Сверхбольшие задачи оптимизации

Ю.Е. Нестеров[1]

PremoDay2, МФТИ, 5 октября 2013г.

[1]ПРЕМОЛАБ Московский физико-технический институт (ГУ)

Phenomena of Big Data (1.9M references by Goggle)

Phenomena of Big Data (1.9M references by Goggle)

1. Due to the fast development of Internet, the daily rate of data accumulation is enormous.

Phenomena of Big Data (1.9M references by Goggle)

1. Due to the fast development of Internet, the daily rate of data accumulation is enormous.

($2.5 \cdot 10^{18}$ bytes per day)

Phenomena of Big Data (1.9M references by Goggle)

1. Due to the fast development of Internet, the daily rate of data accumulation is enormous.

($2.5 \cdot 10^{18}$ bytes per day)

2. This data is easily accessible.

Phenomena of Big Data (1.9M references by Goggle)

1. Due to the fast development of Internet, the daily rate of data accumulation is enormous.

$(2.5 \cdot 10^{18}$ bytes per day)

2. This data is easily accessible.

3. As a result, we see new style of modelling in all fields.

Phenomena of Big Data (1.9M references by Goggle)

1. Due to the fast development of Internet, the daily rate of data accumulation is enormous.

($2.5 \cdot 10^{18}$ bytes per day)

2. This data is easily accessible.

3. As a result, we see new style of modelling in <u>all fields</u>.

4. Explosion of scientific activity right now.

Phenomena of Big Data (1.9M references by Goggle)

1. Due to the fast development of Internet, the daily rate of data accumulation is enormous.

($2.5 \cdot 10^{18}$ bytes per day)

2. This data is easily accessible.

3. As a result, we see new style of modelling in all fields.

4. Explosion of scientific activity right now.

(In 2012, Obama administration announced the Big Data Research and Development Initiative.)

Phenomena of Big Data (1.9M references by Goggle)

1. Due to the fast development of Internet, the daily rate of data accumulation is enormous.

($2.5 \cdot 10^{18}$ bytes per day)

2. This data is easily accessible.

3. As a result, we see new style of modelling in all fields.

4. Explosion of scientific activity right now.

(In 2012, Obama administration announced the Big Data Research and Development Initiative.)

But: We have started this two years ago.

Optimization group in Premolab

Optimization group in Premolab

1. Optimization problems in Large Networks (B.Polyak).

Optimization group in Premolab

1. Optimization problems in Large Networks (B.Polyak).

2. Optimal design of mechanical structures (S.Shpirko).

Optimization group in Premolab

1. Optimization problems in Large Networks (B.Polyak).

2. Optimal design of mechanical structures (S.Shpirko).

3. Modelling congested traffic in large transportation systems (A.Gasnikov).

Optimization group in Premolab

1. Optimization problems in Large Networks (B.Polyak).

2. Optimal design of mechanical structures (S.Shpirko).

3. Modelling congested traffic in large transportation systems (A.Gasnikov).

Our research style:

Optimization group in Premolab

1. Optimization problems in Large Networks (B.Polyak).

2. Optimal design of mechanical structures (S.Shpirko).

3. Modelling congested traffic in large transportation systems (A.Gasnikov).

Our research style:

MODELLING

Optimization group in Premolab

1. Optimization problems in Large Networks (B.Polyak).

2. Optimal design of mechanical structures (S.Shpirko).

3. Modelling congested traffic in large transportation systems (A.Gasnikov).

Our research style:

MODELLING $\Leftrightarrow$ OPTIMIZATION METHODS

Optimization group in Premolab

1. Optimization problems in Large Networks (B.Polyak).

2. Optimal design of mechanical structures (S.Shpirko).

3. Modelling congested traffic in large transportation systems (A.Gasnikov).

Our research style:

MODELLING $\Leftrightarrow$ OPTIMIZATION METHODS $\Leftrightarrow$ COMPLEXITY ANALYSIS

Nonlinear Optimization: problems sizes

Class	Operations	Dimension	Iter.Cost	Memory	
Small-size	All	$10^0 - 10^2$	$n^4 \to n^3$	KBytes:	10^3
Medium-size	A^{-1}	$10^3 - 10^4$	$n^3 \to n^2$	MBytes:	10^6

Nonlinear Optimization: problems sizes

Class	Operations	Dimension	Iter.Cost	Memory	
Small-size	All	$10^0 - 10^2$	$n^4 \to n^3$	KBytes:	10^3
Medium-size	A^{-1}	$10^3 - 10^4$	$n^3 \to n^2$	MBytes:	10^6
Large-scale	Ax	$10^5 - 10^7$	$n^2 \to n$	GBytes:	10^9

Nonlinear Optimization: problems sizes

Class	Operations	Dimension	Iter.Cost	Memory	
Small-size	All	$10^0 - 10^2$	$n^4 \to n^3$	KBytes:	10^3
Medium-size	A^{-1}	$10^3 - 10^4$	$n^3 \to n^2$	MBytes:	10^6
Large-scale	Ax	$10^5 - 10^7$	$n^2 \to n$	GBytes:	10^9
Huge-scale	$x + y$	$10^8 - 10^{12}$	$n \to \log n$	TBytes:	10^{12}

Nonlinear Optimization: problems sizes

Class	Operations	Dimension	Iter.Cost	Memory	
Small-size	All	$10^0 - 10^2$	$n^4 \to n^3$	KBytes:	10^3
Medium-size	A^{-1}	$10^3 - 10^4$	$n^3 \to n^2$	MBytes:	10^6
Large-scale	Ax	$10^5 - 10^7$	$n^2 \to n$	GBytes:	10^9
Huge-scale	$x + y$	$10^8 - 10^{12}$	$n \to \log n$	TBytes:	10^{12}

Sources of Huge-Scale problems

Nonlinear Optimization: problems sizes

Class	Operations	Dimension	Iter.Cost	Memory	
Small-size	All	$10^0 - 10^2$	$n^4 \to n^3$	KBytes:	10^3
Medium-size	A^{-1}	$10^3 - 10^4$	$n^3 \to n^2$	MBytes:	10^6
Large-scale	Ax	$10^5 - 10^7$	$n^2 \to n$	GBytes:	10^9
Huge-scale	$x + y$	$10^8 - 10^{12}$	$n \to \log n$	TBytes:	10^{12}

Sources of Huge-Scale problems

- New: Internet, Telecommunications.

Nonlinear Optimization: problems sizes

Class	Operations	Dimension	Iter.Cost	Memory	
Small-size	All	$10^0 - 10^2$	$n^4 \to n^3$	KBytes:	10^3
Medium-size	A^{-1}	$10^3 - 10^4$	$n^3 \to n^2$	MBytes:	10^6
Large-scale	Ax	$10^5 - 10^7$	$n^2 \to n$	GBytes:	10^9
Huge-scale	$x + y$	$10^8 - 10^{12}$	$n \to \log n$	TBytes:	10^{12}

Sources of Huge-Scale problems

- New: Internet, Telecommunications.
- Very new: Intelligent data mining.

