Master in Statistics and Operative Research

LARGE-SCALE OPTIMIZATION

Cutting Plane Algorithm

Víctor Diví i Cuesta

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Facultat de Matemàtiques i Estadística

Contents

1. Introduction ... 1
   1.1. Problem ... 1
   1.2. Scenario ... 1
2. Base Formulation ... 1
3. Cutting Plane ... 4
   3.1. Generalized Linear Problem ... 6
   3.2. Variants ... 7
4. Conclusions ... 8

List of Tables

Table 1 Cutting Plane iterations ... 5
Table 2 Comparison between dual variables ... 7
Table 3 Cutting Plane iterations (Variant 1) ... 7
Table 4 Cutting Plane iterations (Variant 2) ... 8

List of Summaries

Summary 1 Results of the unconstrained model. ... 2
Summary 2 Results of the capacited model. ... 3
Summary 3 Results of the Cutting Plane. ... 6

1. Introduction

1.1. Problem

The Multi-commodity Network Flow Problem must be solved (in scalar notation):

$$
\begin{aligned}
\text{Min} \quad & \sum_{l\in O}\sum_{a\in A} C_a x_a^l \\
\text{s.t.} \quad & \sum_{(i,j)\in A} x_{ij}^l - \sum_{(j,i)\in A} x_{ji}^l = T_i^l && \forall i \in N, \forall l \in O \\
& \sum_{l\in O} x_a^l \le Y_a && \forall a \in A \\
& x_a^l \ge 0 && \forall a \in A, \forall l \in O
\end{aligned}
\tag{P}
$$

Where given an arc a, an origin l and a node i, x_a^l is the flow at a from l, C_a is the unitary cost of a, Y_a is the maximum capacity of a, and T_i^l is the net output flow at i for flow from l

In Section 2, the problem will be solved directly, and later, in Section 3, the Cutting Plane Algorithm will be used to do so, and two additional variants are implemented and discussed.

1.2. Scenario

The scenario used in this report is Scenario 19, with:

$$
\begin{aligned}
& N = \{1, ..., 24\} \\
& O = \{3, 11\} \\
& D = \{5, 8, 12, 13, 14, 17, 18, 19, 21, 23\} \\
& C_{ij}^l = 95 + \left(x_i - x_j\right)^2 + 8\left(y_i - y_j\right)^2 && \forall (i,j) \in A, \forall l \in O \\
& T_i^l = \begin{cases} 500, & \text{if } i = l \\ -50, & \text{if } i \in D \\ 0, & \text{otherwise} \end{cases} && \forall i \in N, \forall l \in O
\end{aligned}
$$

2. Base Formulation

We start by solving the unconstrained version of the problem:

$$
\begin{aligned}
\text{Min} \quad & \sum_{l\in O}\sum_{a\in A} C_a x_a^l \\
\text{s.t.} \quad & \sum_{(i,j)\in A} x_{ij}^l - \sum_{(j,i)\in A} x_{ji}^l = T_i^l && \forall i \in N, \forall l \in O \\
& x_a^l \ge 0 && \forall a \in A, \forall l \in O
\end{aligned}
\tag{U}
$$

