DCAPS workshop
@ George Washington University
December 5, 2019

Security and Privacy for Distributed Optimization and Learning

Nitin Vaidya
Georgetown University

Secret to happiness is to
lower your expectations to the point
where they're already met

- Hobbes (paraphrased)

Goals

- Problem formulation
- Intuition

Rendezvous


Baseball Fields
GEORGETOWN
Georgetown Day School Lower/Middle School
Georgetown University McDonough School of...
Georgetown University
Foundry Branch Valley Park
Georgetown Cupcake
Filomena Ristorante
Baked & Wired
Trader Joe's
WEST END
M St NW
AMC Georgetown 14
Georgetown Waterfront Park
Three Sisters
Potomac River
Alexandria Aqueduct
Canal Rd NW
George Washington Memorial Pkwy
George Washington University Hospital
Fort CF Smith Park
Fort Bennett Park and Palisades Trail
N Spout Run Pkwy
Key Bridge Marriott
The Mole
Watergate Complex Real Estate
Gateway Park
NORTH HIGHLAND
Custis Memorial Pkwy (Toll road)
McDonald's
John F. Kennedy Center for the Performing Arts
Pho 75
COLONIAL VILLAGE
ROSSLYN
Easby Point
Tupelo Honey
Wilson Blvd
Theodore Roosevelt Bridge
Albert Einstein Memorial
US Marine
Rock Creek and Potomac Pkwy
VIRGINIA
DISTRICT OF COLUMBIA

Rendezvous

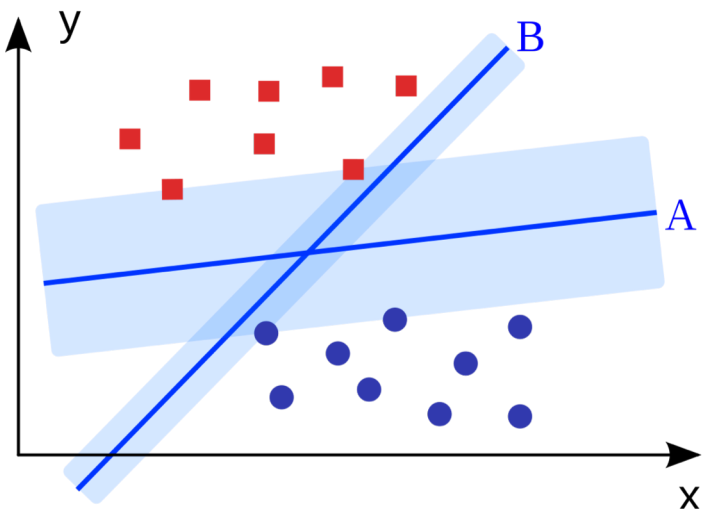

$f_1(x)$
$f_2(x)$
$f_3(x)$
X
Georgetown University
Georgetown Day School Lower/Middle School
Baseball Fields
Georgetown University McDonough School of...
GEORGETOWN
WEST END
Branch Valley Park
Three Sisters
Potomac River
Canal Rd NW
Alexandria Aqueduct
Georgetown Waterfront Park
M St NW
Trader Joe's
Baked & Wired
AMC Georgetown 14
George Washington University Hospital
George Washington Memorial Pkwy
Fort CF Smith Park
Fort Bennett Park and Palisades Trail
N Spout Run Pkwy
Key Bridge Marriott
Gateway Park
NORTH HIGHLAND
Custis Memorial Pkwy (Toll road)
McDonald's
The Mole
Watergate Complex Real Estate
John F. Kennedy Center for the Performing Arts
Easby Point
COLONIAL VILLAGE
Pho 75
Tupelo Honey
Wilson Blvd
Key Blvd
Theodore Roosevelt Bridge
Albert Einstein Memorial
Rock Creek and Potomac Pkwy
Dumbarton St NW
P St NW
O St NW
Q St NW
N St NW
F St NW
34th St
23rd St NW
DISTRICT OF COLUMBIA
VIRGINIA

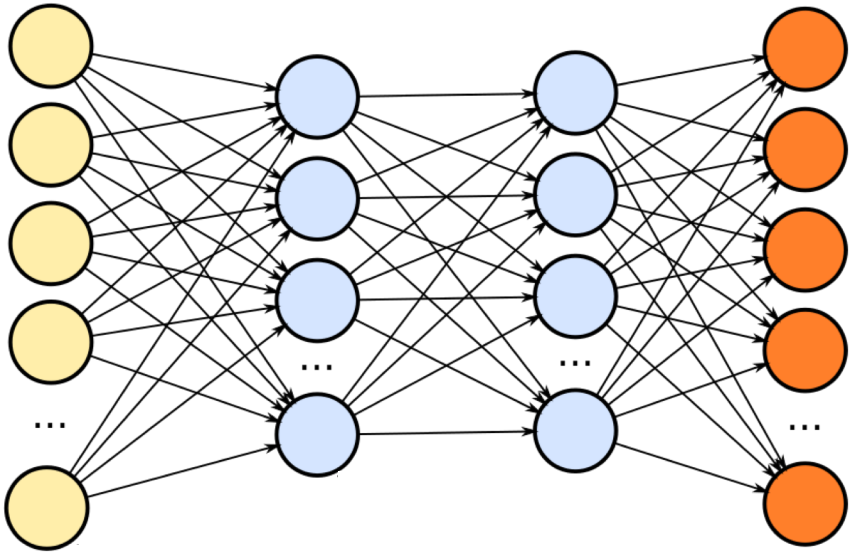

argmin $\sum f_i(x)$

Machine Learning

- Data is distributed across different agents

- Data is distributed across different agents ➔ Collaborate to learn



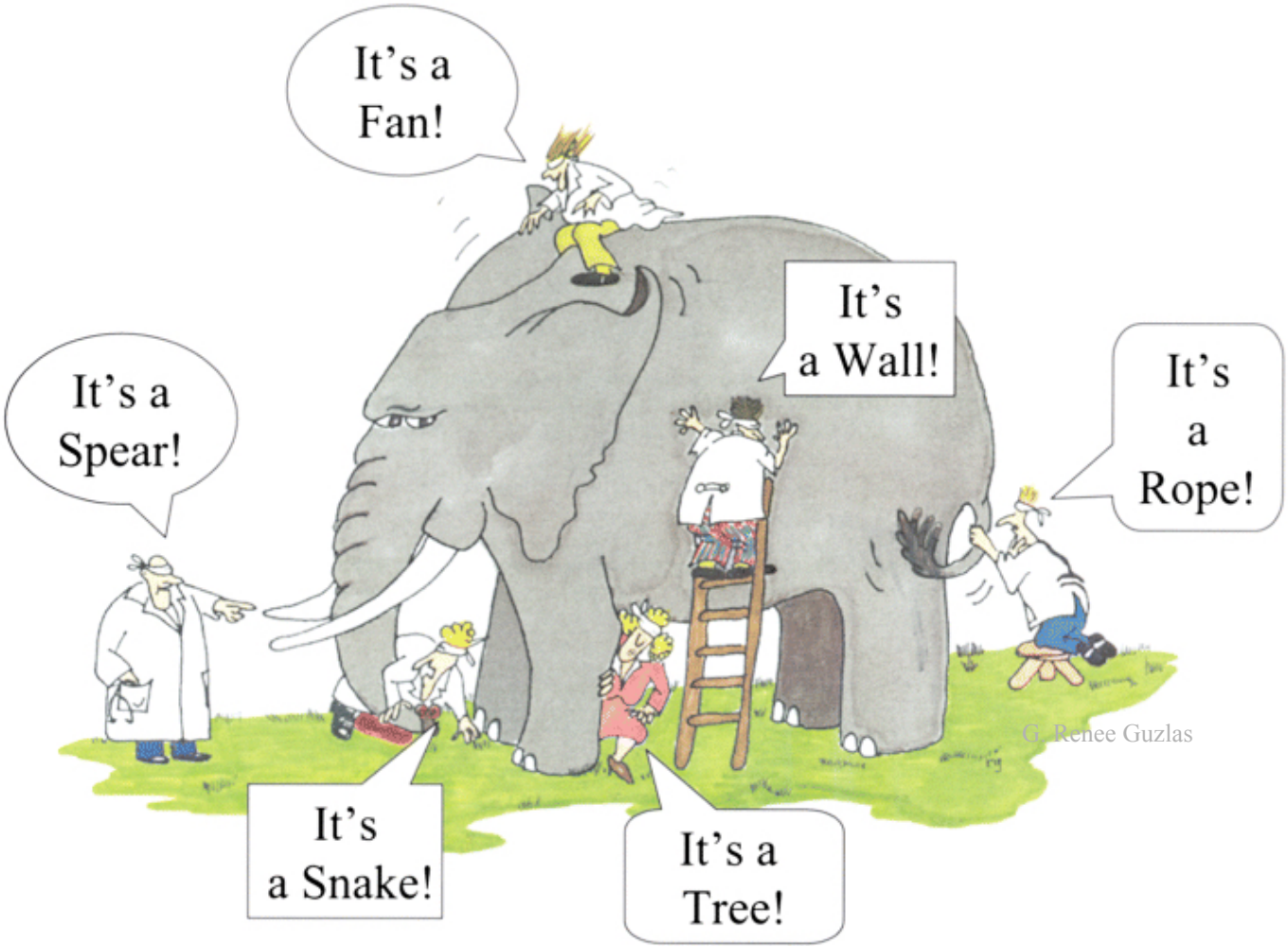


Machine Learning

Minimize global loss

$$\sum f_i(x)$$

Classification



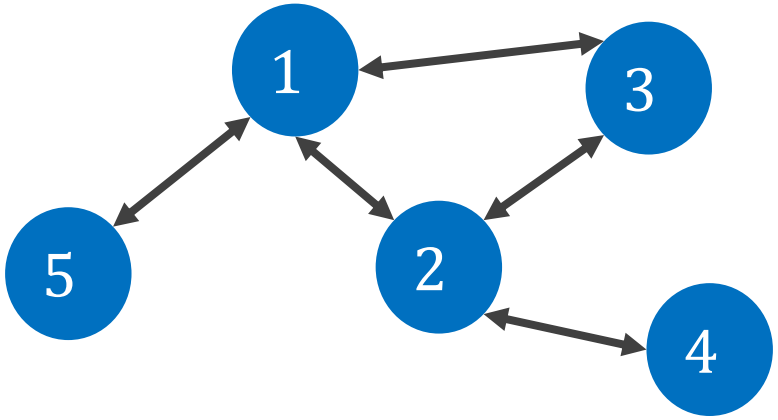

argmin $\sum f_i(x)$

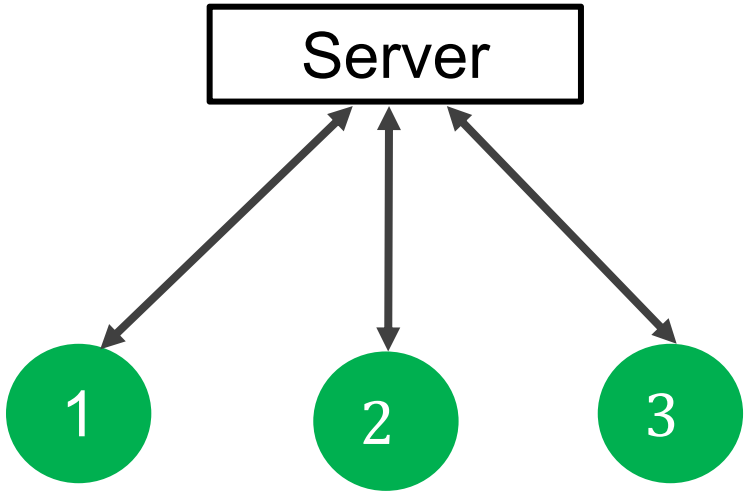
Distributed Optimization

- Each agent i knows own cost function $f_i(x)$
- Need to cooperate to minimize $\sum f_i(x)$

