

Parte I – Usando o ficheiro “Projeto_Parte I”

A *FoodForAll* é uma associação humanitária que se destina à distribuição de alimentos aos habitantes de $m = 20$ aldeias isoladas. Para tal, a associação criou $n = 9$ armazéns, em regiões próximas às aldeias, onde estão armazenados os alimentos. Os armazéns contêm comida suficiente para alimentar todas as aldeias e são abastecidos regularmente através de camiões, uma vez que existem bons acessos até eles. Em contrapartida, a única forma de chegar às aldeias a partir dos armazéns é através de pequenos caminhos de terra batida. Por essa razão, o transporte de alimentos dos armazéns para as aldeias é feito apenas usando motas. Cada mota tem um pequeno reservatório que permite levar alimentos para 10 pessoas. Significa isto que, por exemplo, para distribuir alimento a 40 pessoas de uma aldeia são necessárias 4 viagens de mota de ida e volta entre um armazém e a aldeia.

A *FoodForAll* conta neste momento com 9 pessoas para fazer as entregas diárias dos alimentos. Essas pessoas, sendo voluntárias, têm o seu emprego próprio e, por isso, não estão disponíveis todos os dias para ajudar. Assim sendo, em cada dia, apenas p das 9 pessoas existentes estão disponíveis para a distribuição. Isto significa que, em cada dia, as aldeias podem ser abastecidas a partir de apenas p dos n armazéns existentes.

Além da *FoodForAll*, existe uma outra associação que faz a distribuição de alimentos nas aldeias: a *GiveABitOfUs*. Esta associação faz a distribuição no início de cada dia, enquanto que a *FoodForAll* operada da parte da tarde. Após a *GiveABitOfUs* terminar a distribuição no período da manhã a algumas pessoas de algumas aldeias, é entregue à *FoodForAll* uma tabela com o número de pessoas que ficaram por alimentar em cada aldeia. Essa tabela é depois modificada e convertida numa tabela de custos, conforme exemplificado abaixo. Cada armazém é designado por $C_i, i = 1, \dots, n$ e cada aldeia é designada por $L_j, j = 1, \dots, m$. Os valores da segunda coluna indicam o custo fixo de uso diário de cada armazém, que tem em conta os custos de deslocação da casa dos voluntários que fazem a distribuição até ao armazém e os custos de funcionamento do armazém. Os valores nas restantes células indicam o custo de alimentar todas as pessoas ainda não alimentadas da aldeia L_j a partir do armazém C_i .

Custos	C.F	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16	L17	L18	L19	L20
C1	13	8	9	8	10	8	1	6	2	3	2	7	3	6	0	7	0	1	8	3	8
C2	11	3	1	9	3	6	2	4	8	9	2	6	2	3	9	1	4	3	0	7	5
C3	16	7	8	4	4	8	1	3	7	1	6	4	4	7	1	10	3	10	5	4	8
C4	18	7	3	4	9	4	4	6	7	5	0	8	1	8	8	3	1	4	4	6	4
C5	12	1	0	1	6	3	6	6	6	8	8	4	7	8	5	10	2	6	6	9	5
C6	15	5	10	2	10	9	5	2	1	3	8	9	4	5	4	1	2	1	5	3	4
C7	13	5	5	4	2	3	7	7	4	6	0	10	5	3	9	1	4	8	4	9	0
C8	14	7	1	4	9	9	2	2	7	3	8	1	8	7	2	10	9	3	9	9	7
C9	11	8	0	5	2	8	5	9	5	9	9	1	5	5	4	5	9	7	8	1	9

Uma vez que, em cada dia, as pessoas disponíveis na associação e o número de pessoas que precisam de ser alimentadas nas aldeias variam, os valores apresentados na tabela acima mudam diariamente.

O objetivo da *FoodForAll* é desenvolver uma plataforma, a ser executada diariamente, que, tendo em conta uma tabela de custos como a apresentada acima e o número p de pessoas disponíveis para a entrega, determine quais dos armazéns devem ser abertos nesse dia e quais as aldeias servidas por cada um desses armazém de modo que todas as pessoas por alimentar sejam alimentadas e que o custo total da distribuição de alimentos (custo de uso dos armazéns + custo de entrega) seja o menor possível. É ainda necessário ter em conta que todas as pessoas de uma determinada aldeia têm que ser servidas pelo mesmo armazém.

