Strong Formulations for the Survivable Network Design with Hop Constraints Problem

A. Ridha Mahjoub[1], Luidi Simonetti[2], Eduardo Uchoa[2]

[1]Université Paris-Dauphine
mahjoub@lamsade.dauphine.fr

[2]Universidade Federal Fluminense
luidi@ic.uff.br
uchoa@producao.uff.br

January, 2011

The Survivable Network Design with Hop Constraints (SNDH) Problem

Instance: Undirected graph $G = (V, E)$ with n vertices and m edges, edge costs c_e, a set of demands (pairs of vertices) D, integers $K \geq 1$ and $H \geq 2$.

Solution: A minimum cost subgraph T containing K edge-disjoint paths of length at most H joining the pairs of vertices in each demand.

- K controls the desired level of Network Survivability,
- H controls the Quality of Service requirements.

Instances where all the demands have a common vertex (*the root*) are called *rooted*, the other instances are *unrooted*.

A vertex that does not belong to any demand is a *Steiner vertex*. Instances without Steiner vertices are *spanning*.

The Survivable Network Design with Hop Constraints (SNDH) Problem

Instance: Undirected graph $G = (V, E)$ with n vertices and m edges, edge costs c_e, a set of demands (pairs of vertices) D, integers $K \geq 1$ and $H \geq 2$.

Solution: A minimum cost subgraph T containing K edge-disjoint paths of length at most H joining the pairs of vertices in each demand.

- K controls the desired level of Network Survivability,
- H controls the Quality of Service requirements.

Instances where all the demands have a common vertex (*the root*) are called *rooted*, the other instances are *unrooted*.

A vertex that does not belong to any demand is a *Steiner vertex*. Instances without Steiner vertices are *spanning*.

The Survivable Network Design with Hop Constraints (SNDH) Problem

Instance: Undirected graph $G = (V, E)$ with n vertices and m edges, edge costs c_e, a set of demands (pairs of vertices) D, integers $K \geq 1$ and $H \geq 2$.

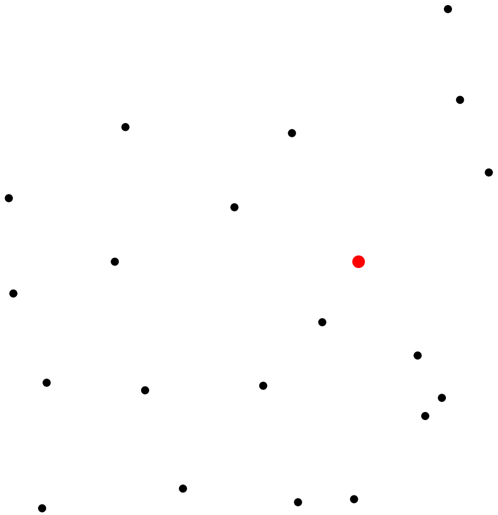
Solution: A minimum cost subgraph T containing K edge-disjoint paths of length at most H joining the pairs of vertices in each demand.

- K controls the desired level of Network Survivability,
- H controls the Quality of Service requirements.

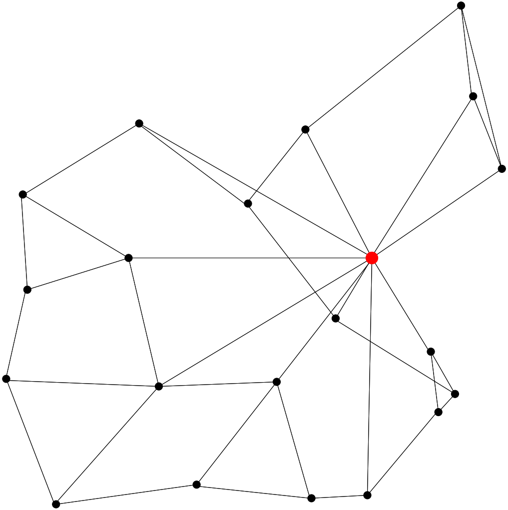
Instances where all the demands have a common vertex (*the root*) are called *rooted*, the other instances are *unrooted*.

A vertex that does not belong to any demand is a *Steiner vertex*. Instances without Steiner vertices are *spanning*.