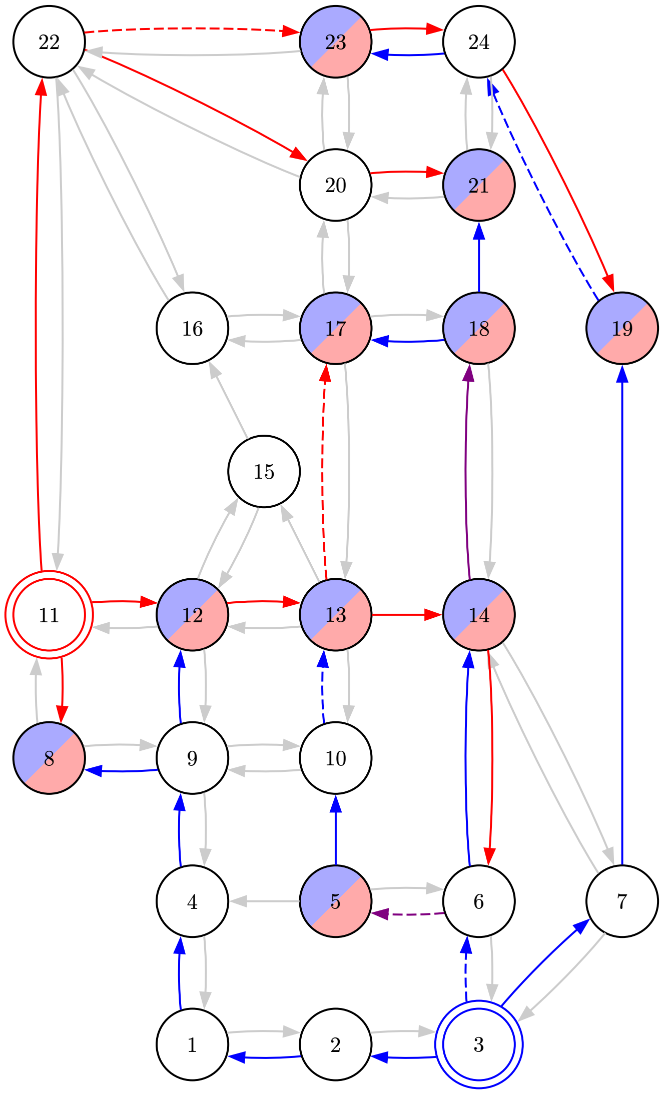
The solution obtained with this version has a total cost of 346240. Its flows are listed in the flow tables of Summary 1 (in parenthesis, the total flow when the link is used by both origins), and which links are used can be seen in the visualization of the network, where edges are painted according to the flow they carry from each origin (blue for origin 3, red for origin 11, and purple for links used by both). Among all the links carrying flow, six has been chosen to be limited by a new cap. The selected links can be identified by dashed edges in the visualization, by orange entries in the flow tables, and are listed, with their new capacities, in the new maximum capacities table in the summary.

Total Cost	346240

Flows for $l = 3$	
Arc	**Flow**
$1 \to 4$	100
$2 \to 1$	100
$3 \to 2$	100
$3 \to 6$	300
$3 \to 7$	100
$4 \to 9$	100
$5 \to 10$	50
$6 \to 5$	100 (150)
$6 \to 14$	200
$7 \to 19$	100
$9 \to 8$	50
$9 \to 12$	50
$10 \to 13$	50
$14 \to 18$	150 (200)
$18 \to 17$	50
$18 \to 21$	50
$19 \to 24$	50
$24 \to 23$	50

Flows for $l = 11$	
Arc	**Flow**
$6 \to 5$	50 (150)
$11 \to 8$	50
$11 \to 12$	300
$11 \to 22$	150
$12 \to 13$	250
$13 \to 14$	150
$13 \to 17$	50
$14 \to 6$	50
$14 \to 18$	50 (200)
$20 \to 21$	50
$22 \to 20$	50
$22 \to 23$	100
$23 \to 24$	50
$24 \to 19$	50

New Maximum Capacities (Y_a)		
Arc	**Total Flow**	**Max Capacity**
$3 \to 6$	300	100
$6 \to 5$	150	100
$10 \to 13$	50	10
$13 \to 17$	50	20
$19 \to 24$	50	30
$22 \to 23$	100	40



Summary 1: Results of the Unconstrained Problem and New Maximum Capacities. Links affected by new caps are represented by dashed edges in the graph and orange rows in the flow tables. Unused links are greyed in the graph and not included in the flow tables.

With the newly added caps, the original (capacited) version of the problem generates a new solution with a total cost of 379245, an increase of 33005 (around 10%) of the uncapacited cost. The new solution is shown in Summary 2. Again, links affected by caps are dashed in the visualization and marked with orange in the flow tables. In the Maximum Capacities table of the summary, we can see how all but one link are at maximum capacity, the only one not being the ($10 \to 13$) link which becomes unused.

