

Proposta de algoritmo genético híbrido para o problema da Árvore Geradora Multiobjetivo baseada no operador OWA

Prof^a. Dr^a. Elizabeth Ferreira Gouvêa Goldberg
Islame Felipe da Costa Fernandes

Departamento de Informática e Matemática Aplicada - DIMAp
Universidade Federal do Rio Grande do Norte - UFRN

Natal, 10 outubro de 2017

Agenda

- 1 Introdução
- 2 Algoritmo memético
- 3 Experimentos computacionais

Introdução

- 1 Abordaremos a Árvore Geradora Multiobjetivo com critério *OWA*.
- 2 Objetivos do trabalho: propor um algoritmo genético hibridizado com *simulated annealing* e compará-lo com o melhor exato da literatura (FERNÁNDEZ et al., 2017);
- 3 Serão relatados experimentos para 39 grafos completos, de 30 a 500 vértices, das classes *correlated* e *anti-correlated*, com critério *k-trimmed*, para 10 objetivos.

Algoritmo memético

Algoritmo 1: M-SA - Algoritmo genético hibridizado com SA

Entrada: Grafo $G(V, E)$

Saída: Árvore com melhor custo OWA encontrado

```
1 P = getInitialPop(#popSize);
2 best =  $\infty$ ;
3 setBest(P, best);
4 para g = 1 ... #max_gen faça
5     Q = {};
6     para r = 1 ... #popSize faça
7         Selecione randomicamente  $p_1, p_2, p_3, p_4$  em P;
8         pai = torneioBinario( $p_1, p_2$ );
9         mae = torneioBinario( $p_3, p_4$ );
10        prob = random(0, 1);
11        se prob < #propCross então
12            | filho = crossover(pai, mae);
13        senão
14            | filho = doRandomWalk()
15        fim
16        prob = random(0, 1);
17        se prop < #propMutation então
18            | mutation(filho);
19        fim
20        SA(filho);
21        Q = Q  $\cup$  {filho};
22    fim
23    P = getElite(P  $\cup$  Q);
24    setBest(P, best);
25    se best não mudar em 5 gerações consecutivas então
26        | renova(P);
27    fim
```

População inicial

- Representação do cromossomo: lista de arestas (RAIDL; JULSTROM, 2003)
- Tamanho: $\#popSize$
- Métodos: *Random Walk* (RAIDL; JULSTROM, 2003) e *rmcPrim* (KNOWLES, 2002)
- Vetores de escalarização ω para o *rmcPrim*:
 - ▶ SPEA/R (JIANG; YANG, 2017)
 - ▶ *k-centrum* : $k \in [1, M]$, $\omega_1 = \dots = \omega_k = 1/k$ e $\omega_{k+1} = \dots = \omega_M = 0$.
 - ▶ *k-trimmed* : $k \in [1, \frac{M}{2}]$, $\omega_1 = \dots = \omega_k = 0$,
 $\omega_{k+1} = \dots = \omega_{M-k} = \frac{1}{M-2k}$ e $\omega_{M-k+1} = \dots = \omega_M = 0$
 - ▶ *Hurwicz* : $\alpha \in \{0.2, 0.3, 0.4, 0.6, 0.7, 0.8\}$, $\omega_1 = \alpha$,
 $\omega_2 = \dots = \omega_{M-1} = 0$ e $\omega_M = 1 - \alpha$.

