

Licenciatura em Engenharia de Sistemas Informáticos

Pesquisa do Melhor Horário

para Funcionários por Turnos

No âmbito da UC de Inteligência Artificial

André Cardoso & Leonel Fernandes
18848 & 18850

Barcelos, 18 de dezembro de 2021

Contents

1	Formulação do Problema	3
2	Pesquisa	5
3	Obstáculos Encontrados	7
4	Resultados Obtidos	8

1 Formulação do Problema

Descrição

O problema tem como objetivo encontrar a melhor distribuição de funcionários de um supermercado por um horário definido em turnos de manhã e de tarde, tendo em conta o número total de funcionários, assim como as previsões de clientes que visitarão o supermercado.

Existem algumas regras que o horário deve cumprir, como por exemplo:

- Não podem existir menos de 3 funcionários a trabalhar por turno.
- Um funcionário só pode trabalhar 5 turnos.
- Um funcionário não pode trabalhar em dois turnos no mesmo dia.
- Todos os funcionários têm de ser colocados no horário.

Assim sendo, se todas as regras acima definidas forem cumpridas, resta apenas que o posicionamento de funcionários pelo horário seja coerente mediante o número de clientes previstos para cada dia.

Estados

Como estado inicial do problema temos um horário vazio, horário este que é composto pelo dia da semana e com os turnos da manhã e de tarde para cada dia, ao total temos 14 turnos diferentes. Este estado é definido por uma lista de dias (de segunda a domingo), para cada um destes dias estarão associados turnos (manhã e tarde), estes turnos serão preenchidos com uma lista funcionários que trabalham no mesmo, como se trata do estado inicial as listas de funcionários estarão vazias.

A cada novo estado posicionaríamos um funcionário nas listas de turnos escolhidos, ou seja, 5 turnos seriam preenchidos, o funcionário seria definido pelo seu número identificativo. Desta forma as listas seriam preenchidas com cada vez mais números de funcionários.

No final da pesquisa ficaríamos com todos os funcionários inseridos em turnos pelo horário e não haveriam turnos vazios.

O formato utilizado para guardar os estados seria um dicionário, e teria uma formatação igual à seguinte:

```
{
  "Segunda ": {
    "manha ": [], "tarde ": []
  },
  "Terca ": {
    "manha ": [], "tarde ": []
  },
  "Quarta ": {
    "manha ": [], "tarde ": []
  },
  "Quinta ": {
    "manha ": [], "tarde ": []
  },
  "Sexta ": {
    "manha ": [], "tarde ": []
  },
  "Sabado ": {
    "manha ": [], "tarde ": []
  },
  "Domingo ": {
    "manha ": [], "tarde ": []
  }
}
```

Onde dentro de cada "manha" ou "tarde" estaria a lista de funcionário daquele dia.

2 Pesquisa

Variáveis importantes

- Margem (margin) - Valor mínimo de funcionários por turno, calculado como $1/4$ do número de funcionários disponíveis.
- Previsões (previsoes) - Dicionário com entradas de previsões de clientes para cada dia por cada turno.
- QtdFunc (len(funcs)) - Número de funcionários disponível.
- ClientesMaximo (clientesMaximo) - Número máximo de clientes da previsão de turnos.

Passagem por estados

De modo a efetuar a pesquisa de estados, como cada estado se refere ao posicionamento de um funcionário em 5 posições de turnos, a abordagem escolhida foi verificar todos os estados imediatamente em seguida ao estado atual, ou seja, todas as combinações possíveis de um funcionário pela semana, para lhes ser atribuída uma pontuação. Depois de atribuída é escolhido o estado com maior pontuação para serem verificados os seus estados seguintes e o processo repete-se até chegar ao último nível da árvore, neste caso quando já posicionamos o último funcionário.

Eliminação de estados

De modo a regular a transição de estados de modo a alcançar a solução mais próxima ao ótimo, são eliminados os estados que se desviem de algum modo do caminho preferencial, um exemplo disso é a distribuição homogênea dos funcionários.

Naturalmente que não é rentável ter o mesmo número de funcionários todos os dias sem distinção, no entanto, todos os dias são necessários para o funcionamento básico um número mínimo de trabalhadores. Como tal, é

avaliada se há uma diferença superior a 1 entre cada turno e o turno com mais funcionários. Neste caso, são aceites apenas os estados em que no mesmo dia haja pelo menos um turno que não esteja vazio com pelo menos x funcionários inseridos em que x é igual à margem calculada.

