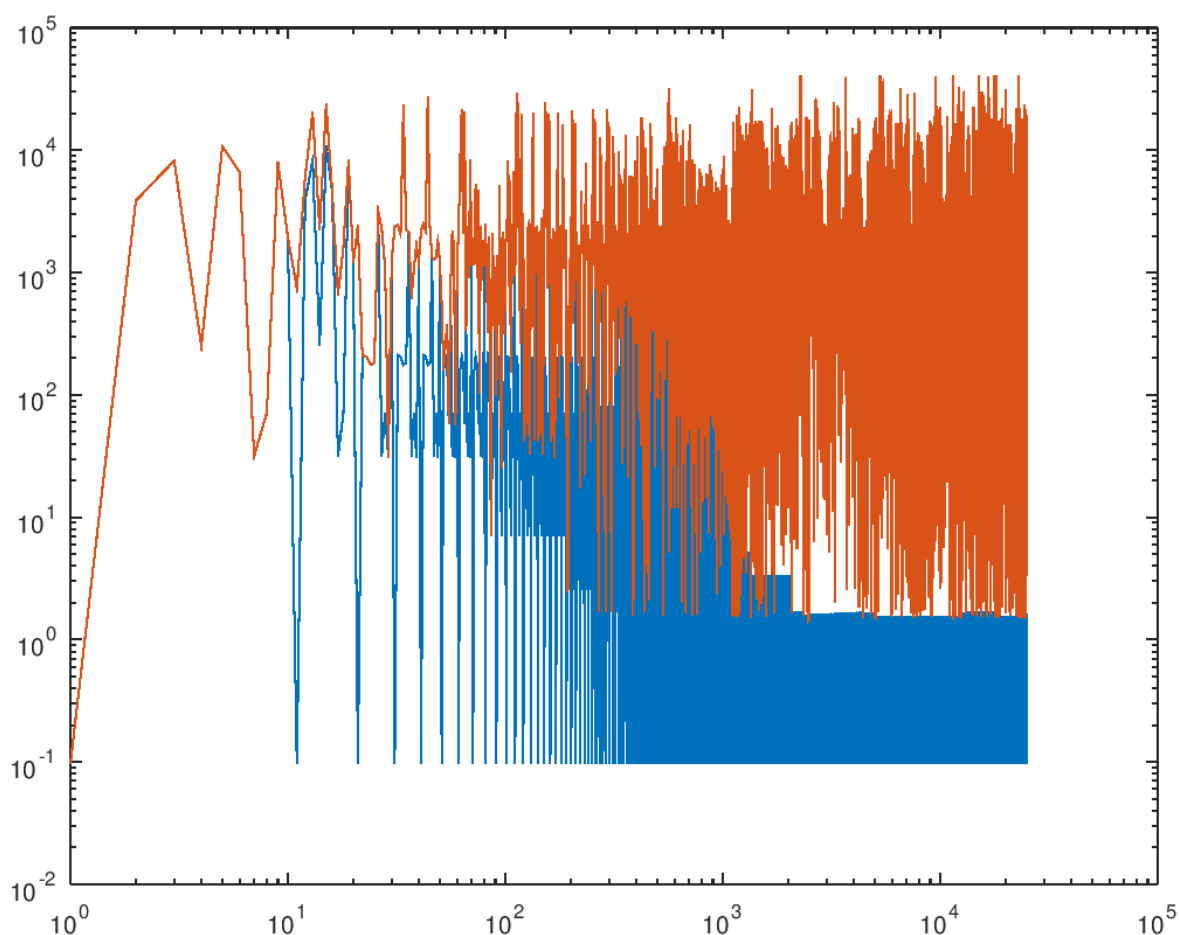
Dado	Valor
Quantidade de indivíduos na população	10
Taxa de confiança no indivíduo	1,55
Taxa de confiança no grupo	2,55
Velocidade máxima	2

Velocidade mínima	-2
Iterações	10000

O valor ótimo obtido da função 1,527100.

