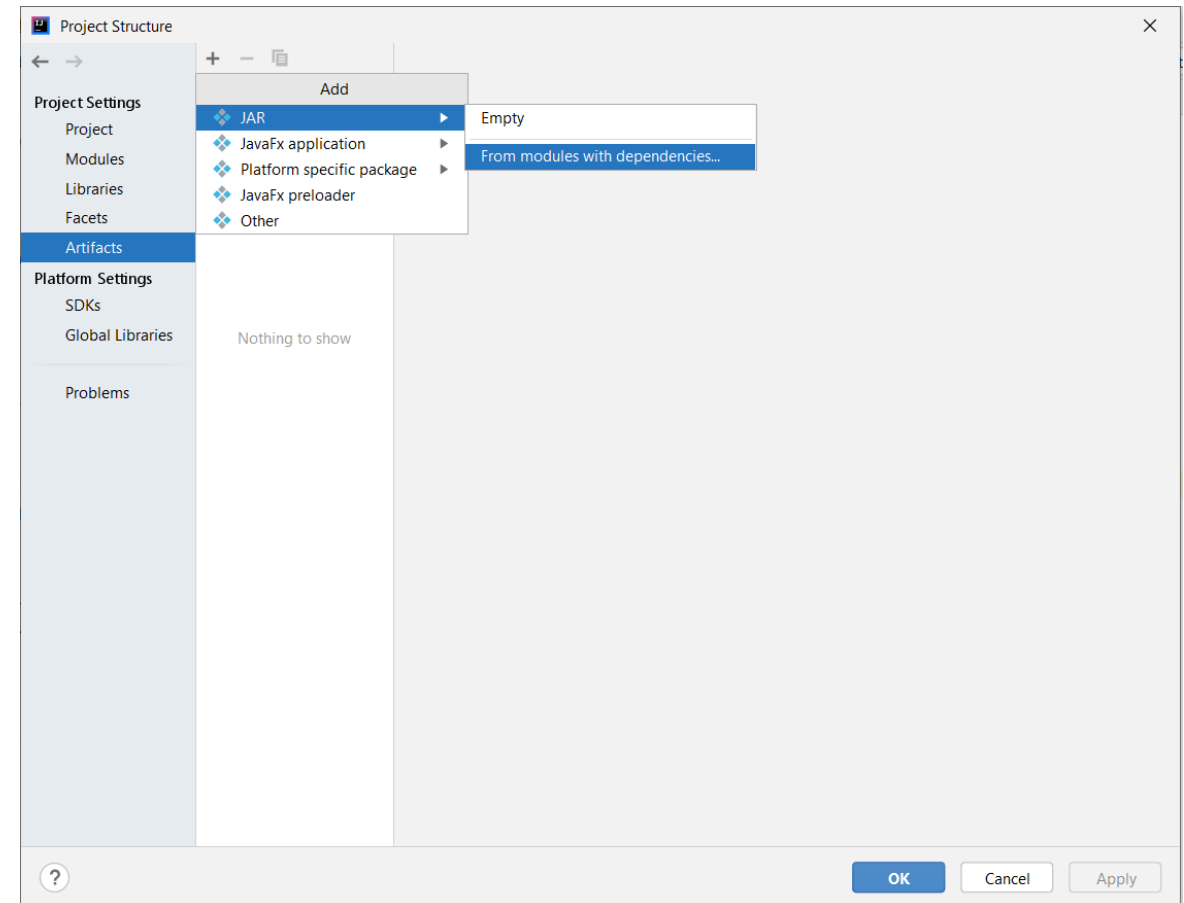


Realização de experiências

Algoritmo genético – projeto *Knapsack*

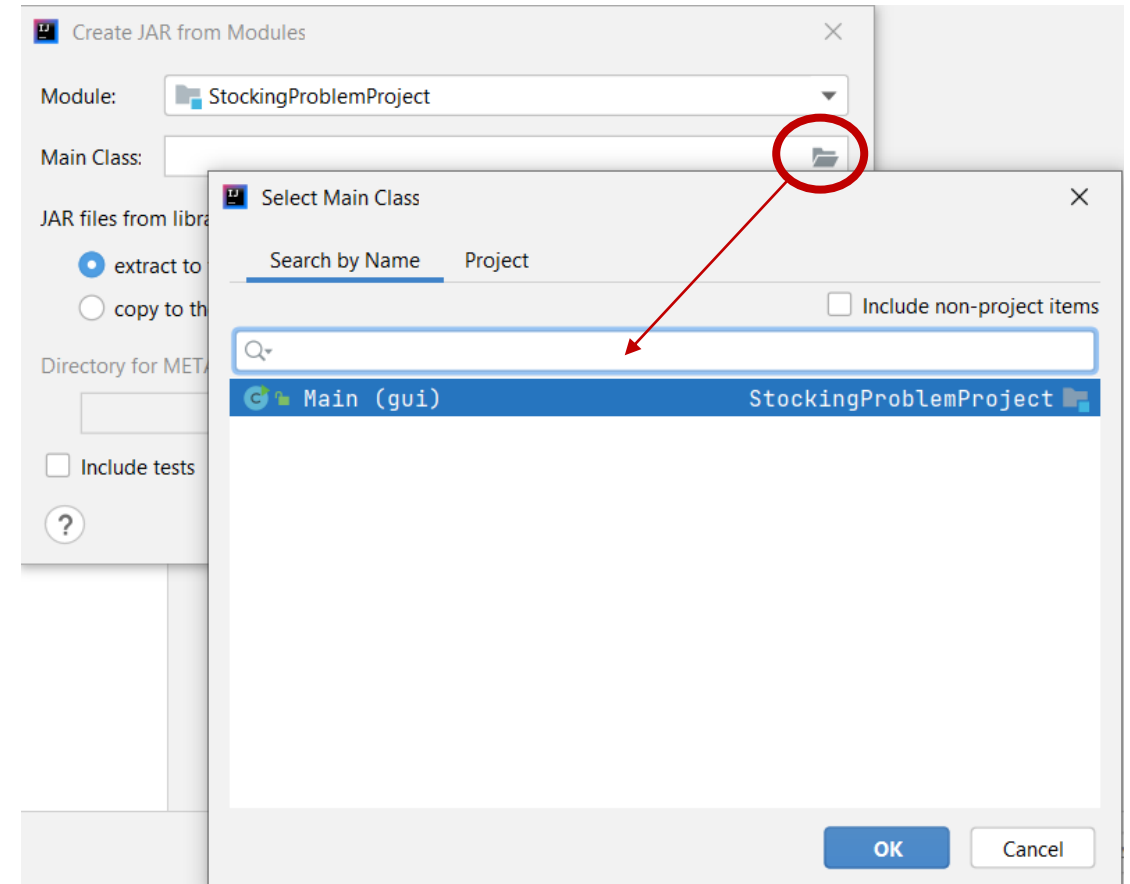
1 – Criação do executável do projeto

- Passos:
 - Abra o projeto no IntelliJ IDEA
 - Vá ao Menu *File* –
Project Structure – *Artifacts* – + – *jar* –
From Modules with dependencies



1 – Criação do executável do (cont.)

- Passos (cont.):
 - Escolha a *Main Class* do Projeto e clique em *ok – Apply*
 - Vá ao Menu *Build – Build Artifacts – Build*
 - Após estes passos, o ficheiro *.jar* pode ser encontrado na pasta *out* do projeto



2 – Criação da estrutura de pastas para as experiências

Testes

Dataset1

TestesGerais

Torneio

População

Recombinação

Mutação

Dataset2

TestesGerais

Torneio

População

Recombinação

Mutação

...

Ficheiros a colocar dentro de cada subpasta:

(o ficheiro .jar é colocado dentro de cada subpasta, para que cada tipo de experiência seja guardada em ficheiros excel diferentes)

DataSet1.txt

StockingProblemProject.jar

DataSet2.txt

StockingProblemProject.jar

...