Total Cost	379245

Flows for $l = 3$

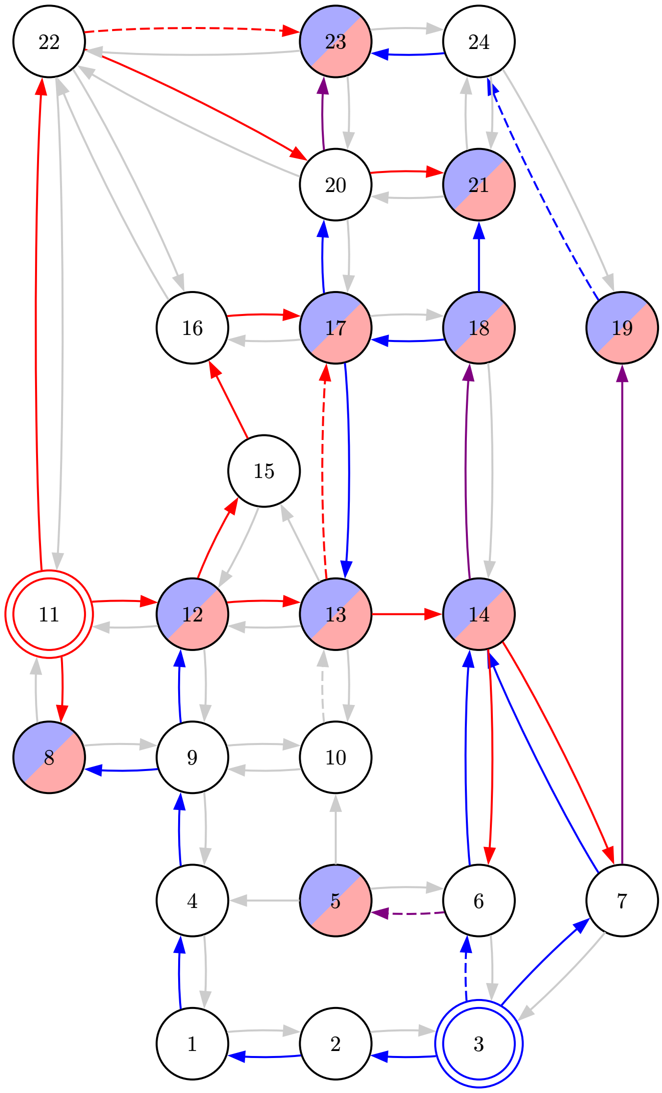
Arc	Flow
1 → 4	100
2 → 1	100
3 → 2	100
3 → 6	100
3 → 7	300
4 → 9	100
6 → 5	50 (100)
6 → 14	50
7 → 14	220
7 → 19	80 (130)
9 → 8	50
9 → 12	50
14 → 18	220 (270)
17 → 13	50
17 → 20	20
18 → 17	120
18 → 21	50
19 → 24	30
20 → 23	20 (30)
24 → 23	30

Flows for $l = 11$

Arc	Flow
6 → 5	50 (100)
7 → 19	50 (130)
11 → 8	50
11 → 12	350
11 → 22	100
12 → 13	270
12 → 15	30
13 → 14	200
13 → 17	20
14 → 6	50
14 → 7	50
14 → 18	50 (270)
15 → 16	30
16 → 17	30
20 → 21	50
20 → 23	10 (30)
22 → 20	60
22 → 23	40

Maximum Capacities (Y_a)

Arc	Total Flow	Max Capacity
3 → 6	100	100
6 → 5	100	100
10 → 13	0	10
13 → 17	20	20
19 → 24	30	30
22 → 23	40	40



Summary 2: Results of the Capacited Problem. Links affected by caps are represented by dashed edges in the graph and orange rows in the flow tables. Unused links are greyed in the graph and not included in the flow tables.

For both (P) and (U), the AMPL formulations and parameters are provided in `base.mod` and `base.dat` respectively. Separate runners are provided in `capacited.run` and `unconstrained.run`. The added caps are listed in `caps.dat`, while flows are listed in `total_flows.csv` (total flows) and `flows.csv` (segregated by origin) in the corresponding folder (`capacited` or `unconstrained`).

3. Cutting Plane

Now, the same problem will be solved using the Dantzig's Cutting Plane Algorithm. The problem that we will approximate is the following:

$$\begin{aligned}
\underset{(z,\mu)}{\text{Min}} \quad & z \\
\text{s.t.} \quad & z \leq \sum_{\substack{a\in A \\ l\in O}} C_a x_a^l - \mu \sum_{a\in A} \left(Y_a - \sum_{l\in O} x_a^l \right) \quad \forall x \in X \\
& \mu \geq 0
\end{aligned} \tag{P_∞}$$

Where $X = \left\{ x \in \mathbb{R}_+^{|A|\times|O|} \mid \sum_{(i,j)\in A} x_{ij}^l - \sum_{(j,i)\in A} x_{ji}^l = T_i^l \ \forall i \in N, \forall l \in O \right\}$.