Operadores genéticos (RAIDL; JULSTROM, 2003)

Figure: Operador *crossover* (RAIDL; JULSTROM, 2003)

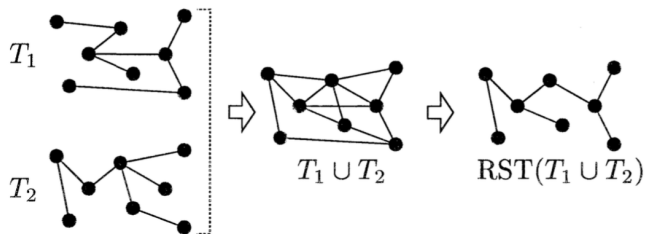
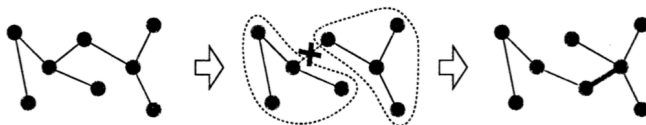


Figure: Operador Mutação (RAIDL; JULSTROM, 2003)



Simulated Annealing

Algoritmo 2: *Simulated Annealing*

Entrada: sol : solução, T_0 : temperatura inicial, L_0 : quantidade inicial de iterações

```
1  $s = sol$ ;  
2  $T_i = T_0$ ;  
3  $L_i = L_0$ ;  
4 enquanto  $T_i > 0$  faça  
5     para  $i=1 \dots L_i$  faça  
6         escolha randomicamente dois pares de aresta  $(v_l, v_{l+1})$  e  $(v_j, v_{j+1})$  em  $s$ ;  
7         remova  $(v_l, v_{l+1})$  e  $(v_j, v_{j+1})$  de  $s$ ;  
8         religue os vértices  $v_l, v_{l+1}, v_j, v_{j+1}$  e produza  $s'$ ;  
9          $\Delta C = OWA(s') - OWA(s)$ ;  
10        se  $\Delta C \leq 0$  então  
11             $s = s'$ ;  
12            se  $OWA(s) < OWA(sol)$  então  
13                 $sol = s$ ;  
14            fim  
15        senão  
16            se  $random(0, 1) < \exp\{-\frac{\Delta C}{T_i}\}$  então  
17                 $s = s'$   
18            fim  
19        fim  
20    fim  
21     $T_i = \frac{T_i}{F_T}$ ;  
22     $L_i = L_i * F_L$ ;  
23 fim
```

Estratégia de renovação da população

- 50% dos indivíduos substituídos por soluções geradas pelo *rmcPrim* com vetor de escalarização w (pesos OWA da instância)
- 1/3 sofrem mutação

Instâncias

- ❶ 60 instâncias
- ❷ 30 a 500 vértices
- ❸ 10 objetivos
- ❹ 3 grupos de instâncias: $n.1$, $n.2$, $n.3$
- ❺ Classes: *correlated* e *anti-correlated* (KNOWLES, 2002)
 - ❶ *correlated* Coeficientes dos respectivos grupos: 0.2, 0.5 e 0.85
 - ❷ *anti-correlated* Coeficientes dos respectivos grupos: -0.2, -0.5 e -0.85
- ❻ Critério do operador OWA: *k-trimmed*
 - ❶ Dado um inteiro $k \in [1, \frac{M}{2}[$ e atribui-se $\omega_1 = \dots = \omega_k = 0$,
 $\omega_{k+1} = \dots = \omega_{M-k} = \frac{1}{M-2k}$ e $\omega_{M-k+1} = \dots = \omega_M = 0$.

Parâmetros

Table: Parâmetros para o *M-SA*

$\#max_gen = 50$	$T_0 = 30$
$\#popSize = 100$	$F_T = 1,8$
$\#propCross = 0,97$	$L_0 = 15$
$\#propMutation = 0,1$	$F_L = 1,8$
$\#perTo = 0,03$	$\kappa = 6$