➔ Distributed algorithms

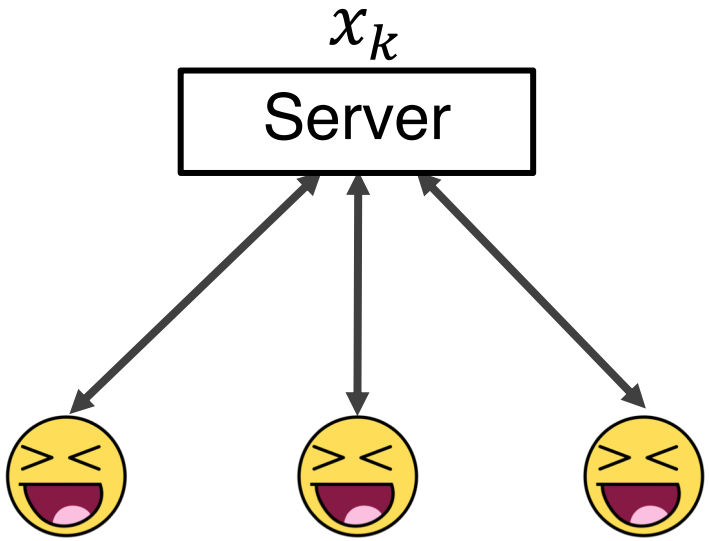
Architectures





Parameter Server

- Server maintains estimate x_k

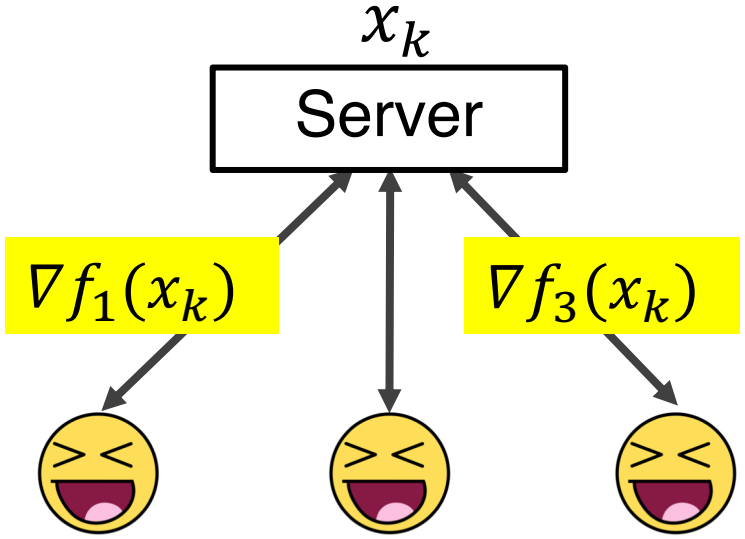


Parameter Server

- Server maintains estimate x_k

In each iteration

- Agent i
  - Receives x_k from server

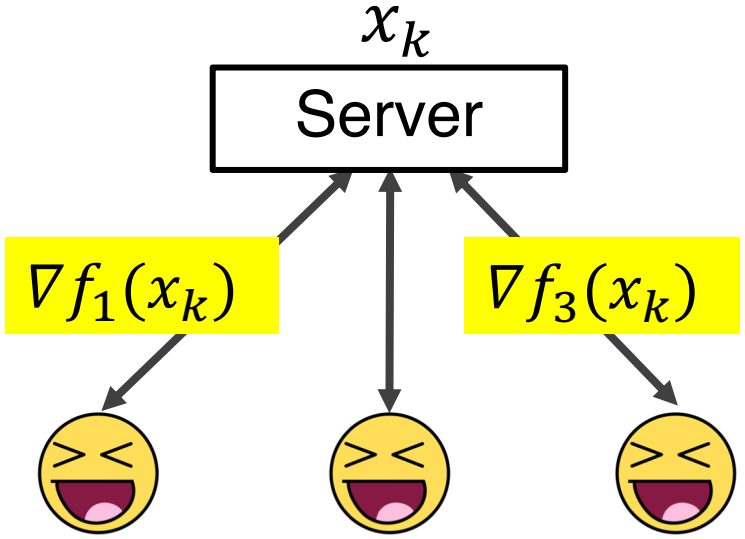


Parameter Server

- Server maintains estimate x_k

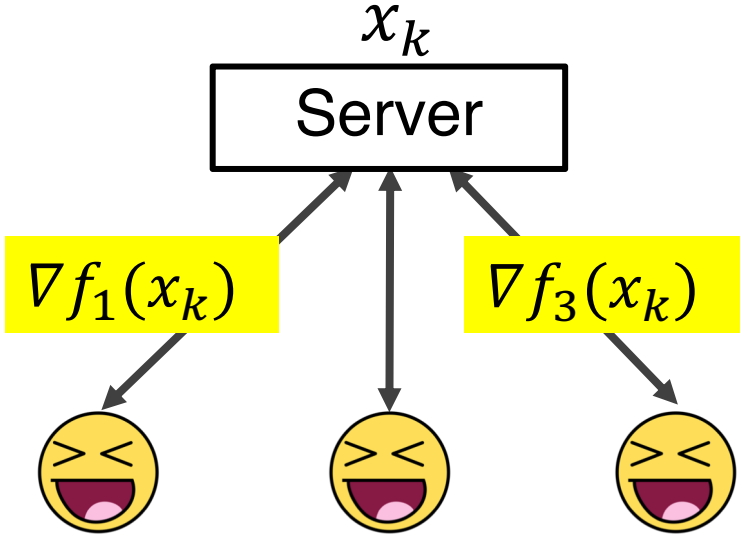
In each iteration

- Agent i
  - Receives x_k from server
  - Uploads gradient $\nabla f_i(x_k)$



Parameter Server

- Server maintains estimate x_k



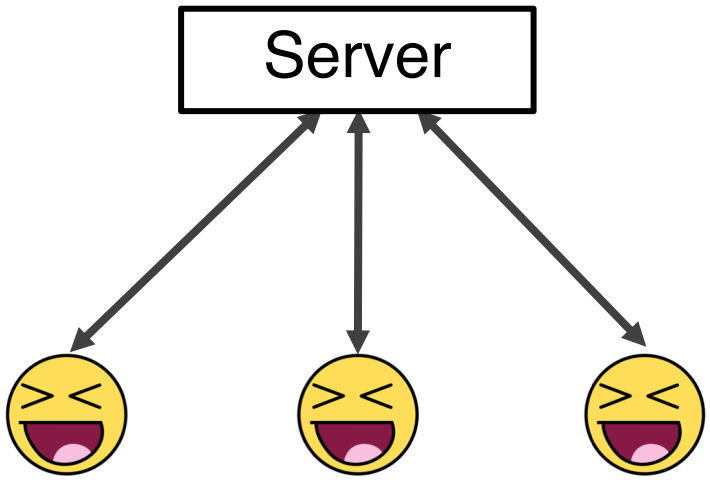
In each iteration

- Agent i
  - Receives x_k from server
  - Uploads gradient $\nabla f_i(x_k)$

- Server updates estimate

$$x_{k+1} \longleftarrow x_k - \alpha \sum \nabla f_i(x_k)$$

Many Variations



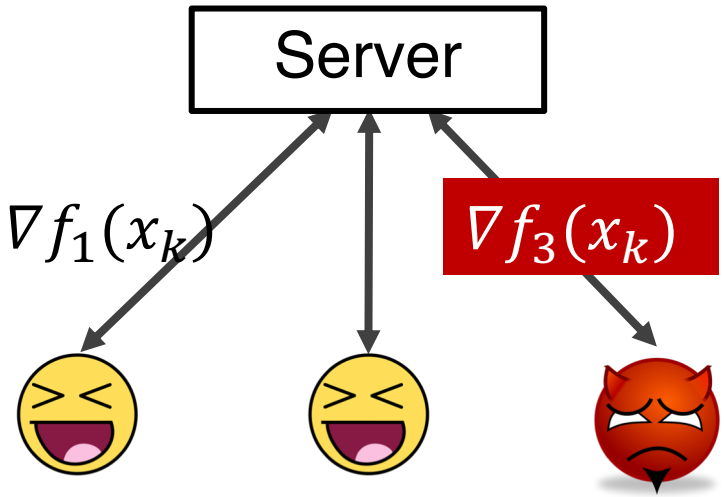
… stochastic optimization
… asynchronous
… gradient compression
… acceleration
… shared memory

Challenges

Challenges

- Fault-tolerant distributed optimization

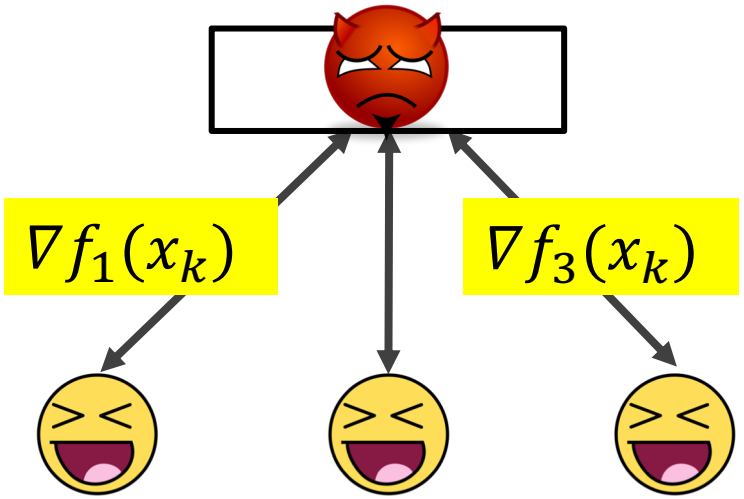
$$f_1(x) + f_2(x) + f_3(x)$$



How to optimize if agents inject bogus information?

Challenges

- Privacy-preserving distributed optimization



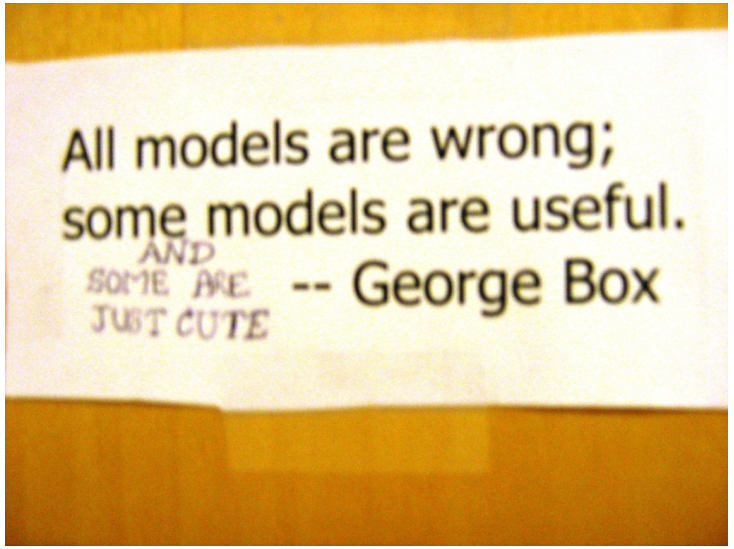
How to collaborate without revealing own cost function?

Fault-Tolerant Optimization

2015 …

Byzantine Fault Model

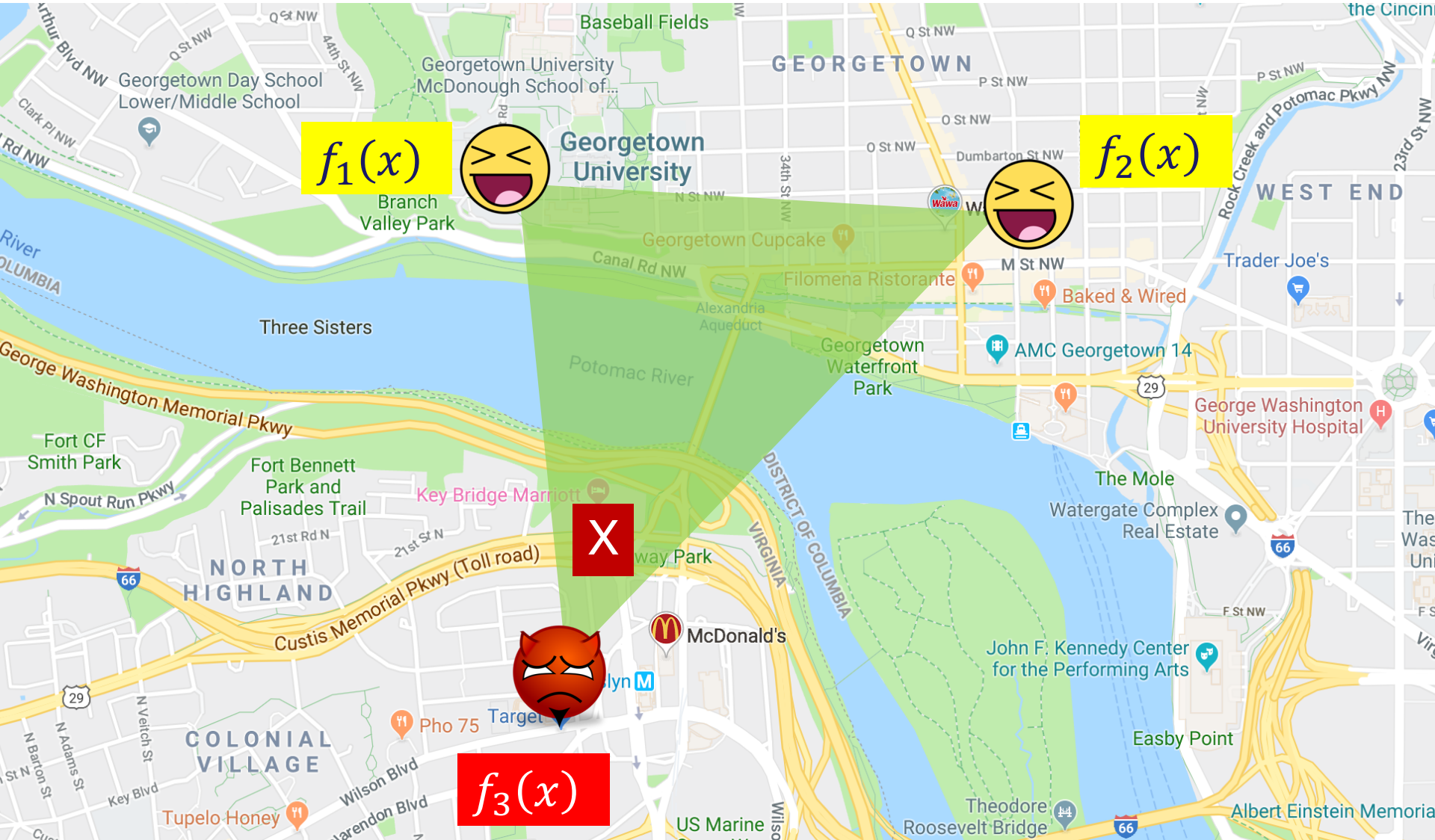
- No constraint
  on misbehavior of *faulty* agents



Rendezvous


$f_1(x)$
$f_2(x)$
$f_3(x)$
X
Georgetown University
Georgetown Day School Lower/Middle School
Baseball Fields
Georgetown University McDonough School of...
GEORGETOWN
WEST END
Branch Valley Park
Three Sisters
Potomac River
Canal Rd NW
Alexandria Aqueduct
Georgetown Waterfront Park
M St NW
Trader Joe's
Baked & Wired
AMC Georgetown 14
George Washington University Hospital
George Washington Memorial Pkwy
Fort CF Smith Park
Fort Bennett Park and Palisades Trail
N Spout Run Pkwy
Key Bridge Marriott
Gateway Park
NORTH HIGHLAND
Custis Memorial Pkwy (Toll road)
McDonald's
The Mole
Watergate Complex Real Estate
John F. Kennedy Center for the Performing Arts
Easby Point
COLONIAL VILLAGE
Pho 75
Tupelo Honey
Wilson Blvd
Key Blvd
Theodore Roosevelt Bridge
Albert Einstein Memorial
Rock Creek and Potomac Pkwy
Dumbarton St NW
P St NW
O St NW
Q St NW
N St NW
F St NW
34th St
23rd St NW
DISTRICT OF COLUMBIA
VIRGINIA

Rendezvous


$f_1(x)$
$f_2(x)$
$f_3(x)$
X
Georgetown University
Georgetown Day School Lower/Middle School
Georgetown University McDonough School of...
Baseball Fields
GEORGETOWN
WEST END
Branch Valley Park
Three Sisters
Potomac River
Georgetown Cupcake
Filomena Ristorante
Baked & Wired
Trader Joe's
AMC Georgetown 14
Georgetown Waterfront Park
George Washington Memorial Pkwy
Fort CF Smith Park
Fort Bennett Park and Palisades Trail
Key Bridge Marriott
NORTH HIGHLAND
Custis Memorial Pkwy (Toll road)
McDonald's
The Mole
Watergate Complex Real Estate
George Washington University Hospital
John F. Kennedy Center for the Performing Arts
Easby Point
COLONIAL VILLAGE
Pho 75
Tupelo Honey
Theodore Roosevelt Bridge
Albert Einstein Memorial
M St NW
Q St NW
P St NW
O St NW
Dumbarton St NW
N St NW
Canal Rd NW
Alexandria Aqueduct
Rock Creek and Potomac Pkwy
23rd St NW
34th St NW
44th St NW
Clark Pl NW
Arthur Blvd NW
N Spout Run Pkwy
21st Rd N
21st St N
Wilson Blvd
Key Blvd
N Veitch St
N Adams St
N Barton St
F St NW
DISTRICT OF COLUMBIA
VIRGINIA
66
29

Machine Learning

Faulty agent can adversely affect model parameters

Minimize global loss

$$\sum f_i(x)$$

Fault-Tolerance

- What should be the objective of fault-tolerant optimization?