Nonlinear Optimization: problems sizes

Class	Operations	Dimension	Iter.Cost	Memory	
Small-size	All	$10^0 - 10^2$	$n^4 \to n^3$	KBytes:	10^3
Medium-size	A^{-1}	$10^3 - 10^4$	$n^3 \to n^2$	MBytes:	10^6
Large-scale	Ax	$10^5 - 10^7$	$n^2 \to n$	GBytes:	10^9
Huge-scale	$x + y$	$10^8 - 10^{12}$	$n \to \log n$	TBytes:	10^{12}

Sources of Huge-Scale problems

- New: Internet, Telecommunications.
- Very new: Intelligent data mining.
- Old: Finite-element schemes, PDE, Weather prediction.

Nonlinear Optimization: problems sizes

Class	Operations	Dimension	Iter.Cost	Memory	
Small-size	All	$10^0 - 10^2$	$n^4 \to n^3$	KBytes:	10^3
Medium-size	A^{-1}	$10^3 - 10^4$	$n^3 \to n^2$	MBytes:	10^6
Large-scale	Ax	$10^5 - 10^7$	$n^2 \to n$	GBytes:	10^9
Huge-scale	$x + y$	$10^8 - 10^{12}$	$n \to \log n$	TBytes:	10^{12}

Sources of Huge-Scale problems

- New: Internet, Telecommunications.
- Very new: Intelligent data mining.
- Old: Finite-element schemes, PDE, Weather prediction.

Main hope: *Sparsity*.

Nonlinear Optimization: problems sizes

Class	Operations	Dimension	Iter.Cost	Memory	
Small-size	All	$10^0 - 10^2$	$n^4 \to n^3$	KBytes:	10^3
Medium-size	A^{-1}	$10^3 - 10^4$	$n^3 \to n^2$	MBytes:	10^6
Large-scale	Ax	$10^5 - 10^7$	$n^2 \to n$	GBytes:	10^9
Huge-scale	$x + y$	$10^8 - 10^{12}$	$n \to \log n$	TBytes:	10^{12}

Sources of Huge-Scale problems

- New: Internet, Telecommunications.
- Very new: Intelligent data mining.
- Old: Finite-element schemes, PDE, Weather prediction.

Main hope: *Sparsity*. **Main tools:** *Right* methods.

Sparse problems

Sparse problems

Problem: $\min_{x \in Q} f(x)$, where Q is closed and convex in R^N,

Sparse problems

Problem: $\min_{x \in Q} f(x)$, where Q is closed and convex in R^N, and

- $f(x) = \Psi(Ax)$, where Ψ is a simple *convex function*:

$$\Psi(y_1) \geq \Psi(y_2) + \langle \Psi'(y_2), y_1 - y_2 \rangle \quad \forall y_1, y_2 \in R^M,$$

Sparse problems

Problem: $\min_{x \in Q} f(x)$, where Q is closed and convex in R^N, and

- $f(x) = \Psi(Ax)$, where Ψ is a simple *convex function*:
  $$\Psi(y_1) \geq \Psi(y_2) + \langle \Psi'(y_2), y_1 - y_2 \rangle \quad \forall y_1, y_2 \in R^M,$$
- $A : R^N \to R^M$ is a *sparse* matrix.

Sparse problems

Problem: $\min_{x \in Q} f(x)$, where Q is closed and convex in R^N, and

- $f(x) = \Psi(Ax)$, where Ψ is a simple *convex function*:
  $$\Psi(y_1) \geq \Psi(y_2) + \langle \Psi'(y_2), y_1 - y_2 \rangle \quad \forall y_1, y_2 \in R^M,$$
- $A : R^N \to R^M$ is a *sparse* matrix.

Let $p(x) = \#$ of $x^{(i)} \neq 0$.

Sparse problems

Problem: $\min\limits_{x \in Q} f(x)$, where Q is closed and convex in R^N, and

- $f(x) = \Psi(Ax)$, where Ψ is a simple *convex function*:
$$\Psi(y_1) \geq \Psi(y_2) + \langle \Psi'(y_2), y_1 - y_2 \rangle \quad \forall y_1, y_2 \in R^M,$$
- $A : R^N \to R^M$ is a *sparse* matrix.

Let $p(x) = \#$ of $x^{(i)} \neq 0$. Sparsity coefficient: $\gamma(A) \stackrel{\text{def}}{=} \frac{p(A)}{MN}$.

Sparse problems

Problem: $\min\limits_{x \in Q} f(x)$, where Q is closed and convex in R^N, and

- $f(x) = \Psi(Ax)$, where Ψ is a simple *convex function*:
  $$\Psi(y_1) \geq \Psi(y_2) + \langle \Psi'(y_2), y_1 - y_2 \rangle \quad \forall y_1, y_2 \in R^M,$$
- $A : R^N \to R^M$ is a *sparse* matrix.

Let $p(x) = \#$ of $x^{(i)} \neq 0$. Sparsity coefficient: $\gamma(A) \stackrel{\text{def}}{=} \frac{p(A)}{MN}$.

Example: Matrix-vector multiplication

Sparse problems

Problem: $\min_{x \in Q} f(x)$, where Q is closed and convex in R^N, and

- $f(x) = \Psi(Ax)$, where Ψ is a simple *convex function*:
  $$\Psi(y_1) \geq \Psi(y_2) + \langle \Psi'(y_2), y_1 - y_2 \rangle \quad \forall y_1, y_2 \in R^M,$$
- $A : R^N \to R^M$ is a *sparse* matrix.

Let $p(x) = \#$ of $x^{(i)} \neq 0$. Sparsity coefficient: $\gamma(A) \stackrel{\text{def}}{=} \frac{p(A)}{MN}$.

Example: Matrix-vector multiplication

- Computation of vector Ax needs $p(A)$ operations.

Sparse problems

Problem: $\min_{x \in Q} f(x)$, where Q is closed and convex in R^N, and

- $f(x) = \Psi(Ax)$, where Ψ is a simple *convex function*:
  $$\Psi(y_1) \geq \Psi(y_2) + \langle \Psi'(y_2), y_1 - y_2 \rangle \quad \forall y_1, y_2 \in R^M,$$
- $A : R^N \to R^M$ is a *sparse* matrix.

Let $p(x) = \#$ of $x^{(i)} \neq 0$. Sparsity coefficient: $\gamma(A) \stackrel{\mathrm{def}}{=} \frac{p(A)}{MN}$.

Example: Matrix-vector multiplication

- Computation of vector Ax needs $p(A)$ operations.
- Initial complexity MN is reduced in $\gamma(A)$ times.

Sparse problems

Problem: $\min_{x \in Q} f(x)$, where Q is closed and convex in R^N, and

- $f(x) = \Psi(Ax)$, where Ψ is a simple *convex function*:
  $$\Psi(y_1) \geq \Psi(y_2) + \langle \Psi'(y_2), y_1 - y_2 \rangle \quad \forall y_1, y_2 \in R^M,$$
- $A : R^N \to R^M$ is a *sparse* matrix.

Let $p(x) = \#$ of $x^{(i)} \neq 0$. Sparsity coefficient: $\gamma(A) \stackrel{\text{def}}{=} \frac{p(A)}{MN}$.

