

Traffic Assignment and Network Design

OR 2023

Kartikey Sharma

Joint work with Mathieu Besancon,
Deborah Hendrych and Sebastian Pokutta





Overview

Overview

Aim: Solve the Network Design problem for the Traffic Assignment Model

Overview

Aim: Solve the Network Design problem for the Traffic Assignment Model

- Introduction

Overview

Aim: Solve the Network Design problem for the Traffic Assignment Model

- Introduction
- Model and Algorithms

Overview

Aim: Solve the Network Design problem for the Traffic Assignment Model

- Introduction
- Model and Algorithms
- Numerical Experiments



Motivation

Motivation

- Boscia.jl library leverages Frank-Wolfe to solve discrete optimization problems with convex objectives using MIPs to compute the descent direction

Motivation

- Boscia.jl library leverages Frank-Wolfe to solve discrete optimization problems with convex objectives using MIPs to compute the descent direction
- Network Design adds a binary problem on top of a network

Motivation

- Boscia.jl library leverages Frank-Wolfe to solve discrete optimization problems with convex objectives using MIPs to compute the descent direction
- Network Design adds a binary problem on top of a network
- For a fixed network, the underlying model can be solved efficiently

Motivation

- Boscia.jl library leverages Frank-Wolfe to solve discrete optimization problems with convex objectives using MIPs to compute the descent direction
- Network Design adds a binary problem on top of a network
- For a fixed network, the underlying model can be solved efficiently
- We leverage this special structure present for efficient computation of these oracles

Traffic Assignment Model

Traffic Assignment Model

- Traffic Assignment models the equilibrium traffic flow between various origins and destinations

$$G = (V, E)$$

d_i^t : flow from origin i to destination t

Traffic Assignment Model

- Traffic Assignment models the equilibrium traffic flow between various origins and destinations
- Equivalent to a network flow problem with convex costs

$$G = (V, E)$$

d_i^t : flow from origin i to destination t

$$\min_x c(x)$$

$$\text{s.t. } x_{ij} = \sum_{d \in \mathcal{D}} x_{ij}^d$$

$$\sum_{(i,j) \in \mathcal{E}} x_{ij}^t - \sum_{(j,i) \in \mathcal{E}} x_{ji}^t = D_i^t$$

$$x_{ij}^t \geq 0$$

$$D_i = \begin{cases} 0 & \text{no net flow at } i \text{ for } t \\ d_i^t & \text{supply from } i \text{ to } t \\ -\sum_{k \in \mathcal{O}} d_k^t & \text{total supply for } t \text{ at } i = t \end{cases}$$

Traffic Assignment Model

- Traffic Assignment models the equilibrium traffic flow between various origins and destinations
- Equivalent to a network flow problem with convex costs
- Solved using the Frank Wolfe algorithm.

$$G = (V, E)$$

d_i^t : flow from origin i to destination t

$$\min_x c(x)$$

$$\text{s.t. } x_{ij} = \sum_{d \in \mathcal{D}} x_{ij}^d$$

$$\sum_{(i,j) \in \mathcal{E}} x_{ij}^t - \sum_{(j,i) \in \mathcal{E}} x_{ji}^t = D_i^t$$

$$x_{ij}^t \geq 0$$

$$D_i = \begin{cases} 0 & \text{no net flow at } i \text{ for } t \\ d_i^t & \text{supply from } i \text{ to } t \\ -\sum_{k \in \mathcal{O}} d_k^t & \text{total supply for } t \text{ at } i = t \end{cases}$$

Traffic Assignment Model

- Traffic Assignment models the equilibrium traffic flow between various origins and destinations
- Equivalent to a network flow problem with convex costs
- Solved using the Frank Wolfe algorithm.
- Linear oracle solved using a sequence of shortest path problems

$$G = (V, E)$$

d_i^t : flow from origin i to destination t

$$\min_x c(x)$$

$$\text{s.t. } x_{ij} = \sum_{d \in \mathcal{D}} x_{ij}^d$$

$$\sum_{(i,j) \in \mathcal{E}} x_{ij}^t - \sum_{(j,i) \in \mathcal{E}} x_{ji}^t = D_i^t$$

$$x_{ij}^t \geq 0$$

$$D_i = \begin{cases} 0 & \text{no net flow at } i \text{ for } t \\ d_i^t & \text{supply from } i \text{ to } t \\ -\sum_{k \in \mathcal{O}} d_k^t & \text{total supply for } t \text{ at } i = t \end{cases}$$