3 – Realização das experiências

Considere o projeto desenvolvido nas aulas para o problema *Knapsack* (ficheiro executável *GeneticAlgorithm.jar*)

1 – Começar por efetuar análises preliminares a cada dataset, de forma a averiguar os valores dos parâmetros que melhor se adequam à resolução do problema. Nesta análise deverá ser possível identificar o número adequado de gerações (o tamanho da população e o número de gerações dependem da complexidade do problema de otimização e devem ser determinados experimentalmente para cada dataset)

2 – Efetuar experiências (automatizadas) com a variação ligeiramente acima/abaixo dos valores encontrados para determinar a melhor combinação de parâmetros

3 – Realizar experiências (automatizadas) que façam variar os diversos parâmetros do AG, para produzir gráficos que mostrem a influência dos seus valores na média de fitness:

- População vs gerações
- Tamanho do torneio
- Métodos e probabilidades de cruzamento
- Métodos e probabilidades de mutação

3 – Realização das experiências (cont.)

Considerações sobre as experiências automatizadas:

- Utilizar entre 30 a 50 runs por cada combinação de parâmetros
- Guardar os ficheiros de *config* criados para cada experiência e os ficheiros resultantes das experiências (que devem ser entregues juntamente com o código do projeto)
- Após a realização de todas as experiências, eliminar das pastas de teste todos os ficheiros .jar

No relatório do projeto deverá constar:

- A informação da melhor combinação de parâmetros para cada dataset
- O melhor resultado obtido (melhor fitness) e a media de fitness (dos *x runs* realizados)
- Um gráfico com os resultados obtidos para cada combinação de parâmetros para cada dataset (torneio, população vs gerações, recombinações, mutações)
- Todos os resultados apresentados em tabelas e gráficos deverão estar devidamente descritos no texto do relatório

3.1 – Testes gerais a cada Dataset

- Após análises preliminares ao *dataset*, é necessário definir o ficheiro *config* que irá ser utilizado pela aplicação para efetuar múltiplos testes com várias combinações de parâmetros. Este ficheiro deverá ser colocado dentro da pasta *TestesGerais* do *Dataset1*
- Neste exemplo (ficheiro *config* apresentado à direita) irão ser realizados 50 runs para cada combinação de parâmetros, para o *Dataset1*
- No total este exemplo irá permitir a realização de 16200 experiências (324 combinações de parâmetros * 50 runs)
- Os resultados destas experiências são guardados em 3 ficheiros:
 - *statistic_average_fitness*
 - *statistic_best_per_experiment*
 - *statistic_best_per_experiment_fitness*
- O ficheiro que deve ser utilizado para análise da melhor combinação de parâmetros deve ser o ficheiro *statistic_average_fitness*

```
Runs: 50

Population_size: 50, 100, 200

Max_generations: 50

//-----Selection: tournament

Tournament_size: 2, 4, 6, 8

//-----

Recombination: one_cut, two_cuts, uniform

Recombination_probability: 0.6, 0.7, 0.8

//-----Mutation: binary

Mutation_probability: 0.01, 0.025, 0.03

//-----

Probability_of_1s: 0.05

Fitness_type: 0

//-----

Problem_file: ./DataSet1.txt

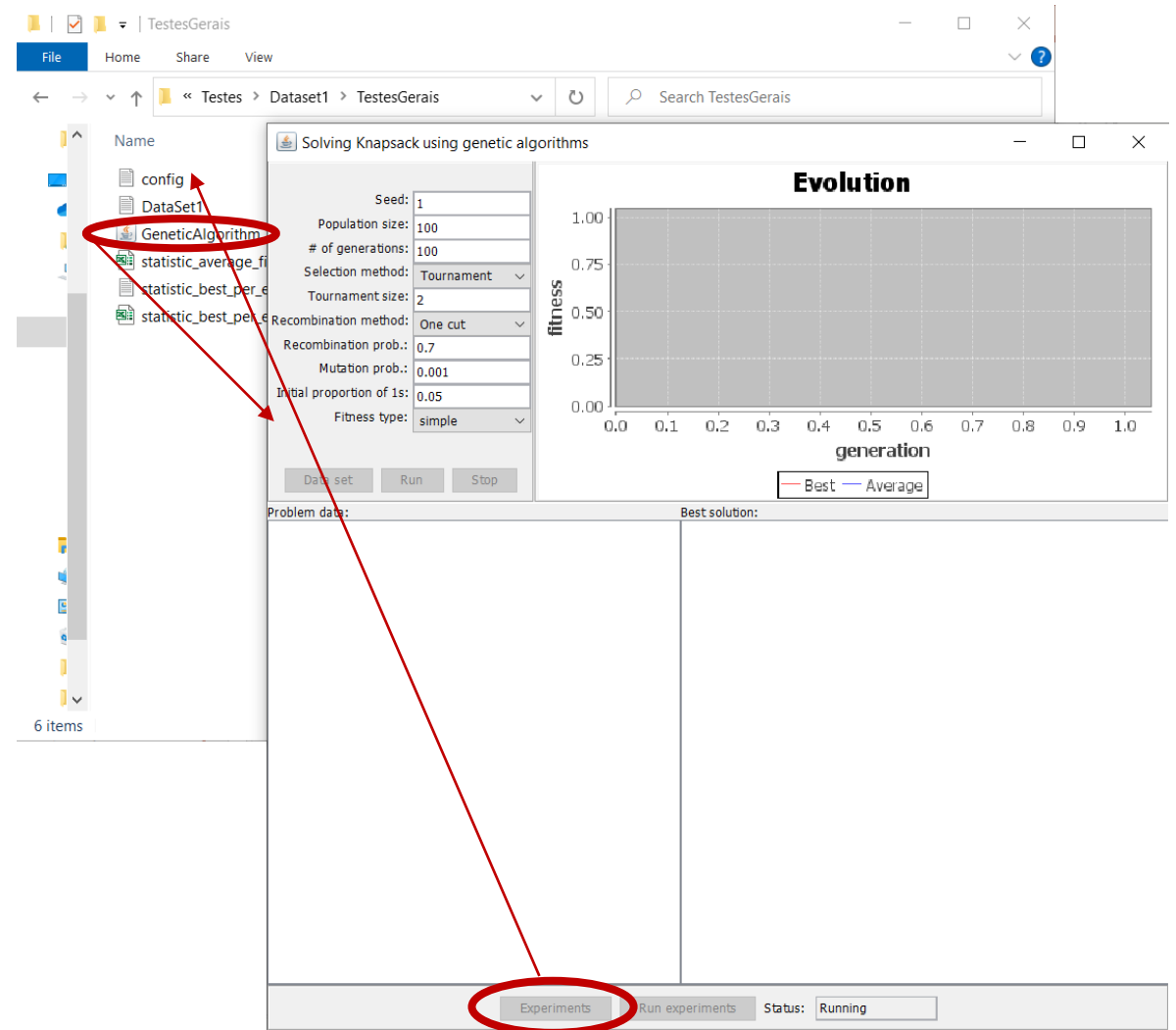
//-----

Statistic: BestIndividual
Statistic: BestAverage
```

3.1 – Testes gerais a cada Dataset (cont.)

- 1 – Clique no ficheiro .jar
- 2 – Clique em *Experiments* e selecione o ficheiro *config*
- 3 – Clique em *Run experiments*

- Será apresentado o *Status: Running* logo após o início das experiências
- Após a conclusão das experiências será apresentado o *Status: Finished*
- Os ficheiros gerados pelas experiências não devem ser abertos enquanto o programa não tiver terminado as experiências (se pretender consultar o seu conteúdo copie os ficheiros para uma pasta temporária)



3.1 – Testes gerais a cada Dataset (cont.)