Example: Matrix-vector multiplication

- Computation of vector Ax needs $p(A)$ operations.
- Initial complexity MN is reduced in $\gamma(A)$ times.

Note: For Large- and Huge-scale problems, we often have
$$\gamma(A) \approx 10^{-4} \dots 10^{-6}.$$

Sparse problems

Problem: $\min_{x \in Q} f(x)$, where Q is closed and convex in R^N, and

- $f(x) = \Psi(Ax)$, where Ψ is a simple *convex function*:
  $$\Psi(y_1) \geq \Psi(y_2) + \langle \Psi'(y_2), y_1 - y_2 \rangle \quad \forall y_1, y_2 \in R^M,$$
- $A : R^N \to R^M$ is a *sparse* matrix.

Let $p(x) = \#$ of $x^{(i)} \neq 0$. Sparsity coefficient: $\gamma(A) \stackrel{\text{def}}{=} \frac{p(A)}{MN}$.

Example: Matrix-vector multiplication

- Computation of vector Ax needs $p(A)$ operations.
- Initial complexity MN is reduced in $\gamma(A)$ times.

Note: For Large- and Huge-scale problems, we often have
$\gamma(A) \approx 10^{-4} \dots 10^{-6}$. **Can we get more?**

Sparse updating strategy

Sparse updating strategy

Main idea

Sparse updating strategy

Main idea

- After update $x_+ = x + d$

Sparse updating strategy

Main idea

- After update $x_+ = x + d$ we have $y_+ \stackrel{\text{def}}{=} Ax_+$

Sparse updating strategy

Main idea

- After update $x_+ = x + d$ we have $y_+ \overset{\text{def}}{=} Ax_+ = \underbrace{Ax}_{y} + Ad.$

Sparse updating strategy

Main idea

- After update $x_+ = x + d$ we have $y_+ \overset{\text{def}}{=} Ax_+ = \underbrace{Ax}_{y} + Ad$.
- What happens if d is *sparse*?

Sparse updating strategy

Main idea

- After update $x_+ = x + d$ we have $y_+ \stackrel{\text{def}}{=} Ax_+ = \underbrace{Ax}_{y} + Ad$.
- What happens if d is *sparse*?

Denote $\sigma(d) = \{j : d^{(j)} \neq 0\}$.

Sparse updating strategy

Main idea

- After update $x_+ = x + d$ we have $y_+ \stackrel{\text{def}}{=} Ax_+ = \underbrace{Ax}_{y} + Ad$.
- What happens if d is *sparse*?

Denote $\sigma(d) = \{j : d^{(j)} \neq 0\}$. Then $y_+ = y + \sum\limits_{j \in \sigma(d)} d^{(j)} \cdot Ae_j$.

Sparse updating strategy

Main idea

- After update $x_+ = x + d$ we have $y_+ \stackrel{\text{def}}{=} Ax_+ = \underbrace{Ax}_{y} + Ad$.
- What happens if d is *sparse*?

Denote $\sigma(d) = \{j : d^{(j)} \neq 0\}$. Then $y_+ = y + \sum\limits_{j \in \sigma(d)} d^{(j)} \cdot Ae_j$.

Its complexity, $\kappa_A(d) \stackrel{\text{def}}{=} \sum\limits_{j \in \sigma(d)} p(Ae_j)$,

Sparse updating strategy

Main idea

- After update $x_+ = x + d$ we have $y_+ \stackrel{\text{def}}{=} Ax_+ = \underbrace{Ax}_{y} + Ad$.
- What happens if d is *sparse*?

Denote $\sigma(d) = \{j : d^{(j)} \neq 0\}$. Then $y_+ = y + \sum\limits_{j \in \sigma(d)} d^{(j)} \cdot Ae_j$.

Its complexity, $\kappa_A(d) \stackrel{\text{def}}{=} \sum\limits_{j \in \sigma(d)} p(Ae_j)$, can be VERY small!

Sparse updating strategy

Main idea

- After update $x_+ = x + d$ we have $y_+ \stackrel{\text{def}}{=} Ax_+ = \underbrace{Ax}_{y} + Ad$.
- What happens if d is *sparse*?

Denote $\sigma(d) = \{j : d^{(j)} \neq 0\}$. Then $y_+ = y + \sum\limits_{j \in \sigma(d)} d^{(j)} \cdot Ae_j$.

Its complexity, $\kappa_A(d) \stackrel{\text{def}}{=} \sum\limits_{j \in \sigma(d)} p(Ae_j)$, can be VERY small!

If $\gamma(d) \leq c\gamma(A)$, $\gamma(A_j) \leq c\gamma(A)$,

Sparse updating strategy

Main idea

- After update $x_+ = x + d$ we have $y_+ \stackrel{\text{def}}{=} Ax_+ = \underbrace{Ax}_{y} + Ad$.
- What happens if d is *sparse*?

Denote $\sigma(d) = \{j : d^{(j)} \neq 0\}$. Then $y_+ = y + \sum\limits_{j \in \sigma(d)} d^{(j)} \cdot Ae_j$.

Its complexity, $\kappa_A(d) \stackrel{\text{def}}{=} \sum\limits_{j \in \sigma(d)} p(Ae_j)$, can be VERY small!

If $\gamma(d) \leq c\gamma(A)$, $\gamma(A_j) \leq c\gamma(A)$, $\Rightarrow$ $\boxed{\kappa_A(d) \leq c^2 \cdot \gamma^2(A) \cdot MN}$

Sparse updating strategy

Main idea

- After update $x_+ = x + d$ we have $y_+ \stackrel{\mathrm{def}}{=} Ax_+ = \underbrace{Ax}_{y} + Ad$.
- What happens if d is *sparse*?

Denote $\sigma(d) = \{j : d^{(j)} \neq 0\}$. Then $y_+ = y + \sum\limits_{j \in \sigma(d)} d^{(j)} \cdot Ae_j$.

Its complexity, $\kappa_A(d) \stackrel{\mathrm{def}}{=} \sum\limits_{j \in \sigma(d)} p(Ae_j)$, can be VERY small!

If $\gamma(d) \leq c\gamma(A)$, $\gamma(A_j) \leq c\gamma(A)$, $\Rightarrow$ $\boxed{\kappa_A(d) \leq c^2 \cdot \gamma^2(A) \cdot MN}$

Expected acceleration:

Sparse updating strategy

Main idea

- After update $x_+ = x + d$ we have $y_+ \stackrel{\text{def}}{=} Ax_+ = \underbrace{Ax}_{y} + Ad$.
- What happens if d is *sparse*?

Denote $\sigma(d) = \{j : d^{(j)} \neq 0\}$. Then $y_+ = y + \sum\limits_{j \in \sigma(d)} d^{(j)} \cdot Ae_j$.

Its complexity, $\kappa_A(d) \stackrel{\text{def}}{=} \sum\limits_{j \in \sigma(d)} p(Ae_j)$, can be VERY small!

