

TSP



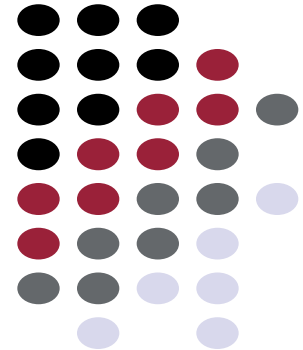
UFOP

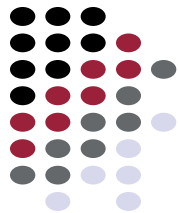
Universidade Federal
de Ouro Preto

Computação Evolucionária

Daniel Reis

Universidade Federal de Ouro Preto
UFOP 2017





- **numexecucoes = 30 [Número de execuções];**
- **geracoes = 100 [Número de gerações];**
- **tamPop = 1000 [Tamanho da população];**
- **pSelecionados = 0.05 [Percentual de selecionados da população];**
- **pCrossover = 0.7 [Percentual de crossover];**
- **pMutacao = 0.05 [Percentual de mutação];**
- **pBuscaLocal = 0.75 [Percentual de busca local];**

Arquivos: berlin52 e att48



- rnd [1,2]
 - Caso 1: AG
 - Caso 2: NOVO

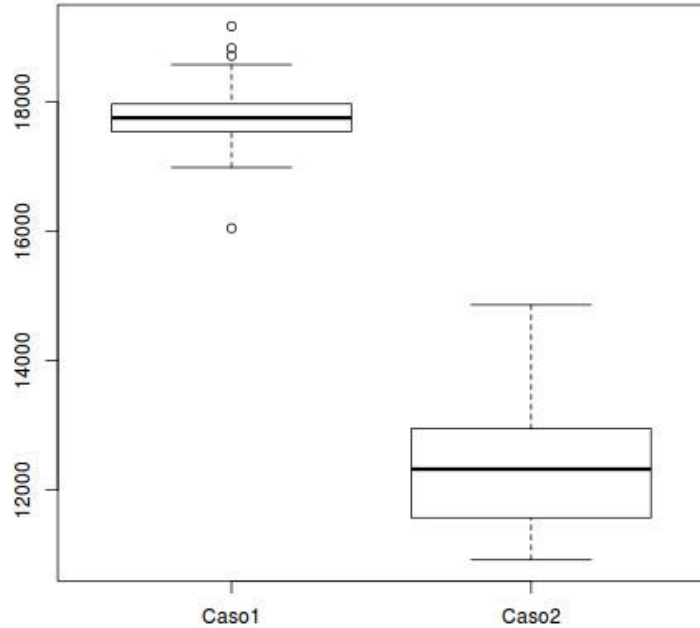
Obs: Busca local -> troca u por v



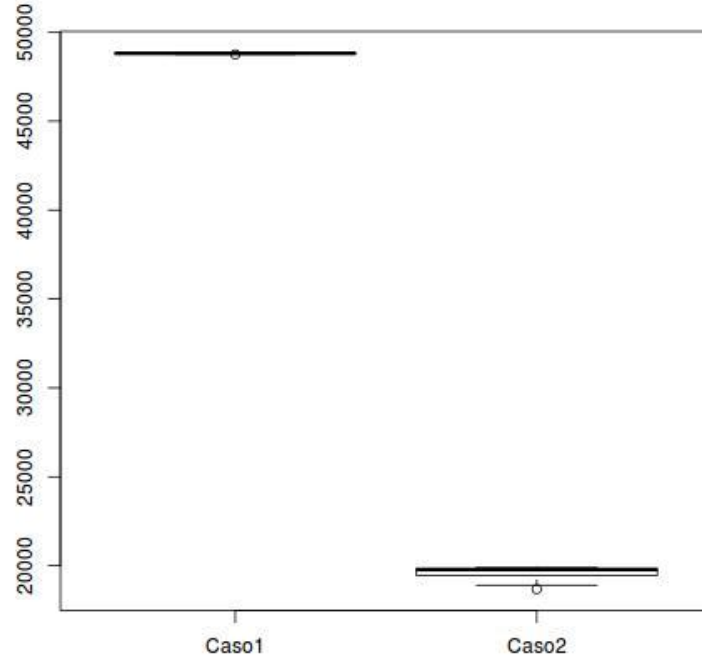
- **Cria e avalia** pop inicial
- Para cada **geração**
 - **Gera** (tamPop)
 - Se **rnd [0,1] <= taxa de crossover**
 - **Seleciona** 2 pais aleatórios na **novaPop**
 - **Crossover OX** -> cria 2 descendentes combinando os 2 pais
 - **Mutação SWAP** no descendente1
 - **Avalia** descendentes
 - Se **rnd [0,1] <= taxa busca local** -> busca local no descendente1
 - Se **rnd [0,1] <= taxa busca local** -> busca local no descendente2
 - **Adiciona** descendentes na **novaPop**
 - Define **sobreviventes**: insere (pop += novaPop) -> ordena (novaPop) -> corta (novaPop [tamPop])