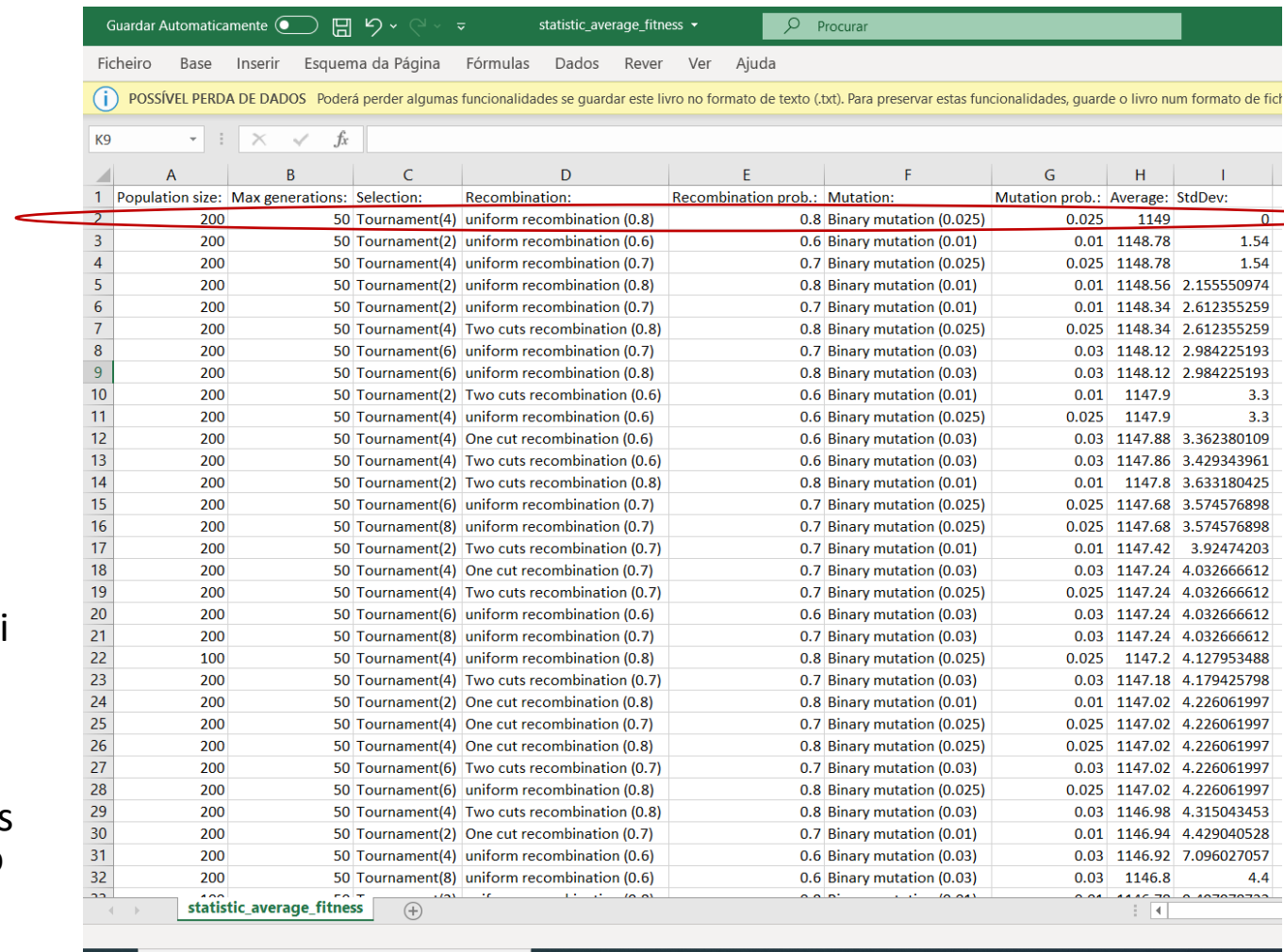
1 – Abra o ficheiro *statistic_average_fitness*

2 – Verifique se os dados da coluna *Average* aparecerem alinhados à direita. Se não estiverem alinhados à direita, é necessário mudar a formatação dos números nas opções do Excel, ou simplesmente selecionar os dados da coluna e trocar ‘.’ por ‘,’

3 – Ordene os dados por ordem decrescente da coluna *Average* (no problema *Knapsack* a melhor combinação de parâmetros é a que tem maior *average fitness*)

Como se pode observar, o melhor resultado obtido foi para a combinação de parâmetros população 200, torneio 4, recombinação uniforme 0.8 e mutação binária 0.025

Todos os 50 runs com esta combinação de parâmetros obtiveram o valor 1149 (uma vez que o desvio padrão é 0)



	A	B	C	D	E	F	G	H	I
1	Population size:	Max generations:	Selection:	Recombination:	Recombination prob.:	Mutation:	Mutation prob.:	Average:	StdDev:
2	200	50	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1149	0
3	200	50	Tournament(2)	uniform recombination (0.6)	0.6	Binary mutation (0.01)	0.01	1148.78	1.54
4	200	50	Tournament(4)	uniform recombination (0.7)	0.7	Binary mutation (0.025)	0.025	1148.78	1.54
5	200	50	Tournament(2)	uniform recombination (0.8)	0.8	Binary mutation (0.01)	0.01	1148.56	2.155550974
6	200	50	Tournament(2)	uniform recombination (0.7)	0.7	Binary mutation (0.01)	0.01	1148.34	2.612355259
7	200	50	Tournament(4)	Two cuts recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1148.34	2.612355259
8	200	50	Tournament(6)	uniform recombination (0.7)	0.7	Binary mutation (0.03)	0.03	1148.12	2.984225193
9	200	50	Tournament(6)	uniform recombination (0.8)	0.8	Binary mutation (0.03)	0.03	1148.12	2.984225193
10	200	50	Tournament(2)	Two cuts recombination (0.6)	0.6	Binary mutation (0.01)	0.01	1147.9	3.3
11	200	50	Tournament(4)	uniform recombination (0.6)	0.6	Binary mutation (0.025)	0.025	1147.9	3.3
12	200	50	Tournament(4)	One cut recombination (0.6)	0.6	Binary mutation (0.03)	0.03	1147.88	3.362380109
13	200	50	Tournament(4)	Two cuts recombination (0.6)	0.6	Binary mutation (0.03)	0.03	1147.86	3.429343961
14	200	50	Tournament(2)	Two cuts recombination (0.8)	0.8	Binary mutation (0.01)	0.01	1147.8	3.633180425
15	200	50	Tournament(6)	uniform recombination (0.7)	0.7	Binary mutation (0.025)	0.025	1147.68	3.574576898
16	200	50	Tournament(8)	uniform recombination (0.7)	0.7	Binary mutation (0.025)	0.025	1147.68	3.574576898
17	200	50	Tournament(2)	Two cuts recombination (0.7)	0.7	Binary mutation (0.01)	0.01	1147.42	3.92474203
18	200	50	Tournament(4)	One cut recombination (0.7)	0.7	Binary mutation (0.03)	0.03	1147.24	4.032666612
19	200	50	Tournament(4)	Two cuts recombination (0.7)	0.7	Binary mutation (0.025)	0.025	1147.24	4.032666612
20	200	50	Tournament(6)	uniform recombination (0.6)	0.6	Binary mutation (0.03)	0.03	1147.24	4.032666612
21	200	50	Tournament(8)	uniform recombination (0.7)	0.7	Binary mutation (0.03)	0.03	1147.24	4.032666612
22	100	50	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1147.2	4.127953488
23	200	50	Tournament(4)	Two cuts recombination (0.7)	0.7	Binary mutation (0.03)	0.03	1147.18	4.179425798
24	200	50	Tournament(2)	One cut recombination (0.8)	0.8	Binary mutation (0.01)	0.01	1147.02	4.226061997
25	200	50	Tournament(4)	One cut recombination (0.7)	0.7	Binary mutation (0.025)	0.025	1147.02	4.226061997
26	200	50	Tournament(4)	One cut recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1147.02	4.226061997
27	200	50	Tournament(6)	Two cuts recombination (0.7)	0.7	Binary mutation (0.03)	0.03	1147.02	4.226061997
28	200	50	Tournament(6)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1147.02	4.226061997
29	200	50	Tournament(4)	Two cuts recombination (0.8)	0.8	Binary mutation (0.03)	0.03	1146.98	4.315043453
30	200	50	Tournament(2)	One cut recombination (0.7)	0.7	Binary mutation (0.01)	0.01	1146.94	4.429040528
31	200	50	Tournament(4)	uniform recombination (0.6)	0.6	Binary mutation (0.03)	0.03	1146.92	7.096027057
32	200	50	Tournament(8)	uniform recombination (0.6)	0.6	Binary mutation (0.03)	0.03	1146.8	4.4

