Optimal Systemic Risk Bailout:
A PGO Approach Based on Neural Network

肖书华 中山大学管理学院

合作者：马家丽博士，夏俐教授，朱书尚教授

中国运筹学会金融工程与金融风险管理分会第十一届学术年会

2022.12.11

Mind Map

Shock
↓
Banking System → Systemic Risk
↑
Optimal Systemic Risk Bailout

$$(P) \quad \begin{aligned} \max \quad & \boxed{f(\boldsymbol{x})} \\ \text{s.t.} \quad & g(\boldsymbol{x}) \leq 0 \\ & h(\boldsymbol{x}) = 0, \end{aligned}$$

GAP: no closed form

$$(P) \quad \begin{aligned} \max \quad & \boxed{f(\boldsymbol{x})} \\ \text{s.t.} \quad & g(\boldsymbol{x}) \leq 0 \\ & h(\boldsymbol{x}) = 0, \end{aligned}$$

⇒ Neural Network / Back-Propagation ⇒

$$(P) \quad \begin{aligned} \max \quad & \hat{f}(\boldsymbol{x}) \\ \text{s.t.} \quad & g(\boldsymbol{x}) \leq 0 \\ & h(\boldsymbol{x}) = 0, \end{aligned}$$

⇒ $\nabla \hat{f}(\boldsymbol{x})$ ⇒

Optimal Systemic Risk Bailout

Optimal Systemic Risk Bailout:
A PGO Approach Based on Neural Network

Overview (Optimal bailout!)

Bailout Measurement (What is it?)

Framework (How to find it?)

Simulating Results

Discussion & Conclusion

Overview: Background

On October 14, 2008, the Treasury Department used $105 billion in TARP funds to launch the Capital Purchase Program, which purchased preferred stock in the eight leading banks. By the time TARP expired on October 3, 2010, Treasury had used the funds in four other areas:[2]

Bailout: costly but necessary!

How to find out optimal **systemic risk** bailout ?

↓

E-N Models

Source: https://www.thebalance.com/what-was-the-bank-bailout-bill-3305675

Overview: Motivation

中山大学管理学院
SCHOOL OF BUSINESS
SUN YAT-SEN UNIVERSITY

How to find out **optimal** systemic risk **bailout** ?

Model	E-N model	E-N +Default costs	E-N + Market value/Cross hold	E-N +Multiple illiquid assets
Source of Model	Eisenberg & Noe (2001)	Rogers & Veraart (2013)	……	Feinstein (2017) Ma et.al(2021)
Property	Linear Programming	Non-deterministic Polynomial hard (NP hard)	……	Objective function **with no closed form**
Research	Pokutta et.al(2011)	Jackson & Pernoud (2020): A simple algorithm (by order)	Demange and Gabrielle(2018): A threat index (by index)	Ma et.al(2021): A heuristic algorithm

Overview: Motivation

How to find out **optimal systemic risk bailout** ?

Model	**E-N model**	**E-N +Default costs**	**E-N + Market value/Cross hold**	**E-N +Multiple illiquid assets**
Source of Model	Eisenberg & Noe (2001)	Rogers & Veraart (2013)		Feinstein (2017) Ma et.al(2021)
Property	Linear Programming	Non-deterministic Polynomial hard (NP hard)	……	Objective function **with no closed form**
Research	Pokutta et.al(2011)	Jackson & Pernoud (2020): A simple algorithm (by order)	Demange and Gabrielle(2018): A threat index (by index)	Ma et.al(2021): A heuristic algorithm

Explore the black box !

Optimal Systemic Risk Bailout:
A PGO Approach Based on Neural Network

Overview (Optimal bailout!)

- Background
- Motivation

Bailout Measurement (What is it?)

Framework (How to find it?)

Simulating Results

Discussion & Conclusion

Optimal Systemic Risk Bailout:
A PGO Approach Based on Neural Network

Overview (Optimal bailout!)

- Background
- Motivation

Bailout Measurement (What is it?)

- Definitions
- Two Cases

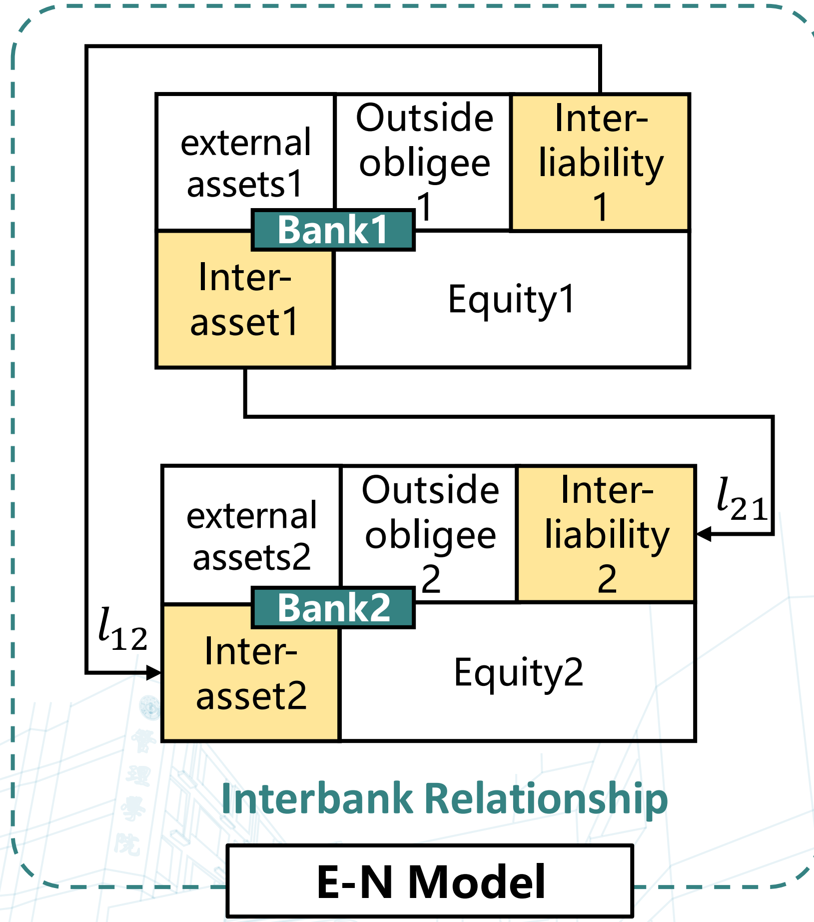
Framework (How to find it?)

