

Modelos y Optimización I

Trabajo práctico: Problema Combinatorio

de Valais, Ezequiel (94463) Rozanec, Matias (97404)

Octubre 2017

Contents

TRABAJO PRÁCTICO

Problema Combinatorio

IngFraTech está evaluando ahora la ubicación de los datacenters necesarios para proveer los nuevos servicios.

Dado que la mayoría de sus clientes se encuentra en Estados Unidos, deciden ubicar allí los datacenter con la idea de hacer mínima la latencia entre los datacenters y los clientes. Como en un principio no saben de dónde serán sus clientes (es decir, a qué estado pertenecerán) la empresa plantea reducir la latencia global del servicio definiéndolo de la siguiente manera:

"La latencia global del sistema es la suma de las latencias de cada estado. El cálculo de latencia de cada estado se hace considerando la distancia con el datacenter más próximo, estimándose 1 ms de latencia por cada milla de distancia."

Actualmente cuenta con 2 datacenters, uno en Oregon y otro en Florida. Por motivos económicos la empresa no puede agregar más de 3 datacenters nuevos.

Dado que no esperan clientes de Hawái y Alaska, estos estados no son tenidos en cuenta para calcular la latencia global.

¿Qué es lo mejor que puede hacer IngFraTech?



Part I

Descripción de la situación problemática

Se trata de un problema de combinatoria, en el que habrá que incluir variables continuas y booleanas.

En esta instancia podemos afirmar que habrá que considerar una variable *latencia* que deberá ser una variable continua; así como una variable booleana que indique si un determinado datacenter se encuentra instalado en un estado específico.

Part II

Objetivo

Determinar en qué estados van a estar los 3 nuevos datacenters (DB, DC, DD) durante un período de tiempo para minimizar la latencia global del sistema.

Part III

Hipótesis y supuestos

- Se instalarán los 3 datacenters puesto que cada datacenter agregado va a reducir la latencia.
- Se tomará un punto en cada estado. El mismo será referente para calcular las distancias entre estados y las respectivas latencias. No hay opción de instalar un datacenter en otro punto del estado que el mencionado.
- Para el cálculo se consideran únicamente las distancias, no la cantidad de usuarios por estado. O expresado de otra forma: para el modelo, la distribución de usuarios es uniforme a lo largo de todos los estados y en cada uno de los estados.
- No puede haber dos datacenters en un mismo estado.
- Los costos de instalación y mantenimiento de datacenters, así como cualquier otro costo que pueda implicar la instalación de los mismos, no serán tenidos en cuenta por el modelo.

Part IV

Definición de variables, incluyendo unidades

- L_i : variable continua que indica la latencia (en ms) correspondiente al estado i . $\forall i \in [1, 48]$
- DC_i : variable continua que indica la distancia (en millas) del datacenter C al estado i . (ídem para datacenters D y E . DA_i y DB_i son datos conocidos). $\forall i \in [1, 48]$
- YA_i : variable bivalente que vale 0 cuando el estado i tiene la menor latencia. (ídem B, C, D, E)
- YCe_i : variable bivalente que vale 1 cuando el datacenter C está en el Estado i . (ídem D, E)

Constantes

- D_{ij} : distancia entre estado i y estado j .
- M : valor mayor a cualquier distancia posible. Su valor se definirá al momento de pasar el modelo a software.

Part V

Modelo de Programación Lineal Continua

$$\sum_i YA_i + YB_i + YC_i + YD_i + YE_i = 1 \quad (1)$$

1 Menor Distancia

Cada L_i tendrá como cota superior la distancia al datacenter más próximo, y como cota inferior esa misma distancia en el caso de que el estado i tenga la menor distancia.

