### Nonuniform Models for Robust Network Design

David Adjiashvili

IFOR, ETH Zürich

#### Joint work with Sebastian Stiller and Rico Zenklusen

July 31, 2013

◆□▶ ◆□▶ ◆注▶ ◆注▶ 注 のへで

- Combinatorial Problems (Shortest Path, Spanning Tree...)
- Adversarial Failure Model ("Robust" Optimization)
- Covering Problems

• Nonuniform failure scenarios ? , ?

Complexity/Algorithms (exact, approximation)

< ロ > < 同 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ >

#### • Combinatorial Problems (Shortest Path, Spanning Tree...)

• Adversarial Failure Model ("Robust" Optimization)

Covering Problems

• Nonuniform failure scenarios ? , ?

Complexity/Algorithms (exact, approximation)

イロト イポト イヨト イヨト

#### • Combinatorial Problems (Shortest Path, Spanning Tree...)

- Adversarial Failure Model ("Robust" Optimization)
- Covering Problems

• Nonuniform failure scenarios ? , ?

Complexity/Algorithms (exact, approximation)

イロト イポト イヨト イヨト

- Combinatorial Problems (Shortest Path, Spanning Tree...)
- Adversarial Failure Model ("Robust" Optimization)
- Covering Problems

• Nonuniform failure scenarios 🦳 ? , ?

Complexity/Algorithms (exact, approximation)

- Combinatorial Problems (Shortest Path, Spanning Tree...)
- Adversarial Failure Model ("Robust" Optimization)
- Covering Problems

• Nonuniform failure scenarios ? , ?

Complexity/Algorithms (exact, approximation)

・ロッ ・雪 ・ ・ ヨ ・ ・ ヨ ・

- Combinatorial Problems (Shortest Path, Spanning Tree...)
- Adversarial Failure Model ("Robust" Optimization)
- Covering Problems

• Nonuniform failure scenarios ? , ?

Complexity/Algorithms (exact, approximation)

- Combinatorial Problems (Shortest Path, Spanning Tree...)
- Adversarial Failure Model ("Robust" Optimization)
- Covering Problems

Nonuniform failure scenarios

Complexity/Algorithms (exact, approximation)

- Combinatorial Problems (Shortest Path, Spanning Tree...)
- Adversarial Failure Model ("Robust" Optimization)
- Covering Problems

• Nonuniform failure scenarios ?

Complexity/Algorithms (exact, approximation)

- Combinatorial Problems (Shortest Path, Spanning Tree...)
- Adversarial Failure Model ("Robust" Optimization)
- Covering Problems

• Nonuniform failure scenarios ? , ?

Complexity/Algorithms (exact, approximation)

イロト イポト イヨト イヨト

- Combinatorial Problems (Shortest Path, Spanning Tree...)
- Adversarial Failure Model ("Robust" Optimization)
- Covering Problems
  - Nonuniform failure scenarios ? , ?
- Complexity/Algorithms (exact, approximation)

イロン 不同 とくほう イロン

- Combinatorial Problems (Shortest Path, Spanning Tree...)
- Adversarial Failure Model ("Robust" Optimization)
- Covering Problems
  - Nonuniform failure scenarios ? , ?
- Complexity/Algorithms (exact, approximation)

・ロト ・ 同ト ・ ヨト ・ ヨト

### What is the cause for failures? depends...

# What does the network represent?

David Adjiashvili Nonuniform Models for Robust Network Design

(人間) (人) (人) (人) (人) (人)

### What is the cause for failures? depends...

# What does the network represent?

David Adjiashvili Nonuniform Models for Robust Network Design

(4 同) (4 日) (4 日)

### What is the cause for failures? depends...

# What does the network represent?

David Adjiashvili Nonuniform Models for Robust Network Design

・ 同 ト ・ ヨ ト ・ ヨ ト

 $\parallel$ 

What is the cause for failures? dependence

What does the network represent?

・ 同 ト ・ ヨ ト ・ ヨ ト

What is the cause for failures? depends...

 $\parallel$ 

What does the network represent?

伺 ト イヨト イヨト

What is the cause for failures? depends...

 $\parallel$ 

What does the network represent?

 $\downarrow$ 

/⊒ > < ∃ >

What is the cause for failures? depends...

 $\parallel$ 

What does the network represent?

 $\downarrow$ 

/⊒ > < ∃ >

Distributed Computer Systems

▲御▶ ▲理▶ ▲理▶

э

**Distributed Computer Systems** 

- Many processes running on various physical machines, using various resources
- Faults: downtime of resources.

・ 同 ト ・ ヨ ト ・ ヨ ト

#### **Distributed Computer Systems**

- Many processes running on various physical machines, using various resources (databases, powerful machines etc.)
- Faults: downtime of resources.

#### **Distributed Computer Systems**

- Many processes running on various physical machines, using various resources (databases, powerful machines etc.)
- Faults: downtime of resources.

・ 同 ト ・ ヨ ト ・ ヨ ト

#### **Distributed Computer Systems**

- Many processes running on various physical machines, using various resources (databases, powerful machines etc.)
- Faults: downtime of resources.



. . . . . . .

#### **Distributed Computer Systems**

- Many processes running on various physical machines, using various resources (databases, powerful machines etc.)
- Faults: downtime of resources.



A B A A B A

#### **Distributed Computer Systems**

- Many processes running on various physical machines, using various resources (databases, powerful machines etc.)
- Faults: downtime of resources.



A B A A B A

#### **Distributed Computer Systems**

- Many processes running on various physical machines, using various resources (databases, powerful machines etc.)
- Faults: downtime of resources.



A B A A B A

#### Distributed Computer Systems

- Many processes running on various physical machines, using various resources (databases, powerful machines etc.)
- Faults: downtime of resources.



• • = • • = •

#### Distributed Computer Systems

- Many processes running on various physical machines, using various resources (databases, powerful machines etc.)
- Faults: downtime of resources.

Similar characteristics in networks representing

- Health care facilities
- Digitally controlled infrastructures
- Hierarchical organizations
- ...

イロン 不同 とくほう イロン

Financial Investment Networks

▲□ ▶ ▲ □ ▶ ▲ □ ▶

#### Financial Investment Networks

- Many companies in a market with mutual dependencies
- Faults: bankruptcy of companies

イロト イポト イヨト イヨト

Financial Investment Networks

- Many companies in a market with mutual dependencies (investment, supply etc.)
- Faults: bankruptcy of companies

Financial Investment Networks

- Many companies in a market with mutual dependencies (investment, supply etc.)
- Faults: bankruptcy of companies

・ 戸 ・ ・ ヨ ・ ・ ヨ ・

Financial Investment Networks

- Many companies in a market with mutual dependencies (investment, supply etc.)
- Faults: bankruptcy of companies  $\Rightarrow$  Causes cascades!

・ 同 ト ・ ヨ ト ・ ヨ ト

Financial Investment Networks

- Many companies in a market with mutual dependencies (investment, supply etc.)
- Faults: bankruptcy of companies  $\Rightarrow$  Causes cascades!



直 と く ヨ と く ヨ と

Financial Investment Networks

- Many companies in a market with mutual dependencies (investment, supply etc.)
- Faults: bankruptcy of companies  $\Rightarrow$  Causes cascades!



直 と く ヨ と く ヨ と
Financial Investment Networks

- Many companies in a market with mutual dependencies (investment, supply etc.)
- Faults: bankruptcy of companies  $\Rightarrow$  Causes cascades!



直 と く ヨ と く ヨ と

Financial Investment Networks

- Many companies in a market with mutual dependencies (investment, supply etc.)
- Faults: bankruptcy of companies  $\Rightarrow$  Causes cascades!



・ 同 ト ・ ヨ ト ・ ヨ ト

Financial Investment Networks

- Many companies in a market with mutual dependencies (investment, supply etc.)
- Faults: bankruptcy of companies  $\Rightarrow$  Causes cascades!



同 ト イヨ ト イヨ ト

Financial Investment Networks

- Many companies in a market with mutual dependencies (investment, supply etc.)
- Faults: bankruptcy of companies  $\Rightarrow$  Causes cascades!



・ 同 ト ・ ヨ ト ・ ヨ ト

#### Financial Investment Networks

- Many companies in a market with mutual dependencies (investment, supply etc.)
- Faults: bankruptcy of companies  $\Rightarrow$  Causes cascades!
- Similar characteristics in networks representing
  - Electricity networks
  - Insurance networks
  - Social networks
  - ...

