

Terceiro Trabalho

Ana Jorge Almeida
201407964

Para a resolução deste trabalho, existe um ficheiro *deliverygame.py* que resolve a primeira alínea.

Alínea 1

Para resolver a alínea 1, existem dois arrays globais, **acceptance** e **compensation** e foram implementadas sete funções:

- **delivery_place()**, que escolhe de forma aleatória o local onde vai ser entregue a encomenda. Existe uma variável, **place**, que escolhe aleatoriamente os números 1 ou 2. Se o número devolvido pela função for 1, a encomenda será para entrega em casa, se o número for 2, a encomenda será para entrega no locker.
- **id_recipient(acceptance)**, que recebe o valor a ser analisado do array **acceptance** e escolhe uma probabilidade aleatória. Se essa probabilidade for inferior à definida pela **acceptance**, a encomenda é aceite pelo OC.
- **delivery_picked()**, que escolhe uma probabilidade aleatória e, se essa probabilidade for inferior à probabilidade constante de 75%, a encomenda é levantada.
- **PF_total_cost**, recebe o número total de encomendas que serão entregues ao PF e faz a atribuição dos custos.
- **simulate(new_acceptance, new_compensation)**, a função principal do programa, que recebe os valores **acceptance** e **compensation** definidos pelo ciclo no main.
- **run_simulate(new_acceptance, new_compensation)**, que faz um ciclo com as 10000 simulações e retorna os custos (**acc_total_cost**) e os itens máximos nos lockers (**status_lckr**).
- **mean_confidence_interval(data, alpha)**, que recebe o data determinado pela função **run_simulate()** e o **alpha**, que é a diferença entre 1 e a confiança de 0.99.

A função **simulate** está definida da seguinte forma com variáveis globais - as variáveis de **accumulated deliveries** tanto do OC, PF, como do locker, as variáveis **left_home** e **left_lckr** que são as entregas para casa e para o locker que não são levantadas no dia ou rejeitadas pelo OC, os **status** e o custo total acumulado. As variáveis locais - as novas entregas diárias da casa e do locker, o custo diário do PF e OC e as entregas esperadas diariamente, que escolhe um valor random entre 10 e 50. Há também variáveis locais auxiliares para as entregas do OC, do locker e do PF.

Depois de contadas as entregas para casa e as do locker, através de um ciclo for, entramos num ciclo que percorre o número de entregas novas de casa e verifica se o OC aceita entregá-las e todas as entregas que não forem aceites são guardadas na variável **left_home** que será colocada na variável **delivery_next_day** para o PF tratar dessas entregas no dia seguinte. De seguida, há novamente um ciclo for que percorre a variável **lock_for**, que é a soma das entregas novas para o locker e das entregas do locker do dia anterior que não foram levantadas, e há um controlo das entregas levantadas no dia e das que não foram levantadas.

A ideia será então, por exemplo, receber as entregas no dia 1, guardá-las juntamente com as entregas não levantadas ou que não foram aceites pelo OC, nas variáveis **status_lckr** e **status_home**. No dia 2, as entregas serão levantadas ou entregues pelo OC e as que não forem entregues pelo OC serão pelo PF, daí achar necessária a criação de variáveis locais auxiliares para garantir que estes ajustes serão feitos no dia depois da recepção das novas encomendas.

O programa demora algum tempo a correr, mas segue um exemplo do output:

```
For acceptance 0.01 with compensation 0.0...
The expected total cost is (19.1835485798, 19.4468514202)
The expected maximum number of items stored in the locker is (14.9369900079, 15.2748099921)
For acceptance 0.25 with compensation 0.5...
The expected total cost is (453.123430714, 469.248819286)
The expected maximum number of items stored in the locker is (14.9543736571, 15.1922263429)
For acceptance 0.5 with compensation 1.0...
The expected total cost is (891.639418899, 913.123481101)
The expected maximum number of items stored in the locker is (15.018213869, 15.2125194643)
For acceptance 0.6 with compensation 1.5...
The expected total cost is (1331.57148683, 1357.07258817)
The expected maximum number of items stored in the locker is (15.0306502061, 15.1985497939)
For acceptance 0.75 with compensation 1.8...
The expected total cost is (1702.10591559, 1728.75750441)
The expected maximum number of items stored in the locker is (15.0561384229, 15.2063815771)
```

Intervalos de confiança observados para cada cenário.

São feitas 10000 simulações para cada valor de acceptance.

Alínea 2

Por observação dos resultados da alínea anterior, melhor cenário para os OCs seria uma compensação de 0.5.