

Empowering Machine Based Next Generation Access Network Design by Gurobi Optimizer

GUROBI Summit 28th Oct. 2025

Peter Bachhiesl

Carinthia University of Applied Sciences

Dpt. Engineering and IT

p.bachhiesl@cuas.at

What means NGA design?

- morning session : **A1** decision system for planning and cost optimization in fiber rollouts
... technically this means to
- **compute spatial and technical layout of access and overlaying backhaul networks** such that
- **all relevant data** are considered (local network installation conditions, existing and usable infrastructure, connection objects ...)
- **networks are operable** – logical consistent cable- and fiber (-signal) paths and according origination, termination and distribution components
- **user driven network design** is supported – parameter driven network topologies (e.g. PON vs. GPON, distribution levels, components ...)
- **networks are optimized** (investment costs, revenues, consideration of quotas ...)

Decision complexity - real world data

... which are mapped on network graphs.

graph BaseData: node attribute character

attribute type	value type	# nodes	sum
easting	valueDouble	381558	19630556278.0941
northing	valueDouble	381558	128354959296.6310
second	valueInteger	8885	45539
muI	valueDouble	15551	15551.0000
KA	valueString	3606	3606
zero	valueInteger	3606	40916
first	valueInteger	1124	3142
p	valueBoolean	12	12

graph BaseData: link attribute character

attribute type	value type	#links	mi
l	valueDouble	547096	0.100
L	valueString	547096	
I	valueString	46613	
tI	valueString	46613	
rhoI	valueDouble	46613	0.0
C	valueString	22001	
tC	valueString	22001	
nuC	valueInteger	22001	2

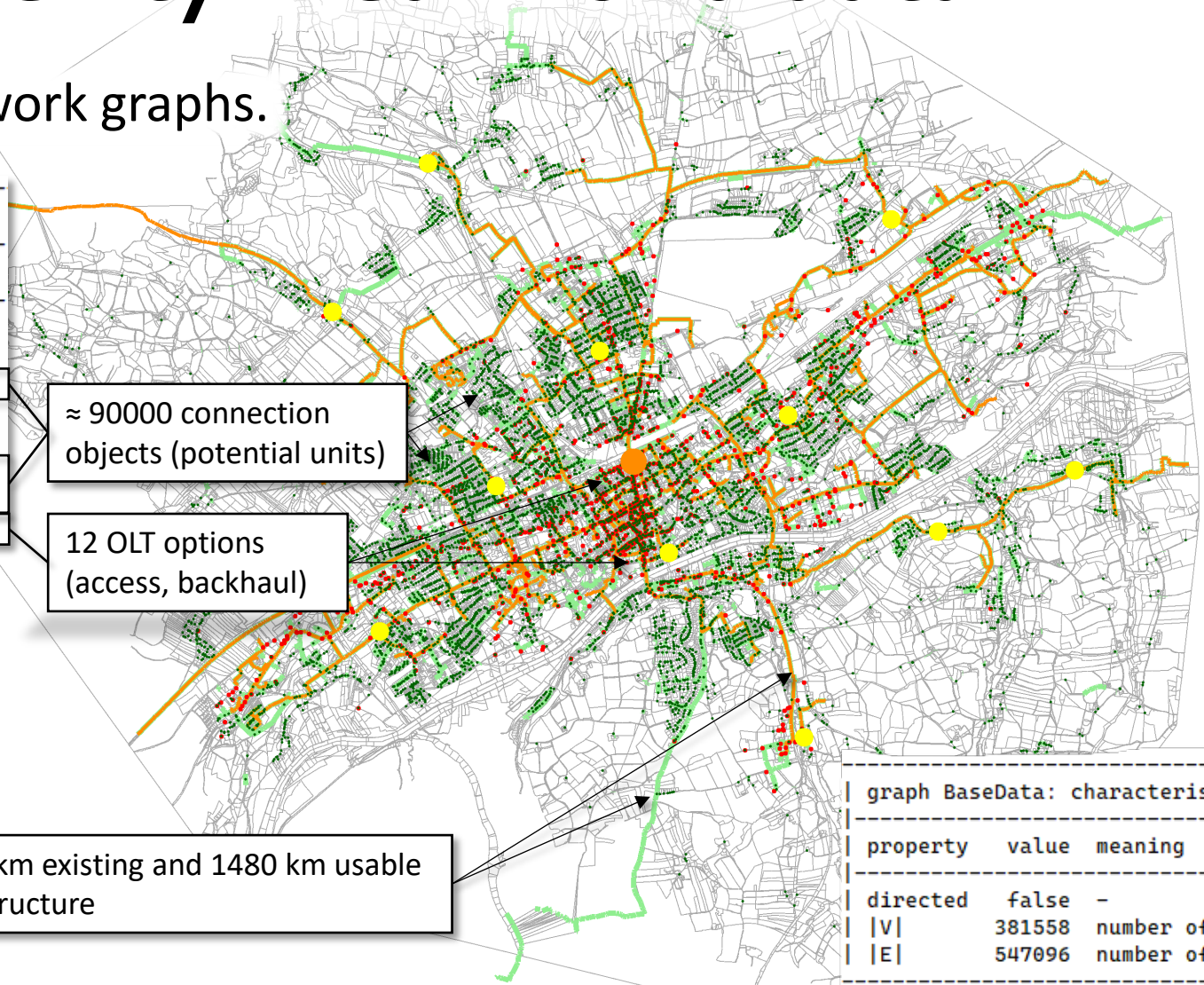
≈ 90000 connection objects (potential units)

12 OLT options (access, backhaul)

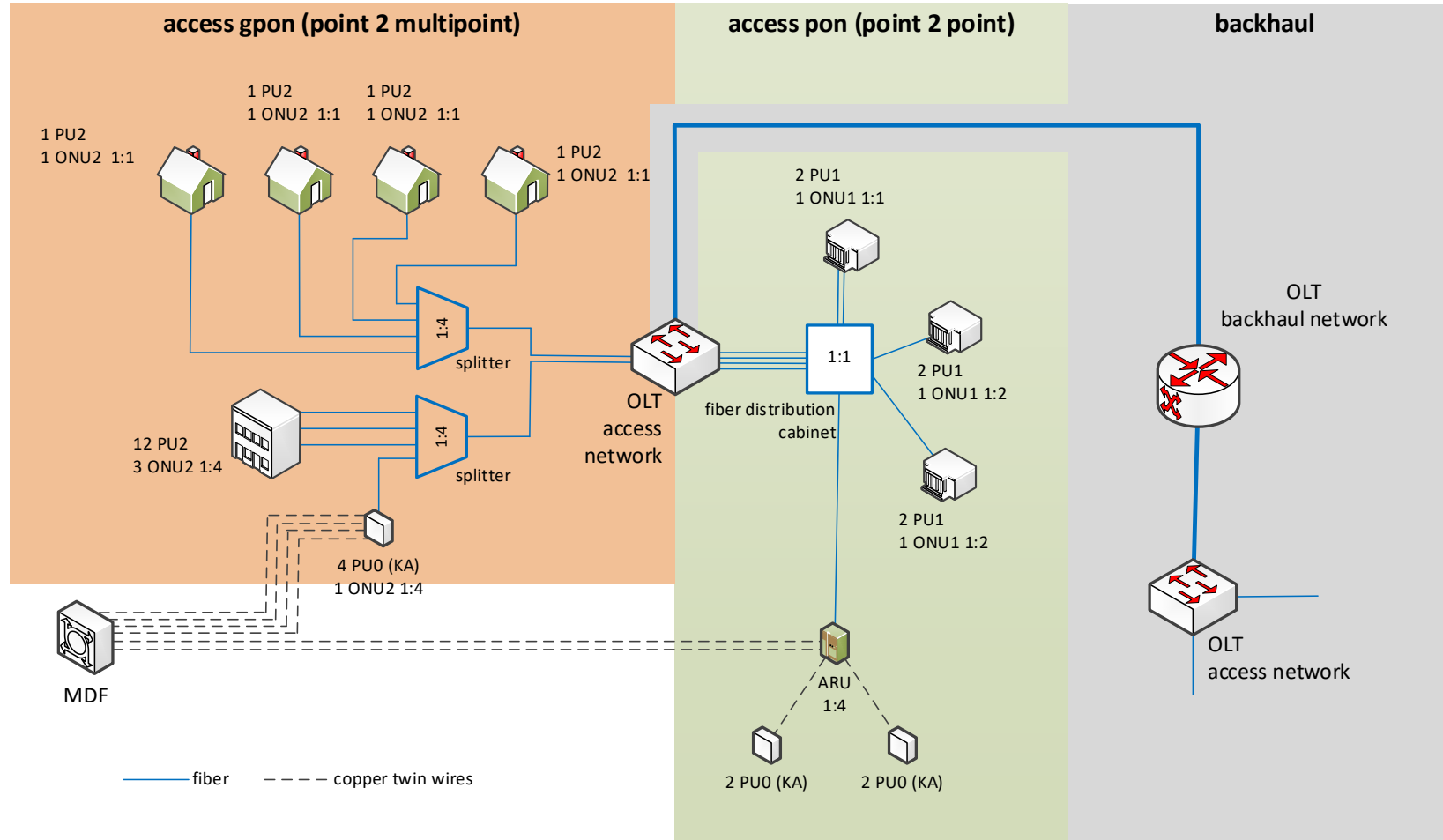
≈ 309 km existing and 1480 km usable infrastructure

graph BaseData: characteristics

property	value	meaning
directed	false	-
V	381558	number of nodes
E	547096	number of links

