Chair of Communication Networks
Department of Electrical and Computer Engineering
Technical University of Munich

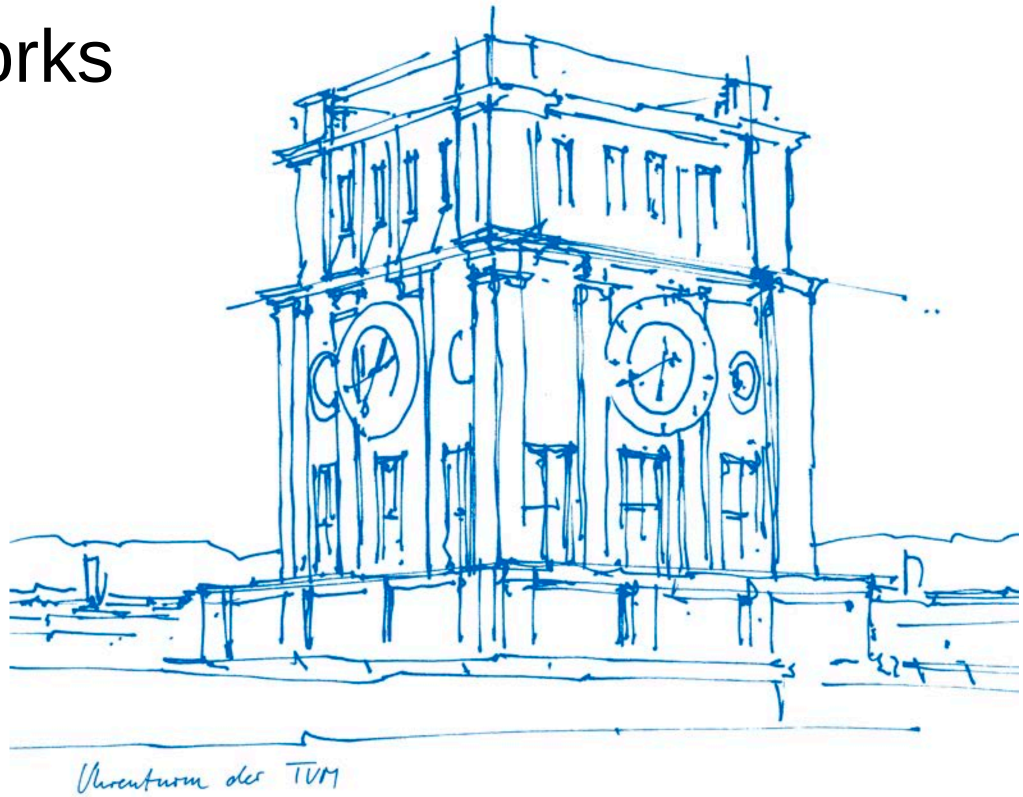
Algorithm-Data Driven Optimization of Adaptive Communication Networks

IEEE ICNP 2017, ML@AI Workshop,
Toronto, Oct. 10, 2017

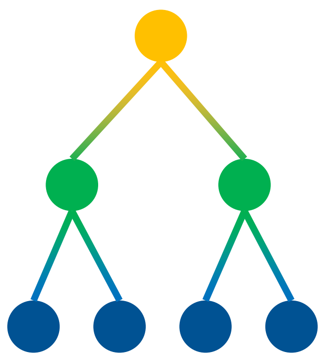
Mu He

Mu.he@tum.de

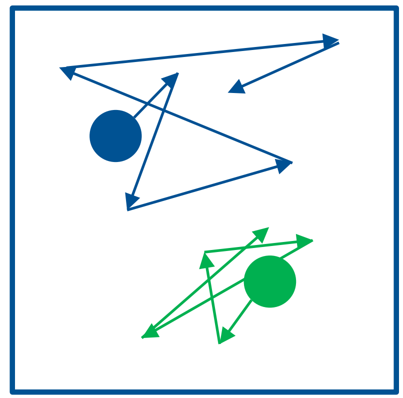
Patrick Kalmbach, Andreas Blenk,
Patrick.kalmbach@tum.de. Andreas.blenk@tum.de
Stefan Schmid, Wolfgang Kellerer
Schmiste@cs.aau.dk, Wolfgang.kellerer@tum.de



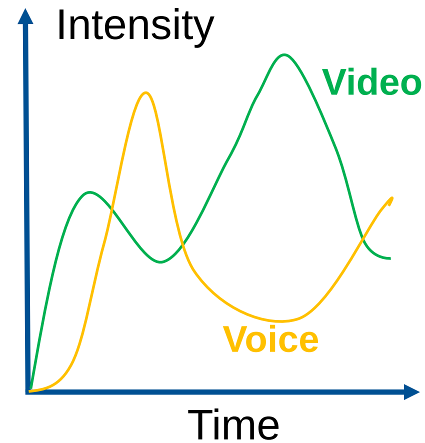
Communication Networks are …



Traffic Patterns



Mobility



Resource Demands

… in constant **flux**.

