

Uma abordagem heurística para o problema da Árvore Geradora Mínima de Grau Restrito

Rubem Kalebe Santos
26 de novembro de 2015

Background

- O problema da AGMGR.
- Soluções exatas implementadas anteriormente.
- Soluções heurísticas implementadas aqui:
 - Algoritmo baseado em Otimização por Colônia de Formigas.
 - Algoritmo heurístico.

Solução 1: Colônia de Formigas (1/3)

- Segue bem o padrão dos algoritmos de Otimização por Colônia de Formigas.
- De maneira geral, um conjunto de arestas $E = \{e_{ij} | i \in V, j \in V, i \neq j\}$, conjunto de todas as arestas possíveis no grafo, contém os componentes das soluções que as formigas construirão de forma incremental.

Solução 1: Colônia de Formigas (2/3)

```
1: function ANTSYSTEM
2:   Ordene, de forma crescente pelo peso, as arestas
3:   for  $t = 1$  to maxiterations do
4:     Coloque cada formiga em uma aresta aleatória
5:     for  $k = 1$  to  $|V| - 1$  do
6:       while formiga  $k$  não construir a solução  $S_k$  do
7:         Selecione a próxima aresta pela regra  $p_{ij}^k$  ou
           aleatoriamente
8:         Calcule o custo  $C_k$  da solução  $S_k$ 
9:         if  $C_k < C^*$  then
10:            $S^* = S_k$ 
11:            $C^* = C_k$ 
12:         Atualize os feromônios
return  $S^*$ 
```

Solução 1: Colônia de Formigas (3/3)

- Parâmetros:
 - *Alpha*: 1.0;
 - *Beta*: 5.0;
 - Quantidade inicial de feromônio: 5.0;
 - Máximo de iterações: 300;
 - Taxa de evaporação do feromônio: 0.5;
 - Fator de número de formigas usadas
($numAnts = fator \times |V|$): 0.8;
 - Taxa de evaporação do feromônio: 0.5;
 - Coeficiente de depósito de feromônio = 500.0;
 - Probabilidade de seleção puramente randômica: 0.1;
- Função `pow()`.

Solução 2: Algoritmo heurístico

- Orientado por algoritmo de *backtracking*.
- Interrompido quando provavelmente não houver mais nenhuma solução “de futuro”.
- $escape = (|E|^2) \times 5$.
- Eventualmente melhora a solução em comparação com o algoritmo guloso.
- Eventualmente retorna a melhor resposta mais rápido do que o algoritmo de *backtracking*.

Análise de resultados

- O algoritmo heurístico produz bons resultados e em pouco tempo, sendo ideal para comparar a eficácia de outros algoritmos heurísticos e interessante para uso.
- Para instâncias euclidianas os algoritmos parecem mais estáveis.
- O algoritmo heurístico tende a encontrar soluções melhores na maioria dos casos, mas, eventualmente, o *Ant System* encontra soluções melhores, mostrando seu poder.
- Percebe-se também que a restrição de grau não afeta tanto o desempenho da solução, sendo o número de vértices o principal fator de impacto no tempo gasto pela solução.

Considerações finais

- Estudo avançado sobre o problema da AGMGR.
- Módulos desenvolvidos:
 - Gerador de instâncias.
 - Validador.
- Algoritmo heurístico produz bons resultados para comparar com outras implementações ou ser usado.

Dúvidas?