$M \rightarrow \infty$

$$DA_i - M * YA_i \leq L_i \leq DA_i, \quad \forall i \in [1, 48] \quad (2)$$

$$DB_i - M * YB_i \leq L_i \leq DB_i, \quad \forall i \in [1, 48] \quad (3)$$

$$DC_i - M * YC_i \leq L_i \leq DC_i, \quad \forall i \in [1, 48] \quad (4)$$

$$DD_i - M * YD_i \leq L_i \leq DD_i, \quad \forall i \in [1, 48] \quad (5)$$

$$DE_i - M * YE_i \leq L_i \leq DE_i, \quad \forall i \in [1, 48] \quad (6)$$

2 Asociación de Datacenter a estado

Se asegura de que cada datacenter pueda ser asignado únicamente a un estado.

$$\sum_{i=1}^{48} YCe_i = 1 \quad (7)$$

$$\sum_{i=1}^{48} YDe_i = 1 \quad (8)$$

$$\sum_{i=1}^{48} YEe_i = 1 \quad (9)$$

3 Asociación de distancia de un estado al datacenter correspondiente

$$DC_i = \sum_{j=1}^{48} D_{ij} * YCe_j \quad (10)$$

$$DD_i = \sum_{j=1}^{48} D_{ij} * YDe_j \quad (11)$$

$$DE_i = \sum_{j=1}^{48} D_{ij} * YEe_j, \quad \forall i \in [1, 48] \quad (12)$$

Part VI

Funcional

$$Z(min) = \sum_{i=1}^{48} L_i \quad (13)$$

Part VII

Modelo y salida

```

1 # conjuntos
2 ##set ESTADOS := {1..48};
3 set ESTADOS;
4 set s dimen 2;
5 set s1;
6
7 # Parametros (constantes)
8 # Distancias entre ESTADOS
9 param Dij{i in ESTADOS, j in ESTADOS};
10
11 # DA i : variable continua que indica la distancia del datacenter A al estado
12 # i. ( dem para datacenter B)
13 param DAi{i in ESTADOS};
14 param DBi{i in ESTADOS};
15
16 # M : valor mayor a cualquier distancia posible.
17 param M := 10000;
18
19
20 # all distances
21 table tab_distances IN "CSV" "distances.csv" : s <- [state1, state2], Dij~
    distance;
22
23 # distances from datacenters A and B, which correspond to states
24 # of Florida and Oregon respectively.
25 table tab_florida_distances IN "CSV" "florida_distances.csv" : ESTADOS <- [
    state2], DAi~distance;
26 table tab_oregon_distances IN "CSV" "oregon_distances.csv" : s1 <- [state2],
    DBi~distance;
27
28 # Variables
29 # Li: variable continua que indica la latencia correspondiente al estado i
30 var Li{i in ESTADOS} >= 0;
31
32 # DC i : variable continua que indica la distancia del datacenter C
33 # al estado i. Idem para D y E.
34 var DCi{i in ESTADOS} >= 0;
35 var DDi{i in ESTADOS} >= 0;
36 var DEi{i in ESTADOS} >= 0;
37
38 # YAi: variable bivalente que vale 0 cuando la distancia del Datacenter
39 # A al Estado i es la menor respecto al resto de los Datacenters. ( dem
40 # B, C, D, E)
41 var YAi{i in ESTADOS} >= 0, binary;
42 var YBi{i in ESTADOS} >= 0, binary;
43 var YCi{i in ESTADOS} >= 0, binary;
44 var YDi{i in ESTADOS} >= 0, binary;
45 var YEi{i in ESTADOS} >= 0, binary;
46
47 # Y Ce i : variable bivalente que vale 1 cuando el datacenter C est en el
48 # Estado i.