Fault-Tolerance

- What should be the objective of fault-tolerant optimization?

- Optimize over only good agents … set *G*

Fault-Tolerance

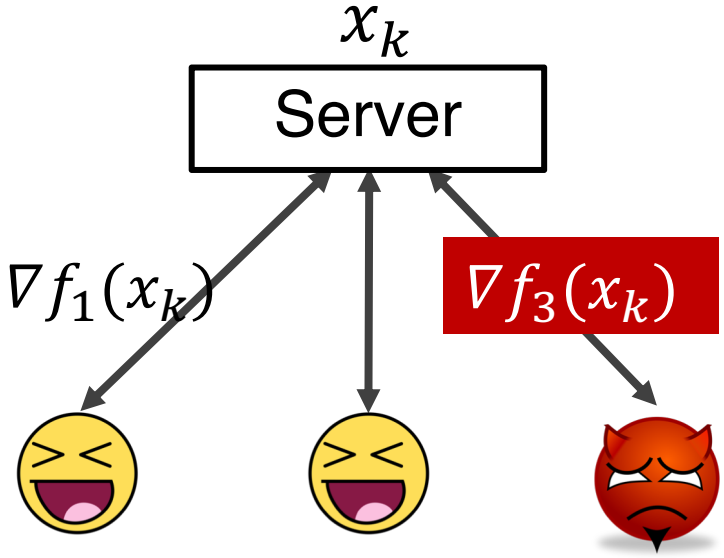
- What should be the objective of fault-tolerant optimization?

- Optimize over only good agents … set *G*

$$\text{argmin} \sum_{i \in G} f_i(x)$$

Parameter Server

- Server maintains estimate x_k



In each iteration

- Agent i
  - Downloads x_k from server
  - Uploads gradient $\nabla f_i(x_k)$

- Server updates estimate

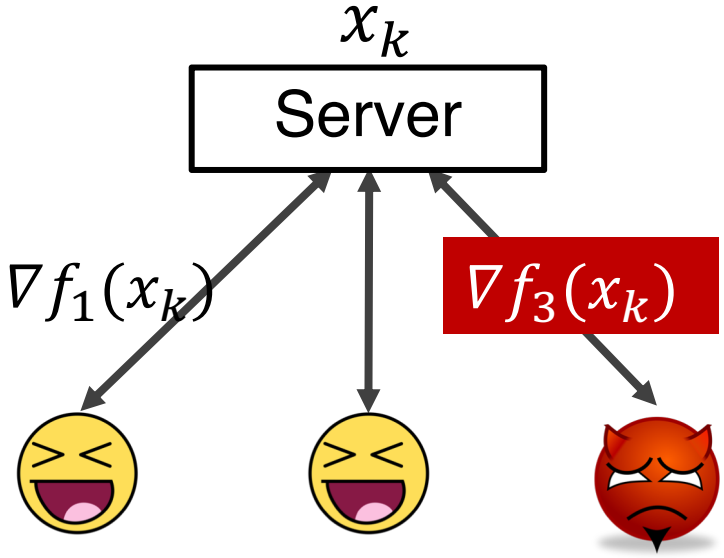
$$x_{k+1} \longleftarrow x_k - \alpha \sum \nabla f_i(x_k)$$

Parameter Server

- Server maintains estimate x_k

<u>In each iteration</u>

- Agent i
  - Downloads x_k from server
  - Uploads gradient $\nabla f_i(x_k)$



- Server updates estimate

$$x_{k+1} \longleftarrow x_k - \alpha \, \text{Filtered—Gradient}$$

$$\operatorname{argmin} \sum_{i \in G} f_i(x)$$

Is this achievable?

$$\operatorname{argmin} \sum_{i \in G} f_i(x)$$

Is this achievable?

It Depends

$$\operatorname{argmin} \sum_{i \in G} f_i(x)$$

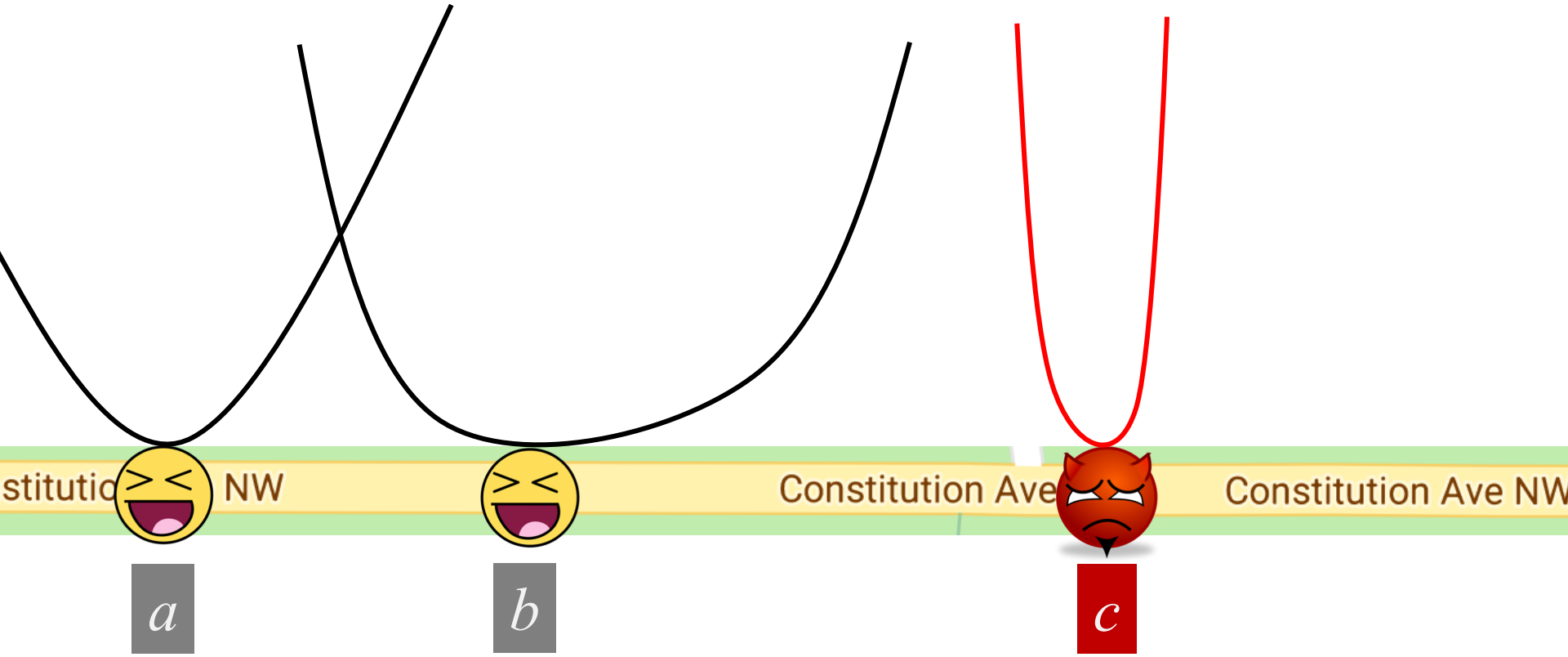
Is this achievable?

