2020

Su mažiau degalų ir laiko į žaliąją ekonomiką: „keliaujančio pardavėjo“ problema ir jos sprendimas

Raimondas Kuodis

LB, VU

[Versija 0.9]

Turinys

TSP trumpai

TSP (traveling salesman problem) **esmė**:

- jums reikia aplankyti *n* taškų...;
- ...pradedant iš bazinio taško n_0;
- tą reikia padaryti trumpiausiu maršrutu;
- maršruto gale grįžti į bazę n_0.

Pavyzdžiui, kaip važiuotumėt iš, tarkim, Alytaus, kad aplankytumėt visus Lietuvos rajonų centrus?

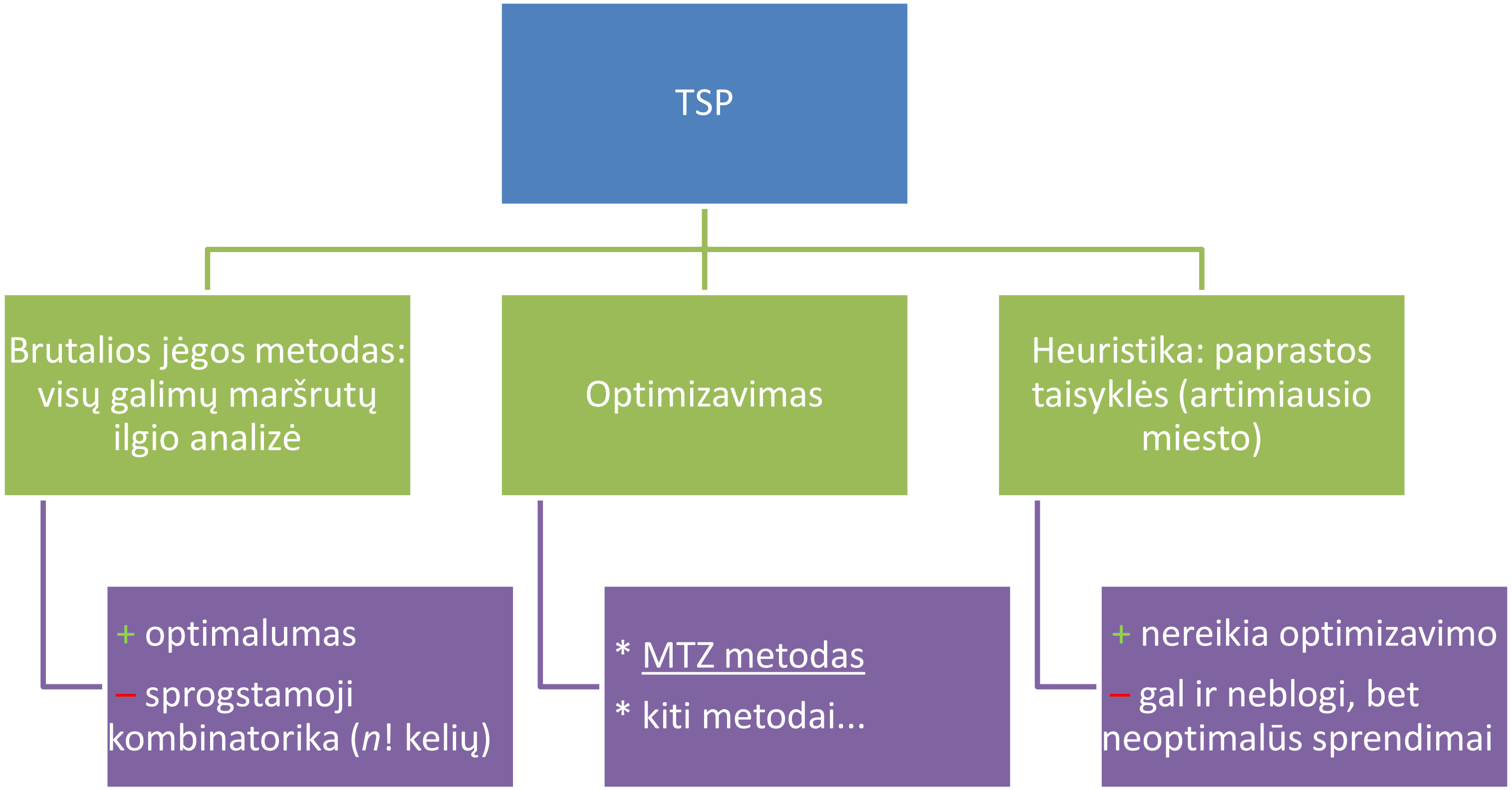
TSP modelis jums sudėlios optimalų maršrutą.