```

```

49 var YCe{i in ESTADOS} >= 0, binary;
50 var YDe{i in ESTADOS} >= 0, binary;
51 var YEe{i in ESTADOS} >= 0, binary;
52
53 # Restricciones
54 # Cada L i tendr como cota superior la distancia al datacenter m s prximo
    , y
55 # como cota inferior esa misma distancia en el caso de que el estado i tenga
    la
56 # menor distancia.
57 s.t. cota_sup_dcA{i in ESTADOS}: Li[i] <= DAi[i];
58 s.t. cota_sup_dcB{i in ESTADOS}: Li[i] <= DBi[i];
59 s.t. cota_sup_dcC{i in ESTADOS}: Li[i] <= DCi[i];
60 s.t. cota_sup_dcD{i in ESTADOS}: Li[i] <= DDi[i];
61 s.t. cota_sup_dcE{i in ESTADOS}: Li[i] <= DEi[i];
62
63 s.t. cota_inf_dcA{i in ESTADOS}: Li[i] >= DAi[i] - M * YAi[i];
64 s.t. cota_inf_dcB{i in ESTADOS}: Li[i] >= DBi[i] - M * YBi[i];
65 s.t. cota_inf_dcC{i in ESTADOS}: Li[i] >= DCi[i] - M * YCi[i];
66 s.t. cota_inf_dcD{i in ESTADOS}: Li[i] >= DDi[i] - M * YDi[i];
67 s.t. cota_inf_dcE{i in ESTADOS}: Li[i] >= DEi[i] - M * YEi[i];
68
69 # sumatoria de YXi debe ser 4
70 s.t. total.datacenters{i in ESTADOS}: YAi[i] + YBi[i] + YCi[i] + YDi[i] + YEi[
    i] = 4;
71
72 # Asociacion de datacenter a estado
73 s.t. asoc_dcC: sum{i in ESTADOS} YCe{i} = 1;
74 s.t. asoc_dcD: sum{i in ESTADOS} YDe{i} = 1;
75 s.t. asoc_dcE: sum{i in ESTADOS} YEe{i} = 1;
76
77 # Asociaci n de distancia de un estado al datacenter correspondiente
78 s.t. asoc_dist_dcC_estadoi{i in ESTADOS}: sum{(i,j) in s} Dij[i,j] * YCe{j} =
    DCi[i];
79 s.t. asoc_dist_dcD_estadoi{i in ESTADOS}: sum{(i,j) in s} Dij[i,j] * YDe{j} =
    DDi[i];
80 s.t. asoc_dist_dcE_estadoi{i in ESTADOS}: sum{(i,j) in s} Dij[i,j] * YEe{j} =
    DEi[i];
81
82 # funcional
83 minimize z: sum{i in ESTADOS} Li[i];
84
85 end;

```



```

1 Problem:      comb
2 Rows:        116
3 Columns:     96 (64 integer , 64 binary)
4 Non-zeros:   432
5 Status:      INTEGER OPTIMAL
6 Objective:   z = 1457.681632 (MINimum)

```

No.	Row name	Activity	Lower bound	Upper bound
1	cota_sup_dcA [Alabama]	223.749		223.749
2	cota_sup_dcA [Arizona]	817.982		3090.64
3	cota_sup_dcA [Arkansas]	0		895.532
4	cota_sup_dcA [California]	415.951		4130.54
5	cota_sup_dcA [Colorado]	0		2296.9
6	cota_sup_dcA [Connecticut]	0		1307.57
7	cota_sup_dcA [Florida]	0		-0
8	cota_sup_dcA [Oregon]	0		4291.9
9	cota_sup_dcB [Alabama]	223.749		4077.38
10	cota_sup_dcB [Arizona]	817.982		1348.72
11	cota_sup_dcB [Arkansas]	0		3417.83
12	cota_sup_dcB [California]	415.951		415.951
13	cota_sup_dcB [Colorado]	0		2018.37
14	cota_sup_dcB [Connecticut]	0		5596.86
15	cota_sup_dcB [Florida]	0		4291.9
16	cota_sup_dcB [Oregon]	0		-0
17	cota_sup_dcC [Alabama]	-1853.4		-0
18	cota_sup_dcC [Arizona]	0		-0
19	cota_sup_dcC [Arkansas]	-1408.12		-0
20	cota_sup_dcC [California]	-1417.72		-0
21	cota_sup_dcC [Colorado]	0		-0
22	cota_sup_dcC [Connecticut]	-3591.82		-0
23	cota_sup_dcC [Florida]	-2296.9		-0
24	cota_sup_dcC [Oregon]			

