

Modelos y Optimización I
Trabajo práctico: Problema Combinatorio

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

Octubre 2017

Contents

I	Descripción de la situación problemática	4
II	Objetivo	4
III	Hipótesis y supuestos	4
IV	Definición de variables, incluyendo unidades	5
V	Modelo de Programación Lineal Continua	5
1	Menor Distancia	5
2	Asociación de Datacenter a estado	6
3	Asociación de distancia de un estado al datacenter correspondiente	6
VI	Funcional	6
VII	Modelo y salida	7
VIII	Conclusiones	17
IX	Heurística	18
X	Resultados de Heurística	41

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

$$YA_i + YB_i + YC_i + YD_i + YE_i = 1 \quad \forall i \in [1, 48] \quad (1)$$

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

$$DC_i = \sum_{j=1}^{48} D_{ij} * YCe_j \quad \forall i \in [1, 48] \quad (10)$$

$$DD_i = \sum_{j=1}^{48} D_{ij} * YDe_j \quad \forall i \in [1, 48] \quad (11)$$

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

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 YDi[Alabama] *	1	0	1
336	66 YDi[Arizona] *	1	0	1
337	67 YDi[Arkansas]			
338		1	0	1
339	68 YDi[California]			
340		1	0	1
341	69 YDi[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	YCeI[Alabama]			
348		*	0	0	1
349	74	YCeI[Arizona]			
350		*	0	0	1
351	75	YCeI[Arkansas]			
352		*	0	0	1
353	76	YCeI[California]			
354		*	0	0	1
355	77	YCeI[Colorado]			
356		*	1	0	1
357	78	YCeI[Connecticut]			
358		*	0	0	1
359	79	YCeI[Florida]			
360		*	0	0	1
361	80	YCeI[Oregon]	0	0	1
362	81	YDei[Alabama]			
363		*	0	0	1
364	82	YDei[Arizona]			
365		*	0	0	1
366	83	YDei[Arkansas]			
367		*	1	0	1
368	84	YDei[California]			
369		*	0	0	1
370	85	YDei[Colorado]			
371		*	0	0	1
372	86	YDei[Connecticut]			
373		*	0	0	1
374	87	YDei[Florida]			
375		*	0	0	1
376	88	YDei[Oregon]	0	0	1
377	89	YEei[Alabama]			
378		*	0	0	1
379	90	YEei[Arizona]			
380		*	0	0	1
381	91	YEei[Arkansas]			
382		*	0	0	1
383	92	YEei[California]			
384		*	0	0	1
385	93	YEei[Colorado]			
386		*	0	0	1
387	94	YEei[Connecticut]			
388		*	1	0	1
389	95	YEei[Florida]			
390		*	0	0	1
391	96	YEei[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

heuristic.cpp

```
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 }

```

distances.csv

```

1 48
2 5
3 2868.1784177465
4 672.8151726255
5 3910.7411305068
6 2077.1452377083
7 1519.5077579655

```

8	1199.4986222794
9	223.7493394677
10	210.4252102536
11	3324.1942219342
12	375.2257089259
13	56.4357710091
14	813.7155779585
15	1044.7398456498
16	165.0340299937
17	540.4773059684
18	1847.1499908138
19	1091.0322829094
20	1703.4547303348
21	212.816842606
22	753.2459183061
23	436.760545128
24	656.2792698469
25	2852.0588142381
26	1153.1140642009
27	3719.6061504499
28	1647.0895439219
29	1286.3997375109
30	2188.3611696487
31	1398.2602236424
32	850.7371193588
33	1599.6208068671
34	372.2093675161
35	1251.1843173346
36	4077.3821775963
37	1051.0725631423
38	1658.9964185501
39	583.4074521691
40	1556.4006060984
41	61.6660433356
42	1275.3164983935
43	2844.2698934059
44	1535.3683688696
45	983.0281549644
46	4058.9796061032
47	520.6289399976
48	346.722732452
49	2055.7517642547
50	2195.3633128392
51	1074.6564829771
52	817.9816362195
53	4368.8983013915
54	4058.8562357351
55	3090.6378199187
56	3078.2955554998
57	652.9574510614
58	2493.1423425089
59	2880.6046484923
60	2056.6878450176
61	1825.8003590143
62	3024.0858609986
63	2327.8933566076
64	4682.3639246028