Problemos **mastas**: ši problema iškyla daugeliui verslų (ypač transporte, kaip paštas, išvežiojamoji prekyba etc.) ir žmonių, ir iškyla nuolat. Pasikartojantys procesai tiesiog prašosi būti optimizuoti.

TSP istorija

...

TSP sprendimo būdų taksonomija



1	1
2	2
3	6
4	24
5	120
6	720
7	5040
8	40320
9	362880
10	3628800

Lentelė 1. *n* faktorialas (*n*!), Excel funkcija =fact(n).

Matematinis optimizavimo modelis

Aibės.

Tarkim, turim aibę *iv* miestų – Lietuvos rajonų centrų.

Aibės *iv* poaibis *i* – miestai, kuriuos norime aplankyt (nebūtinai visus *iv*).

Aibė *j* yra dubliuojanti aibę *i*.

Parametrai.

d_{ij} – atstumas tarp miesto *i* ir *j*

Kintamieji.

$$x_{ij} = \begin{cases} 1, \text{jei vykstam iš } i \text{ į } j \\ 0, \text{ priešingu atveju} \end{cases}$$

dtot – bendras viso turo ilgis (km)

Obj – tikslo funkcijos kintamasis

Lygtys.

Tikslo funkcija:

min *Obj* = *dtot*

kur:

dtot – visas turo ilgis (km)

$$dtot = \sum_{i=0}^{iv} \sum_{j=0}^{iv} d_{ij} x_{ij}$$

Apribojimai:

Iš miesto *i* vykstama tik į vieną kitą miestą *j*:

$$\sum_{i=0}^{n} x_{ij} = 1$$

Iš miesto *j* atvykstama tik į vieną kitą miestą *i*:

$$\sum_{j=0}^{n} x_{ij} = 1$$

Nevažiuojam iš savęs į save:

$$x_{ii} = 0.$$

Sub-turų eliminavimas: MTZ metodas

Šių lygčių neužtenka – modelis gali duoti optimalų sprendinį, kuris apims vadinamuosius sub-turus:

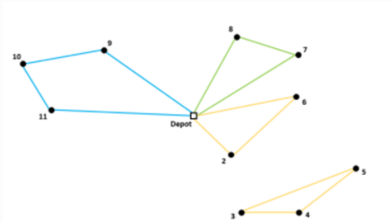
- visi norimi miestai bus „aplankyti“,
- bet trajektorija bus su trūkiais, t.y. atsiras sub-turai (vietoj vieno „rato“ keli mažesni „ratukai“).

Mūsų gi tikslas – turėti trajektoriją, „nupieštą nepakeliant rankos“.

Vienas sprendimų: Miller-Tucker-Zemlin metodas[1].

[1] C.E. Miller, A.W. Tucker, R.A. Zemlin (1960), Integer Programming Formulation of Traveling Salesman Problems, *Journal of the ACM*.

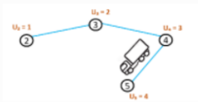
The library in AIMMS that solves a **Capacitated Vehicle Routing Problem** (**CVRP**) contains four different formulation options. The formulations have different methods of eliminating subtours. In this article the Miller-Tucker-Zemlin formulation is discussed. This is an example of a subtour in a route for a CVRP:



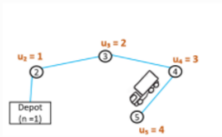
Idea behind the formulation

The Miller-Tucker-Zemlin (MTZ) formulation uses an extra variable. The variable is called u_i and gets a value for each node, except for the depot. If a vehicle drives from node i to node j, the value of u_j has to be bigger than the value of u_i.