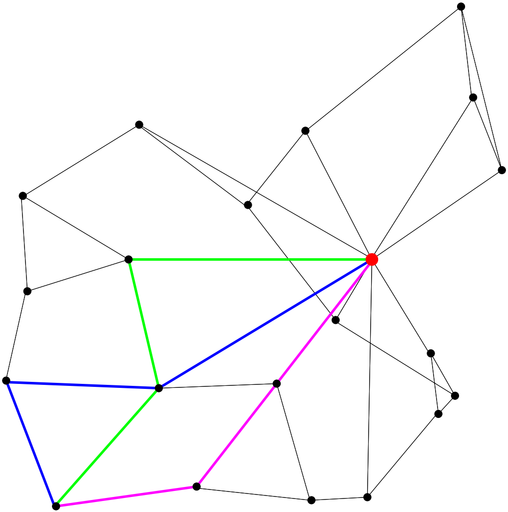
Example of a rooted spanning instance with $K = 3$ and $H = 3$; complete graph, Euclidean costs.



Example of a rooted spanning instance with $K = 3$ and $H = 3$; complete graph, Euclidean costs.



Example of a rooted spanning instance with $K = 3$ and $H = 3$; complete graph, Euclidean costs.



The Survivable Network Design with Hop Constraints (SNDH) Problem

A more general version considers potentially distinct values $K(d)$ and $H(d)$ for each $d \in D$ in order to model demand importance.

There is an even more general version where each demand has its required profile of Survivability $\times$ QoS.

- For example, an important demand may require a primary path of length ≤ 2 and two secondary paths of length ≤ 3.
- A less important demand may require a primary path of length ≤ 3 and a secondary path of length at ≤ 4.

Complexity of the SNDH Problem

Even some very particular cases are already NP-hard.

- Case $|D| = 1$ (single demand):
  - Polynomial for $H = 2$ or 3;
  - NP-hard for $H \geq 4$.
- Case $K = 1$, rooted and spanning (equivalent to the Spanning Tree with Hop Constraints Problem):
  - NP-hard for $H \geq 2$.

Some recent algorithmic work on the SNDH Problem

- Case $K = 2$:
  - Huygens, Labbé, Mahjoub and Pesneau (2007) – Facet-defining inequalities on the natural variables, branch-and-cut.
- Case $K = 3$:
  - Diarrassouba, Gabrel and Mahjoub (2010) – Facet-defining inequalities on the natural variables, branch-and-cut.
- General SNDH:
  - Botton, Fortz, Gouveia and Poss (2010) – Extended formulation, Benders decomposition.

Significant gaps, some instances with only 20 demands can be very challenging.

Some recent algorithmic work on the SNDH Problem

- Case $K = 2$:
  - Huygens, Labbé, Mahjoub and Pesneau (2007) – Facet-defining inequalities on the natural variables, branch-and-cut.
- Case $K = 3$:
  - Diarrassouba, Gabrel and Mahjoub (2010) – Facet-defining inequalities on the natural variables, branch-and-cut.
- General SNDH:
  - Botton, Fortz, Gouveia and Poss (2010) – Extended formulation, Benders decomposition.

Significant gaps, some instances with only 20 demands can be very challenging.

Hop Multi-Commodity Flow Formulation (Hop-MCF), BFGP10

- Each edge $(i, j) \in E$ defines a binary design variable x_{ij}.
- Each demand $d = (u, v) \in D$ defines an auxiliary network with H layers, with associated binary variables f_{ij}^{dh} (a path serving demand d goes from i to j at hop h). There must be K units of flow in each network.
- The f variables are coupled to the x variables.

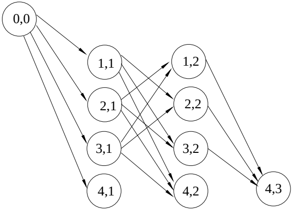


Figure: Example of network with $d = (0, 4)$, $H = 3$.