65	3951.0378063587
66	4550.2158738859
67	3054.42205873
68	2118.4265973449
69	2431.4866187941
70	2212.1368745122
71	547.8271383216
72	1720.8529029881
73	897.0547741451
74	4489.1977165938
75	4142.2868098361
76	683.9429951775
77	4243.9969329672
78	3717.1929493347
79	1335.176983076
80	3228.9139729094
81	1617.4340416483
82	1348.7170068791
83	3907.1834067229
84	4507.7632004432
85	3451.3284563862
86	1342.8449481975
87	2811.7489989948
88	1594.7492824989
89	304.6040016489
90	4372.3388621472
91	3846.5508340495
92	1383.7138875033
93	3382.3197560766
94	2522.0070767663
95	851.0210240327
96	3240.5930582521
97	1408.1193462358
98	2183.7128516647
99	1867.8921975774
100	895.5322444057
101	882.9858748922
102	2661.6694918504
103	297.9545644178
104	686.9378827219
105	148.5196443215
106	374.724802599
107	830.0793441766
108	133.1107231847
109	2505.9449694443
110	1759.60566892
111	2366.8704387846
112	863.818072177
113	100.1013188851
114	236.2140420465
115	23.0538798924
116	2196.2325277927
117	485.2828349181
118	3050.1034594321
119	2308.321592842
120	1953.2153440986
121	1516.0107427025

122	2060.4684236256
123	1522.5705106878
124	949.2174855858
125	1036.6914655118
126	578.6759461004
127	3417.8319636982
128	1717.5448495975
129	2323.2052980925
130	1256.0544480086
131	894.8381828103
132	616.765049642
133	603.6759142219
134	2176.7297626343
135	2193.9499122674
136	1653.3308235565
137	3404.6887339159
138	1189.1402961561
139	327.0826055126
140	1388.6180685584
141	1833.6679973809
142	5423.3472152311
143	5108.4715702844
144	4130.5385102755
145	4121.1002845619
146	641.4336791534
147	3538.0528310694
148	3927.4529733155
149	3097.1937543928
150	2866.5128820872
151	4070.3712586562
152	3373.7037732339
153	5741.7610750756
154	5000.1586332735
155	5605.9389092081
156	4104.165847916
157	3157.4969452398
158	3476.3045536175
159	3255.7477674731
160	1122.1693629777
161	2757.6338063377
162	193.7582962079
163	5546.2706377884
164	5193.7171224844
165	1728.3472481221
166	5299.3955074081
167	4761.4435405175
168	2328.5909304151
169	4277.281128777
170	2662.8775648463
171	415.9506567564
172	4958.0687708492
173	5562.7014963965
174	4493.5222460133
175	2358.3271992773
176	3856.5024348272
177	2649.0475568585
178	1070.3124627028

179	5430.5327664793
180	4893.4060165068
181	526.5427934998
182	4429.5152900794
183	3567.4224858431
184	1856.2968125313
185	3591.8183125852
186	3275.4722802973
187	2296.901617194
188	2287.4751061824
189	1259.8310612474
190	1705.0400877621
191	2094.5210942818
192	1263.817123616
193	1033.5395038933
194	2237.2042155199
195	1541.1679976617
196	3913.0019859645
197	3167.0941116809
198	3774.899349793
199	2271.9304862058
200	1323.9084208644
201	1643.3413275397
202	1422.7850433852
203	818.3775515832
204	924.1301791049
205	1642.4800997284
206	3716.0608213891
207	3361.172953584
208	162.6602002741
209	3468.4424433252
210	2927.8065958521
211	517.5119335309
212	2444.4518295122
213	832.8939032589
214	2018.36986971
215	3125.4945554517
216	3731.2998785239
217	2659.8543607334
218	528.5735398543
219	2023.1906755159
220	828.1079325294
221	768.9867917093
222	3601.2023686552
223	3059.9776640214
224	2014.8472999087
225	2596.3207443201
226	1734.4985664447
227	45.0050041801
228	326.585097466
229	1307.5706360011
230	1311.3018619164
231	4843.0912545545
232	1887.3767301518
233	1498.1971567642
234	2328.8500611693
235	2558.3839762036

