

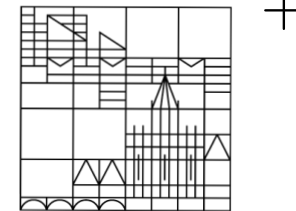
Approximating Pareto Sum via Bounded Monotone $(\min, +)$ -Convolution

SoCG 2026, New Brunswick, New Jersey

Geri Gokaj^{*}, Marvin Künnemann^{*}, Sabine Storandt⁺, Carina Truschel⁺



Universität
Konstanz



Pareto Sum

Pareto Sum

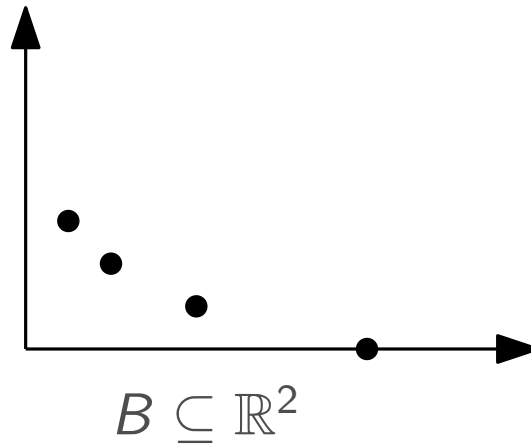
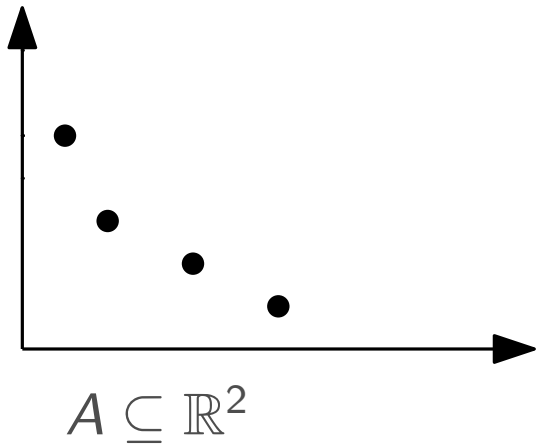
Given sets $A, B \subseteq \mathbb{R}^2$ of size n .

Pareto Sum

Given sets $A, B \subseteq \mathbb{R}^2$ of size n . Compute the **minima** of $A + B := \{a + b \mid a \in A, b \in B\}$

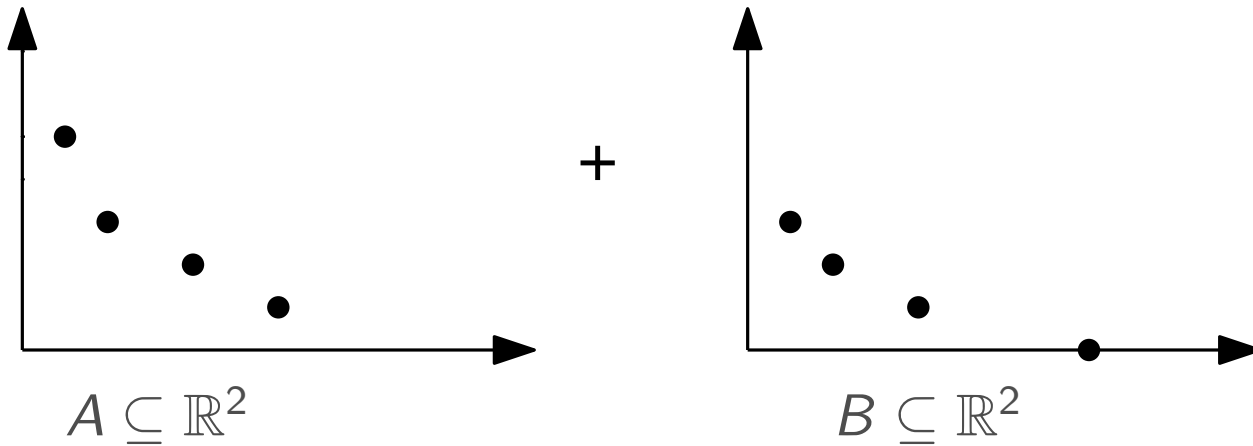
Pareto Sum

Given sets $A, B \subseteq \mathbb{R}^2$ of size n . Compute the **minima** of $A + B := \{a + b \mid a \in A, b \in B\}$



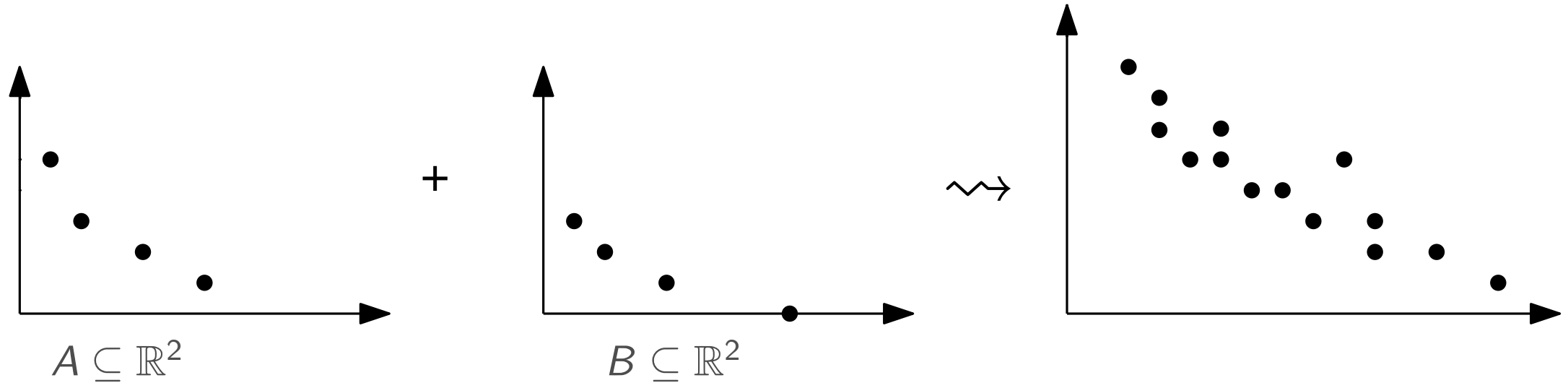
Pareto Sum

Given sets $A, B \subseteq \mathbb{R}^2$ of size n . Compute the **minima** of $A + B := \{a + b \mid a \in A, b \in B\}$



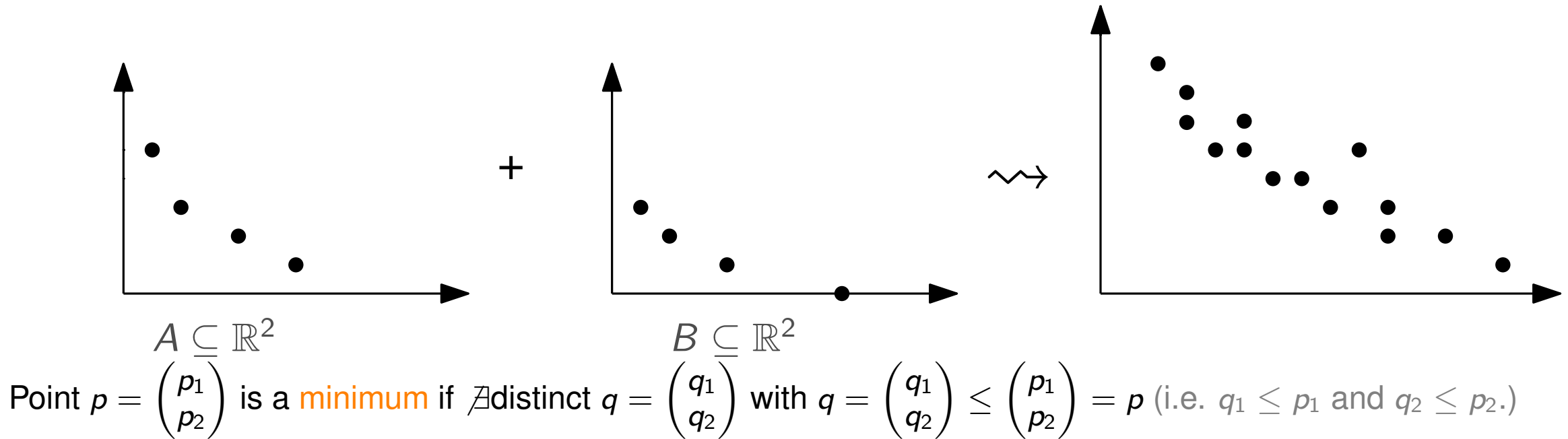
Pareto Sum

Given sets $A, B \subseteq \mathbb{R}^2$ of size n . Compute the **minima** of $A + B := \{a + b \mid a \in A, b \in B\}$