Hop Multi-Commodity Flow Formulation (Hop-MCF), BFGP10

$$\min \sum_{(i,j)\in E} c_{ij} x_{ij} \qquad (1)$$

s.t.

$$\sum_{[j,i,h]\in\delta^-(i,h)} f_{ji}^{dh} - \sum_{[i,j,h+1]\in\delta^+(i,h)} f_{ij}^{d(h+1)} = 0 \qquad d \in D; (i,h) \in V_H^d, i \notin \{o_d, d_d\} \qquad (2)$$

$$\sum_{[o_d,j,1]\in\delta^+(o_d,0)} f_{o_d j}^{d1} = K \qquad d \in D \qquad (3)$$

$$\sum_{h=1}^{H} \sum_{[j,d_d,h]\in\delta^-(d_d,h)} f_{jd_d}^{dh} = K \qquad d \in D \qquad (4)$$

$$f_{o_d j}^{d1} \le x_{o_d j} \qquad d \in D; (o_d, j) \in \delta(o_d) \qquad (5)$$

$$\sum_{h=2}^{H-1} (f_{ji}^{dh} + f_{ij}^{dh}) \le x_{ij} \qquad d \in D; (i,j) \in E \setminus (\delta(o_d) \cup \delta(d_d)) \qquad (6)$$

$$\sum_{h=2}^{H} f_{jd_d}^{dh} \le x_{jd_d} \qquad d \in D; (j, d_d) \in \delta(o_d) \qquad (7)$$

Hop Multi-Commodity Flow Formulation (Hop-MCF), BFGP10

- Only known formulation for the most general versions of the SNDH.
- Quite large size: $O(|D|.H.m)$ variables and $O(|D|.H.n)$ constraints.
- Typical duality gaps: 5% – 25%.

Our Goal

Introduce formulations significantly stronger than Hop-MCF for the general SNDH problem.

- It is well-known that extending a formulation may yield smaller gaps. Even automatic extension schemes (e.g. Sherali and Adams' RLT) do exist.
- However we do not want to increase the formulation size by a large factor that may depend on n or m, but only by a **small constant factor**, that can be even **controlled**.

Our Goal

Introduce formulations significantly stronger than Hop-MCF for the general SNDH problem.

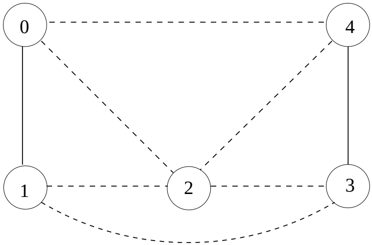
- It is well-known that extending a formulation may yield smaller gaps. Even automatic extension schemes (e.g. Sherali and Adams' RLT) do exist.
- However we do not want to increase the formulation size by a large factor that may depend on n or m, but only by a **small constant factor**, that can be even **controlled**.

Our Goal

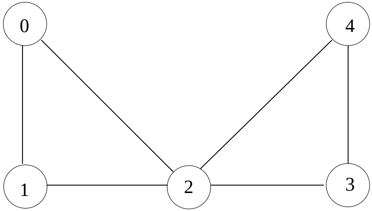
Introduce formulations significantly stronger than Hop-MCF for the general SNDH problem.

- It is well-known that extending a formulation may yield smaller gaps. Even automatic extension schemes (e.g. Sherali and Adams' RLT) do exist.
- However we do not want to increase the formulation size by a large factor that may depend on n or m, but only by a **small constant factor**, that can be even **controlled**.

Spanning instance rooted at 0 with $K = 2$ and $H = 3$; complete graph, Euclidean costs.



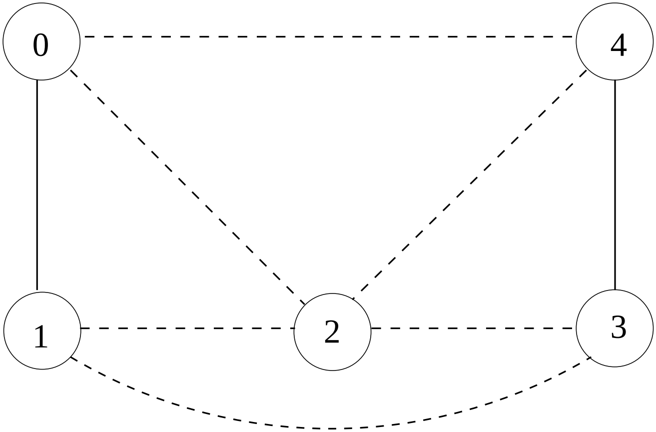
Linear relaxation of Hop-MCF (cost 641).



Optimal integral solution (cost 682).