Traffic Assignment and Network Design

Traffic Assignment and Network Design

- Implements the Network Design on top of the traffic assignment model

$$\min_x c(x) + r^\top y$$

$$\text{s.t. } x \in \mathcal{X}$$

$$x_j \leq M y_j \quad \forall j \in J$$

$$y_j \in \{0, 1\}$$

Traffic Assignment and Network Design

- Implements the Network Design on top of the traffic assignment model
- Goal: find a new arc which reduces the overall traffic cost at equilibrium

$$\min_x c(x) + r^\top y$$

$$\text{s.t. } x \in \mathcal{X}$$

$$x_j \leq M y_j \quad \forall j \in J$$

$$y_j \in \{0, 1\}$$

Traffic Assignment and Network Design

- Implements the Network Design on top of the traffic assignment model
- Goal: find a new arc which reduces the overall traffic cost at equilibrium
- Leads to a discrete optimization problem with convex costs

$$\begin{aligned} \min_x \quad & c(x) + r^\top y \\ \text{s.t.} \quad & x \in \mathcal{X} \end{aligned}$$

$$\begin{aligned} x_j &\leq M y_j \quad \forall j \in J \\ y_j &\in \{0, 1\} \end{aligned}$$

Traffic Assignment and Network Design

- Implements the Network Design on top of the traffic assignment model
- Goal: find a new arc which reduces the overall traffic cost at equilibrium
- Leads to a discrete optimization problem with convex costs
- Due to graphs and multiple sources and destinations, the problem size is large

$$\begin{aligned} \min_x \quad & c(x) + r^\top y \\ \text{s.t.} \quad & x \in \mathcal{X} \end{aligned}$$

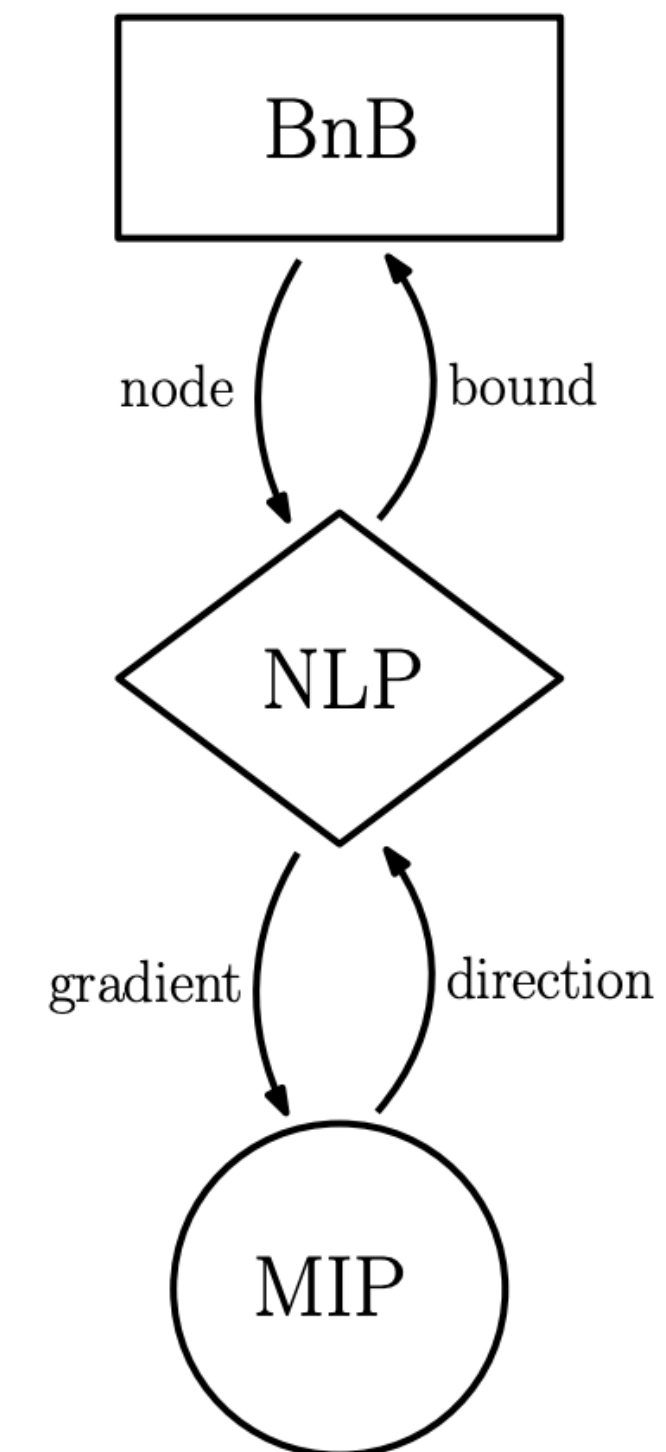
$$x_j \leq M y_j \quad \forall j \in J$$