If $\gamma(d) \leq c\gamma(A)$, $\gamma(A_j) \leq c\gamma(A)$, $\Rightarrow$ $\boxed{\kappa_A(d) \leq c^2 \cdot \gamma^2(A) \cdot MN}$

Expected acceleration: $(10^{-6})^2$

Sparse updating strategy

Main idea

- After update $x_+ = x + d$ we have $y_+ \stackrel{\text{def}}{=} Ax_+ = \underbrace{Ax}_{y} + Ad$.
- What happens if d is *sparse*?

Denote $\sigma(d) = \{j : d^{(j)} \neq 0\}$. Then $y_+ = y + \sum\limits_{j \in \sigma(d)} d^{(j)} \cdot Ae_j$.

Its complexity, $\kappa_A(d) \stackrel{\text{def}}{=} \sum\limits_{j \in \sigma(d)} p(Ae_j)$, can be VERY small!

If $\gamma(d) \leq c\gamma(A)$, $\gamma(A_j) \leq c\gamma(A)$, $\Rightarrow$ $\boxed{\kappa_A(d) \leq c^2 \cdot \gamma^2(A) \cdot MN}$

Expected acceleration: $(10^{-6})^2 \Rightarrow 1\text{sec}$

Sparse updating strategy

Main idea

- After update $x_+ = x + d$ we have $y_+ \stackrel{\text{def}}{=} Ax_+ = \underbrace{Ax}_{y} + Ad$.
- What happens if d is *sparse*?

Denote $\sigma(d) = \{j : d^{(j)} \neq 0\}$. Then $y_+ = y + \sum\limits_{j \in \sigma(d)} d^{(j)} \cdot Ae_j$.

Its complexity, $\kappa_A(d) \stackrel{\text{def}}{=} \sum\limits_{j \in \sigma(d)} p(Ae_j)$, can be VERY small!

If $\gamma(d) \leq c\gamma(A)$, $\gamma(A_j) \leq c\gamma(A)$, $\Rightarrow$ $\boxed{\kappa_A(d) \leq c^2 \cdot \gamma^2(A) \cdot MN}$

Expected acceleration: $(10^{-6})^2 \Rightarrow 1\text{sec} \approx 32000$ years!

When it can work? Problem: $\min_x f(x)$.

When it can work? Problem: $\min_x f(x)$.

1. Simple methods: $x_+ = x - hf'(x)$.

When it can work? Problem: $\min\limits_x f(x)$.

1. Simple methods: $x_+ = x - hf'(x)$. Full-vector operations?

When it can work? Problem: $\min_x f(x)$.

1. Simple methods: $x_+ = x - hf'(x)$. Full-vector operations?
2. Simple problems:

When it can work? Problem: $\min_x f(x)$.

1. Simple methods: $x_+ = x - hf'(x)$. Full-vector operations?
2. Simple problems: Functions with *sparse* gradients.

When it can work? Problem: $\min\limits_x f(x)$.

1. Simple methods: $x_+ = x - hf'(x)$. Full-vector operations?

2. Simple problems: Functions with *sparse* gradients.

Example: Piece-wise linear function

$$f(x) = \max_{1 \leq i \leq m} [\langle a_i, x \rangle - b^{(i)}].$$

When it can work? Problem: $\min\limits_x f(x)$.

1. Simple methods: $x_+ = x - hf'(x)$. Full-vector operations?

2. Simple problems: Functions with *sparse* gradients.

Example: Piece-wise linear function

$$f(x) = \max_{1 \le i \le m} [\langle a_i, x \rangle - b^{(i)}].$$

Its *subgradient* $f'(x) = a_{i(x)}$, $i(x):\ f(x) = \langle a_{i(x)}, x \rangle - b^{(i(x))}$,

can be sparse if a_i are sparse!

When it can work? Problem: $\min_x f(x)$.

1. Simple methods: $x_+ = x - hf'(x)$. Full-vector operations?

2. Simple problems: Functions with *sparse* gradients.

Example: Piece-wise linear function

$$f(x) = \max_{1 \leq i \leq m} [\langle a_i, x \rangle - b^{(i)}].$$

Its *subgradient* $f'(x) = a_{i(x)}, \;\; i(x) : \;\; f(x) = \langle a_{i(x)}, x \rangle - b^{(i(x))}$,

can be sparse if a_i are sparse!

But: Do we have a fast procedure for updating *max-type operations*?

When it can work? Problem: $\min_x f(x)$.

1. Simple methods: $x_+ = x - hf'(x)$. Full-vector operations?

2. Simple problems: Functions with *sparse* gradients.

Example: Piece-wise linear function

$$f(x) = \max_{1 \le i \le m} [\langle a_i, x \rangle - b^{(i)}].$$

Its *subgradient* $f'(x) = a_{i(x)}, \;\; i(x) : \;\; f(x) = \langle a_{i(x)}, x \rangle - b^{(i(x))}$,

can be sparse if a_i are sparse!

But: Do we have a fast procedure for updating *max-type operations*?

YES: change of one entry needs only $\log_2 m$ a.o.

When it can work? Problem: $\min_x f(x)$.

1. Simple methods: $x_+ = x - hf'(x)$. Full-vector operations?

2. Simple problems: Functions with *sparse* gradients.

Example: Piece-wise linear function

$$f(x) = \max_{1 \leq i \leq m} [\langle a_i, x \rangle - b^{(i)}].$$

Its *subgradient* $f'(x) = a_{i(x)}, \;\; i(x) : \;\; f(x) = \langle a_{i(x)}, x \rangle - b^{(i(x))}$,

can be sparse if a_i are sparse!

But: Do we have a fast procedure for updating *max-type operations*?

YES: change of one entry needs only $\log_2 m$ a.o.

Big change in modelling: Max-type functions.

Google problem: rank the agents by social weights

Google problem: rank the agents by social weights

Known: σ_i, the set of friends of agent i.

Google problem: rank the agents by social weights

Known: σ_i, the set of friends of agent i.
Unknown: $x_i \geq 0$, the social influence of agent $i = 1, \dots, N$.

Google problem: rank the agents by social weights

Known: σ_i, the set of friends of agent i.
Unknown: $x_i \geq 0$, the social influence of agent $i = 1, \ldots, N$.

Hypothesis:

- Agent i shares his support among all friends by equal parts.

Google problem: rank the agents by social weights

Known: σ_i, the set of friends of agent i.
Unknown: $x_i \geq 0$, the social influence of agent $i = 1, \dots, N$.

Hypothesis:

- Agent i shares his support among all friends by equal parts.
- The influence of agent i is equal to the total support obtained from his friends.

Google problem: rank the agents by social weights

Known: σ_i, the set of friends of agent i.
Unknown: $x_i \geq 0$, the social influence of agent $i = 1, \ldots, N$.

Hypothesis:

- Agent i shares his support among all friends by equal parts.
- The influence of agent i is equal to the total support obtained from his friends.

Let $E \in R^{N \times N}$ be an incidence matrix of the connections graph.

Google problem: rank the agents by social weights

Known: σ_i, the set of friends of agent i.
Unknown: $x_i \geq 0$, the social influence of agent $i = 1, \ldots, N$.

Hypothesis:

- Agent i shares his support among all friends by equal parts.
- The influence of agent i is equal to the total support obtained from his friends.