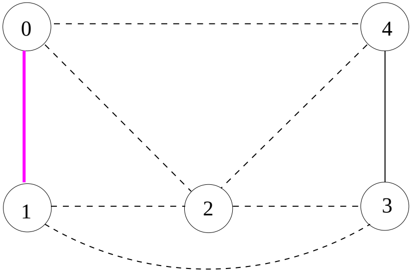
How Hop-MCF is cheating?

There are fractional $u - v$ paths with length ≤ 3 summing 2 for each demand (u, v).



How Hop-MCF is cheating?

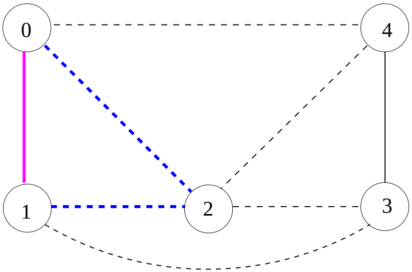
For example, take demand $(0,1)$:



- Path 0-1 with value 1;

How Hop-MCF is cheating?

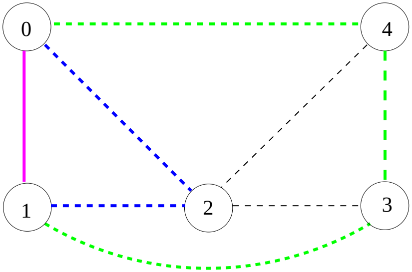
For example, take demand $(0, 1)$:



- Path 0-1 with value 1;
- Path 0-2-1 with value $1/2$;

How Hop-MCF is cheating?

For example, take demand $(0, 1)$:



- Path 0-1 with value 1;
- Path 0-2-1 with value $1/2$;
- Path 0-4-3-1 with value $1/2$.

The Main Idea: Sort the vertices by their distances to a source.

Given a solution T, a chosen source vertex $s \in V$ and a chosen positive integer L, we can partition V into $L + 2$ levels, according to their **distance from** s **in** T, as follows:

- Level 0 only contains s;
- Level i, $1 \leq i \leq L - 1$, contains vertices with distance i;
- Level L contains the vertices with finite distance $\geq L$;
- Level $L + 1$ contains the vertices with infinite distance.

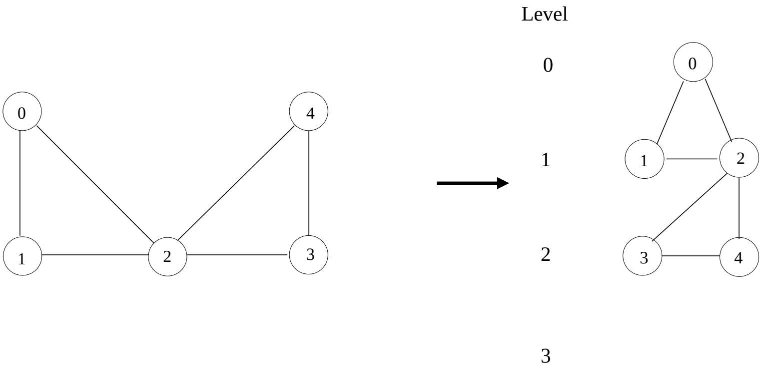
The Hop-Level Multi-Commodity Flow Formulation (HL-MCF)

The proposed formulation, besides the m edge variables x, also has:

- $O(L.n)$ binary variables w_i^l indicating if vertex i is in level l;
- $O(L.m)$ binary variables $y_{ij}^{l_1 l_2}$ indicating that edge (i, j) belongs to T and that i is in level l_1 and j in level l_2; (remark that $|l_1 - l_2| \leq 1$)
- $O(|D|.L.H.m)$ binary flow variables $g_{ij}^{dhl_1 l_2}$ associated to $|D|$ auxiliary Hop-Level networks.

Translating an integral x solution into (w, y) variables.