236	1356.5939196617
237	2050.7018038666
238	336.3808942182
239	433.0479639672
240	184.1299991242
241	1319.9751142956
242	2269.8139753056
243	1948.9384488258
244	2169.3298117922
245	4371.5396371781
246	2668.4185602294
247	5233.3800993119
248	134.8662928399
249	236.1363915436
250	3696.0237152242
251	126.0904353252
252	682.2375452439
253	3119.1068885677
254	1148.5768664668
255	2760.5243315717
256	5596.8571667682
257	468.8687752232
258	139.5483963457
259	947.8423762924
260	3074.8270522217
261	1570.7108590263
262	2775.8463016135
263	4360.4035958162
264	83.8078237662
265	545.1168342183
266	5578.0923308836
267	999.0426116404
268	1857.8014740628
269	3572.0880788388
270	984.4912652411
271	990.0245354913
272	4523.5879331381
273	1570.4770087236
274	1181.0285097404
275	2011.6824057504
276	2242.0495117761
277	1038.2798503708
278	1734.7828127772
279	662.5228532703
280	108.4673752219
281	509.6971423607
282	1004.8450458454
283	1951.9362922725
284	1632.2071044874
285	1852.7773708623
286	4050.1995254668
287	2351.4037170474
288	4917.8315863269
289	460.1475154303
290	90.8579506791
291	3382.9232740122
292	219.6053845926

293	355.9300331087
294	2798.0254120361
295	831.2043839341
296	2446.0556363783
297	5276.0615101782
298	152.8303673399
299	463.9961442837
300	622.884511836
301	2755.8941299552
302	1252.3742017086
303	2464.2749379258
304	4043.3130182772
305	363.4051293214
306	219.0343716871
307	5255.7856620816
308	679.2350698168
309	1541.0711839978
310	3254.7648540675
311	38.5957550036
312	3540.3606598785
313	598.5097189689
314	225.1222104865
315	1034.7935901023
316	1265.9625162574
317	104.5311310143
318	763.6265991019
319	1638.5668678011
320	876.1955743307
321	1491.6992377109
322	136.1923797168
323	973.724289722
324	659.8444602257
325	878.5058989683
326	3065.8801241031
327	1373.4844964715
328	3939.0469543248
329	1437.2215322092
330	1072.7653827284
331	2410.0592273679
332	1188.4448548723
333	632.0274290731
334	1813.9809891673
335	183.2456780818
336	1473.2915434574
337	4291.9009521456
338	838.7886200665
339	1446.6960453081
340	363.3706172477
341	1774.050971501
342	283.2508131238
343	1498.5976808441
344	3062.9322351388
345	1327.9596767817
346	766.387623655
347	4271.3003370573
348	311.9695714787
349	570.242171347

350	2274.6933479968
351	3533.5918604076
352	585.155587498
353	202.8556213887
354	1024.1363317816
355	1255.1648255148
356	69.9808395963
357	750.4161575914
358	1640.593395648
359	881.5695314583
360	1495.3797230559
361	97.6445067398
362	963.6084721415
363	646.813876975
364	866.6222804588
365	3060.4905013792
366	1363.4933045664
367	3929.9121733828
368	1439.8521147739
369	1077.435006026
370	2398.7252011568
371	1190.9516272504
372	640.3467793485
373	1808.1542168378
374	171.8366546745
375	1461.4943960597
376	4286.167198558
377	842.5052564178
378	1450.6774025461
379	373.0735275944
380	1766.1916264885
381	266.9364134244
382	1484.8887071445
383	3054.4071885736
384	1329.2917908134
385	773.0151374319
386	4266.7925471723
387	312.5151298808
388	556.3693201117
389	2265.9275221027
390	2956.2846384657
391	3345.0116729509
392	2515.1599633646
393	2287.3268486904
394	3486.5263497804
395	2794.1719644534
396	5167.390896967
397	4415.123611807
398	5026.7659657486
399	3523.8128423301
400	2573.5617605511
401	2894.9252956937
402	2675.0775123376
403	487.2930359531
404	2176.3995070289
405	454.6092109608
406	4969.1072315811