Let $E \in R^{N \times N}$ be an incidence matrix of the connections graph.
Denote $e = (1, \ldots, 1)^T \in R^N$ and $\bar{E} = E \cdot \mathrm{diag}\,(E^T e)^{-1}$.

Google problem: rank the agents by social weights

Known: σ_i, the set of friends of agent i.
Unknown: $x_i \geq 0$, the social influence of agent $i = 1, \ldots, N$.

Hypothesis:

- Agent i shares his support among all friends by equal parts.
- The influence of agent i is equal to the total support obtained from his friends.

Let $E \in R^{N \times N}$ be an incidence matrix of the connections graph.
Denote $e = (1, \ldots, 1)^T \in R^N$ and $\bar{E} = E \cdot \mathrm{diag}\,(E^T e)^{-1}$.
This matrix is stochastic.

Google problem: rank the agents by social weights

Known: σ_i, the set of friends of agent i.
Unknown: $x_i \geq 0$, the social influence of agent $i = 1, \ldots, N$.

Hypothesis:

- Agent i shares his support among all friends by equal parts.
- The influence of agent i is equal to the total support obtained from his friends.

Let $E \in R^{N \times N}$ be an incidence matrix of the connections graph.
Denote $e = (1, \ldots, 1)^T \in R^N$ and $\bar{E} = E \cdot \mathrm{diag}\,(E^T e)^{-1}$.
This matrix is stochastic.

Problem: Find $x^* \geq 0 : \quad \bar{E} x^* = x^*, \quad x^* \neq 0.$

Google problem: rank the agents by social weights

Known: σ_i, the set of friends of agent i.
Unknown: $x_i \geq 0$, the social influence of agent $i = 1, \ldots, N$.

Hypothesis:

- Agent i shares his support among all friends by equal parts.
- The influence of agent i is equal to the total support obtained from his friends.

Let $E \in R^{N \times N}$ be an incidence matrix of the connections graph.
Denote $e = (1, \ldots, 1)^T \in R^N$ and $\bar{E} = E \cdot \mathrm{diag}\,(E^T e)^{-1}$.
This matrix is stochastic.

Problem: Find $x^* \geq 0 : \quad \bar{E} x^* = x^*, \quad x^* \neq 0$.

$$f(x) = \max_{1 \leq i \leq N} [\langle e_i, \bar{E} x \rangle - x^{(i)}] \quad \to \quad \min_{x \geq 0}.$$

Computational experiments: Iteration Cost

Computational experiments: Iteration Cost

Compare GM with sparse update (GM_s) with the standard one.

Computational experiments: Iteration Cost

Compare GM with sparse update (GM_s) with the standard one.

Problem: matrix with p nonzero diagonals.

Thus, $\kappa(A) \stackrel{\text{def}}{=} \max\limits_{1 \leq i \leq M} \kappa_A(A^T e_i) = p^2$.

Computational experiments: Iteration Cost

Compare GM with sparse update (GM_s) with the standard one.

Problem: matrix with p nonzero diagonals.

Thus, $\kappa(A) \stackrel{\text{def}}{=} \max\limits_{1 \leq i \leq M} \kappa_A(A^T e_i) = p^2$.

Iteration Cost: $GM_s \leq \kappa(A) \log_2 N$

Computational experiments: Iteration Cost

Compare GM with sparse update (GM_s) with the standard one.

Problem: matrix with p nonzero diagonals.

Thus, $\kappa(A) \stackrel{\mathrm{def}}{=} \max\limits_{1 \le i \le M} \kappa_A(A^T e_i) = p^2$.

Iteration Cost: $GM_s \le \kappa(A) \log_2 N \approx p^2 \log_2 N$,

Computational experiments: Iteration Cost

Compare GM with sparse update (GM_s) with the standard one.

Problem: matrix with p nonzero diagonals.

Thus, $\kappa(A) \stackrel{\text{def}}{=} \max\limits_{1 \le i \le M} \kappa_A(A^T e_i) = p^2$.

Iteration Cost: $GM_s \le \kappa(A) \log_2 N \approx p^2 \log_2 N, \quad GM \approx pN$.

Computational experiments: Iteration Cost

Compare GM with sparse update (GM_s) with the standard one.

Problem: matrix with p nonzero diagonals.

Thus, $\kappa(A) \stackrel{\text{def}}{=} \max\limits_{1 \leq i \leq M} \kappa_A(A^T e_i) = p^2$.

Iteration Cost: $GM_s \leq \kappa(A) \log_2 N \approx p^2 \log_2 N, \quad GM \approx pN$.

$(\log_2 \; 10^3 = 10,$

Computational experiments: Iteration Cost

Compare GM with sparse update (GM_s) with the standard one.

Problem: matrix with p nonzero diagonals.

Thus, $\kappa(A) \stackrel{\mathrm{def}}{=} \max\limits_{1 \leq i \leq M} \kappa_A(A^T e_i) = p^2$.

Iteration Cost: $GM_s \leq \kappa(A) \log_2 N \approx p^2 \log_2 N, \quad GM \approx pN.$

$(\log_2\ 10^3 = 10, \quad \log_2\ 10^6 = 20,$

Computational experiments: Iteration Cost

Compare GM with sparse update (GM_s) with the standard one.

Problem: matrix with p nonzero diagonals.

Thus, $\kappa(A) \stackrel{\mathrm{def}}{=} \max\limits_{1 \leq i \leq M} \kappa_A(A^T e_i) = p^2$.

Iteration Cost: $GM_s \leq \kappa(A) \log_2 N \approx p^2 \log_2 N, \quad GM \approx pN.$

$$(\log_2 \ 10^3 = 10, \quad \log_2 \ 10^6 = 20, \quad \log_2 \ 10^9 = 30)$$

Computational experiments: Iteration Cost

Compare GM with sparse update (GM_s) with the standard one.

Problem: matrix with p nonzero diagonals.

Thus, $\kappa(A) \stackrel{\mathrm{def}}{=} \max\limits_{1 \le i \le M} \kappa_A(A^T e_i) = p^2$.

Iteration Cost: $GM_s \le \kappa(A) \log_2 N \approx p^2 \log_2 N, \quad GM \approx pN$.

$(\log_2 \ 10^3 = 10, \quad \log_2 \ 10^6 = 20, \quad \log_2 \ 10^9 = 30)$

Time for 10^4 iter. ($p = 32$)

N	$\kappa(A)$	GM_s	GM
1024	1632	3.00	2.98
2048	1792	3.36	6.41
4096	1888	3.75	15.11
8192	1920	4.20	139.92
16384	1824	4.69	408.38

Computational experiments: Iteration Cost

Compare GM with sparse update (GM_s) with the standard one.

Problem: matrix with p nonzero diagonals.

Thus, $\kappa(A) \stackrel{\text{def}}{=} \max_{1 \le i \le M} \kappa_A(A^T e_i) = p^2$.

Iteration Cost: $GM_s \le \kappa(A) \log_2 N \approx p^2 \log_2 N, \quad GM \approx pN$.