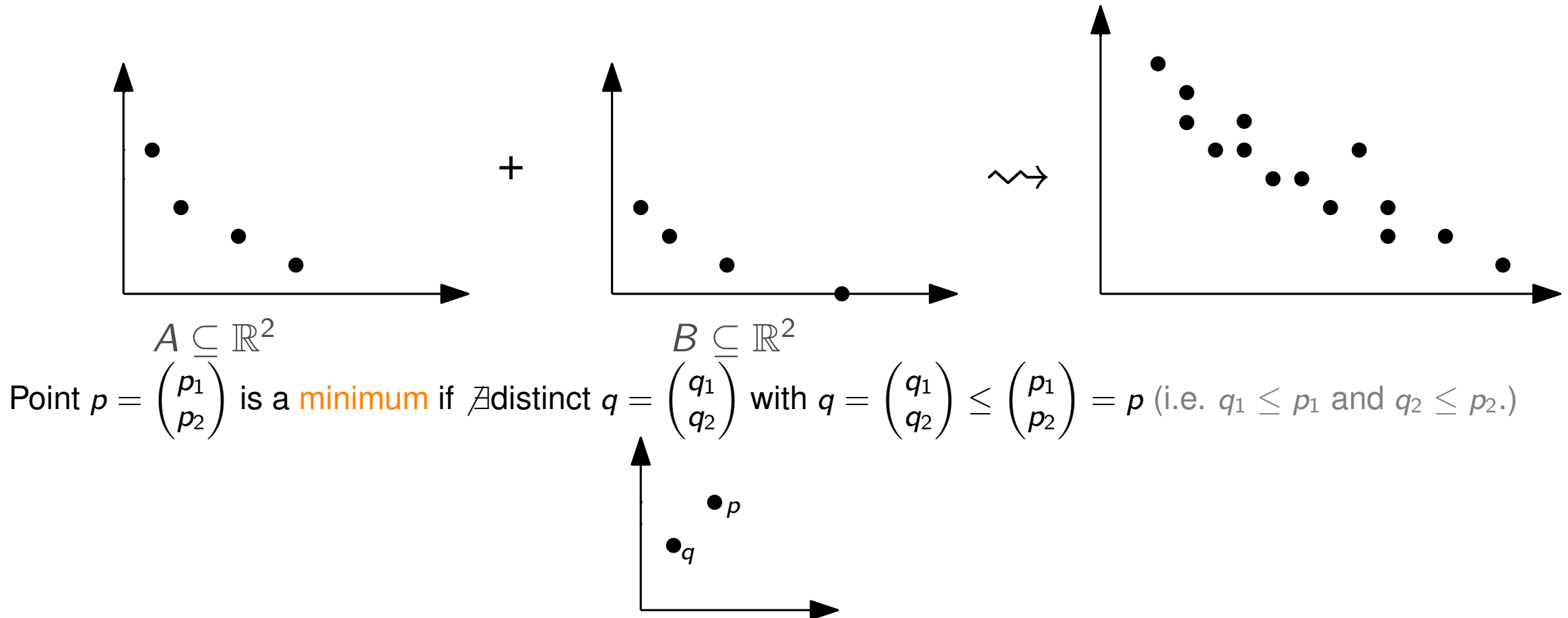
Pareto Sum

Given sets $A, B \subseteq \mathbb{R}^2$ of size n . Compute the **minima** of $A + B := \{a + b \mid a \in A, b \in B\}$



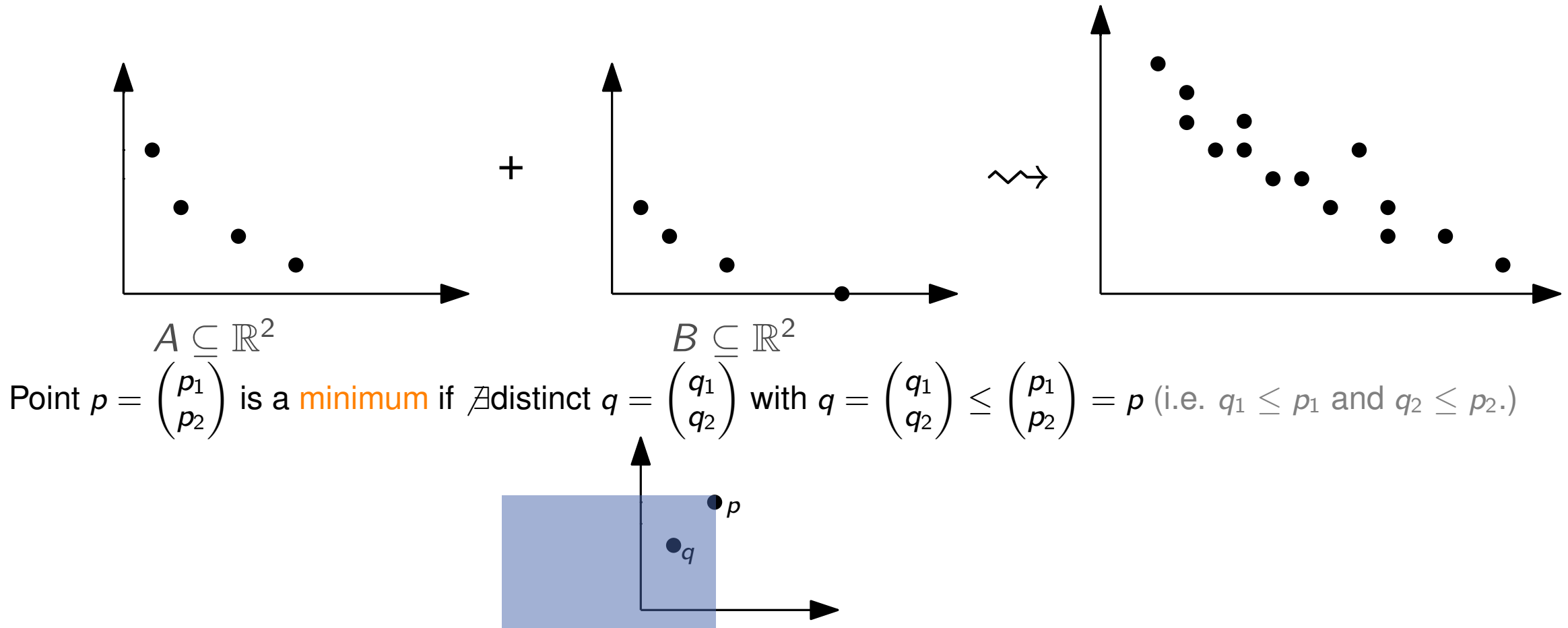
Pareto Sum

Given sets $A, B \subseteq \mathbb{R}^2$ of size n . Compute the **minima** of $A + B := \{a + b \mid a \in A, b \in B\}$



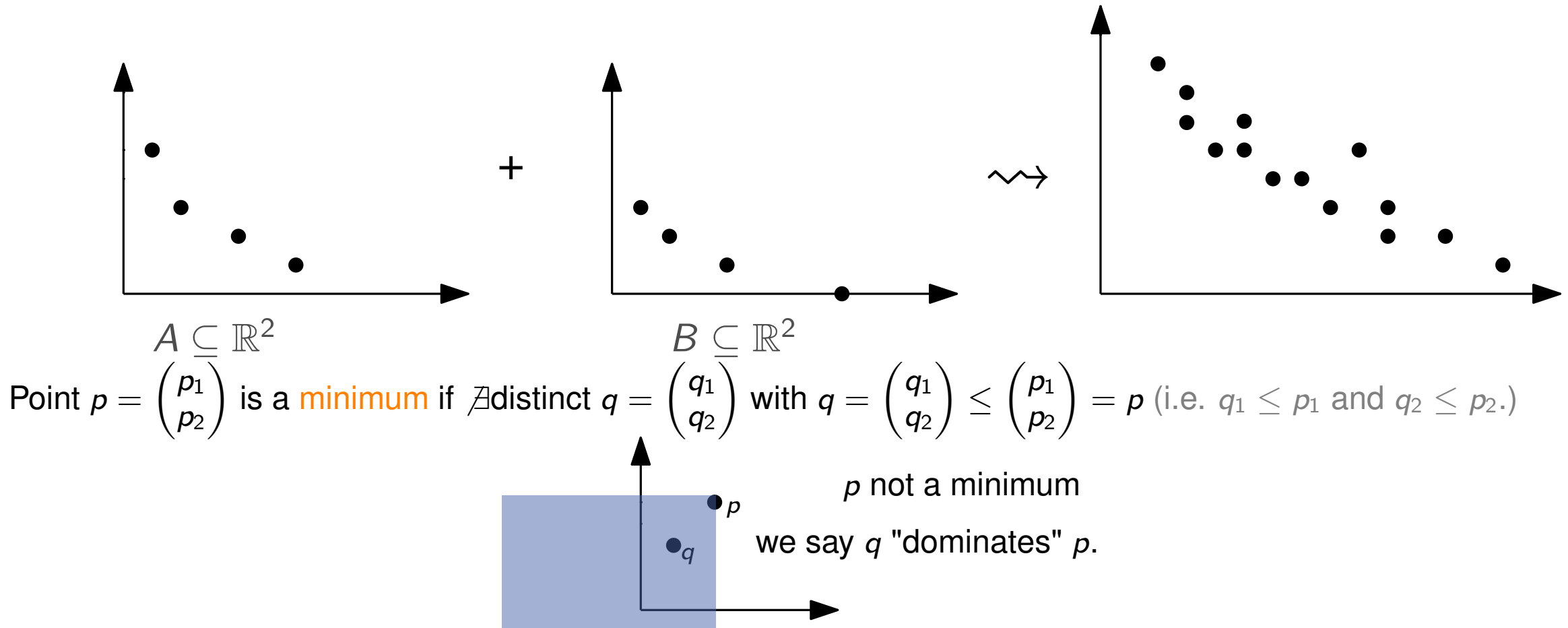
Pareto Sum

Given sets $A, B \subseteq \mathbb{R}^2$ of size n . Compute the **minima** of $A + B := \{a + b \mid a \in A, b \in B\}$