3.2 – Tamanho da população

- O número de gerações está relacionado com o tamanho da população e com o tempo computacional disponível para a execução do algoritmo
- O utilizador deve escolher entre usar uma população pequena com muitas gerações ou uma população maior com menos gerações
- Uma vez que nos testes gerais o melhor resultado foi obtido com uma população de 200 indivíduos e 50 gerações, vamos testar vários tamanhos da população, para confirmar se de facto este é o melhor valor:
 - Pop 50 → 200 gerações
 - Pop 100 → 100 gerações
 - Pop 200 → 50 gerações
 - Pop 400 → 25 gerações
- O ficheiro de *config* apresentado à direita faz as combinações de todos os tipos de população, com todos os números máximos de gerações. Nos restantes parâmetros é colocada a melhor combinação obtida nos testes gerais do *Dataset1*. **Do gráfico a criar deverão constar apenas os resultados correspondentes às 4 combinações acima, sendo que os restantes dados devem ser eliminados**

```
Runs: 50

Population_size: 50, 100, 200, 400

Max_generations: 25, 50, 100, 200

//-----Selection: tournament

Tournament_size: 4

//-----

Recombination: uniform

Recombination_probability: 0.8

//-----Mutation: binary

Mutation_probability: 0.025

//-----

Probability_of_1s: 0.05

Fitness_type: 0

//-----

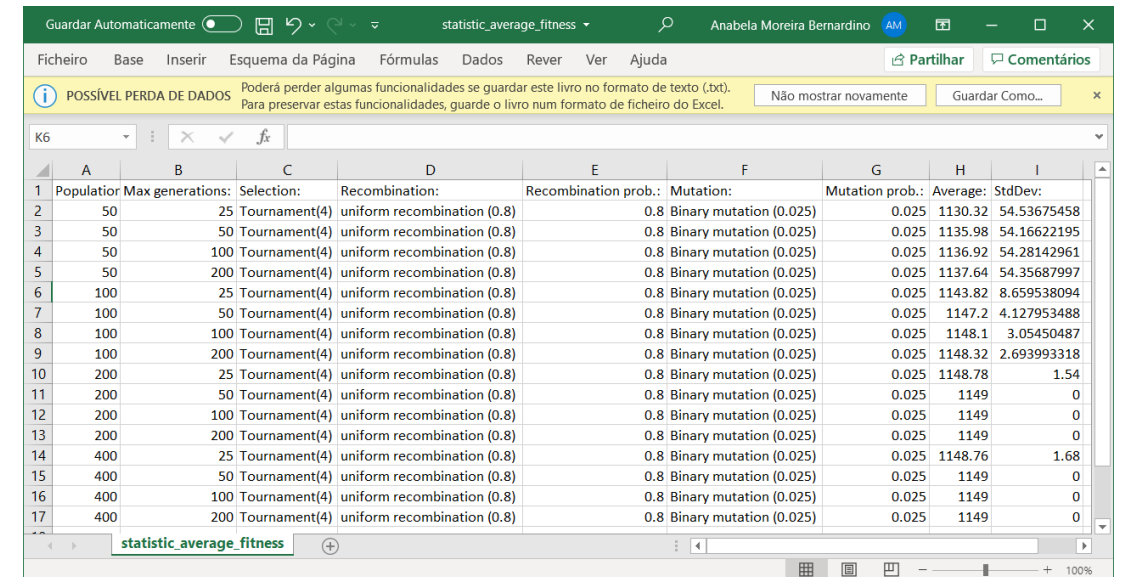
Problem_file: ./DataSet1.txt

//-----

Statistic: BestIndividual
Statistic: BestAverage
```

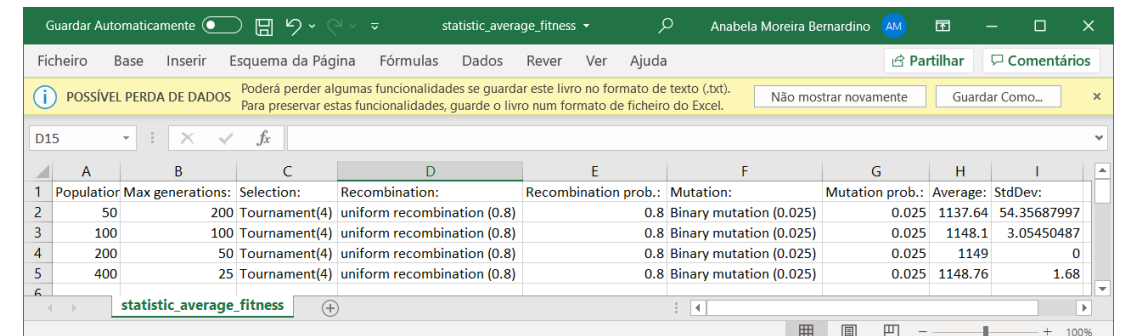
3.2 – Tamanho da população (cont.)

- Após a conclusão dos testes deverá abrir o ficheiro *statistic_average_fitness*



	A	B	C	D	E	F	G	H	I
1	Population	Max generations:	Selection:	Recombination:	Recombination prob.:	Mutation:	Mutation prob.:	Average:	StdDev:
2	50	25	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1130.32	54.53675458
3	50	50	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1135.98	54.16622195
4	50	100	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1136.92	54.28142961
5	50	200	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1137.64	54.35687997
6	100	25	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1143.82	8.659538094
7	100	50	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1147.2	4.127953488
8	100	100	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1148.1	3.05450487
9	100	200	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1148.32	2.693993318
10	200	25	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1148.78	1.54
11	200	50	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1149	0
12	200	100	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1149	0
13	200	200	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1149	0
14	400	25	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1148.76	1.68
15	400	50	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1149	0
16	400	100	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1149	0
17	400	200	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1149	0

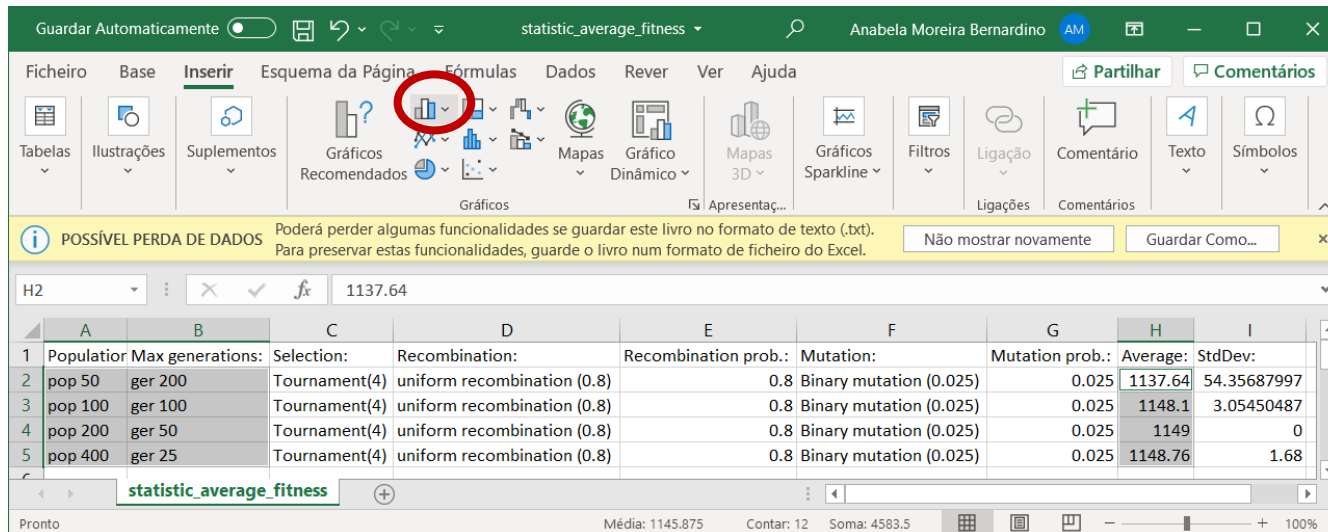
- Elimine as linhas a mais:



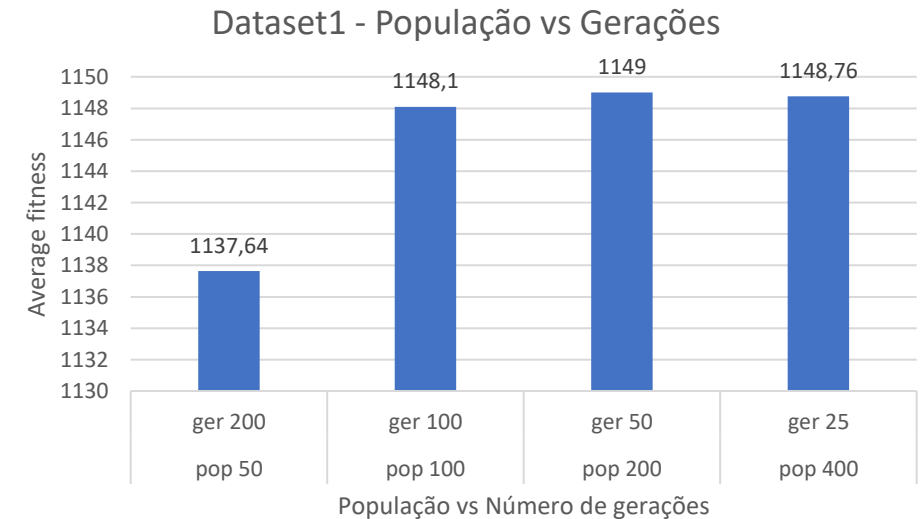
	A	B	C	D	E	F	G	H	I
1	Population	Max generations:	Selection:	Recombination:	Recombination prob.:	Mutation:	Mutation prob.:	Average:	StdDev:
18	50	200	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1137.64	54.35687997
19	100	100	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1148.1	3.05450487
20	200	50	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1149	0
21	400	25	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1148.76	1.68

3.2 – Tamanho da população (cont.)

- Para criar o gráfico de forma automática:
 - Converta as colunas das gerações e da população para texto (no exemplo em baixo foi acrescentado o texto pop na coluna da população e ger na coluna das gerações)
 - Converta a coluna do *average fitness* para número (se os números não aparecerem alinhados à direita, é necessário mudar a formatação dos números nas opções do Excel, ou simplesmente selecionar os dados da coluna e trocar ‘:’ por ‘;’)
 - Selecione as colunas de dados do tamanho da população, do número de gerações e do *average fitness* e crie um gráfico de colunas
 - Adicione títulos nos eixos vertical e horizontal e adicione um título ao gráfico
 - Adicione rótulos de dados às colunas do gráfico



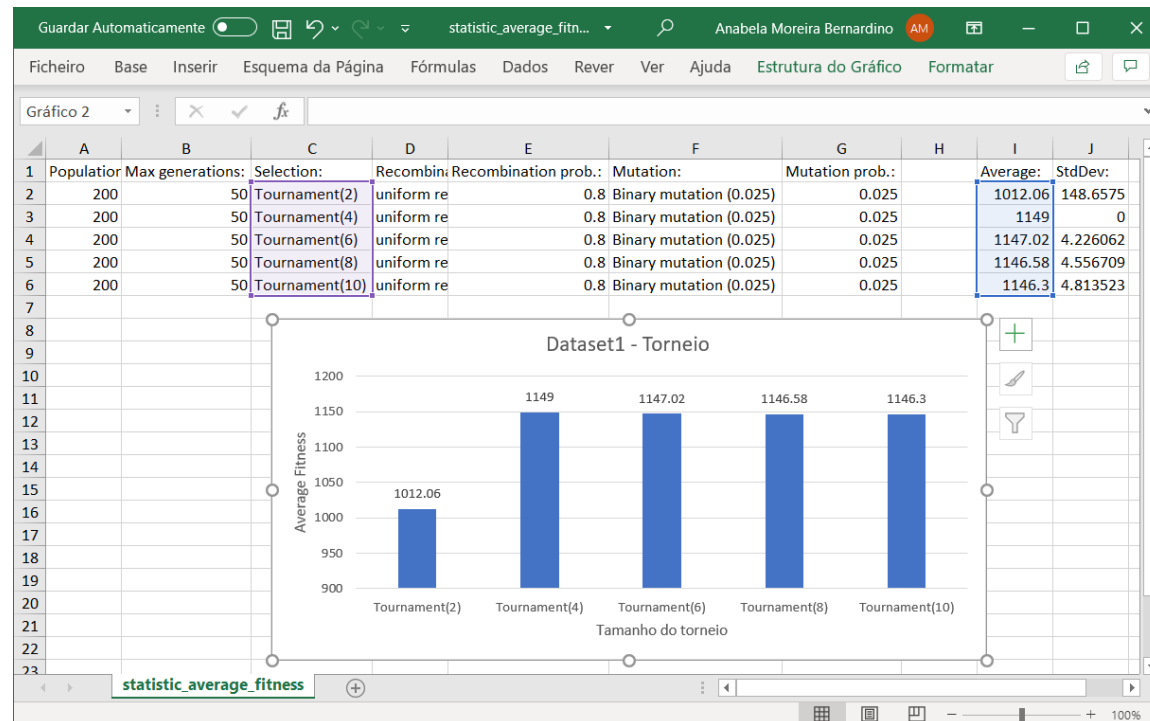
	A	B	C	D	E	F	G	H	I
1	Population	Max generations:	Selection:	Recombination:	Recombination prob.:	Mutation:	Mutation prob.:	Average:	StdDev:
2	pop 50	ger 200	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1137.64	54.35687997
3	pop 100	ger 100	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1148.1	3.05450487
4	pop 200	ger 50	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1149	0
5	pop 400	ger 25	Tournament(4)	uniform recombination (0.8)	0.8	Binary mutation (0.025)	0.025	1148.76	1.68



Como se pode observar pelo gráfico, a população que apresenta os melhores resultados é a população 200

3.3 – Tamanho do torneio

- O ficheiro de config apresentado à direita contém a variação do tamanho do torneio
- Nos restantes parâmetros é colocada a melhor combinação, obtida nos testes gerais
- Após a realização dos testes, deverá ser aberto o ficheiro *statistic_average_fitness* e criado um gráfico idêntico ao apresentado em baixo (verificar antes de criar o gráfico se os dados da coluna *Average* estão alinhados à direita)