Metodologia dos experimentos

- Para cada instância, 30 execuções
- Dados:
 - ▶ Qualidade da solução retornada;
 - ▶ Evolução da qualidade ao longo das gerações do processo evolucionário
 - ▶ Tempo para atingir a melhor solução
 - ▶ tempo total
 - ▶ Quantidade de vezes em que a população precisou ser renovada
 - ▶ Contribuição dos operadores de *crossover*, mutação e do SA.
- Desvio percentual:

$$d_p = \frac{OWA(s_h) - OWA(s_s)}{OWA(s_s)} * 100 \quad (1)$$

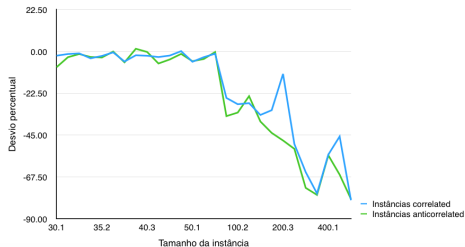
onde s_s e s_h são, respectivamente, a solução retornada pelo *solver* e pelo *M-SA*.

Resultado - Qualidade de soluções

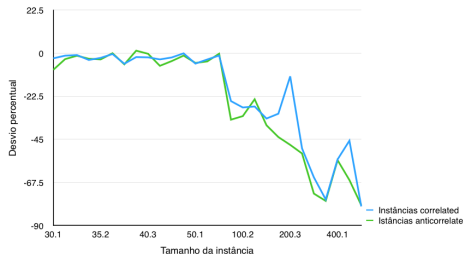
- O *solver* atingiu 1 hora em todas as instâncias:
 - ▶ Solução factível de 30 a 400 vértices, tanto na classe *correlated* quanto na *anticorrelated*
 - ▶ *gap* na classe *correlated*: de 81,68% a 100%
 - ▶ *gap* na classe *anticorrelated*: de 84,23% a 100%

Resultado - Qualidade de soluções

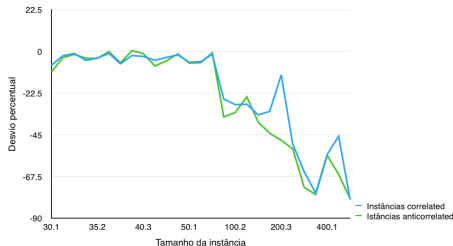
Desvio percentual da solução mediana em relação à solução do solver



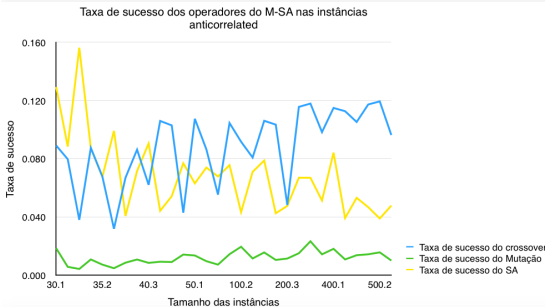
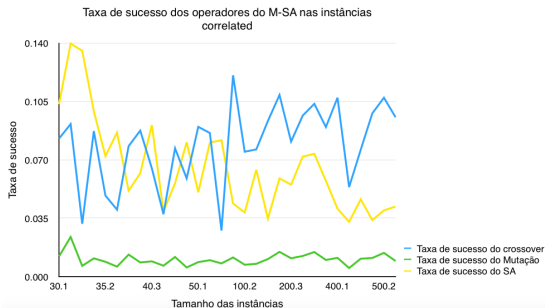
Desvio percentual da solução média em relação à solução do solver



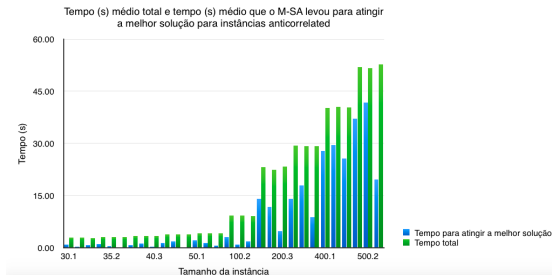
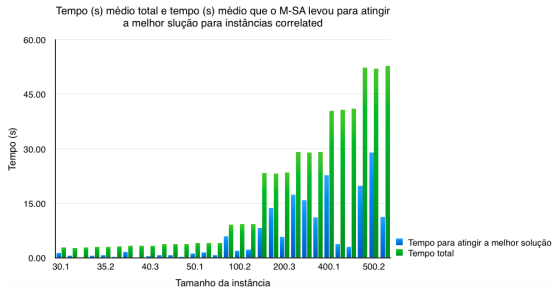
Desvio percentual da solução mínima em relação à solução do solver