- **Cria e avalia** pop inicial binária
- **Calcula numr (melhores indivíduos selecionados)** para novaPop)
- Para cada **geração**
 - **Move** os numr (pop -> novaPop)
 - **Gera** (tamPop - numr) -> descendentes com **base nos x melhores**
 - Se **rnd [0,1] <= taxa crossover**
 - **Seleciona** 2 pais aleatórios na **novaPop**
 - **Crossover OX** -> cria 2 descendentes combinando os 2 pais
 - **Mutação SWAP** no descendente1
 - **Avalia** descendentes
 - Se **rnd [0,1] <= taxa busca local** -> busca local no descendente1
 - Se **rnd [0,1] <= taxa busca local** -> busca local no descendente2
 - **Adiciona** descendentes na **novaPop**
 - Se **rnd [0,1] >= taxa mutação**
 - **Cria** descendente usando pai aleatório na **pop**
 - **Mutação SWAP** no descendente
 - **Avalia** descendente
 - Se **rnd [0,1] <= taxa busca local** -> busca local no descendente
 - **Adiciona** descendente na **novaPop**
 - Define **sobreviventes**: insere (pop += novaPop) -> ordena (novaPop) -> corta (novaPop [tamPop])

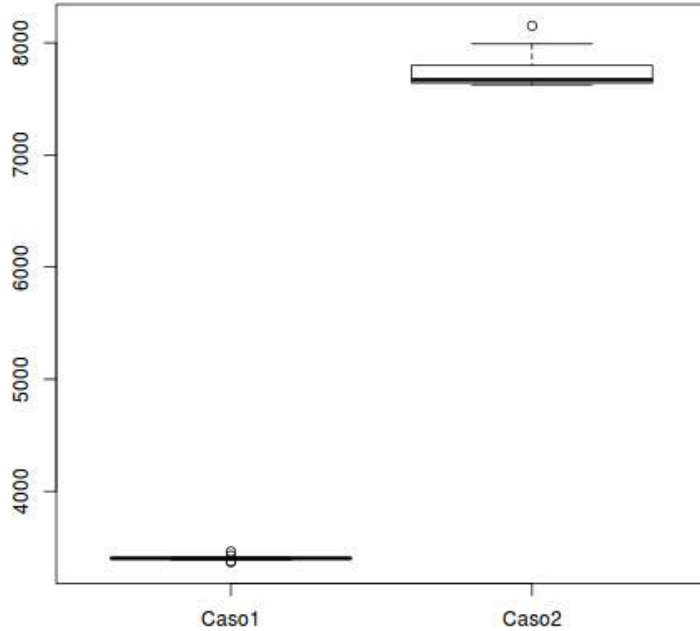
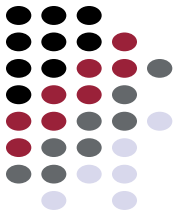


Melhor Resultado ~ Caso

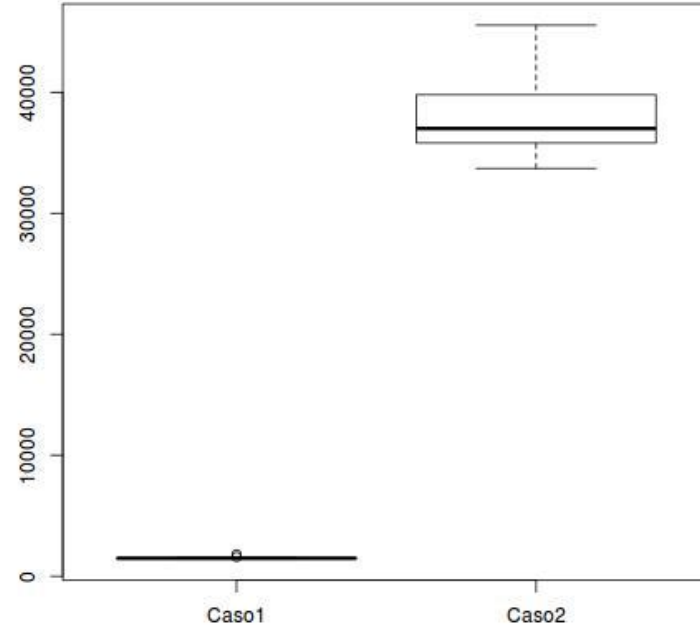


Média ~ Caso

Boxplot [ATT 48]