$$y_j \in \{0, 1\}$$

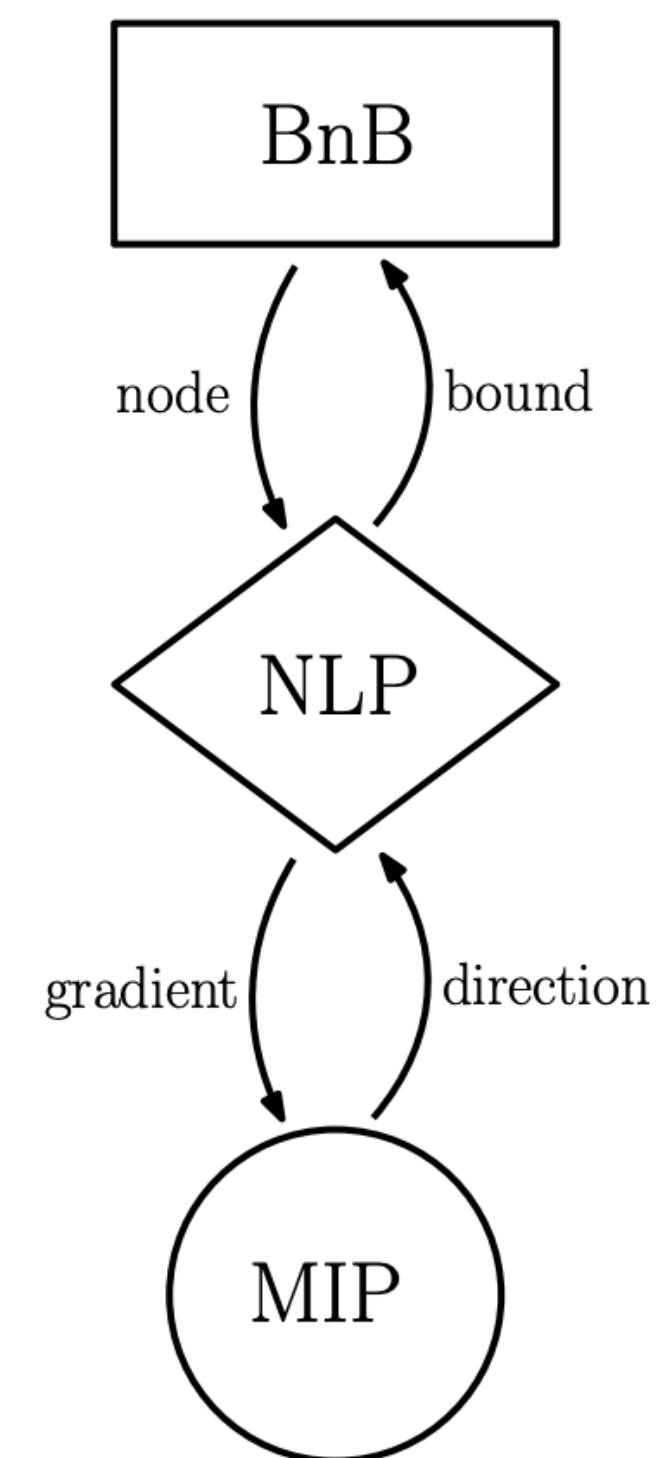


Boscia.jl

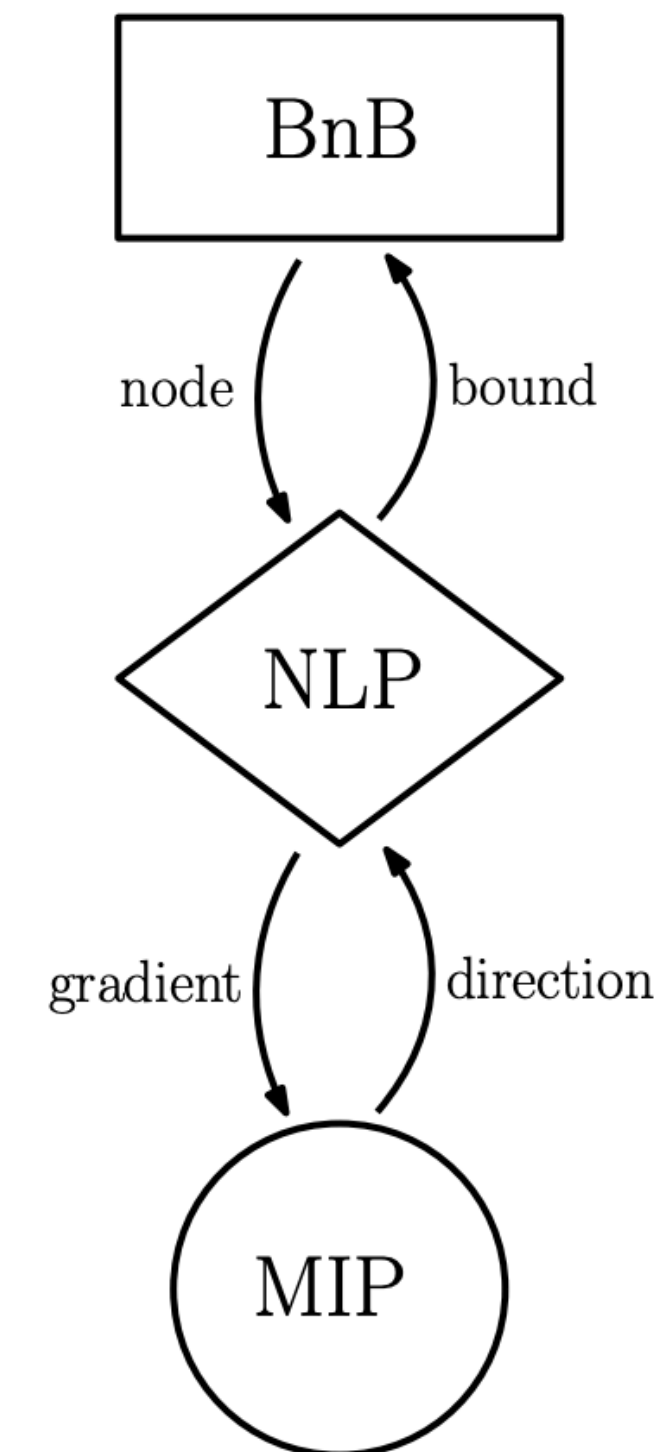
- Solves the MINLP using a B&B approach with an NLP being solved at every node



- Solves the MINLP using a B&B approach with an NLP being solved at every node
- The NLP is solved using the FW algorithm



- Solves the MINLP using a B&B approach with an NLP being solved at every node
- The NLP is solved using the FW algorithm
- The FW algorithm requires a MIP oracle





Integer LMO : IFW

Integer LMO : IFW

- Solves an integer optimization problem to find the descent direction.