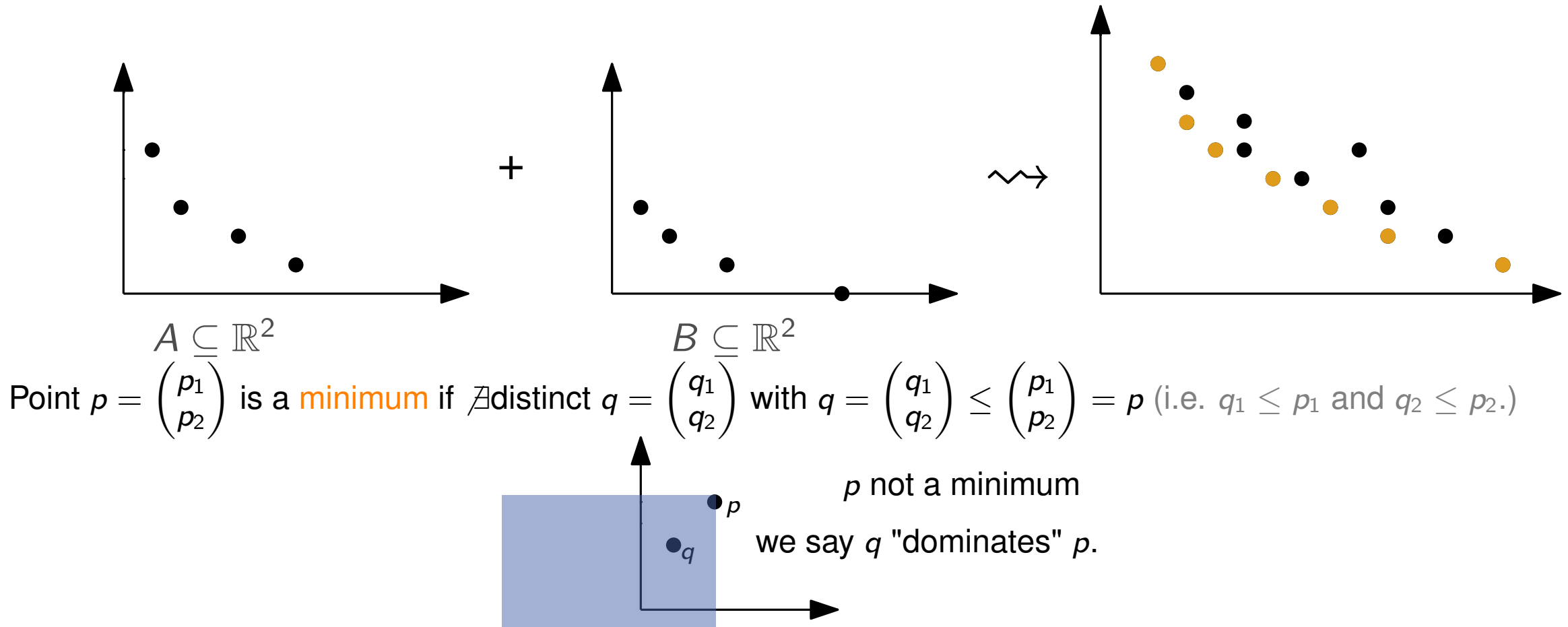
Pareto Sum

Given sets $A, B \subseteq \mathbb{R}^2$ of size n . Compute the **minima** of $A + B := \{a + b \mid a \in A, b \in B\}$



Pareto Sum

Given sets $A, B \subseteq \mathbb{R}^2$ of size n . Compute the **minima** of $A + B := \{a + b \mid a \in A, b \in B\}$



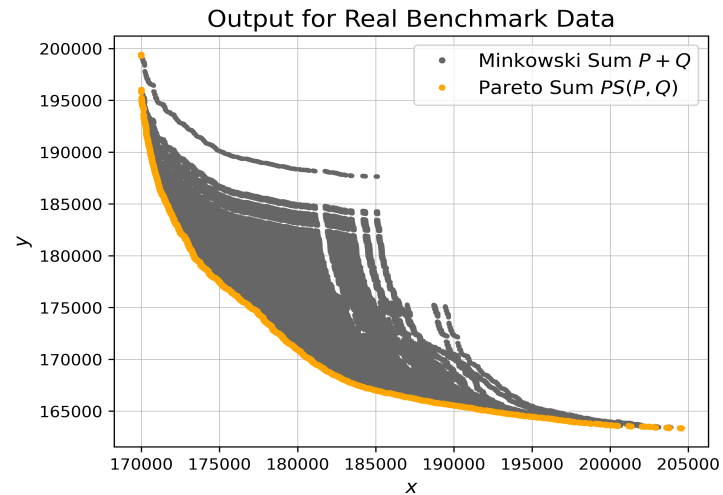
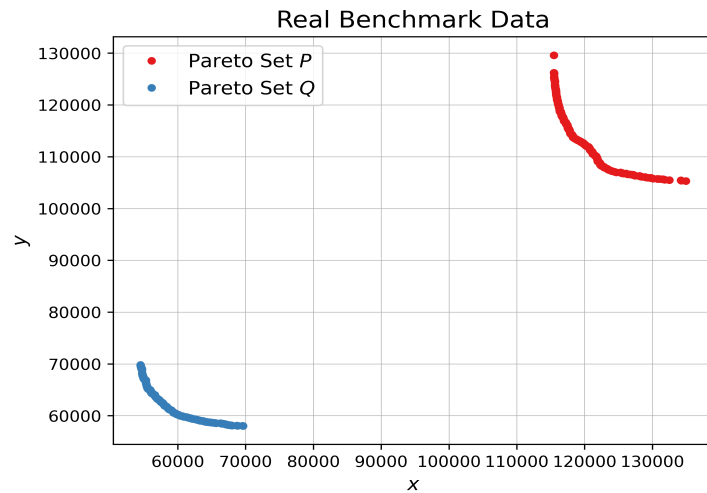
Previous work on Pareto Sum

Previous work on Pareto Sum

Problem stems from bicriteria-route planning.

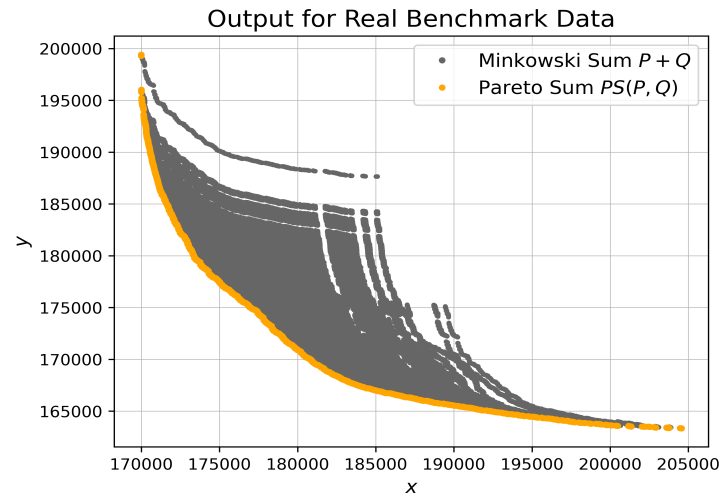
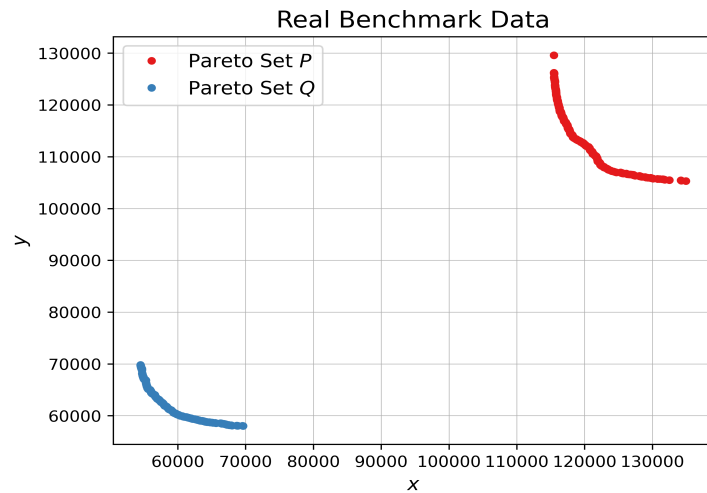
Previous work on Pareto Sum

Problem stems from bicriteria-route planning.