Pontuação

Foram utilizados vários fatores para determinar a pontuação de um estado, de modo conseguirmos obter o melhor estado final tendo em conta o problema, como por exemplo:

- É calculado um número ideal de funcionários mediante o número previsto de clientes para o dia e atribuído um score tendo em conta a proximidade do número de funcionários ao número ideal.

– O cálculo do número ideal é feito com a seguinte expressão:

$$(\text{previsoes} * \text{qtdFunc}) / \text{clientesMaximo}$$

- É atribuído um score positivo aos dias que se igualem ao valor ideal calculado.
- É retirado score aos dias que se encontram abaixo do número mínimo de funcionários para cada turno.
- São retirados pontos ao score a turnos que tenham como funcionários trabalhar um número superior ou inferior ao ideal calculado.

3 Obstáculos Encontrados

Ao desenvolver o algoritmo de pesquisa encontramos alguns problemas que tornaram a resolução do problema um pouco mais complexa.

Inicialmente deparamo-nos com o facto de que seria impossível pesquisar todos os nodos da árvore, pois a sua quantidade faz com que seja uma tarefa impossível de resolver numa duração compreensível.

Posto isto, decidimos tomar como estratégia fazer uma pesquisa de profundidade 1, ou seja, apenas verificar o estado seguinte ao atual. No entanto, ao utilizar essa estratégia, seria necessário fazer com que a pesquisa seguisse o caminho de estados correto para alcançar o estado final desejável, pelo que tivemos de adicionar pontuação não apenas tendo em conta o estado final desejável, mas também pelo caminho de estados que queríamos que o algoritmo percorresse.

Ainda assim, os resultados não eram os desejados, como tal, procedemos à eliminação de estados indesejáveis (referido no capítulo anterior), como por exemplo, estados que dessem prioridade a preencher turnos onde já tinham funcionários colocados mesmo ainda havendo outros turnos sem funcionários.

No entanto, devido ao modo como era guardado o melhor estado por cada nível, foi verificada uma tendência a ser preenchido preferencialmente o final da semana. Para amenizar este efeito, passou a ser alternado o método de escolha, para níveis pares aceitamos valores maiores ou iguais ao score mais alto registado, para níveis ímpares aceitamos apenas valores maiores.

4 Resultados Obtidos

Para observação dos dados obtidos, é importante saber:

- Assumimos que é aceitável 1 funcionário por cada 100 clientes previstos.
- Quando o valor de previsões não é divisível por 100 com resto 0, por exemplo 80 ou 250, são aceites as abordagens de 'arredondamento' por excesso e por defeito.
- Uma lista de 14 funcionários e uma margem de 3.
- Uma lista de 10 funcionários e uma margem de 2.

Previsões

```
{
  "Segunda" : {"manha":100 , "tarde": 300},
  "Terca" : { "manha":200 , "tarde": 250},
  "Quarta" : { "manha":80 , "tarde": 325},
  "Quinta" : { "manha":300 , "tarde": 200},
  "Sexta" : { "manha":100 , "tarde": 400},
  "Sabado" : { "manha":700 , "tarde": 300},
  "Domingo" : {"manha":200 , "tarde": 500}
}
```

Estado Final 1

```
{
  "Segunda": { "manha":3 , "tarde":5},
  "Terca": {"manha":4 , "tarde":4},
  "Quarta": {"manha":5 , "tarde":5},
  "Quinta": {"manha":5 , "tarde":5},
  "Sexta": {"manha":5 , "tarde":6},
  "Sabado": {"manha":7 , "tarde":5},
  "Domingo": {"manha":4 , "tarde":7}
}
```


É verificada uma quase constante situação de overflow, pelo que se pode concluir que para o atual plano de previsões, há um excesso de funcionários disponíveis.

Estado Final 2

```
{
  "Segunda ": { "manha ":2 , "tarde ":3 } ,
  "Terca ": { "manha ":3 , "tarde ":3 } ,
  "Quarta ": { "manha ":4 , "tarde ":4 } ,
  "Quinta ": { "manha ":3 , "tarde ":3 } ,
  "Sexta ": { "manha ":4 , "tarde ":4 } ,
  "Sabado ": { "manha ":6 , "tarde ":3 } ,
  "Domingo ": { "manha ":3 , "tarde ":5 }
}
```

São verificáveis situações de overflow, no entanto, é perceptível que estamos mais próximos do número de funcionários desejado para as previsões dadas.