$(\log_2\ 10^3 = 10, \quad \log_2\ 10^6 = 20, \quad \log_2\ 10^9 = 30)$

Time for 10^4 iter. ($p = 32$)

N	$\kappa(A)$	GM_s	GM
1024	1632	3.00	2.98
2048	1792	3.36	6.41
4096	1888	3.75	15.11
8192	1920	4.20	139.92
16384	1824	4.69	408.38

Time for 10^3 iter. ($p = 16$)

N	$\kappa(A)$	GM_s	GM
131072	576	0.19	213.9
262144	592	0.25	477.8
524288	592	0.32	1095.5
1048576	608	0.40	2590.8

Computational experiments: Iteration Cost

Compare GM with sparse update (GM_s) with the standard one.

Problem: matrix with p nonzero diagonals.

Thus, $\kappa(A) \stackrel{\text{def}}{=} \max\limits_{1 \le i \le M} \kappa_A(A^T e_i) = p^2$.

Iteration Cost: $GM_s \le \kappa(A) \log_2 N \approx p^2 \log_2 N, \quad GM \approx pN$.

$(\log_2 \ 10^3 = 10, \quad \log_2 \ 10^6 = 20, \quad \log_2 \ 10^9 = 30)$

Time for 10^4 iter. ($p = 32$)

N	$\kappa(A)$	GM_s	GM
1024	1632	3.00	2.98
2048	1792	3.36	6.41
4096	1888	3.75	15.11
8192	1920	4.20	139.92
16384	1824	4.69	408.38

Time for 10^3 iter. ($p = 16$)

N	$\kappa(A)$	GM_s	GM
131072	576	0.19	213.9
262144	592	0.25	477.8
524288	592	0.32	1095.5
1048576	608	0.40	2590.8

1 sec $\approx$ 100 min!

Optimal design of mechanical structures

Optimal design of mechanical structures

Problem: $$\max_{y \in R^m} \left\{ \langle b, y \rangle : \; g(y) \stackrel{\text{def}}{=} \max_{1 \le j \le n} g_j(y) \; \le \; 1 \right\},$$

Optimal design of mechanical structures

Problem: $$\max_{y \in R^m} \left\{ \langle b, y \rangle : \; g(y) \stackrel{\text{def}}{=} \max_{1 \leq j \leq n} g_j(y) \; \leq \; 1 \right\},$$

where $y^{(i)}$ are displacements of nodes, $i = 1, \ldots, m$,

Optimal design of mechanical structures

Problem: $$\max_{y \in R^m} \left\{ \langle b, y \rangle : \; g(y) \stackrel{\text{def}}{=} \max_{1 \le j \le n} g_j(y) \; \le \; 1 \right\},$$

where $y^{(i)}$ are displacements of nodes, $i = 1, \ldots, m$,

Optimal design of mechanical structures

Problem: $$\max_{y \in R^m} \left\{ \langle b, y \rangle : \; g(y) \stackrel{\text{def}}{=} \max_{1 \le j \le n} g_j(y) \; \le \; 1 \right\},$$

where $y^{(i)}$ are displacements of nodes, $i = 1, \dots, m$,

- b is the external force,

Optimal design of mechanical structures

Problem: $$\max_{y \in R^m} \left\{ \langle b, y \rangle : \; g(y) \stackrel{\text{def}}{=} \max_{1 \le j \le n} g_j(y) \; \le \; 1 \right\},$$

where $y^{(i)}$ are displacements of nodes, $i = 1, \dots, m$,

- b is the external force,
- $g_j(y)$ describes the tension in the design element j (connecting some nodes).

Optimal design of mechanical structures

Problem: $\max\limits_{y \in R^m} \left\{ \langle b, y \rangle : \; g(y) \stackrel{\mathrm{def}}{=} \max\limits_{1 \leq j \leq n} g_j(y) \; \leq \; 1 \right\}$,

where $y^{(i)}$ are displacements of nodes, $i = 1, \dots, m$,

- b is the external force,
- $g_j(y)$ describes the tension in the design element j (connecting some nodes).

Method: Choose $h > 0$. Define $y_0 = 0$. For $k \geq 0$ do:

if $g(y_k) \leq 1 + h$, **then:** $y_{k+1} = y_k + hb$,

else: $y_{k+1} = y_k - h g'_{j_k}(y_k)$.

Optimal design of mechanical structures

Problem: $\max\limits_{y \in R^m} \left\{ \langle b, y \rangle : \; g(y) \stackrel{\text{def}}{=} \max\limits_{1 \leq j \leq n} g_j(y) \; \leq \; 1 \right\},$

where $y^{(i)}$ are displacements of nodes, $i = 1, \ldots, m$,

- b is the external force,
- $g_j(y)$ describes the tension in the design element j (connecting some nodes).

Method: Choose $h > 0$. Define $y_0 = 0$. For $k \geq 0$ do:

if $g(y_k) \leq 1 + h$, **then:** $\quad y_{k+1} = y_k + hb$,

else: $\quad y_{k+1} = y_k - hg'_{j_k}(y_k)$.

Size of the element $\approx$ the number of times it was the worst.

Optimal design of mechanical structures

Problem: $$\max_{y \in R^m} \left\{ \langle b, y \rangle : \; g(y) \stackrel{\mathrm{def}}{=} \max_{1 \le j \le n} g_j(y) \; \le \; 1 \right\},$$

where $y^{(i)}$ are displacements of nodes, $i = 1, \dots, m$,

- b is the external force,
- $g_j(y)$ describes the tension in the design element j (connecting some nodes).

Method: Choose $h > 0$. Define $y_0 = 0$. For $k \ge 0$ do:

if $g(y_k) \le 1 + h$, **then:** $y_{k+1} = y_k + hb$,

else: $y_{k+1} = y_k - h g'_{j_k}(y_k)$.

Size of the element $\approx$ the number of times it was the worst.

Interpretation: The natural "growth" of construction under an action of increasing force.

Optimal design of mechanical structures

Problem: $\max\limits_{y \in R^m} \left\{ \langle b, y \rangle : \; g(y) \stackrel{\mathrm{def}}{=} \max\limits_{1 \le j \le n} g_j(y) \; \le \; 1 \right\}$,

where $y^{(i)}$ are displacements of nodes, $i = 1, \dots, m$,

- b is the external force,
- $g_j(y)$ describes the tension in the design element j (connecting some nodes).

Method: Choose $h > 0$. Define $y_0 = 0$. For $k \ge 0$ do:

$$\textbf{if } g(y_k) \le 1 + h, \textbf{ then: } \quad y_{k+1} = y_k + hb,$$
$$\textbf{else: } \quad y_{k+1} = y_k - h g'_{j_k}(y_k).$$

Size of the element $\approx$ the number of times it was the worst.

Interpretation: The natural "growth" of construction under an action of increasing force. **NB:** Full theoretical justification!

Modelling congested traffic

Modelling congested traffic

Data: network $\mathcal{R} = (\mathcal{N}, \mathcal{E})$, $\quad n = |\mathcal{N}|$, $\quad m = |\mathcal{E}| \approx 3n$.