Desvio Padrão ~ Caso



Tempo ~ Caso

Teste T + Resultados [ATT 48]

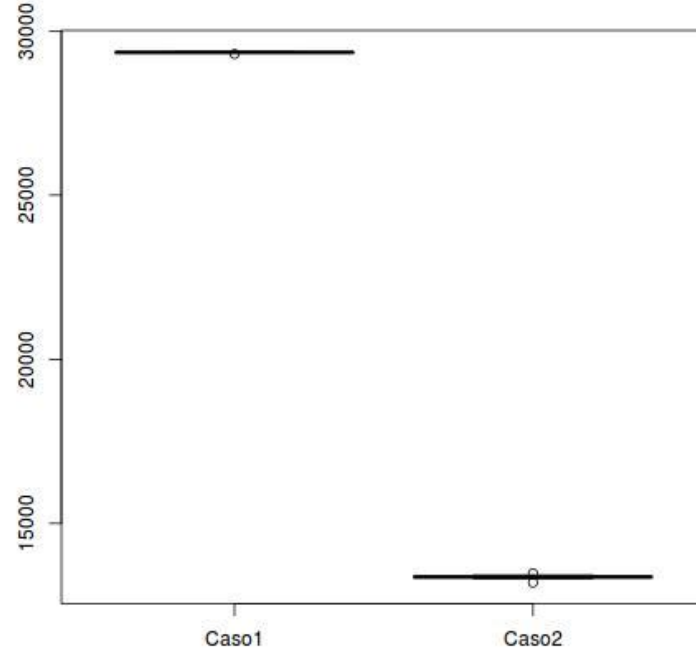
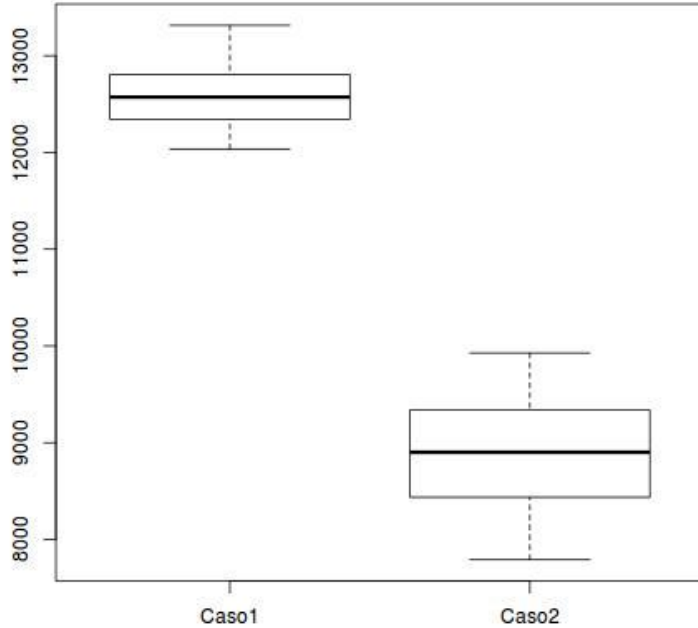
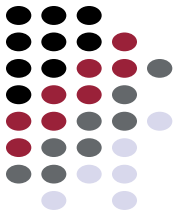


Caso	melhor	dp	media
1	16050	606.783815580619	17749.2333333333
2	10922	1025.70663559268	12426.0333333333

Caso	Tipo	pvalor	pvalorL	pvalorG
1-2	MelhorResultado ~ Caso	2.03889502566081e-28	1	1.0194475128304e-28
1-2	PiorResultado ~ Caso	0.505436978197011	0.747281510901495	0.252718489098505
1-2	Media ~ Caso	3.16471474048073e-60	1	1.58235737024036e-60
1-2	DesvioPadrao ~ Caso	4.75067441788163e-48	2.37533720894082e-48	1
1-2	Tempo ~ Caso	6.41376583146505e-33	3.20688291573252e-33	1