No gráfico a linha em azul indica os melhores resultados em cada ciclo e a linha vermelha indica a média dos valores para cada ciclo, ou seja, a soma do menor valor com o valor obtido dividido por 2.

No gráfico percebe-se que até aproximadamente a décima interação ambos valores praticamente se equivalem, no entanto, a partir daí percebe-se que os valores em azul vão convergindo para o mínimo enquanto que os outros valores flutuam como era de se esperar.



ACO

Os experimentos usando ACO tem como objetivo atender ao enunciado abaixo:

Utilize o ACO para encontrar o menor caminho no problema do Caixeiro Viajante com 52 cidades do arquivo berlin52.tsp, disponível no Moodle, e representado na figura ao lado. Inclua em seu relatório um gráfico que mostre a distância total do menor caminho encontrado ao longo das iterações, além das configurações e outros resultados que julgar interessantes.

Inclua também um gráfico mostrando as cidades em suas respectivas coordenadas e o traçado do menor caminho encontrado, para uma análise visual do resultado.

Antes que o valor ótimo de parâmetros fosse encontrado foram feitos experimentos. Os dados dos mesmos estão expostos na tabela abaixo:

A mudança de cores em cada coluna indica uma mudança de valor.

Iterações	taxa de decaimento	feromônio inicial	feromônio heurístico inicial	influência do feromônio	influência da distância	taxa de atualização de feromônio	distância ótima alcançada
15	0,1	0,01	0,01	1	2	0,5	8980,2
15	0,1	0,01	0,05	1	2	0,5	9312,97
15	0,1	0,01	0,1	1	2	0,5	8980,92
15	0,1	0,1	0,2	1	2	0,5	8980,92
15	0,9	0,1	0,1	1	2	0,9	9585,89
15	0,9	0,1	0,1	1	1	0,5	9190,07
15	0,9	0,1	0,1	1	5	0,5	9138,24
15	0,9	0,1	0,1	1	3	0,5	8651,95
16	0,9	0,1	0,1	1	3	0,5	9764,08

Heurística

Neste trabalho foi escolhido um método heurístico guloso, as arestas foram selecionadas levando-se em consideração a menor distância euclidiana disponível, ou seja, cada próxima aresta foi selecionada levando-se em consideração a menor distância euclidiana que representa, e não a distância total do caminho.