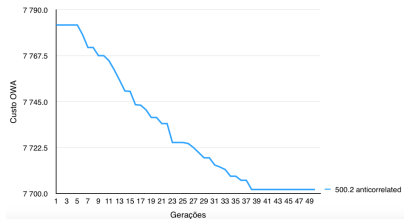
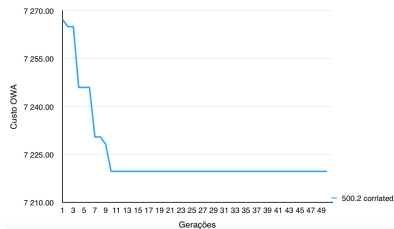
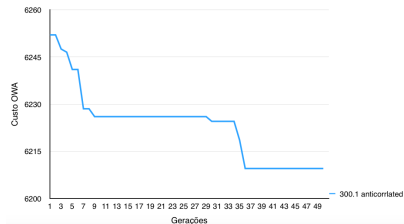
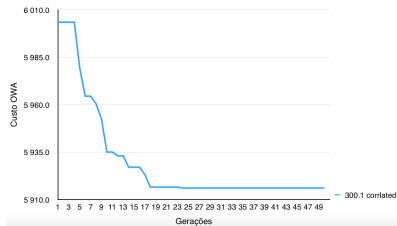
Resultado - taxa de sucesso dos operadores



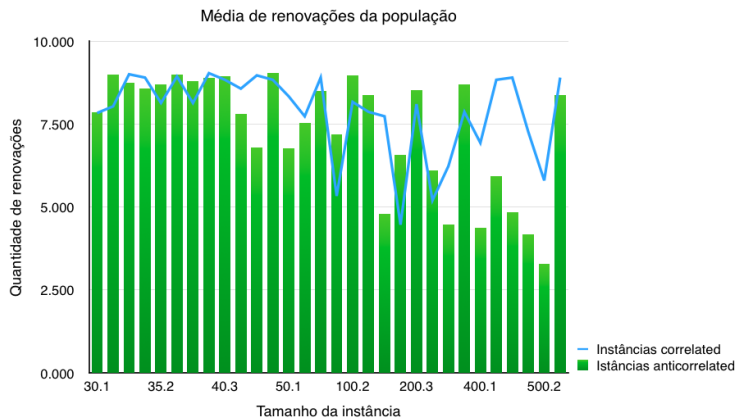
Resultado - Tempo



Resultado - Exemplos do evolução




Resultado - Quantidade de renovações





Considerações Finais e trabalhos futuros


- O *M-SA* conseguiu ser melhor que o exato!!
- Evolução algumas vezes foi tímida
- Propor algoritmo transgenético
- Testar com mais instâncias (até 1000 vértices) e outros critérios OWA
- Parâmetros com IRACE

Referências Bibliográficas I

 FERNÁNDEZ, E. et al. Ordered weighted average optimization in multiobjective spanning tree problem. *European Journal of Operational Research*, Elsevier, v. 260, n. 3, p. 886–903, 2017.

 JIANG, S.; YANG, S. A strength pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 21, n. 3, p. 329–346, 2017.

 KNOWLES, J. *Local-search and hybrid evolutionary algorithms for Pareto optimization*. Tese (Doutorado) — Department of Computer Science, University of Reading, Reading, UK, 2002.

 RAIDL, G. R.; JULSTROM, B. A. Edge sets: an effective evolutionary coding of spanning trees. *IEEE Transactions on Evolutionary Computation*, v. 7, n. 3, p. 225–239, 2003.