$$\begin{aligned} \min_v \quad & \nabla_x c(x_t)^\top v^x + r^\top v^y \\ \text{s.t.} \quad & v^x \in \mathcal{X} \\ & v_j^x \leq M v_j^y \quad \forall j \in J \\ & v_j^y \in \{0, 1\} \end{aligned}$$

Integer LMO : IFW

- Solves an integer optimization problem to find the descent direction.
- Solves the math programming model without leveraging the special structure

$$\begin{aligned} \min_v \quad & \nabla_x c(x_t)^\top v^x + r^\top v^y \\ \text{s.t.} \quad & v^x \in \mathcal{X} \\ & v_j^x \leq M v_j^y \quad \forall j \in J \\ & v_j^y \in \{0, 1\} \end{aligned}$$

Network LMO with Penalty : NLMO-P

Network LMO with Penalty : NLMO-P

- x and y only connected by network design constraints.

Network LMO with Penalty : NLMO-P

- x and y only connected by network design constraints.
- Move the constraint into the objective and penalise its violation

$$\min_{x,y} c(x) + r^\top y + \lambda \sum_{j \in J} \max(x_j - My_j, 0)^p$$

$$\text{s.t. } x \in \mathcal{X}$$

$$y_j \in \{0, 1\}$$

Network LMO with Penalty : NLMO-P

- x and y only connected by network design constraints.
- Move the constraint into the objective and penalise its violation
- Allows the descent direction problem to be separated into network flow and a selection problems.

$$\min_{x,y} c(x) + r^\top y + \lambda \sum_{j \in J} \max(x_j - My_j, 0)^p$$

$$\text{s.t. } x \in \mathcal{X}$$

$$y_j \in \{0, 1\}$$

$$\min_x (\nabla c(x_t) + \lambda g_t)^\top v^x \quad \min_y (r + \lambda h_t)^\top v^y$$

$$\text{s.t. } v^x \in \mathcal{X}$$

$$\text{s.t. } v_j^y \in \{0, 1\}$$

$$g_{ti} = \begin{cases} 0 & i \notin J \\ 0 & i \in J \text{ \& } x_{ti} \leq My_{ti} \\ p(x_{ti} - My_{ti})^{p-1} & i \in J \text{ \& } x_{ti} > My_{ti} \end{cases}$$

$$h_{ki} = \begin{cases} 0 & i \in J \text{ \& } x_{ti} \leq My_{ti} \\ -Mp(x_{ti} - My_{ti})^{p-1} & i \in J \text{ \& } x_{ti} > My_{ti} \end{cases}$$

Network LMO with Penalty : NLMO-P

- x and y only connected by network design constraints.
- Move the constraint into the objective and penalise its violation
- Allows the descent direction problem to be separated into network flow and a selection problems.
- 2 parameters: λ (penalty) and p (power)

$$\min_{x,y} c(x) + r^\top y + \lambda \sum_{j \in J} \max(x_j - My_j, 0)^p$$

$$\text{s.t. } x \in \mathcal{X}$$

$$y_j \in \{0, 1\}$$

$$\min_x (\nabla c(x_t) + \lambda g_t)^\top v^x \quad \min_y (r + \lambda h_t)^\top v^y$$

$$\text{s.t. } v^x \in \mathcal{X}$$

$$\text{s.t. } v_j^y \in \{0, 1\}$$

$$g_{ti} = \begin{cases} 0 & i \notin J \\ 0 & i \in J \text{ \& } x_{ti} \leq My_{ti} \\ p(x_{ti} - My_{ti})^{p-1} & i \in J \text{ \& } x_{ti} > My_{ti} \end{cases}$$

$$h_{ki} = \begin{cases} 0 & i \in J \text{ \& } x_{ti} \leq My_{ti} \\ -Mp(x_{ti} - My_{ti})^{p-1} & i \in J \text{ \& } x_{ti} > My_{ti} \end{cases}$$