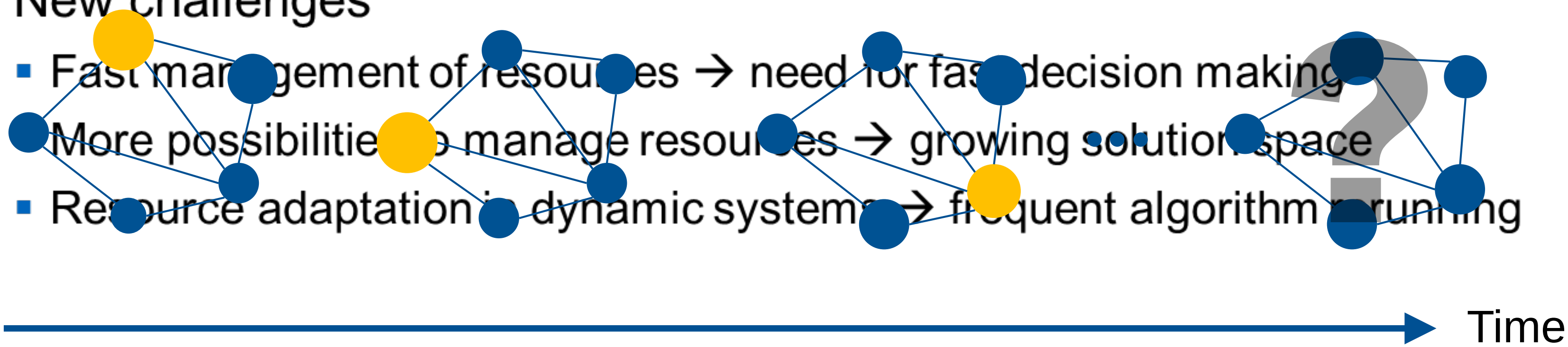
Motivation 1/2

- SDN, NV, NFV claim to provide more flexibility in communication networks, especially in terms of resource allocation.

- New challenges
  - Fast management of resources → need for fast decision making
  - More possibilities to manage resources → growing solution space
  - Resource adaptation in dynamic systems → frequent algorithm rerunning

Motivation 2/2

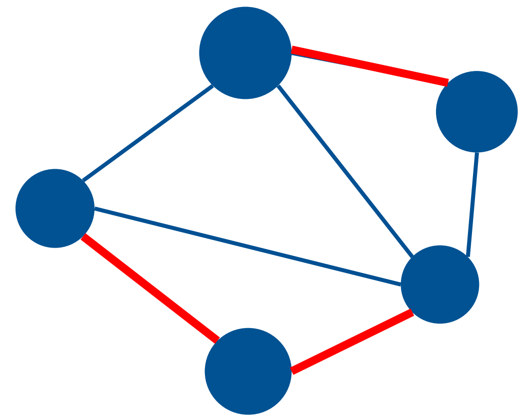
- New challenges
  - Fast management of resources → need for fast decision making
  - More possibilities to manage resources → growing solution space
  - Resource adaptation in dynamic systems → frequent algorithm rerunning



Repeatedly solve problem instances with similar structure

- We need fast and efficient design of optimization systems
  - Can we leverage algorithm data?

The Optimization Problem



- Weighted Controller Placement Problem
- Objective

$$\min \sum_{n \in \mathcal{N}, c \in \mathcal{C}} \mathcal{R}(n) \cdot \mathcal{L}(n,c) \cdot a_{n,c}$$

- Constraints

$$\sum_{c \in \mathcal{C}} p_c = \boldsymbol{k}$$

Total number of controllers to be placed are *k*.

$$\sum_{c \in \mathcal{C}} a_{n,c} = 1, \forall n \in \mathcal{N}$$

A switch is only assigned to one controller.