Simulating Results

Discussion & Conclusion

Bailout Measurement

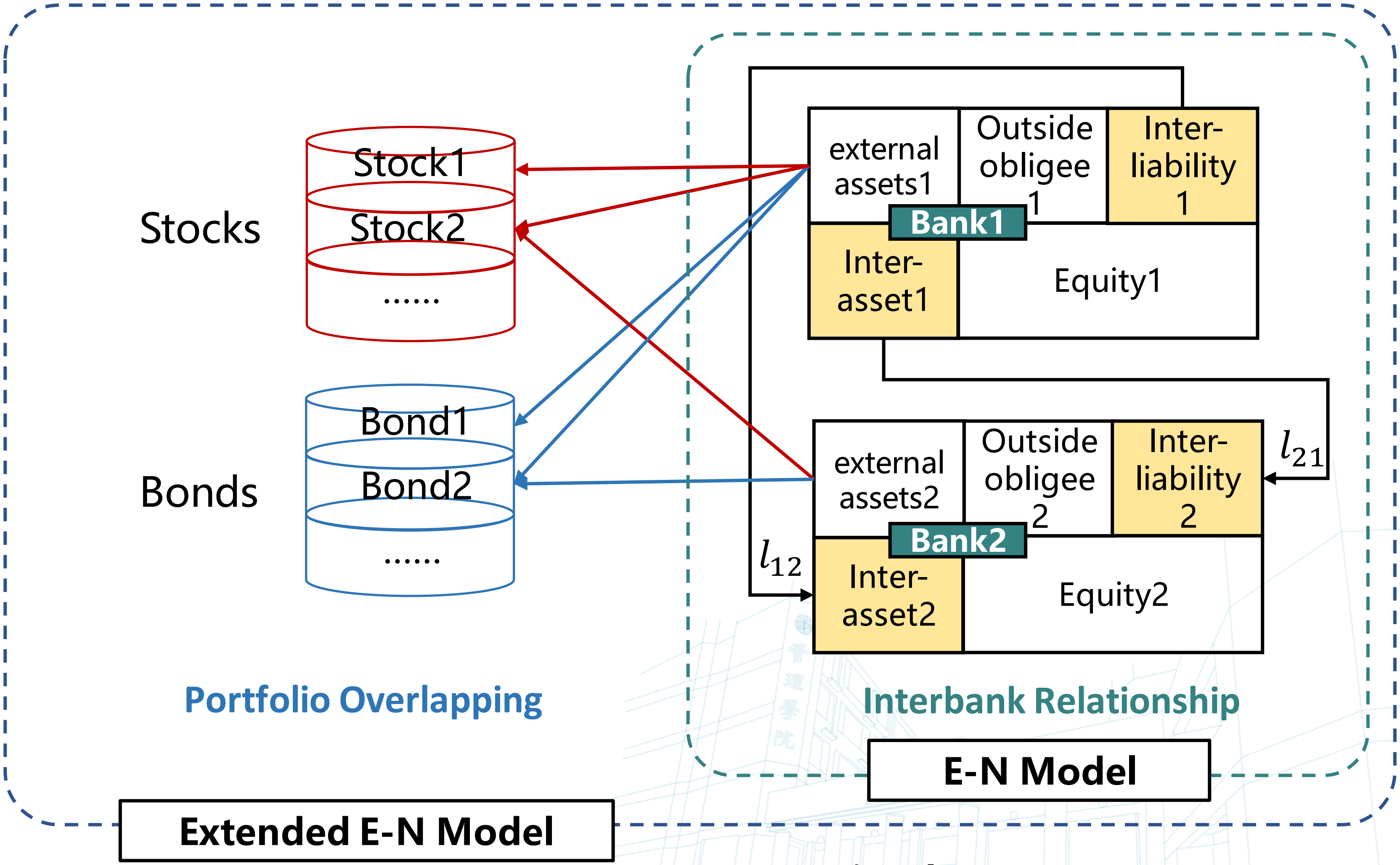
- ## Definitions of the Financial System



Eisenberg & Noe(2001)

Bailout Measurement

- **Definitions of the Financial System**



Eisenberg & Noe(2001)

Feinstein (2017)

Ma et.al (2021)

Bailout Measurement

- ## Definitions of the Financial System

E-N Model

Interbank Relationship

Fixed−point System ⟹ Clearing Vector $\boldsymbol{l}^*$

Extended E-N Model

Interbank Relationship
\+ Portfolio Overlapping

Fixed−point System ⟹ Clearing Vector $\boldsymbol{l}^*$; Price Vector $\boldsymbol{p}^*$