57			-2018.37		-0
58	25	cota_sup_dcD [Alabama]			
59			-449.066		-0
60	26	cota_sup_dcD [Arizona]			
61			-1377.38		-0
62	27	cota_sup_dcD [Arkansas]			
63			0		-0
64	28	cota_sup_dcD [California]			
65			-2824.64		-0
66	29	cota_sup_dcD [Colorado]			
67			-1408.12		-0
68	30	cota_sup_dcD [Connecticut]			
69			-2183.71		-0
70	31	cota_sup_dcD [Florida]			
71			-895.532		-0
72	32	cota_sup_dcD [Oregon]			
73			-3417.83		-0
74	33	cota_sup_dcE [Alabama]			
75			-1295.76		-0
76	34	cota_sup_dcE [Arizona]			
77			-3550.92		-0
78	35	cota_sup_dcE [Arkansas]			
79			-2183.71		-0
80	36	cota_sup_dcE [California]			
81			-5007.4		-0
82	37	cota_sup_dcE [Colorado]			
83			-3591.82		-0
84	38	cota_sup_dcE [Connecticut]			
85			0		-0
86	39	cota_sup_dcE [Florida]			
87			-1307.57		-0
88	40	cota_sup_dcE [Oregon]			
89			-5596.86		-0
90	41	cota_inf_dcA [Alabama]			
91			223.749	223.749	
92	42	cota_inf_dcA [Arizona]			
93			10818	3090.64	
94	43	cota_inf_dcA [Arkansas]			
95			10000	895.532	
96	44	cota_inf_dcA [California]			
97			10416	4130.54	
98	45	cota_inf_dcA [Colorado]			
99			10000	2296.9	
100	46	cota_inf_dcA [Connecticut]			
101			10000	1307.57	
102	47	cota_inf_dcA [Florida]			
103			0	-0	
104	48	cota_inf_dcA [Oregon]			
105			10000	4291.9	
106	49	cota_inf_dcB [Alabama]			
107			10223.7	4077.38	
108	50	cota_inf_dcB [Arizona]			
109			10818	1348.72	
110	51	cota_inf_dcB [Arkansas]			
111			10000	3417.83	
112	52	cota_inf_dcB [California]			
113			415.951	415.951	

114	53	cota_inf_dcB [Colorado]		
115			10000	2018.37
116	54	cota_inf_dcB [Connecticut]		
117			10000	5596.86
118	55	cota_inf_dcB [Florida]		
119			10000	4291.9
120	56	cota_inf_dcB [Oregon]		
121			0	-0
122	57	cota_inf_dcC [Alabama]		
123			8146.6	-0
124	58	cota_inf_dcC [Arizona]		
125			0	-0
126	59	cota_inf_dcC [Arkansas]		
127			8591.88	-0
128	60	cota_inf_dcC [California]		
129			8582.28	-0
130	61	cota_inf_dcC [Colorado]		
131			0	-0
132	62	cota_inf_dcC [Connecticut]		
133			6408.18	-0
134	63	cota_inf_dcC [Florida]		
135			7703.1	-0
136	64	cota_inf_dcC [Oregon]		
137			7981.63	-0
138	65	cota_inf_dcD [Alabama]		
139			9550.93	-0
140	66	cota_inf_dcD [Arizona]		
141			8622.62	-0
142	67	cota_inf_dcD [Arkansas]		
143			0	-0
144	68	cota_inf_dcD [California]		
145			7175.36	-0
146	69	cota_inf_dcD [Colorado]		
147			8591.88	-0
148	70	cota_inf_dcD [Connecticut]		
149			7816.29	-0
150	71	cota_inf_dcD [Florida]		
151			9104.47	-0
152	72	cota_inf_dcD [Oregon]		
153			6582.17	-0
154	73	cota_inf_dcE [Alabama]		
155			8704.24	-0
156	74	cota_inf_dcE [Arizona]		
157			6449.08	-0
158	75	cota_inf_dcE [Arkansas]		
159			7816.29	-0
160	76	cota_inf_dcE [California]		
161			4992.6	-0
162	77	cota_inf_dcE [Colorado]		
163			6408.18	-0
164	78	cota_inf_dcE [Connecticut]		
165			0	-0
166	79	cota_inf_dcE [Florida]		
167			8692.43	-0
168	80	cota_inf_dcE [Oregon]		
169			4403.14	-0
170	81	total_datacenters [Alabama]		