So each time a new node is being visited, the value for u_i increases.



The node that the vehicle will visit after node 5, should again have a larger value of u_i. It would not be possible to go from node 5 to node 2, because that node already has a lower value of u_i. This ensures that a vehicle will not drive in a circle. Since that would make it impossible for every value of u_i to be larger than the previous one. Since the depot does not get a value of u_i, it is possible to drive in a circle if the vehicle starts and ends at the depot.

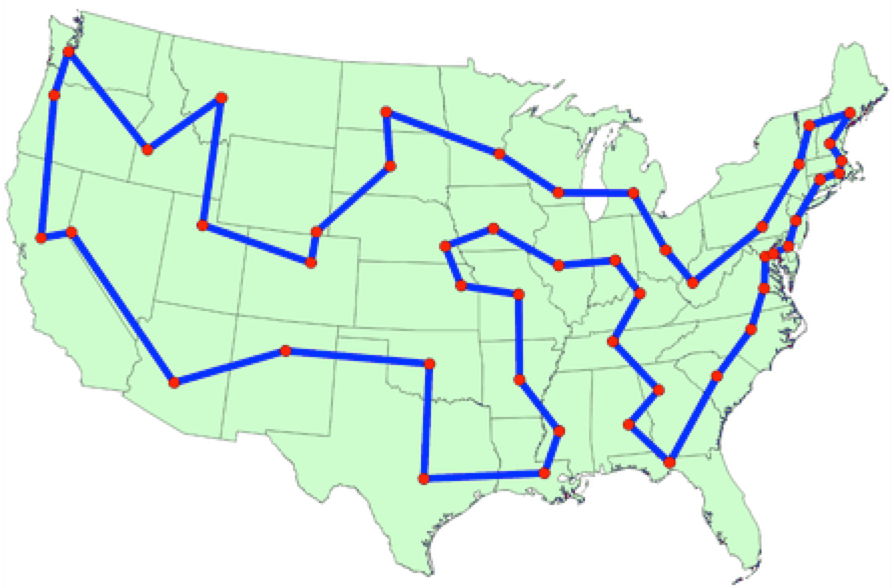


The vehicle can now drive from node 5 back to the depot and u_j is always larger than u_i. So the only circles permitted to be driven are the ones passing the depot. All the other circles would be subtours and are eliminated by this formulation.

https://how-to.aimms.com/Articles/332/332-Miller-Tucker-Zemlin-formulation.html

Praktiniai TSP sprendimai

JAV valstijos. Kaip aplankyt visas JAV valstijas (be Havajų ir Aliaskos)? Ar be modelio važiuotumėt taip, kaip parodyta šiame paveiksle?



https://optimization.mccormick.northwestern.edu/index.php/Traveling_salesman_problems