- $x_{01} = x_{02} = x_{12} = x_{23} = x_{24} = x_{34} = 1.$



- $w_0^0 = w_1^1 = w_2^1 = w_3^2 = w_4^2 = 1.$
- $y_{01}^{01} = y_{02}^{01} = y_{12}^{11} = y_{23}^{12} = y_{24}^{12} = y_{34}^{22}.$

The Hop-Level Multi-Commodity Flow Formulation (HL-MCF)

$$\min \sum_{(i,j)\in E} c_{ij} x_{ij} \tag{10}$$

The x and (w, y) variables are linked by the following constraints:

$$w_s^0 = 1 \tag{11}$$

$$\sum_{l=1}^{L+1} w_i^l = 1 \qquad i \in V \setminus s \tag{12}$$

$$w_j^1 = y_{sj}^{01} = x_{sj} \qquad (s,j) \in \delta(s) \tag{13}$$

$$\sum_{l=1}^{L-1} (y_{ij}^{l(l+1)} + y_{ji}^{l(l+1)}) + \sum_{l=1}^{L+1} y_{ij}^{ll} = x_{ij} \qquad (i,j) \in E \setminus \delta(s) \tag{14}$$

The Hop-Level Multi-Commodity Flow Formulation (HL-MCF)

$$\begin{aligned} y_{ij}^{ll} + y_{ij}^{l(l+1)} &\leq w_i^l \\ y_{ij}^{ll} + y_{ji}^{l(l+1)} &\leq w_j^l \end{aligned} \qquad (i,j) \in E \setminus \delta(s);\, l = 1 \qquad (15)$$

$$\begin{aligned} y_{ij}^{ll} + y_{ij}^{l(l+1)} + y_{ji}^{(l-1)l} &\leq w_i^l \\ y_{ij}^{ll} + y_{ji}^{l(l+1)} + y_{ij}^{(l-1)l} &\leq w_j^l \end{aligned} \qquad (i,j) \in E \setminus \delta(s);\, l = 2, \ldots, L-1 \qquad (16)$$

$$\begin{aligned} y_{ij}^{ll} + y_{ji}^{(l-1)l} &\leq w_i^l \\ y_{ij}^{ll} + y_{ij}^{(l-1)l} &\leq w_j^l \end{aligned} \qquad (i,j) \in E \setminus \delta(s);\, l = L \qquad (17)$$

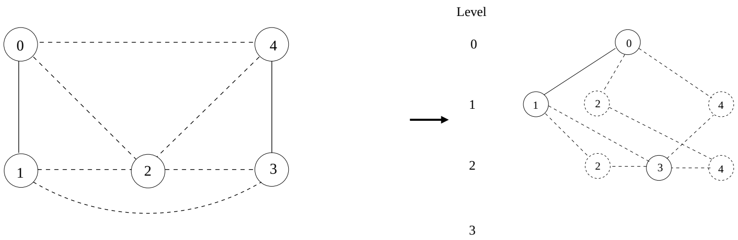
$$\begin{aligned} y_{ij}^{ll} &\leq w_i^l \\ y_{ij}^{ll} &\leq w_j^l \end{aligned} \qquad (i,j) \in E \setminus \delta(s);\, l = L+1 \qquad (18)$$

$$w_i^l \leq \sum_{j \in \delta(i), j \neq s} y_{ji}^{(l-1)l} \qquad i \in V \setminus s;\, l = 2, \ldots, L-1 \qquad (19)$$

$$w_i^l \leq \sum_{j \in \delta(i), j \neq s} (y_{ji}^{(l-1)l} + y_{ij}^{ll}) \qquad i \in V \setminus s;\, l = L \qquad (20)$$

Translating a fractional x solution into (w, y) variables.

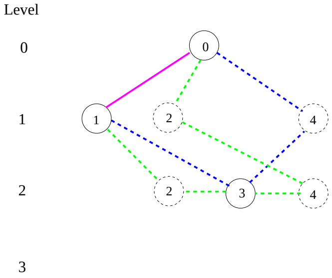
- $x_{01} = x_{34} = 1$; $x_{02} = x_{04} = x_{12} = x_{13} = x_{23} = x_{24} = 1/2$.



- $w_0^0 = w_1^1 = w_3^1 = 1$; $w_2^1 = w_2^2 = w_4^1 = w_4^2 = 1/2$.
- $y_{01}^{01} = 1$;
  $y_{02}^{01} = y_{04}^{01} = y_{12}^{12} = y_{13}^{12} = y_{24}^{12} = y_{43}^{12} = y_{23}^{22} = y_{34}^{22} = 1/2$.

What is wrong with that (w, y) solution?