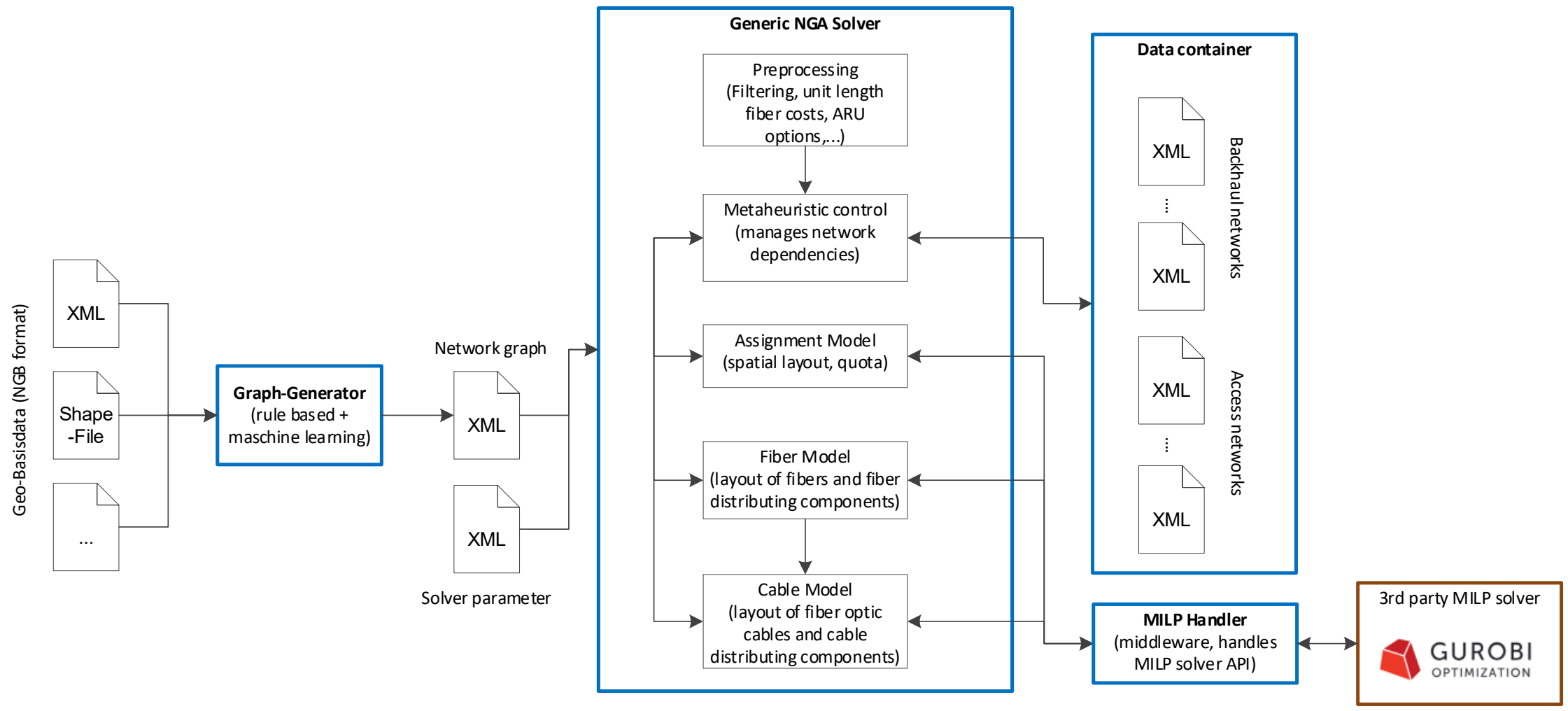


Network design layout



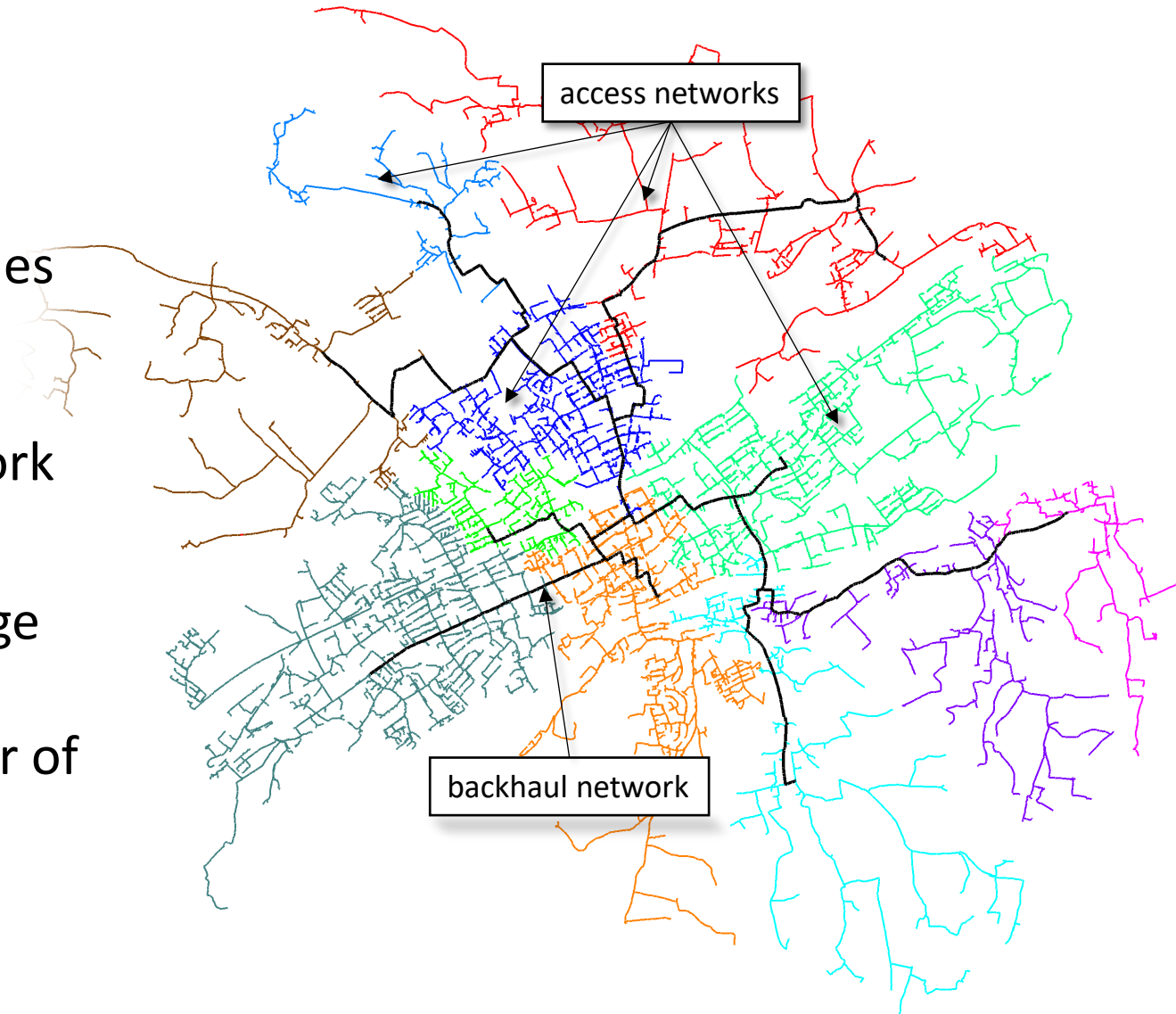
... but note:
everything here
(components,
distribution levels,
network topologies
...) might be
influenced by the
user!