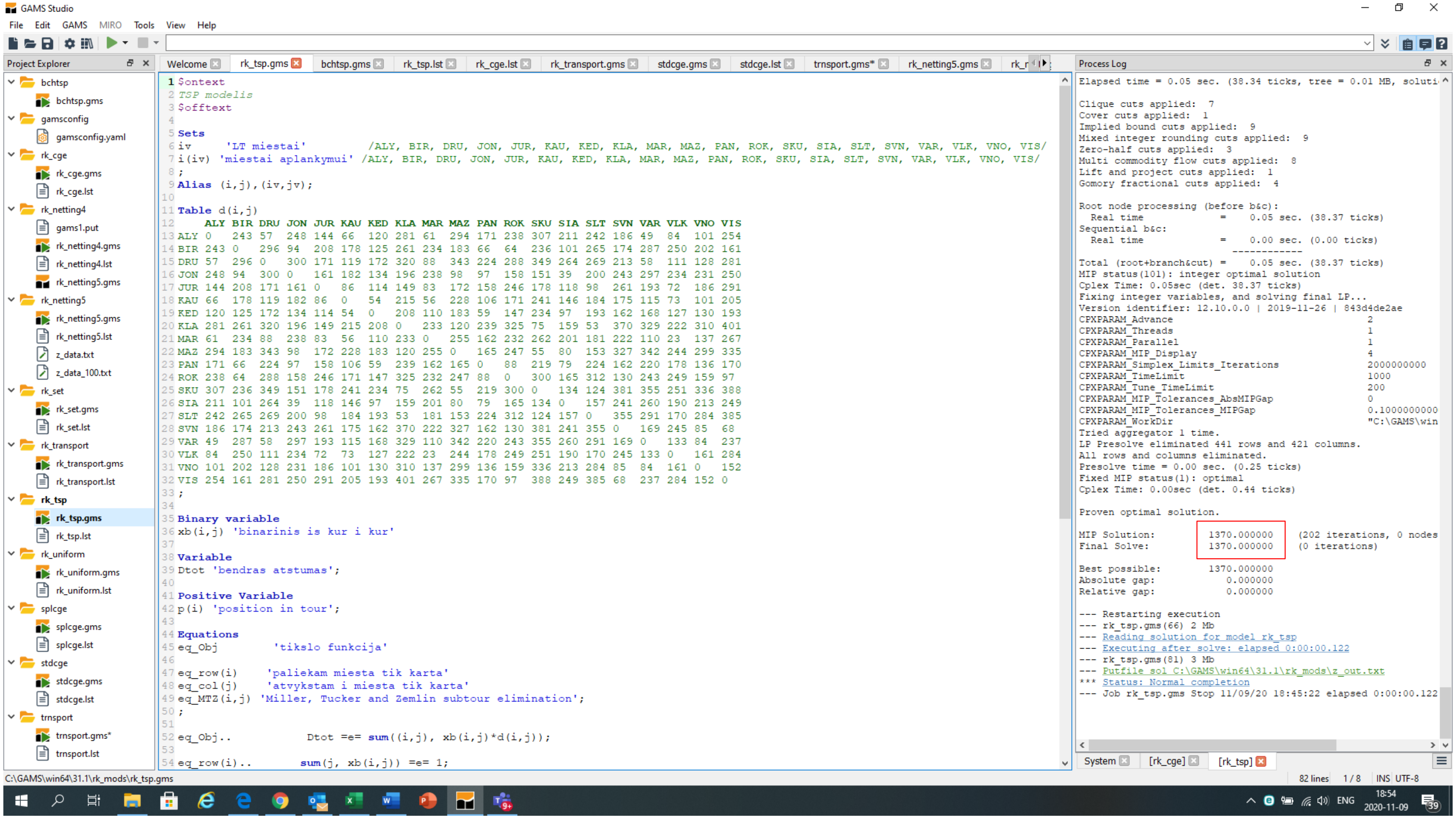
Lietuva.

Pagūglinus „atstumai tarp Lietuvos miestų“, galima rast tokią ar panašią lentelę:

	ALY	BIR	DRU	JON	JUR	KAU	KED	KLA	MAR	MAZ	PAN	ROK	SKU	SIA	SLT	SVN	VAR	VLK	VNO	VIS
ALY	0	243	57	248	144	66	120	281	61	294	171	238	307	211	242	186	49	84	101	254
BIR	243	0	296	94	208	178	125	261	234	183	66	64	236	101	265	174	287	250	202	161
DRU	57	296	0	300	171	119	172	320	88	343	224	288	349	264	269	213	58	111	128	281
JON	248	94	300	0	161	182	134	196	238	98	97	158	151	39	200	243	297	234	231	250
JUR	144	208	171	161	0	86	114	149	83	172	158	246	178	118	98	261	193	72	186	291
KAU	66	178	119	182	86	0	54	215	56	228	106	171	241	146	184	175	115	73	101	205
KED	120	125	172	134	114	54	0	208	110	183	59	147	234	97	193	162	168	127	130	193
KLA	281	261	320	196	149	215	208	0	233	120	239	325	75	159	53	370	329	222	310	401
MAR	61	234	88	238	83	56	110	233	0	255	162	232	262	201	181	222	110	23	137	267
MAZ	294	183	343	98	172	228	183	120	255	0	165	247	55	80	153	327	342	244	299	335
PAN	171	66	224	97	158	106	59	239	162	165	0	88	219	79	224	162	220	178	136	170
ROK	238	64	288	158	246	171	147	325	232	247	88	0	300	165	312	130	243	249	159	97
SKU	307	236	349	151	178	241	234	75	262	55	219	300	0	134	124	381	355	251	336	388
SIA	211	101	264	39	118	146	97	159	201	80	79	165	134	0	157	241	260	190	213	249
SLT	242	265	269	200	98	184	193	53	181	153	224	312	124	157	0	355	291	170	284	385
SVN	186	174	213	243	261	175	162	370	222	327	162	130	381	241	355	0	169	245	85	68
VAR	49	287	58	297	193	115	168	329	110	342	220	243	355	260	291	169	0	133	84	237
VLK	84	250	111	234	72	73	127	222	23	244	178	249	251	190	170	245	133	0	161	284
VNO	101	202	128	231	186	101	130	310	137	299	136	159	336	213	284	85	84	161	0	152
VIS	254	161	281	250	291	205	193	401	267	335	170	97	388	249	385	68	237	284	152	0