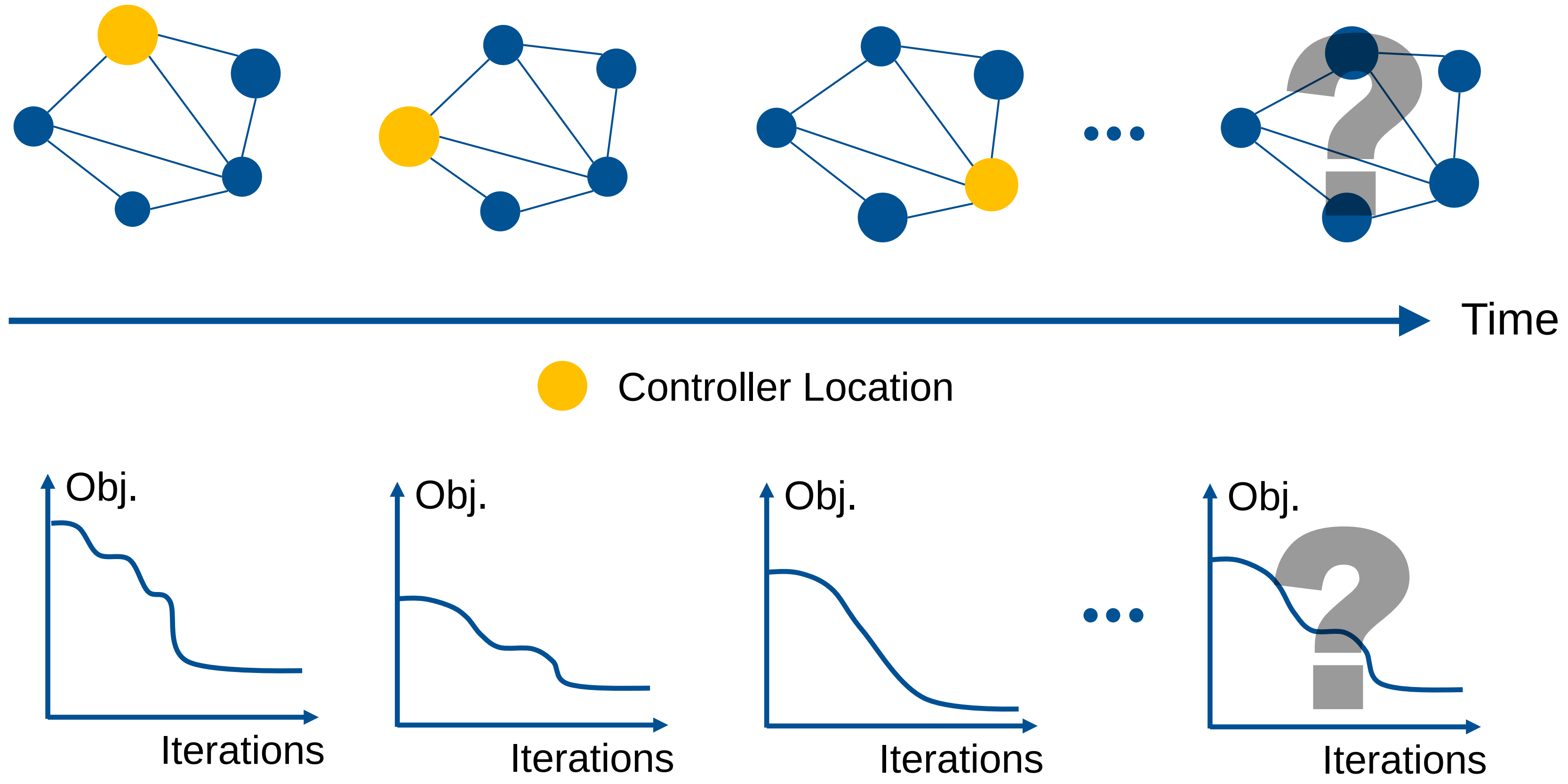
$$\sum_{n \in \mathcal{N}} a_{n,c} \leq |\mathcal{N}| \cdot p_c, \forall c \in \mathcal{C}$$

A switch is only assigned to one controller if that controller has been placed.

$$p_c, a_{n,c} \in \{0,1\}, \forall c \in \mathcal{C}, n \in \mathcal{N}$$

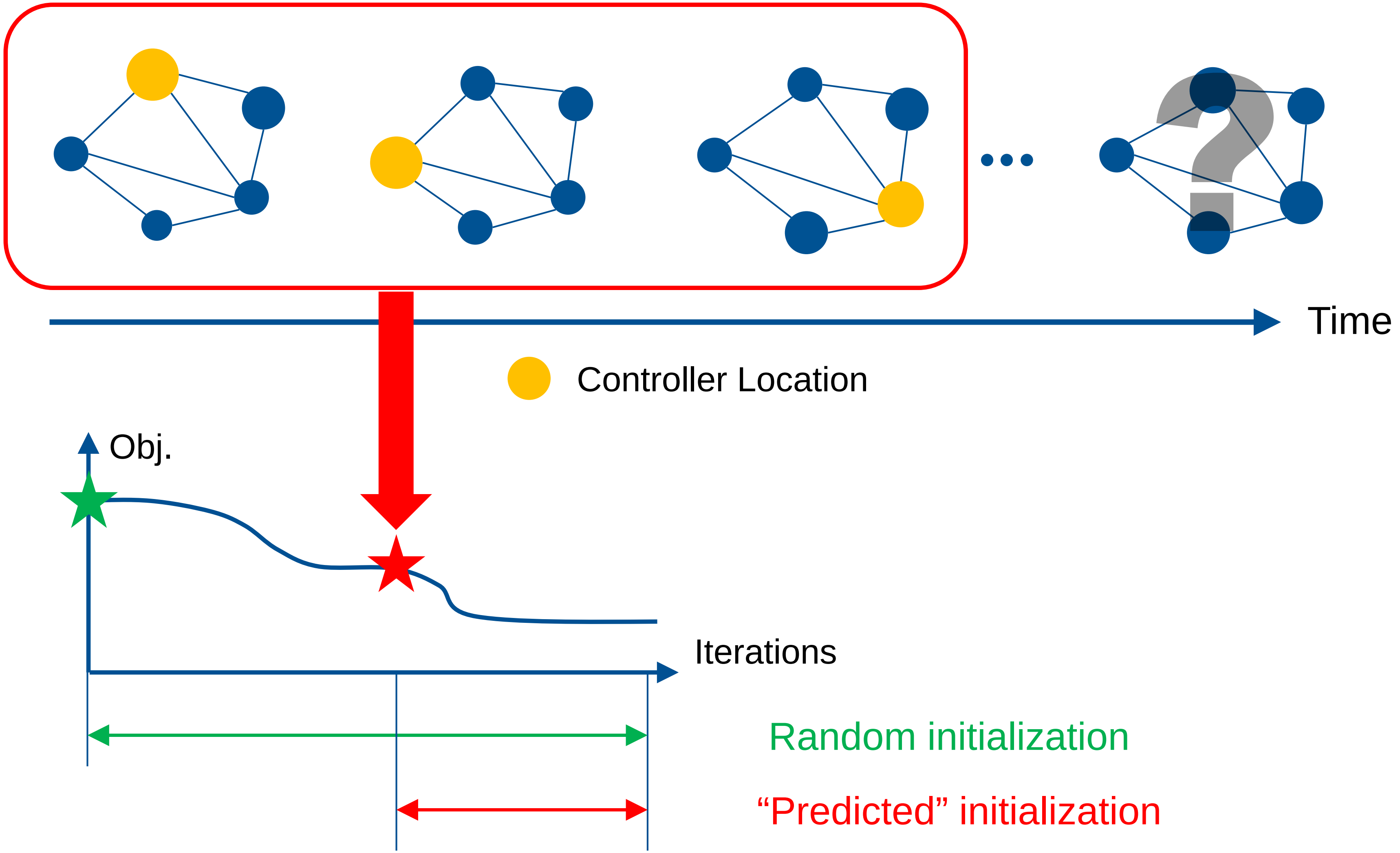
Binary decision variables. One for controller location, the other for switch-controller association.

The Limitation

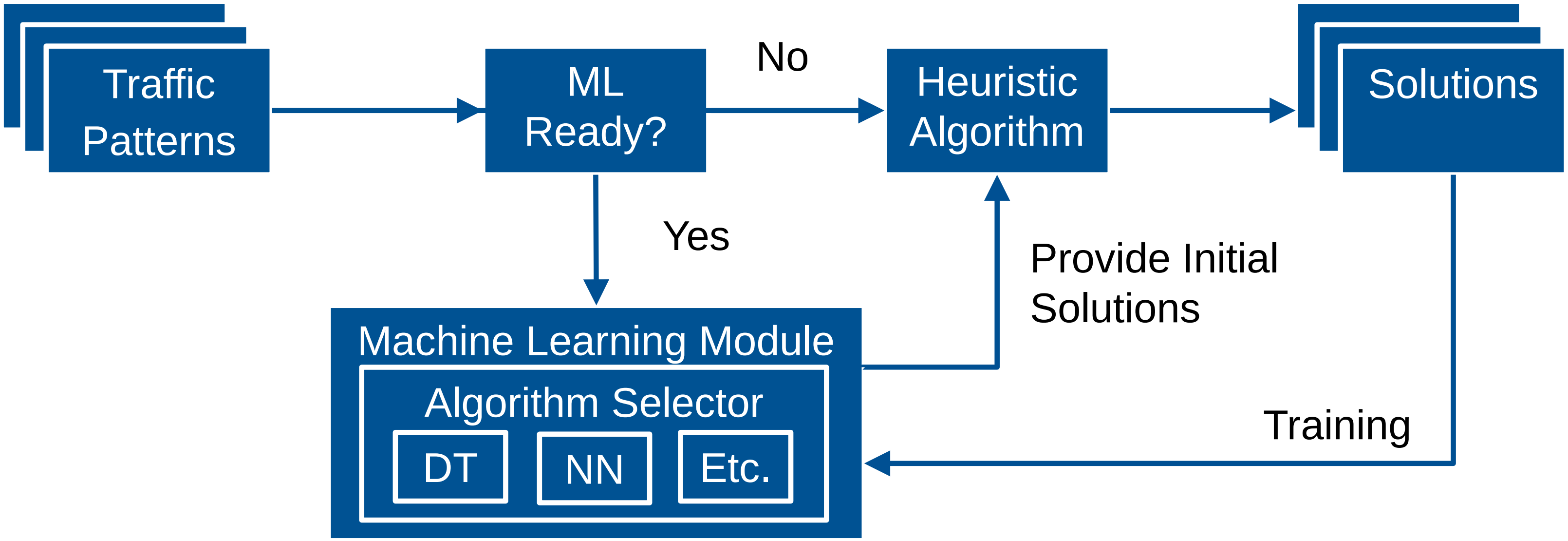


Algorithm always starts from random initial solution (from scratch). And prior solutions are wasted.

The Opportunity



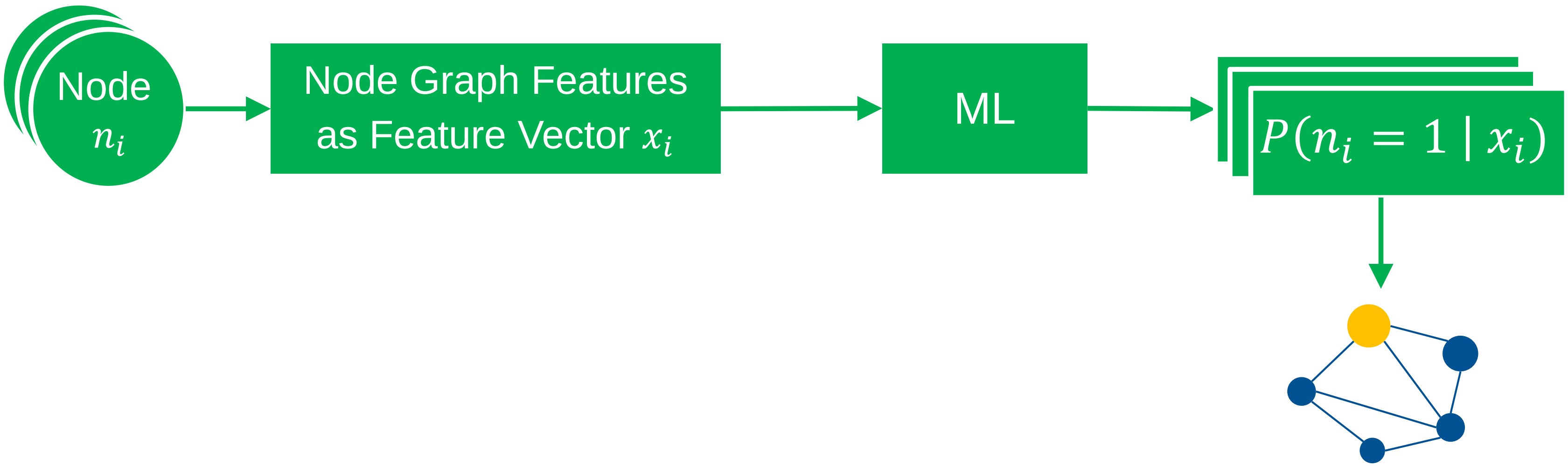
Proposed System Architecture



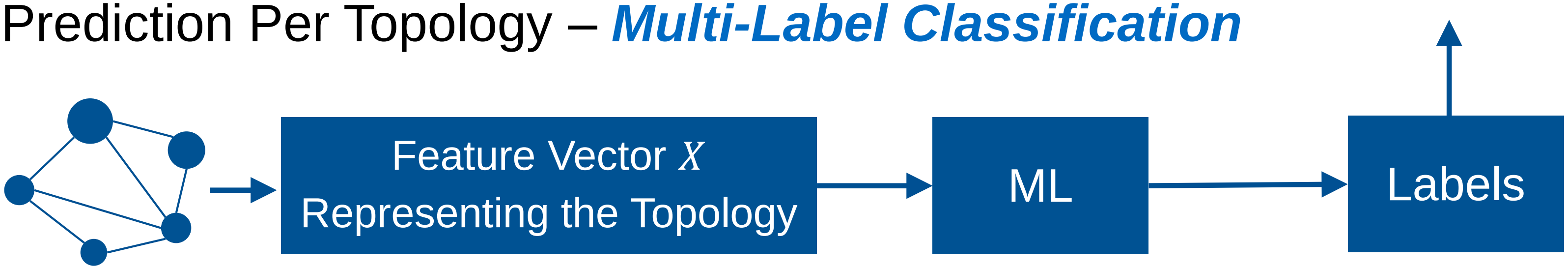
- Little modification to existing system architecture
- Training happens offline

The Machine Learning Approach 1/2

Prediction Per Node – Former Approach [1]



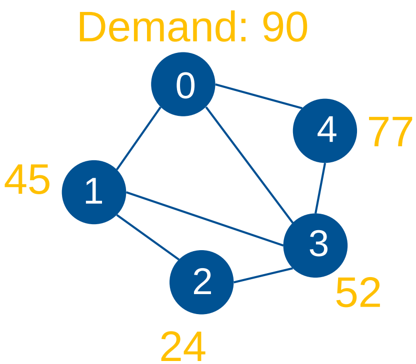
Prediction Per Topology – ***Multi-Label Classification***



[1] A. Blenk, P. Kalmbach, S. Schmid, and W. Kellerer, o'zapft is : Tap your network algorithms big data!," in Proc. of ACM SIGCOMM Big-DAMA, ACM, 2017.

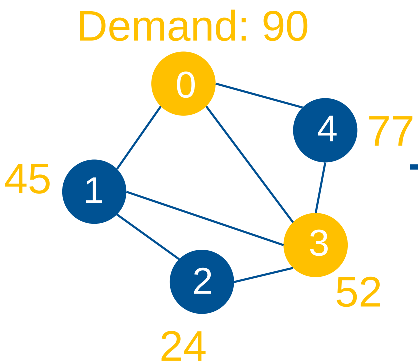
The Machine Learning Approach 2/2

- What is our feature vector?



Traffic Distribution of All Nodes → Feature Vector X $\begin{bmatrix} 90 \\ 45 \\ 24 \\ 52 \\ 77 \end{bmatrix}$

- How do we represent the multiple labels?



→ Labels: {0, 3} → Binary Prediction Vector Y $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$

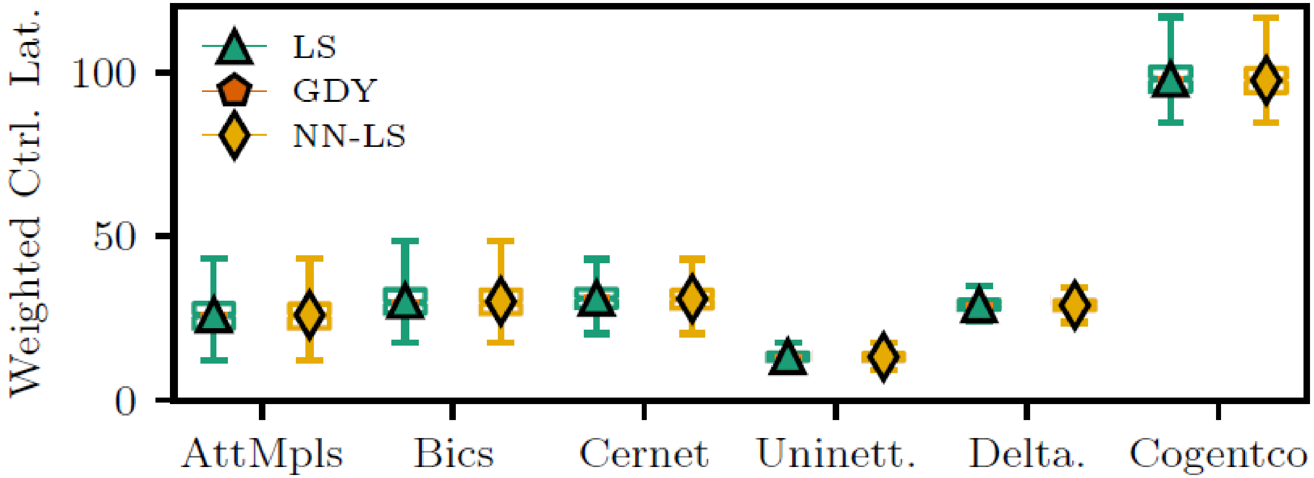
[1] A. Blenk, P. Kalmbach, S. Schmid, and W. Kellerer, o'zapft is : Tap your network algorithms big data!," in Proc. of ACM SIGCOMM Big-DAMA, ACM, 2017.

Evaluation Settings

ML Algorithms	Decision Tree [1], Logistic Regression [1], Neural Network [2]
Data Samples	6500 for training, 500 for evaluation
Evaluation Metrics	Hamming Loss, Objective Function Value
Topologies	From Topology Zoo [3], 25 – 180 nodes
# of Controllers	5, 10, 15, 20
Traffic Distribution	Uniform (1, 100)

[1] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
[2] TensorFlow: https://www.tensorflow.org
[3] Knight et al., The Internet Topology Zoo. *IEEE J. on Sel. Areas in Communica-tions 29*, 9 (2011).

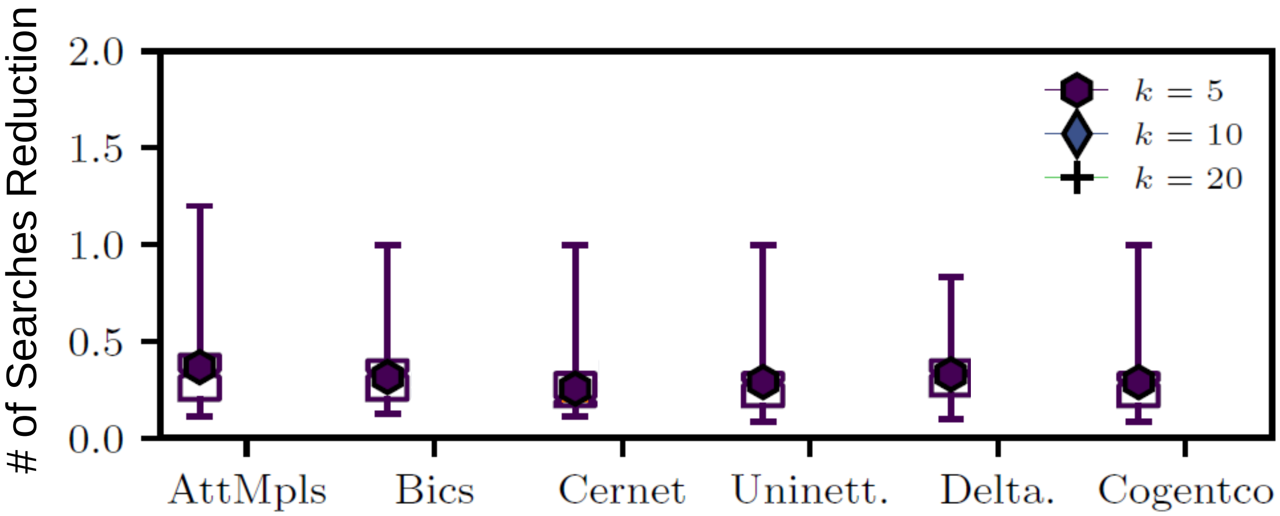
Could we improve the solution quality?



k =10

- Local search with neural network achieves similar performance as plain local search
- Better than reference greedy algorithm

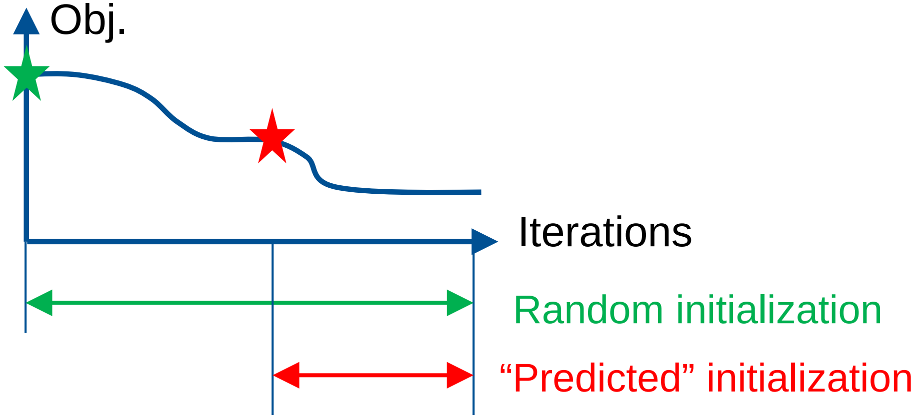
Could we speed up the search?



- Local search with neural network needs only 1/3 the number of searches of plain local search of most samples.
- Efficient search space reduction applies to all tested topologies and different *k*s.

Conclusion

- Learn from past executions of network algorithms
- Multi-label classification → predict initial solutions
  - Different ways to apply ML to a problem with trade-offs
- Algorithm-data driven optimization → performance boost (same solution quality with shorter runtime)



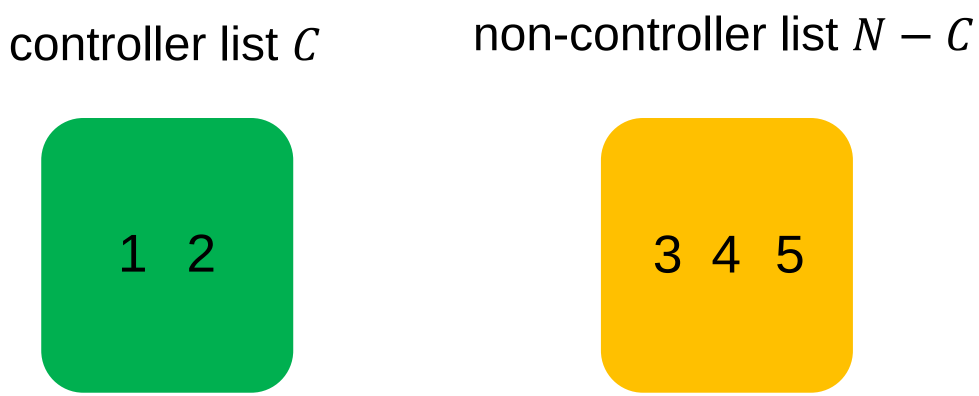
Future Work

- Explore more representative feature vector
- Generalize topologies
- Investigate the performance of more advanced machine learning algorithms
- Apply to other network optimization use cases

Questions?

Heuristic: Local Search

- Algorithm intuition
  - Two lists : controller list C and non-controller list $N - C$
  - Random initialization: C and $N - C$
  - Swap one element in C and one element in $N - C$ → One local search move
  - Stop condition: no more swaps can further improve the objective

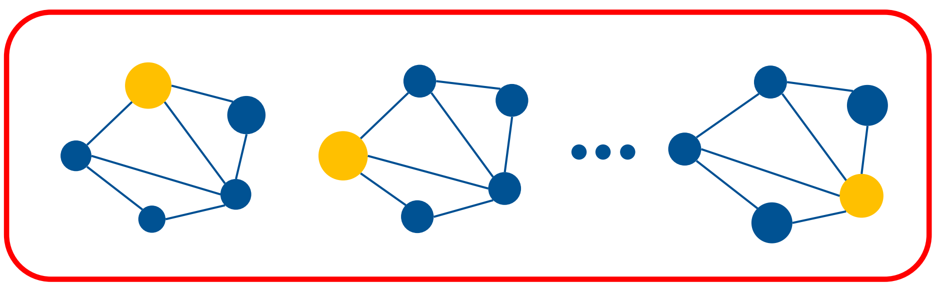


The Machine Learning Approach - Backup

- What machine learning algorithms can be used?
  - Decision tree, logistic regression, neural network, etc.

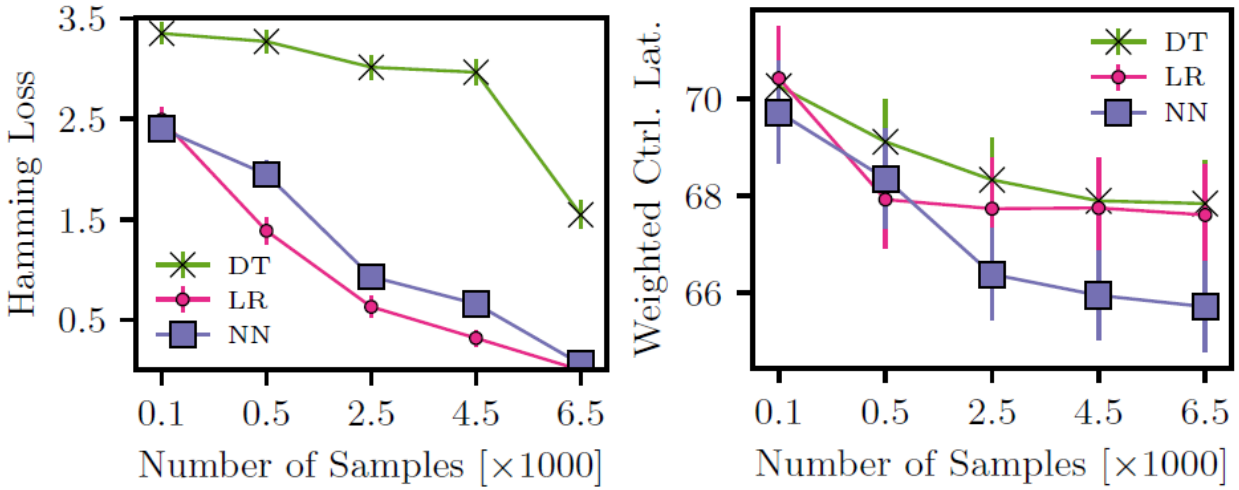
- Which cost function?
  - Bernoulli cross entropy loss

- How do we train the classifiers?



Problem instances solved by the heuristic as training data

Could we learn from the past solutions?

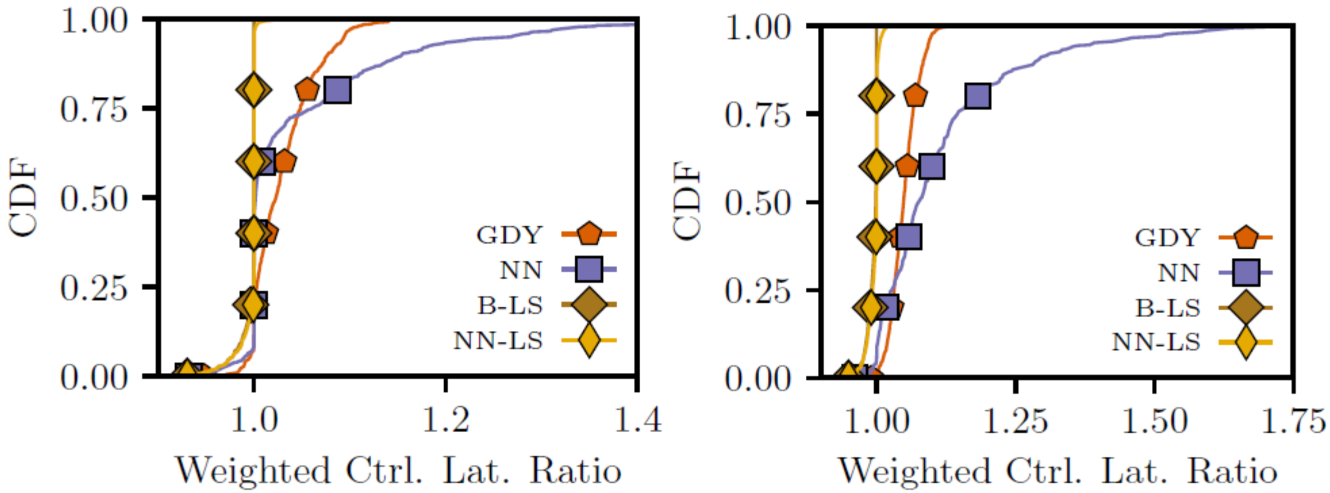


(a) Prediction accuracy (b) Objective function

- Prediction is more accurate for more samples.
- ML evaluation metrics do not fully represent performance.
  - Mispredictions lead to different obj. function values.
- Neural network performs the best in most cases.

Evaluation Results

What are the benefits of applying machine learning in heuristics?



(a) Uninett2010, $\boldsymbol{k} = 5$ (b) Deltacom, $\boldsymbol{k} = 10$

- Neural network prediction does not always guarantee good performance.
  - (a): 60% percent good solutions
  - (b): only 25% are good