

Algoritmia:

AirScheduling Problem:

El problema a tractar en aquesta entrega es conegut com "AirScheduling" que consisteix en, donat un conjunt de vols entre ciutats, assignar la mínima quantitat de pilots que puguin realitzar tots els vols a les hores previstes.

Per tal de resoldre aquest problema em modelat el problema com una xarxa de fluxe i hem calculat el mínim valor que satisfà el MaxFlow associat a la xarxa.

Implementació:

La implementació d'aquest projecte ha consistit en:

Primer de tot modelar l'entrada (conjunt de vols) com una xarxa de fluxe, per fer-ho hem associat a cada vol una aresta que uneix les dues ciutats entre les quals s'efectua el vol. Per fer-ho més senzill hem utilitzat una estructura de dades *Vuelo* que conté la ciutat inicial, destí, el instant de temps inicial i el d'arribada.

Després hem afegit arestes entre ciutats que son destí d'un vol i origen d'un altre i el pilot arriba com a destí té temps de pilotar el següent vol.

I a més, per la segona versió, hem afegit encara més arestes considerant que el pilot viatge com a passatger.

Després, donada aquesta xarxa, l'hem considerat com una xarxa amb demandes ja que tots els vols s'han de realitzar i hem fet les transformacions convenients per modelar la xarxa com un graf de fluxe amb demandes.

I un cop fet tota la modelització hem calculat la K mínima de dues maneres:

1. Cerca linial:

Com sabiem que la K mínima estava entre 1 i nVols, anavem fent una busqueda linial començant en 1 fins a nVols del Max-Flow que hem resolt amb l'algoritme «Edmonds-Karp». A continuació parlarem del cost.

2. Cerca dicotomica:

Aquesta cerca, a diferencia de l'anterior ens a fet guanyar un factor logarítmic, que s'explicarà amb més detall a la secció cost .

En les dues cerques hem calculat el Max-Flow amb l'algoritme «Edmonds-Karp».

Finalment, el programa crea dos archius amb ea sortida de la versió 1 i de la versió 2.

Cost:

El cost temporal associat aquest problema es:

1. Cerca linial del valor de K:

$$O(nVols_1 * (nCiuatats_2 * nVols^2)).$$

1. nVols es la cuantiat de vols del problema, i a més la K màxima. Per tant el rang de valors de K està entre 1 i nVols.
2. nCiuatats es el nombre de ciutats que hi ha en el problema, com a màxim $2 * nVols$.

2. Cerca dicotòmica del valor K:
 $O(\log(nVols_1) * (nCitats_2 * nVols^2))$.

Com veiem hi ha una millora logarítmica.

Diferències entre versions:

La principal diferència entre les dues versions es que la segona té més arestes ja que els pilots poden viatjar com a passatger.

Hem observat que aquest fet equilibra més la distribució de viatges i per tant hi ha menys diferència entre el pilot que fa menys vols i el que en fa més.

Com que la segona versió2 és la mateixa xarxa de fluxe que la versió1 però amb més arestes, ens assegura que com a mínim tindren tants pilots com la versió1 però en moltes instàncies hem observat millores ja que la K s'ha reduït.

En el següent exemple es veu el resultat de les dues versions sobre la mateixa instància:

Versió 1	Versió 2
56 26 3 4 5 6 24 25 68 173 27 28 231 206 169 281 120 70 55 12 13 126 279 32 174 71 48 137 93 81 50 119 257 104 103 277 111 268 220 284 159 117 114 163 256 282 122 56 34 35 36 37 128 213 136 124 96 152 208 142 100 101 171 198 143 89 66 42 43 64 69 153 145 146 141 200 155 62 63 138 139 148 216 161 112 113 127 180 215 165 72 73 7 44 149 178 166 151 170 57 58 53 54 160 182 33 17 224 197 183 132 52 49 14 236 185 130 131 192 186 157 158 234 187 116 21 22 1 2 75 76 150 193 217 162 195 201 211 18 19 20 176 283 202 88 82 45 46 47 203 273 274 275 205 105 106 175 207 107 164 209 102 196 210 154 95 67 84 85	45 22 48 253 46 3 55 37 20 144 161 40 239 223 219 141 142 41 19 4 5 6 17 34 44 233 237 162 56 27 28 29 105 57 245 140 111 151 76 170 62 250 202 11 69 120 65 8 212 213 93 94 66 42 60 21 54 143 72 59 90 159 84 73 246 188 189 95 51 199 168 203 167 255 96 97 98 175 217 204 114 9 23 196 254 12 80 115 50 38 1 2 13 116 70 155 156 235 185 125 47 209 106 35 216 91 187 126 78 112 36 163 130 7 222 160 146 200 131 89 240 122 138 77 68 39 107 108 109 139 86 127 128 225 226 148 149 136 10 30 228 153 58 45 25 26 87 79 164 67 64 71 18 104 165 150 152 227 171 99 100 101 174 177 251 252 63 134 83 186 182 229 119 169 190 85 206 178 179 180

1. nVols es la quantitat de vols del problema, i a més la K màxima. Per tant el rang de valors de K està entre 1 i nVols.
2. nCiutats es el nombre de ciutats que hi ha en el problema, com a màxim $2 * nVols$.

219 223	194 195 242 32 33 92
225 212 199	197 24 110 102 147
227 108 109 110 140	198 132 14 15 16 121
228	201 135 137 193
229 184 39 40 23 30 191	205 117 118 145 61
235 226 251	210 123 124
237 232 245	211 247 43 103 208 218
239 247	214 215 172 81 207 243 31
240 238	220 154 192 230
241 59 97 98 115 177	231 49 221 166 176
242 233 254	232 74 75 234 181
243 194 262 270	238 113 173 158 157
244 168 147	241 236 52 53 248
249 79 41 65 80	244 88 133 82 224 129
250 218 179 221	
258 144 167 118 222 31 38	
259 156 121 172 248	
260 83 99	
261 252 253 129	
263 264	
265 204 51 29 60 74 246	
266 188 133 134 135 255 10 11 181	
267 125 90 91 92	
271 123 94 78 8 9 276 61	
272 230 189 190 86 87 15 16 214	

Per tant podem concloure que la segona versió es com a mínim igual de bona que la primera, per tant la segona és millor.

Taula de Resultats:

A continuació mostrem la taula de resultats obtinguts:

Versió 1:

Instància	K òptima	Temps CPU (s) amb Cerca linial
instance_100_10_10.air	53	47
instance_100_10_1.air	57	61
instance_100_10_2.air	53	73
instance_100_10_3.air	49	44
instance_100_10_4.air	57	44
instance_100_10_5.air	53	68
instance_100_10_6.air	56	65
instance_100_10_7.air	56	67
instance_100_10_8.air	51	46
instance_100_10_9.air	53	52

1. nVols es la quantitat de vols del problema, i a més la K màxima. Per tant el rang de valors de K està entre 1 i nVols.
2. nCiutats es el nombre de ciutats que hi ha en el problema, com a màxim $2 \cdot nVols$.

Instancia	K òptima	Temps CPU (s) amb Cerca dicotomica
instance_100_10_10.air	53	4
instance_100_10_1.air	57	6
instance_100_10_2.air	53	7
instance_100_10_3.air	49	4
instance_100_10_4.air	57	4
instance_100_10_5.air	53	6
instance_100_10_6.air	56	6
instance_100_10_7.air	56	6
instance_100_10_8.air	51	4
instance_100_10_9.air	53	5

De la segona versió que també està implementada no tenim el temps de CPU pero a l'archiu *Resultats2.txt* que adjuntem amb la pràctica estan totes les K obtingudes.

Bibliografia:

- Apunts de l'assignatura.
- Llibre: "Kleinberg and Tardos, 2005"
- http://es.wikipedia.org/wiki/Algoritmo_de_Edmonds-Karp

1. nVols es la quantitat de vols del problema, i a més la K màxima. Per tant el rang de valors de K està entre 1 i nVols.
2. nCiutats es el nombre de ciutats que hi ha en el problema, com a màxim 2^n Vols.