Bailout Measurement

- **Definitions of the Objective Function**

$$Pay_{all} = \mathbf{1}^T \boldsymbol{l}^*(\tilde{\boldsymbol{c}})$$

$$Save_{all} = \tilde{\boldsymbol{c}} + \Pi^T[\boldsymbol{l}^*(\tilde{\boldsymbol{c}}) - \boldsymbol{l}^*(\boldsymbol{s})] + (\mathbf{1}^T - \mathbf{1}^T\Pi)[\boldsymbol{l}^*(\tilde{\boldsymbol{c}}) - \boldsymbol{l}^*(\boldsymbol{s})] + A[\boldsymbol{p}^*(\tilde{\boldsymbol{c}}) - \boldsymbol{p}^*(\boldsymbol{s})]$$

$$Ratio = \frac{Save_{all}}{\tau}$$

$\boldsymbol{l}^*$: Clearing Vector

$\boldsymbol{p}^*$: Price Vector

Π: The relative liability matrix

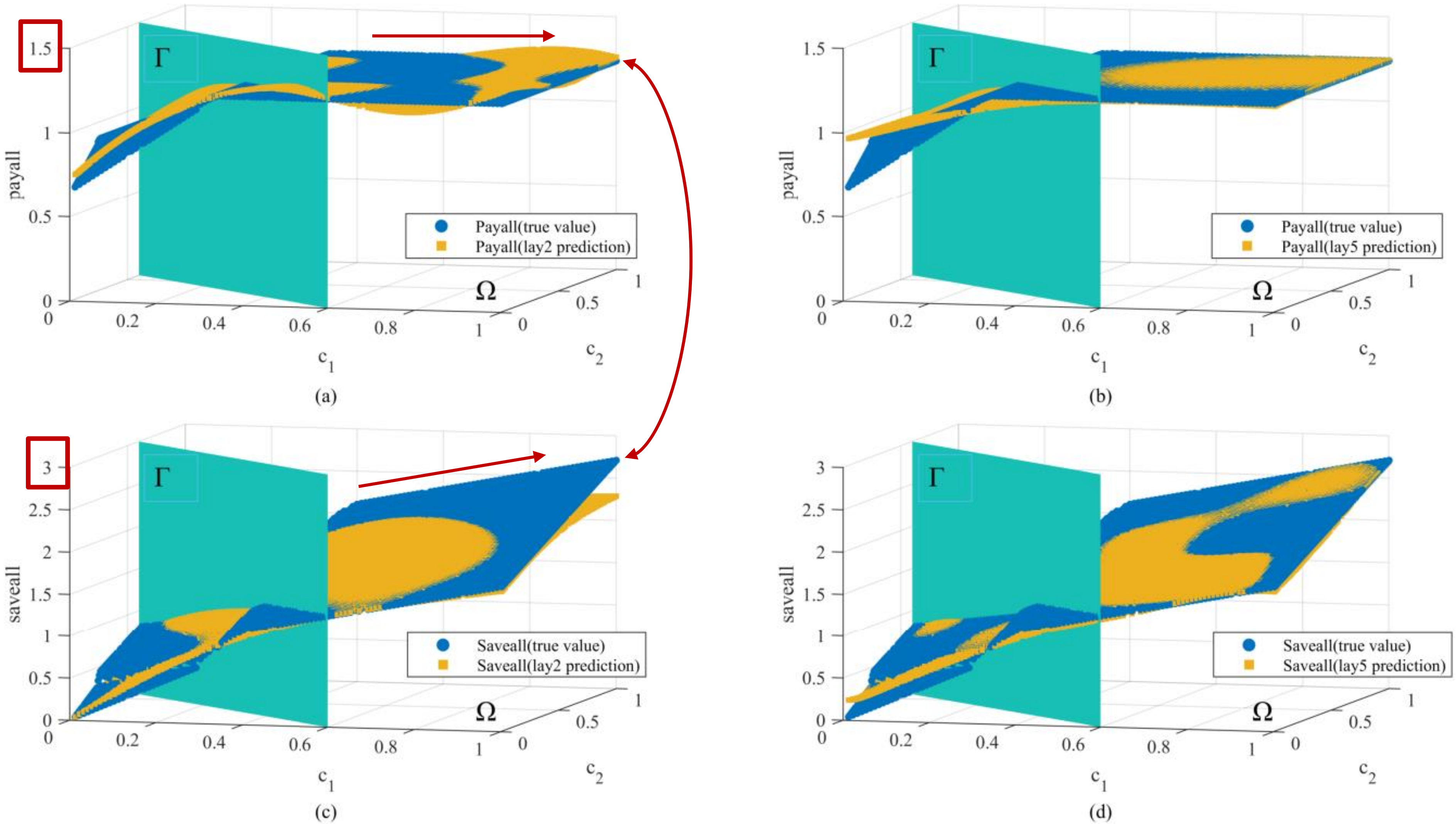
$\tilde{\boldsymbol{c}}$: The bailout vector

A: The portfolio holdings

$\boldsymbol{s} = (s_i) \in \mathbb{R}_+^n$: The initial shock

Bailout Measurement

- **Definitions of the Objective Function**



Optimal Systemic Risk Bailout:
A PGO Approach Based on Neural Network

Overview (Optimal bailout!)

- Background
- Motivation

Bailout Measurement (What is it?)

- Definitions
- Two Cases

Framework (How to find it?)

Simulating Results

Discussion & Conclusion

Optimal Systemic Risk Bailout:
A PGO Approach Based on Neural Network

Overview (Optimal bailout!)

- Background
- Motivation

Bailout Measurement (What is it?)

- Definitions
- Two Cases

Framework (How to find it?)

