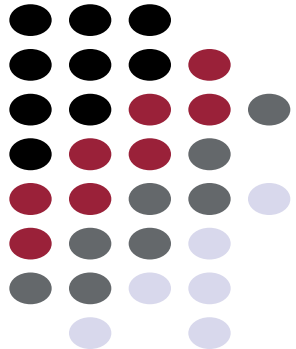
Rastrigin

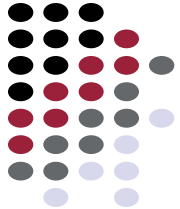
Computação Evolucionária

Daniel Reis

Universidade Federal de Ouro Preto
UFOP 2017

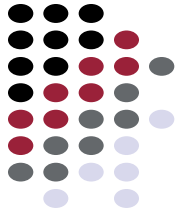


Código



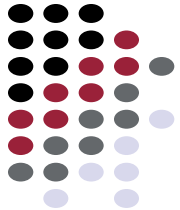
- População inicial -> aleatória no intervalo de -5,12 a 5,12;
- Estratégia utilizada:
  - Elitismo (mantendo 20% dos pais);
  - Crossover
    - 1 pt = Parte 1 do pai1 e parte 2 do pai2
    - 2 pts = (Parte 1 e 3 do pai com > FO e parte 2 do pai com < FO)
  - Mutação por bit.
- crossover = 0,8; mutação = 0,05; mu = 10; lambda = 100;
- tampop = 100, gerações = 300, nvar = 100, execuções = 30;
- precisão = 10; numBest = 100 ou 50; numEst = 100 ou 50;

Main



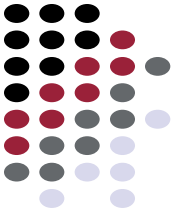
- rnd [1,8]
  - Caso 1: AG -> **crossover 1 pt** + mutação + elitismo
  - Caso 2: AG -> **crossover 2 pts** + mutação + elitismo
  - Caso 3: ES -> mutação + mu + lambda + **sem elitismo**
  - Caso 4: ES -> mutação + mu + lambda + **elitismo**
  - Caso 5: DE -> crossover + mutação + **sem elitismo**
  - Caso 6: DE -> crossover + mutação + **elitismo**
  - Caso 7: AED -> precisão = 10; numBest = 100; numEst = 100;
  - Caso 8: AED -> precisão = 10; numBest = 50; numEst = 50;

- **Cria e avalia** pop inicial
- **Salva melhor e pior** indivíduo e as **FO**
- Calcula o número de pais (% **elitismo**)
- Para cada **geração**
  - **Move** os x% melhores indivíduos (**pop -> elite**)
  - **Gera** os (tamPop - numPais)
    - Se **rnd [0,1]** < taxa de **crossover**
      - Seleciona dois **pais aleatórios** (diferentes)
      - **descendente = pai1**
      - **Crossover** (1 ou 2 pontos)
      - **Mutação por bit** no descendente
      - **Avalia** descendente
      - **Adiciona** FO do descendente na lista e descendente na **novapop**
  - Define **sobreviventes**: limpa (pop) -> ordena (novaPop) -> corta (novaPop [tam = tamPop - numPais]) -> insere (pop = elite + novaPop)
  - **Atualiza** e salva **melhor e pior** indivíduo



- **Cria e avalia** pop inicial
- **Salva melhor e pior** indivíduo e as **FO**
- Calcula o número de pais (% **elitismo**)
- Para cada **geração**
  - **Move** os x% melhores indivíduos (**pop -> elite**)
  - **Gera** os (tamPop - numPais)
    - Para cada pai, **gerar lambda/mu filhos**
      - Se **rnd [0,1]** < taxa de **mutação**
        - **descendente = pai**
        - **Mutação por bit** no descendente
        - **Avalia** descendente
        - **Adiciona** FO do descendente na lista e descendente na **novapop**
  - Define **sobreviventes**: limpa (pop) -> ordena (novaPop) -> corta (novaPop [tam = tamPop - numPais]) -> insere (pop = elite + novaPop)
  - **Atualiza** e salva **melhor e pior** indivíduo