171		4	4	=
172	82 total_datacenters [Arizona]	4	4	=
173		4	4	=
174	83 total_datacenters [Arkansas]	4	4	=
175		4	4	=
176	84 total_datacenters [California]	4	4	=
177		4	4	=
178	85 total_datacenters [Colorado]	4	4	=
179		4	4	=
180	86 total_datacenters [Connecticut]	4	4	=
181		4	4	=
182	87 total_datacenters [Florida]	4	4	=
183		4	4	=
184	88 total_datacenters [Oregon]	4	4	=
185		4	4	=
186	89 asoc_dcC	1	1	=
187	90 asoc_dcD	1	1	=
188	91 asoc_dcE	1	1	=
189	92 asoc_dist_dcC_estadoi [Alabama]	0	-0	=
190		0	-0	=
191	93 asoc_dist_dcC_estadoi [Arizona]	0	-0	=
192		0	-0	=
193	94 asoc_dist_dcC_estadoi [Arkansas]	0	-0	=
194		0	-0	=
195	95 asoc_dist_dcC_estadoi [California]	0	-0	=
196		0	-0	=
197	96 asoc_dist_dcC_estadoi [Colorado]	0	-0	=
198		0	-0	=
199	97 asoc_dist_dcC_estadoi [Connecticut]	0	-0	=
200		0	-0	=
201	98 asoc_dist_dcC_estadoi [Florida]	0	-0	=
202		0	-0	=
203	99 asoc_dist_dcC_estadoi [Oregon]	0	-0	=
204		0	-0	=
205	100 asoc_dist_dcD_estadoi [Alabama]	0	-0	=
206		0	-0	=
207	101 asoc_dist_dcD_estadoi [Arizona]	0	-0	=
208		0	-0	=
209	102 asoc_dist_dcD_estadoi [Arkansas]	0	-0	=
210		0	-0	=
211	103 asoc_dist_dcD_estadoi [California]	0	-0	=
212		0	-0	=
213	104 asoc_dist_dcD_estadoi [Colorado]	0	-0	=
214		0	-0	=
215	105 asoc_dist_dcD_estadoi [Connecticut]	0	-0	=
216		0	-0	=
217	106 asoc_dist_dcD_estadoi [Florida]	0	-0	=
218		0	-0	=
219	107 asoc_dist_dcD_estadoi [Oregon]	0	-0	=
220		0	-0	=
221	108 asoc_dist_dcE_estadoi [Alabama]	0	-0	=
222		0	-0	=
223	109 asoc_dist_dcE_estadoi [Arizona]	0	-0	=
224		0	-0	=
225	110 asoc_dist_dcE_estadoi [Arkansas]	0	-0	=
226		0	-0	=
227	111 asoc_dist_dcE_estadoi [California]			

228			0	-0	=
229	112	asoc_dist_dcE_estadoi [Colorado]			
230			0	-0	=
231	113	asoc_dist_dcE_estadoi [Connecticut]			
232			0	-0	=
233	114	asoc_dist_dcE_estadoi [Florida]			
234			0	-0	=
235	115	asoc_dist_dcE_estadoi [Oregon]			
236			0	-0	=
237	116	z	1457.68		
238					
239	No.	Column name	Activity	Lower bound	Upper bound
240					
241	1	Li [Alabama]	223.749	0	
242	2	Li [Arizona]	817.982	0	
243	3	Li [Arkansas]	0	0	
244	4	Li [California]			
245			415.951	0	
246	5	Li [Colorado]	0	0	
247	6	Li [Connecticut]			
248			0	0	
249	7	Li [Florida]	0	0	
250	8	Li [Oregon]	0	0	
251	9	DCi [Alabama]	2077.15	0	
252	10	DCi [Arizona]	817.982	0	
253	11	DCi [Arkansas]			
254			1408.12	0	
255	12	DCi [California]			
256			1833.67	0	
257	13	DCi [Colorado]			
258			0	0	
259	14	DCi [Connecticut]			
260			3591.82	0	
261	15	DCi [Florida]	2296.9	0	
262	16	DCi [Oregon]	2018.37	0	
263	17	DDi [Alabama]	672.815	0	
264	18	DDi [Arizona]	2195.36	0	
265	19	DDi [Arkansas]			
266			0	0	
267	20	DDi [California]			
268			3240.59	0	
269	21	DDi [Colorado]			
270			1408.12	0	
271	22	DDi [Connecticut]			
272			2183.71	0	
273	23	DDi [Florida]	895.532	0	
274	24	DDi [Oregon]	3417.83	0	
275	25	DEi [Alabama]	1519.51	0	
276	26	DEi [Arizona]	4368.9	0	
277	27	DEi [Arkansas]			
278			2183.71	0	
279	28	DEi [California]			
280			5423.35	0	
281	29	DEi [Colorado]			
282			3591.82	0	
283	30	DEi [Connecticut]			
284			0	0	