Previous work on Pareto Sum

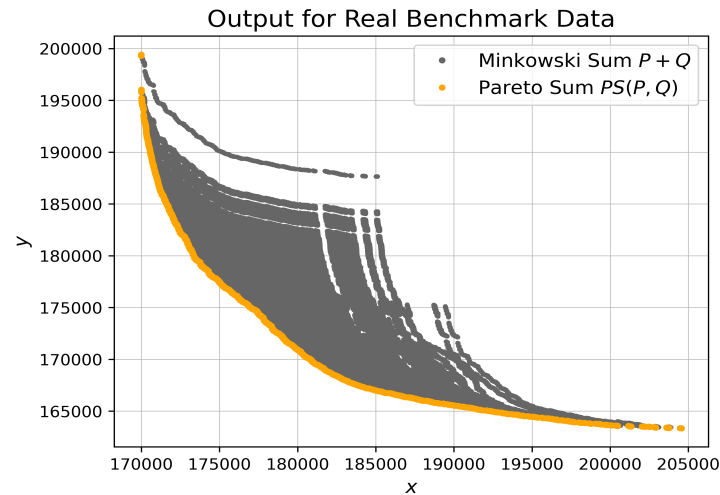
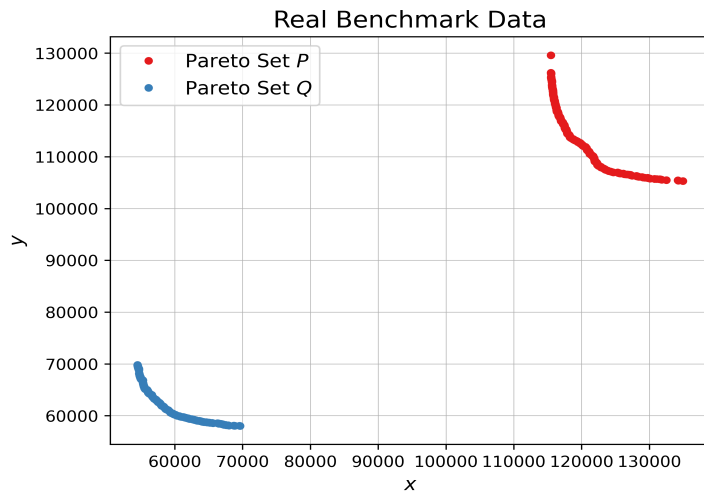
Problem stems from bicriteria-route planning.



Algorithm in time $\tilde{O}(\min(n^2, nk))$ exist [Funke, Hespe, Sanders, Storandt, Truschel;23]

Previous work on Pareto Sum

Problem stems from bicriteria-route planning.

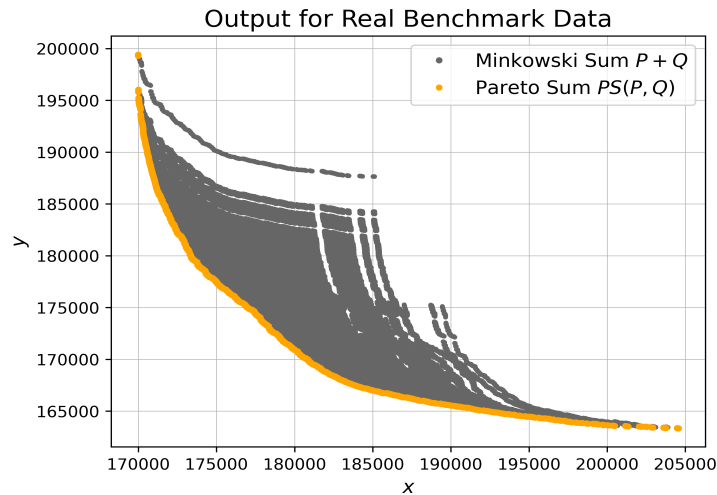
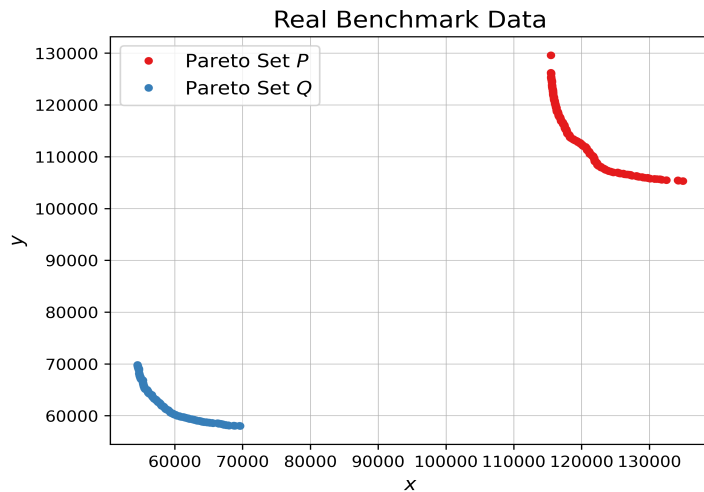


Algorithm in time $\tilde{O}(\min(n^2, nk))$ exist [Funke, Hespe, Sanders, Storandt, Truschel;23]

$\rightsquigarrow$ Conditionally optimal under 3-SUM hypothesis [Gokaj, Künnemann, Storandt, Truschel;25]

Previous work on Pareto Sum

Problem stems from bicriteria-route planning.



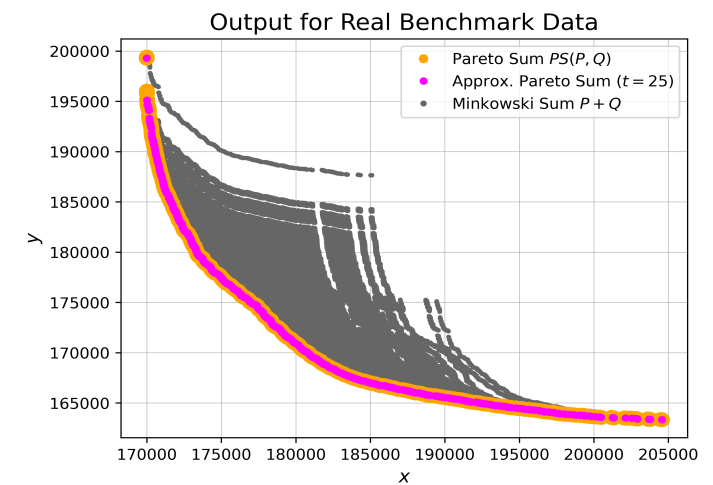
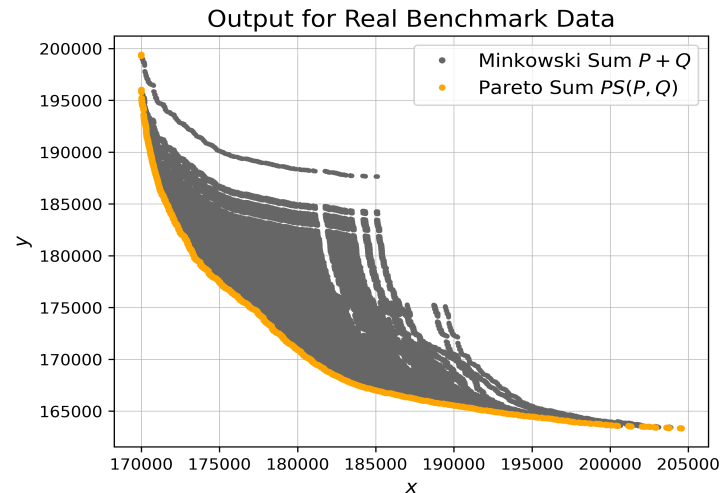
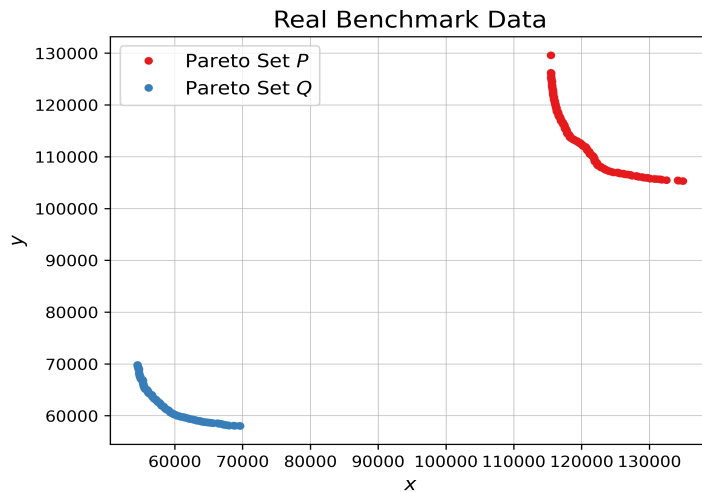
Algorithm in time $\tilde{O}(\min(n^2, nk))$ exist [Funke, Hespe, Sanders, Storandt, Truschel;23]

$\rightsquigarrow$ Conditionally optimal under 3-SUM hypothesis [Gokaj, Künnemann, Storandt, Truschel;25]

Question: Can we circumvent the conditional lower bound (and to what extent)?

Previous work on Pareto Sum

Problem stems from bicriteria-route planning.



Algorithm in time $\tilde{O}(\min(n^2, nk))$ exist [Funke, Hespe, Sanders, Storandt, Truschel;23]

$\rightsquigarrow$ Conditionally optimal under 3-SUM hypothesis [Gokaj, Künnemann, Storandt, Truschel;25]

Question: Can we circumvent the conditional lower bound (and to what extent)?

Our results:

Our results:

Theorem: Pareto Sum with bounded entries (entries in $\{0, \dots, W\}$) can be solved in time $\tilde{O}(W^{1.5})$.