The algorithm requires an initial feasible solution. To obtain such solution, we add an artificial point to N, namely node 0, and connect it to every origin and every destination. This means adding a total of $|O| + |D|$ arcs to the problem, and a total of $|O| + |O| \times |D|$ (flows of origins to fake node and of fake node to destinations). Then, the initial feasible solution is such that all previously existing arcs are unused, and all the flow goes through the new artificial arcs. Therefore, the initial solution is:

$$x_0 = \left[\overbrace{\mathbb{0}^{|A|\times|O|}}^{\text{Original arcs}}, \overbrace{\left[T_l^l, l \in O\right]}^{\text{Origins to Fake}}, \overbrace{\left[-T_i^l, i \in N, l \in O\right]}^{\text{Fake to Destinations}} \right]$$

From this point on, variables x without subindices or with one subindex refer to this kind of vector variables, while variables x_i^l with a subindex and superindex refer to its elements.

The cost for these new artificial arcs (their C_a) is a value large enough so that using any combination of real arcs is cheaper (Big-M method), so that if the optimal solution obtained uses any of these arcs would mean that no feasible solution exists in the original problem.

With that initial point x_0, we then proceed to solve our Master Problem and SubProblem iteratively until the difference between their objective functions is lower than a given tolerance ε (set to $1e^{-6}$). The Master Problem formulation is as follows:

$$\begin{aligned}
\underset{(z,\mu)}{\text{Max}} \quad & z \\
\text{s.t.} \quad & z \leq \sum_{\substack{a\in A \\ l\in O}} C_a x_a^l - \mu \sum_{a\in A} \left(Y_a - \sum_{l\in O} x_a^l \right) \quad \forall x \in \{x_0, ..., x_k\} \\
& \mu \geq 0
\end{aligned} \tag{MP}$$

Where $\{x_0, ..., x_k\}$ is the set of existing cuts. The SubProblem is defined as follows:

$$\begin{aligned}
\underset{(x)}{\text{Min}} \quad & \sum_{\substack{a\in A \\ l\in O}} C_a x_a^l - \mu \sum_{a\in A} \left(Y_a - \sum_{l\in O} x_a^l \right) \\
\text{s.t.} \quad & x \in X
\end{aligned} \tag{SP}$$

At each iteration k we start solving (MP), which yields a vector of dual variables μ_k. These are used as parameters to solve (SP) which, in turn, yields a new vector x_k, which gets added to the cut set.

This process is repeated until the termination criteria is met (i.e. the difference falls below the accepted tolerance ε). Once finished, the last solution x obtained is not guaranteed to be the optimum solution for (P), it is not even guaranteed to be feasible. To obtain the actual optimum solution we must solve the corresponding Generalized Linear Problem (where α_j are the dual variables of cut number j) :

$$\left.\begin{array}{lll} z_k = \underset{(\alpha)}{\text{Min}} & \sum_{j=0}^{k} \alpha_j f(\hat{x}_j) & \\ \text{s.t.} & \sum_{j=0}^{k} \alpha_j g(\hat{x}_j) \geq 0 & \\ & \sum_{j=0}^{k} \alpha_j h(\hat{x}_j) = 0 & \\ & \sum_{j=0}^{k} \alpha_j = 1 & \\ & \alpha_j \geq 0 & j = 1, ..., k \end{array}\right\} \longrightarrow \quad \tilde{x} = \sum_{j=0}^{k} \alpha^* \hat{x}_j \qquad (GLP)$$

The AMPL implementation, parameters, and runner for both problems is provided in `cutting_plane.mod`, `cutting_plane.dat`, `cutting_plane.run` respectively, with the maximum capacities again in `caps.dat`.

Running our implementation with Scenario 19, we reach the stopping criteria in 17 iterations. Progress of the Master Problem and SubProblem costs (z and w respectively), their gap, the cost of the flow assignation and whether it is feasible or not is shown in Table 1.