It Depends

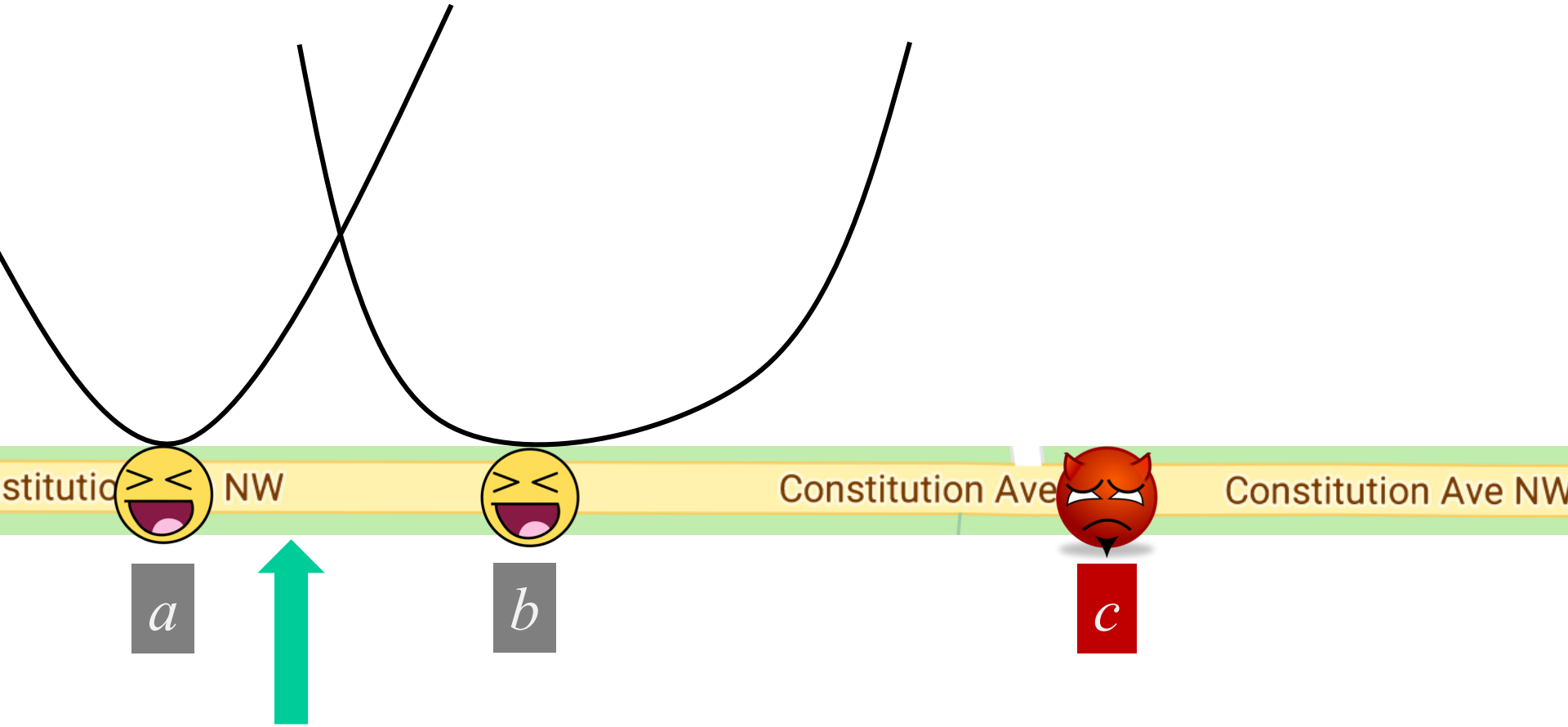
Independent functions

"Enough" redundancy

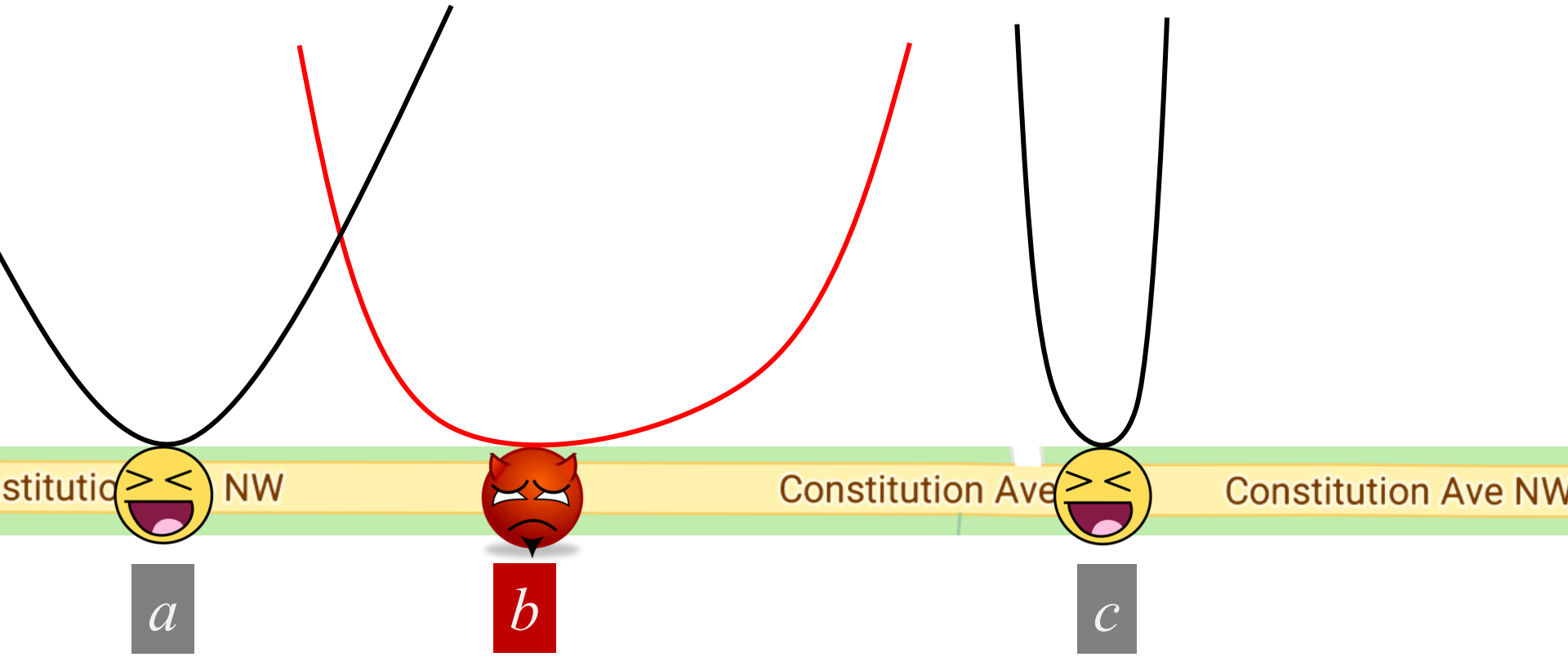
Independent Functions



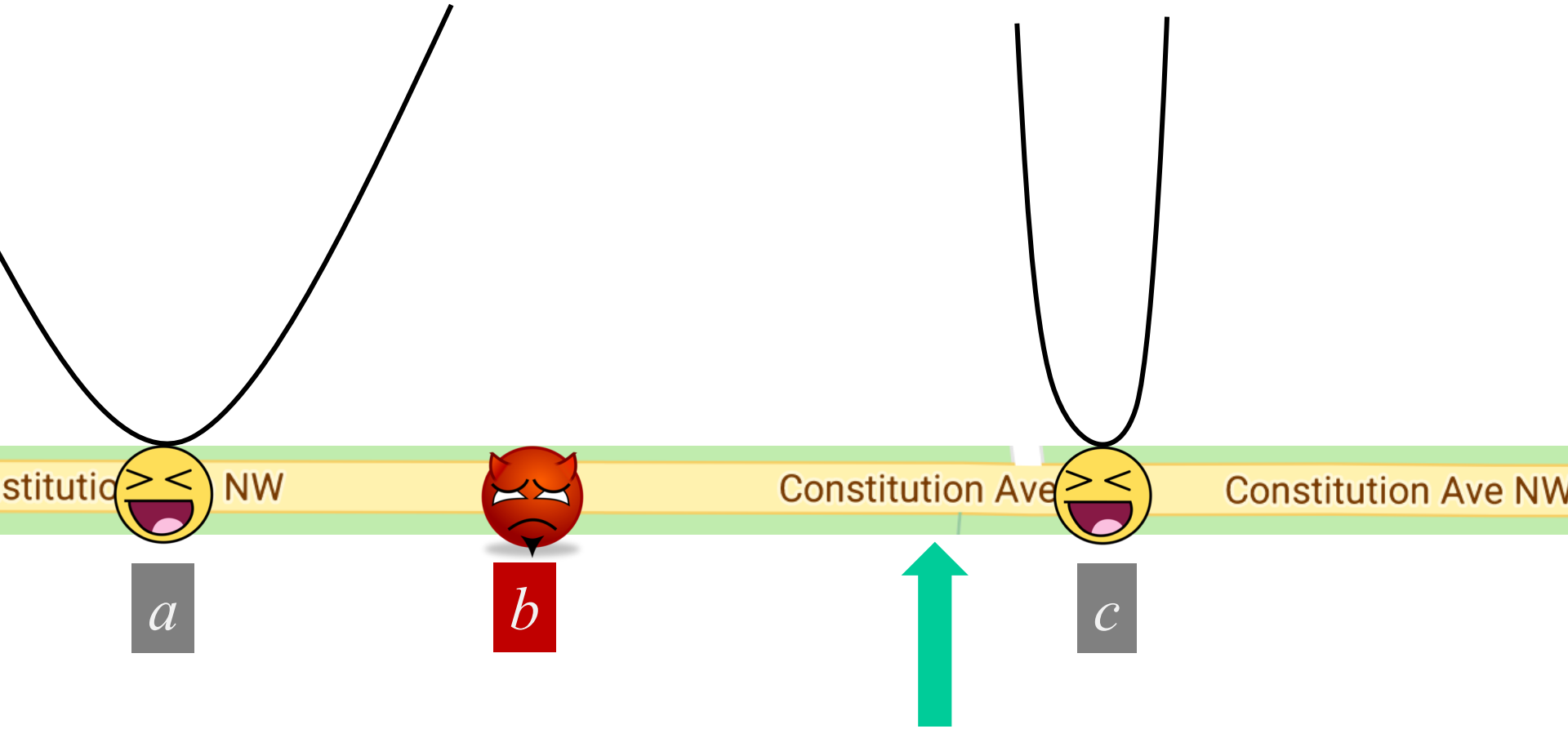
Independent Functions



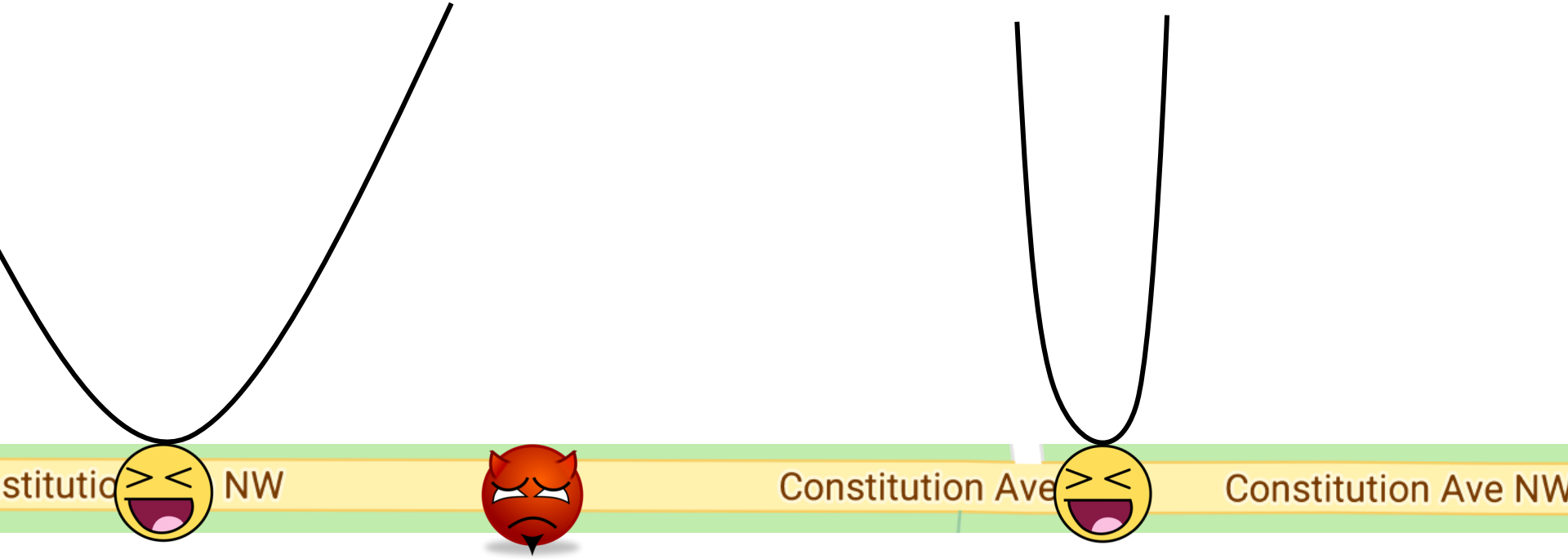
Independent Functions



Independent Functions



Independent Functions



Provably impossible to compute

$$\operatorname{argmin} \sum_{i \in G} f_i(x)$$

$$\operatorname{argmin} \sum_{i \in G} f_i(x)$$

Is this achievable?

Independent functions

"Enough" redundancy

Approximate

Exact

$$\operatorname{argmin} \sum_{i \in G} f_i(x)$$

Is this achievable?

Independent functions

"Enough" redundancy

Approximate

Exact

An Example of Redundancy

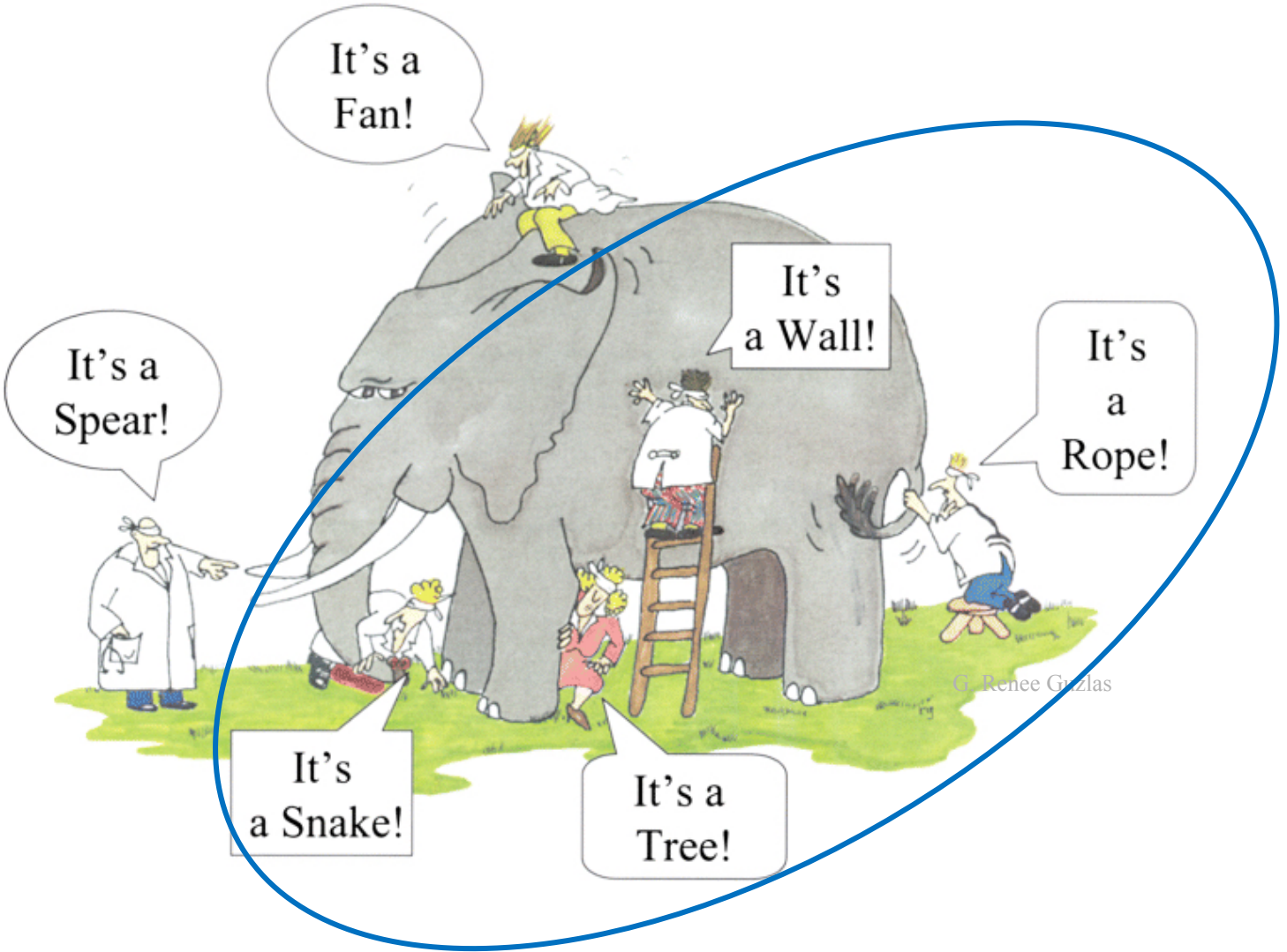
n agents
t bad agents

- Aggregate cost of ANY $n - 2t$ agents has

  argmin identical to desired $\operatorname{argmin} \sum_{i \in G} f_i(x)$

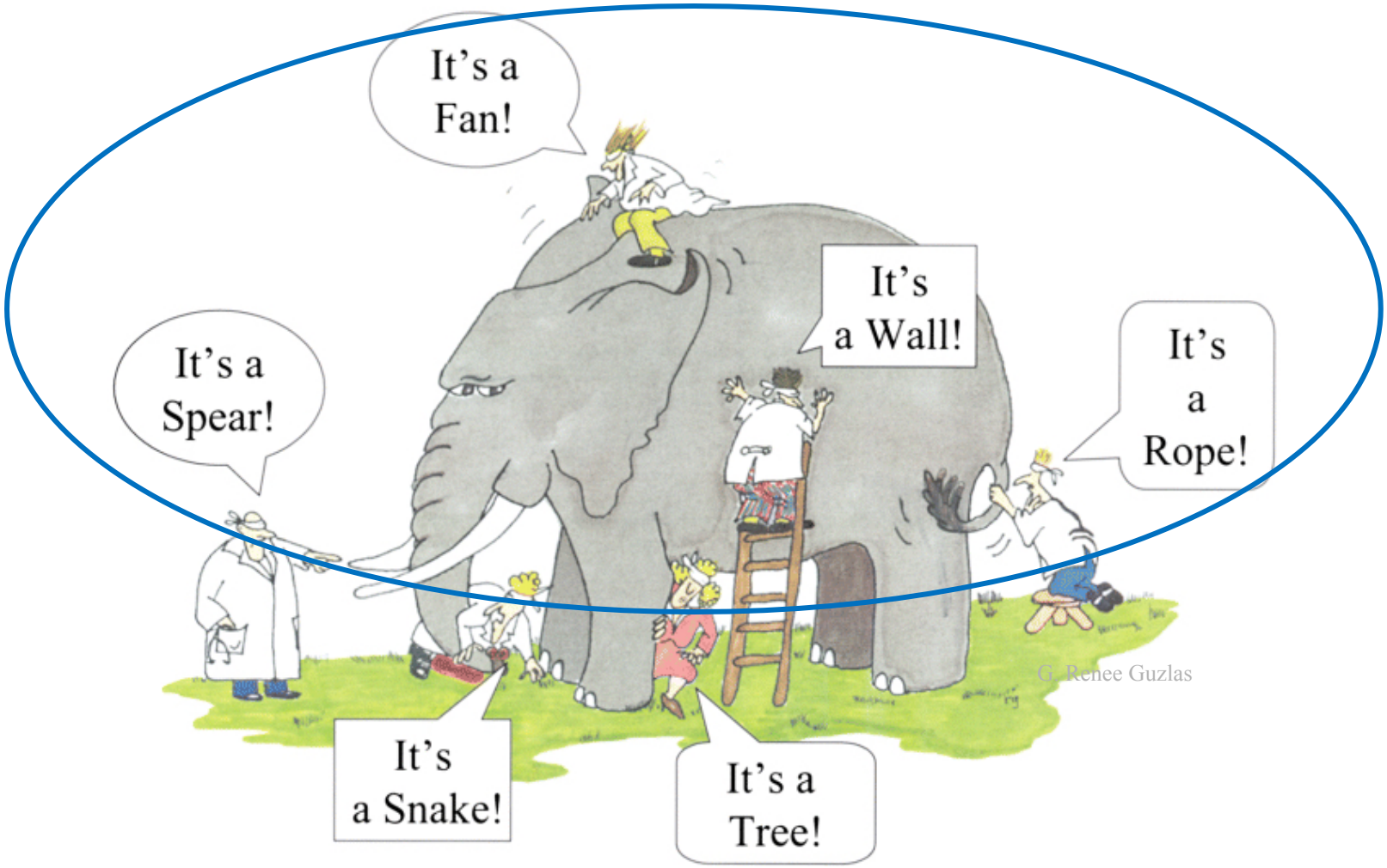
$n = 6$ (number of agents)