Network LMO with Benders : NLMO-B

Network LMO with Benders : NLMO-B

- We solve the descent problem using Benders's decomposition

$$\begin{aligned} \min_v \quad & \nabla_x c(x_t)^\top v^x + r^\top v^y \\ \text{s.t.} \quad & v^x \in \mathcal{X} \\ & v_j^x \leq M v_j^y \quad \forall j \in J \\ & v_j^y \in \{0, 1\} \end{aligned}$$

Network LMO with Benders : NLMO-B

- We solve the descent problem using Benders's decomposition
- The subproblem at each step then simplifies to network flow problem

$$\begin{aligned} \min_v & \nabla_x c(x_t)^\top v^x + r^\top v^y \\ \text{s.t. } & v^x \in \mathcal{X} \\ & v_j^x \leq M v_j^y \quad \forall j \in J \\ & v_j^y \in \{0, 1\} \end{aligned}$$

$$\begin{aligned} \min_{x,y} & r^\top v^y + \eta \\ \text{s.t. } & \eta \geq p_k^\top b - s_k^\top M v^y \quad \forall k \in OPT \\ & 0 \geq p_k^\top b - s_k^\top M v^y \quad \forall k \in FEAS \\ & v_j^y \in \{0, 1\} \quad \forall j \in J \end{aligned}$$

Network LMO with Benders : NLMO-B

- We solve the descent problem using Benders's decomposition
- The subproblem at each step then simplifies to network flow problem
- Disadvantage: Requires a iterative algorithm for the descent problem which can be slow

$$\begin{aligned} \min_v \quad & \nabla_x c(x_t)^\top v^x + r^\top v^y \\ \text{s.t.} \quad & v^x \in \mathcal{X} \\ & v_j^x \leq M v_j^y \quad \forall j \in J \\ & v_j^y \in \{0, 1\} \end{aligned}$$

$$\begin{aligned} \min_{x,y} \quad & r^\top v^y + \eta \\ \text{s.t.} \quad & \eta \geq p_k^\top b - s_k^\top M v^y \quad \forall k \in OPT \\ & 0 \geq p_k^\top b - s_k^\top M v^y \quad \forall k \in FEAS \\ & v_j^y \in \{0, 1\} \quad \forall j \in J \end{aligned}$$



Experimental Setup

Experimental Setup

- Transportation Networks repository

Experimental Setup

- Transportation Networks repository
- Instances created out of existing datasets by removing arcs at random

Experimental Setup

- Transportation Networks repository
- Instances created out of existing datasets by removing arcs at random
- Model solved with a time limit of 3 hours with no post processing

Experimental Setup

- Transportation Networks repository
- Instances created out of existing datasets by removing arcs at random
- Model solved with a time limit of 3 hours with no post processing

Methods Compared

Experimental Setup

- Transportation Networks repository
- Instances created out of existing datasets by removing arcs at random
- Model solved with a time limit of 3 hours with no post processing