407	4610.5747761482
408	1183.2984790398
409	4720.7649828841
410	4172.3764079138
411	1727.0367690408
412	3694.5147679004
413	2091.1899670821
414	758.6048796501
415	4375.1416961574
416	4982.6369260547
417	3903.65064933
418	1768.3006914047
419	3272.3837865658
420	2087.8439800333
421	500.1025968152
422	4855.3262658782
423	4306.3939442963
424	763.3098168329
425	3844.7840206964
426	2985.9374181776
427	1274.9443436159
428	389.481135604
429	441.4810313766
430	671.5727989649
431	532.335335561
432	165.2707426363
433	2211.2476202183
434	1462.1300978077
435	2070.8052340606
436	567.5575885725
437	382.7718845964
438	61.7714968431
439	282.306431651
440	2487.90041838
441	781.135314856
442	3347.3546501761
443	2012.8379438493
444	1656.1645328456
445	1813.9583123861
446	1764.606107622
447	1224.618271866
448	1236.9803426437
449	739.4159944139
450	876.6251934408
451	3711.3963033762
452	1420.4899404651
453	2026.9130233531
454	958.1915857917
455	1188.1091654373
456	318.8130599424
457	900.125209431
458	2473.2197416198
459	1899.1564085869
460	1355.5382649841
461	3696.1113229597
462	891.4712096441
463	29.7180307086

464	1684.7779823557
465	830.8772505799
466	1061.0350292791
467	143.497951839
468	553.8484737606
469	1823.2832837846
470	1072.7057367457
471	1681.7731788497
472	179.9308640898
473	771.6207257908
474	451.1897487024
475	671.7599441957
476	2875.0599803171
477	1170.5793197189
478	3736.8148540304
479	1624.2476539696
480	1266.6983391328
481	2202.6384231136
482	1375.7744028854
483	836.6247985918
484	1623.0026760455
485	349.9389059956
486	1265.4073023591
487	4099.4692194838
488	1031.0284320605
489	1637.7443070971
490	572.3116898175
491	1576.711160495
492	74.9585239397
493	1286.1564220596
494	2862.6544014988
495	1511.2183796248
496	966.5011285539
497	4082.7924067328
498	502.2277157448
499	360.0448799892
500	2074.1743107299
501	231.169488889
502	973.4030306081
503	279.2936558186
504	2652.2309619476
505	1903.2917527517
506	2512.2461498969
507	1008.9530166804
508	62.5107939947
509	379.9233150564
510	160.0357725889
511	2048.4935134586
512	339.7241165877
513	2906.2658730847
514	2454.1284109851
515	2097.5686264907
516	1375.3193217172
517	2205.9971293631
518	1664.4375519392
519	800.7258034044
520	1180.8069906494

521	439.5980687214
522	3270.8116344217
523	1861.9028822658
524	2468.3838026859
525	1396.960581787
526	747.4555075785
527	759.3747289989
528	473.5820806274
529	2031.777532267
530	2340.1680234936
531	1796.2219955592
532	3256.9420305565
533	1332.5061461735
534	471.0490128308
535	1243.3060614656
536	1203.8626721023
537	507.682524458
538	2880.575536141
539	2133.7016825642
540	2741.5796964907
541	1238.4331042097
542	292.6355375731
543	609.8458434683
544	389.275098876
545	1823.8933143728
546	111.5837197317
547	2675.7827507129
548	2683.042000756
549	2327.6403801501
550	1144.1664996257
551	2435.1884600521
552	1895.4052420326
553	584.881261015
554	1410.9462826956
555	210.0535847952
556	3043.9326511078
557	2091.9619398209
558	2697.886031386
559	1628.056700845
560	522.6636871945
561	990.0138719682
562	255.2872207111
563	1802.0272884529
564	2568.6069811184
565	2026.9332775286
566	3032.1431652339
567	1563.0069117362
568	700.9926873203
569	1013.9006060591
570	697.037498525
571	1683.1245155173
572	929.8906507309
573	1540.3735639218
574	58.3235510656
575	913.7400968134
576	594.0971288437
577	814.6295959333

578	3015.4610295407
579	1313.1238672019
580	3879.6062770542
581	1483.4548521078
582	1124.3773390026
583	2345.9510142482
584	1234.7470993842
585	693.1291139865
586	1763.0956285687
587	208.1086571519
588	1408.6664146538
589	4240.3988217295
590	888.7786789787
591	1496.1331529255
592	429.2225297998
593	1718.3247208279
594	214.1928823183
595	1429.6543400527
596	3005.0784138733
597	1371.20676542
598	823.2515173718
599	4222.7558029258
600	359.1443525251
601	503.0308342742
602	2216.5423624723
603	2373.2798868679
604	1626.4993476995
605	2233.9087831167
606	730.7640664565
607	223.5442431313
608	103.7851766124
609	119.2687749474
610	2327.7558272547
611	617.9587449396
612	3183.204812454
613	2175.473004539
614	1820.1153509532
615	1648.9548448082
616	1927.5375700966
617	1389.6886758486
618	1078.9637993687
619	903.5812348985
620	711.6387032568
621	3550.036229175
622	1584.4431528927
623	2190.2035444095
624	1123.4350998294
625	1026.7307351433
626	483.9262416775
627	734.9738304061
628	2309.7083848376
629	2061.2505683992
630	1520.2838596126
631	3536.1943180551
632	1056.0647226128
633	194.1580213618
634	1521.5012656032