285	31	DEi[Florida]	1307.57	0	
286	32	DEi[Oregon]	5596.86	0	
287	33	YAi[Alabama] *	0	0	1
288	34	YAi[Arizona] *	1	0	1
289	35	YAi[Arkansas]			
290		*	1	0	1
291	36	YAi[California]			
292		*	1	0	1
293	37	YAi[Colorado]			
294		*	1	0	1
295	38	YAi[Connecticut]			
296		*	1	0	1
297	39	YAi[Florida] *	0	0	1
298	40	YAi[Oregon] *	1	0	1
299	41	YBi[Alabama] *	1	0	1
300	42	YBi[Arizona] *	1	0	1
301	43	YBi[Arkansas]			
302		*	1	0	1
303	44	YBi[California]			
304		*	0	0	1
305	45	YBi[Colorado]			
306		*	1	0	1
307	46	YBi[Connecticut]			
308		*	1	0	1
309	47	YBi[Florida] *	1	0	1
310	48	YBi[Oregon] *	0	0	1
311	49	YCi[Alabama] *	1	0	1
312	50	YCi[Arizona] *	0	0	1
313	51	YCi[Arkansas]			
314		*	1	0	1
315	52	YCi[California]			
316		*	1	0	1
317	53	YCi[Colorado]			
318		*	0	0	1
319	54	YCi[Connecticut]			
320		*	1	0	1
321	55	YCi[Florida] *	1	0	1
322	56	YCi[Oregon] *	1	0	1
323	57	YDi[Alabama] *	1	0	1
324	58	YDi[Arizona] *	1	0	1
325	59	YDi[Arkansas]			
326		*	0	0	1
327	60	YDi[California]			
328		*	1	0	1
329	61	YDi[Colorado]			
330		*	1	0	1
331	62	YDi[Connecticut]			
332		*	1	0	1
333	63	YDi[Florida] *	1	0	1
334	64	YDi[Oregon] *	1	0	1
335	65	YEi[Alabama] *	1	0	1
336	66	YEi[Arizona] *	1	0	1
337	67	YEi[Arkansas]			
338		*	1	0	1
339	68	YEi[California]			
340		*	1	0	1
341	69	YEi[Colorado]			

342		*	1	0	1
343	70	YEi[Connecticut]			
344		*	0	0	1
345	71	YEi[Florida]	1	0	1
346	72	YEi[Oregon]	1	0	1
347	73	YCe[Alabama]			
348		*	0	0	1
349	74	YCe[Arizona]			
350		*	0	0	1
351	75	YCe[Arkansas]			
352		*	0	0	1
353	76	YCe[California]			
354		*	0	0	1
355	77	YCe[Colorado]			
356		*	1	0	1
357	78	YCe[Connecticut]			
358		*	0	0	1
359	79	YCe[Florida]			
360		*	0	0	1
361	80	YCe[Oregon]	0	0	1
362	81	YDe[Alabama]			
363		*	0	0	1
364	82	YDe[Arizona]			
365		*	0	0	1
366	83	YDe[Arkansas]			
367		*	1	0	1
368	84	YDe[California]			
369		*	0	0	1
370	85	YDe[Colorado]			
371		*	0	0	1
372	86	YDe[Connecticut]			
373		*	0	0	1
374	87	YDe[Florida]			
375		*	0	0	1
376	88	YDe[Oregon]	0	0	1
377	89	YEe[Alabama]			
378		*	0	0	1
379	90	YEe[Arizona]			
380		*	0	0	1
381	91	YEe[Arkansas]			
382		*	0	0	1
383	92	YEe[California]			
384		*	0	0	1
385	93	YEe[Colorado]			
386		*	0	0	1
387	94	YEe[Connecticut]			
388		*	1	0	1
389	95	YEe[Florida]			
390		*	0	0	1
391	96	YEe[Oregon]	0	0	1
392					
393	Integer feasibility conditions:				
394					
395	KKT.PE: max.abs.err = 1.82e-12 on row 74				
396	max.rel.err = 8.41e-17 on row 74				
397	High quality				
398					

```
399 KKT.PB: max.abs.err = 0.00e+00 on row 0
400         max.rel.err = 0.00e+00 on row 0
401         High quality
402
403 End of output
```


Part VIII

Conclusiones

Se decidió realizar la corrida sobre los siguientes estados:

Alabama, Arizona, Arkansas, California, Colorado, Connecticut, Florida, Oregon

Fuera de Florida (Datacenter A) y Oregon (Datacenter B), las 3 nuevas datacenters son las siguientes