Iter	z	w	Gap	Cost	Feas
1	100000000	359750	9.96e+07	359750	×
2	80071950	-19559550	9.96e+07	368500	×
3	66787750	-32830450	9.96e+07	381800	×
4	60146536	-39478271	9.96e+07	372975	×
5	60146008	-39473897	9.96e+07	379200	×
6	40222385	-59398290	9.96e+07	379325	×
7	20300895	-79317485	9.96e+07	374500	×
8	11762120	-87836405	9.96e+07	401475	✓
9	383544	-79263880	7.96e+07	352575	×
10	383544	369045	1.45e+04	362525	×

Iter	z	w	Gap	Cost	Feas
11	383544	370145	1.34e+04	388075	×
12	381056	370170	1.09e+04	388675	✓
13	379605	379155	4.50e+02	383125	×
14	379477	378630	8.47e+02	377125	×
15	379365	377805	1.56e+03	348100	×
16	379365	378765	6.00e+02	373650	×
17	379245	379245	0.00e+00	373650	×

Table 1: Evolution of z, w, Absolute Gap, Cost, and Feasibility (Feas) through the iterations of the Cutting Plane Algorithm.

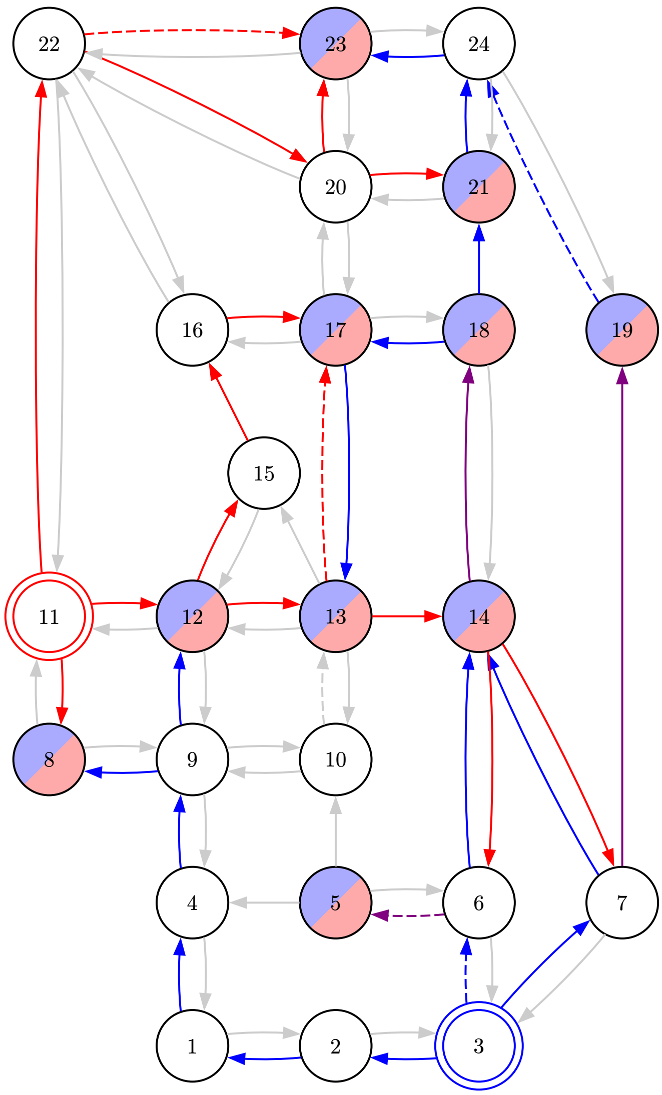
Note how of all the points generated through the iterations, only 2 are feasible, but neither is guaranteed to be the optimal solution. To get it, we will compute an approximation using the dual variables obtained from each cut in the right-hand-side equation in (GLP). The final solution is shown in Summary 3, showing also said dual variables. Note that all of them are positive, and their sum is 1.

Note that while the cost of the final solution equals the one obtained by solving the (P) directly (Summary 2), the actual solution differs slightly. For example, arc $(20 \rightarrow 23)$ is used by both origins in the direct solution, but only by origin 3 in the cutting plane one.