イロン 不同 とくほう イロン

- Simultaneous failure of variable-size parts of the network
- Failure of a single resource that causes failure of multiple network components
- Propagation effects

Certainly,

# $\Omega \neq \{A \subset N : w(A) \leq k\} =$ "Uniform"

(1)

- Simultaneous failure of variable-size parts of the network
- Failure of a single resource that causes failure of multiple network components
- Propagation effects

Certainly,

# $\Omega \neq \{A \subset N : w(A) \leq k\} =$ "Uniform"

(1)

- Simultaneous failure of variable-size parts of the network
- Failure of a single resource that causes failure of multiple network components
- Propagation effects

Certainly,

## $\Omega \neq \{A \subset N : w(A) \leq k\} =$ "Uniform"

- Simultaneous failure of variable-size parts of the network
- Failure of a single resource that causes failure of multiple network components
- Propagation effects

Certainly,

# $\Omega \neq \{A \subset N : w(A) \leq k\} =$ "Uniform"

イロト 不得 とくほ とくほう

- Simultaneous failure of variable-size parts of the network
- Failure of a single resource that causes failure of multiple network components
- Propagation effects

Certainly,

# $\Omega \neq \{A \subset N : w(A) \leq k\} =$ "Uniform"

ヘロト ヘヨト ヘヨト

- Simultaneous failure of variable-size parts of the network
- Failure of a single resource that causes failure of multiple network components
- Propagation effects

Certainly,

# $\Omega \neq \{A \subset N : w(A) \leq k\} =$ "Uniform"

- Simultaneous failure of variable-size parts of the network
- Failure of a single resource that causes failure of multiple network components
- Propagation effects

Certainly,

## $\Omega \neq \{A \subset N : w(A) \leq k\} =$ "Uniform"

イロト 不得 トイヨト イヨト

- Simultaneous failure of variable-size parts of the network
- Failure of a single resource that causes failure of multiple network components
- Propagation effects

Certainly,

## $\Omega \neq \{A \subset N : w(A) \leq k\} =$ "Uniform"

イロト 不得 トイヨト イヨト

Combinatorial (covering) problem P:  $\min\{c(X) : X \in S\}$   $(S \subset 2^N$  feasible set,  $c : N \to \mathbb{Z}_+)$ 

Scenario set:

 $\Omega = \{F_1, \cdots, F_m\} \quad (F_1, \cdots, F_m \subset N \text{ scenarios})$ 

Bulk-Robust counterpart Bulk(P): $\min\{c(X):X\setminus F_i\in\mathcal{S}\mid orall F_i\in\Omega\}$ 

イロト 不得 とくほとう ほうとう

### Nonuniform Models - Bulk Robustness

Combinatorial (covering) problem P:

 $\min\{c(X): X \in S\}$   $(S \subset 2^N \text{ feasible set, } c: N \to \mathbb{Z}_+)$ 

Scenario set:

 $\Omega = \{F_1, \cdots, F_m\} \quad (F_1, \cdots, F_m \subset N \text{ scenarios})$ 

Bulk-Robust counterpart Bulk(P): $\min\{c(X):X\setminus F_i\in\mathcal{S} \mid orall F_i\in \mathcal{G}\}$ 

Combinatorial (covering) problem P:

 $\min\{c(X): X \in S\}$   $(S \subset 2^N \text{ feasible set, } c: N \to \mathbb{Z}_+)$ 

Scenario set:

 $\Omega = \{F_1, \cdots, F_m\} \quad (F_1, \cdots, F_m \subset N \text{ scenarios})$ 

Bulk-Robust counterpart Bulk(P): $\min\{c(X):X\setminus F_i\in \mathcal{S} \mid \forall F_i\in \Omega\}$ 

Combinatorial (covering) problem P:

 $\min\{c(X): X \in S\}$   $(S \subset 2^N \text{ feasible set, } c: N \to \mathbb{Z}_+)$ 

Scenario set:

 $\Omega = \{F_1, \cdots, F_m\} \quad (F_1, \cdots, F_m \subset N \text{ scenarios})$ 

Bulk-Robust counterpart Bulk(P):

 $\min\{c(X): X \setminus F_i \in \mathcal{S} \quad \forall F_i \in \Omega\}$ 

◆□ ▶ ◆□ ▶ ◆ 三 ▶ ◆ 三 ● ● ● ●

### $\min\{c(X): (V, X \setminus F_i) \text{ is connected } \forall F_i \in \Omega\}$

### $\min\{c(X): (V, X \setminus F_i) \text{ is connected } \forall F_i \in \Omega\}$



- ( 同 ) ( 回 ) ( 回 ) - 回

 $\min\{c(X): (V, X \setminus F_i) \text{ is connected } \forall F_i \in \Omega\}$ 





・日・ ・日・

- ∢ ⊒ →

 $\min\{c(X): (V, X \setminus F_i) \text{ is connected } \forall F_i \in \Omega\}$ 





▲ 同 ▶ → 目 ▶

 $\bullet \; \Rightarrow$  e.g. edge-sets failures, vertex-sets failures...

- Bulk(P) instance feasible iff  $N \setminus F \in S$  for all  $F \in \Omega$ .
- $\beta$ -approximation for P implies a  $|\Omega|\beta$ -approximation for Bulk(P):
  - Compute  $\beta$ -approximate solution  $X_F$  for relaxation

 $\min\{c(X): X \setminus F \in \mathcal{S}\}$ 

• Return  $X = \bigcup_{F \in \Omega} X_F$ 

イロト イポト イヨト イヨト

 $\bullet \Rightarrow$  e.g. edge-sets failures, vertex-sets failures...

- Bulk(P) instance feasible iff  $N \setminus F \in S$  for all  $F \in \Omega$ .
- $\beta$ -approximation for P implies a  $|\Omega|\beta$ -approximation for Bulk(P):
  - Compute  $\beta$ -approximate solution  $X_F$  for relaxation

 $\min\{c(X):X\setminus F\in \mathcal{S}\}$ 

• Return  $X = \bigcup_{F \in \Omega} X_F$ 

・ロト ・ 一下・ ・ 日 ・ ・ 日 ・

 $\bullet \ \Rightarrow$  e.g. edge-sets failures, vertex-sets failures...

- Bulk(P) instance feasible iff  $N \setminus F \in S$  for all  $F \in \Omega$ .
- $\beta$ -approximation for P implies a  $|\Omega|\beta$ -approximation for Bulk(P):
  - Compute  $\beta$ -approximate solution  $X_F$  for relaxation

 $\min\{c(X):X\setminus F\in \mathcal{S}\}$ 

• Return  $X = \bigcup_{F \in \Omega} X_F$ 

・ロト ・ 日 ・ ・ 日 ・ ・ 日 ・

 $\bullet \ \Rightarrow$  e.g. edge-sets failures, vertex-sets failures...

- Bulk(P) instance feasible iff  $N \setminus F \in S$  for all  $F \in \Omega$ .
- $\beta$ -approximation for P implies a  $|\Omega|\beta$ -approximation for Bulk(P):

• Compute  $\beta$ -approximate solution  $X_F$  for relaxation

 $\min\{c(X):X\setminus F\in \mathcal{S}\}$ 

• Return  $X = \bigcup_{F \in \Omega} X_F$ 

・ロッ ・雪 ・ ・ ヨ ・ ・ ヨ ・

 $\bullet \ \Rightarrow$  e.g. edge-sets failures, vertex-sets failures...

- Bulk(P) instance feasible iff  $N \setminus F \in S$  for all  $F \in \Omega$ .
- $\beta$ -approximation for P implies a  $|\Omega|\beta$ -approximation for Bulk(P):

• Compute  $\beta$ -approximate solution  $X_F$  for relaxation

 $\min\{c(X):X\setminus F\in\mathcal{S}\}$ 

• Return  $X = \bigcup_{F \in \Omega} X_F$ 

 $\bullet \ \Rightarrow$  e.g. edge-sets failures, vertex-sets failures...

- Bulk(P) instance feasible iff  $N \setminus F \in S$  for all  $F \in \Omega$ .
- $\beta$ -approximation for P implies a  $|\Omega|\beta$ -approximation for Bulk(P):
  - Compute  $\beta$ -approximate solution  $X_F$  for relaxation

 $\min\{c(X): X \setminus F \in \mathcal{S}\}$ 

• Return  $X = \bigcup_{F \in \Omega} X_F$ 

Let  $N = \{1, \cdots, n\}$  and  $\mathbf{P}^U$ :  $\min\{c(X) : X \subset N, |X| \ge 1\}$  $\Rightarrow \operatorname{Bulk}(\mathbf{P}^U) \equiv \operatorname{Set}$  Cover

⇒ Bulk(Spanning Tree), Bulk(Shortest Path), etc. unlikely to admit polynomial constant-factor approximation algorithms

(not better than  $\ln |\Omega|$ )

At the same time, e.g., Bulk(Set Cover) pprox Set Cover

Bulk(Shortest Path)? Bulk(Spanning Tree)?