Datacenter C: Colorado (77 YCe[Colorado] 1)

Datacenter D: Arkansas (83 YDe[Arkansas] 1)

Datacenter E: Connecticut (94 YEE[Connecticut] 1)

La latencia global del sistema es: 1457.681632

Las latencias de cada estado son las siguientes:

Li[Alabama] 223.749

Li[Arizona] 817.982

Li[Arkansas] 0

Li[California] 415.951

Li[Colorado] 0

Li[Connecticut] 0

Li[Florida] 0

Li[Oregon] 0

Esto tiene sentido pues solo Alabama, Arizona y California no tienen un datacenter en el estado.

El datacenter C de Colorado es el mas cercano a Arizona (la distancia del datacenter al estado es la misma que su latencia mínima):

DCi[Arizona] 817.982

El datacenter A de Florida es el mas cercano a Alabama (define la cota inferior a la latencia del estado)

cota_inf_dcA[Alabama] 223.749

El datacenter B de Oregon es el mas cercano a California (define la cota inferior a la latencia del estado)

cota_inf_dcB[California] 415.951

Part IX

Heurística

compilar: g++ -std=c++11 heuristic.cpp -o heuristic
correr: cat distances.csv | ./heuristic

```
1 #include <iostream>
2 #include <map>
3 #include <cmath>
4
5 void get_problem_values(int &num_of_states, int &num_of_datacenters){
6     std::cout << "Insert number of states" << std::endl;
7     std::cin >> num_of_states;
8     std::cout << "Insert number of datacenters" << std::endl;
9     std::cin >> num_of_datacenters;
10 }
11
12 void fill_distances_matrix(int num_of_states,
13                           std::map<int, std::map<int, double> > &distances){
14     for(int i = 1; i <= num_of_states; ++i){
15         for(int j = i + 1; j <= num_of_states; ++j){
16             double current_distance;
17             std::cout << "insert distance between state_"
18                     << i << " and " << j << std::endl;
19             std::cin >> current_distance;
20             distances[i][j] = current_distance;
21             distances[j][i] = current_distance;
22         }
23         /* distance from a state to itself is 0 */
24         distances[i][i] = 0;
25     }
26 }
27
28 double calculate_current_latency(int num_of_states,
29                                 const std::map<int, int> &datacenters,
30                                 const std::map<int, std::map<int, double> >
31                                 &distances){
32     double total_latency = 0;
33     /* for each state, check the minimum distance to each datacenter */
34     for(int i = 1; i <= num_of_states; ++i){
35         double this_state_lat = DBL_MAX;
36         for(auto it = datacenters.begin(); it != datacenters.end(); ++it){
37             double read_lat = distances.at(i).at(it->second);
38             if (read_lat < this_state_lat){
39                 this_state_lat = read_lat;
40             }
41         }
42         total_latency += this_state_lat;
43     }
44     return total_latency;
45 }
46
47 void manage_manually_allocated_datacenters(
```

```

48     int num_of_states, int &num_of_manually_allocated_datacenters,
49     std::map<int, int> &datacenters,
50     std::map<int, std::map<int, double> > &distances){
51     std::cout << "How_many_datacenters_are_manually_allocated?" << std::endl;
52     std::cin >> num_of_manually_allocated_datacenters;
53
54     for (int i = 1; i <= num_of_manually_allocated_datacenters; ++i){
55         int state;
56         std::cout << "Insert_number_of_state_where_datacenter_" << i
57             << "_should_be." << std::endl;
58         std::cin >> state;
59         datacenters[i] = state;
60         /* display current latency */
61         std::cout << "latency_after_datacenter_" << i
62             << "_was_allocated:_"
63             << calculate_current_latency(num_of_states, datacenters,
64                                         distances) << std::endl;
65     }
66 }
67
68 bool state_already_has_datacenter(int state,
69                                 const std::map<int, int> &datacenters){
70     for (auto it = datacenters.begin(); it != datacenters.end(); ++it){
71         if (it->second == state){
72             return true;
73         }
74     }
75     return false;
76 }
77
78 int locate_new_datacenter(int num_of_datacenter,
79                          int num_of_states, std::map<int, int> &datacenters,
80                          const std::map<int, std::map<int, double> >
81                          &distances){
82     /* returns number of state which minimizes latency */
83     double current_lat = DBL_MAX;
84     int state_that_minimizes_lat;
85     for(int i = 1; i <= num_of_states; ++i){
86         if (!state_already_has_datacenter(i, datacenters)){
87             datacenters[num_of_datacenter] = i;
88             if (calculate_current_latency(num_of_states, datacenters,
89                                         distances)
90                 < current_lat){
91                 state_that_minimizes_lat = i;
92             }
93         }
94     }
95     return state_that_minimizes_lat;
96 }
97
98 void locate_remaining_datacenters(int num_of_states, int num_of_datacenters,
99                                 int num_of_manually_allocated_datacenters,
100                                 std::map<int, int> &datacenters,
101                                 std::map<int, std::map<int, double> >
102                                 &distances){
103     for (int i = num_of_manually_allocated_datacenters + 1;
104          i <= num_of_datacenters; ++i){