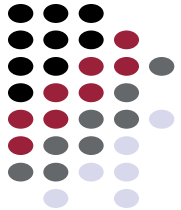
DE



- **Cria e avalia** pop inicial
- **Salva melhor e pior** indivíduo e as **FO**
- Calcula o número de pais (% **elitismo**)
- Para cada **geração**
  - **Move** os x% melhores indivíduos (**pop -> elite**)
  - **Gera** os (tamPop - numPais)
    - **rnd r0, r1, r2** (Seleciona três indivíduos aleatórios)
    - **Cria** um novo indivíduo (**trial**)
    - **Analisa a perturbação** (diferença entre xr1 e xr2) e insere no trial
    - **Mutação** (trial = trial * fator de mutação + xro)
    - **Pega** o indivíduo **target**
    - **Crossover** (**Combina** trial e target)
    - **Seleção** (**Maior FO** entre trial e target)
    - **Adiciona** FO do selecionado na lista e selecionado na **novapop**
  - Define **sobreviventes**: limpa (pop) -> ordena (novaPop) -> corta (novaPop [tam = tamPop - numPais]) -> insere (pop = elite + novaPop)
  - **Atualiza** e salva **melhor e pior** indivíduo

- **Cria e avalia** pop inicial binária
- **Salva melhor e pior** indivíduo e as **FO** e imprime a solução inicial
- Para cada **geração**
  - **Calcula** percentual **best** -> **proporcao[ ]**. Obs: considera qtde de “1”
  - **Limpa** novaPop
  - **Estimar** (gerar **est**). Obs: considera **proporcao[ ]**
    - **Gera cromossomos** do indivíduo
      - **rnd [0,1] <= proporcao [p]** -> 1. Senão 0.
      - **Avalia** descendente
      - **Adiciona** FO do descendente na lista e descendente na **novapop**
  - Define **sobreviventes**: insere (pop += novaPop) -> ordena (novaPop) -> corta (novaPop [tamPop])
  - **Atualiza** e salva **melhor e pior** indivíduo