O que a associação já fez:

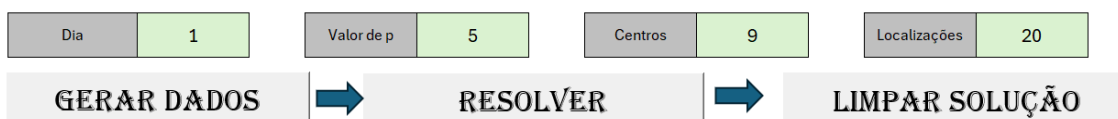
A *FoodForAll* começou por formular o problema de distribuição de alimentos (de forma geral) para um único dia como um problema de programação linear (PL).

Apesar da construção do modelo não ser alvo de avaliação no projeto, aconselha-se que tentem fazer por vocês a formulação em PL do problema, definindo as variáveis de decisão do problema e considerando d_i o custo de uso do armazém i e c_{ij} o custo total de servir a aldeia j a partir do armazém i , $i = 1, \dots, n, j = 1, \dots, m$. Fazer e perceber bem a formulação será muito útil para as fases seguintes do projeto.

Depois, a *FoodForAll* usou o Solver do Excel para resolver o problema e obter o plano ótimo de distribuição de alimentos para cada dia. Assim sendo, ao meio-dia de cada dia, a associação atualiza a tabela de custos no Excel, após receber a informação da *GiveABitOfUs*, e chama o Solver para determinar o plano ótimo de distribuição de alimentos tendo em conta os p voluntários disponíveis nesse dia. Todo o código associado a esta etapa foi criado no módulo “Nao_Mexer” do ficheiro Excel “Projeto_Parte I” e não pode ser modificado.

Devem brincar com o ficheiro Excel que receberam alterando o dia e o número de voluntários disponíveis. De seguida, clicam em “gerar dados” para preencher automaticamente a tabela de custos. Carregam de seguida em “Resolver” para obter a solução ótima do problema. Por fim, carregam em “Limpar Solução” para visualizar a solução de forma mais clara. Nunca carreguem no botão “Comparar Soluções”.

Nota: Para que o solver funcione, é necessário ativá-lo. Para tal, abram o editor VBA e vão a tools/referências/solver.



O que se pretende fazer:

No futuro, a *FoodForAll* pretende aumentar o número de aldeias que apoia e, eventualmente, abrir novos armazéns. Isto fará com que o número de variáveis de decisão do problema aumente e, conseqüentemente, que deixe de ser possível utilizar o Solver para obter o plano ótimo de distribuição diário. Assim sendo, a *FoodForAll* pretende desenvolver um algoritmo (heurística) que sirva como alternativa ao modelo, isto é, que permita determinar um plano de distribuição cujo custo total seja o mais próximo possível do custo mínimo que seria obtido se se pudesse utilizar o Solver para resolver o problema de forma ótima.

Uma vez que a *FoodForAll* consegue usar o Solver para obter soluções ótimas para o cenário atual (com 20 aldeias e, no máximo, 9 armazéns), foi decidido que a heurística seria primeiramente aplicada ao cenário atual. Desta forma, é possível ter uma ideia clara sobre a qualidade das soluções que a heurística obtém, através do cálculo do gap relativamente ao valor ótimo obtido pelo Solver.

No módulo “Heurísticas” devem implementar uma heurísticas à vossa escolha que permita obter um *bom* plano de distribuição diário. Devem fazer isso aqui dentro:

```
Function Heuristical() As Long
    Call GerarNumeros 'Isto preenche o valor de todos os parâmetros

    Heuristical = ...
End Function
```

Notem que os parâmetros do problema ficam automaticamente preenchidos através da primeira instrução, e são os que se encontram abaixo:

```
Public d() As Integer ' Custo de uso dos armazens
Public c() As Integer ' Custo aldeia-armazem
Public n As Integer ' #armazens
Public m As Integer ' #aldeias
Public p As Integer ' #voluntarios
```

Usando estes parâmetros, a heurística deve mostrar uma mensagem de texto que indique quais os armazéns de onde será feita a distribuição naquele dia, quais as aldeias servidas por cada um deles e qual o custo total da distribuição. O retorno da função é o valor do plano de distribuição. Não podem usar qualquer mecanismo aleatório na implementação desta heurística. Devem implementar uma função que, dado o conjunto de armazéns a abrir num dia, devolva o menor custo de afetação das aldeias a esses armazéns.

Antes do código relativo à implementação da heurística, devem deixar descrito em comentário qual a ideia geral da heurística, usando linguagem natural e clara.