Kokiu trumpiausiu atstumu aplankytumėt šiuos 20 miestų?

Ir šiame kelyje įveiksit 1370 km.


GAMS Studio
File Edit GAMS MIRO Tools View Help
Project Explorer
bchtsp
bchtsp.gms
gamsconfig
gamsconfig.yaml
rk_cge
rk_cge.gms
rk_cge.lst
rk_netting4
gams1.put
rk_netting4.gms
rk_netting4.lst
rk_netting5.gms
rk_netting5
rk_netting5.gms
rk_netting5.lst
z_data.txt
z_data_100.txt
rk_set
rk_set.gms
rk_set.lst
rk_transport
rk_transport.gms
rk_transport.lst
rk_tsp
rk_tsp.gms
rk_tsp.lst
rk_uniform
rk_uniform.gms
rk_uniform.lst
splcge
splcge.gms
splcge.lst
stdcge
stdcge.gms
stdcge.lst
trnsport
trnsport.gms*
trnsport.lst
Welcome rk_tsp.gms bchtsp.gms rk_tsp.lst rk_cge.lst rk_transport.gms stdcge.gms stdcge.lst trnsport.gms* rk_netting5.gms
$ontext
TSP modelis
$offtext
Sets
iv 'LT miestai' /ALY, BIR, DRU, JON, JUR, KAU, KED, KLA, MAR, MAZ, PAN, ROK, SKU, SIA, SLT, SVN, VAR, VLK, VNO, VIS/
i(iv) 'miestai aplankymui' /ALY, BIR, DRU, JON, JUR, KAU, KED, KLA, MAR, MAZ, PAN, ROK, SKU, SIA, SLT, SVN, VAR, VLK, VNO, VIS/
;
Alias (i,j),(iv,jv);
Table d(i,j)
ALY BIR DRU JON JUR KAU KED KLA MAR MAZ PAN ROK SKU SIA SLT SVN VAR VLK VNO VIS
ALY 0 243 57 248 144 66 120 281 61 294 171 238 307 211 242 186 49 84 101 254
BIR 243 0 296 94 208 178 125 261 234 183 66 64 236 101 265 174 287 250 202 161
DRU 57 296 0 300 171 119 172 320 88 343 224 288 349 264 269 213 58 111 128 281
JON 248 94 300 0 161 182 134 196 238 98 97 158 151 39 200 243 297 234 231 250
JUR 144 208 171 161 0 86 114 149 83 172 158 246 178 118 98 261 193 72 186 291
KAU 66 178 119 182 86 0 54 215 56 228 106 171 241 146 184 175 115 73 101 205
KED 120 125 172 134 114 54 0 208 110 183 59 147 234 97 193 162 168 127 130 193
KLA 281 261 320 196 149 215 208 0 233 120 239 325 75 159 53 370 329 222 310 401
MAR 61 234 88 238 83 56 110 233 0 255 162 232 262 201 181 222 110 23 137 267
MAZ 294 183 343 98 172 228 183 120 255 0 165 247 55 80 153 327 342 244 299 335
PAN 171 66 224 97 158 106 59 239 162 165 0 88 219 79 224 162 220 178 136 170
ROK 238 64 288 158 246 171 147 325 232 247 88 0 300 165 312 130 243 249 159 97
SKU 307 236 349 151 178 241 234 75 262 55 219 300 0 134 124 381 355 251 336 388
SIA 211 101 264 39 118 146 97 159 201 80 79 165 134 0 157 241 260 190 213 249
SLT 242 265 269 200 98 184 193 53 181 153 224 312 124 157 0 355 291 170 284 385
SVN 186 174 213 243 261 175 162 370 222 327 162 130 381 241 355 0 169 245 85 68
VAR 49 287 58 297 193 115 168 329 110 342 220 243 355 260 291 169 0 133 84 237
VLK 84 250 111 234 72 73 127 222 23 244 178 249 251 190 170 245 133 0 161 284
VNO 101 202 128 231 186 101 130 310 137 299 136 159 336 213 284 85 84 161 0 152
VIS 254 161 281 250 291 205 193 401 267 335 170 97 388 249 385 68 237 284 152 0
;
Binary variable
xb(i,j) 'binarinis is kur i kur'
Variable
Dtot 'bendras atstumas';
Positive Variable
p(i) 'position in tour';
Equations
eq_Obj 'tikslo funkcija'
eq_row(i) 'paliekam miesta tik karta'
eq_col(j) 'atvykstam i miesta tik karta'
eq_MTZ(i,j) 'Miller, Tucker and Zemlin subtour elimination';
;
eq_Obj.. Dtot =e= sum((i,j), xb(i,j)*d(i,j));
eq_row(i).. sum(j, xb(i,j)) =e= 1;
Process Log
Elapsed time = 0.05 sec. (38.34 ticks, tree = 0.01 MB, soluti
Clique cuts applied: 7
Cover cuts applied: 1
Implied bound cuts applied: 9
Mixed integer rounding cuts applied: 9
Zero-half cuts applied: 3
Multi commodity flow cuts applied: 8
Lift and project cuts applied: 1
Gomory fractional cuts applied: 4
Root node processing (before b&c):
Real time = 0.05 sec. (38.37 ticks)
Sequential b&c:
Real time = 0.00 sec. (0.00 ticks)