Design workflow



Assignment model

- computes **spatial layout of networks**:
 - which access network supplies which potential unit
 - which backhaul network supplies which access network
- **Steiner forest problem**
 - quota constraints (percentage of potential units per level)
 - capacity constraints (number of potential units per level is restricted per network)



Assignment model ILP

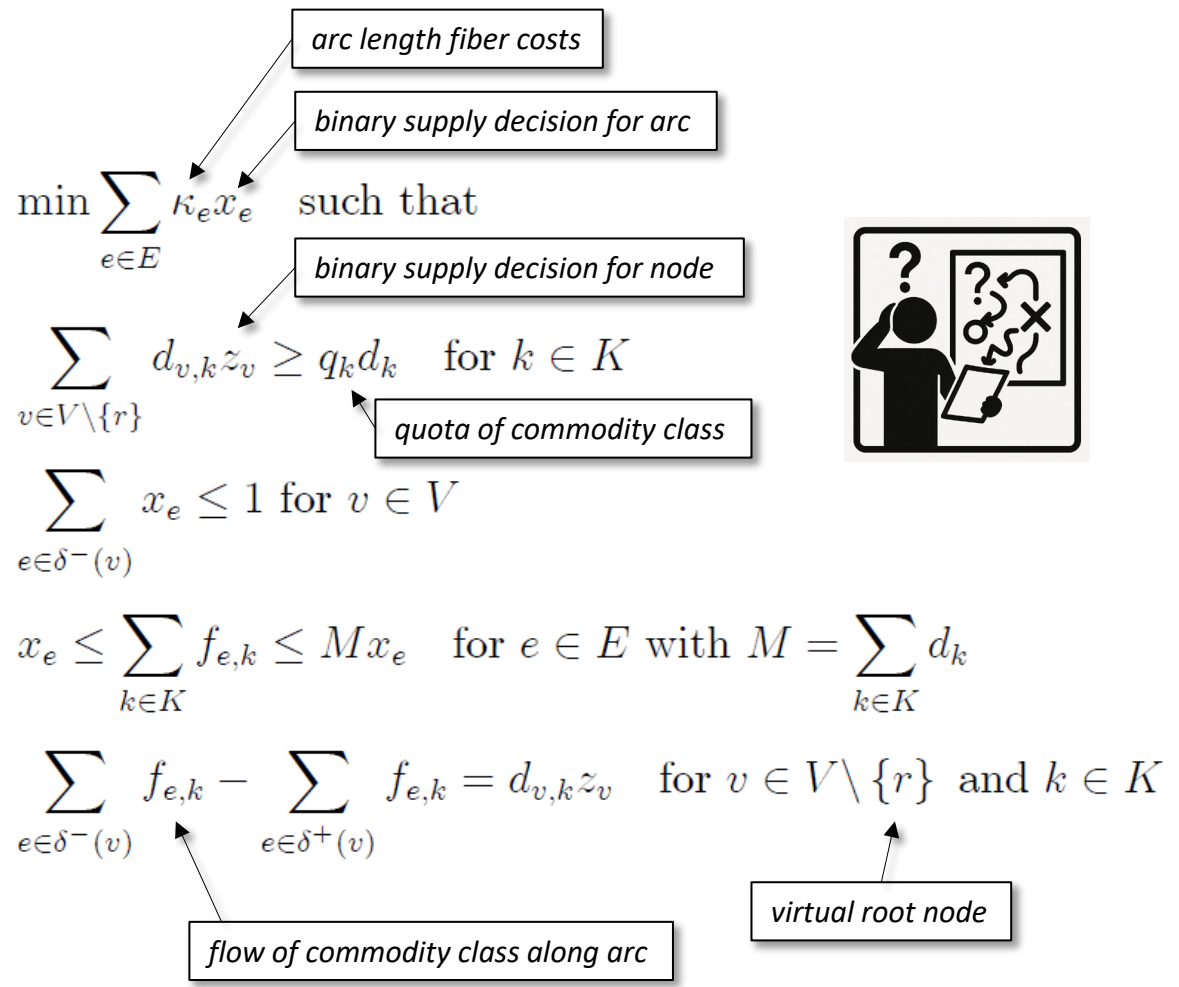
characteristics:

- multicommodity flow formulation on augmented network
- commodity classes are levels of potential units
- network capacities correspond to inflow restriction for OLT options, i.e.

$$f_{e,k} \leq p_{\omega(e),k} \text{ if } e \in \delta^+(r)$$

provision of commodity class on end node of arc (=OLT-option)

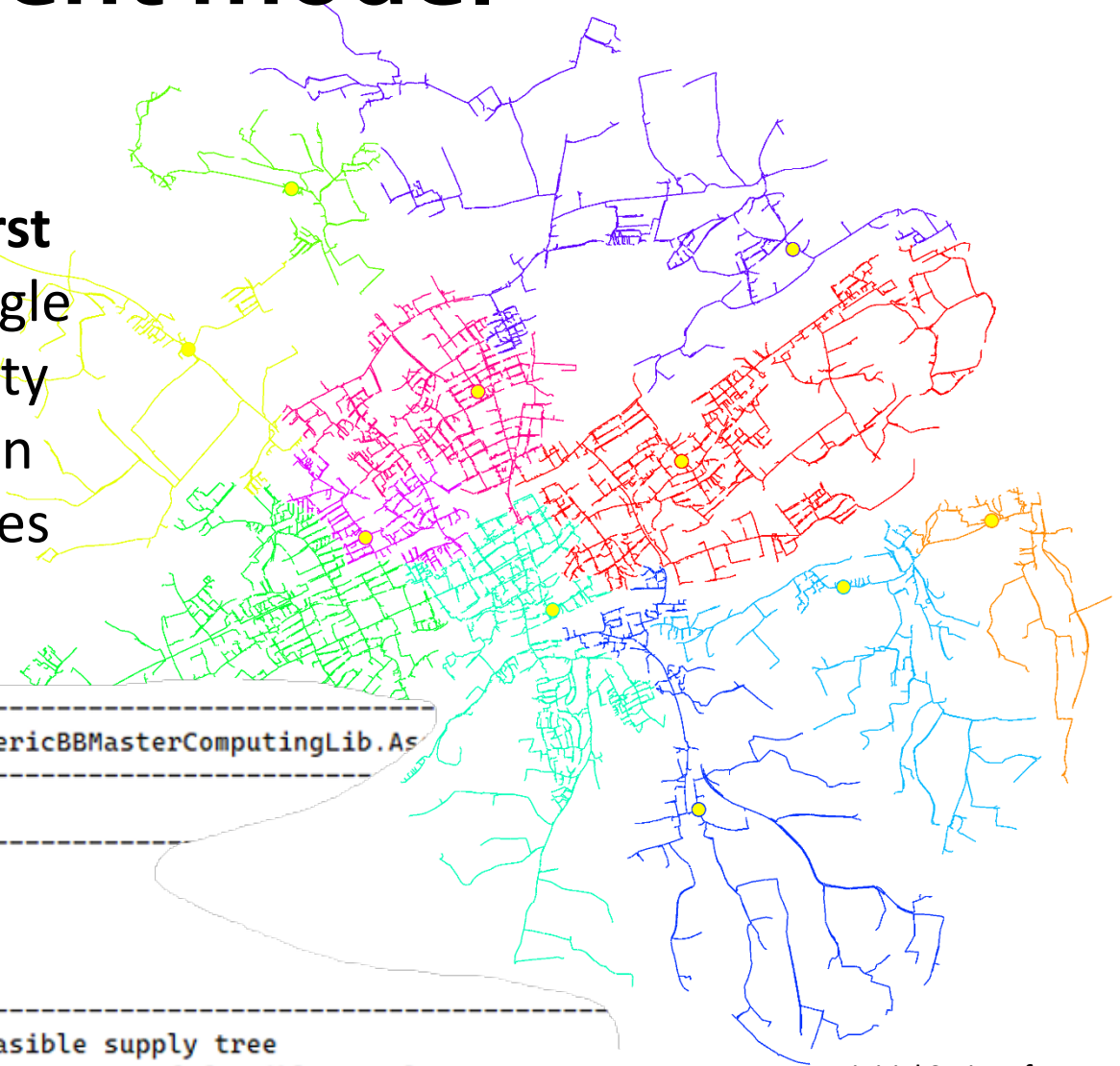
- **Solving model directly on graph** (populate and solve) usually **fails** (there is a Steiner forest problem inside!)