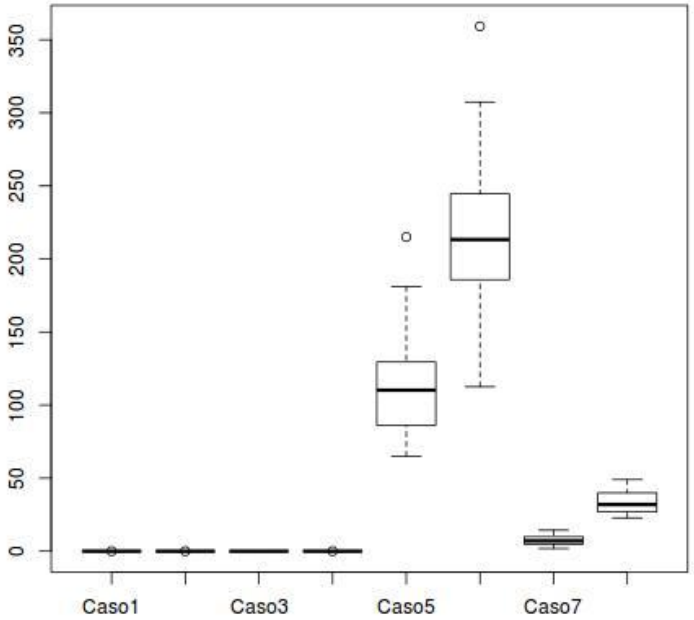
Resultados



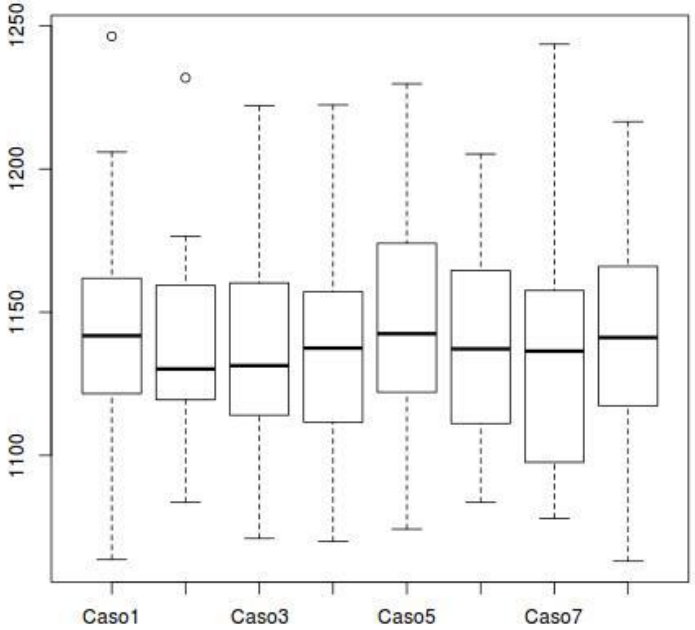
Caso	melhor	pior	dpmelhor	dppior
1	0.00050374424859001	1246.29973762838	0.0005031463461956 83	34.8175662513752
2	0.0003833473857639 55	1231.79719919037	0.00047519293245215	31.2847239886392
3	5.06914830111782e-08	1222.00833977536	1.23636101996351e-07	35.4473146979814
4	6.97943960403791e-08	1222.35107889754	1.72427765857092e-07	36.5419531375395
5	65.0232202033893	1229.71637402057	35.7277947123267	40.9149142047642
6	112.546745689816	1205.21088597165	53.4437075049772	33.7709396985736
7	1.87061598894741	1243.52271590101	3.3871596227479	40.0675418656033
8	22.8043180333814	1216.5203214165	7.25243929713469	38.022695067849

Caso	mediamelhor	mediapior	mediageral
1	0.00115148186208292	1142.84021157569	70.581843399614
2	0.00112543377568954	1138.28521639651	70.9745743700014
3	2.12922585281679e-07	1137.00265936556	45.6004658944708
4	2.57984781152724e-07	1135.74660849141	45.095722725319
5	116.146827141984	1147.61593232002	148.64563010013
6	218.27490226009	1141.01441051622	239.144919103583
7	7.53179033235407	1135.25401099169	332.112892886325
8	33.690744757209	1143.80125717346	223.616837464559