Modelling congested traffic

Data: network $\mathcal{R} = (\mathcal{N}, \mathcal{E}), \quad n = |\mathcal{N}|, \quad m = |\mathcal{E}| \approx 3n.$

- Performance: time $t_\alpha \geq \bar{t}_\alpha$, flow $f_\alpha \in [0, \bar{f}_\alpha]$, $\alpha \in \mathcal{E}$.

Modelling congested traffic

Data: network $\mathcal{R} = (\mathcal{N}, \mathcal{E})$, $\quad n = |\mathcal{N}|$, $\quad m = |\mathcal{E}| \approx 3n$.

- Performance: time $t_\alpha \geq \bar{t}_\alpha$, flow $f_\alpha \in [0, \bar{f}_\alpha]$, $\alpha \in \mathcal{E}$.
- Congestion: $(t_\alpha - \bar{t}_\alpha)(\bar{f}_\alpha - f_\alpha) = 0$, $\alpha \in \mathcal{E}$.

Modelling congested traffic

Data: network $\mathcal{R} = (\mathcal{N}, \mathcal{E}), \quad n = |\mathcal{N}|, \quad m = |\mathcal{E}| \approx 3n.$

- Performance: time $t_\alpha \geq \bar{t}_\alpha$, flow $f_\alpha \in [0, \bar{f}_\alpha]$, $\alpha \in \mathcal{E}$.
- Congestion: $(t_\alpha - \bar{t}_\alpha)(\bar{f}_\alpha - f_\alpha) = 0$, $\alpha \in \mathcal{E}$.
- OD-flow: $d_{(i,j)} > 0$, $(i, j) \in \mathcal{OD} \subseteq \mathcal{N} \times \mathcal{N}$.

Modelling congested traffic

Data: network $\mathcal{R} = (\mathcal{N}, \mathcal{E})$, $\quad n = |\mathcal{N}|$, $\quad m = |\mathcal{E}| \approx 3n$.

- Performance: time $t_\alpha \geq \bar{t}_\alpha$, flow $f_\alpha \in [0, \bar{f}_\alpha]$, $\alpha \in \mathcal{E}$.
- Congestion: $(t_\alpha - \bar{t}_\alpha)(\bar{f}_\alpha - f_\alpha) = 0$, $\alpha \in \mathcal{E}$.
- OD-flow: $d_{(i,j)} > 0$, $(i,j) \in \mathcal{OD} \subseteq \mathcal{N} \times \mathcal{N}$.

Problem: $\max\limits_{t \geq \bar{t}} \{ \sum\limits_{(i,j) \in \mathcal{OD}} d_{(i,j)} SP_{i,j}(t) - \langle \bar{f}, t \rangle \}$,

Modelling congested traffic

Data: network $\mathcal{R} = (\mathcal{N}, \mathcal{E})$, $\quad n = |\mathcal{N}|$, $\quad m = |\mathcal{E}| \approx 3n$.

- Performance: time $t_\alpha \geq \bar{t}_\alpha$, flow $f_\alpha \in [0, \bar{f}_\alpha]$, $\alpha \in \mathcal{E}$.
- Congestion: $(t_\alpha - \bar{t}_\alpha)(\bar{f}_\alpha - f_\alpha) = 0$, $\alpha \in \mathcal{E}$.
- OD-flow: $d_{(i,j)} > 0$, $(i,j) \in \mathcal{OD} \subseteq \mathcal{N} \times \mathcal{N}$.

Problem: $\quad \max\limits_{t \geq \bar{t}} \{ \sum\limits_{(i,j) \in \mathcal{OD}} d_{(i,j)} SP_{i,j}(t) - \langle \bar{f}, t \rangle \}$, where

$SP_{i,j}(t)$ is the *shortest path function*: $SP_{i,j}(t) = \min\limits_{r \in \mathcal{R}_{i,j}} \sum\limits_{\alpha \in r} t_\alpha$.

Modelling congested traffic

Data: network $\mathcal{R} = (\mathcal{N}, \mathcal{E})$, $\quad n = |\mathcal{N}|$, $\quad m = |\mathcal{E}| \approx 3n$.

- Performance: time $t_\alpha \geq \bar{t}_\alpha$, flow $f_\alpha \in [0, \bar{f}_\alpha]$, $\alpha \in \mathcal{E}$.
- Congestion: $(t_\alpha - \bar{t}_\alpha)(\bar{f}_\alpha - f_\alpha) = 0$, $\alpha \in \mathcal{E}$.
- OD-flow: $d_{(i,j)} > 0$, $(i,j) \in \mathcal{OD} \subseteq \mathcal{N} \times \mathcal{N}$.

Problem: $\max_{t \geq \bar{t}} \{ \sum_{(i,j) \in \mathcal{OD}} d_{(i,j)} SP_{i,j}(t) - \langle \bar{f}, t \rangle \}$, where

$SP_{i,j}(t)$ is the *shortest path function*: $SP_{i,j}(t) = \min_{r \in \mathcal{R}_{i,j}} \sum_{\alpha \in r} t_\alpha$.

It is concave, nondifferentiable, piecewise linear, etc.

Modelling congested traffic

Data: network $\mathcal{R} = (\mathcal{N}, \mathcal{E})$, $\quad n = |\mathcal{N}|$, $\quad m = |\mathcal{E}| \approx 3n$.

- Performance: time $t_\alpha \geq \bar{t}_\alpha$, flow $f_\alpha \in [0, \bar{f}_\alpha]$, $\alpha \in \mathcal{E}$.
- Congestion: $(t_\alpha - \bar{t}_\alpha)(\bar{f}_\alpha - f_\alpha) = 0$, $\alpha \in \mathcal{E}$.
- OD-flow: $d_{(i,j)} > 0$, $(i,j) \in \mathcal{OD} \subseteq \mathcal{N} \times \mathcal{N}$.

Problem: $\max\limits_{t \geq \bar{t}} \{ \sum\limits_{(i,j) \in \mathcal{OD}} d_{(i,j)} SP_{i,j}(t) - \langle \bar{f}, t \rangle \}$, where

$SP_{i,j}(t)$ is the *shortest path function*: $SP_{i,j}(t) = \min\limits_{r \in \mathcal{R}_{i,j}} \sum\limits_{\alpha \in r} t_\alpha$.

It is concave, nondifferentiable, piecewise linear, etc.

Solution: equilibrium travel time t^*_α, $\alpha \in \mathcal{E}$,

Modelling congested traffic

Data: network $\mathcal{R} = (\mathcal{N}, \mathcal{E})$, $\quad n = |\mathcal{N}|$, $\quad m = |\mathcal{E}| \approx 3n$.

- Performance: time $t_\alpha \geq \bar{t}_\alpha$, flow $f_\alpha \in [0, \bar{f}_\alpha]$, $\alpha \in \mathcal{E}$.
- Congestion: $(t_\alpha - \bar{t}_\alpha)(\bar{f}_\alpha - f_\alpha) = 0$, $\alpha \in \mathcal{E}$.
- OD-flow: $d_{(i,j)} > 0$, $(i,j) \in \mathcal{OD} \subseteq \mathcal{N} \times \mathcal{N}$.