635	767.8684617417
636	160.5076110545
637	1643.9370356833
638	2594.019469641
639	2272.5192193032
640	2492.3267241041
641	4698.328100307
642	2991.2007718508
643	5552.6863442492
644	202.4687531939
645	571.7647331991
646	4013.4760160411
647	450.2566813654
648	1017.4236025976
649	3446.1156449231
650	1475.099967874
651	3080.4764270314
652	5922.5909676521
653	800.6843545261
654	206.0530874934
655	1281.2701632745
656	3399.352878525
657	1896.6673828135
658	3092.7448008842
659	4681.9729687585
660	312.0951690811
661	880.2767736178
662	5905.8595121799
663	1328.1053758134
664	2181.5703607129
665	3894.4040340351
666	616.8021176371
667	896.7951747789
668	1843.5112941506
669	1523.8675011163
670	1744.4349000458
671	3941.8064150552
672	2243.0097489814
673	4809.477229164
674	565.4162340198
675	197.0634806598
676	3274.7986326508
677	320.0344370268
678	249.5716813809
679	2689.6039700852
680	722.9489215673
681	2337.8477401705
682	5167.6411313368
683	53.2523892211
684	571.3459021288
685	515.0580599383
686	2647.4290553141
687	1143.960988262
688	2356.3792417895
689	3934.8884180972
690	463.8039472044
691	112.4909042176

692	5147.4929537378
693	570.8107881781
694	1432.7396844436
695	3146.3387129357
696	1503.2982966733
697	2453.3488410221
698	2132.3228487991
699	2352.6352793385
700	4555.5111369764
701	2851.7381229516
702	5416.1573221222
703	67.8110729611
704	419.7442395506
705	3878.2917016873
706	306.543431971
707	865.6174194468
708	3303.073443869
709	1332.3101180732
710	2943.2741077994
711	5780.7348596648
712	652.9903433858
713	46.2878950273
714	1131.7755561843
715	3258.4725755097
716	1754.4405611046
717	2957.8748595168
718	4543.6001779525
719	185.633800316
720	728.6081071817
721	5762.1568050529
722	1183.0689232982
723	2041.206626199
724	3755.3855809483
725	950.2518744619
726	629.0312394577
727	849.3580143158
728	3054.5555533202
729	1348.4550036042
730	3913.8465430824
731	1445.2949012553
732	1089.559884518
733	2378.0928440715
734	1197.0542751001
735	665.5074097148
736	1802.6699501027
737	175.0821143151
738	1441.4044309292
739	4278.6641342248
740	853.9374389528
741	1459.4925038196
742	407.9528949802
743	1755.5696431931
744	254.4237628364
745	1459.6874619606
746	3040.4347056053
747	1331.8453043085
748	792.3675356717

749	4262.4243213185
750	328.9884434001
751	537.9339569903
752	2252.1196744734
753	321.6160602172
754	105.6222606217
755	2105.4209565759
756	399.893794525
757	2966.3821915595
758	2395.5455002396
759	2038.0990233268
760	1436.6924854143
761	2147.237953059
762	1603.9551769325
763	855.6255948244
764	1121.4238749952
765	501.6858046338
766	3328.6378295137
767	1802.4699585027
768	2409.3601813706
769	1336.2217471432
770	805.3409421741
771	699.5778172419
772	536.0563029715
773	2091.3772938803
774	2281.9280455849
775	1736.1062787605
776	3313.7567282224
777	1272.7021492122
778	412.4532251344
779	1302.8285653103
780	220.5709901648
781	2426.993830121
782	719.5073604943
783	3285.6277288001
784	2074.2499574502
785	1717.8417687898
786	1752.2021652153
787	1826.0809276427
788	1286.3731040278
789	1176.5182438411
790	801.1112663981
791	814.8717880654
792	3650.2159362217
793	1482.1647534935
794	2088.4694504286
795	1019.8701947055
796	1126.8525729122
797	380.5715510538
798	838.7585138974
799	2411.5823578352
800	1960.4362965718
801	1417.3077923961
802	3635.285808722
803	953.2283209688
804	91.1609926341
805	1623.1709042559