Start ILP for assignment model

initialization:

build **skeleton** (of graph instance) by a **first feasible Steiner forest** - e.g. compute single source shortest paths from the commodity providing nodes (=OLT options) and assign **heuristically** commodity demanding nodes along the computed paths to the forest



```

-----
| status graph [GenericBBMasterComputingLib.As
-----
| item  value
-----
| |V|    205255
| |E|    370793
| |F|    59167
| |H|     0
-----
| F ... edges in feasible supply tree
| H ... edges in enhancement of feasible supply tree
-----

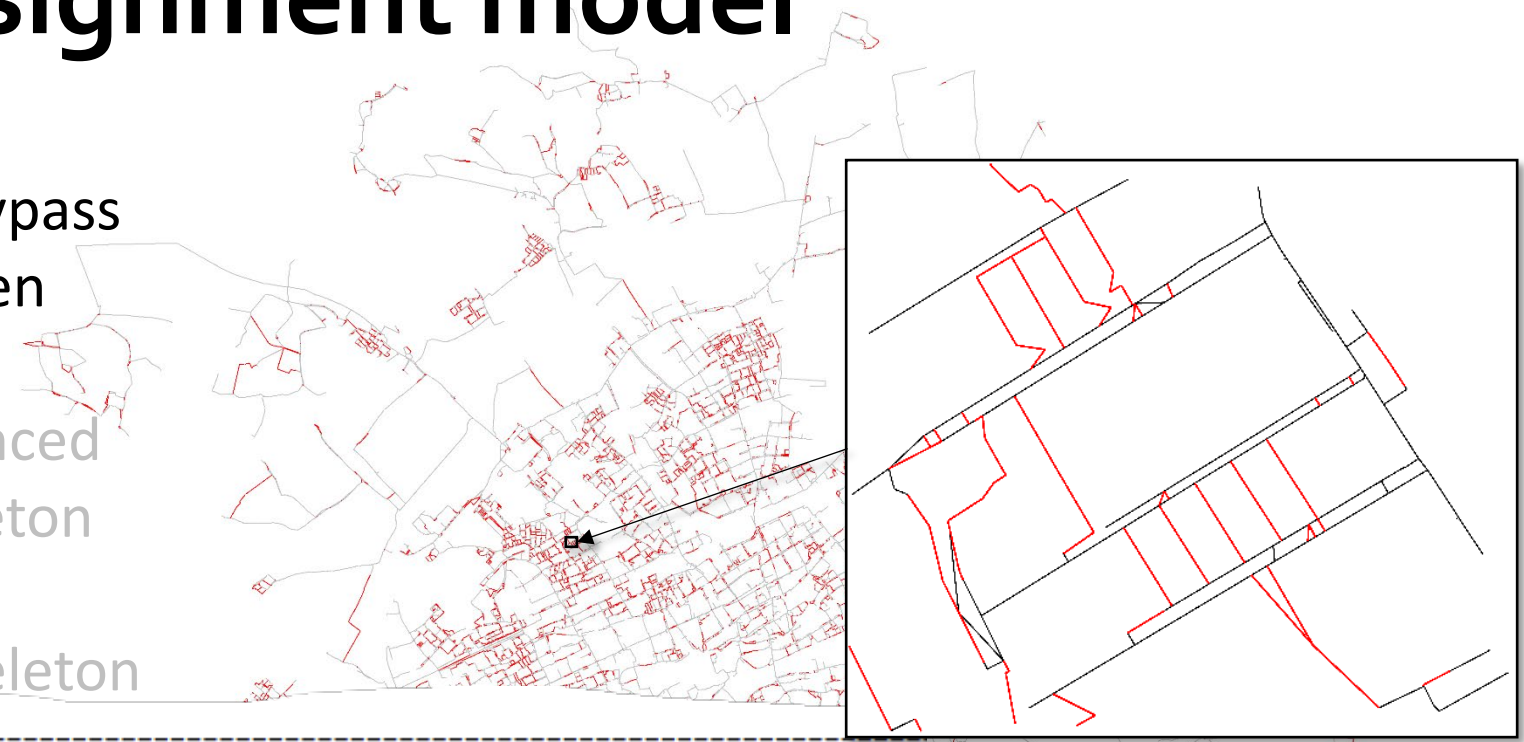
```

initial Steiner forest

Start ILP for assignment model

iteration:

1. enhance skeleton - e.g. bypass the skeleton (by cost driven search)
2. (re)solve ILP on the enhanced skeleton and update skeleton based on ILP solution
3. go back to 1 , until the skeleton becomes steady



```

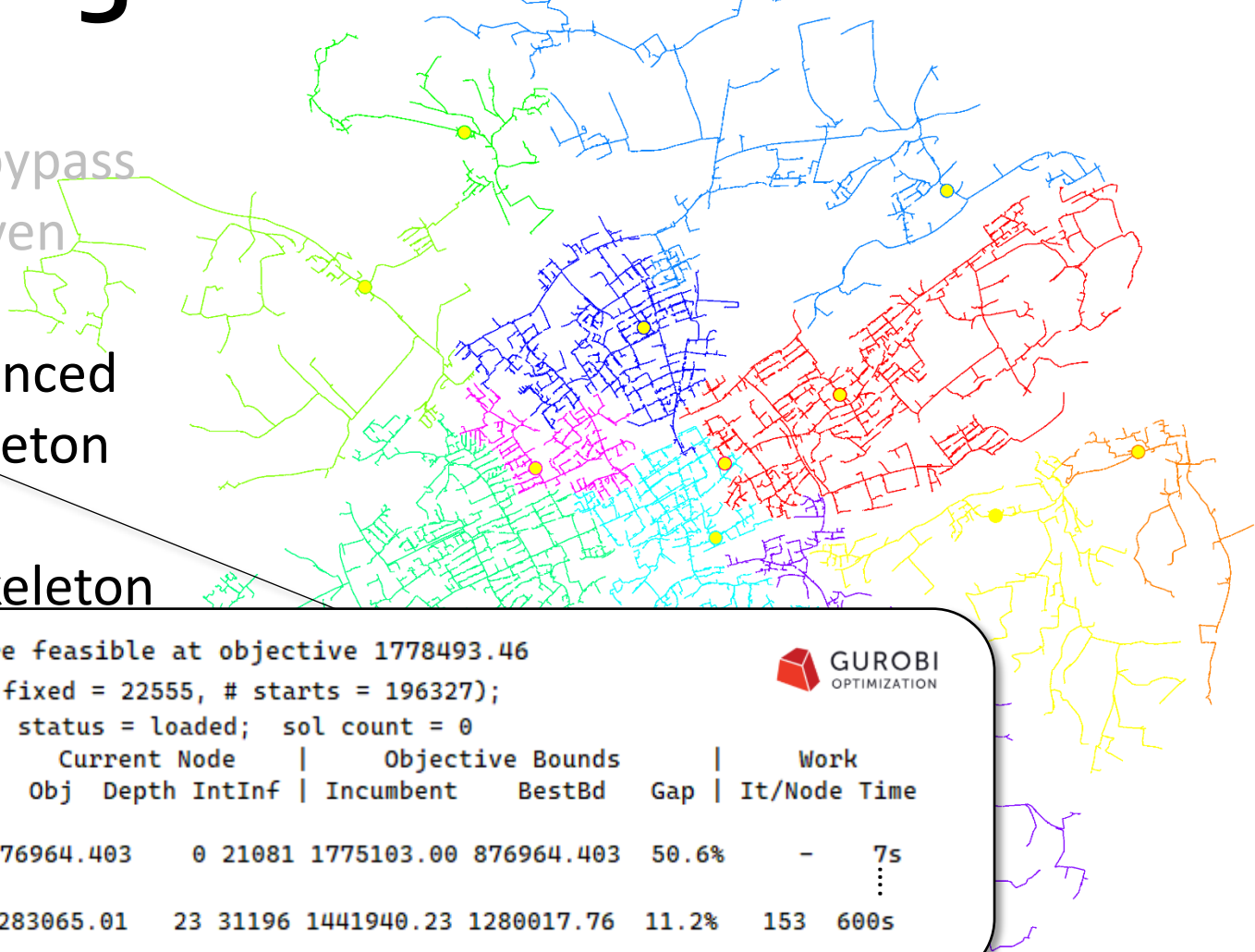
-----
| status graph [GenericBBMasterComputingLib.AssignmentScalingMo
-----
| item value
-----
| |V| 205255
| |E| 370793
| |F| 59167
| |H| 15776
-----
| F ... edges in feasible supply tree
| H ... edges in enhancement of feasible supply tree
-----
    
```

enhancement of skeleton
 — feasible Steiner forest
 — enhancement of feasible Steiner forest

Start ILP for assignment model

iteration:

1. enhance skeleton - e.g. bypass the skeleton (by cost driven search)
2. (re)solve ILP on the enhanced skeleton and update skeleton based on ILP solution
3. go back to 1, until the skeleton becomes st



```
variable starts are feasible at objective 1778493.46
# vars = 196327 (# fixed = 22555, # starts = 196327);
# constr = 205530; status = loaded; sol count = 0
[GRB] Nodes | Current Node | Objective Bounds | Work
[GRB] Expl Unexpl | Obj Depth IntInf | Incumbent BestBd Gap | It/Node Time
[GRB]
[GRB] 0 0 876964.403 0 21081 1775103.00 876964.403 50.6% - 7s
⋮
[GRB] 324 234 1283065.01 23 31196 1441940.23 1280017.76 11.2% 153 600s
```



feasible Steiner forest update

Assignment model – infeasibilities

user upper bounds for commodity provision and/or skeletonization may contradict to quotas → relax model and prove feasibility of current skeleton