Total Cost	379245
Iterations	17

Flows for $l = 3$	
Arc	**Flow**
$1 \to 4$	100
$2 \to 1$	100
$3 \to 2$	100
$3 \to 6$	100
$3 \to 7$	300
$4 \to 9$	100
$6 \to 5$	50 (100)
$6 \to 14$	50
$7 \to 14$	220
$7 \to 19$	80 (130)
$9 \to 8$	50
$9 \to 12$	50
$14 \to 18$	220 (270)
$17 \to 13$	50
$18 \to 17$	100
$18 \to 21$	70
$19 \to 24$	30
$21 \to 24$	20
$24 \to 23$	50

Flows for $l = 11$	
Arc	**Flow**
$6 \to 5$	50 (100)
$7 \to 19$	50 (130)
$11 \to 8$	50
$11 \to 12$	350
$11 \to 22$	100
$12 \to 13$	270
$12 \to 15$	30
$13 \to 14$	200
$13 \to 17$	20
$14 \to 6$	50
$14 \to 7$	50
$14 \to 18$	50 (270)
$15 \to 16$	30
$16 \to 17$	30
$20 \to 21$	50
$20 \to 23$	10
$22 \to 20$	60
$22 \to 23$	40



Maximum Capacities (Y_a)		
Arc	**Total Flow**	**Max Capacity**
$3 \to 6$	100	100
$6 \to 5$	100	100
$10 \to 13$	0	10
$13 \to 17$	20	20
$19 \to 24$	30	30
$22 \to 23$	40	40

Dual Variables				
Cut	13	14	15	17
μ	0.2	0.2	0.2	0.4

Summary 3: Results of the Cutting Plane. Links affected by caps are represented by dashed edges in the graph and orange rows in the flow tables. Unused links are greyed in the graph and not included in the flow tables.

All data from the iterations table is provided in `iterations.csv`, while the flows are provided as in the basic version (`total_flows.csv` and `flows.csv`), all in the `cutting_plane` folder.

3.1. Generalized Linear Problem

Now, instead of computing an approximation using the dual variables from the cuts, we will compute it solving the corresponding Generalized Linear Problem (GLP) to obtain the optimal dual variables. The AMPL implementation of the code is provided in `dual.mod`, `dual.dat`, and `dual.run` (although it also uses `caps.dat` and `cutting_plane.dat` to read the scenario data).

The cost of the solution obtained for the (GLP) is 379245. Due to space limitations, a summary of the solution to the primal problem generated from the solution of the (GLP) is not shown, but it is identical to the one obtained by the Cutting Plane Algorithm (Summary 3), with the same objective cost of 379245 (the list of total flows is provided in `total_flows_dual.csv`). However, the obtained α^* do differ from the ones used before (see Table 2), although they are still all positive and adding up to 1.

Dual Variables (CP)				
Cut	13	14	15	17
μ	0.2	0.2	0.2	0.4

Dual Variables (GLP)				
Cut	6	14	15	18
μ	0.2	0.4	0.2	0.2

Table 2: Comparison between the dual variables obtained through the Cutting Plane Algorithm (left) and those obtained solving the Generalized Linear Problem (right)

3.2. Variants

To try and improve the convergence rate of the Cutting Plane Algorithm, we are going to explore two variants in which instead of passing directly the dual variables μ_k from the Master Problem to the SubProblem, we apply some smoothing to it:

$$\mu_{k+1} = \hat{\mu}_k + a^k(\mu_k - \hat{\mu}_k)$$

Where $\hat{\mu}_k$ are the dual variables obtained as a result of the Master Problem at iteration k, and μ_k are the values that will be passed to the subproblem at iteration k. The two variants differ in how a^k is computed:

$$1) \quad a^k = \frac{1}{k+1} \qquad\qquad 2) \quad a^k = \frac{k^2}{\sum_{i=1}^{k} i^2}$$

While they both obtain an equivalent optimal solution, with the same objective cost of 379425, the number of iterations for both variants is larger than using the basic version (from 17 to 25 for the Variant 1 and 32 for Variant 2). The evolution of the values at each iteration is shown in Table 3 for Variant 1 and in Table 4 for Variant 2.

Due to size limitations, the summaries of the obtained solutions are not shown here, but are provided as in the basic version in the folders `variant_1` and `variant_2`.