Como se pode observar no gráfico, o melhor resultado foi obtido para o torneio 4

```
Runs: 50

Population_size: 200

Max_generations: 50

//-----Selection: tournament

Tournament_size: 2, 4, 6, 8, 10

//-----

Recombination: uniform

Recombination_probability: 0.8

//-----Mutation: binary

Mutation_probability: 0.025

//-----

Probability_of_1s: 0.05

Fitness_type: 0

//-----

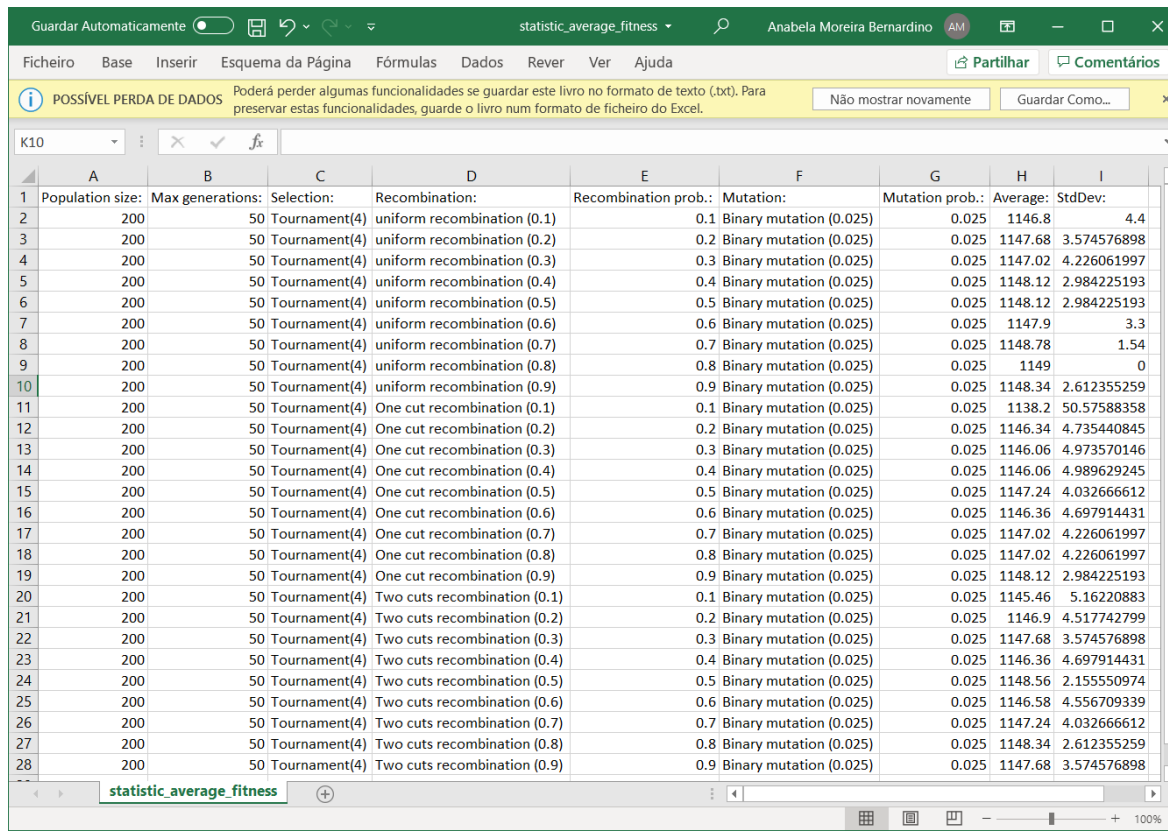
Problem_file: ./DataSet1.txt

//-----

Statistic: BestIndividual
Statistic: BestAverage
```

3.4 – Recombinação

- O ficheiro de config apresentado à direita contém a variação dos vários métodos de recombinação com as várias probabilidades de recombinação
- Nos restantes parâmetros é colocada a melhor combinação, obtida nos testes gerais
- Após a realização dos testes deverá ser aberto o ficheiro *statistic_average_fitness*



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I
1	Population size:	Max generations:	Selection:	Recombination:	Recombination prob.:	Mutation:	Mutation prob.:	Average:	StdDev:
2	200	50	Tournament(4)	uniform recombination (0.1)	0.025	0.1 Binary mutation (0.025)	0.025	1146.8	4.4
3	200	50	Tournament(4)	uniform recombination (0.2)	0.025	0.2 Binary mutation (0.025)	0.025	1147.68	3.574576898
4	200	50	Tournament(4)	uniform recombination (0.3)	0.025	0.3 Binary mutation (0.025)	0.025	1147.02	4.226061997
5	200	50	Tournament(4)	uniform recombination (0.4)	0.025	0.4 Binary mutation (0.025)	0.025	1148.12	2.984225193
6	200	50	Tournament(4)	uniform recombination (0.5)	0.025	0.5 Binary mutation (0.025)	0.025	1148.12	2.984225193
7	200	50	Tournament(4)	uniform recombination (0.6)	0.025	0.6 Binary mutation (0.025)	0.025	1147.9	3.3
8	200	50	Tournament(4)	uniform recombination (0.7)	0.025	0.7 Binary mutation (0.025)	0.025	1148.78	1.54
9	200	50	Tournament(4)	uniform recombination (0.8)	0.025	0.8 Binary mutation (0.025)	0.025	1149	0
10	200	50	Tournament(4)	uniform recombination (0.9)	0.025	0.9 Binary mutation (0.025)	0.025	1148.34	2.612355259
11	200	50	Tournament(4)	One cut recombination (0.1)	0.025	0.1 Binary mutation (0.025)	0.025	1138.2	50.57588358
12	200	50	Tournament(4)	One cut recombination (0.2)	0.025	0.2 Binary mutation (0.025)	0.025	1146.34	4.735440845
13	200	50	Tournament(4)	One cut recombination (0.3)	0.025	0.3 Binary mutation (0.025)	0.025	1146.06	4.973570146
14	200	50	Tournament(4)	One cut recombination (0.4)	0.025	0.4 Binary mutation (0.025)	0.025	1146.06	4.989629245
15	200	50	Tournament(4)	One cut recombination (0.5)	0.025	0.5 Binary mutation (0.025)	0.025	1147.24	4.032666612
16	200	50	Tournament(4)	One cut recombination (0.6)	0.025	0.6 Binary mutation (0.025)	0.025	1146.36	4.697914431
17	200	50	Tournament(4)	One cut recombination (0.7)	0.025	0.7 Binary mutation (0.025)	0.025	1147.02	4.226061997
18	200	50	Tournament(4)	One cut recombination (0.8)	0.025	0.8 Binary mutation (0.025)	0.025	1147.02	4.226061997
19	200	50	Tournament(4)	One cut recombination (0.9)	0.025	0.9 Binary mutation (0.025)	0.025	1148.12	2.984225193
20	200	50	Tournament(4)	Two cuts recombination (0.1)	0.025	0.1 Binary mutation (0.025)	0.025	1145.46	5.16220883
21	200	50	Tournament(4)	Two cuts recombination (0.2)	0.025	0.2 Binary mutation (0.025)	0.025	1146.9	4.517742799
22	200	50	Tournament(4)	Two cuts recombination (0.3)	0.025	0.3 Binary mutation (0.025)	0.025	1147.68	3.574576898
23	200	50	Tournament(4)	Two cuts recombination (0.4)	0.025	0.4 Binary mutation (0.025)	0.025	1146.36	4.697914431
24	200	50	Tournament(4)	Two cuts recombination (0.5)	0.025	0.5 Binary mutation (0.025)	0.025	1148.56	2.155550974
25	200	50	Tournament(4)	Two cuts recombination (0.6)	0.025	0.6 Binary mutation (0.025)	0.025	1146.58	4.556709339
26	200	50	Tournament(4)	Two cuts recombination (0.7)	0.025	0.7 Binary mutation (0.025)	0.025	1147.24	4.032666612
27	200	50	Tournament(4)	Two cuts recombination (0.8)	0.025	0.8 Binary mutation (0.025)	0.025	1148.34	2.612355259
28	200	50	Tournament(4)	Two cuts recombination (0.9)	0.025	0.9 Binary mutation (0.025)	0.025	1147.68	3.574576898

```
Runs: 50

Population_size: 200

Max_generations: 50

//-----Selection: tournament

Tournament_size: 4

//-----

Recombination: uniform, one_cut, two_cuts

Recombination_probability: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9

//-----Mutation: binary

Mutation_probability: 0.025

//-----

Probability_of_1s: 0.05

Fitness_type: 0

//-----

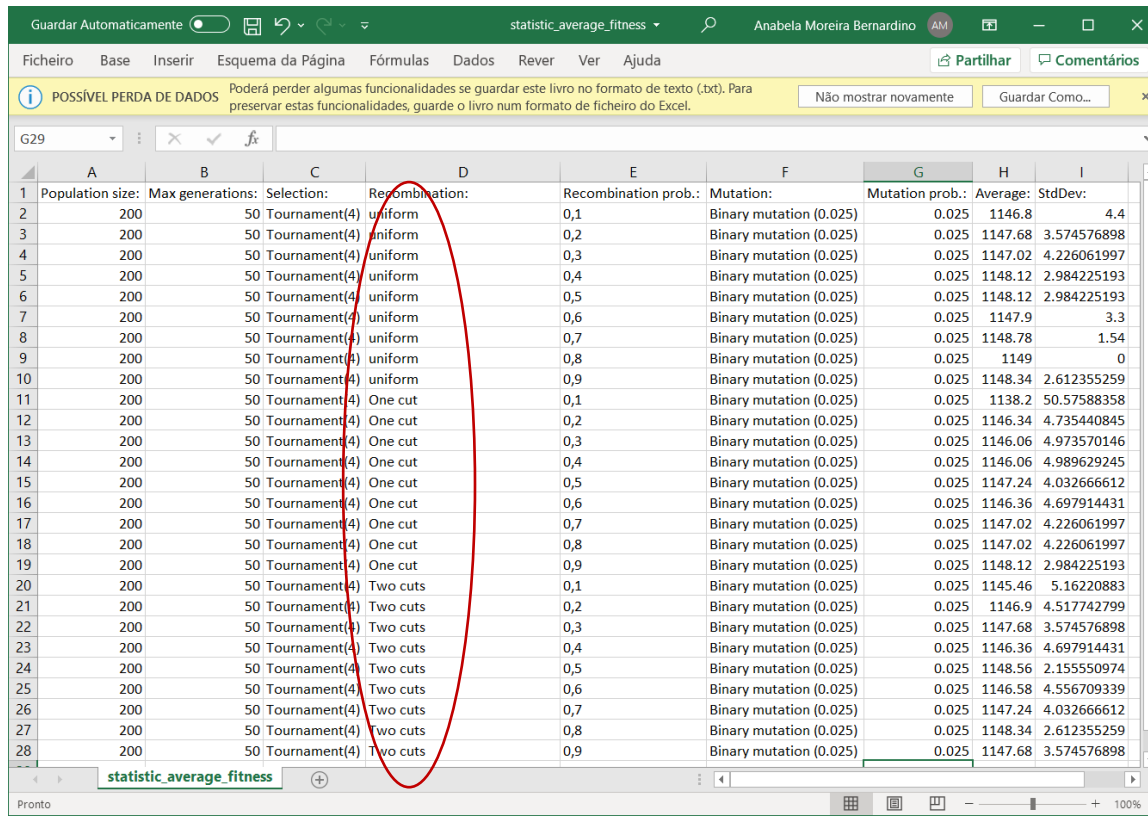
Problem_file: ./DataSet1.txt

//-----

Statistic: BestIndividual
Statistic: BestAverage
```