イロト イポト イヨト イヨト

# Let $N = \{1, \dots, n\}$ and $P^U$ : $\min\{c(X) : X \subset N, |X| \ge 1\}$ $\Rightarrow$ Bulk $(P^U) =$ Set Cover

⇒ Bulk(Spanning Tree), Bulk(Shortest Path), etc. unlikely to admit polynomial constant-factor approximation algorithms

(not better than  $\ln |\Omega|$ )

At the same time, e.g., Bulk(Set Cover) pprox Set Cover

Bulk(Shortest Path)? Bulk(Spanning Tree)?

イロト イポト イヨト イヨト

### Bulk(P) and Set Cover

## Let $N = \{1, \dots, n\}$ and $P^U$ : $\min\{c(X) : X \subset N, |X| \ge 1\}$ $\Rightarrow \text{Bulk}(P^U) \equiv \text{Set Cover}$

⇒ Bulk(Spanning Tree), Bulk(Shortest Path), etc. unlikely to admit polynomial constant-factor approximation algorithms

(not better than  $\ln |\Omega|$ )

At the same time, e.g., Bulk(Set Cover) pprox Set Cover

Bulk(Shortest Path)? Bulk(Spanning Tree)?

・ロト ・ 一下・ ・ 日 ・ ・ 日 ・

Let  $N = \{1, \dots, n\}$  and  $P^U$ :  $\min\{c(X) : X \subset N, |X| \ge 1\}$  $\Rightarrow \operatorname{Bulk}(P^U) \equiv \operatorname{Set} \operatorname{Cover}$ 

 $\Rightarrow$  Bulk(Spanning Tree), Bulk(Shortest Path), etc. unlikely to admit polynomial constant-factor approximation algorithms

(not better than  $\ln |\Omega|$ )

At the same time, e.g., Bulk(Set Cover) pprox Set Cover

Bulk(Shortest Path)? Bulk(Spanning Tree)?

Let  $N = \{1, \dots, n\}$  and  $P^U$ :  $\min\{c(X) : X \subset N, |X| \ge 1\}$  $\Rightarrow \operatorname{Bulk}(P^U) \equiv \operatorname{Set} \operatorname{Cover}$ 

 $\Rightarrow$  Bulk(Spanning Tree), Bulk(Shortest Path), etc. unlikely to admit polynomial constant-factor approximation algorithms

(not better than  $\ln |\Omega|$ )

At the same time, e.g., Bulk(Set Cover) pprox Set Cover

Bulk(Shortest Path)? Bulk(Spanning Tree)?

イロト イポト イヨト イヨト

Let  $N = \{1, \dots, n\}$  and  $P^U$ :  $\min\{c(X) : X \subset N, |X| \ge 1\}$  $\Rightarrow \operatorname{Bulk}(P^U) \equiv \operatorname{Set} \operatorname{Cover}$ 

 $\Rightarrow$  Bulk(Spanning Tree), Bulk(Shortest Path), etc. unlikely to admit polynomial constant-factor approximation algorithms

(not better than  $\ln |\Omega|$ )

At the same time, e.g., Bulk(Set Cover)  $\approx$  Set Cover

### Bulk(Shortest Path)? Bulk(Spanning Tree)?

イロト 不得 トイヨト イヨト 二日

Let  $N = \{1, \dots, n\}$  and  $P^U$ :  $\min\{c(X) : X \subset N, |X| \ge 1\}$  $\Rightarrow \text{Bulk}(P^U) \equiv \text{Set Cover}$ 

 $\Rightarrow$  Bulk(Spanning Tree), Bulk(Shortest Path), etc. unlikely to admit polynomial constant-factor approximation algorithms

(not better than  $\ln |\Omega|$ )

At the same time, e.g., Bulk(Set Cover)  $\approx$  Set Cover

Bulk(Shortest Path)? Bulk(Spanning Tree)?

Let  $N = \{1, \dots, n\}$  and  $P^U$ :  $\min\{c(X) : X \subset N, |X| \ge 1\}$  $\Rightarrow \text{Bulk}(P^U) \equiv \text{Set Cover}$ 

 $\Rightarrow$  Bulk(Spanning Tree), Bulk(Shortest Path), etc. unlikely to admit polynomial constant-factor approximation algorithms

(not better than  $\ln |\Omega|$ )

At the same time, e.g., Bulk(Set Cover)  $\approx$  Set Cover

## Bulk(Shortest Path)? Bulk(Spanning Tree)?

イロト 不得 トイヨト イヨト 二日

**Theorem.** There is a polynomial  $(\log |\Omega| + \log r)$ -approximation algorithm for Bulk(Minimum Matroid Basis).

 $\Rightarrow (\log |\Omega| + \log |V|) \text{-approximation for Bulk(Spanning Tree)}.$ 

Proof sketch:

Let  $r(\cdot)$  denote the rank function of the matroid  $\mathcal{M}$ . For  $F \in \Omega$  define  $r^F(X) = r(X \setminus F)$ .  $\Rightarrow$  Submodular

Define:

$$f(X) = \sum_{F \in \Omega} r^F(X) \qquad \Rightarrow$$
Submodular

#### Note: $S \subset N$ feasible iff $f(S) = f(N) = r(\mathcal{M})|\Omega|$

イロト イポト イヨト イヨト
$\Rightarrow (\log |\Omega| + \log |V|) \text{-approximation for Bulk(Spanning Tree)}.$ Proof sketch: Let  $r(\cdot)$  denote the rank function of the matroid  $\mathcal{M}$ . For  $F \in \Omega$  define  $r^F(X) = r(X \setminus F)$ .  $\Rightarrow$  Submodular Define:  $f(X) = \sum r^F(X) \Rightarrow$  Submodular

#### Note: $S \subset N$ feasible iff $f(S) = f(N) = r(\mathcal{M})|\Omega|$

< ロ > < 同 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ >

 $\Rightarrow (\log |\Omega| + \log |V|)$ -approximation for Bulk(Spanning Tree).

Proof sketch: Let  $r(\cdot)$  denote the rank function of the matroid  $\mathcal{M}$ . For  $F \in \Omega$  define  $r^F(X) = r(X \setminus F)$ .  $\Rightarrow$  Submodular Define:

$$f(X) = \sum_{F \in \Omega} r^F(X) \qquad \Rightarrow \mathsf{Submodular}$$

#### Note: $S \subset N$ feasible iff $f(S) = f(N) = r(\mathcal{M})|\Omega|$

(a)

 $\Rightarrow (\log |\Omega| + \log |V|)$ -approximation for Bulk(Spanning Tree).

#### Proof sketch:

Let  $r(\cdot)$  denote the rank function of the matroid  $\mathcal{M}$ . For  $F \in \Omega$  define  $r^F(X) = r(X \setminus F)$ .  $\Rightarrow$  Submodular

Define:

$$f(X) = \sum_{F \in \Omega} r^F(X) \qquad \Rightarrow \mathsf{Submodular}$$

#### Note: $S \subset N$ feasible iff $f(S) = f(N) = r(\mathcal{M})|\Omega|$

(a)

 $\Rightarrow (\log |\Omega| + \log |V|)$ -approximation for Bulk(Spanning Tree).

Proof sketch:

Let  $r(\cdot)$  denote the rank function of the matroid  $\mathcal{M}$ . For  $F \in \Omega$  define  $r^F(X) = r(X \setminus F)$ .  $\Rightarrow$  Submodular Define:

$$f(X) = \sum_{F \in \Omega} r^F(X) \qquad \Rightarrow \text{Submodular}$$

#### Note: $S \subset N$ feasible iff $f(S) = f(N) = r(\mathcal{M})|\Omega|$

 $\Rightarrow (\log |\Omega| + \log |V|)$ -approximation for Bulk(Spanning Tree).

Proof sketch:

Let  $r(\cdot)$  denote the rank function of the matroid  $\mathcal{M}$ . For  $F \in \Omega$  define  $r^F(X) = r(X \setminus F)$ .  $\Rightarrow$  Submodular

Define:

 $f(X) = \sum_{F \in \Omega} r^F(X) \quad \Rightarrow$ Submodular

#### Note: $S \subset N$ feasible iff $f(S) = f(N) = r(\mathcal{M})|\Omega|$

< ロ > < 同 > < 回 > < 回 > < □ > <

 $\Rightarrow (\log |\Omega| + \log |V|)$ -approximation for Bulk(Spanning Tree).

Proof sketch:

Let  $r(\cdot)$  denote the rank function of the matroid  $\mathcal{M}$ . For  $F \in \Omega$  define  $r^F(X) = r(X \setminus F)$ .  $\Rightarrow$  Submodular

Define:

 $f(X) = \sum_{F \in \Omega} r^F(X) \quad \Rightarrow$ Submodular

#### Note: $S \subset N$ feasible iff $f(S) = f(N) = r(\mathcal{M})|\Omega|$

< ロ > < 同 > < 回 > < 回 > < □ > <

 $\Rightarrow (\log |\Omega| + \log |V|)$ -approximation for Bulk(Spanning Tree).