Our results:

Theorem: Pareto Sum with bounded entries (entries in $\{0, \dots, W\}$) can be solved in time $\tilde{O}(W^{1.5})$.

Corollary: We can compute a $2t$ - Approximation of Pareto Sum in time $\tilde{O}(n + (\frac{W}{t})^{1.5})$.

Our results:

Theorem: Pareto Sum with bounded entries (entries in $\{0, \dots, W\}$) can be solved in time $\tilde{O}(W^{1.5})$.

Corollary: We can compute a $2t$ - Approximation of Pareto Sum in time $\tilde{O}(n + (\frac{W}{t})^{1.5})$.

Theorem [(Weak) Hardness of Approximation]:

-For any t an algo in time $O(n + (\frac{W}{t})^{1.5-\epsilon})$ leads to $O(n^{1.5-\epsilon'})$ for bounded monotone min-plus convolution.

Our results:

Theorem: Pareto Sum with bounded entries (entries in $\{0, \dots, W\}$) can be solved in time $\tilde{O}(W^{1.5})$.

Corollary: We can compute a $2t$ - Approximation of Pareto Sum in time $\tilde{O}(n + (\frac{W}{t})^{1.5})$.

Theorem [(Weak) Hardness of Approximation]:

-For any t an algo in time $O(n + (\frac{W}{t})^{1.5-\epsilon})$ leads to $O(n^{1.5-\epsilon'})$ for bounded monotone min-plus convolution.

Engineering results: Approx. Pareto Sum also faster in practice.

Our results:

Theorem: Pareto Sum with bounded entries (entries in $\{0, \dots, W\}$) can be solved in time $\tilde{O}(W^{1.5})$.

Corollary: We can compute a $2t$ - Approximation of Pareto Sum in time $\tilde{O}(n + (\frac{W}{t})^{1.5})$.

Theorem [(Weak) Hardness of Approximation]:

-For any t an algo in time $O(n + (\frac{W}{t})^{1.5-\epsilon})$ leads to $O(n^{1.5-\epsilon'})$ for bounded monotone min-plus convolution.

Engineering results: Approx. Pareto Sum also faster in practice.

Approx. enables an effective choice of runtime and output size vs approx. quality.

Our results:

Theorem: Pareto Sum with bounded entries (entries in $\{0, \dots, W\}$) can be solved in time $\tilde{O}(W^{1.5})$.

Corollary: We can compute a $2t$ - Approximation of Pareto Sum in time $\tilde{O}(n + (\frac{W}{t})^{1.5})$.

Theorem [(Weak) Hardness of Approximation]:

-For any t an algo in time $O(n + (\frac{W}{t})^{1.5-\epsilon})$ leads to $O(n^{1.5-\epsilon'})$ for bounded monotone min-plus convolution.

Engineering results: Approx. Pareto Sum also faster in practice.

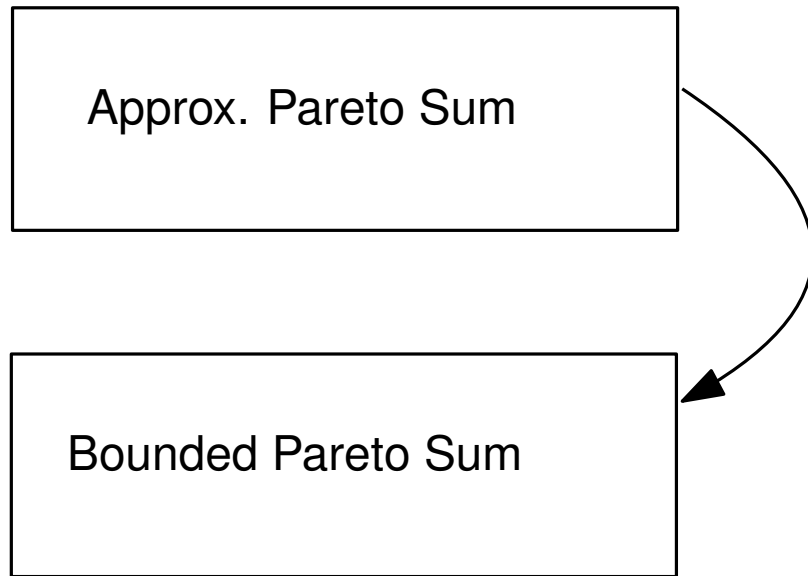
Approx. enables an effective choice of runtime and output size vs approx. quality.

A series of simple and efficient reductions:

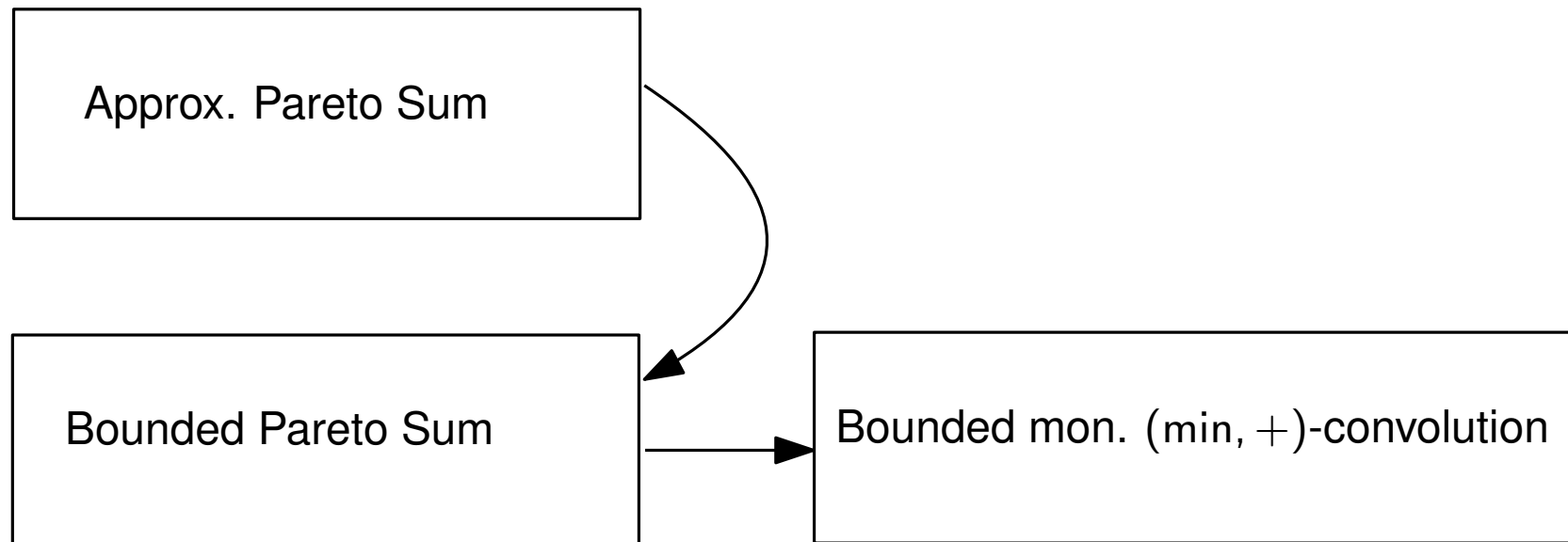
A series of simple and efficient reductions:

Approx. Pareto Sum

A series of simple and efficient reductions:



A series of simple and efficient reductions:



On $(\min, +)$ -convolution

On (min, +)-convolution

Min-plus Convolution: Given Arrays A and B of length n , compute array C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

A	3 $i=0$	1 $i=1$	4 $i=2$	2 $i=3$	5 $i=4$
B	2 $j=0$	4 $j=1$	1 $j=2$	3 $j=3$	0 $j=4$
i		$j = k - i$		$A[i] + B[j]$	
	0		3		6
	1		2		$2 \leftarrow \min$
	2		1		8
	3		0		4

On $(\min, +)$ -convolution

Min-plus Convolution: Given Arrays A and B of length n , compute array C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

Conjectured to require $n^{2-o(1)}$ time.

A	3 $i=0$	1 $i=1$	4 $i=2$	2 $i=3$	5 $i=4$
B	2 $j=0$	4 $j=1$	1 $j=2$	3 $j=3$	0 $j=4$
i			$j = k - i$		$A[i] + B[j]$
	0		3		6
	1		2		$2 \leftarrow \min$
	2		1		8
	3		0		4

On $(\min, +)$ -convolution

Min-plus Convolution: Given Arrays A and B of length n , compute array C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

Conjectured to require $n^{2-o(1)}$ time.

$\tilde{O}(n\Delta)$ -time algo for $(\min, +)$ -convolution [Bringmann, Cassis;23]

(Δ describes dissimilarity of A and B to convex functions)

A	3 $i=0$	1 $i=1$	4 $i=2$	2 $i=3$	5 $i=4$
B	2 $j=0$	4 $j=1$	1 $j=2$	3 $j=3$	0 $j=4$
	i	$j = k - i$		$A[i] + B[j]$	
	0	3		6	
	1	2		2 $\leftarrow \min$	
	2	1		8	
	3	0		4	

On (min, +)-convolution

Min-plus Convolution: Given Arrays A and B of length n , compute array C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

Conjectured to require $n^{2-o(1)}$ time.