O Xavier, um dos voluntário mais antigos da *FoodForAll*, tem alguns conhecimentos de IO e sugeriu que fosse implementada uma segunda heurística com mecanismos aleatórios. A ideia da heurística que ele sugeriu é a seguinte. Primeiro deve escolher-se aleatoriamente quais os p armazens que irão operar naquele dia. De seguida, é feita a afetação das aldeias aos armazens selecionados de forma a minimizar os custos de afetação. Por fim, altera-se aleatoriamente o conjunto dos armazens selecionados para operar e o processo é repetido 100 vezes. Em cada iteração, é calculado o custo total da solução obtida e, no final do processo, deve ser indicada qual a melhor solução encontrada durante o processo e qual o seu custo total de distribuição.

Tarefa 4
Até 2/11
6%

No módulo “Heurísticas” devem implementar a heurística proposta pelo Xavier. Devem fazer isso aqui dentro:

```
Function Heuristica2() As Long
    Call GerarNumeros 'Isto preenche o valor de todos os parâmetros

    Heuristica2 = ...
End Function
```

A heurística deve mostrar uma mensagem de texto que indique quais os armazéns de onde será feita a distribuição naquele dia, quais as aldeias servidas por cada um deles e qual o custo total da distribuição. Além disso, deve ainda indicar quantas vezes é que a melhor solução foi atualizada no decorrer da heurística. O retorno da função é o valor do melhor plano de distribuição encontrado.

Será que o algoritmo do Xavier é melhor do que o vosso?

Corram o Solver e as duas heurísticas usando a sequência dos quatro botões apresentada na folha inicial do Excel para um conjunto de 10 dias e verifiquem qual das heurísticas parece funcionar melhor.

Parte II – Usando o ficheiro “Projeto_Parte II”

Tarefa 5
Até 18/11
17%

A *FoodForAll* pretende modernizar a sua plataforma através da implementação de uma interface gráfica. A plataforma deve ser acedida através de um botão “Iniciar Aplicação” contido na folha Excel. Ao carregar no botão, deve ser pedida ao utilizador uma password através de um Userform. A password correta é “FFA”. Caso a password introduzida pelo utilizador não esteja correta, deve ser pedido sucessivamente ao utilizador que insira a password correta, até um máximo de três tentativas. Caso a password esteja correta, a aplicação deve chamar um procedimento inicial (que terão que implementar) que faça a leitura dos dados contidos na folha inicial e que abra o menu inicial da plataforma. Esse menu deverá conter:

- Um *SpinButton* combinado com *TextBox* onde o utilizador indica qual o número p de armazéns que vão funcionar nesse dia.
- Nove *CheckBoxs*, uma para cada armazén.
- Um botão “Determinar Solução” que, após verificar se o valor de p está de acordo com os valores das *CheckBoxs*, indique quais as aldeias afetas a cada armazém, qual o custo total de servir cada aldeia e qual o custo total do plano de distribuição. Toda esta informação deve aparecer dentro de uma label ou textbox (integrada no menu inicial) que apenas deverá ficar visível quando se carregar no botão referido anteriormente. A solução obtida (centros abertos+aldeias associadas+custo total) deverá ser escrita numa folha Excel com o nome “Soluções”.
- Uma Imagem.
- Um botão “Alterar plano de distribuição” que apenas fica visível após ter sido carregada uma primeira vez no botão “Determinar Solução” e que, quando carregado, abra um novo menu “Alterações”. Este novo menu contém uma textbox/label com o plano de distribuição atual que foi apresentado no menu principal, e uma outra textbox com o plano alterado. Além disso, deve existir uma outra textbox/label que indique qual o acréscimo ou redução no valor total da distribuição do novo plano de distribuição face ao antigo. As alterações à solução atual são feitas através de três botões: um que permita abrir um novo centro e reafetar as aldeias aos centros, outro que permita fechar um dos centros abertos e reafetar as aldeias, e outro que permita afetar uma aldeia específica a um centro específico. A informação de quais os centros a abrir/fechar e de quais as aldeias a reafetar deve ser pedida ao utilizador através de userforms. Para o terceiro botão, devem usar os *OptionButton*. O menu alterações terá ainda um botão “Voltar” que permitirá voltar ao menu principal. Ao carregar nesse botão, o programa deve perguntar ao utilizador se quer gravar a solução alterada ou não. Em caso afirmativo, a solução alterada (centros abertos+aldeias associadas+custo total) deverá ser escrita na folha Excel com o nome “Soluções”.
- Um botão para sair da aplicação.

Neste projeto tudo será tido em conta na avaliação: o correto funcionamento de todo o programa (obviamente...), a parte estética e funcional da interface, o modo de interação com o utilizador, simplicidade e clareza do código (incluindo aspetos como: indentações, nomes de variáveis/funções/procedimentos, comentários, estruturas de dados usadas etc.).