Methods Compared

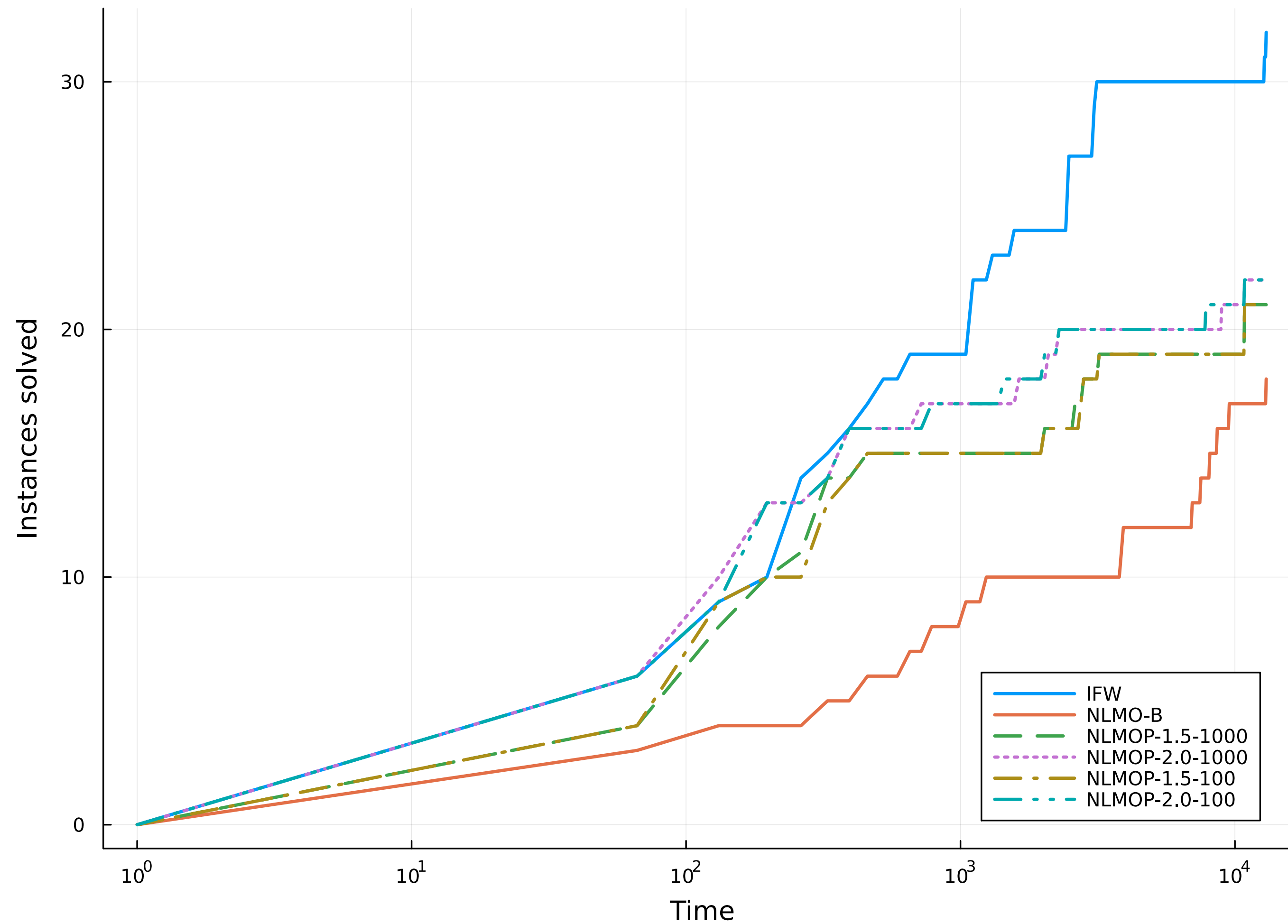
- IFW
- NLMO-P
- NLMO-B



Results

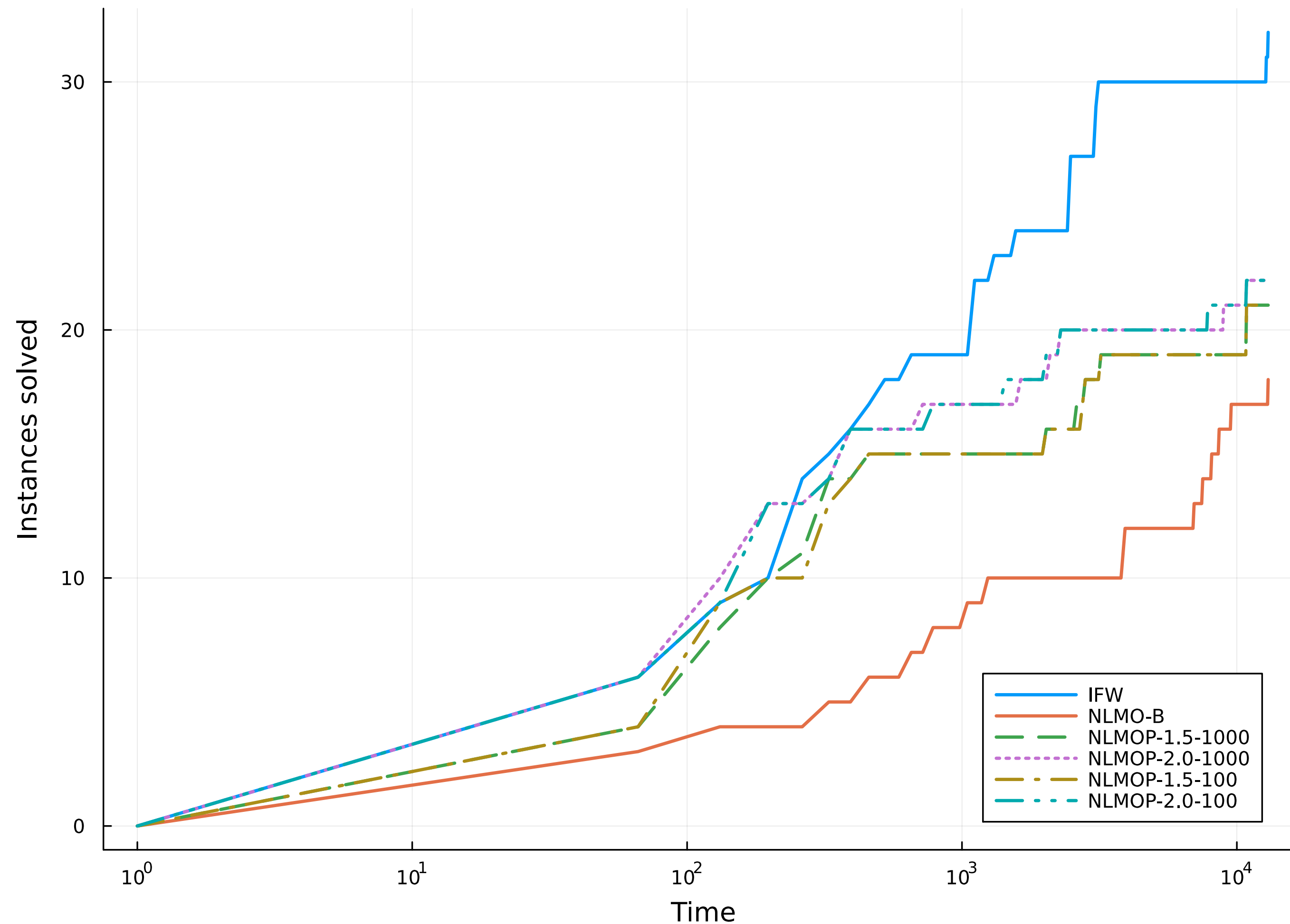
Results

Instances solved vs Time



Results

Instances solved vs Time



The IFW algorithm solves more instances than the penalty based or Bender's approaches

Results : Common Instances

Results : Common Instances

Algorithm	Time	Number Of Nodes	Time Per Node	Rel Dual Gap	Total Const. Viol. (1e-3)
NLMO-P-2-1000	1746.59	448.50	3.27	0.0038	1.2089
NLMO-P-1.5-1000	2133.50	549.88	3.65	0.0354	0.0024
NLMO-P-2-100	1672.28	445.50	3.26	0.0040	1.2089
NLMO-P-1.5-100	2143.29	576.63	3.63	0.0351	0.0024
NLMO-B	3307.55	3.00	1086.92	0.0031	
IFW	423.28	10.50	40.29	0.0017	



Conclusion

Conclusion

- We leverage the Boscia.jl library to solve the network design problem for the traffic assignment model

Conclusion

- We leverage the Boscia.jl library to solve the network design problem for the traffic assignment model
- The IFW LMO which solves an integer MIP leads to the fastest algorithm

Conclusion

- We leverage the Boscia.jl library to solve the network design problem for the traffic assignment model
- The IFW LMO which solves an integer MIP leads to the fastest algorithm
- The Penalty-based LMO is much faster but explores more of the B&B tree before converging

Conclusion

- We leverage the Boscia.jl library to solve the network design problem for the traffic assignment model
- The IFW LMO which solves an integer MIP leads to the fastest algorithm
- The Penalty-based LMO is much faster but explores more of the B&B tree before converging
- More experiments needed to identify situations in which the benefits of the penalty based or Bender's based approach are realised

thank you!