Boxplot

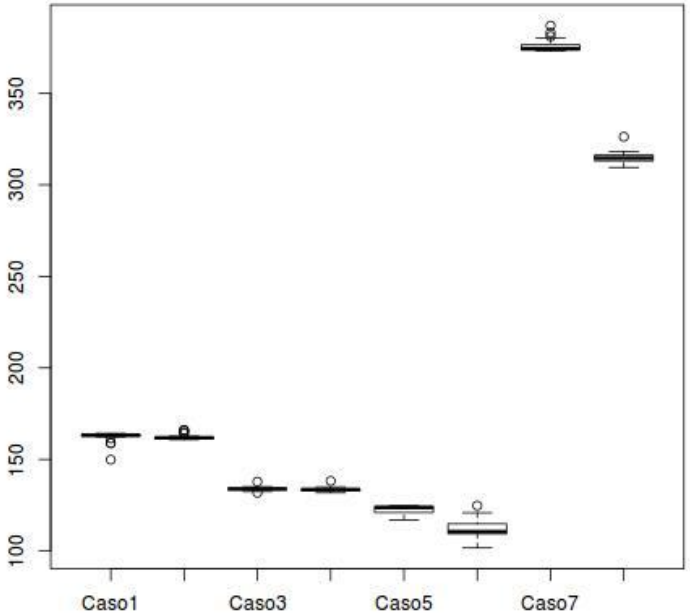


Melhor Resultado ~ Caso

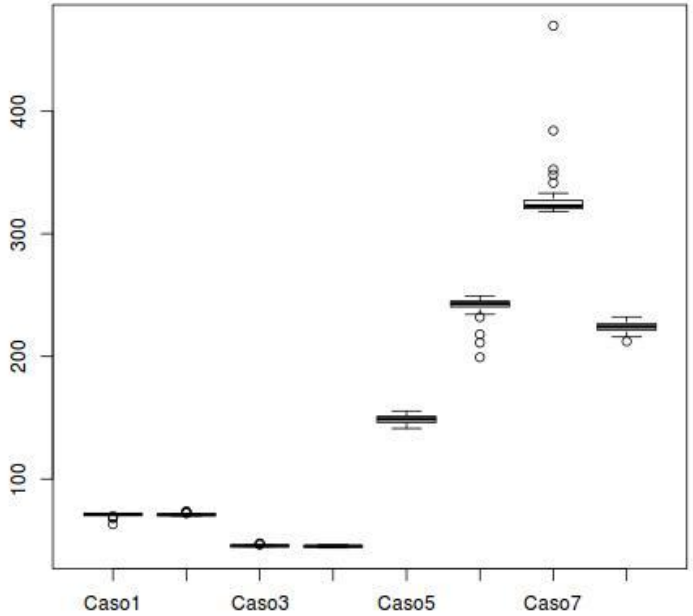


Pior Resultado ~ Caso

Boxplot

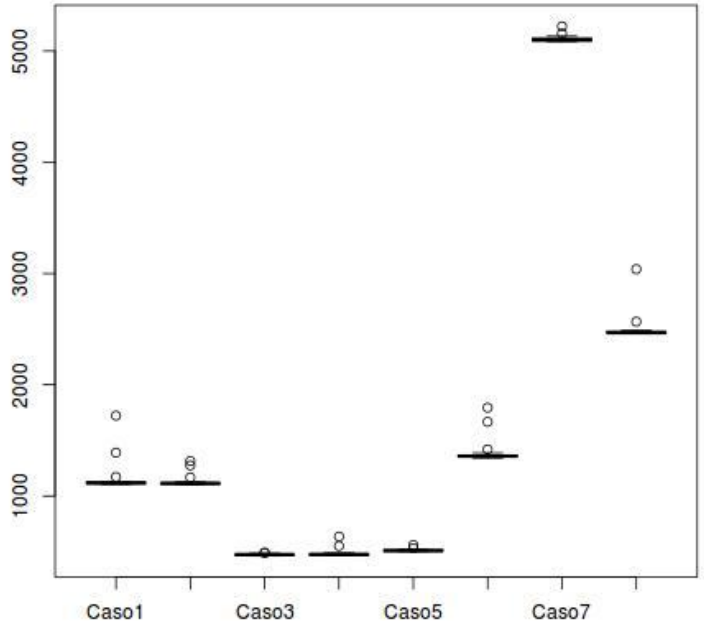


Desvio Padrão ~ Caso



Média ~ Caso

Boxplot



Tempo ~ Caso

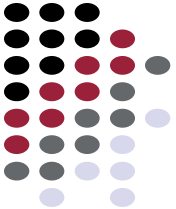
Teste T

	Caso	Tipo	pvalor	pvalorL	pvalorG
1	1-2	MelhorResultado ~ Caso	0.837397500557454	0.581301249721273	0.418698750278727
2	1-2	PiorResultado ~ Caso	0.596092707764019	0.701953646117991	0.298046353882009
3	1-2	Media ~ Caso	0.241685300310749	0.120842650155374	0.879157349844626
4	1-2	DesvioPadrao ~ Caso	0.546789502995648	0.726605248502176	0.273394751497824
5	1-2	Tempo ~ Caso	0.378797262613479	0.81060136869326	0.18939863130674
6	3-4	MelhorResultado ~ Caso	0.249968151259033	0.124984075629516	0.875015924370484
7	3-4	PiorResultado ~ Caso	0.892974182930605	0.553512908534698	0.446487091465302
8	3-4	Media ~ Caso	0.000999499699610394	0.999500250150195	0.000499749849805197
9	3-4	DesvioPadrao ~ Caso	0.140577048031777	0.929711475984111	0.0702885240158886
10	3-4	Tempo ~ Caso	0.175438483097599	0.0877192415487994	0.912280758451201
11	5-6	MelhorResultado ~ Caso	1.26292434542278e-11	6.3146217271139e-12	0.999999999993685
12	5-6	PiorResultado ~ Caso	0.498327434859925	0.750836282570037	0.249163717429963
13	5-6	Media ~ Caso	5.501775083418 92e-31	2.75088754170946e-31	1
14	5-6	DesvioPadrao ~ Caso	4.656123316130 49e-14	0.999999999999977	2.32806165806524e-14
15	5-6	Tempo ~ Caso	4.193496959677 56e-30	2.096748479838 78e-30	1
16	7-8	MelhorResultado ~ Caso	5.237197144422 81e-21	2.6185985722114e-21	1
17	7-8	PiorResultado ~ Caso	0.400191920881 28	0.200095960440 64	0.799904039559 36
18	7-8	Media ~ Caso	5.146186608592 06e-19	1	2.573093304296 03e-19
19	7-8	DesvioPadrao ~ Caso	9.399019227853 23e-59	1	4.699509613926 62e-59