```

```

104         datacenters[i] = locate_new_datacenter(i, num_of_states,
105                                                datacenters, distances);
106         /* display current latency */
107         std::cout << "latency_after_datacenter_" << i << "_located:" <<
108             calculate_current_latency(
109                 num_of_states, datacenters, distances) << std::endl;
110     }
111 }
112
113 void inform_datacenter_positions(const std::map<int, int> &datacenters){
114     for (auto it = datacenters.begin(); it != datacenters.end(); ++it){
115         std::cout << "datacenter\t" << it->first <<
116             "\tlocated_in_state\t" << it->second << std::endl;
117     }
118 }
119
120 int main(){
121     int num_of_states;
122     int num_of_datacenters;
123
124     /* A map is used to store de distance matrix.
125      * States are identified with numbers from 1 to n,
126      * where n is the number of states to consider. */
127     std::map<int, std::map<int, double> > distances;
128
129     /* A map is used to link each datacenter to a state.
130      * K = datacenter.
131      * V = state. */
132     std::map<int, int> datacenters;
133
134     get_problem_values(num_of_states, num_of_datacenters);
135
136     fill_distances_matrix(num_of_states, distances);
137
138     int num_of_manually_allocated_datacenters;
139     manage_manually_allocated_datacenters(
140         num_of_states, num_of_manually_allocated_datacenters, datacenters,
141         distances);
142
143     locate_remaining_datacenters(num_of_states, num_of_datacenters,
144                                 num_of_manually_allocated_datacenters,
145                                 datacenters, distances);
146
147     /* inform results */
148     std::cout << "***_Final_datacenter_positions_***" << std::endl;
149     inform_datacenter_positions(datacenters);
150     return 0;
151 }

```

Part X

Resultados de Heurística

Se corrió la heurística con los 8 estados y se consiguieron los siguientes resultados:

```
Insert number of state where datacenter 1 should be.
latency after datacenter 1 was allocated: 16236.8
Insert number of state where datacenter 2 should be.
latency after datacenter 2 was allocated: 6209.89
latency after datacenter 3 located: 4902.32
latency after datacenter 4 located: 2353.21
latency after datacenter 5 located: 1937.26
*** Final datacenter positions ***
datacenter 1 located in state 7
datacenter 2 located in state 8
datacenter 3 located in state 6
datacenter 4 located in state 5
datacenter 5 located in state 4
```

Los datacenters seleccionados quedaron en los estados California Colorado y Connecticut (4, 5 y 6) dejando la latencia en 1937.26 Este resultado es peor comparado con el modelo de GLPK que dejó la latencia en 1457

Se corrió la heurística con los 48 estados y se consiguieron los siguientes resultados:

```
Insert number of state where datacenter 1 should be.
latency after datacenter 1 was allocated: 68827.2
Insert number of state where datacenter 2 should be.
latency after datacenter 2 was allocated: 44071.9
latency after datacenter 3 located: 32303.7
latency after datacenter 4 located: 27360.3
latency after datacenter 5 located: 22764.1
*** Final datacenter positions ***
datacenter 1 located in state 8
datacenter 2 located in state 35
datacenter 3 located in state 48
datacenter 4 located in state 47
datacenter 5 located in state 46
```

Los datacenters seleccionados quedaron en los estados Wyoming, Wisconsin y West Virginia dejando la latencia en 22764.1