- 1 e 2 -> melhor, media ($1 > 2$); dp, tempo ($1 < 2$); **[2]**

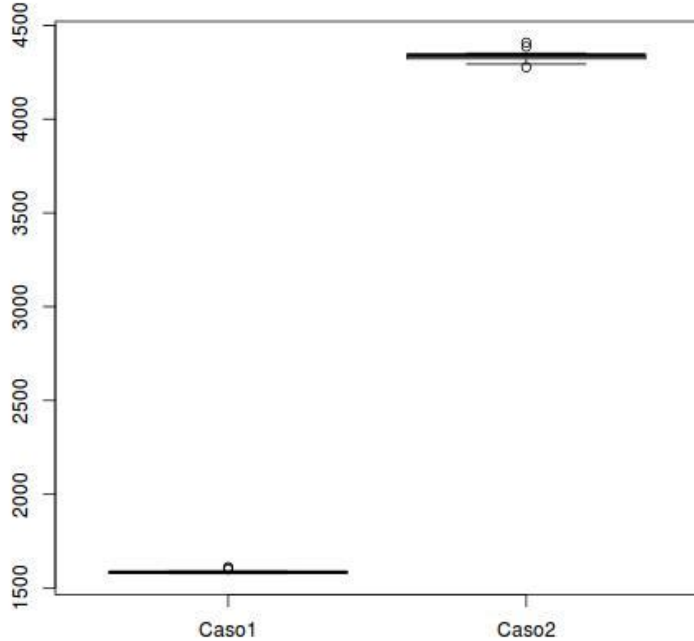
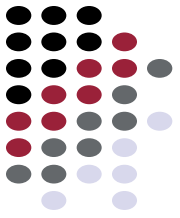
Boxplot [BERLIN 52]



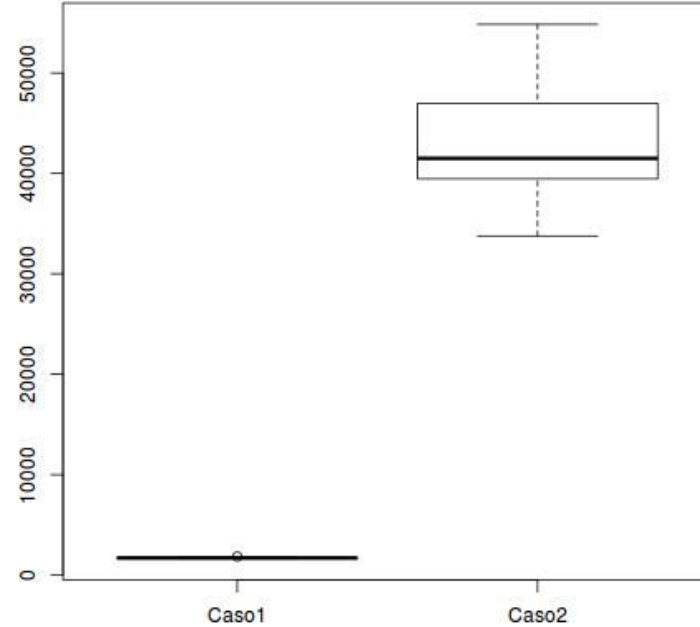
Melhor Resultado ~ Caso

Média ~ Caso

Boxplot [BERLIN 52]



Desvio Padrão ~ Caso



Tempo ~ Caso

Teste T + Resultados [BERLIN 52]



Caso	melhor	dp	media
1	12036.2264517108	329.651926583879	12588.5014243297
2	7790.49461584654	571.940478501201	8921.91627832543

Caso	Tipo	pvalor	pvalorL	pvalorG
1-2	MelhorResultado ~ Caso	2.80927319269287e-32	1	1.40463659634644e-32
1-2	PiorResultado ~ Caso	0.884862563812905	0.557568718093547	0.442431281906453
1-2	Media ~ Caso	6.48208737002271e-86	1	3.24104368501136e-86
1-2	DesvioPadrao ~ Caso	1.19090465416907e-71	5.95452327084537e-72	1
1-2	Tempo ~ Caso	2.91821624834869e-28	1.45910812417435e-28	1

- 1 e 2 -> melhor, media ($1 > 2$); dp, tempo ($1 < 2$); **[2]**

Referências



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