Proof sketch:

Let  $r(\cdot)$  denote the rank function of the matroid  $\mathcal{M}$ . For  $F \in \Omega$  define  $r^F(X) = r(X \setminus F)$ .  $\Rightarrow$  Submodular

Define:

$$f(X) = \sum_{F \in \Omega} r^F(X) \implies$$
Submodular

#### Note: $S \subset N$ feasible iff $f(S) = f(N) = r(\mathcal{M})|\Omega|$

イロン 不同 とくほう イロン

 $\Rightarrow (\log |\Omega| + \log |V|)$ -approximation for Bulk(Spanning Tree).

Proof sketch:

Let  $r(\cdot)$  denote the rank function of the matroid  $\mathcal{M}$ . For  $F \in \Omega$  define  $r^F(X) = r(X \setminus F)$ .  $\Rightarrow$  Submodular

Define:

$$f(X) = \sum_{F \in \Omega} r^F(X) \qquad \Rightarrow \text{Submodular}$$

Note:  $S \subset N$  feasible iff  $f(S) = f(N) = r(\mathcal{M})|\Omega|$ 

イロン 不同 とくほう イロン

 $\Rightarrow (\log |\Omega| + \log |V|)$ -approximation for Bulk(Spanning Tree).

Proof sketch:

Let  $r(\cdot)$  denote the rank function of the matroid  $\mathcal{M}$ . For  $F \in \Omega$  define  $r^F(X) = r(X \setminus F)$ .  $\Rightarrow$  Submodular

Define:

$$f(X) = \sum_{F \in \Omega} r^F(X) \quad \Rightarrow \text{Submodular}$$

Note:  $S \subset N$  feasible iff  $f(S) = f(N) = r(\mathcal{M})|\Omega|$ 

Proof sketch (cont.):  $f(X) = \sum_{F \in \Omega} r^F(X) = \sum_{F \in \Omega} r(X \setminus F)$ 

Try to maximize *f* without paying more than *OPT*.

 $\max\{f(X): c(X) \le OPT\}$ 

 $\Rightarrow$  Submodular Function maximization with knapsack constraint

NP-hard, but constant-factor approximations exist e.g.  $1 - \frac{1}{e} - \epsilon$  [Kulik, Shachnai, Tamir 10']

 $\Rightarrow$  Obtain a set  $Y_1$  with

•  $f(Y_1) \ge (1 - \alpha)r(\mathcal{M})|\Omega|$  for some  $\alpha \in (0, 1)$ •  $c(Y_1) \le OPT$ 

Iterate...

Proof sketch (cont.):  $f(X) = \sum_{F \in \Omega} r^F(X) = \sum_{F \in \Omega} r(X \setminus F)$ 

Try to maximize f without paying more than OPT.

 $\max\{f(X): c(X) \le OPT\}$ 

 $\Rightarrow$  Submodular Function maximization with knapsack constraint

NP-hard, but constant-factor approximations exist e.g.  $1 - \frac{1}{e} - \epsilon$  [Kulik, Shachnai, Tamir 10']

 $\Rightarrow$  Obtain a set  $Y_1$  with

•  $f(Y_1) \ge (1 - \alpha)r(\mathcal{M})|\Omega|$  for some  $\alpha \in (0, 1)$ •  $c(Y_1) \le OPT$ 

Iterate...

イロン 不同 とくほう イロン

Proof sketch (cont.):  $f(X) = \sum_{F \in \Omega} r^F(X) = \sum_{F \in \Omega} r(X \setminus F)$ 

Try to maximize f without paying more than OPT.

 $\max\{f(X): c(X) \le OPT\}$ 

 $\Rightarrow$  Submodular Function maximization with knapsack constraint

NP-hard, but constant-factor approximations exist e.g.  $1 - \frac{1}{e} - \epsilon$  [Kulik, Shachnai, Tamir 10']

 $\Rightarrow$  Obtain a set  $Y_1$  with

•  $f(Y_1) \ge (1 - \alpha)r(\mathcal{M})|\Omega|$  for some  $\alpha \in (0, 1)$ •  $c(Y_1) \le OPT$ 

Iterate...

Proof sketch (cont.):  $f(X) = \sum_{F \in \Omega} r^F(X) = \sum_{F \in \Omega} r(X \setminus F)$ 

Try to maximize f without paying more than OPT.

 $\max\{f(X): c(X) \le OPT\}$ 

#### $\Rightarrow$ Submodular Function maximization with knapsack constraint

NP-hard, but constant-factor approximations exist e.g.  $1 - rac{1}{e} - \epsilon$  [Kulik, Shachnai, Tamir 10']

 $\Rightarrow$  Obtain a set  $Y_1$  with

•  $f(Y_1) \ge (1 - \alpha)r(\mathcal{M})|\Omega|$  for some  $\alpha \in (0, 1)$ •  $c(Y_1) \le OPT$ 

Iterate...

Proof sketch (cont.):  $f(X) = \sum_{F \in \Omega} r^F(X) = \sum_{F \in \Omega} r(X \setminus F)$ 

Try to maximize f without paying more than OPT.

 $\max\{f(X): c(X) \le OPT\}$ 

#### $\Rightarrow$ Submodular Function maximization with knapsack constraint

**NP-hard,** but constant-factor approximations exist e.g.  $1 - \frac{1}{\epsilon} - \epsilon$  [Kulik, Shachnai, Tamir 10']

 $\Rightarrow$  Obtain a set  $Y_1$  with

•  $f(Y_1) \ge (1 - \alpha)r(\mathcal{M})|\Omega|$  for some  $\alpha \in (0, 1)$ •  $c(Y_1) \le OPT$ 

Iterate...

Proof sketch (cont.):  $f(X) = \sum_{F \in \Omega} r^F(X) = \sum_{F \in \Omega} r(X \setminus F)$ 

Try to maximize f without paying more than OPT.

 $\max\{f(X): c(X) \le OPT\}$ 

 $\Rightarrow$  Submodular Function maximization with knapsack constraint

NP-hard, but constant-factor approximations exist e.g.  $1 - \frac{1}{e} - \epsilon$  [Kulik, Shachnai, Tamir 10']

 $\begin{array}{l} \Rightarrow \ \mathsf{Obtain} \ \mathsf{a} \ \mathsf{set} \ Y_1 \ \mathsf{with} \\ \bullet \ f(Y_1) \geq (1-\alpha) r(\mathcal{M}) |\Omega| \ \mathsf{for} \ \mathsf{some} \ \alpha \in (0,1) \\ \bullet \ c(Y_1) \leq OPT \end{array}$ 

Iterate...

Proof sketch (cont.):  $f(X) = \sum_{F \in \Omega} r^F(X) = \sum_{F \in \Omega} r(X \setminus F)$ 

Try to maximize f without paying more than OPT.

 $\max\{f(X): c(X) \le OPT\}$ 

 $\Rightarrow$  Submodular Function maximization with knapsack constraint

NP-hard, but constant-factor approximations exist e.g.  $1 - \frac{1}{e} - \epsilon$  [Kulik, Shachnai, Tamir 10']

 $\Rightarrow$  Obtain a set  $Y_1$  with

•  $f(Y_1) \ge (1 - \alpha)r(\mathcal{M})|\Omega|$  for some  $\alpha \in (0, 1)$ •  $c(Y_1) \le OPT$ 

Iterate...

Proof sketch (cont.):  $f(X) = \sum_{F \in \Omega} r^F(X) = \sum_{F \in \Omega} r(X \setminus F)$ 

Try to maximize f without paying more than OPT.

 $\max\{f(X): c(X) \le OPT\}$ 

 $\Rightarrow$  Submodular Function maximization with knapsack constraint

NP-hard, but constant-factor approximations exist e.g.  $1 - \frac{1}{e} - \epsilon$  [Kulik, Shachnai, Tamir 10']

 $\Rightarrow$  Obtain a set  $Y_1$  with

•  $f(Y_1) \ge (1 - \alpha)r(\mathcal{M})|\Omega|$  for some  $\alpha \in (0, 1)$ •  $c(Y_1) \le OPT$ 

Iterate...

Proof sketch (cont.):  $f(X) = \sum_{F \in \Omega} r^F(X) = \sum_{F \in \Omega} r(X \setminus F)$ 

Try to maximize f without paying more than OPT.

 $\max\{f(X): c(X) \le OPT\}$ 

 $\Rightarrow$  Submodular Function maximization with knapsack constraint

NP-hard, but constant-factor approximations exist e.g.  $1 - \frac{1}{e} - \epsilon$  [Kulik, Shachnai, Tamir 10']

 $\Rightarrow$  Obtain a set  $Y_1$  with

•  $f(Y_1) \ge (1 - \alpha)r(\mathcal{M})|\Omega|$  for some  $\alpha \in (0, 1)$ 

•  $c(Y_1) \leq OPT$ 

lterate...

◆□ ▶ ◆□ ▶ ◆ 三 ▶ ◆ 三 ● ● ● ●

Proof sketch (cont.):  $f(X) = \sum_{F \in \Omega} r^F(X) = \sum_{F \in \Omega} r(X \setminus F)$ 

Try to maximize f without paying more than OPT.

 $\max\{f(X): c(X) \le OPT\}$ 

 $\Rightarrow$  Submodular Function maximization with knapsack constraint

NP-hard, but constant-factor approximations exist e.g.  $1 - \frac{1}{e} - \epsilon$  [Kulik, Shachnai, Tamir 10']

 $\Rightarrow$  Obtain a set  $Y_1$  with

- $f(Y_1) \ge (1 \alpha)r(\mathcal{M})|\Omega|$  for some  $\alpha \in (0, 1)$
- $c(Y_1) \leq OPT$

Iterate...

◆□ ▶ ◆□ ▶ ◆ 三 ▶ ◆ 三 ● ● ● ●

#### Proof sketch (cont.):

- Update N:  $N' = N \setminus Y_1$
- Update  $f: \quad f'(X) = \sum_{F \in \Omega} \left[ r^F(X \cup Y_1) r^F(Y_1) \right]$

(Contraction operation...)

⇒ Resolve approximately SFM problem to obtain  $Y_2$  with •  $f(Y_1 \cup Y_2) \ge (1 - \alpha^2) r(\mathcal{M}) |\Omega|$ 

•  $c(Y_1 \cup Y_2) \leq 20PT$ .

... after  $O(\log r(\mathcal{M})|\Omega|)$  iterations the solution is feasible!

*OPT* can be replaced with any  $T \leq OPT$  (binary search)

イロト イポト イヨト イヨト

Proof sketch (cont.):

Update N:  $N' = N \setminus Y_1$ 

Update f:  $f'(X) = \sum_{F \in \Omega} \left[ r^F(X \cup Y_1) - r^F(Y_1) \right]$ 

(Contraction operation...)

⇒ Resolve approximately SFM problem to obtain  $Y_2$  with •  $f(Y_1 \cup Y_2) \ge (1 - \alpha^2) r(\mathcal{M}) |\Omega|$ 

... after  $O(\log r(\mathcal{M})|\Omega|)$  iterations the solution is feasible!

*OPT* can be replaced with any  $T \leq OPT$  (binary search)

イロト イポト イヨト イヨト

Proof sketch (cont.):

Update N:  $N' = N \setminus Y_1$ 

Update  $f: \quad f'(X) = \sum_{F \in \Omega} \left[ r^F(X \cup Y_1) - r^F(Y_1) \right]$ 

(Contraction operation...)

⇒ Resolve approximately SFM problem to obtain  $Y_2$  with •  $f(Y_1 \cup Y_2) \ge (1 - \alpha^2) r(\mathcal{M}) |\Omega|$ 

... after  $O(\log r(\mathcal{M})|\Omega|)$  iterations the solution is feasible!

*OPT* can be replaced with any  $T \leq OPT$  (binary search)

イロン 不同 とくほう イロン

Proof sketch (cont.):

Update N:  $N' = N \setminus Y_1$ 

Update  $f: f'(X) = \sum_{F \in \Omega} [r^F(X \cup Y_1) - r^F(Y_1)]$ 

(Contraction operation...)

⇒ Resolve approximately SFM problem to obtain  $Y_2$  with •  $f(Y_1 \cup Y_2) \ge (1 - \alpha^2)r(\mathcal{M})|\Omega|$ •  $c(Y_1 \cup Y_2) \le 2OPT$ .

... after  $O(\log r(\mathcal{M})|\Omega|)$  iterations the solution is feasible!

*OPT* can be replaced with any  $T \leq OPT$  (binary search)

イロト イポト イヨト イヨト

Proof sketch (cont.):

Update N:  $N' = N \setminus Y_1$ 

Update  $f: f'(X) = \sum_{F \in \Omega} \left[ r^F(X \cup Y_1) - r^F(Y_1) \right]$ 

(Contraction operation...)

⇒ Resolve approximately SFM problem to obtain Y<sub>2</sub> with
f(Y<sub>1</sub> ∪ Y<sub>2</sub>) ≥ (1 − α<sup>2</sup>)r(M)|Ω|
c(Y<sub>1</sub> ∪ Y<sub>2</sub>) < 20PT.</li>

... after  $O(\log r(\mathcal{M})|\Omega|)$  iterations the solution is feasible!

*OPT* can be replaced with any *T* ≤ *OPT* (binary search)

Proof sketch (cont.):

Update N:  $N' = N \setminus Y_1$ 

Update  $f: f'(X) = \sum_{F \in \Omega} \left[ r^F(X \cup Y_1) - r^F(Y_1) \right]$ 

(Contraction operation...)

⇒ Resolve approximately SFM problem to obtain Y<sub>2</sub> with
 f(Y<sub>1</sub> ∪ Y<sub>2</sub>) ≥ (1 − α<sup>2</sup>)r(M)|Ω|
 c(Y<sub>1</sub> ∪ Y<sub>2</sub>) < 20PT.</li>

... after  $O(\log r(\mathcal{M})|\Omega|)$  iterations the solution is feasible!

*OPT* can be replaced with any *T* ≤ *OPT* (binary search)

イロン 不同 とくほう イロン

Proof sketch (cont.):

Update N:  $N' = N \setminus Y_1$ 

Update  $f: f'(X) = \sum_{F \in \Omega} \left[ r^F(X \cup Y_1) - r^F(Y_1) \right]$ 

(Contraction operation...)

⇒ Resolve approximately SFM problem to obtain Y<sub>2</sub> with
f(Y<sub>1</sub> ∪ Y<sub>2</sub>) ≥ (1 − α<sup>2</sup>)r(M)|Ω|
c(Y<sub>1</sub> ∪ Y<sub>2</sub>) ≤ 2OPT.
... after O(log r(M)|Ω|) iterations the solution is feasible!

*OPT* can be replaced with any  $T \leq OPT$  (binary search)

Proof sketch (cont.):

Update N:  $N' = N \setminus Y_1$ 

Update  $f: f'(X) = \sum_{F \in \Omega} \left[ r^F(X \cup Y_1) - r^F(Y_1) \right]$ 

(Contraction operation...)

⇒ Resolve approximately SFM problem to obtain  $Y_2$  with •  $f(Y_1 \cup Y_2) \ge (1 - \alpha^2) r(\mathcal{M}) |\Omega|$ 

... after  $O(\log r(\mathcal{M})|\Omega|)$  iterations the solution is feasible!

*OPT* can be replaced with any  $T \leq OPT$  (binary search)

Proof sketch (cont.):

Update N:  $N' = N \setminus Y_1$ 

Update  $f: f'(X) = \sum_{F \in \Omega} \left[ r^F(X \cup Y_1) - r^F(Y_1) \right]$ 

(Contraction operation...)

 $\Rightarrow$  Resolve approximately SFM problem to obtain  $Y_2$  with

- $f(Y_1 \cup Y_2) \ge (1 \alpha^2) r(\mathcal{M}) |\Omega|$
- $c(Y_1 \cup Y_2) \leq 2OPT$ .

... after  $O(\log r(\mathcal{M})|\Omega|)$  iterations the solution is feasible!

*OPT* can be replaced with any  $T \leq OPT$  (binary search)

Proof sketch (cont.):

Update N:  $N' = N \setminus Y_1$ 

Update  $f: f'(X) = \sum_{F \in \Omega} \left[ r^F(X \cup Y_1) - r^F(Y_1) \right]$ 

(Contraction operation...)

 $\Rightarrow$  Resolve approximately SFM problem to obtain  $\emph{Y}_2$  with

- $f(Y_1 \cup Y_2) \ge (1 \alpha^2) r(\mathcal{M}) |\Omega|$
- $c(Y_1 \cup Y_2) \le 2OPT$ .

... after  $O(\log r(\mathcal{M})|\Omega|)$  iterations the solution is feasible!

*OPT* can be replaced with any  $T \leq OPT$  (binary search)

Proof sketch (cont.):

Update N:  $N' = N \setminus Y_1$ 

Update  $f: f'(X) = \sum_{F \in \Omega} \left[ r^F(X \cup Y_1) - r^F(Y_1) \right]$ 

(Contraction operation...)

 $\Rightarrow$  Resolve approximately SFM problem to obtain  $Y_2$  with

- $f(Y_1 \cup Y_2) \ge (1 \alpha^2)r(\mathcal{M})|\Omega|$
- $c(Y_1 \cup Y_2) \leq 2OPT$ .

... after  $O(\log r(\mathcal{M})|\Omega|)$  iterations the solution is feasible!

*OPT* can be replaced with any  $T \leq OPT$  (binary search)

Proof sketch (cont.):

Update N:  $N' = N \setminus Y_1$ 

Update  $f: f'(X) = \sum_{F \in \Omega} \left[ r^F(X \cup Y_1) - r^F(Y_1) \right]$ 

(Contraction operation...)

 $\Rightarrow$  Resolve approximately SFM problem to obtain  $Y_2$  with

- $f(Y_1 \cup Y_2) \ge (1 \alpha^2)r(\mathcal{M})|\Omega|$
- $c(Y_1 \cup Y_2) \leq 2OPT$ .

... after  $O(\log r(\mathcal{M})|\Omega|)$  iterations the solution is feasible!

*OPT* can be replaced with any  $T \leq OPT$  (binary search)

◆□ ▶ ◆□ ▶ ◆ □ ▶ ◆ □ ● ● ● ● ● ●

Proof sketch (cont.):

Update N:  $N' = N \setminus Y_1$ 

Update  $f: f'(X) = \sum_{F \in \Omega} \left[ r^F(X \cup Y_1) - r^F(Y_1) \right]$ 

(Contraction operation...)

 $\Rightarrow$  Resolve approximately SFM problem to obtain  $Y_2$  with

- $f(Y_1 \cup Y_2) \ge (1 \alpha^2)r(\mathcal{M})|\Omega|$
- $c(Y_1 \cup Y_2) \leq 2OPT$ .

... after  $O(\log r(\mathcal{M})|\Omega|)$  iterations the solution is feasible!

*OPT* can be replaced with any  $T \leq OPT$  (binary search)

Proof sketch (cont.):

Update N:  $N' = N \setminus Y_1$ 

Update  $f: f'(X) = \sum_{F \in \Omega} \left[ r^F(X \cup Y_1) - r^F(Y_1) \right]$ 

(Contraction operation...)

 $\Rightarrow$  Resolve approximately SFM problem to obtain  $Y_2$  with

- $f(Y_1 \cup Y_2) \ge (1 \alpha^2)r(\mathcal{M})|\Omega|$
- $c(Y_1 \cup Y_2) \leq 2OPT$ .

... after  $O(\log r(\mathcal{M})|\Omega|)$  iterations the solution is feasible!

*OPT* can be replaced with any  $T \leq OPT$  (binary search)

Bulk(Shortest Path) can be even harder:

Steiner Forest can be modeled as Bulk(Shortest Path)

 $\Rightarrow$  No  $O(2^{\log^{1-\epsilon}|V|})$ -approximations for directed graphs.

However: when  $k = \max_{F \in \Omega} |F|$  bounded

**Theorem.** Bulk(Shortest Path) is

polynomial when  $k \leq 1$ .

APX-complete when k = 2.

 $O(k \log |\Omega|)$ -approximable when  $k = O(\log |V|)$  (Undir.).

< ロ > < 同 > < 回 > < 回 >

#### Bulk(Shortest Path)

#### Bulk(Shortest Path) can be even harder:

< ロ > < 同 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ >

Bulk(Shortest Path) can be even harder:

#### Steiner Forest can be modeled as Bulk(Shortest Path)

 $\Rightarrow$  No  $O(2^{\log^{1-\epsilon}|V|})$ -approximations for directed graphs.

However: when  $k = \max_{F \in \Omega} |F|$  bounded

Theorem. Bulk(Shortest Path) is

polynomial when  $k \leq 1$ .

APX-complete when k = 2.

 $O(k \log |\Omega|)$ -approximable when  $k = O(\log |V|)$  (Undir.).

(1)
Steiner Forest can be modeled as Bulk(Shortest Path)

 $\Rightarrow$  No  $O(2^{\log^{1-\epsilon}|V|})$ -approximations for directed graphs.

However: when  $k = \max_{F \in \Omega} |F|$  bounded

Theorem. Bulk(Shortest Path) is

polynomial when  $k \leq 1$ .

APX-complete when k = 2.

 $O(k \log |\Omega|)$ -approximable when  $k = O(\log |V|)$  (Undir.).

・ロト ・ 日 ・ ・ 日 ・ ・ 日 ・

Steiner Forest can be modeled as Bulk(Shortest Path)

 $\Rightarrow$  No  $O(2^{\log^{1-\epsilon}|V|})$ -approximations for directed graphs.

However: when  $\mathbf{k} = \max_{F \in \Omega} |F|$  bounded

**Theorem.** Bulk(Shortest Path) is polynomial when  $k \le 1$ . APX-complete when k = 2.  $O(k \log |\Omega|)$ -approximable when  $k = O(\log |V|)$  (Undir.).

イロト 不得 とくほ とくほう

Steiner Forest can be modeled as Bulk(Shortest Path)

 $\Rightarrow$  No  $O(2^{\log^{1-\epsilon}|V|})$ -approximations for directed graphs.

However: when  $k = \max_{F \in \Omega} |F|$  bounded

**Theorem.** Bulk(Shortest Path) is polynomial when  $k \leq 1$ . APX-complete when k = 2.  $O(k \log |\Omega|)$ -approximable when  $k = O(\log |V|)$  (Undir.).

イロト 不得 トイヨト イヨト

Steiner Forest can be modeled as Bulk(Shortest Path)

 $\Rightarrow$  No  $O(2^{\log^{1-\epsilon}|V|})$ -approximations for directed graphs.

However: when  $\mathbf{k} = \max_{\mathbf{F} \in \Omega} |\mathbf{F}|$  bounded

**Theorem.** Bulk(Shortest Path) is

polynomial when  $k \leq 1$ .

APX-complete when k = 2.

 $O(k \log |\Omega|)$ -approximable when  $k = O(\log |V|)$  (Undir.).

・ロト ・ 一下・ ・ ヨト ・ ヨト

Steiner Forest can be modeled as Bulk(Shortest Path)

 $\Rightarrow$  No  $O(2^{\log^{1-\epsilon}|V|})$ -approximations for directed graphs.

However: when  $\mathbf{k} = \max_{\mathbf{F} \in \Omega} |\mathbf{F}|$  bounded

**Theorem.** Bulk(Shortest Path) is polynomial when  $k \le 1$ . APX-complete when k = 2.  $O(k \log |\Omega|)$ -approximable when  $k = O(\log |V|)$  (Undir.).

Theorem. Bulk(Shortest Path) is

APX-complete when k = 2.

- (目) - (目) - (目)

э

**Theorem.** Bulk(Shortest Path)

admits a polynomial 13-apx when k = 2.

3

Proof sketch.

- 4 回 2 - 4 □ 2 - 4 □

æ

*Proof sketch.* (*G undirected*)

・ロト ・回ト ・ヨト ・ヨト

æ

Proof sketch. (G undirected)

**Lemma 1.** When k = 1 a minimal feasible solution is a "simple" union of two *s*-*t* paths.

< ロ > < 同 > < 回 > < 回 > < □ > <

3

*Proof sketch.* (*G undirected*)

**Lemma 1.** When k = 1 a minimal feasible solution is a "simple" union of two *s*-*t* paths.



< 回 > < 回 > < 回 >

*Proof sketch.* (*G undirected*)

**Lemma 1.** When k = 1 a minimal feasible solution is a "simple" union of two *s*-*t* paths.



**Step 1:** Solve for  $\Omega_1 = \{\{e\} \subset E : \exists F \in \Omega : e \in F\}$ 

(人間) ト く ヨ ト く ヨ ト

*Proof sketch.* (*G undirected*)

**Lemma 1.** When k = 1 a minimal feasible solution is a "simple" union of two *s*-*t* paths.



**Step 1:** Solve for  $\Omega_1 = \{\{e\} \subset E : \exists F \in \Omega : e \in F\} \Rightarrow S_1$ 

- 4 同 6 4 日 6 4 日 6

*Proof sketch.* (*G undirected*)

**Lemma 1.** When k = 1 a minimal feasible solution is a "simple" union of two *s*-*t* paths.



**Step 1:** Solve for  $\Omega_1 = \{\{e\} \subset E : \exists F \in \Omega : e \in F\} \Rightarrow S_1$  $\Rightarrow c(S_1) \leq OPT$ 

- 4 同 6 4 日 6 4 日 6

*Proof sketch.* (*G undirected*)

**Lemma 1.** When k = 1 a minimal feasible solution is a "simple" union of two *s*-*t* paths.



**Step 1:** Solve for  $\Omega_1 = \{\{e\} \subset E : \exists F \in \Omega : e \in F\} \Rightarrow S_1$  $\Rightarrow c(S_1) \leq OPT$ 

 $\Rightarrow$  If  $F \in \Omega$  is an *s*-*t* in *S*<sub>1</sub> then *F* is contained in a simple cycle.

(日) (同) (三) (三)

Proof sketch (cont.)

- 4 回 > - 4 回 > - 4 回 >

æ

Proof sketch (cont.)

**Lemma 2.** Minimal X such that  $X \cup S_1$  is feasible is a forest.

・ロト ・回ト ・ヨト ・ヨト

3

Proof sketch (cont.)

**Lemma 2.** Minimal X such that  $X \cup S_1$  is feasible is a forest.

(Call such X an *augmenting set*)

・ 同 ト ・ ヨ ト ・ ヨ ト

**Lemma 2.** Minimal X such that  $X \cup S_1$  is feasible is a forest.

(Call such X an *augmenting set*)

**Lemma 3.** Let X be an augmenting set. There exist paths  $P_1, \dots, P_k \subset E$  such that

- $\forall F \in \Omega \exists i \in [k]$  such that F is not an s-t cut in  $S_1 \cup P_i$ .
- $c(P_1) + \cdots + c(P_k) \leq 2c(X)$ .

イロン 不同 とくほう イロン

**Lemma 2.** Minimal X such that  $X \cup S_1$  is feasible is a forest.

(Call such X an *augmenting set*)

**Lemma 3.** Let X be an augmenting set. There exist paths  $P_1, \dots, P_k \subset E$  such that

- $\forall F \in \Omega \ \exists i \in [k]$  such that F is not an s-t cut in  $S_1 \cup P_i$ .
- $c(P_1) + \cdots + c(P_k) \leq 2c(X)$ .

Furthermore,  $P_i$  can be replaced by any shortest path between its endpoints.

イロン 不同 とくほう イロン

**Lemma 2.** Minimal X such that  $X \cup S_1$  is feasible is a forest.

(Call such X an *augmenting set*)

**Lemma 3.** Let X be an augmenting set. There exist paths  $P_1, \dots, P_k \subset E$  such that

- $\forall F \in \Omega \ \exists i \in [k]$  such that F is not an s-t cut in  $S_1 \cup P_i$ .
- $c(P_1) + \cdots + c(P_k) \leq 2c(X)$ .

Furthermore,  $P_i$  can be replaced by any shortest path between its endpoints.

**Step 2:** Compute shortest paths  $Q_{u,v}$  for all  $u, v \in V[S_1]$ 

< ロ > < 同 > < 回 > < 回 > < □ > <

**Lemma 2.** Minimal X such that  $X \cup S_1$  is feasible is a forest.

(Call such X an *augmenting set*)

**Lemma 3.** Let X be an augmenting set. There exist paths  $P_1, \dots, P_k \subset E$  such that

- $\forall F \in \Omega \exists i \in [k]$  such that F is not an s-t cut in  $S_1 \cup P_i$ .
- $c(P_1) + \cdots + c(P_k) \leq 2c(X)$ .

Furthermore,  $P_i$  can be replaced by any shortest path between its endpoints.

**Step 2:** Compute shortest paths  $Q_{u,v}$  for all  $u, v \in V[S_1]$ 

 $\Rightarrow$  Solve Set Cover problem

**Lemma 2.** Minimal X such that  $X \cup S_1$  is feasible is a forest.

(Call such X an *augmenting set*)

**Lemma 3.** Let X be an augmenting set. There exist paths  $P_1, \dots, P_k \subset E$  such that

- $\forall F \in \Omega \exists i \in [k]$  such that F is not an s-t cut in  $S_1 \cup P_i$ .
- $c(P_1) + \cdots + c(P_k) \leq 2c(X)$ .

Furthermore,  $P_i$  can be replaced by any shortest path between its endpoints.

**Step 2:** Compute shortest paths  $Q_{u,v}$  for all  $u, v \in V[S_1]$ 

 $\Rightarrow$  Solve Set Cover problem ( $\Rightarrow O(\log n)$ -apx)

< ロ > < 同 > < 回 > < 回 > < □ > <

Proof sketch (cont.)

- 4 回 > - 4 回 > - 4 回 >

æ

Proof sketch (cont.)

 $\Rightarrow$  Want a O(1)-apx for that Set Cover problem

3

Proof sketch (cont.)

 $\Rightarrow$  Want a O(1)-apx for that Set Cover problem

**Step 3:** "Unfold"  $S_1$  into two disjoint paths.

イロト イポト イヨト イヨト

3

Proof sketch (cont.)

 $\Rightarrow$  Want a O(1)-apx for that Set Cover problem

**Step 3:** "Unfold"  $S_1$  into two disjoint paths.



A 10

3 N

Proof sketch (cont.)

 $\Rightarrow$  Want a O(1)-apx for that Set Cover problem

**Step 3:** "Unfold"  $S_1$  into two disjoint paths.



 $F = \{f_1, f_2\}$ 

▲ □ ▶ ▲ □ ▶

Proof sketch (cont.)

 $\Rightarrow$  Want a O(1)-apx for that Set Cover problem

**Step 3:** "Unfold"  $S_1$  into two disjoint paths.



イロト 不得 トイヨト イヨト 二日

Proof sketch (cont.)

 $\Rightarrow$  Want a O(1)-apx for that Set Cover problem

**Step 3:** "Unfold"  $S_1$  into two disjoint paths.



Proof sketch (cont.)

- 4 回 > - 4 回 > - 4 回 >

æ

Proof sketch (cont.)

Every path in  $S_1$  defines an ordering of  $\Omega$ ,

э

Proof sketch (cont.)

Every path in  $S_1$  defines an ordering of  $\Omega$ , but...



(4 同) (4 日) (4 日)

Proof sketch (cont.)

Every path in  $S_1$  defines an ordering of  $\Omega$ , but...



Restricted to such paths Q - Interval cover!

直 ト イヨ ト イヨ ト

Proof sketch (cont.)

Every path in  $S_1$  defines an ordering of  $\Omega$ , but...



(4 同) (4 日) (4 日)

Proof sketch (cont.)

Every path in  $S_1$  defines an ordering of  $\Omega$ , but...



Restricted to such paths Q...

- **→** → **→**
Proof sketch (cont.)

Every path in  $S_1$  defines an ordering of  $\Omega$ , but...



Restricted to such paths Q... also Interval cover!

- **→** → **→** 

Proof sketch (cont.)

Every path in  $S_1$  defines an ordering of  $\Omega$ , but...



Which scenarios F to cover with which paths?

- **→** → **→** 

Proof sketch (cont.)

Every path in  $S_1$  defines an ordering of  $\Omega$ , but...



#### Which scenarios *F* to cover with which paths? Ask the LP!

- 4 周 ト 4 戸 ト 4 戸 ト

Proof sketch (cont.)

- 4 回 > - 4 回 > - 4 回 >

æ

Proof sketch (cont.)

Step 4: Solve

 $\min\left\{\sum_{u,v\in V} c(Q_{u,v}) x_{u,v} \ : \ \sum_{F\times (u,v)} x_{u,v} \ge 1 \ \forall F \in \Omega\right\}$ 

イロン 不同 とくほう イロン

Proof sketch (cont.)

Step 4: Solve

 $\min\left\{\sum_{u,v\in V} c(Q_{u,v}) x_{u,v} \ : \ \sum_{F\times (u,v)} x_{u,v} \ge 1 \ \forall F \in \Omega\right\}$ 

 $\Rightarrow x^*$ 

イロン 不同 とくほう イロン

Proof sketch (cont.)

Step 4: Solve

 $\min\left\{\sum_{u,v\in V} c(Q_{u,v}) x_{u,v} : \sum_{F\times(u,v)} x_{u,v} \ge 1 \ \forall F \in \Omega\right\}$  $\Rightarrow x^* \qquad (\text{Lemma 3:} \quad c(x^*) \le 2OPT)$ 

イロト 不得 トイヨト イヨト 二日

Proof sketch (cont.)

Step 4: Solve

 $\min\left\{\sum_{u,v\in V} c(Q_{u,v}) x_{u,v} \ : \ \sum_{F\times (u,v)} x_{u,v} \ge 1 \ \forall F \in \Omega\right\}$ 

イロン 不同 とくほう イロン

Proof sketch (cont.)

Step 4: Solve

 $\min\left\{\sum_{u,v\in V} c(Q_{u,v}) x_{u,v} \ : \ \sum_{F\times (u,v)} x_{u,v} \geq 1 \ \forall F \in \Omega\right\}$ 





▲ □ ▶ ▲ □ ▶ ▲ □ ▶

Proof sketch (cont.)

Step 4: Solve

 $\min\left\{\sum_{u,v\in V} c(Q_{u,v}) x_{u,v} \ : \ \sum_{F\times (u,v)} x_{u,v} \ge 1 \ \forall F \in \Omega\right\}$ 



・ 同 ト ・ ヨ ト ・ ヨ ト

Proof sketch (cont.)