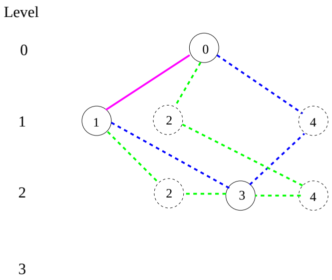
For example, take demand $(0, 1)$:



- Path 0-1 with value 1;
- Path 0-4-3-1 with value 1/2.
- The only remaining path 0-2-4-3-2-1 has length 5. Wrong! The splitting of vertex 2 removed path 0-2-1.
- Cutting the (w, y) solution indirectly cuts the x solution.

What is wrong with that (w, y) solution?

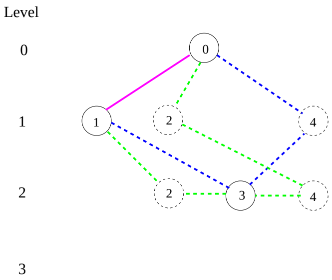
For example, take demand $(0, 1)$:



- Path 0-1 with value 1;
- Path 0-4-3-1 with value $1/2$.
- The only remaining path 0-2-4-3-2-1 has length 5. Wrong! The splitting of vertex 2 removed path 0-2-1.
- Cutting the (w, y) solution indirectly cuts the x solution.

What is wrong with that (w, y) solution?

For example, take demand $(0, 1)$:



- Path 0-1 with value 1;
- Path 0-4-3-1 with value $1/2$.
- The only remaining path 0-2-4-3-2-1 has length 5. Wrong! The splitting of vertex 2 removed path 0-2-1.
- Cutting the (w, y) solution indirectly cuts the x solution.

The Hop-Level Multi-Commodity Flow Formulation (HL-MCF)

The HL-MCF is completed by enforcing, for each demand $d = (u, v)$, the existence of K (u, v)-paths with length $\leq H$ in the network induced by the (w, y) solution.

This is done by building $|D|$ auxiliary hop-level indexed networks.

- One variable for each arc in those networks: $g_{ij}^{dhl_1 l_2}$ indicates that a path serving demand d goes from i at level l_1 to j at level l_2 in its hop h.

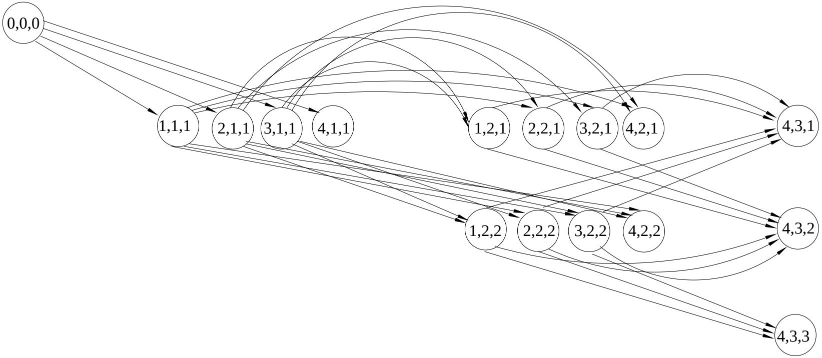
The Hop-Level Multi-Commodity Flow Formulation (HL-MCF)

The HL-MCF is completed by enforcing, for each demand $d = (u, v)$, the existence of K (u, v)-paths with length $\leq H$ in the network induced by the (w, y) solution.

This is done by building $|D|$ auxiliary hop-level indexed networks.

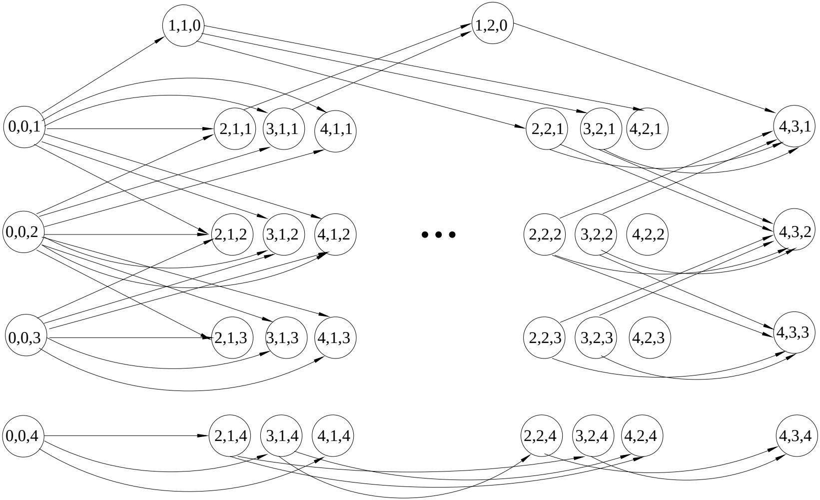
- One variable for each arc in those networks: $g_{ij}^{dhl_1 l_2}$ indicates that a path serving demand d goes from i at level l_1 to j at level l_2 in its hop h.