20	7-8	Tempo ~ Caso	1.379179352080 51e-46	1	6.895896760402 53e-47

Teste T

- 1 e 2 -> **pvalor > 5%**; melhor = **2** **[2]**
- 3 e 4 -> media **3 > 4**; melhor = **3** **[3]**
- 5 e 6 -> melhor, media, tempo (**5 < 6**); dp (**5 > 6**); **[5]**
- 7 e 8 -> melhor (**7 < 8**); media, dp, tempo (**7 > 8**); **[7]**

Teste T



	Caso	Tipo	pvalor	pvalorL	pvalorG
1	2-3	MelhorResultado ~ Caso	1.34087946343624e-13	0.999999999999933	6.7043973171812e-14
2	2-3	PiorResultado ~ Caso	0.882404539875438	0.558797730062281	0.441202269937719
3	2-3	Media ~ Caso	1.04606610259173e-66	1	5.23033051295867e-67
4	2-3	DesvioPadrao ~ Caso	1.37050236785273e-60	1	6.85251183926366e-61
5	2-3	Tempo ~ Caso	7.50046339871408e-36	1	3.75023169935704e-36
6	5-7	MelhorResultado ~ Caso	1.69293495240917e-16	1	8.46467476204585e-17
7	5-7	PiorResultado ~ Caso	0.241892611090042	0.879053694454979	0.120946305545021
8	5-7	Media ~ Caso	2.36574083510996e-25	1.18287041755498e-25	1
9	5-7	DesvioPadrao ~ Caso	1.45482935855653e-88	7.27414679278266e-89	1
10	5-7	Tempo ~ Caso	2.86287824558848e-83	1.43143912279424e-83	1

- 2 e 3 -> melhor, media, dp, tempo (**2 > 3**); **[3]**
- 5 e 7 -> melhor (**5 > 7**); media, dp, tempo (**5 < 7**); **[7]**

Teste T

	Caso	Tipo	pvalor	pvalorL	pvalorG
1	3-7	MelhorResultado ~ Caso	6.31381153776235e-13	3.15690576888117e-13	0.999999999999684
2	3-7	PiorResultado ~ Caso	0.858544577745654	0.570727711127173	0.429272288872827
3	3-7	Media ~ Caso	1.49221136675282e-30	7.4610568337641e-31	1
4	3-7	DesvioPadrao ~ Caso	7.37737079570971e-64	3.68868539785485e-64	1
5	3-7	Tempo ~ Caso	5.95363347681477e-73	2.97681673840738e-73	1

- 3 e 7 -> melhor (**3 < 7**); media, dp, tempo (**3 > 7**); **[3]**

Referências

- Código disponível em:
  https://github.com/UFOP-CSI557/2017-02-atividades-danieel-reis