$t = 1$ (faulty agents)



$n = 6$ (number of agents)

$t = 1$ (faulty agents)

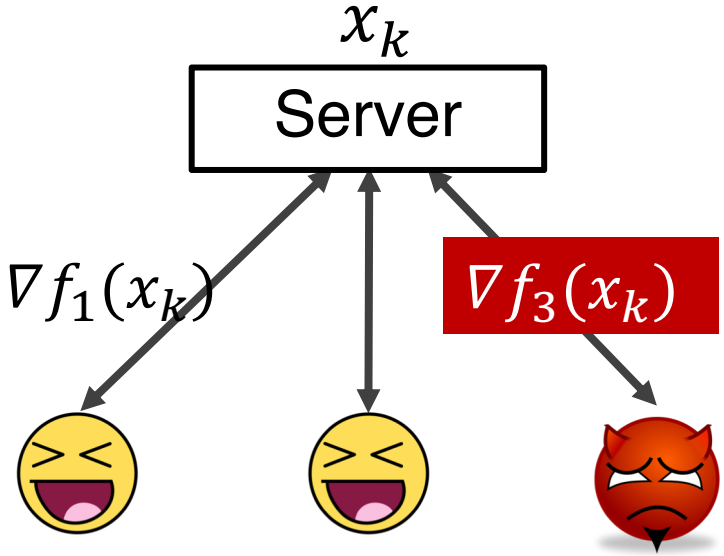


Parameter Server

- Server maintains estimate x_k

<u>In each iteration</u>

- Agent i
  - Downloads x_k from server
  - Uploads gradient $\nabla f_i(x_k)$



- Server updates estimate

$$x_{k+1} \longleftarrow x_k - \alpha \, \text{Filtered—Gradient}$$

Norm Filter

- Clip the largest t norms to equal $t + 1^{\text{th}}$ norm

$$|\nabla f_1(x_k)| = 1$$

$$|\nabla f_2(x_k)| = 3$$

$$|\nabla f_3(x_k)| = 2$$

Norm Filter

- Clip the largest t norms to equal $t + 1^{\text{th}}$ norm

$$|\nabla f_1(x_k)| = 1$$

$$|\nabla f_2(x_k)| = 3$$

$$|\nabla f_3(x_k)| = 2$$

Filtered gradient = $\nabla f_1(x_k) + \frac{2}{3}\nabla f_2(x_k) + \nabla f_3(x_k)$

Norm Filter

- Clip the largest t norms to equal $t+1^{\text{th}}$ norm

$$|\nabla f_1(x_k)| = 1$$

$$|\nabla f_2(x_k)| = 3$$

$$|\nabla f_3(x_k)| = 2$$

Filtered gradient = $\nabla f_1(x_k) + \frac{2}{3}\nabla f_2(x_k) + \nabla f_3(x_k)$

Exact optimum computed despite faulty agents

Another Example of Redundancy

Another Example of Redundancy

- Machine learning
- Agents draw samples from identical data distribution

- Filter on stochastic gradients

$$\operatorname{argmin} \sum_{i \in G} f_i(x)$$

Is this achievable?

Independent functions

"Enough" Redundancy

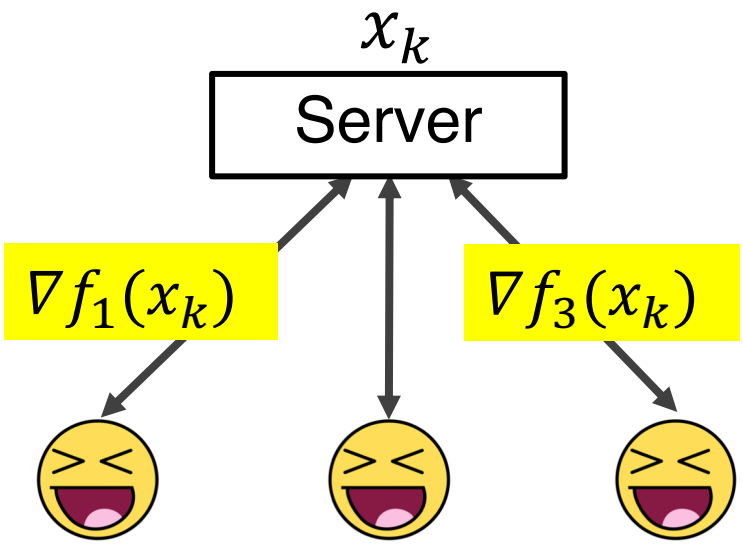
Approximation

Exact

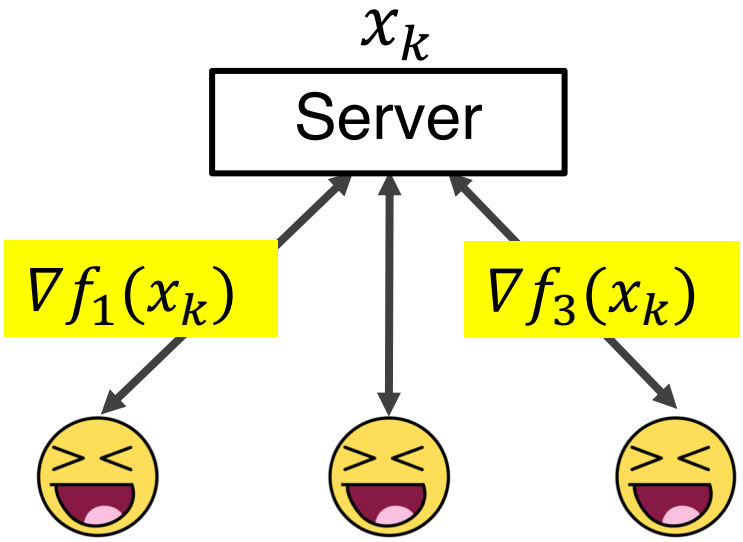
Privacy-Preserving Optimization

2016 …

Communication Leaks Information



Communication Leaks Information



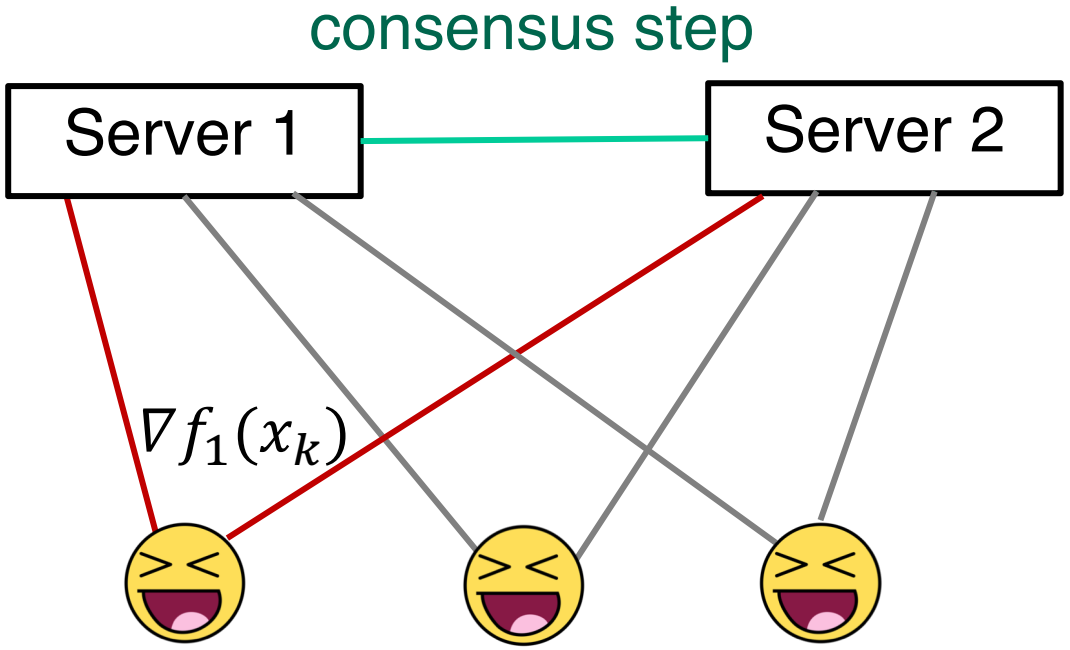
Server can use gradients to infer polynomial cost functions (up to a constant)

Our Approach

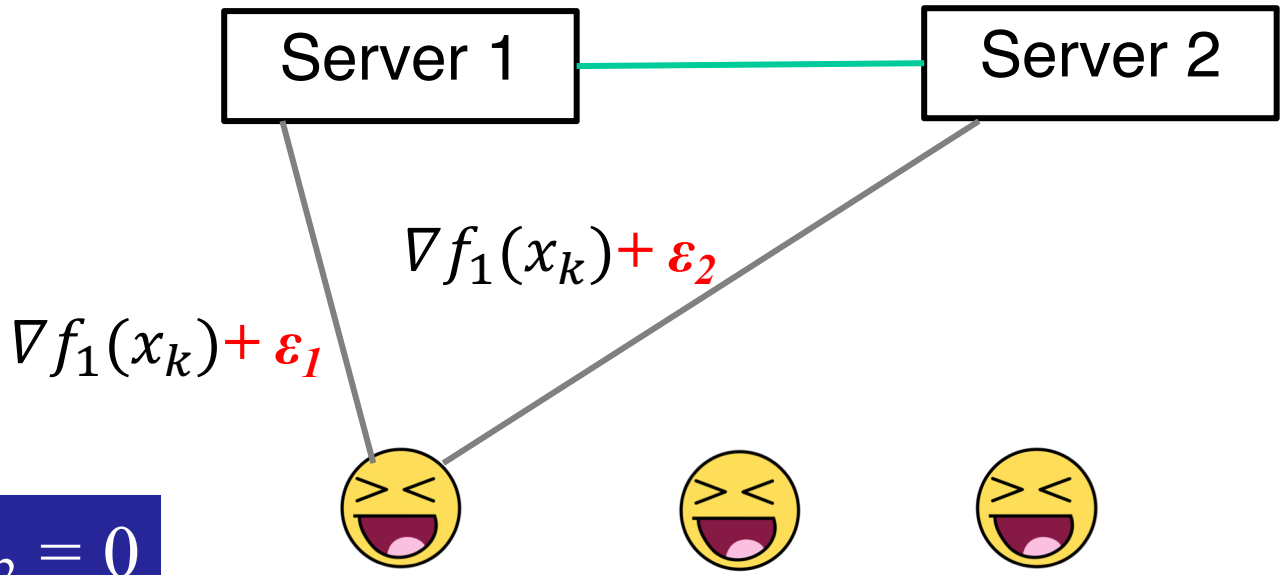
- Motivated by secret sharing & differential privacy

➔ Add cancellable noise

Multiple Parameter Servers

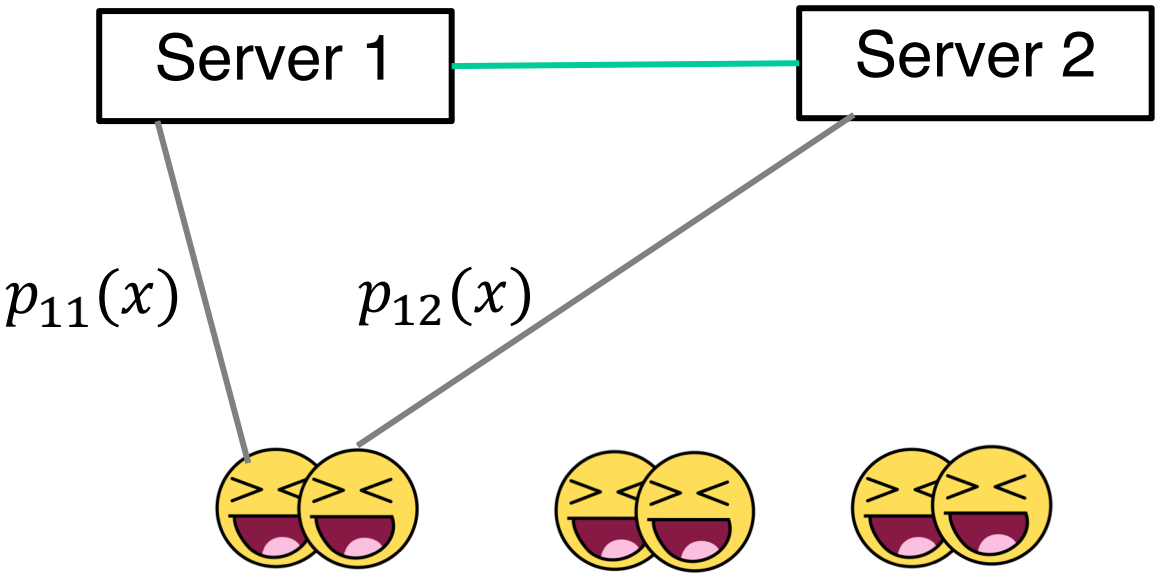


Improving Privacy



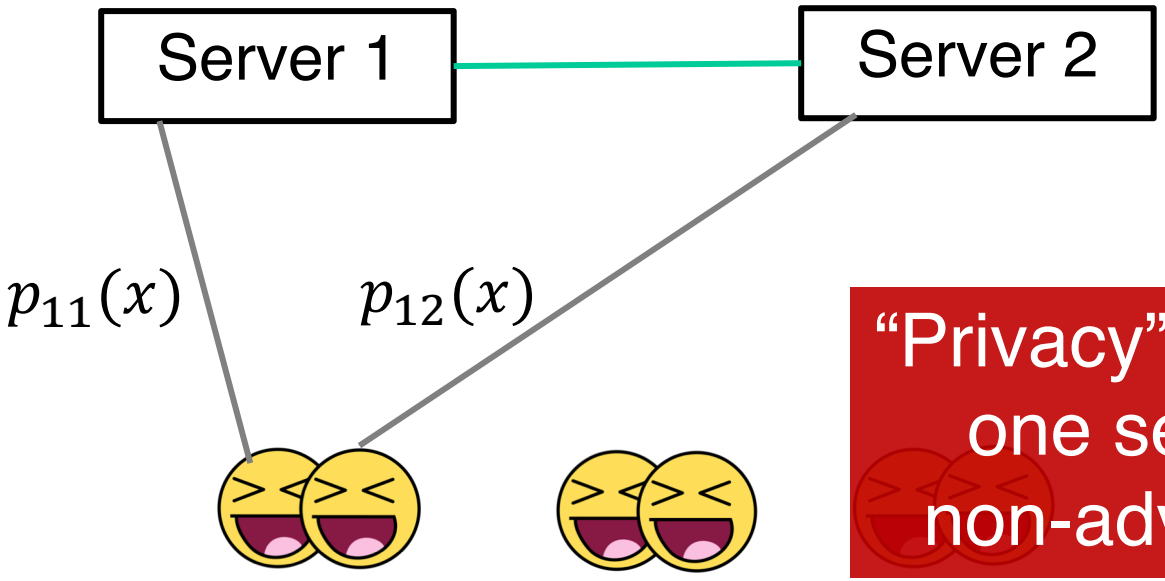
$\varepsilon_1 + \varepsilon_2 = 0$
over time