... good quality relaxations are provided by *grb.feasrelax* .. you can even select the relaxation criterion!

class	pAttr	P	total provision	v in P	p(v)	dAttr	quota	quota * total demand	supplied demand	actual quota
first	p(first)	11	7700	17611 258502	700 700	d(first)	0.402610	1265.000000	1265	0.402610
		⋮								
second	p(second)	11	7700	17611 258502	700 700	d(second)	0.150618	6859.000000	6859	0.150618

quotas (originally 1.0) have been relaxed

```

MILP start and solving ...
check starts for all model variables ...
| [GRB]Gurobi Optimizer version 12.0.1 build v12.0.1rc0 (win64 - Windows Server 2019.0 (17763.2))
:
| [GRB]User-callback calls 31, time in user-callback 0.08 sec
variable starts are feasible at objective 437354.405361111
    
```

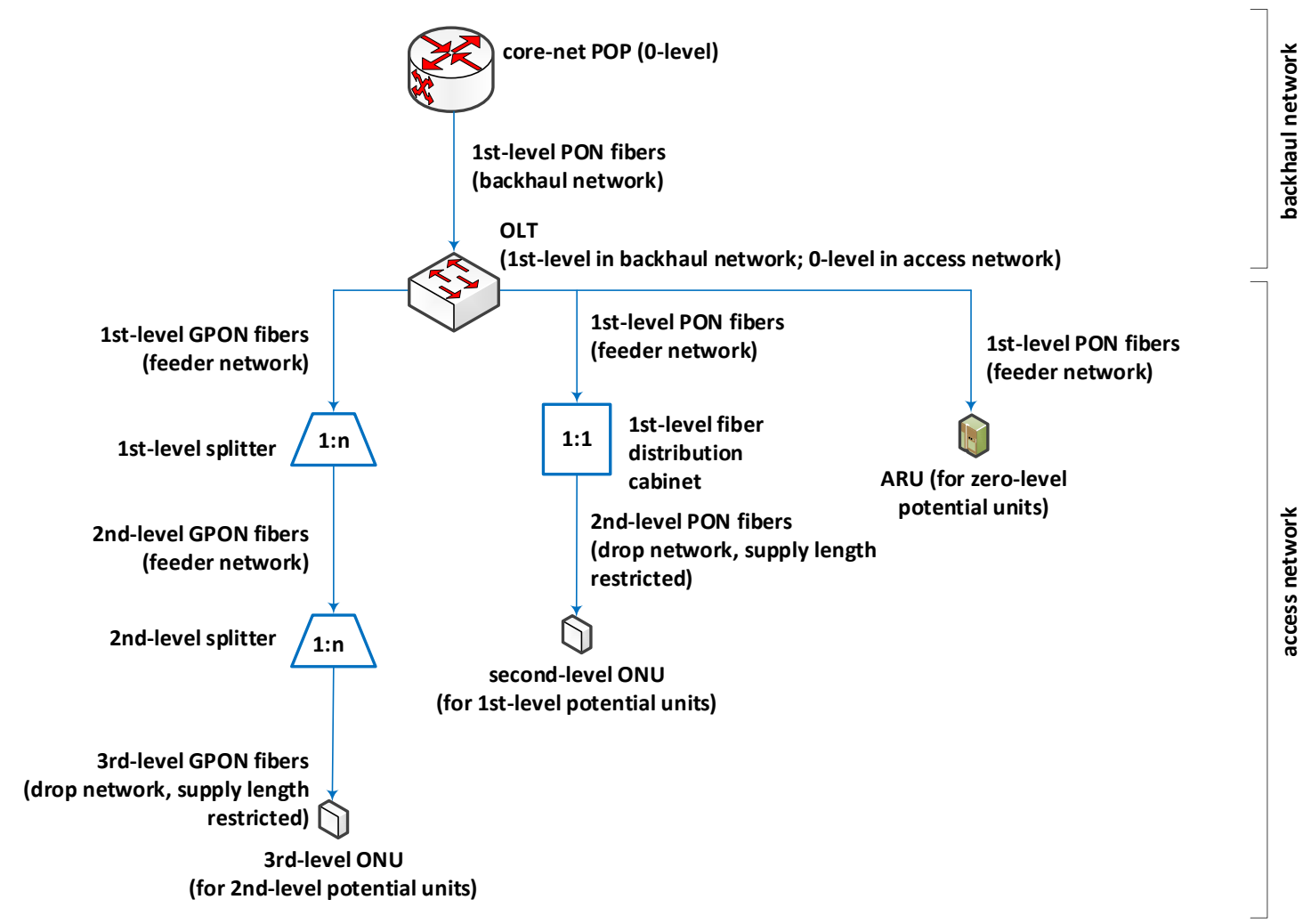


Fiber model - specification

computes **fibers** and **fiber-distributing components** on network trees under consideration of declared

- network topologies
- distribution levels
- components

and yields consistent up-/downstream fiber paths



Fiber model - MILP



multicommodity flow
formulation with

- fiber requirements
- component requirements
- distance requirements

For $n \in \{p, g\}$, $l = 1, \dots, l^n$ and $v \in V$ do

$$d_v^{n,l} \leq d_{\max}^{n,l} (1 - q_v^{n,l}).$$

For $n \in \{p, g\}$, $l = 1, \dots, l^n$, $v \in V \setminus \{r\}$ and $e \in \delta^-(v)$ do

$$d_v^{n,l} \geq (d_{\alpha(e)}^{n,l} + l_e) \bar{f}_e^{n,l+1}. \quad \leftarrow \text{nonlinearity}$$

For $n \in \{p, g\}$, $v \in V$ and $l = 1, \dots, l^n + 1$ do

$$\sum_{e \in \delta^-(v)} \bar{f}_e^{n,l} \leq 1.$$

For $n \in \{p, g\}$, $l = 1, \dots, l^n + 1$ and pairs of antiparallel arcs $(e, e') \in E \times E$ do

$$\bar{f}_e^{n,l} + \bar{f}_{e'}^{n,l} \leq 1 \quad \leftarrow \text{prohibits back-feeding at the same network level}$$

For $n \in \{p, g\}$ if $l^n > 0$ then for $v \in V \setminus \{r\}$ do

$$\sum_{e \in \delta^-(v)} f_e^{n,1} - \sum_{e \in \delta^+(v)} f_e^{n,1} = \sum_{t \in SODS_{n,1}} y_{v,t}^{n,1} + \sum_{t \in SODS_{n,1}} r_{tS_{v,t}}^{n,1}.$$

For $n \in \{p, g\}$, $v \in V$ and $l = 2, \dots, l^n$ do

$$\sum_{e \in \delta^+(v)} f_e^{n,l} - \sum_{e \in \delta^-(v)} f_e^{n,l} \leq \sum_{t \in SODS_{n,l-1}} \theta_t y_{v,t}^{n,l-1} - \sum_{t \in SODS_{n,l}} y_{v,t}^{n,l} - \sum_{t \in SODS_{n,l}} r_{tS_{v,t}}^{n,l}.$$