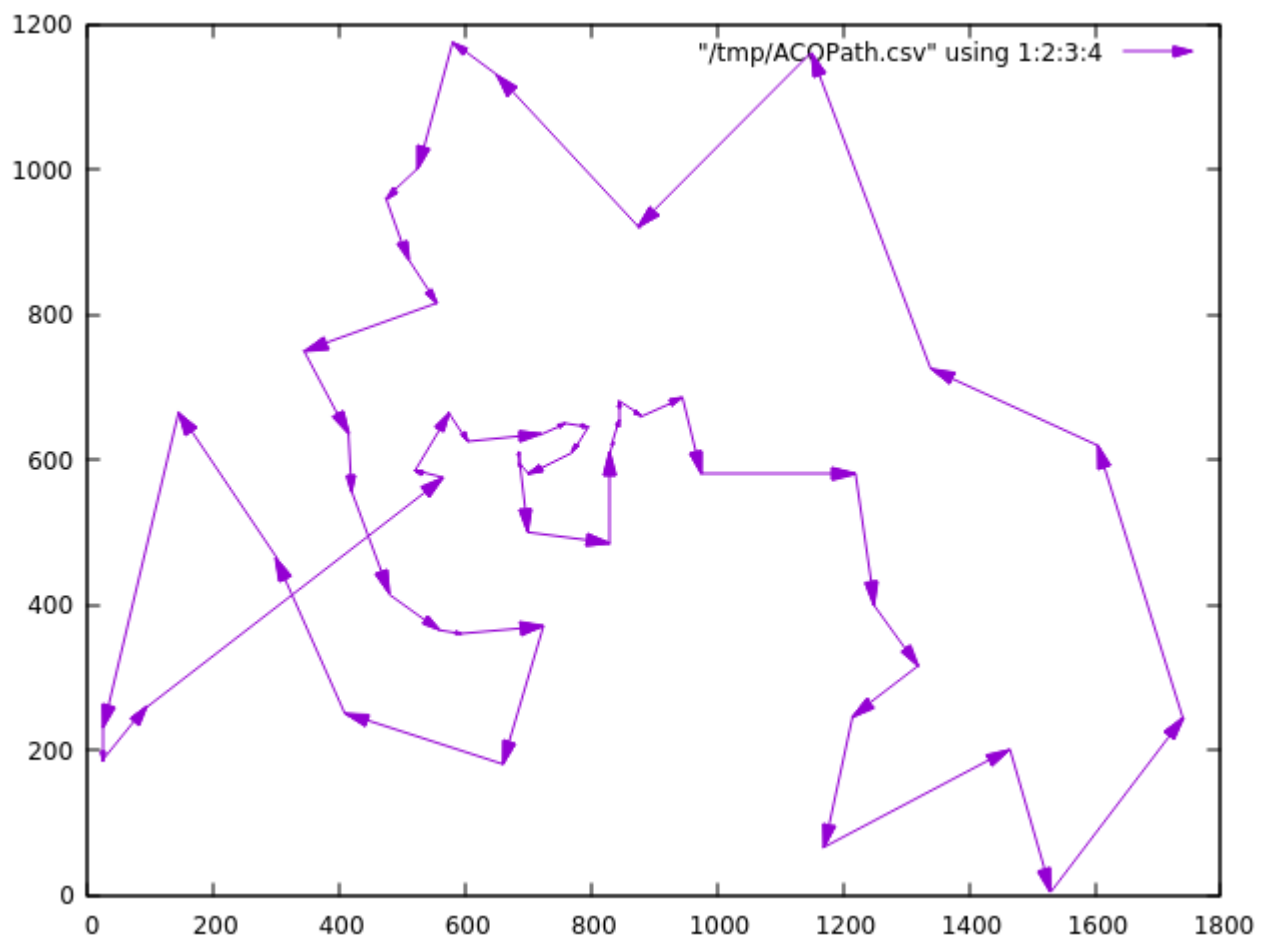
Experimento com maior taxa de sucesso

Após as várias tentativas descritas acima, chegou-se os valores abaixo:

Dado	Valor
Taxa de decaimento do feromônio	0,9
Quantidade de feromônio inicial em cada aresta	0,1
Quantidade de feromônio inicial em cada aresta heurística	0,1
Influência do feromônio na escolha	1
Influência da distância na escolha	2

Iterações	15
Quantidade de coordenadas	52
taxa de atualização de feromônio	0.5

Esses parâmetros resultaram em uma **distância ótima de 8194,03** e no gráfico abaixo, percebe-se que ainda que há dois cruzamentos nos caminhos indicando que este resultado pode ser melhorado, no entanto, ao se tentar executar o algoritmo mais vezes o mesmo hora chegava em uma solução pior, hora chegava na mesma solução. Como se pode perceber na tabela que lista os experimentos anteriores a mudança dos parâmetros parece influenciar mais do que o número de iterações a partir de um certo número de ter iterações;



É interessante notar que, nesse caso, o algoritmo se comporta de forma melhor quando a quantidade de feromônios depositada pelo método heurístico é igual a quantidade padrão de feromônio em todas as arestas. Aparentemente a deposição inicial de feromônios no mapa tem grande influência na convergência do algoritmo, isso, é claro, dependendo do método heurístico escolhido. Vai abaixo o gráfico representativo da convergência do algoritmo para esse problema:

