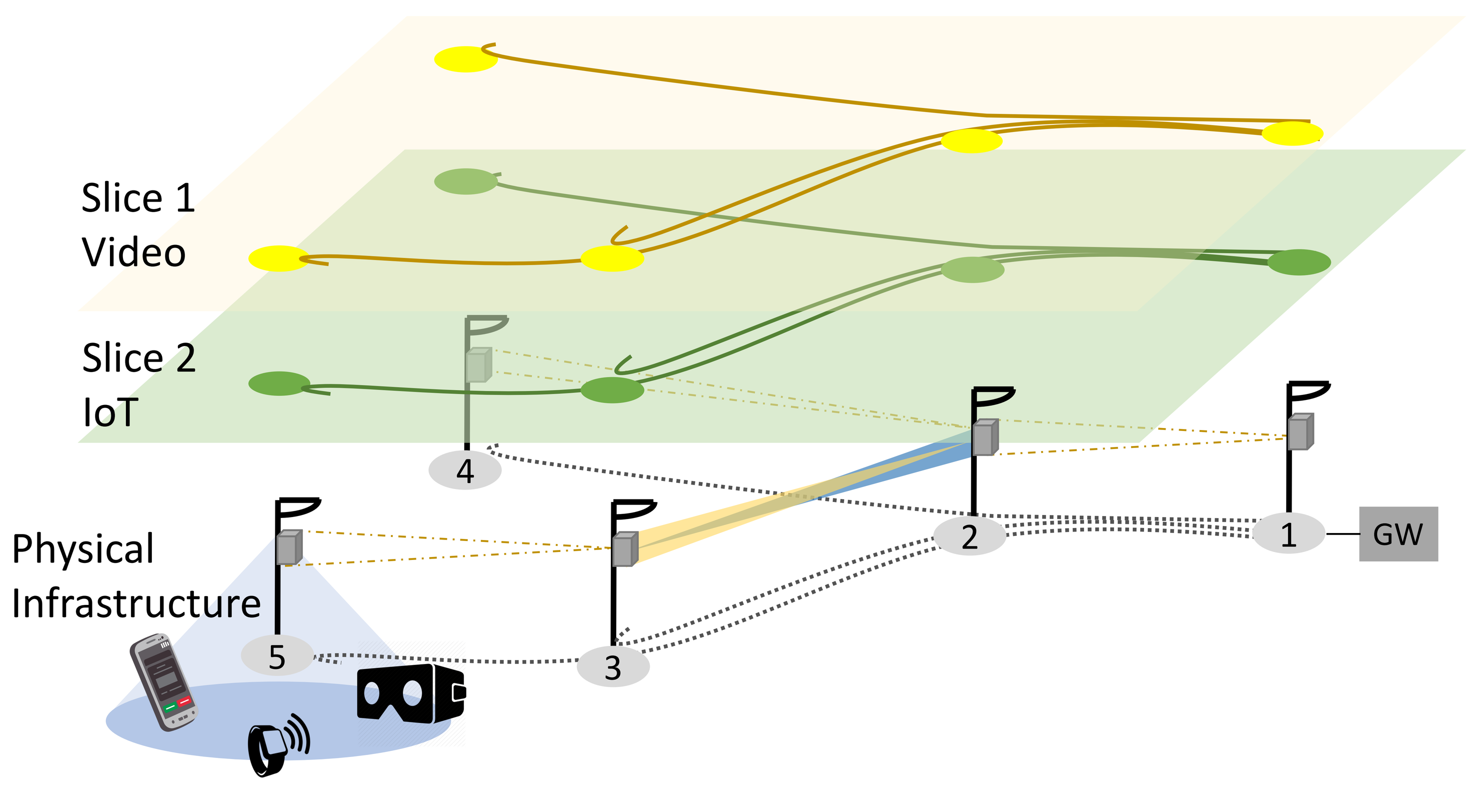
A Deep Learning Approach to Maximising the Utility of 5G Backhaul Networks

THE UNIVERSITY *of* EDINBURGH informatics

Rui Li, Chaoyun Zhang, Pan Cao Paul Patras and John S. Thompson | School of Informatics and School of Engineering, The University of Edinburgh | Email: rui.li@ed.ac.uk | Webpage:

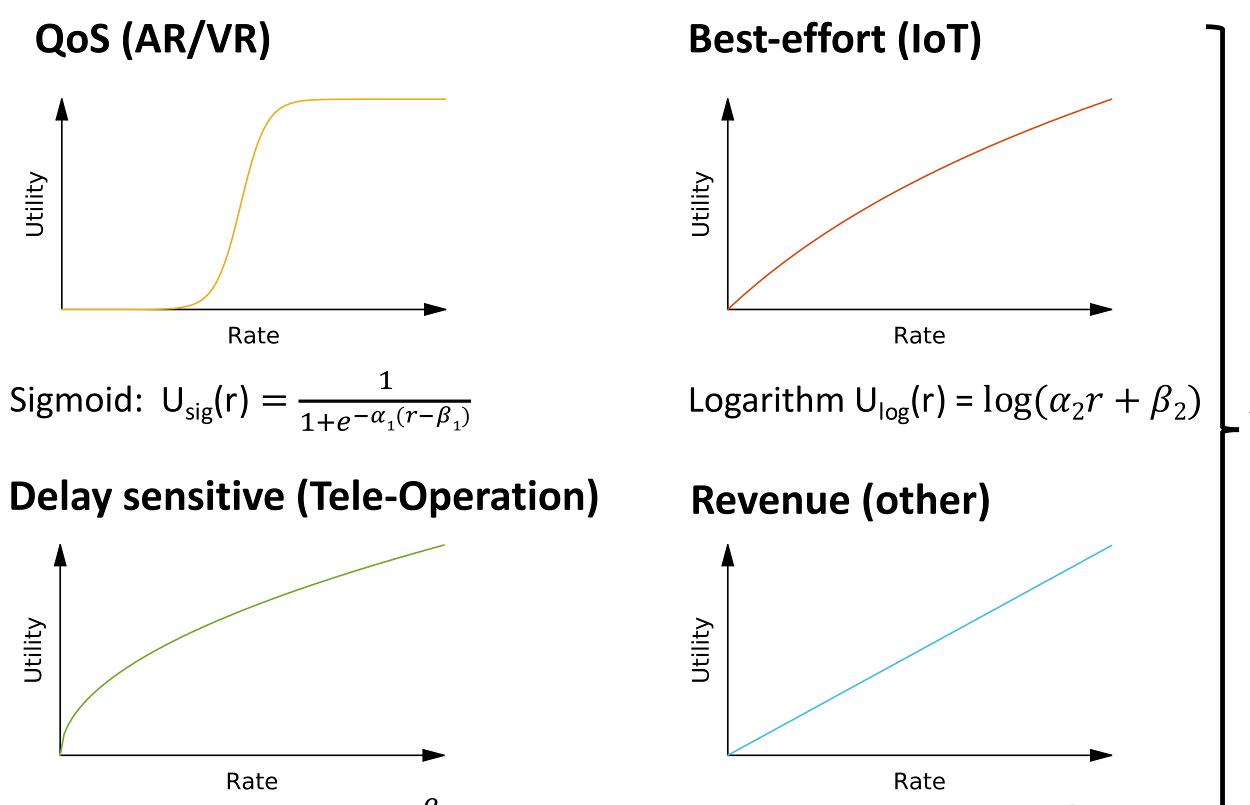
5G Networks

- Ramified use cases and distinct performance requirements
- Network virtualisation and densification
- Millimetre-wave (mm-wave) technologies
  - Multi-Gbps link rates -> Tangible backhauling solution



Utility Framework

- Application scenarios -> utility functions
- Combing all known types of utility functions



Sigmoid: $U_{sig}(r) = \frac{1}{1+e^{-\alpha_1(r-\beta_1)}}$

Logarithm $U_{log}(r) = \log(\alpha_2 r + \beta_2)$

Polynomial: $U_{ply}(r) = \alpha_3(r^{\beta_3})$

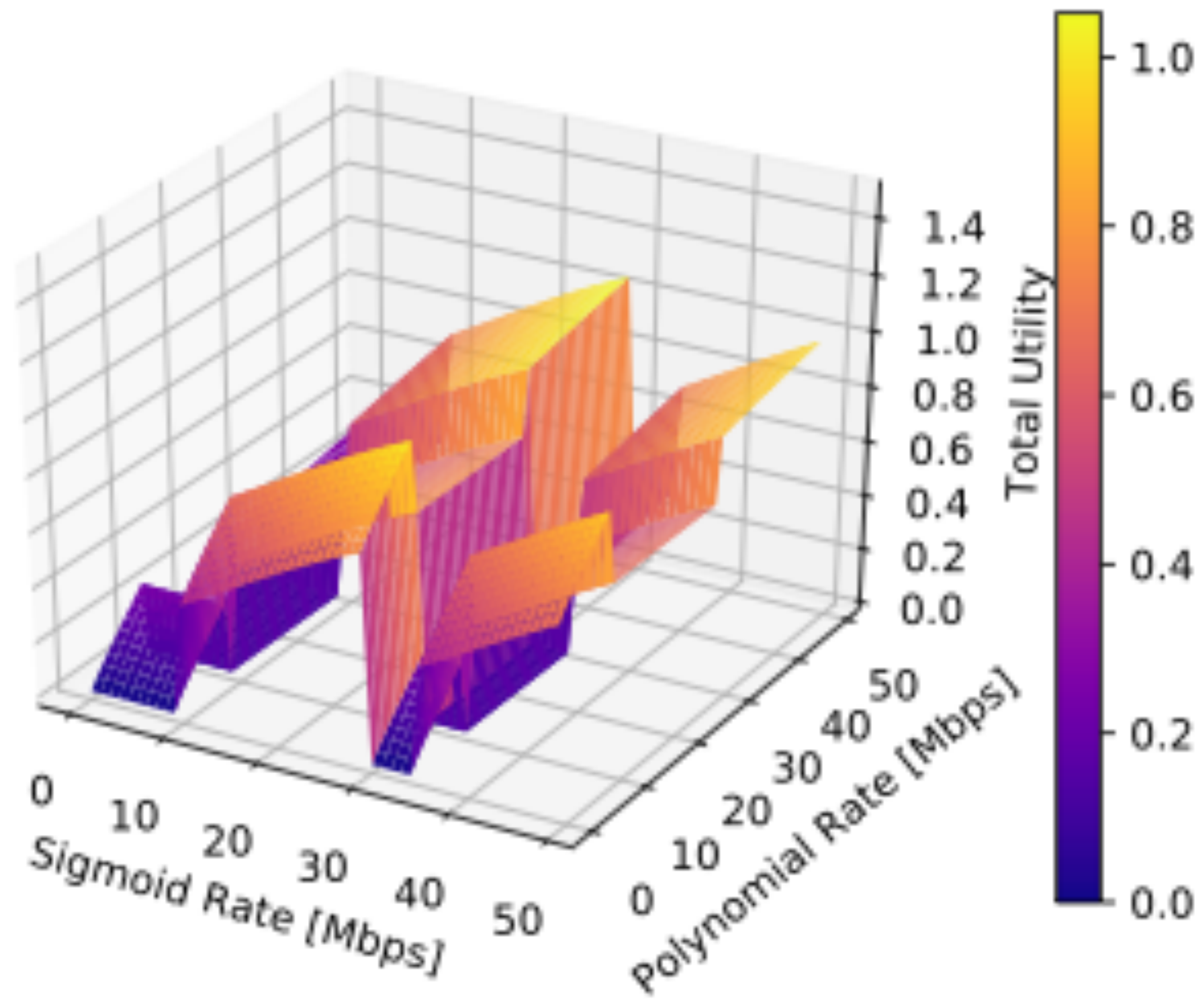
Linear $U_{lin}(r) = \alpha_4 r + \beta_4$

$max \sum U_i$

Finding the Optimal Rate Allocation

- High-dimensional highly non-convex problem
- Global search can be time consuming
- Heuristic method can solve but sub-optimal

Acknowledgement: We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research

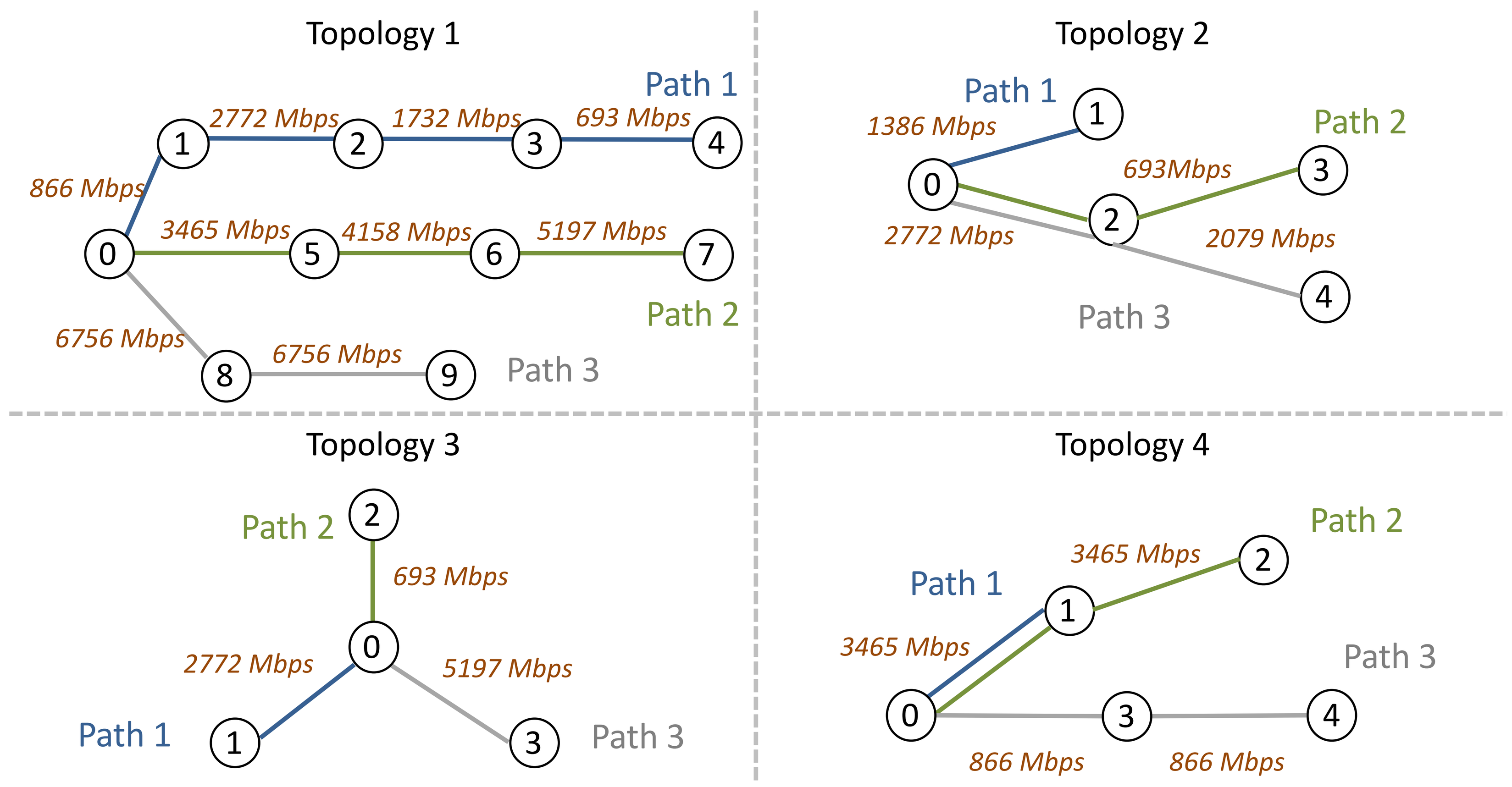


The Deep Learning Approach

- Supervised learning with convolutional neural network (CNN)
  - 10 stacks of convolutional layers + batch normalisation + SeLU
  - Input: Flow demands ($d_{i,j}$) and minimum service rates ($\delta_{i,j}$)
  - Output: Predicted flow rate ($r_{i,j}$)

Numerical Analysis

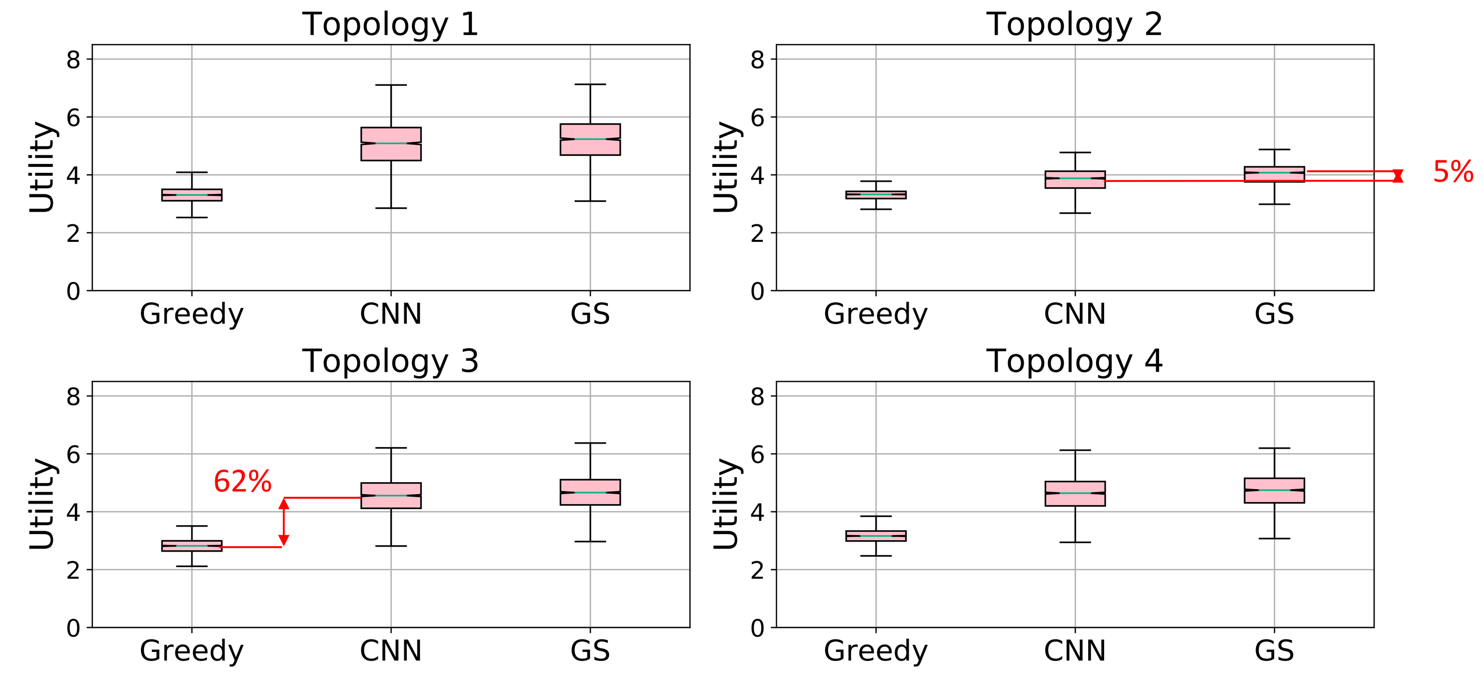
- 10,000 data points $\{d_{i,j}, \delta_{i,j}, r_{i,j}\}$
  - Optimal solutions obtained from Global Search (GS)*
  - Benchmark: Light-weight greedy solution



* Optimality of GS is proven in Z. Ugray et al. Scatter search and local NLP solvers: *A multistart framework for global optimization*. Journal on Computing, 19(3), 2007.

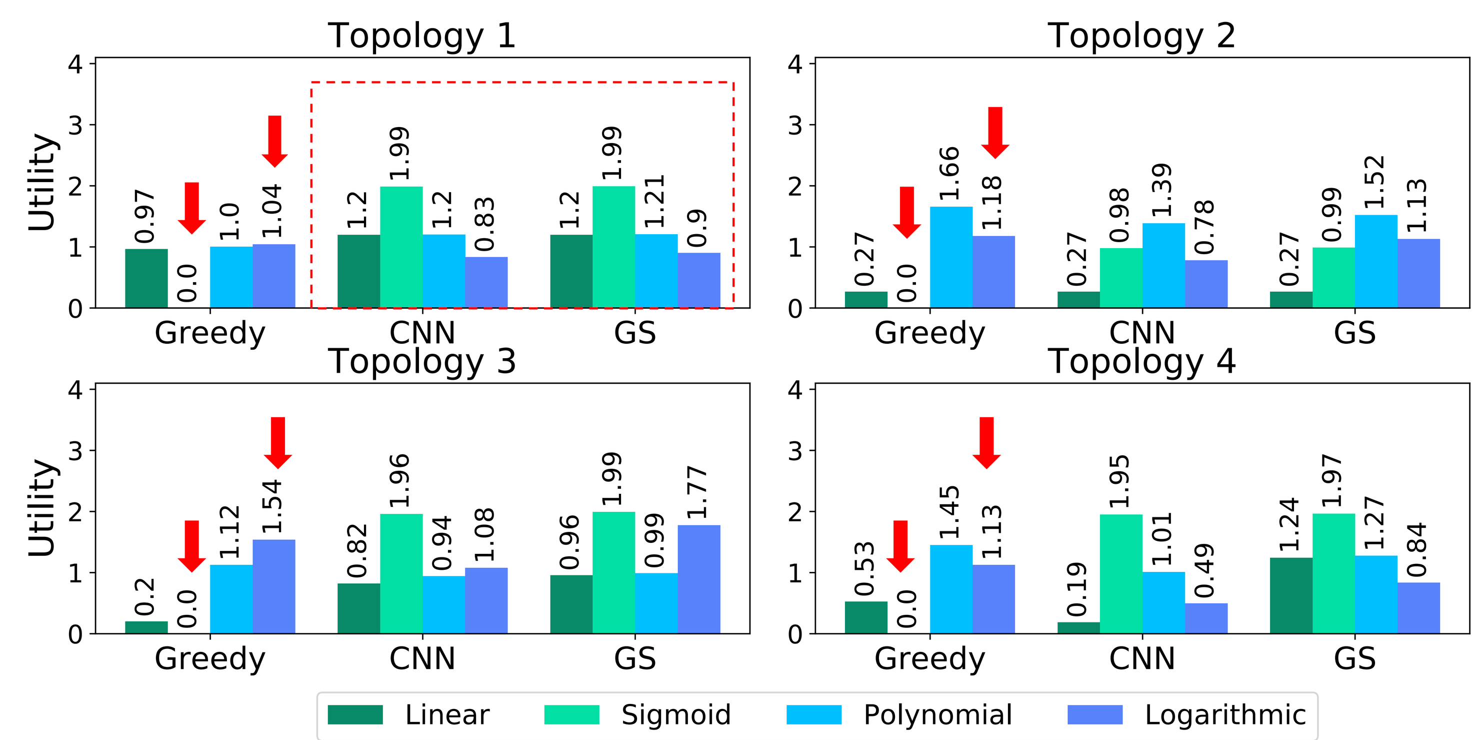
Performance

- Total network utility distributions over 2k instances



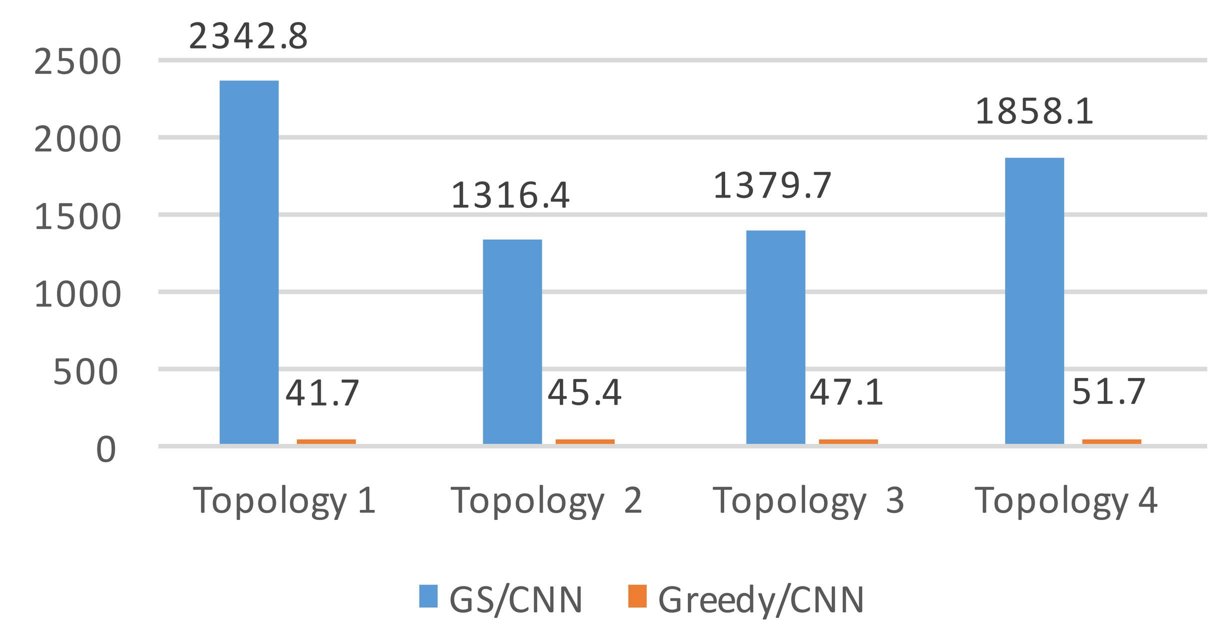
  - CNN: close to optimal in terms of median, quartiles, etc.
  - Achieves up to 62% total utility gain over greedy

- Per type utility allocation in a single instance



  - CNN attains close to optimal allocations
  - Greedy tends to starve Sigmoidal utilities

- Computation Time



  - 2000x faster than GS
  - 50x faster than Greedy