For $n \in \{p, g\}$ and if $l^n > 0$ then for $v \in V$ else

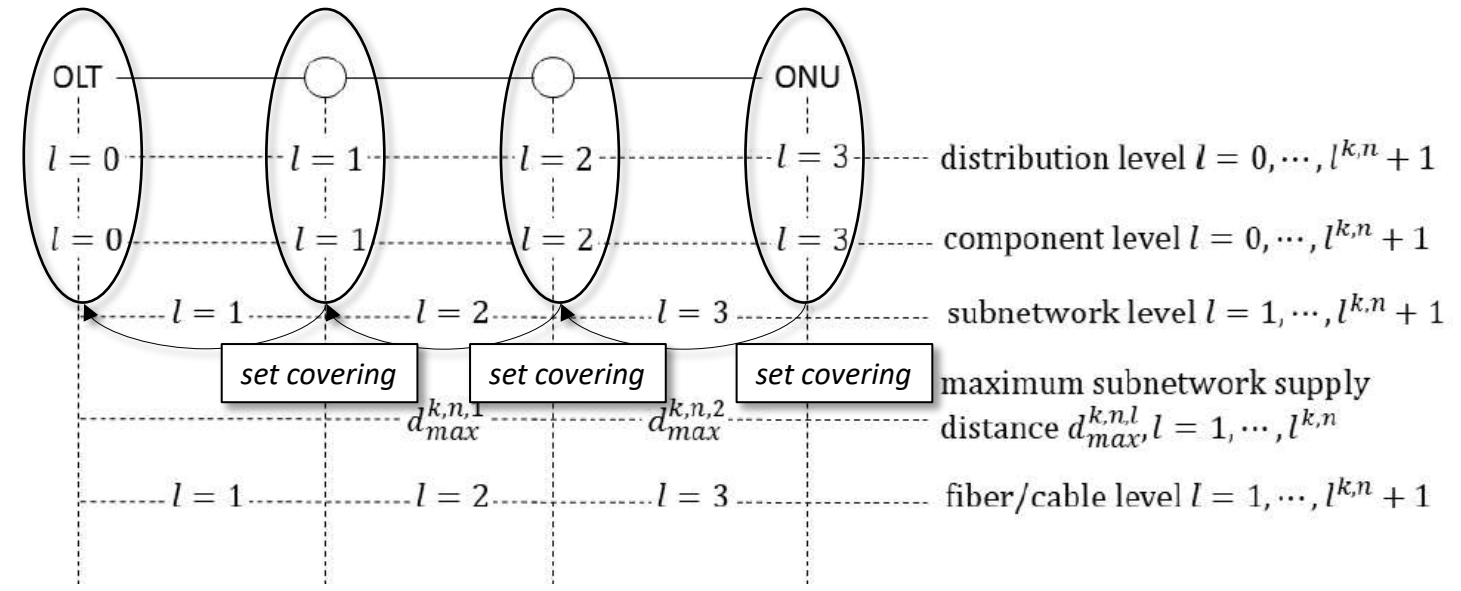
$$\sum_{e \in \delta^+(v)} f_e^{n,l^n+1} - \sum_{e \in \delta^-(v)} f_e^{n,l^n+1} \leq \sum_{t \in SODS_{n,l^n}} \theta_t y_{v,t}^{n,l^n} - \sum_{t \in SONU_{n,l^n}} z_{v,t}^{n,l^n} - \sum_{t \in SONU_{n,l^n}} r_{tt_{v,t}}^{n,l^n}.$$

⋮

Start MILP for fiber model

... solving model directly on network (populate and solve) comes sometimes with **poor performance** ... (its hard to find a first incumbent)

... **initial guess** by bottom up set covering approach (i.e. cover components of subsequent network level by components of current level)



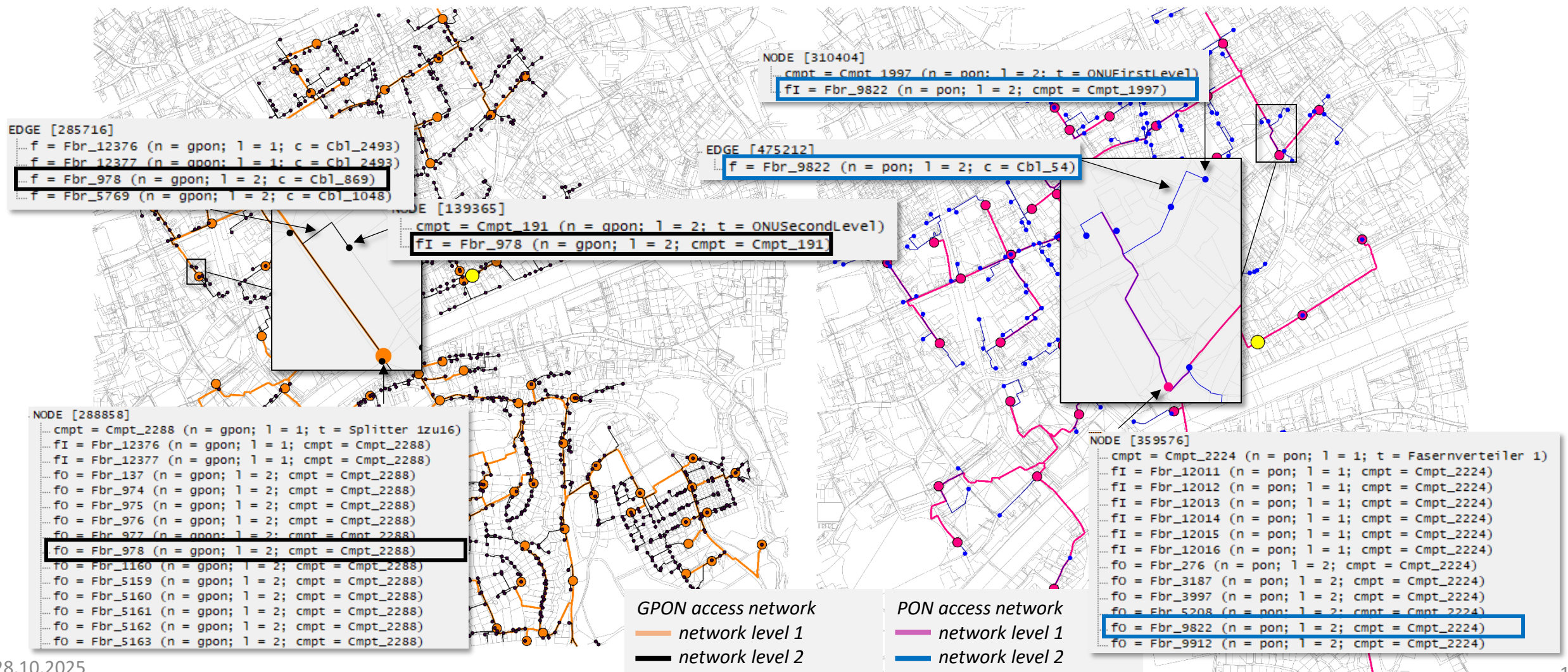
```

variable starts are feasible at objective 458265.33386111
solve MILP for HdlrBaseDataNetwAccess258502FiberMdl ...
| solve model HdlrBaseDataNetwAccess258502FiberMdl --> # vars = 154790 (# fixed = 12952, # starts = 154790);
| obj = 458265.333861 (gap = 71.621790[%], rem. time = 585.56, mean objDiff = NaN%)
|
|
|
| model solution exists; objective = 443534.853194443; solver status = time_limit; gap = 0.262776018640498
    
```



Fiber model result

... just for one access network (note: we have 12 networks in this example instance!)

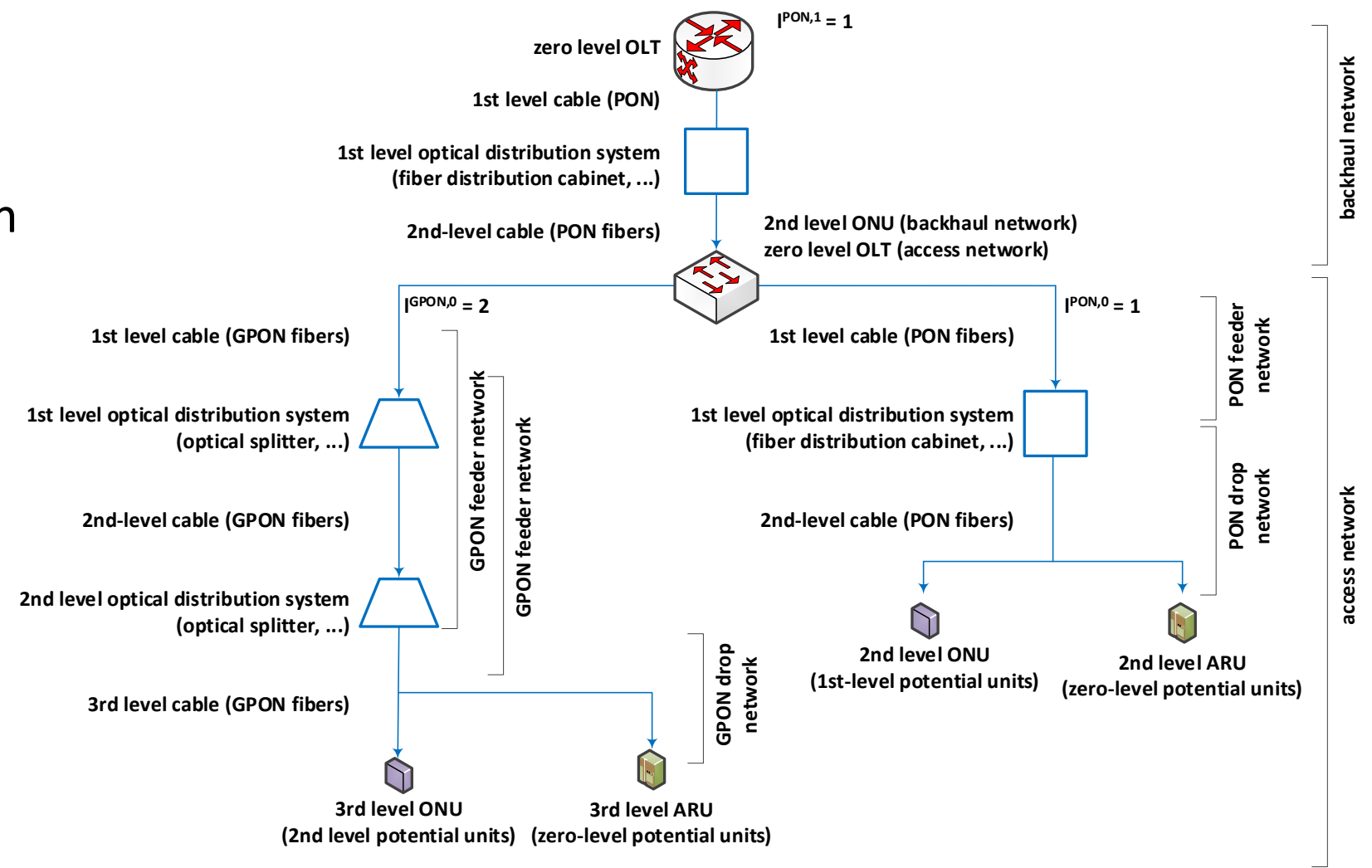


Cable model - specification

... computes **cables** and **cable distributing components** on network trees under consideration of