Total (root+branch&cut) = 0.05 sec. (38.37 ticks)
MIP status(101): integer optimal solution
Cplex Time: 0.05sec (det. 38.37 ticks)
Fixing integer variables, and solving final LP...
Version identifier: 12.10.0.0 | 2019-11-26 | 843d4de2ae
CPXPARAM_Advance 2
CPXPARAM_Threads 1
CPXPARAM_Parallel 1
CPXPARAM_MIP_Display 4
CPXPARAM_Simplex_Limits_Iterations 2000000000
CPXPARAM_TimeLimit 1000
CPXPARAM_Tune_TimeLimit 200
CPXPARAM_MIP_Tolerances_AbsMIPGap 0
CPXPARAM_MIP_Tolerances_MIPGap 0.1000000000
CPXPARAM_WorkDir "C:\GAMS\win
Tried aggregator 1 time.
LP Presolve eliminated 441 rows and 421 columns.
All rows and columns eliminated.
Presolve time = 0.00 sec. (0.25 ticks)
Fixed MIP status(1): optimal
Cplex Time: 0.00sec (det. 0.44 ticks)
Proven optimal solution.
MIP Solution: 1370.000000 (202 iterations, 0 nodes
Final Solve: 1370.000000 (0 iterations)
Best possible: 1370.000000
Absolute gap: 0.000000
Relative gap: 0.000000
--- Restarting execution
--- rk_tsp.gms(66) 2 Mb
--- Reading solution for model rk_tsp
--- Executing after solve: elapsed 0:00:00.122
--- rk_tsp.gms(81) 3 Mb
--- Putfile sol C:\GAMS\win64\31.1\rk_mods\z_out.txt
*** Status: Normal completion
--- Job rk_tsp.gms Stop 11/09/20 18:45:22 elapsed 0:00:00.122
System [rk_cge] [rk_tsp]
C:\GAMS\win64\31.1\rk_mods\rk_tsp.gms
82 lines 1 / 8 INS UTF-8
ENG 18:54 2020-11-09