Hop-Level Network, source is one of the vertices of the demand: $d = (0, 4)$, $H = 3$, $L = 3$, and $s = 0$.



- There must be $K.w_4^l$ units of flow from $(0, 0, 0)$ to vertices $(4, h, l)$;
- The g variables are constrained by the corresponding y variables.

Hop-Level Network, general case: $d = (0, 4)$, $H = 3$, $L = 3$, and $s = 1$.



- Level $L+1$ usually not necessary.

Hop-MCF × HL-MCF: 72 rooted instances, complete graphs with $n = 21$, Euc. cost, 5 to 20 demands.

Table: Rooted instances: average percentual duality gaps.

	H	$K = 1$	$K = 2$	$K = 3$
Hop	2	14.99	12.92	7.38
HL		0.00	0.40	0.08
Hop	3	23.91	13.13	7.64
HL		0.00	2.83	3.27
Hop	4	25.82	13.05	7.00
HL		0.83	5.62	5.20
Hop	5	26.94	9.60	6.01
HL		1.93	5.08	5.27

In the HL-MCF, L is taken as $\min\{H, 4\}$, the source is the root.

Hop-MCF × HL-MCF: 72 rooted instances, complete graphs with $n = 21$, Euc. cost, 5 to 20 demands.

	H	$K = 1$	$K = 2$	$K = 3$
Hop	2	14.99	12.92	7.38
HL		0.00	0.40	0.08
Hop	3	23.91	13.13	7.64
HL		0.00	2.83	3.27
Hop	4	25.82	13.05	7.00
HL		0.83	5.62	5.20
Hop	5	26.94	9.60	6.01
HL		1.93	5.08	5.27

- Gap reduction $> 75\%$
- $50 \leq$ Gap reduction ≤ 75
- Gap reduction $< 50\%$

Explaining the results of HL-MCF.

	H	$K=1$	$K=2$	$K=3$
Hop	2	14.99	12.92	7.38
HL		0.00	0.40	0.08
Hop	3	23.91	13.13	7.64
HL		0.00	2.83	3.27
Hop	4	25.82	13.05	7.00
HL		0.83	5.62	5.20
Hop	5	26.94	9.60	6.01
HL		1.93	5.08	5.27

HL-MCF works by splitting vertices, cutting shorter paths in the fractional solution; the remaining paths have length $> H$.

- As K increases, the solution gets denser, vertices are concentrated on levels 1 and 2 and are not sufficiently split.
- As H increases, the hop-constraints are looser and it is easier to find alternative paths with length $\leq H$.

Hop-MCF $\times$ HL-MCF: 38 unrooted instances, sparse graphs, 9 to 48 demands.

Table: Unrooted instances: average percentual duality gaps.

	H	$K = 1$	$K = 2$	$K = 3$
Hop	2	18.03	10.68	11.03
HL		9.01	7.46	8.00
Hop	3	19.70	18.70	10.19
HL		6.07	11.63	8.17
Hop	4	24.79	12.89	5.21
HL		3.18	8.82	4.39
Hop	5	30.67	10.67	2.86
HL		3.81	6.97	2.86

In the HL-MCF, $L = 5$, the source is vertex 0.

Hop-MCF × HL-MCF: 38 unrooted instances, sparse graphs, 9 to 48 demands.

Table: Unrooted instances: average percentual duality gaps.

	H	$K=1$	$K=2$	$K=3$
Hop	2	18.03	10.68	11.03
HL		9.01	7.46	8.00
Hop	3	19.70	18.70	10.19
HL		6.07	11.63	8.17
Hop	4	24.79	12.89	5.21
HL		3.18	8.82	4.39
Hop	5	30.67	10.67	2.86
HL		3.81	6.97	2.86

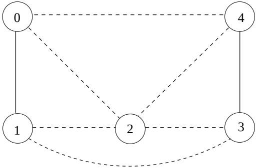
Less satisfactory results.