- fiber information
- component types
- existing and usable infrastructures

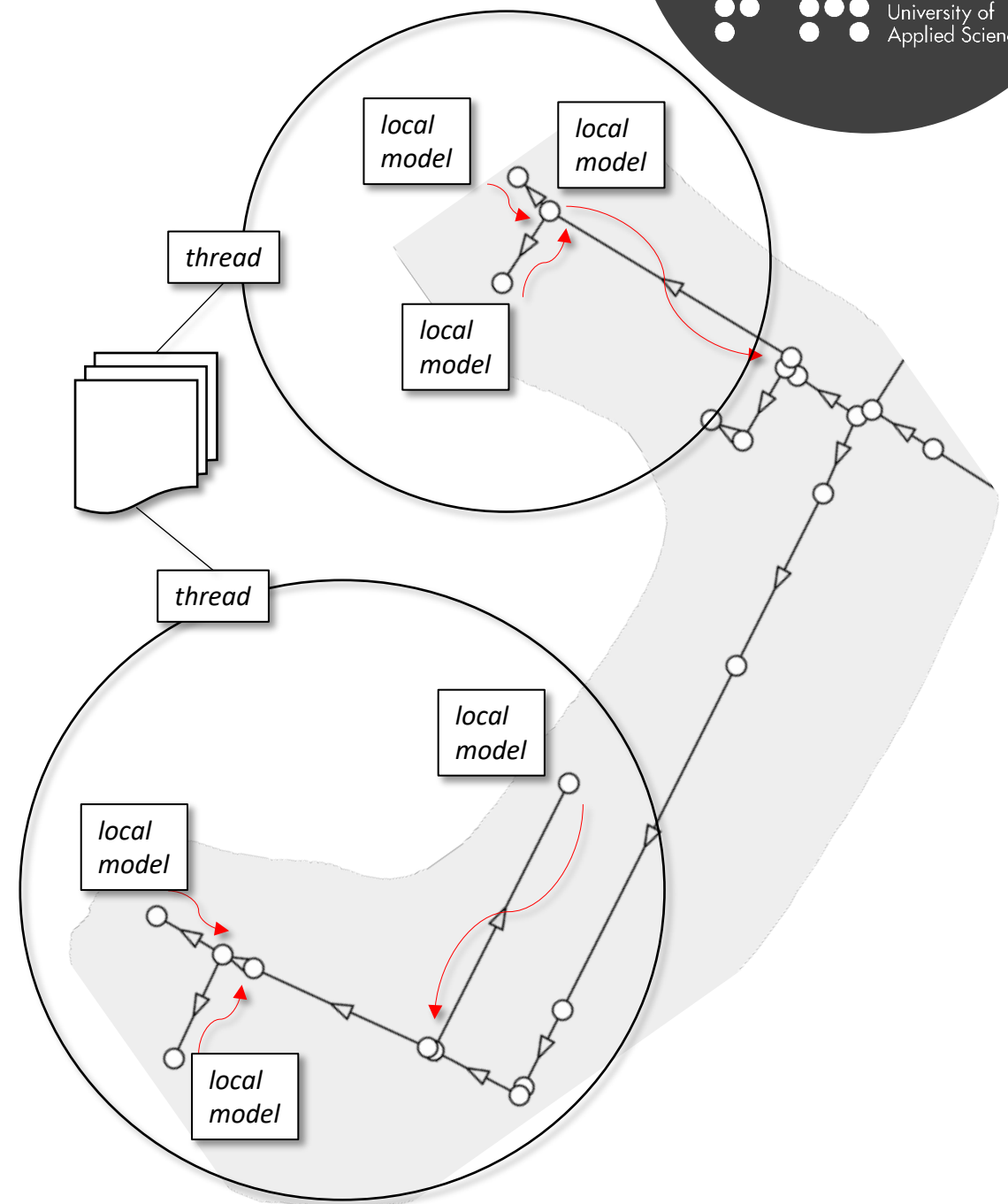
and yields a consistent downstream cable paths



Cable model

... downstream cable paths via cable distributing components which are consistent to the already know fiber paths and consideration of existing infrastructures – **to much complexity just for one model;** hence **perform bottom up local modelling:**

- for all network topologies k
 - for all network levels l in topology k
 - for all nodes in network segment l and topology k , sorted towards the upstream direction
 - do local cable modelling on node and its unique ingoing arc



Local cable model (just some data)

```

:
(99.83%) v = 258584 [secure mode = false, # decvar = 114, # constr = 204, status = optimal, solution exists = true, obj = 41860.800000, gap = 0.000000, cpu = 0.000000]
*****
* DEBUG: v = 354816
*****
data:
e = 462245 = delta_Minus
|   |F_e| = 119
|   F_e = { ... }
|   J_e = {1024977662 [t=K104, rho=0.5, kappa=0.01, comesfrom=ExistingUsableInfrastructure]
|         1026863499 [t=K104, rho=0.166666666666667, kappa=0.01, comesfrom=ExistingUsableInfrastructure]
|         1093803560 [t=PE36, rho=0, kappa=0.01, comesfrom=ExistingUsableInfrastructure]
|         1093803561 [t=PE28, rho=0, kappa=0.01, comesfrom=ExistingUsableInfrastructure]
|         1093803562 [t=PE28, rho=0, kappa=0.01, comesfrom=ExistingUsableInfrastructure]
|         1093803572 [t=PE36, rho=0.33, kappa=0.01, comesfrom=ExistingUsableInfrastructure]
|         TRENCH [t=TRENCH, rho=0, kappa=80, comesfrom=Trenching]}
|   C_e = {Mikrokabel [tau=Mikrokabel, kappa=1, nu=12, rho(1024977662)=0.0208333333333333, rho(1026863499)=0.0208333333333333, rho(1093803560)=0.125, rho(1093803561)=0.25,
|   I_e = {cbl_462245_1 ... cbl_462245_119}
delta_Plus = {723687 723688}
e+ = 723687 in delta_Plus, |F_e+| = 18
e+ = 723688 in delta_Plus, |F_e+| = 101
M_v = {mfl_354816_1 ... mfl_354816_119}
C+ = {cbl_723687_1 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1026863499(K104), F={Fbr_5596}]
      cbl_723687_2 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1026863499(K104), F={Fbr_1115}]
      cbl_723687_3 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1026863499(K104), F={Fbr_8910}]
      cbl_723687_4 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1026863499(K104), F={Fbr_1118 Fbr_1119 Fbr_1120 Fbr_1121 Fbr_1122 Fbr_1123 Fbr_1124 Fbr_1125}
      cbl_723687_5 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1026863499(K104), F={Fbr_1358 Fbr_1359 Fbr_1360 Fbr_1361 Fbr_1362 Fbr_1363}]
      cbl_723687_6 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1026863499(K104), F={Fbr_229}]
      cbl_723688_1 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_8510 Fbr_8511 Fbr_8512 Fbr_8513 Fbr_8514 Fbr_8515}]
      cbl_723688_2 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_4289 Fbr_4290 Fbr_4291 Fbr_4292 Fbr_4293 Fbr_4294 Fbr_4295 Fbr_4296}
      cbl_723688_3 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_8835 Fbr_8836 Fbr_8837 Fbr_8838 Fbr_8839 Fbr_8840 Fbr_8841}]
      cbl_723688_4 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_3728 Fbr_3729 Fbr_3730 Fbr_3731 Fbr_3732}]
      cbl_723688_5 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_3739 Fbr_3740}]

```

119 fibers which have to be installed in cable systems

7 usable infrastructures in which cable systems may be installed or not

1 existing cable system (dark fibers) and its allocation rates

119 cable system options which have to be typed or not

119 muffle options which have to be typed

downstream originating cable systems and their installed fibers (already taken decisions in the bottom up process)