Note how, while during the first 9 iterations the gap evolution is similar in all three cases, in the basic version starts rapidly decreasing (although not consistently, it increases at iteration 15), while for the two variants it is much more steady and consistent, although also much slower.

Iter	z	w	**Gap**	**Cost**	**Feas**
1	100000000	359750	9.96e+07	359750	×
2	80071950	-9595525	8.97e+07	368500	×
3	66787750	-25081041	9.19e+07	381800	×
4	60146536	-35870032	9.60e+07	385775	×
5	60146536	-38737957	9.89e+07	395975	×
6	40225731	-55932737	9.62e+07	401475	✓

Iter	z	w	**Gap**	**Cost**	**Feas**
7	388071	-73076222	7.35e+07	353175	✓
8	388071	-8794965	9.18e+06	353175	✓
9	388071	-632265	1.02e+06	353175	✓
10	388071	286037	1.02e+05	353175	✓
11	388071	370613	1.75e+04	357575	×
12	388071	373763	1.43e+04	384700	×

Iter	z	w	Gap	Cost	Feas
13	387225	374437	1.28e+04	383125	×
14	382135	373726	8.41e+03	382675	×
15	381565	376024	5.54e+03	373650	×
16	379245	371783	7.46e+03	372975	×
17	379245	378509	7.35e+02	377125	×
18	379245	378361	8.83e+02	377625	×
19	379245	379198	4.65e+01	377625	×

Iter	z	w	Gap	Cost	Feas
20	379245	379242	2.32e+00	377625	×
21	379245	379244	1.11e-01	377625	×
22	379245	379244	5.03e-03	377625	×
23	379245	379244	2.19e-04	377625	×
24	379245	379244	9.11e-06	377625	×
25	379245	379244	3.59e-07	377625	×

Table 3: Evolution of z, w, Absolute Gap, Cost, and Feasibility (Feas) through the iterations of the Cutting Plane Algorithm with 1st Variant.

Iter	z	w	Gap	Cost	Feas
1	100000000	359750	9.96e+07	359750	×
2	80071950	-3617110	8.37e+07	368500	×
3	66787750	-14041895	8.08e+07	381800	×
4	60146536	-25905145	8.61e+07	385775	×
5	60146536	-33292696	9.34e+07	395975	×
6	40225731	-49053216	8.93e+07	401475	✓
7	388071	-61703720	6.21e+07	353175	✓
8	388071	-19091706	1.95e+07	353175	✓
9	388071	-5148286	5.54e+06	353175	✓
10	388071	-1049943	1.44e+06	353175	✓
11	388071	44198	3.44e+05	353175	✓
12	388071	311890	7.62e+04	353175	✓
13	388071	366088	2.20e+04	363125	✓
14	388071	376385	1.17e+04	383125	×
15	385340	370245	1.51e+04	379200	×
16	380485	374737	5.75e+03	382675	×
17	380098	372479	7.62e+03	347425	×
18	380098	373473	6.62e+03	372975	×
19	379425	378339	1.09e+03	377625	×

Iter	z	w	Gap	Cost	Feas
20	379365	378694	6.71e+02	347600	×
21	379365	378740	6.25e+02	378550	×
22	379270	378570	7.00e+02	373650	×
23	379245	379162	8.25e+01	373650	×
24	379245	379235	9.69e+00	373650	×
25	379245	379243	1.10e+00	373650	×
26	379245	379244	1.20e-01	373650	×
27	379245	379244	1.26e-02	373650	×
28	379245	379244	1.28e-03	373650	×
29	379245	379244	1.26e-04	373650	×
30	379245	379244	1.20e-05	373650	×
31	379245	379244	1.10e-06	373650	×
32	379245	379244	1.01e-07	373650	×

Table 4: Evolution of z, w, Absolute Gap, Cost, and Feasibility (Feas) through the iterations of the Cutting Plane Algorithm with 2nd Variant.

4. Conclusions

In this report we have solved a scenario of the Multi-Commodity Network Design Problem using two different methods: solving directly the classical LP formulation and using the Dantzig's Cutting Plane Algorithm.

Moreover, the corresponding Generalized Linear Problem has also been solved to find the optimal solution for the primal problem, and two variants of the Cutting Plane algorithm have been implemented and compared with the basic version.

Experimental results show that all 5 methods tried an optimal solution, although the actual flows within each solution differ from each other.