Problem: $\quad \max\limits_{t \geq \bar{t}} \{ \sum\limits_{(i,j) \in \mathcal{OD}} d_{(i,j)} SP_{i,j}(t) - \langle \bar{f}, t \rangle \}$, where

$SP_{i,j}(t)$ is the *shortest path function*: $SP_{i,j}(t) = \min\limits_{r \in \mathcal{R}_{i,j}} \sum\limits_{\alpha \in r} t_\alpha$.

It is concave, nondifferentiable, piecewise linear, etc.

Solution: equilibrium travel time t^*_α, $\alpha \in \mathcal{E}$, AND

- equilibrium OD-flows $f^*_{i,j} \in R^m$, $(i,j) \in \mathcal{OD}$.

Modelling congested traffic

Data: network $\mathcal{R} = (\mathcal{N}, \mathcal{E})$, $\quad n = |\mathcal{N}|$, $\quad m = |\mathcal{E}| \approx 3n$.

- Performance: time $t_\alpha \geq \bar{t}_\alpha$, flow $f_\alpha \in [0, \bar{f}_\alpha]$, $\alpha \in \mathcal{E}$.
- Congestion: $(t_\alpha - \bar{t}_\alpha)(\bar{f}_\alpha - f_\alpha) = 0$, $\alpha \in \mathcal{E}$.
- OD-flow: $d_{(i,j)} > 0$, $(i,j) \in \mathcal{OD} \subseteq \mathcal{N} \times \mathcal{N}$.

Problem: $\max\limits_{t \geq \bar{t}} \{ \sum\limits_{(i,j) \in \mathcal{OD}} d_{(i,j)} SP_{i,j}(t) - \langle \bar{f}, t \rangle \}$, where

$SP_{i,j}(t)$ is the *shortest path function*: $SP_{i,j}(t) = \min\limits_{r \in \mathcal{R}_{i,j}} \sum\limits_{\alpha \in r} t_\alpha$.

It is concave, nondifferentiable, piecewise linear, etc.

Solution: equilibrium travel time t^*_α, $\alpha \in \mathcal{E}$, AND

- equilibrium OD-flows $f^*_{i,j} \in R^m$, $(i,j) \in \mathcal{OD}$.

Moscow: $m \approx 10^4$,

Modelling congested traffic

Data: network $\mathcal{R} = (\mathcal{N}, \mathcal{E})$, $\quad n = |\mathcal{N}|$, $\quad m = |\mathcal{E}| \approx 3n$.

- Performance: time $t_\alpha \geq \bar{t}_\alpha$, flow $f_\alpha \in [0, \bar{f}_\alpha]$, $\alpha \in \mathcal{E}$.
- Congestion: $(t_\alpha - \bar{t}_\alpha)(\bar{f}_\alpha - f_\alpha) = 0$, $\alpha \in \mathcal{E}$.
- OD-flow: $d_{(i,j)} > 0$, $(i,j) \in \mathcal{OD} \subseteq \mathcal{N} \times \mathcal{N}$.

Problem: $\max\limits_{t \geq \bar{t}} \{ \sum\limits_{(i,j) \in \mathcal{OD}} d_{(i,j)} SP_{i,j}(t) - \langle \bar{f}, t \rangle \}$, where

$SP_{i,j}(t)$ is the *shortest path function*: $SP_{i,j}(t) = \min\limits_{r \in \mathcal{R}_{i,j}} \sum\limits_{\alpha \in r} t_\alpha$.

It is concave, nondifferentiable, piecewise linear, etc.

Solution: equilibrium travel time t^*_α, $\alpha \in \mathcal{E}$, AND

- equilibrium OD-flows $f^*_{i,j} \in R^m$, $(i,j) \in \mathcal{OD}$.

Moscow: $m \approx 10^4$, $|\mathcal{OD}| \approx \frac{1}{2} 1000 \times 1000 = 5 \cdot 10^5$.

Modelling congested traffic

Data: network $\mathcal{R} = (\mathcal{N}, \mathcal{E})$, $\quad n = |\mathcal{N}|$, $\quad m = |\mathcal{E}| \approx 3n$.

- Performance: time $t_\alpha \geq \bar{t}_\alpha$, flow $f_\alpha \in [0, \bar{f}_\alpha]$, $\alpha \in \mathcal{E}$.
- Congestion: $(t_\alpha - \bar{t}_\alpha)(\bar{f}_\alpha - f_\alpha) = 0$, $\alpha \in \mathcal{E}$.
- OD-flow: $d_{(i,j)} > 0$, $(i,j) \in \mathcal{OD} \subseteq \mathcal{N} \times \mathcal{N}$.

Problem: $\max\limits_{t \geq \bar{t}} \{ \sum\limits_{(i,j) \in \mathcal{OD}} d_{(i,j)} SP_{i,j}(t) - \langle \bar{f}, t \rangle \}$, where

$SP_{i,j}(t)$ is the *shortest path function*: $SP_{i,j}(t) = \min\limits_{r \in \mathcal{R}_{i,j}} \sum\limits_{\alpha \in r} t_\alpha$.

It is concave, nondifferentiable, piecewise linear, etc.

Solution: equilibrium travel time t^*_α, $\alpha \in \mathcal{E}$, AND

- equilibrium OD-flows $f^*_{i,j} \in R^m$, $(i,j) \in \mathcal{OD}$.

Moscow: $m \approx 10^4$, $|\mathcal{OD}| \approx \frac{1}{2} 1000 \times 1000 = 5 \cdot 10^5$.

Challenges: Numerical methods,

Modelling congested traffic

Data: network $\mathcal{R} = (\mathcal{N}, \mathcal{E})$, $\quad n = |\mathcal{N}|$, $\quad m = |\mathcal{E}| \approx 3n$.

- Performance: time $t_\alpha \geq \bar{t}_\alpha$, flow $f_\alpha \in [0, \bar{f}_\alpha]$, $\alpha \in \mathcal{E}$.
- Congestion: $(t_\alpha - \bar{t}_\alpha)(\bar{f}_\alpha - f_\alpha) = 0$, $\alpha \in \mathcal{E}$.
- OD-flow: $d_{(i,j)} > 0$, $(i,j) \in \mathcal{OD} \subseteq \mathcal{N} \times \mathcal{N}$.

Problem: $\max\limits_{t \geq \bar{t}} \{ \sum\limits_{(i,j) \in \mathcal{OD}} d_{(i,j)} SP_{i,j}(t) - \langle \bar{f}, t \rangle \}$, where

$SP_{i,j}(t)$ is the *shortest path function*: $SP_{i,j}(t) = \min\limits_{r \in \mathcal{R}_{i,j}} \sum\limits_{\alpha \in r} t_\alpha$.

It is concave, nondifferentiable, piecewise linear, etc.

Solution: equilibrium travel time t^*_α, $\alpha \in \mathcal{E}$, AND

- equilibrium OD-flows $f^*_{i,j} \in R^m$, $(i,j) \in \mathcal{OD}$.

Moscow: $m \approx 10^4$, $|\mathcal{OD}| \approx \frac{1}{2} 1000 \times 1000 = 5 \cdot 10^5$.

Challenges: Numerical methods, Behavioral methods.