Local cable model (just some data)

```

cbl_723688_6 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_9083}]
cbl_723688_7 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_1364}]
cbl_723688_8 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_4067}]
cbl_723688_9 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_9292}]
cbl_723688_10 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_3071}]
cbl_723688_11 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_231 Fbr_232 Fbr_233 Fbr_234 Fbr_235 Fbr_236 Fbr_237 Fbr_238 Fbr_239 Fbr_240}]
cbl_723688_12 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_255 Fbr_256 Fbr_257 Fbr_258 Fbr_259 Fbr_260 Fbr_261 Fbr_262 Fbr_263 Fbr_264}]
cbl_723688_13 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_3059 Fbr_3060 Fbr_3061 Fbr_3062 Fbr_3063 Fbr_3064 Fbr_3065 Fbr_3066 Fbr_3067}]
cbl_723688_14 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_3034 Fbr_3035}]
cbl_723688_15 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_3037 Fbr_3038 Fbr_3039 Fbr_3040 Fbr_3041 Fbr_3042 Fbr_3043 Fbr_3044 Fbr_3045}]
cbl_723688_16 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_3037 Fbr_3038 Fbr_3039 Fbr_3040 Fbr_3041 Fbr_3042 Fbr_3043 Fbr_3044 Fbr_3045}]
cbl_723688_17 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_3037 Fbr_3038 Fbr_3039 Fbr_3040 Fbr_3041 Fbr_3042 Fbr_3043 Fbr_3044 Fbr_3045}]
cbl_723688_18 [t=Mikrokabel(tau=Mikrokabel, comesfrom=Parameter, nu=12), j=1103175695(PE42), F={Fbr_3037 Fbr_3038 Fbr_3039 Fbr_3040 Fbr_3041 Fbr_3042 Fbr_3043 Fbr_3044 Fbr_3045}]

```

*further downstream originating cable systems and their installed fibers
 (already taken decisions in the bottom up process)*

... obviously not too much data for just one node ... in order to get good performance we filter out non-decisions (e.g. cable system types which are not allocable with respect to given infrastructure ...) ... hence this model is easy to solve:

```

solve model HdlrBaseDataNetwAccess258502_gpon_2_MdlBinary_354816 --> # vars = 746
(# fixed = 0, # starts = 0); # constr = 1373; status = loaded; sol count = 0
model HdlrBaseDataNetwAccess258502_gpon_2_MdlBinary_354816 solved; obj = 102.424000, gap = 0.000000, cpu = 0.016000

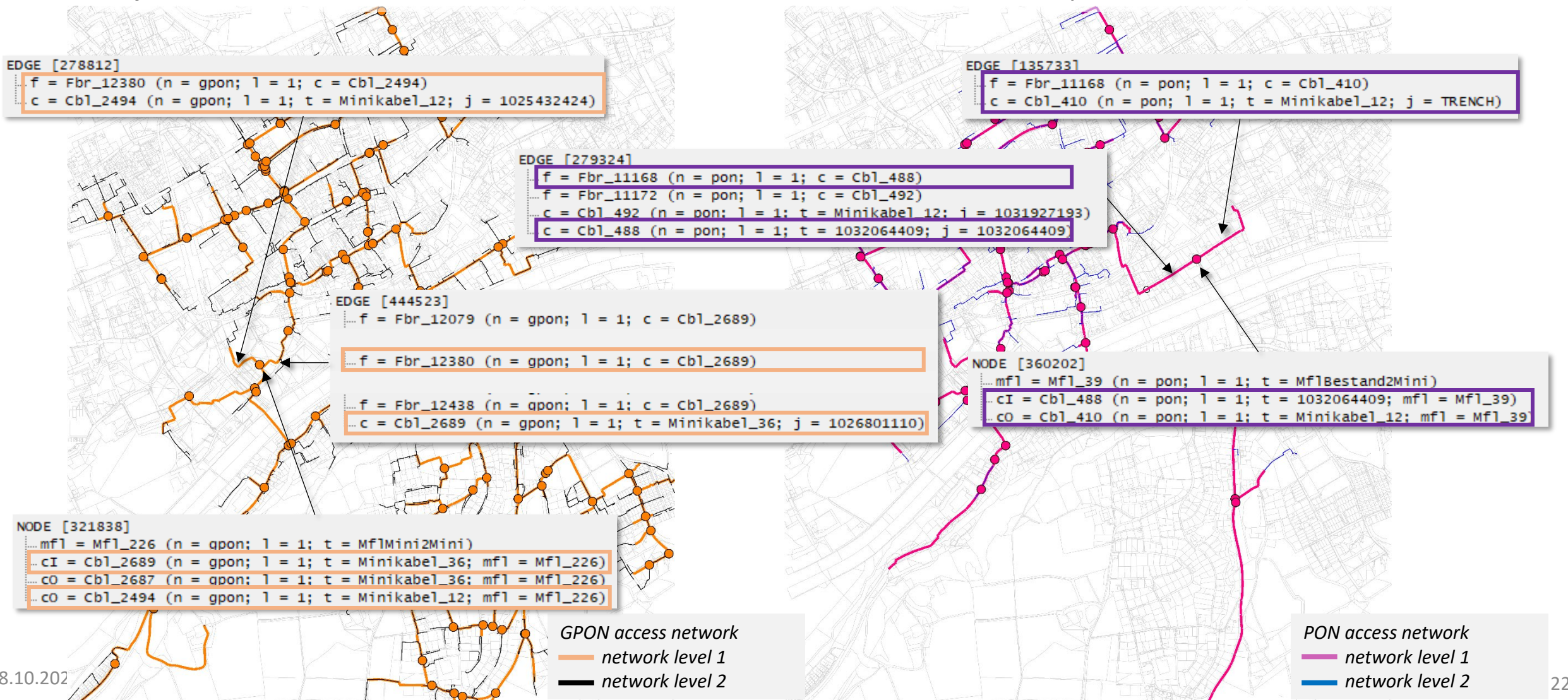
```



... but note - quiet a lot of them have to be solved (in this instance approx. 40000 ones!)

Cable model result

... just for one access network (note: we have 12 networks in this example instance!)



Overall performance

... considering the sketched example with characteristics

```
# node = 381558
# edges = 547096
# connections nodes (PE1 / PE2) = 1124 / 8885
# potential units (PE1 / PE2) = 3142 / 45539
# computed access networks = 11
# distribution levels per access network = 2
network topologies per access network = PON, GPON
# computed backhaul networks = 1
# distribution levels per backhaul network = 0
network topologies per backhaul network = PON
```

```
platform = Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz
          10 physical cores, 20 logical processors
```

```
-----
| SOLVER START (5/29/2024 11:42:34 PM)
|
| preprocessing time = approx. 3 min
|
|-----
| assignment model (5/29/2024 11:45:41 PM)
|-----
|
| assignment model time (two hierarchies) = approx. 11
| min
|
|-----
| initialize network handler (5/29/2024 11:56:57 PM)
|-----
|
| fiber and cable modelling on 12
| networks = approx. 101 min
|
|-----
| SOLVER END (5/30/2024 1:37:04 AM)
|-----
|
| hrs = 1
| min = 54
| sec = 29
```

... but note, that this one of the largest instances ... typical instances are computed within minutes ...

What's the conclusion of all this modelling?

Gurobi Optimizer, as a third party solver for local modelling, is a useful and powerful tool which saves a lot of work ...

...however it is probably not designed for just a “populate and solve” strategy on sophisticated models ...

... hence, provide good quality initial solutions and if not possible simplify, simplify, simplify

MILP solver middle ware

... there are two good reasons for such a tool:

- simplified handling of third party solver
 - supports model population, MILP-Start handling and model access (usage of model outside containers allow access without updating the model)
 - supports conflict management (exploits `grb.feasrelax` in order to generate user feedback for useful parameter changes)
 - allows descent control and user abort (e.g exploits `grb.callback` event information in order to avoid flat spot optimization near the optimum)
 - Allows user / system integrator to handle parallelism of Gurobi Solver threads with respect to parallel NGA solver processes
- one point maintenance (that's the main reason for us!)

NGA solver – bold decisions?

... we think **yes**, since it allows

- automated planning of many different network scenarios in acceptable time
- strategic decisions (relative advantages of investment decisions – e.g. comparison of different levels of network deployments)
- estimation of investment risks and/or funding needs

and

- works on detailed real world data
- offers interfaces to most common geo data sources

NGA solver – is it risky?

... we are not sure, it depends but probably not ...

- solving local models not towards the optimum clearly means to waste money ... but data are usually noisy and it makes no sense to optimize model noise
- planning results are treated as rough first approaches ... they are validated and corrected in detail by experienced (human!) network planners
- for strategic decisions different scenarios are always considered in relation ... hence solving inaccuracies, if they are in the same range, vanish in such a relation ... this is provable non-risky!

**... and finally, let me thank you
for your attention!**