Analysis of HL-MCF

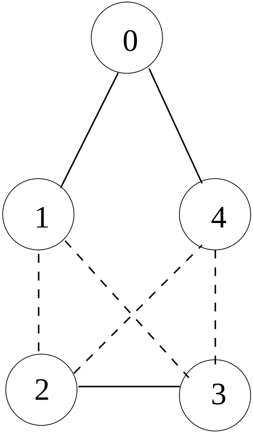
The gap reductions obtained by HL-MCF are very significant in some cases, specially for the rooted instances, allowing dramatic reductions in the overall time taken by a B&B algorithm.

In other cases, specially for non-rooted instances, the reductions are not enough to compensate for the increase in formulation size.

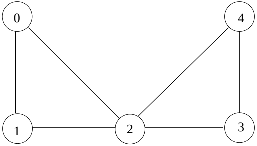
Spanning instance rooted at 0 with $K = 2$ and $H = 3$; complete graph, Euclidean costs.



Linear relaxation of Hop-MCF (cost 641).

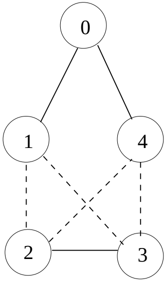


Linear relaxation of HL-MCF (cost 672).



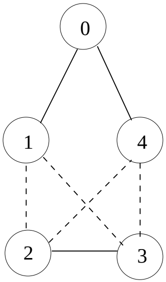
Optimal integral solution (cost 683).

How HL-MCF is cheating?



- The fractional edges are arranged in order to avoid vertex splitting. For example, $w_2^2 = 1$ because 2 is connected to source 0 by paths $0 - 1 - 2$ and $0 - 4 - 2$ with value $1/2$.

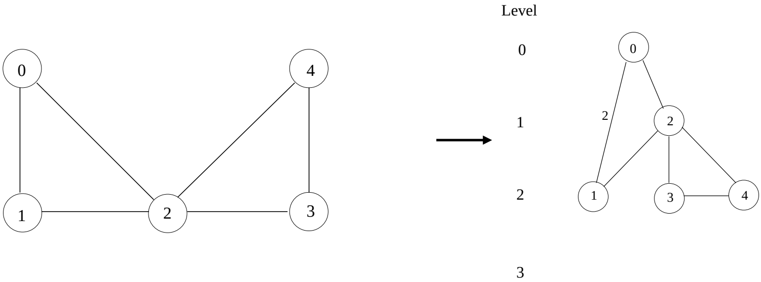
Forcing more vertex splittings: Non-unitary HL-MCF



- By setting $d(0,1) = 2$ (other distances remain 1), the same x solution would force $w_2^2 = w_2^3 = 1/2$, which would lead to infeasibility.
- The relaxation of the modified HL-MCF with that non-unitary distance is integral in this instance.

Translating an integral x solution into (w, y) variables, in case of non-unitary distances

- $x_{01} = x_{02} = x_{12} = x_{23} = x_{24} = x_{34} = 1.$



- $w_0^0 = w_1^2 = w_2^1 = w_3^2 = w_4^2 = 1.$
- $y_{01}^{02} = y_{02}^{01} = y_{12}^{21} = y_{23}^{12} = y_{24}^{12} = y_{34}^{22}.$

Non-unitary HL-MCF

In this generalization, one can choose not only s and L, but also the distances: an edge-vector of (small) integers between 0 and L.

The choice of the distance makes a lot of difference:

- It is typical that a good choice (at the moment, a lucky choice) closes half of the gap, while several poor choices are worse than unitary HL-MCF.

Non-unitary HL-MCF

We are now trying to find systematic ways of choosing good distance vectors.

- Some amount of trial and error is not unreasonable.
- Perhaps one can even use a few distinct distance vectors at once.
    - For each vector there would be a distinct set of variables (w, y, g), the x solution should be compatible with all of them.

Conclusions

- The proposed unitary distance HL-MCF already proved to yield significant algorithmic improvements upon existing methods for solving some kinds of SNDH instances.
- The non-unitary version of HL-MCF is currently being investigated.
- The idea of trying to extend an existing formulation by only multiplying its size by a constant factor may be useful on other problems.

Thank you!