806	2208.4879869009
807	499.1028840943
808	3065.056752232
809	2294.3621601802
810	1938.3888453189
811	1532.1039812192
812	2046.322523178
813	1506.5892626106
814	959.9757226715
815	1021.6751085877
816	594.9139921347
817	3430.8291063504
818	1702.710654501
819	2308.8499635565
820	1239.6754295852
821	907.4890402421
822	600.9120500817
823	621.8663327598
824	2191.1064429048
825	2180.2777195339
826	1637.7877675223
827	3416.9280268589
828	1173.7709523569
829	311.7189791788
830	1402.7617674801
831	1712.3663184148
832	930.4037643452
833	4498.915877958
834	4137.9169614643
835	778.8634058313
836	4250.1834826544
837	3697.4426676846
838	1252.4380446168
839	3223.5669511981
840	1634.7558462701
841	1226.2576639681
842	3902.9176798551
843	4511.0024206422
844	3428.7135254682
845	1301.5395840847
846	2801.698040862
847	1639.4078661613
848	243.2531535814
849	4386.2752739595
850	3832.2663842614
851	1208.4582877292
852	3372.5174482027
853	2517.6149260905
854	825.0358672788
855	2566.6038973724
856	2793.4195906141
857	2437.2771341865
858	1037.9825152409
859	2545.4201729265
860	2003.8394291937
861	474.7849031363
862	1520.5152681335

863	120.3451456535
864	2932.7609056971
865	2201.6076922963
866	2807.9426115695
867	1736.1044740002
868	411.0801762494
869	1099.0727667552
870	189.1172901891
871	1692.0844874237
872	2679.1881123267
873	2135.8533244357
874	2920.6550499285
875	1672.2169444549
876	810.6622253275
877	903.6704471313
878	5356.7558815453
879	5003.3187856131
880	1540.0665343415
881	5109.6231908554
882	4570.2192111341
883	2135.5055895601
884	4086.7097455554
885	2472.885150276
886	493.9501178398
887	4767.6478385072
888	5372.788894195
889	4302.1483322109
890	2166.158375195
891	3665.6375453522
892	2460.367237201
893	877.4597765199
894	5241.2681880934
895	4702.4563157669
896	573.330325229
897	4238.740690071
898	3376.7701856004
899	1664.4451060252
900	369.3489898707
901	3818.3016543695
902	248.900964195
903	814.9623482503
904	3246.5353380534
905	1275.3494629699
906	2883.9968930543
907	5723.6920390394
908	598.7798311973
909	51.2914930258
910	1079.1351658443
911	3200.8454932036
912	1697.2968713496
913	2897.6930347667
914	4484.9168727043
915	119.2391991171
916	677.8123064829
917	5706.0669243994
918	1127.3429197487
919	1983.1969931458

920	3696.9396504151
921	3467.4110409999
922	130.9421795857
923	446.101424982
924	2885.5872851763
925	916.7621729832
926	2530.952642539
927	5363.5836377701
928	235.6784834247
929	374.2933098
930	712.0976546697
931	2842.5705040807
932	1338.5666527979
933	2548.0800892602
934	4129.3438554208
935	273.7024189044
936	308.9840202125
937	5343.933168585
938	765.7907643961
939	1626.6817326129
940	3340.8409913293
941	3571.7719691482
942	3038.5752692426
943	664.8953747022
944	2551.9348302484
945	937.3348048837
946	1939.4954580812
947	3231.9048617405
948	3835.2627003426
949	2771.7680009407
950	661.6960114503
951	2132.7713385247
952	920.731247187
953	683.6387989442
954	3702.3582288231
955	3169.1382682304
956	1945.5337835715
957	2704.8369181275
958	1843.0757913125
959	206.8050914407
960	569.1460883434
961	2997.7766911149
962	1026.6389913278
963	2636.7427982764
964	5475.1018050696
965	350.5186884526
966	263.7947377007
967	831.2322832023
968	2952.4682417756
969	1448.6747240025
970	2651.3955807321
971	4237.1870529177
972	144.0634265889
973	432.4075059967
974	5457.2374705567
975	878.4419663848
976	1734.9879793142