Convex Sum of Non-Convex Functions



$$p_{11}(x) + p_{12}(x) = f_1(x)$$

Convex Sum of Non-Convex Functions



"Privacy" if at least one server is non-adversarial

$$p_{11}(x) + p_{12}(x) = f_1(x)$$

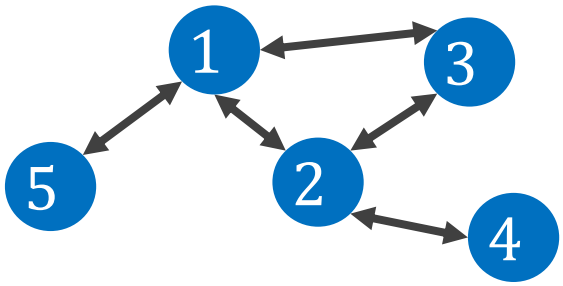
Summary: argmin $\sum f_i(x)$

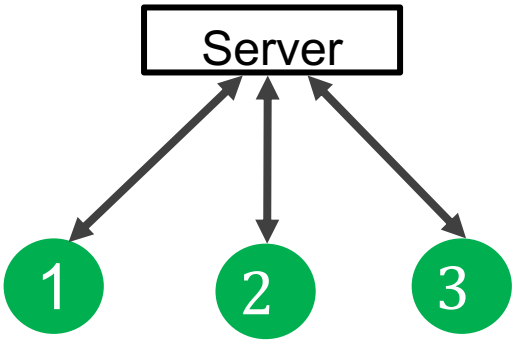
Fault-tolerance

➔ Gradient filters

Privacy

➔ Cancellable noise





Lili Su

Shripad Gade

Nirupam Gupta

Dimitrios Pylorof

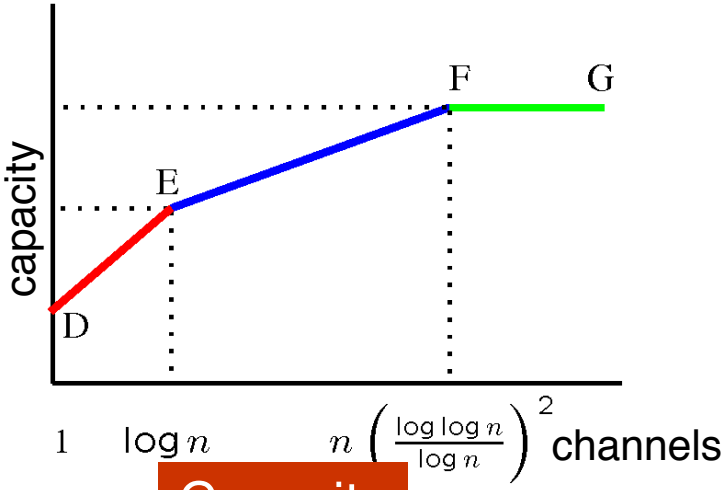
Shuo Liu

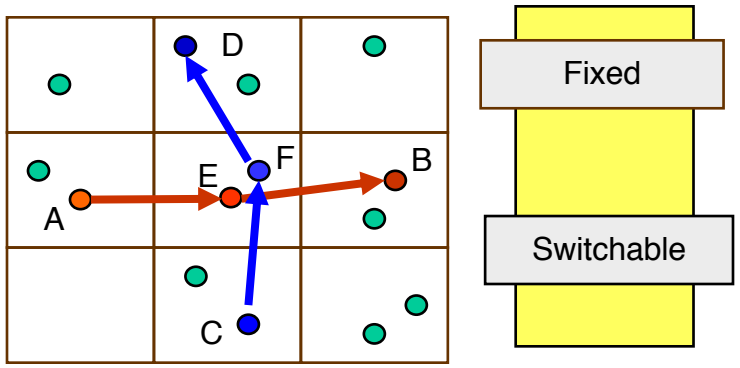
Thanks!

disc.georgetown.domains

Net-X: Multi-Channel Mesh

Theory to Practice





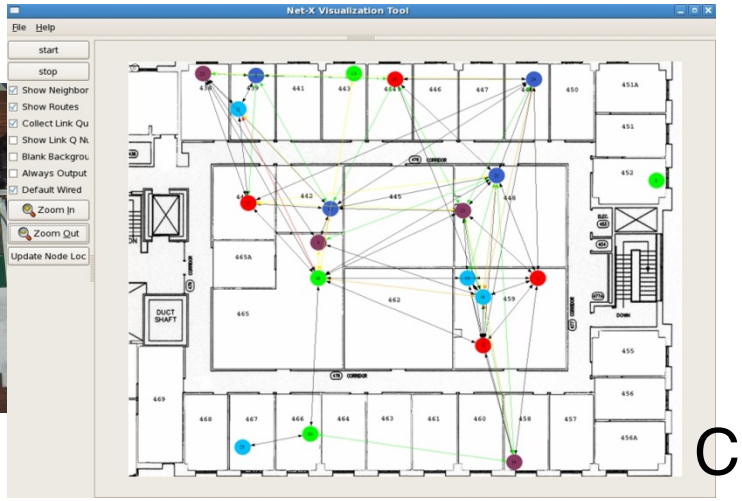
Capacity bounds

Insights on protocol design

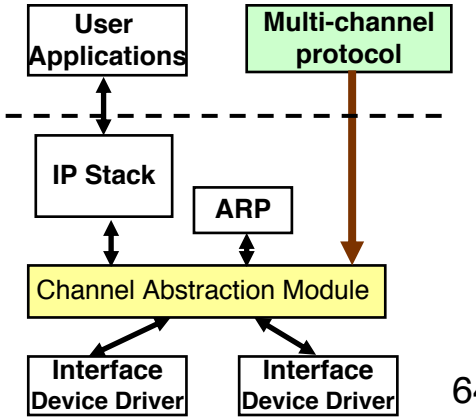
OS improvements Software architecture

Net-X testbed

Linux box



CSL



Decentralized Control/Optimization

Distributed Computing

Thanks!

disc.georgetown.domains

Averaging ➔ Optimization

- Input of agent i = a_i

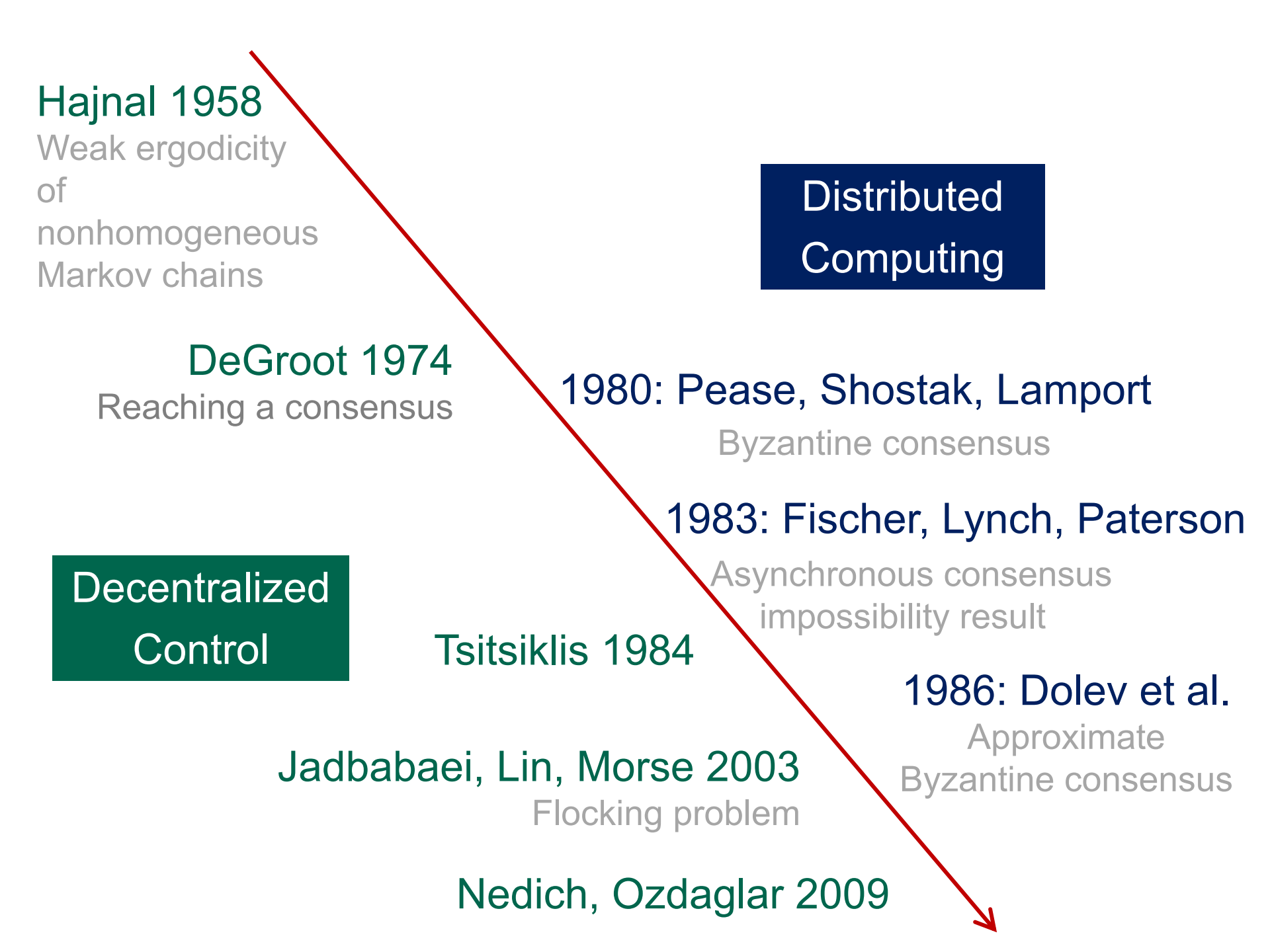
- Average of $a_i's$

Averaging ➔ Optimization

- Input of agent i = a_i

- Average of $a_i's$ = $\operatorname{argmin} \sum f_i(x)$

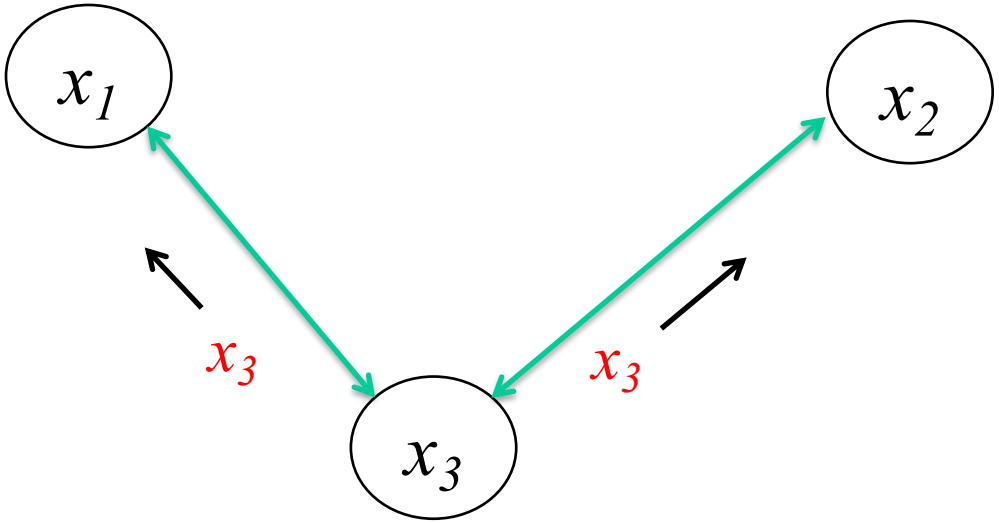
where $f_i(x) = (x - a_i)^2$


Hajnal 1958
Weak ergodicity of nonhomogeneous Markov chains
Distributed Computing
DeGroot 1974
Reaching a consensus
1980: Pease, Shostak, Lamport
Byzantine consensus
1983: Fischer, Lynch, Paterson
Asynchronous consensus impossibility result
Decentralized Control
Tsitsiklis 1984
1986: Dolev et al.
Approximate Byzantine consensus
Jadbabaei, Lin, Morse 2003
Flocking problem
Nedich, Ozdaglar 2009

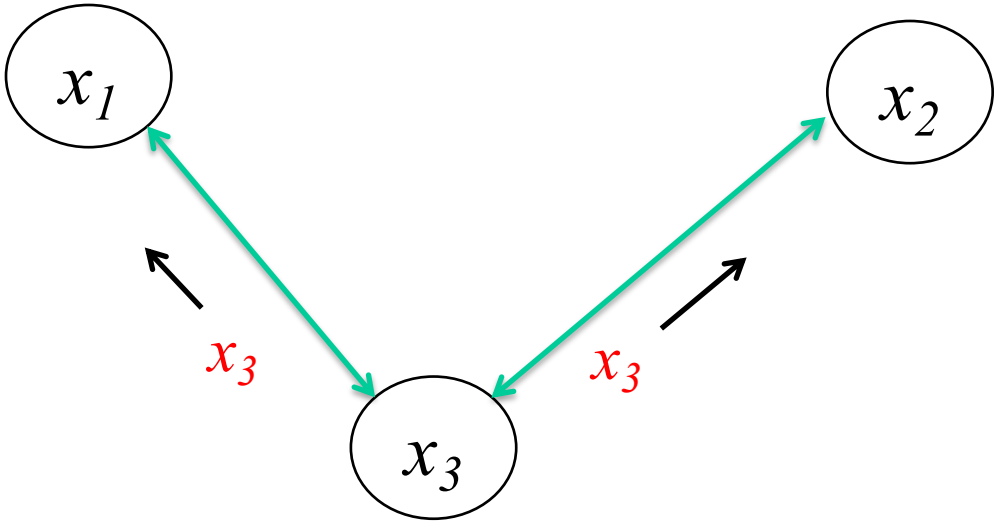
Distributed Optimization

$$f(x) = \sum f_i(x)$$

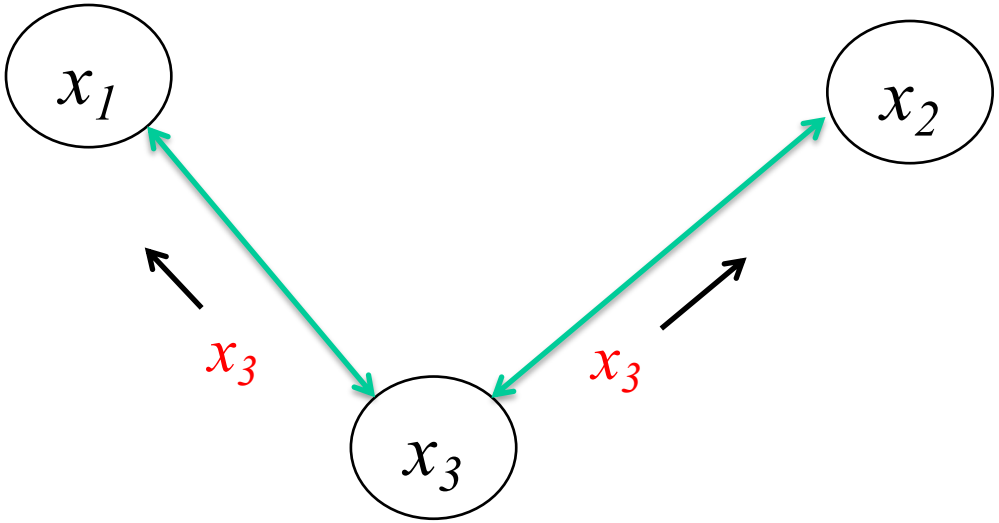
- Each agent maintains an estimate
- Local estimates shared with neighbors & updated
- Estimates converge to optimum



Example based on [Nedic and Ozdaglar, 2009]



$$x_1 \leftarrow \frac{2}{3}x_1 + \frac{1}{3}x_3 - \alpha \nabla f_1(x_1)$$



$$x_1 \leftarrow \frac{2}{3}x_1 + \frac{1}{3}x_3 - \alpha \nabla f_1(x_1)$$

$$x_3 \leftarrow \frac{1}{3}x_1 + \frac{1}{3}x_2 + \frac{1}{3}x_3 - \alpha \nabla f_3(x_3)$$

Distributed Optimization

As time → ∞

- Consensus: All agents converge to same estimate
- Optimality

$$\operatorname{argmin} \sum f_i(x)$$

Filter for Scalar Parameter x

t faulty agents

Filter for Scalar Parameter x

t faulty agents

- Discard smallest t and largest t gradients
- Average the rest

Filter for Scalar Parameter x

t faulty agents

- Discard smallest t and largest t gradients
- Average the rest

- t = 1 Filter(1, 4, 6, 0, 2) = $\frac{1+2+4}{3}$

Filter for Scalar Parameter x

t faulty agents

- Discard smallest t and largest t gradients
- Average the rest

- t = 1 Filter(1, 4, 6, 0, 2) $= \frac{1+2+4}{3}$

What does this achieve?

Our Ideal Goal

Optimize $\sum_{i \in G} f_i(x)$

- Equal importance to each good agent's cost
- Each f_i has identical weight in the aggregate cost

Independent Functions
How good an approximation?

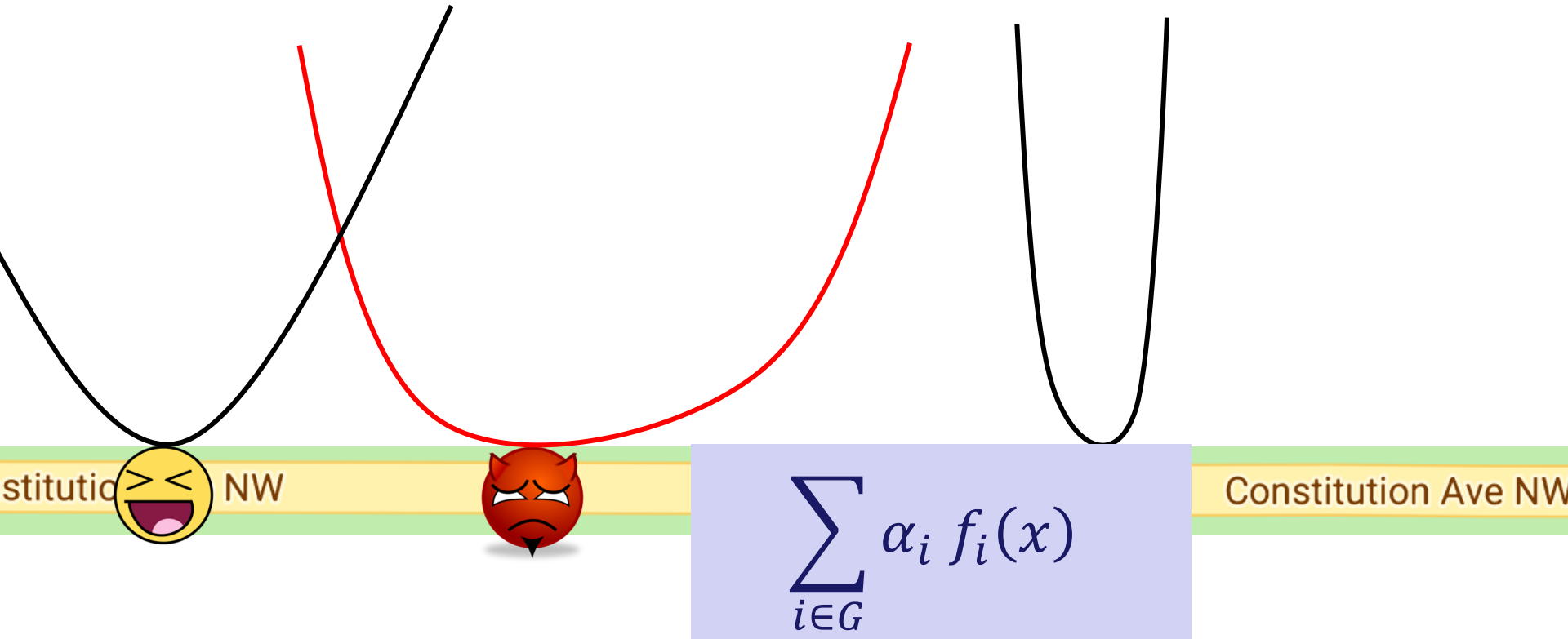
- Instead of uniform weights

$$\sum_{i \in G} f_i(x)$$

the filter achieves unequal weights

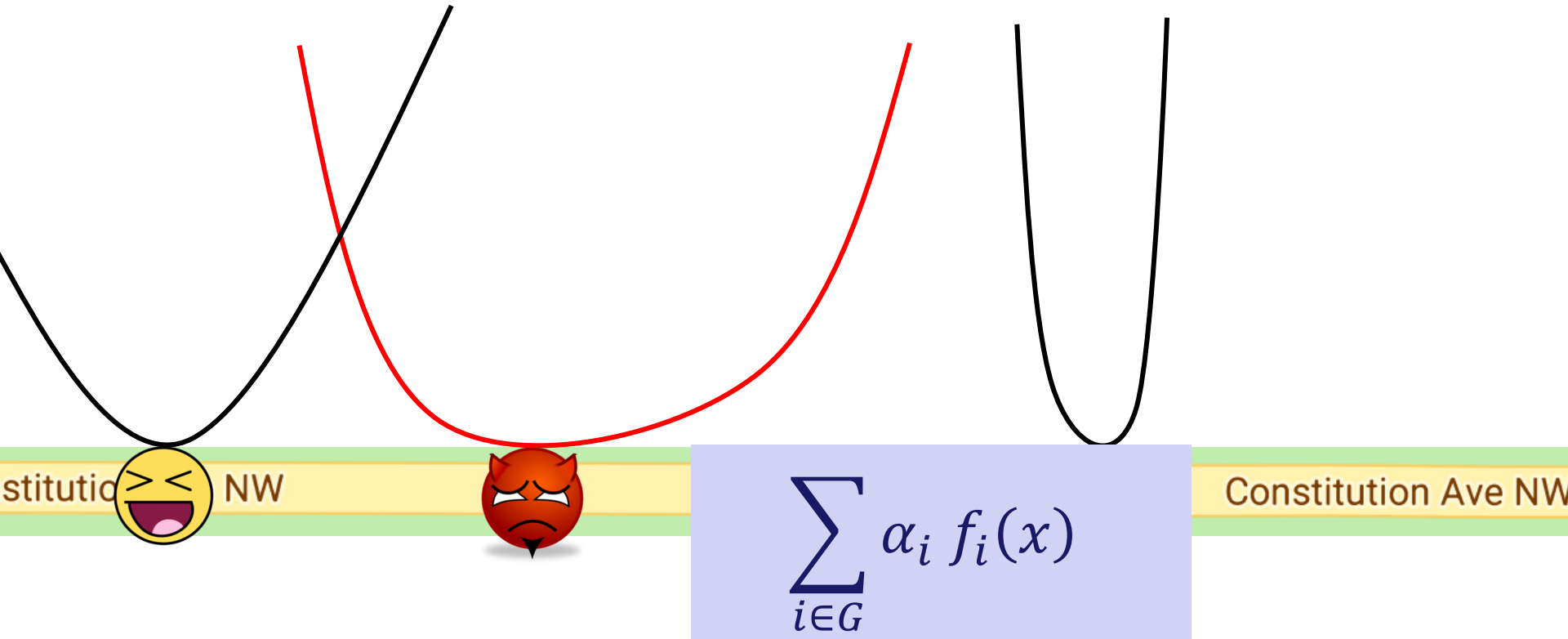
$$\sum_{i \in G} \alpha_i \, f_i(x)$$

Independent Functions



Cost functions of faulty nodes “filtered away”

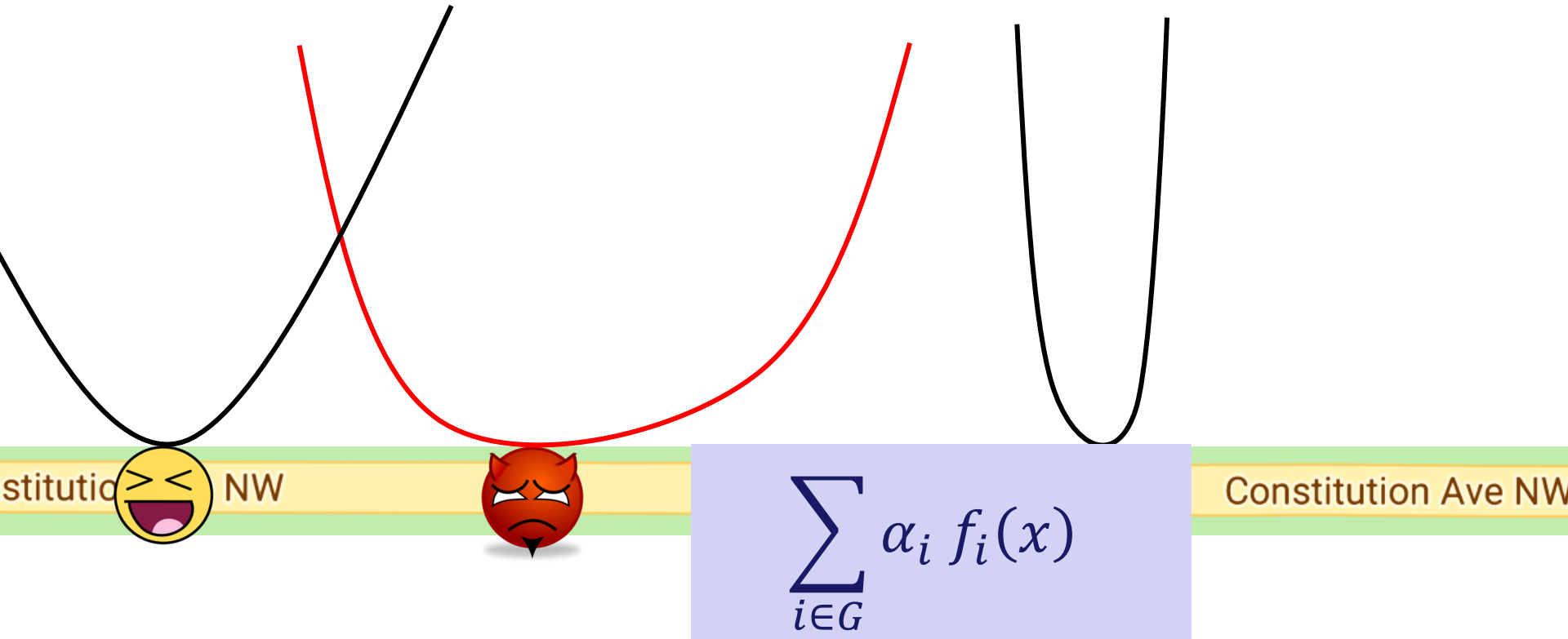
Independent Functions



Cost functions of faulty nodes “filtered away”

At most t good cost functions also filtered away

Independent Functions



Cost functions of faulty nodes “filtered away”

At most t good cost functions also filtered away

Nearly uniform importance to the remaining costs

Independent Functions
How good an approximation?

$$\sum_{i \in G} \alpha_i \, f_i(x)$$

α	0	0	¼	¼	¼	¼	0	0

Independent Functions
How good an approximation?