3.4 – Recombinação (cont.)

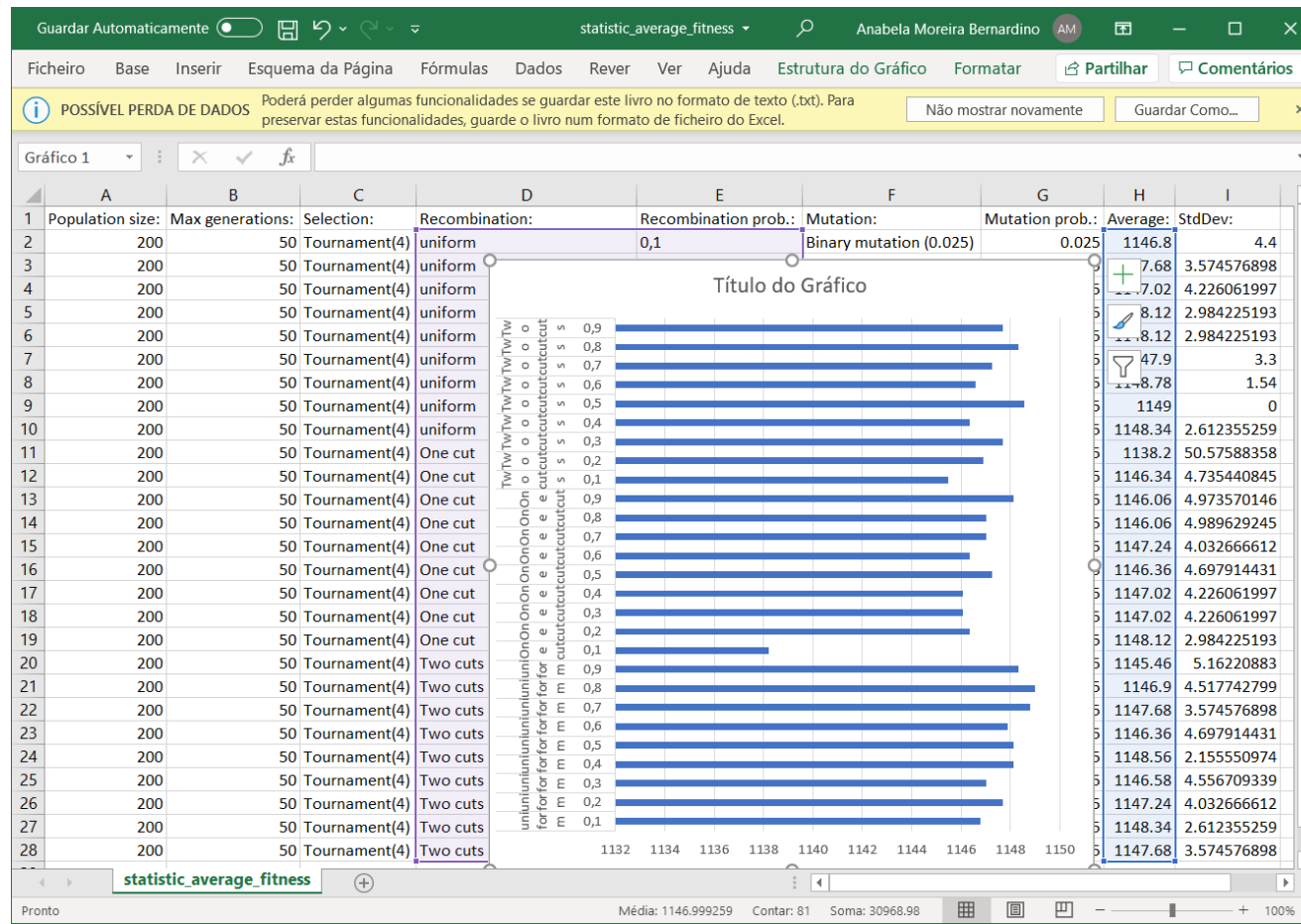
- Verifique se os dados da coluna *Average* estão alinhados à direita
- Converta a coluna *Recombination prob.* para texto, se os dados estiverem alinhados à direita (no exemplo em baixo, foi substituído o “.” por “,” uma vez que os dados desta coluna eram reconhecidos como número)
- Formate a coluna *Recombination* como apresentado:



	A	B	C	D	E	F	G	H	I
1	Population size:	Max generations:	Selection:	Recombination:	Recombination prob.:	Mutation:	Mutation prob.:	Average:	StdDev:
2	200	50	Tournament(4)	uniform	0,1	Binary mutation (0.025)	0.025	1146.8	4.4
3	200	50	Tournament(4)	uniform	0,2	Binary mutation (0.025)	0.025	1147.68	3.574576898
4	200	50	Tournament(4)	uniform	0,3	Binary mutation (0.025)	0.025	1147.02	4.226061997
5	200	50	Tournament(4)	uniform	0,4	Binary mutation (0.025)	0.025	1148.12	2.984225193
6	200	50	Tournament(4)	uniform	0,5	Binary mutation (0.025)	0.025	1148.12	2.984225193
7	200	50	Tournament(4)	uniform	0,6	Binary mutation (0.025)	0.025	1147.9	3.3
8	200	50	Tournament(4)	uniform	0,7	Binary mutation (0.025)	0.025	1148.78	1.54
9	200	50	Tournament(4)	uniform	0,8	Binary mutation (0.025)	0.025	1149	0
10	200	50	Tournament(4)	uniform	0,9	Binary mutation (0.025)	0.025	1148.34	2.612355259
11	200	50	Tournament(4)	One cut	0,1	Binary mutation (0.025)	0.025	1138.2	50.57588358
12	200	50	Tournament(4)	One cut	0,2	Binary mutation (0.025)	0.025	1146.34	4.735440845
13	200	50	Tournament(4)	One cut	0,3	Binary mutation (0.025)	0.025	1146.06	4.973570146
14	200	50	Tournament(4)	One cut	0,4	Binary mutation (0.025)	0.025	1146.06	4.989629245
15	200	50	Tournament(4)	One cut	0,5	Binary mutation (0.025)	0.025	1147.24	4.032666612
16	200	50	Tournament(4)	One cut	0,6	Binary mutation (0.025)	0.025	1146.36	4.697914431
17	200	50	Tournament(4)	One cut	0,7	Binary mutation (0.025)	0.025	1147.02	4.226061997
18	200	50	Tournament(4)	One cut	0,8	Binary mutation (0.025)	0.025	1147.02	4.226061997
19	200	50	Tournament(4)	One cut	0,9	Binary mutation (0.025)	0.025	1148.12	2.984225193
20	200	50	Tournament(4)	Two cuts	0,1	Binary mutation (0.025)	0.025	1145.46	5.16220883
21	200	50	Tournament(4)	Two cuts	0,2	Binary mutation (0.025)	0.025	1146.9	4.517742799
22	200	50	Tournament(4)	Two cuts	0,3	Binary mutation (0.025)	0.025	1147.68	3.574576898
23	200	50	Tournament(4)	Two cuts	0,4	Binary mutation (0.025)	0.025	1146.36	4.697914431
24	200	50	Tournament(4)	Two cuts	0,5	Binary mutation (0.025)	0.025	1148.56	2.155550974
25	200	50	Tournament(4)	Two cuts	0,6	Binary mutation (0.025)	0.025	1146.58	4.556709339
26	200	50	Tournament(4)	Two cuts	0,7	Binary mutation (0.025)	0.025	1147.24	4.032666612
27	200	50	Tournament(4)	Two cuts	0,8	Binary mutation (0.025)	0.025	1148.34	2.612355259
28	200	50	Tournament(4)	Two cuts	0,9	Binary mutation (0.025)	0.025	1147.68	3.574576898

3.4 – Recombinação (cont.)

- Selecione os dados das colunas *Recombination*, *Recombination prob* e *Average* e crie um gráfico de barras
- Aumente o tamanho do gráfico na vertical, para que apareçam todas as probabilidades de 0.1 a 0.9

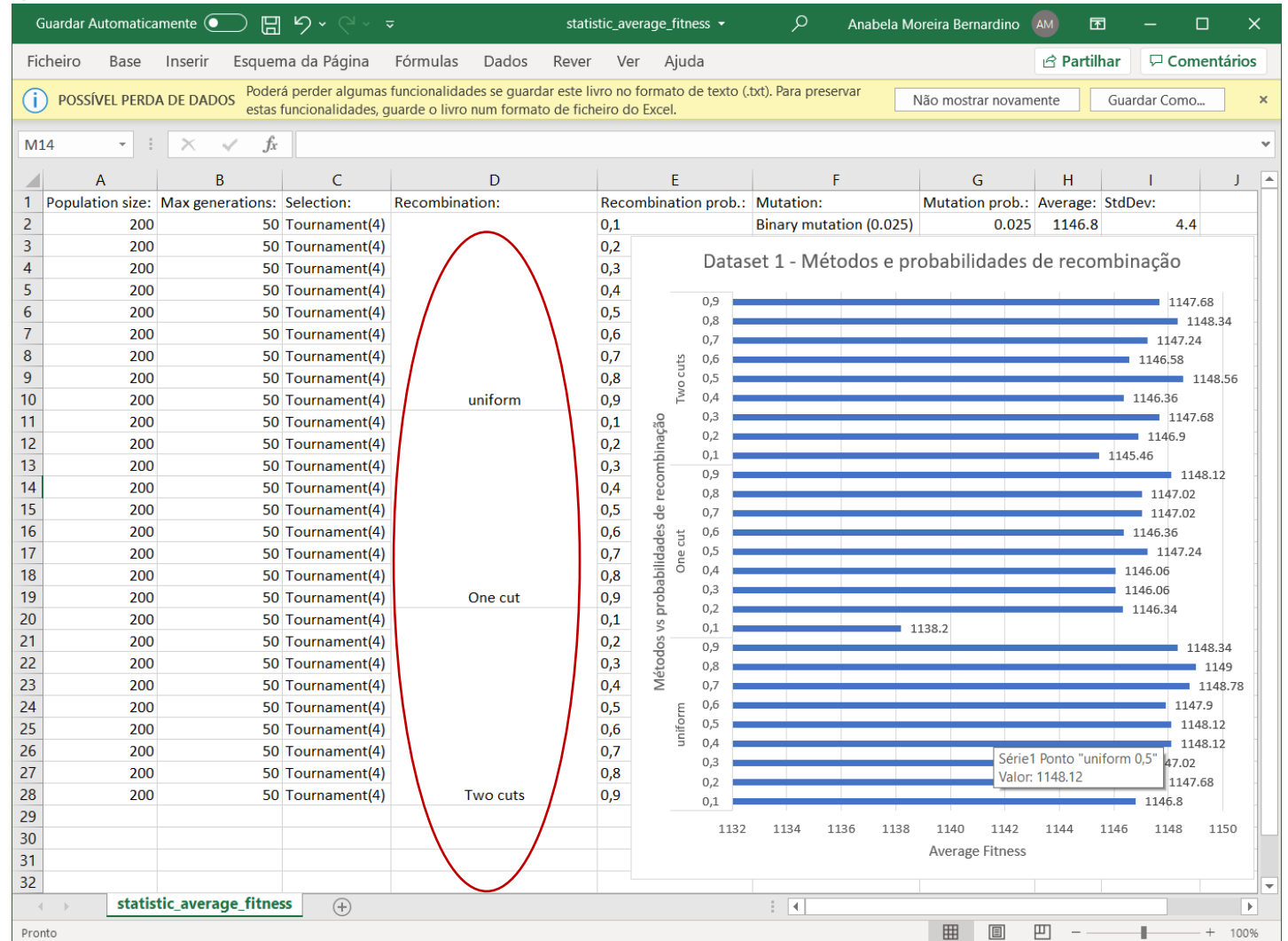


3.4 – Recombinação (cont.)

- Após a criação do gráfico, uma as colunas de cada método de recombinação
- Adicione títulos nos eixos vertical e horizontal e adicione um título ao gráfico
- Adicione rótulos de dados às colunas do gráfico

Observando o gráfico podemos concluir:

- O melhor resultado é obtido para a recombinação uniforme, probabilidade 0.8
- A melhor probabilidade para a recombinação de dois cortes é 0.5 e para a recombinação de um corte é 0.9



3.5 – Mutação

- No projeto *Knapsack* só estava uma mutação implementada, pelo que só era necessário variar as várias probabilidades de mutação
- No projeto a desenvolver na UC de IA deve ser elaborado um *config* idêntico ao criado para a recombinação e deve ser criado um gráfico que mostre a influência das várias probabilidades para cada método de mutação

Boas experiências!