977	3449.0367090338
978	2445.8504037139
979	490.4519620209
980	2101.2434107296
981	4923.5876682437
982	222.6974900364
983	819.9231466122
984	268.7588024299
985	2405.8963832824
986	905.8056216911
987	2122.7724913352
988	3694.5174936524
989	712.4827865342
990	137.1597353641
991	4902.0720672582
992	336.5816027834
993	1195.5366671591
994	2906.1100058434
995	1971.1924339354
996	428.7161668556
997	2478.0371628795
998	2650.5166646187
999	3258.5757461724
1000	2177.0919490755
1001	103.0416812049
1002	1549.4515945053
1003	468.8738465671
1004	1258.3351410831
1005	3134.1053102328
1006	2580.316322758
1007	2459.7935853126
1008	2120.0970432893
1009	1266.6925062661
1010	484.5904327876
1011	1614.9057821321
1012	4448.4942895817
1013	681.0965068337
1014	1288.122168727
1015	236.8828079578
1016	1926.263450401
1017	422.1362542352
1018	1634.1692469108
1019	3212.5837350948
1020	1163.2366425069
1021	617.6497835515
1022	4430.8205985406
1023	153.9065557433
1024	709.9564856443
1025	2424.0906240947
1026	2849.1418460569
1027	2295.3417192438
1028	2899.9156789349
1029	1834.5622231484
1030	346.3601944895
1031	1195.4381540135
1032	80.7169531074
1033	1601.8017068722

1034	2768.8026478099
1035	2231.8967027187
1036	2841.1037968378
1037	1767.6328646177
1038	905.7446287748
1039	816.2532946757
1040	5128.4229427837
1041	5736.3636277762
1042	4654.8433805327
1043	2523.4001277993
1044	4026.4272072051
1045	2846.4269243584
1046	1256.0035384836
1047	5610.4959989871
1048	5058.2804499141
1049	128.617421435
1050	4597.9719227079
1051	3741.0862755944
1052	2032.9424868198
1053	608.1798790638
1054	480.7343126948
1055	2607.0504742968
1056	1102.9715598076
1057	2312.7889064307
1058	3893.6800813886
1059	492.3875711093
1060	93.6169876793
1061	5109.2852822689
1062	530.4533321873
1063	1391.0045115489
1064	3105.1850414348
1065	1086.3909558252
1066	3214.3747680084
1067	1710.2550601512
1068	2914.9883031118
1069	4499.9102720069
1070	152.9260444796
1071	682.9839915535
1072	5717.4645087381
1073	1138.4869113918
1074	1997.3348236797
1075	3711.610654282
1076	2137.3984395622
1077	639.6564051975
1078	1857.6351317037
1079	3426.198000522
1080	972.8633499281
1081	404.0641696743
1082	4633.5240215646
1083	98.6271340175
1084	929.3347201011
1085	2637.8822798
1086	1504.1531760481
1087	376.7291991654
1088	1289.2183281474
1089	3087.2634917531
1090	2539.2051408378

1091	2509.9526632337
1092	2076.8270538426
1093	1217.7830319968
1094	502.4549081661
1095	1218.0214439773
1096	2790.9440944807
1097	1584.6545267825
1098	1036.9194056848
1099	4009.2247575983
1100	573.1532159957
1101	289.7702141855
1102	2002.3999234775
1103	1594.3446846904
1104	2781.6527316345
1105	2251.8126698752
1106	2842.1578583073
1107	1787.616771576
1108	928.6275270096
1109	815.6519748167
1110	4370.1516054191
1111	3827.2972416413
1112	1263.2188187674
1113	3364.0794007967
1114	2502.7405548462
1115	788.5509102707
1116	576.0257628302
1117	5593.8749025832
1118	1016.9258643426
1119	1869.4804778997
1120	3582.4679541654
1121	5037.4429747477
1122	464.325078767
1123	1326.2729461281
1124	3038.7730259976
1125	4579.1020807542
1126	3725.8286081701
1127	2026.64005841
1128	862.1394981692
1129	2575.5304987784
1130	1714.321397239
1131	2
1132	8
1133	35

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