$$\sum_{i \in G} \alpha_i \, f_i(x)$$

α	0	0	¼	¼	¼	¼	0	0
α	⅛	⅛	⅛	⅛	¼	¼	0	0

Independent Functions

- Necessary condition in general

$$n > (d+1)t$$ for d-dimensional parameter x

Bad news for large d

Good News

Cost functions often naturally redundant

Good News

Cost functions often naturally redundant

- Data sets at different agents may be drawn from same distribution
  - ➔ In expectation, all cost functions are identical

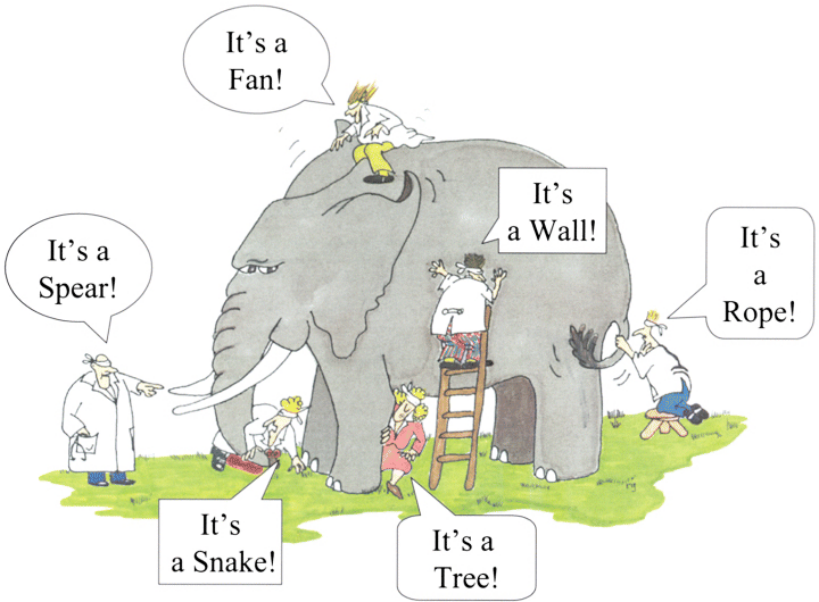
Good News

Cost functions often naturally redundant

- Data sets at different agents may be drawn from same distribution

  ➔ In expectation, all cost functions are identical

- Observations by different agents conditioned on the same ground truth



Good News

Cost functions often naturally redundant

- With redundancy in functions, possible to compute

$$\operatorname{argmin} \sum_{i \in G} f_i(x)$$

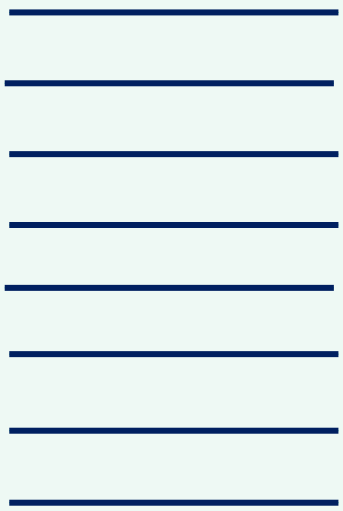
with far fewer than $(d+1)t$ agents

Linear Regression

Linear Regression

- d-dimensional x^*

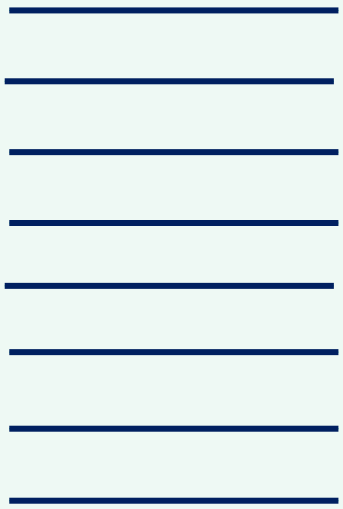
- n-by-d matrix A



Linear Regression

- d-dimensional x^*

- n-by-d matrix A

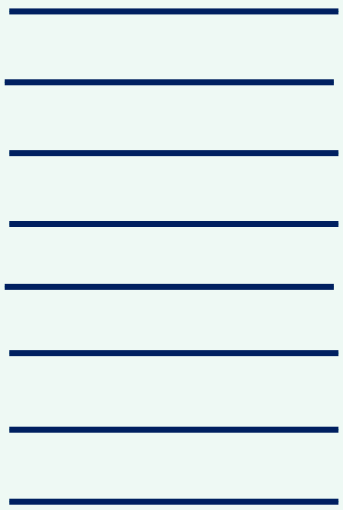


- n-dimensional y

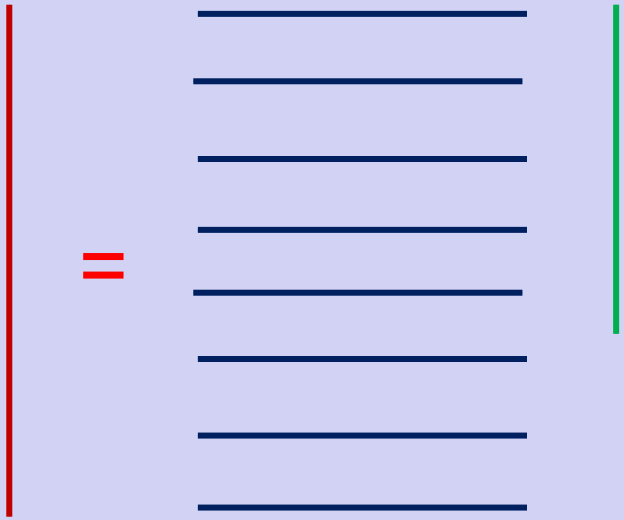
Linear Regression

- d-dimensional x^*

- n-by-d matrix A



- n-dimensional y



$$y = A\,x^*$$

Distributed Linear Regression

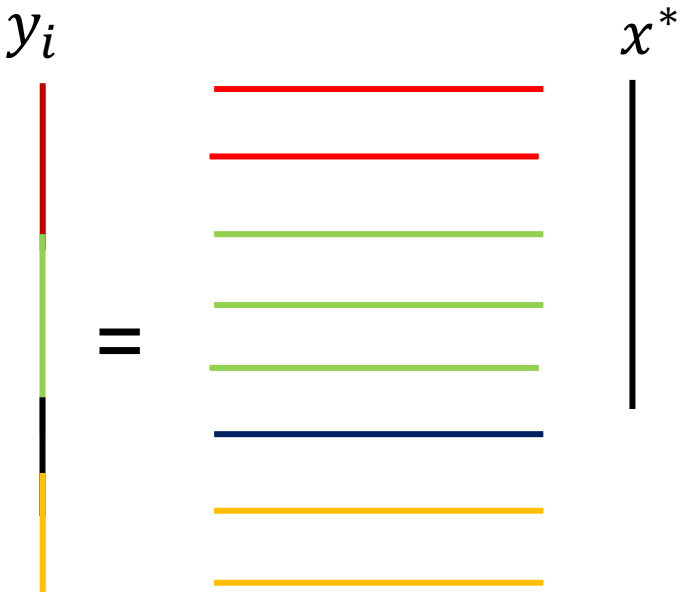
- Agent i knows A_i and y_i where

$$y_i = A_i x^*$$

Distributed Linear Regression

- Agent i knows A_i and y_i where

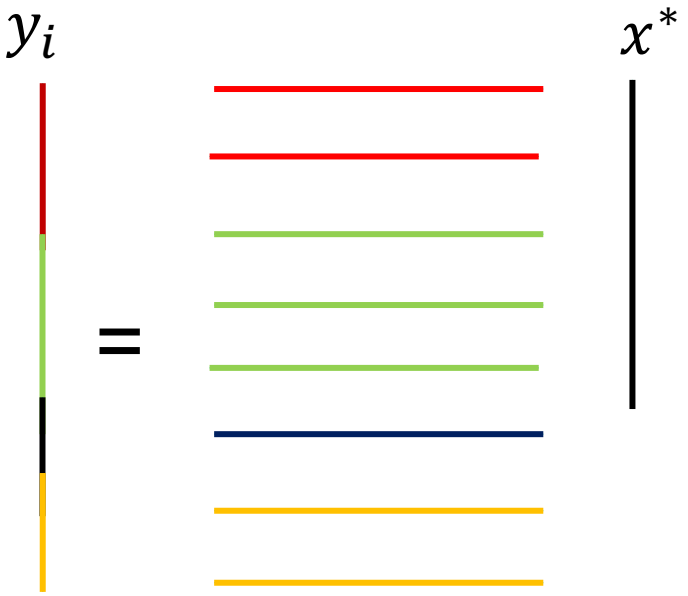
$$y_i = A_i\, x^*$$



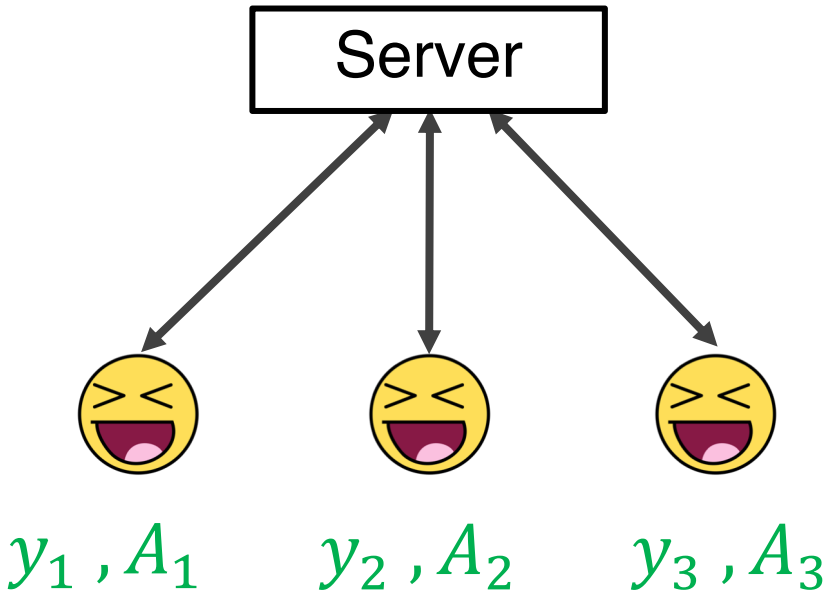
Distributed Linear Regression

- Agent i knows A_i and y_i where

$$y_i = A_i\, x^*$$



- Determine x^* allowing for t agents to be faulty


Server
y_1 , A_1
y_2 , A_2
y_3 , A_3

2t-Redundancy … Linear Regression

- Any $n - 2t$ agents have enough information to compute x^*

2t-Redundancy … Linear Regression

- Any $n - 2t$ agents have enough information to compute x^*

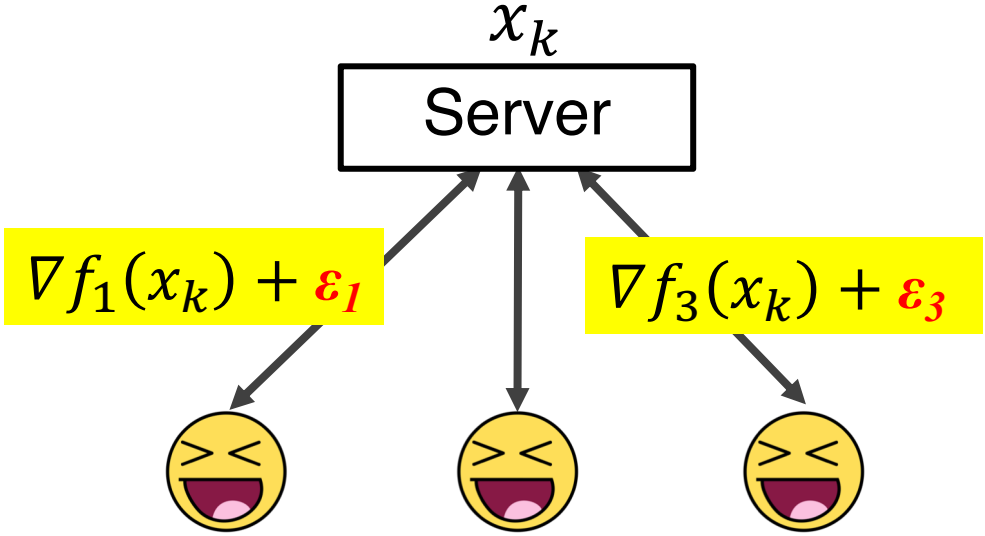
- Define cost function of agent i as

$$f_i(x) = (y_i - A_i x)^2$$

- The cost functions satisfy $2t$-redundancy

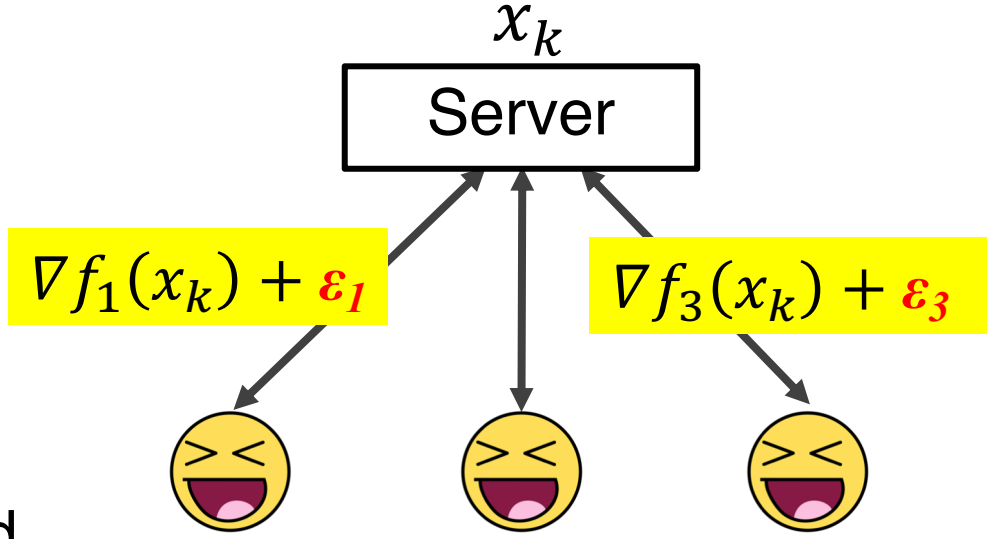
Privacy Techniques

Differential privacy … add noise ε



Privacy Techniques

Differential privacy … add noise ε



- Optimality compromised due to the noise

$$\neq \operatorname{argmin} \sum f_i(x)$$

Privacy Techniques

Homomorphic encryption

- Expensive