Step 4: Solve

 $\min\left\{\sum_{u,v\in V} c(Q_{u,v}) x_{u,v} \ : \ \sum_{F\times (u,v)} x_{u,v} \geq 1 \ \forall F \in \Omega\right\}$ 



type(F) = 3

(人間) ト く ヨ ト く ヨ ト

Proof sketch (cont.)

Step 4: Solve

 $\min\left\{\sum_{u,v\in V} c(Q_{u,v}) x_{u,v} \ : \ \sum_{F\times (u,v)} x_{u,v} \ge 1 \ \forall F \in \Omega\right\}$ 



(人間) ト く ヨ ト く ヨ ト

Partition  $\Omega$  by type  $\Rightarrow \Omega_1, \cdots, \Omega_4$ .

・ロト ・回ト ・ヨト ・ヨト

Partition  $\Omega$  by type  $\Rightarrow \Omega_1, \cdots, \Omega_4$ .

 $\Omega_i$  is fractionally covered by  $6x^*$  for i = 1, 2, and

イロン 不同 とくほう イロン

Partition  $\Omega$  by type  $\Rightarrow \Omega_1, \cdots, \Omega_4$ .

 $\Omega_i$  is fractionally covered by  $6x^*$  for i = 1, 2, and

 $\Omega_i$  is fractionally covered by  $3x^*$  for i = 3, 4.

イロト イポト イヨト イヨト

Partition  $\Omega$  by type  $\Rightarrow \Omega_1, \cdots, \Omega_4$ .

 $\Omega_i$  is fractionally covered by  $6x^*$  for i = 1, 2, and

 $\Omega_i$  is fractionally covered by  $3x^*$  for i = 3, 4.

Step 5: Remove redundant scenarios for types 3,4

イロン 不同 とくほう イロン

Partition  $\Omega$  by type  $\Rightarrow \Omega_1, \cdots, \Omega_4$ .

 $\Omega_i$  is fractionally covered by  $6x^*$  for i = 1, 2, and

 $\Omega_i$  is fractionally covered by  $3x^*$  for i = 3, 4.

Step 5: Remove redundant scenarios for types 3,4

**Step 6:** Solve Interval Cover problems  $\Rightarrow$   $T_1$ ,  $\cdots$ ,  $T_4$ 

- Partition  $\Omega$  by type  $\Rightarrow \Omega_1, \cdots, \Omega_4$ .
- $\Omega_i$  is fractionally covered by  $6x^*$  for i = 1, 2, and
- $\Omega_i$  is fractionally covered by  $3x^*$  for i = 3, 4.

Step 5: Remove redundant scenarios for types 3,4

- **Step 6:** Solve Interval Cover problems  $\Rightarrow$   $T_1$ ,  $\cdots$ ,  $T_4$
- **Step 6:** Return  $S_1 \cup T_1 \cup \cdots \cup T_4$

- Partition  $\Omega$  by type  $\Rightarrow \Omega_1, \cdots, \Omega_4$ .
- $\Omega_i$  is fractionally covered by  $6x^*$  for i = 1, 2, and
- $\Omega_i$  is fractionally covered by  $3x^*$  for i = 3, 4.

Step 5: Remove redundant scenarios for types 3,4

- **Step 6:** Solve Interval Cover problems  $\Rightarrow$   $T_1$ ,  $\cdots$ ,  $T_4$
- **Step 6:** Return  $S_1 \cup T_1 \cup \cdots \cup T_4$

Partitioning of x\* plus integrality of Interval Cover

• Given  $\Omega = \{F_1, \cdots, F_m\}$  and  $k \in \mathbb{N}$  find, the scenario set is

$$\Omega^k = \left\{ Y = \bigcup_{F \in \Omega'} F \; : \; \Omega' \subset \Omega, \; |\Omega'| \leq k 
ight\}$$

Perhaps more importantly: reliability analysis (interdiction) • Find  $\Omega' \subset \Omega^k$  such that C = [1] = -E

Interdiction in networks with a diffusion dynamic

(Threshold model, Cascade model...)

- 4 同 2 4 日 2 4 日 2

• Given  $\Omega = \{F_1, \cdots, F_m\}$  and  $k \in \mathbb{N}$  find, the scenario set is

$$\Omega^k = \left\{ Y = \bigcup_{F \in \Omega'} F \; : \; \Omega' \subset \Omega, \; |\Omega'| \leq k 
ight\}$$

Perhaps more importantly: reliability analysis (interdiction) • Find  $\Omega' \subset \Omega^k$  such that  $G = \bigcup_{n \in \Omega} F_n$ 

• Interdiction in networks with a diffusion dynamics

(Threshold model, Cascade model...)

< ロ > < 同 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ >

• Given  $\Omega = \{F_1, \cdots, F_m\}$  and  $k \in \mathbb{N}$  find, the scenario set is

$$\Omega^{\boldsymbol{k}} = \left\{ \boldsymbol{Y} = \bigcup_{\boldsymbol{F} \in \Omega'} \boldsymbol{F} : \ \Omega' \subset \Omega, \ |\Omega'| \leq \boldsymbol{k} \right\}$$

Perhaps more importantly: reliability analysis (interdiction)
 Find Ω' ⊂ Ω<sup>k</sup> such that G − U<sub>F∈Ω'</sub> F...

• Interdiction in networks with a diffusion dynamics

(Threshold model, Cascade model...)

< ロ > < 同 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ >

• Given  $\Omega = \{F_1, \cdots, F_m\}$  and  $k \in \mathbb{N}$  find, the scenario set is

$$\Omega^{\boldsymbol{k}} = \left\{ \boldsymbol{Y} = \bigcup_{\boldsymbol{F} \in \Omega'} \boldsymbol{F} : \ \Omega' \subset \Omega, \ |\Omega'| \leq \boldsymbol{k} \right\}$$

Perhaps more importantly: reliability analysis (interdiction)

- Find  $\Omega' \subset \Omega^k$  such that  $G \bigcup_{F \in \Omega'} F$ ...
- Interdiction in networks with a diffusion dynamics

(Threshold model, Cascade model...)

• Given  $\Omega = \{F_1, \cdots, F_m\}$  and  $k \in \mathbb{N}$  find, the scenario set is

$$\Omega^{\boldsymbol{k}} = \left\{ \boldsymbol{Y} = \bigcup_{\boldsymbol{F} \in \Omega'} \boldsymbol{F} : \ \Omega' \subset \Omega, \ |\Omega'| \leq \boldsymbol{k} \right\}$$

Perhaps more importantly: reliability analysis (interdiction)

- Find  $\Omega' \subset \Omega^k$  such that  $G \bigcup_{F \in \Omega'} F$ ...
- Interdiction in networks with a diffusion dynamics

(Threshold model, Cascade model...)

• Given  $\Omega = \{F_1, \cdots, F_m\}$  and  $k \in \mathbb{N}$  find, the scenario set is

$$\Omega^{\boldsymbol{k}} = \left\{ \boldsymbol{Y} = \bigcup_{\boldsymbol{F} \in \Omega'} \boldsymbol{F} \; : \; \Omega' \subset \Omega, \; |\Omega'| \leq \boldsymbol{k} \right\}$$

Perhaps more importantly: reliability analysis (interdiction)

• Find  $\Omega' \subset \Omega^k$  such that  $G - \bigcup_{F \in \Omega'} F$ ...

Interdiction in networks with a diffusion dynamics

(Threshold model, Cascade model...)

イロン 不同 とくほう イロン

• Given  $\Omega = \{F_1, \cdots, F_m\}$  and  $k \in \mathbb{N}$  find, the scenario set is

$$\Omega^{\pmb{k}} = \left\{ Y = \bigcup_{F \in \Omega'} F \; : \; \Omega' \subset \Omega, \; |\Omega'| \le \pmb{k} \right\}$$

Perhaps more importantly: reliability analysis (interdiction)

- Find  $\Omega' \subset \Omega^k$  such that  $G \bigcup_{F \in \Omega'} F$ ...
- Interdiction in networks with a diffusion dynamics

(Threshold model, Cascade model...)

• Given  $\Omega = \{F_1, \cdots, F_m\}$  and  $k \in \mathbb{N}$  find, the scenario set is

$$\Omega^{\pmb{k}} = \left\{ Y = \bigcup_{F \in \Omega'} F \; : \; \Omega' \subset \Omega, \; |\Omega'| \le \pmb{k} \right\}$$

Perhaps more importantly: reliability analysis (interdiction)

- Find  $\Omega' \subset \Omega^k$  such that  $G \bigcup_{F \in \Omega'} F$ ...
- Interdiction in networks with a diffusion dynamics

(Threshold model, Cascade model...)

イロト 不得 トイヨト イヨト 二日

#### Nonuniform...

# $\mathsf{T} H \mathsf{A} \mathsf{N} \mathsf{K} \mathsf{Y} \mathsf{O} \mathsf{U}$

・ロン ・四 と ・ ヨ と ・ ヨ と

æ