- Prediction
- Gradient
- Optimization

Simulating Results

Discussion & Conclusion

Framework: "Prediction-Gradient-Optimization" (PGO)

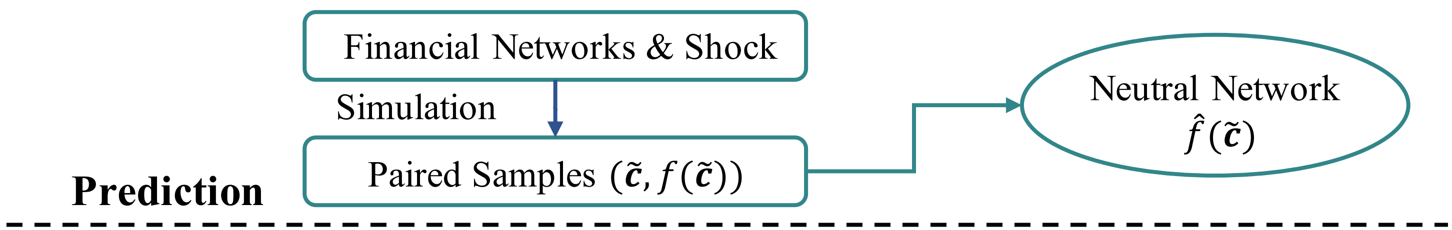
中山大学管理学院
SCHOOL OF BUSINESS
SUN YAT-SEN UNIVERSITY

$$
(P) \quad \begin{aligned} \min \quad & \boxed{f(\boldsymbol{x})} \\ \text{s.t.} \quad & g(\boldsymbol{x}) \le 0 \\ & h(\boldsymbol{x}) = 0, \end{aligned}
$$

GAP: no closed form

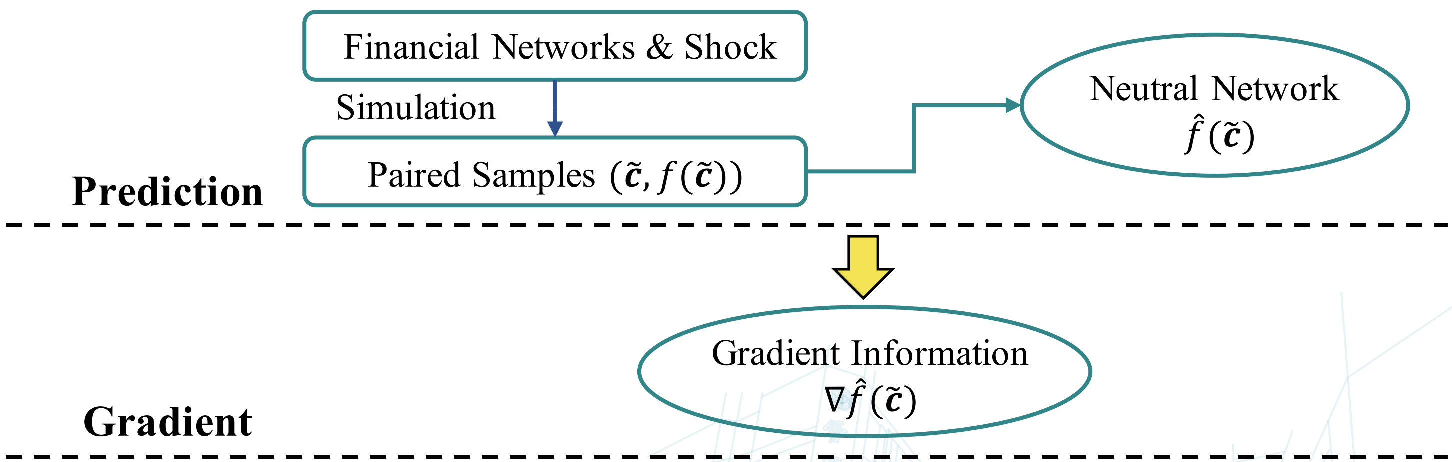
Framework: "Prediction-Gradient-Optimization" (PGO)

$$
\begin{aligned}
& & \min \quad & f(\boldsymbol{x}) \\
(P) & & \text{s.t.} \quad & g(\boldsymbol{x}) \leq 0 \\
& & & h(\boldsymbol{x}) = 0,
\end{aligned}
$$



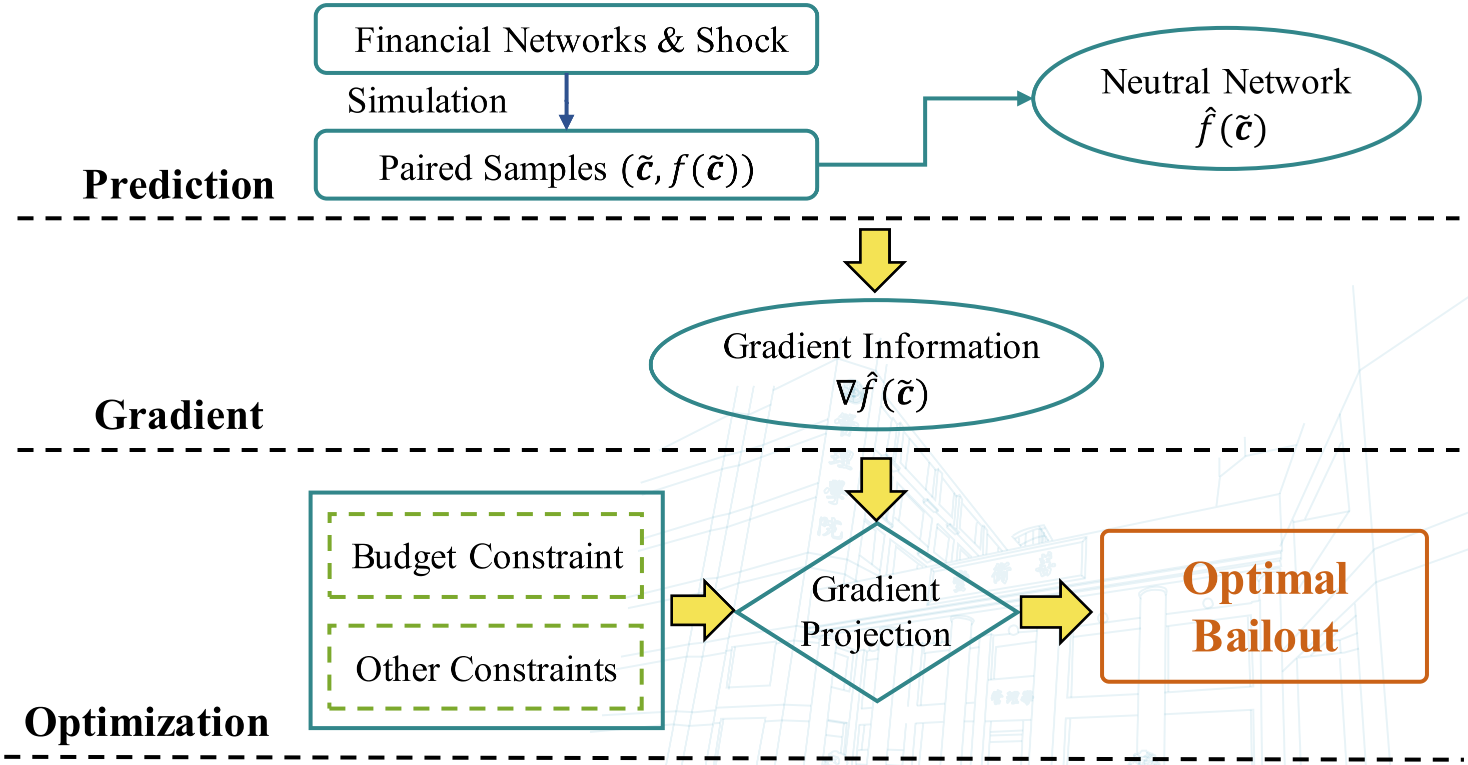
Framework: “Prediction-Gradient-Optimization” (PGO)

$$
(P) \quad \begin{aligned} \min \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g(\boldsymbol{x}) \leq 0 \\ & h(\boldsymbol{x}) = 0, \end{aligned}
$$



Framework: “Prediction-Gradient-Optimization” (PGO)

$$
(P) \quad \begin{aligned} \min \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g(\boldsymbol{x}) \le 0 \\ & h(\boldsymbol{x}) = 0, \end{aligned}
$$



Framework: “Prediction-Gradient-Optimization” (PGO)

Algorithm 1 The Prediction-Gradient-Optimization

Input: Training set of $(\boldsymbol{x}, f(\boldsymbol{x}))$ generated by black-box system; Times of training $\mathcal{T}$; Inequality constraint $g(\boldsymbol{x})$; Equality constraint $h(\boldsymbol{x})$; Initial point $\boldsymbol{x}_0$

Output: $\boldsymbol{x}^*$

```
Initialize W and b randomly;
while the times in T do
    According to the optimization of the loss function based on (x, f(x)), update W and b;
end while
function PREDICTION(W, b, x)
    According to the forward-propagation process, predict f̂(x);
    return f̂(x);
end function
function GRADIENT(W, b, x)
    According to Eq.(A1), compute ∇f̂(x);
    return ∇f̂(x);
end function
function OPTIMIZATION(f̂(x), ∇f̂(x), g(x), h(x), x_0)
    Starting from x_0, use one constrained optimization algorithm to optimize x;
    return x*.
end function
```

Framework: “Prediction-Gradient-Optimization” (PGO)

SCHOOL OF BUSINESS SUN YAT-SEN UNIVERSITY 中山大学管理学院

Algorithm 1 The Prediction-Gradient-Optimization

Input: Training set of $(\boldsymbol{x}, f(\boldsymbol{x}))$ generated

constraint $g(\boldsymbol{x})$; Equality constraint h

Output: $\boldsymbol{x}^*$

Initialize $\boldsymbol{W}$ and $\boldsymbol{b}$ randomly;

while the times in $\mathcal{T}$ **do**

According to the optimization of

end while

function PREDICTION($\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{x}$)

According to the forward-propag

return $\hat{f}(\boldsymbol{x})$;

end function

function GRADIENT($\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{x}$)

According to Eq.(A1), compute $\nabla \hat{f}(\boldsymbol{x})$;

return $\nabla \hat{f}(\boldsymbol{x})$;

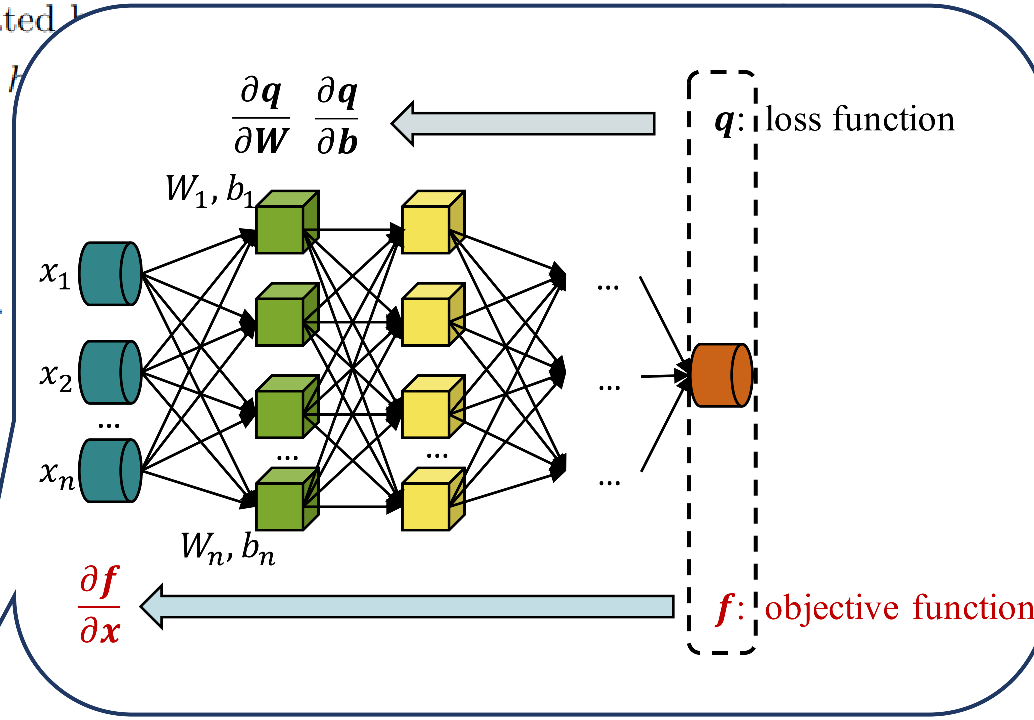
end function

function OPTIMIZATION($\hat{f}(\boldsymbol{x})$, $\nabla \hat{f}(\boldsymbol{x})$, $g(\boldsymbol{x})$, $h(\boldsymbol{x})$, $\boldsymbol{x}_0$)

Starting from $\boldsymbol{x}_0$, use one constrained optimization algorithm to optimize $\boldsymbol{x}$;

return $\boldsymbol{x}^*$.

end function



Framework: “Prediction-Gradient-Optimization” (PGO)

Algorithm 1 The Prediction-Gradient-Optimization

Input: Training set of $(\boldsymbol{x}, f(\boldsymbol{x}))$ generated by black-box system; Times of training $\mathcal{T}$; Inequality constraint $g(\boldsymbol{x})$; Equality constraint $h(\boldsymbol{x})$; Initial point $\boldsymbol{x}_0$

Output: $\boldsymbol{x}^*$

Initialize $\boldsymbol{W}$ and $\boldsymbol{b}$ randomly;

while the times in $\mathcal{T}$ **do**

    According to the optimization of the loss function based on $(\boldsymbol{x}, f(\boldsymbol{x}))$, update $\boldsymbol{W}$ and $\boldsymbol{b}$;

end while

function PREDICTION($\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{x}$)

    According to the forward-propagation process, predict $\hat{f}(\boldsymbol{x})$;

    return $\hat{f}(\boldsymbol{x})$;

end function

function GRADIENT($\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{x}$)

    According to Eq.(A1), compute $\nabla\hat{f}(\boldsymbol{x})$;

    return $\nabla\hat{f}(\boldsymbol{x})$;

end function

function OPTIMIZATION($\hat{f}(\boldsymbol{x})$, $\nabla\hat{f}(\boldsymbol{x})$, $g(\boldsymbol{x})$, $h(\boldsymbol{x})$, $\boldsymbol{x}_0$)

    Starting from $\boldsymbol{x}_0$, use one constrained optimization algorithm to optimize $\boldsymbol{x}$;

    return $\boldsymbol{x}^*$.

end function

Gradient Projection

Optimal Systemic Risk Bailout:
A PGO Approach Based on Neural Network

Overview (Optimal bailout!)

- Background
- Motivation

Bailout Measurement (What is it?)

- Definitions
- Two Cases

Framework (How to find it?)

- Prediction
- Gradient
- Optimization

Simulating Results

Discussion & Conclusion

Optimal Systemic Risk Bailout:
A PGO Approach Based on Neural Network

Overview (Optimal bailout!)

- Background
- Motivation

Bailout Measurement (What is it?)

- Definitions
- Two Cases

Framework (How to find it?)

- Prediction
- Gradient
- Optimization

Simulating Results

- Case1
- Case2(Known/Unknown Budget)

Discussion & Conclusion

Simulating Results

- **Two cases**

Table 1: A brief summary of two cases.

Case	Model	Method	Objective Function	Decision Variable
Case 1	E-N Model	LP	Pay_{all}	$\boldsymbol{l}, \tilde{\boldsymbol{c}}$
		PGO	Pay_{all}	$\tilde{\boldsymbol{c}}$
Case 2	Extended E-N Model	Heuristic	Pay_{all}	$\tilde{\boldsymbol{c}}$
		PGO	Pay_{all}	$\tilde{\boldsymbol{c}}$
		PGO	$Save_{all}$	$\tilde{\boldsymbol{c}}$

Order:

Heuristic: $\widetilde{\boldsymbol{c_1}} \Longrightarrow \begin{pmatrix} l_1 \\ p_1 \end{pmatrix} \Longrightarrow Pay_{all_1} \Longrightarrow \widetilde{\boldsymbol{c_1}} + \Delta \boldsymbol{c} \Longrightarrow \begin{pmatrix} l_2 \\ p_2 \end{pmatrix} \Longrightarrow Pay_{all_2} \overset{\cdots}{\Rightarrow} \widetilde{\boldsymbol{c}}^* \Longrightarrow \begin{cases} Pay_{all} \\ Save_{all} \end{cases}$

PGO: $\widetilde{\boldsymbol{c_o}} \Longrightarrow \hat{f}(\widetilde{\boldsymbol{c_0}}) \Longrightarrow \widetilde{\boldsymbol{c_1}} \Longrightarrow \hat{f}(\widetilde{\boldsymbol{c_1}}) \overset{\cdots}{\Rightarrow} \widetilde{\boldsymbol{c}}^* \Longrightarrow \begin{cases} Pay_{all} \\ Save_{all} \end{cases}$

Simulating Results: Case1(E-N Model)

中山大学管理学院
SCHOOL OF BUSINESS
SUN YAT-SEN UNIVERSITY

$$\begin{aligned} \max_{\tilde{\boldsymbol{c}}} \quad & Pay_{all}(\tilde{\boldsymbol{c}}) \\ \text{s.t.} \quad & \boldsymbol{1}^T\tilde{\boldsymbol{c}} \le \tau \\ & \tilde{\boldsymbol{c}} \ge \boldsymbol{0}, \end{aligned}$$

$$\begin{aligned} \max_{\boldsymbol{l},\tilde{\boldsymbol{c}}} \quad & \boldsymbol{1}^T\boldsymbol{l} \\ \text{s.t.} \quad & \tilde{\boldsymbol{c}} + \boldsymbol{c} - \boldsymbol{s} + \Pi^T\boldsymbol{l} \ge \boldsymbol{l} \\ & \boldsymbol{0} \le \boldsymbol{l} \le \bar{\boldsymbol{l}} \\ & \boldsymbol{1}^T\tilde{\boldsymbol{c}} \le \tau \\ & \tilde{\boldsymbol{c}} \ge \boldsymbol{0} \end{aligned}$$

Table 2: Results of LINPROG and PGO in *Case 1*.

n	LINPROG	PGO			
		lay=2	lay=3	lay=4	lay=5
$n = 10$	4.146	3.979 95.98%	4.059 97.89%	4.136 99.76%	4.046 97.58%
$n = 100$	28.859	27.932 96.79%	28.787 99.75%	27.785 96.281%	28.751 99.63%
$n = 1000$	312.853	311.030 99.42%	311.852 99.68%	311.553 99.58%	311.936 99.71%

Simulating Results: Case2(Extended E-N Model)

- ## Case2.1 Known Budget

Heuristic
(Objective Function: Pay_{all})

$$\begin{aligned} \max_{\tilde{\boldsymbol{c}}_{\mathcal{D}^1}} \quad & Pay_{all}(\tilde{\boldsymbol{c}}_{\mathcal{D}^1}) = \mathbf{1}^T \boldsymbol{l}^*_{\mathcal{D}^1}(\tilde{\boldsymbol{c}}_{\mathcal{D}^1}) \\ \text{s.t.} \quad & \mathbf{1}^T \tilde{\boldsymbol{c}}_{\mathcal{D}^1} \leq \tau \\ & \tilde{\boldsymbol{c}}_{\mathcal{D}^1} \geq \mathbf{0}, \end{aligned}$$

PGO
(Objective Function: Pay_{all})

$$\begin{aligned} \max_{\tilde{\boldsymbol{c}}} \quad & Pay_{all}(\tilde{\boldsymbol{c}}) \\ \text{s.t.} \quad & \mathbf{1}^T \tilde{\boldsymbol{c}} \leq \tau \\ & \tilde{\boldsymbol{c}} \geq \mathbf{0}, \end{aligned}$$

PGO
(Objective Function: $Save_{all}$)

$$\begin{aligned} \max_{\tilde{\boldsymbol{c}}} \quad & Save_{all}(\tilde{\boldsymbol{c}}) \\ \text{s.t.} \quad & \mathbf{1}^T \tilde{\boldsymbol{c}} \leq \tau \\ & \tilde{\boldsymbol{c}} \geq \mathbf{0} \end{aligned}$$

Simulating Results: Case2.1 Known Budget

中山大学管理学院
SCHOOL OF BUSINESS
SUN YAT-SEN UNIVERSITY

$$Budget = \min\{budget, \tau_{\max}\}$$

Table 3: Results of the Heuristic and the PGO in *Case 2* with the known budget.

Different Approach	**Result**	$n = 10$	$n = 100$	$n = 1000$
Initial State	Pay_{all}	5.487	55.308	522.313
	$Budget$	0.076	3.080	54.237
Heuristic (Objective Function: Pay_{all})	Pay_{all}	5.600	59.765	590.471
	$Save_{all}$	0.644	16.930	261.939
	$Ratio$	8.472	5.497	4.830
PGO (Objective Function: Pay_{all})	Pay_{all}	5.600	59.477	590.471
	$Save_{all}$	0.719	15.876	303.902
	$Ratio$	9.458	5.155	5.603
PGO (Objective Function: $Save_{all}$)	Pay_{all}	5.600	59.420	590.471
	$Save_{all}$	0.719	16.525	307.152
	$Ratio$	9.458	5.365	5.663

Simulating Results: Case2.2 Unknown Budget

中山大学管理学院
SCHOOL OF BUSINESS
SUN YAT-SEN UNIVERSITY

$$Budget_{\max} = \tau_{\max}$$

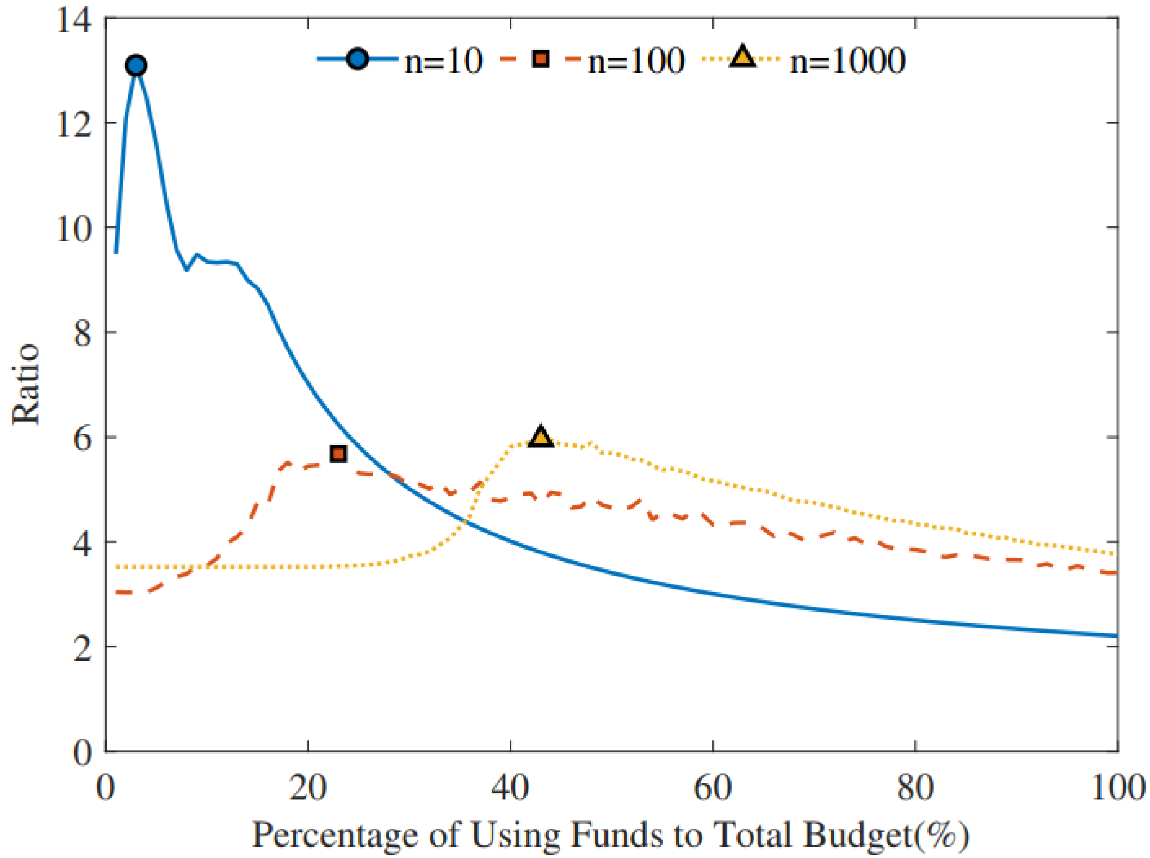
Table 4: Results of the Heuristic and the PGO in *Case 2* with the unknown budget.

	$n=10$			$n=100$			$n=1000$		
	Sample	**PGO**	**Ratio**	**Sample**	**PGO**	**Ratio**	**Sample**	**PGO**	**Ratio**
$0.1\tau_{max}$	0.851	0.875	9.608	3.237	3.303	3.619	31.801	32.095	3.553
$0.2\tau_{max}$	1.278	1.278	7.021	9.944	10.018	5.489	63.603	63.931	3.539
$0.3\tau_{max}$	1.369	1.369	5.014	13.872	14.172	5.176	100.962	100.227	3.698
$0.4\tau_{max}$	1.460	1.460	4.010	17.646	19.079	5.227	210.075	212.494	5.881
$0.5\tau_{max}$	1.551	1.551	3.408	21.202	20.895	4.579	257.300	257.760	5.707
$0.6\tau_{max}$	1.642	1.642	3.007	23.700	23.354	4.265	280.043	281.901	5.201
$0.7\tau_{max}$	1.733	1.733	2.720	26.024	25.808	4.040	299.973	300.155	4.747
$0.8\tau_{max}$	1.824	1.824	2.505	28.107	27.931	3.826	313.486	313.895	4.344
$0.9\tau_{max}$	1.915	1.915	2.338	30.054	29.535	3.596	326.595	327.472	4.028
τ_{max}	2.006	2.006	2.204	31.145	30.940	3.390	340.400	340.812	3.773

Simulating Results: Case2.2 Unknown Budget

The Highest Ratio

$$Ratio = \frac{Save_{all}}{\tau}$$



Simulating Results: Case2.3 More Constraints

中山大学管理学院
SCHOOL OF BUSINESS
SUN YAT-SEN UNIVERSITY

- $Budget = 0.1\tau_{\max}$
- The number of bailout funds for each bank $\leq \frac{\xi}{n_s}\tau_{\max}$

ξ : a parameter(=1.5)
n_s : the number of banks bailed out

Table 5: Results of the random generation and the PGO in *Case 2* with $0.1\tau_{max}$ budget and more constraints.

	$n = 10$		$n = 100$		$n = 1000$	
	Sample	**PGO**	**Sample**	**PGO**	**Sample**	**PGO**
$Save_{all}$	0.260	0.327	5.037	6.209	31.793	32.094
Time(s)	0.66	0.87	3.18	3.27	457.97	283.22

Optimal Systemic Risk Bailout:
A PGO Approach Based on Neural Network

Overview (Optimal bailout!)

- Background
- Motivation

Bailout Measurement (What is it?)

- Definitions
- Two Cases

Framework (How to find it?)

- Prediction
- Gradient
- Optimization

Simulating Results

- Case1
- Case2(Known/Unknown Budget)

Discussion & Conclusion

Optimal Systemic Risk Bailout:
A PGO Approach Based on Neural Network

Overview (Optimal bailout!)

- Background
- Motivation

Bailout Measurement (What is it?)

- Definitions
- Two Cases

Framework (How to find it?)

- Prediction
- Gradient
- Optimization

Simulating Results

- Case1
- Case2(Known/Unknown Budget)

Discussion & Conclusion

Conclusion

E-N model + Multiple asset Banking System

Shock → Fire Sale Liability - - - - → Systemic Risk

$$(P)\quad \begin{aligned} \max \quad & \boxed{f(\boldsymbol{x})} \\ \text{s.t.} \quad & g(\boldsymbol{x}) \leq 0 \\ & h(\boldsymbol{x}) = 0, \end{aligned}$$

Neural Network ⇒ Back-Propagation

$$(P)\quad \begin{aligned} \max \quad & \hat{f}(\boldsymbol{x}) \\ \text{s.t.} \quad & g(\boldsymbol{x}) \leq 0 \\ & h(\boldsymbol{x}) = 0, \end{aligned}$$

$\nabla \hat{f}(\boldsymbol{x})$ ⇒ **Optimal Systemic Risk Bailout**

Discussion

中山大学管理学院
SCHOOL OF BUSINESS
SUN YAT-SEN UNIVERSITY

- Generalization
- NP-hard & Combinatorial Optimization

Model	**E-N model**	**E-N +Default costs**	**E-N + Market value/Cross hold**	**E-N +Multiple illiquid assets**
Source of Model	Eisenberg & Noe (2001)	Rogers & Veraart (2013)	……	Feinstein (2017) Ma et.al(2021)
Property	Linear Programming	Non-deterministic Polynomial hard (NP hard)	……	Objective function **with no closed form**
Research	Pokutta et.al(2011)	Jackson & Pernoud (2020): A simple algorithm (by order)	Demange and Gabrielle(2018): A threat index (by index)	Ma et.al(2021): A heuristic algorithm

Thanks for your listening!