$\tilde{O}(n\Delta)$ -time algo for (min, +)-convolution [Bringmann, Cassis;23]

(Δ describes dissimilarity of A and B to convex functions)

A	3 $i=0$	1 $i=1$	4 $i=2$	2 $i=3$	5 $i=4$
B	2 $j=0$	4 $j=1$	1 $j=2$	3 $j=3$	0 $j=4$
	i	$j = k - i$			$A[i] + B[j]$
	0	3			6
	1	2			$2 \leftarrow \min$
	2	1			8
	3	0			4

Bounded Monotone Min-plus Convolution:

Given monotone arrays A and B with entries bounded by $O(n)$, compute C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

On (min, +)-convolution

Min-plus Convolution: Given Arrays A and B of length n , compute array C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

Conjectured to require $n^{2-o(1)}$ time.

$\tilde{O}(n\Delta)$ -time algo for (min, +)-convolution [Bringmann, Cassis;23]

(Δ describes dissimilarity of A and B to convex functions)

A	3 $i=0$	1 $i=1$	4 $i=2$	2 $i=3$	5 $i=4$
B	2 $j=0$	4 $j=1$	1 $j=2$	3 $j=3$	0 $j=4$
	i	$j = k - i$		$A[i] + B[j]$	
	0	3		6	
	1	2		2 $\leftarrow \min$	
	2	1		8	
	3	0		4	

Bounded Monotone Min-plus Convolution:

Given monotone arrays A and B with entries bounded by $O(n)$, compute C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

First breakthrough for monotone bounded case $O(n^{1.859})$ via additive combinatorics [Chan, Lewenstein;13]

On (min, +)-convolution

Min-plus Convolution: Given Arrays A and B of length n , compute array C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

Conjectured to require $n^{2-o(1)}$ time.

$\tilde{O}(n\Delta)$ -time algo for (min, +)-convolution [Bringmann, Cassis;23]

(Δ describes dissimilarity of A and B to convex functions)

A	3 $i=0$	1 $i=1$	4 $i=2$	2 $i=3$	5 $i=4$
B	2 $j=0$	4 $j=1$	1 $j=2$	3 $j=3$	0 $j=4$
	i	$j = k - i$		$A[i] + B[j]$	
	0	3		6	
	1	2		2 $\leftarrow \min$	
	2	1		8	
	3	0		4	

Bounded Monotone Min-plus Convolution:

Given monotone arrays A and B with entries bounded by $O(n)$, compute C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

First breakthrough for monotone bounded case $O(n^{1.859})$ via additive combinatorics [Chan, Lewenstein;13]

Current best $\tilde{O}(n^{1.5})$ randomized algorithm via FFT + Hashing [Chi, Duan, Xie, Zhang;22]

On (min, +)-convolution

Min-plus Convolution: Given Arrays A and B of length n , compute array C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

Conjectured to require $n^{2-o(1)}$ time.

$\tilde{O}(n\Delta)$ -time algo for (min, +)-convolution [Bringmann, Cassis;23]

(Δ describes dissimilarity of A and B to convex functions)

A	3 $i=0$	1 $i=1$	4 $i=2$	2 $i=3$	5 $i=4$
B	2 $j=0$	4 $j=1$	1 $j=2$	3 $j=3$	0 $j=4$
	i	$j = k - i$	$A[i] + B[j]$		
	0	3	6		
	1	2	2 $\leftarrow$ min		
	2	1	8		
	3	0	4		

Bounded Monotone Min-plus Convolution:

Given monotone arrays A and B with entries bounded by $O(n)$, compute C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

First breakthrough for monotone bounded case $O(n^{1.859})$ via additive combinatorics [Chan, Lewenstein;13]

Current best $\tilde{O}(n^{1.5})$ randomized algorithm via FFT + Hashing [Chi, Duan, Xie, Zhang;22]

Recently determinized by [Jin, Park, Saha, Xu;26]

On (min, +)-convolution

Min-plus Convolution: Given Arrays A and B of length n , compute array C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

Conjectured to require $n^{2-o(1)}$ time.

$\tilde{O}(n\Delta)$ -time algo for (min, +)-convolution [Bringmann, Cassis;23]

(Δ describes dissimilarity of A and B to convex functions)

A	3 $i=0$	1 $i=1$	4 $i=2$	2 $i=3$	5 $i=4$
B	2 $j=0$	4 $j=1$	1 $j=2$	3 $j=3$	0 $j=4$
	i	$j = k - i$		$A[i] + B[j]$	
	0	3		6	
	1	2		2 $\leftarrow \min$	
	2	1		8	
	3	0		4	

Bounded Monotone Min-plus Convolution:

Given monotone arrays A and B with entries bounded by $O(n)$, compute C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

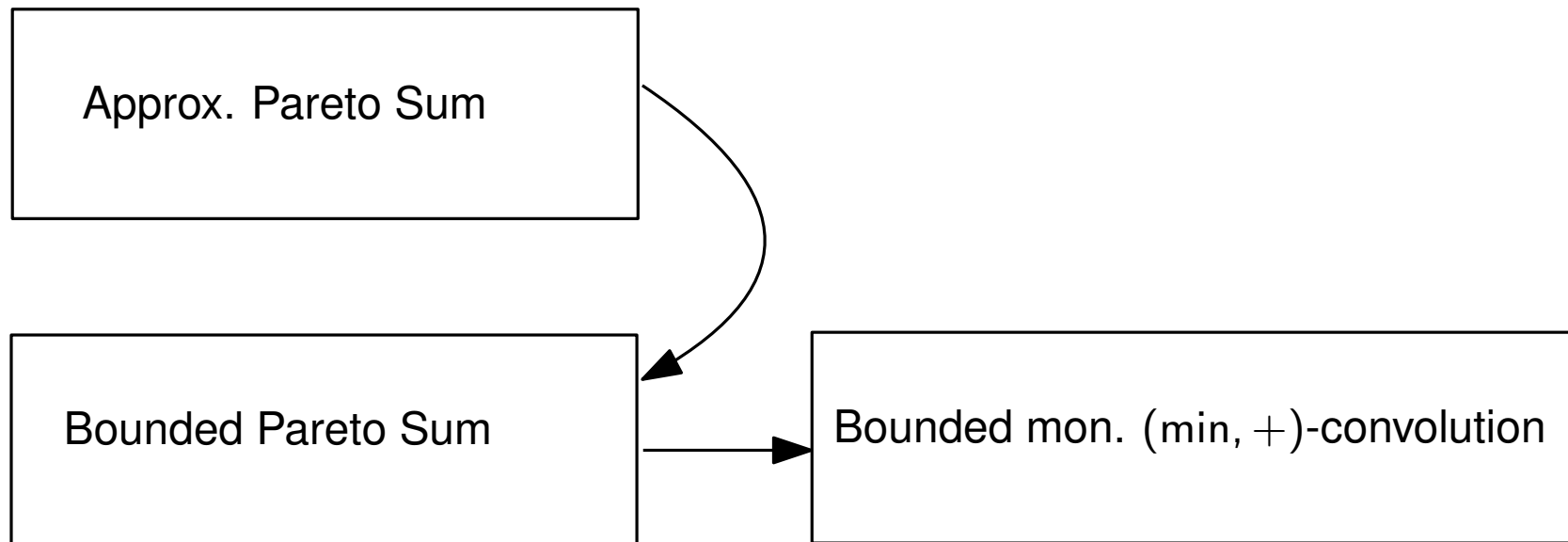
First breakthrough for monotone bounded case $O(n^{1.859})$ via additive combinatorics [Chan, Lewenstein;13]

Current best $\tilde{O}(n^{1.5})$ randomized algorithm via FFT + Hashing [Chi, Duan, Xie, Zhang;22]

Recently determinized by [Jin, Park, Saha, Xu;26]

Many other algorithms and applications to Knapsack...

A series of simple and efficient reductions:



Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing.

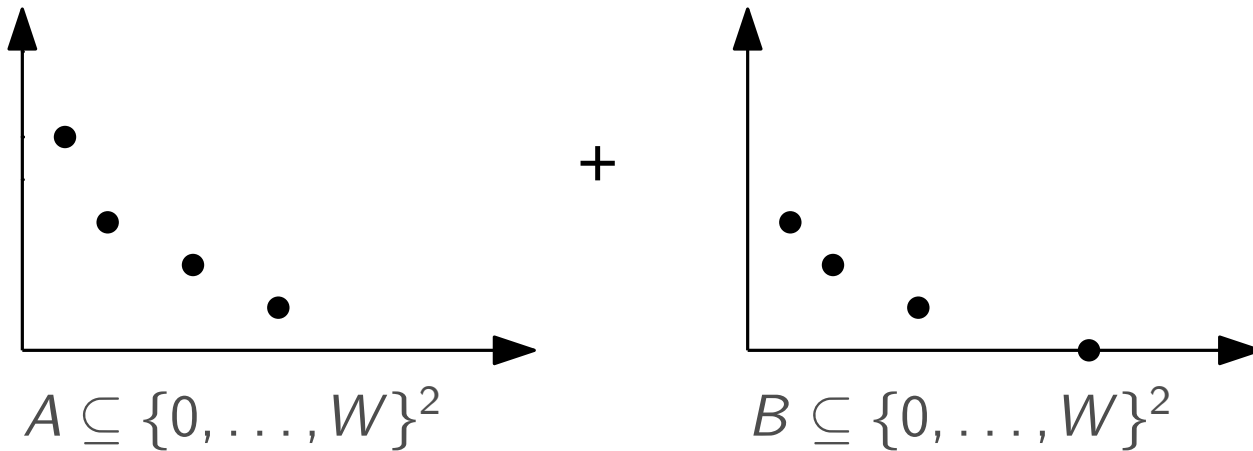
Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point.

Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing.

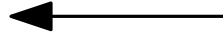
Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point.



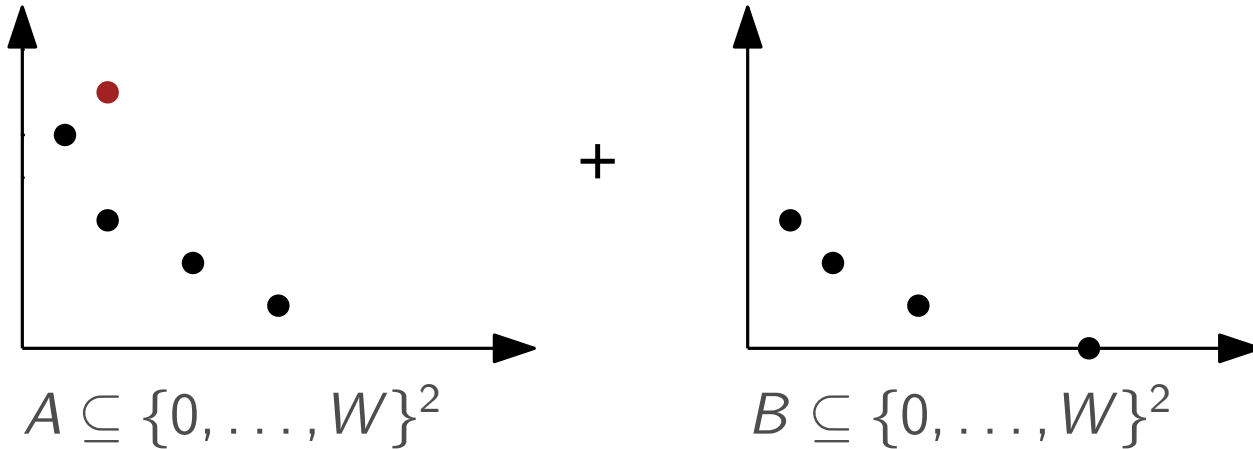
Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing.



Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point.

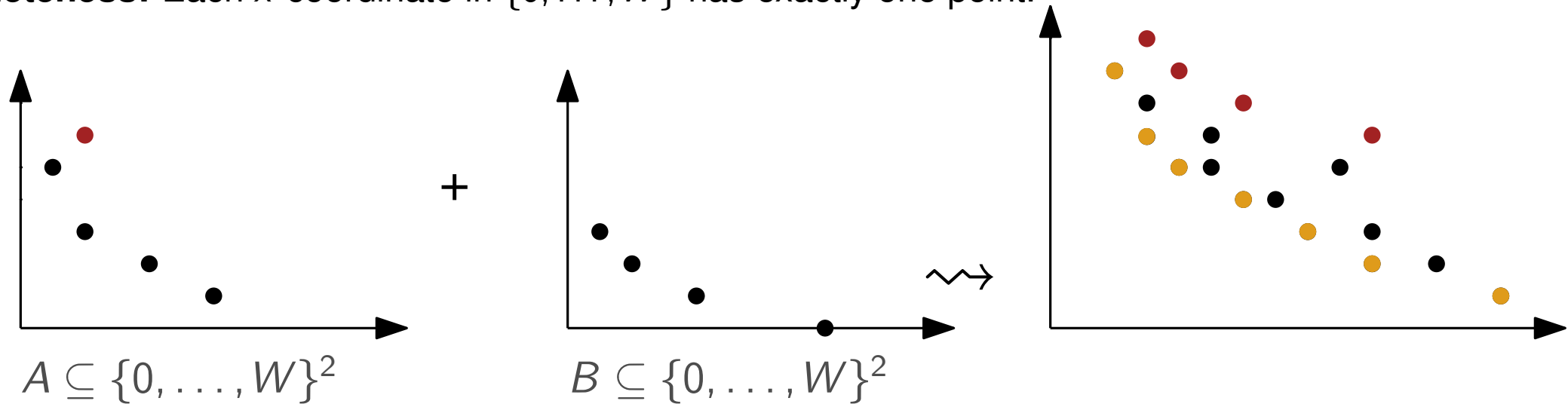


Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing. $\leftarrow$

Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point.

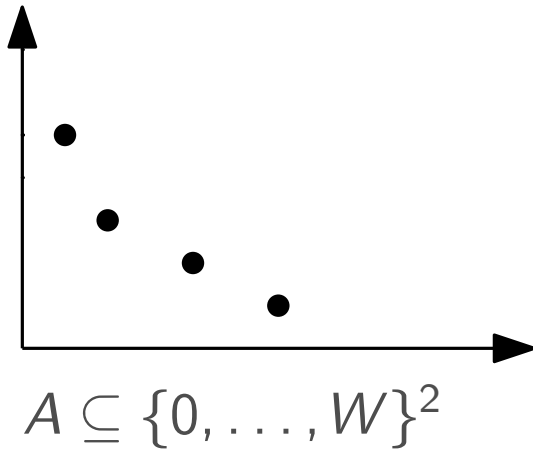


Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

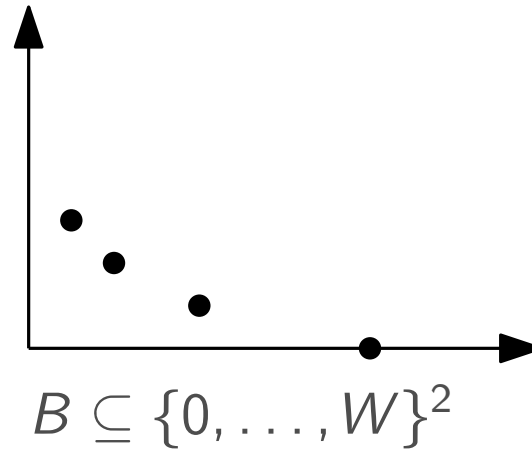
Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing. [Preprocessing](#)

Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point.



+

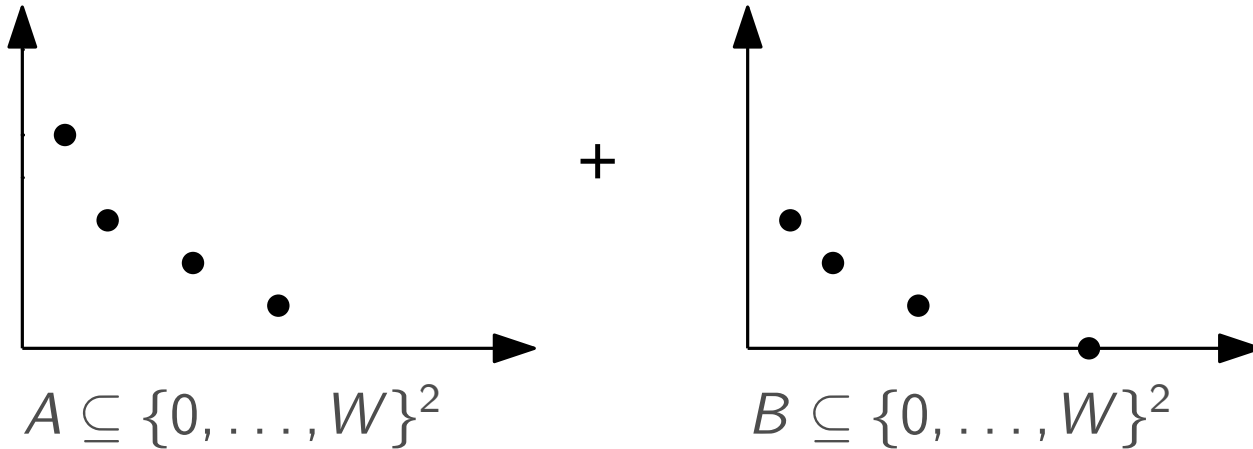


Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing. [Preprocessing](#)

Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point. $\longleftarrow$ Translate + Densify

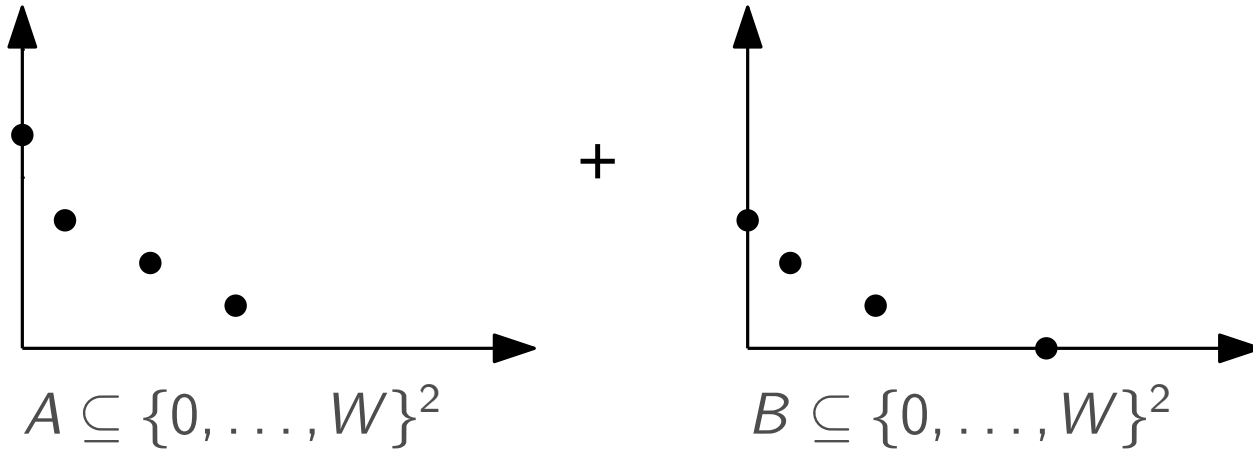


Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing. [Preprocessing](#)

Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point. $\longleftarrow$ Translate + Densify

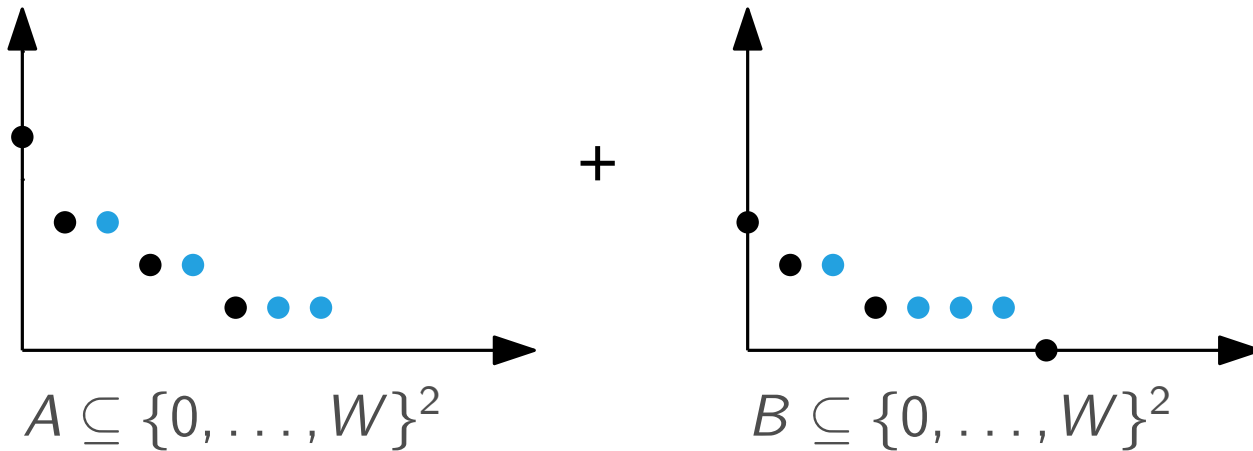


Bounded Pareto Sum $\rightarrow$ Bounded Monotone (min, +)-convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing. [Preprocessing](#)

Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point. $\longleftarrow$ Translate + Densify

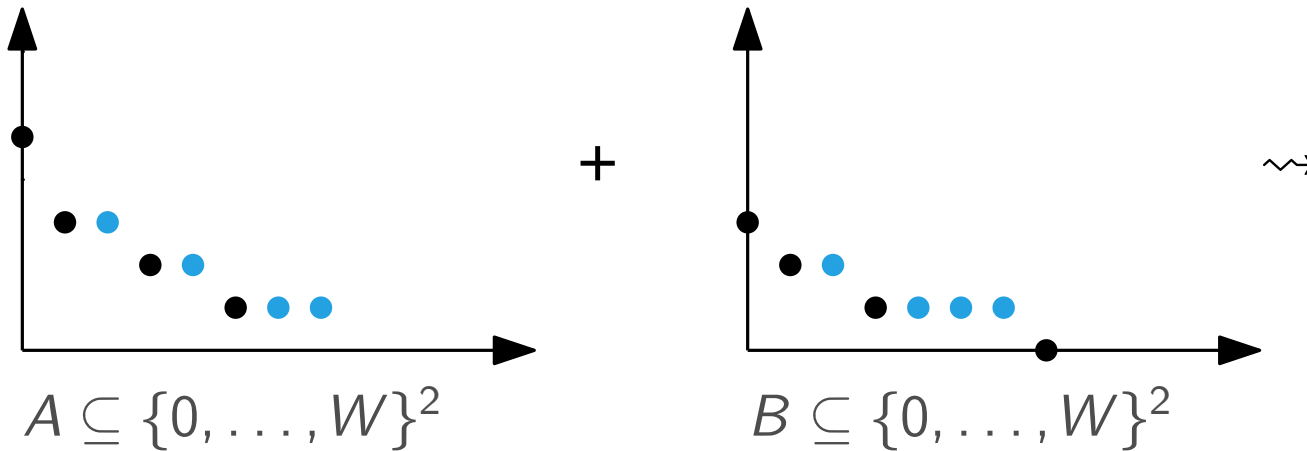


Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing. [Preprocessing](#)

Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point. [Translate + Densify](#)

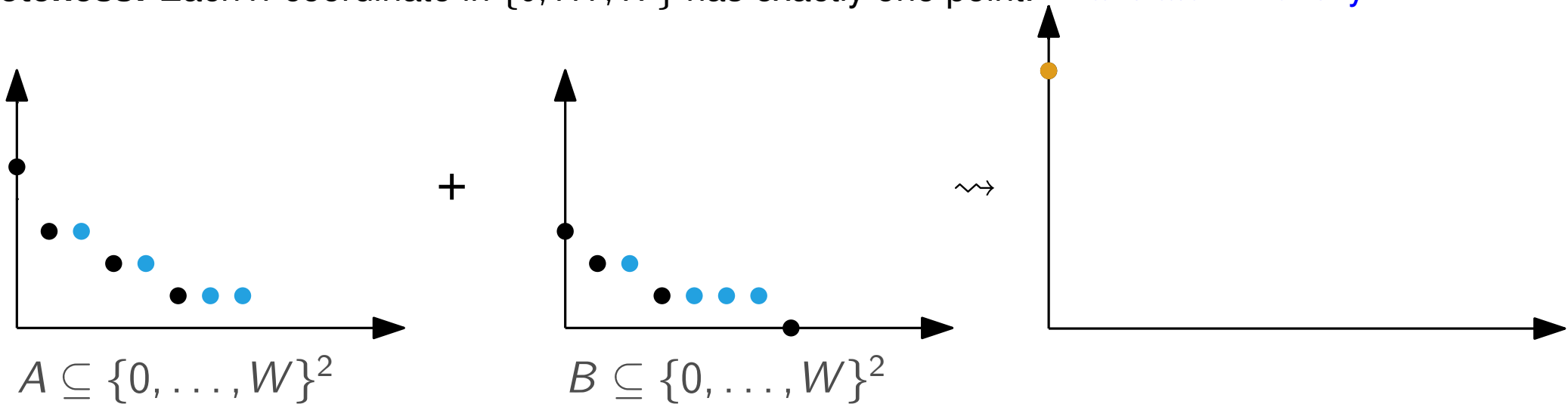


Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing. [Preprocessing](#)

Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point. [Translate + Densify](#)

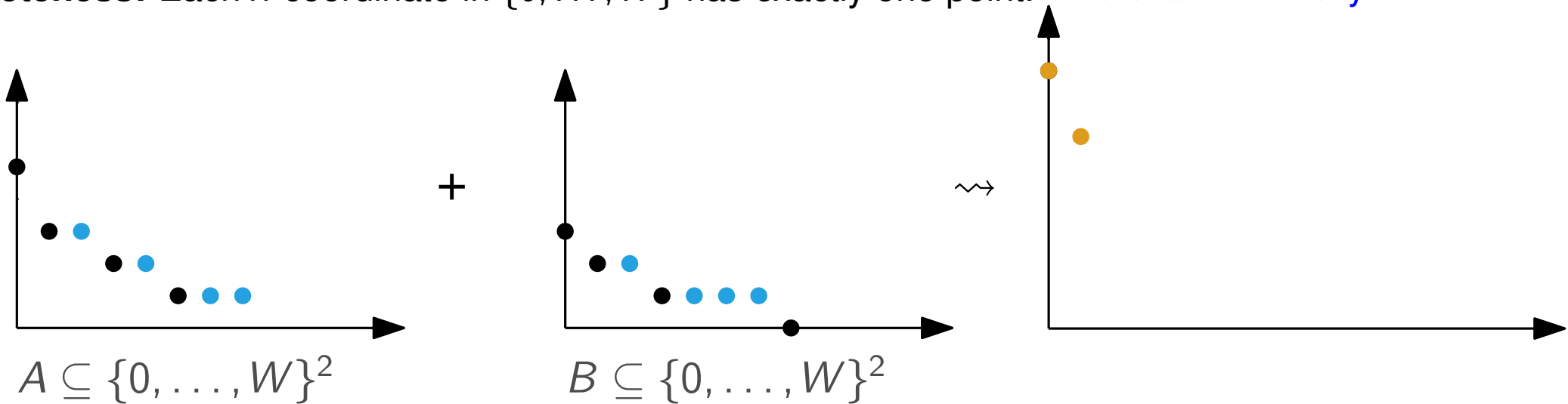


Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing. [Preprocessing](#)

Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point. [Translate + Densify](#)

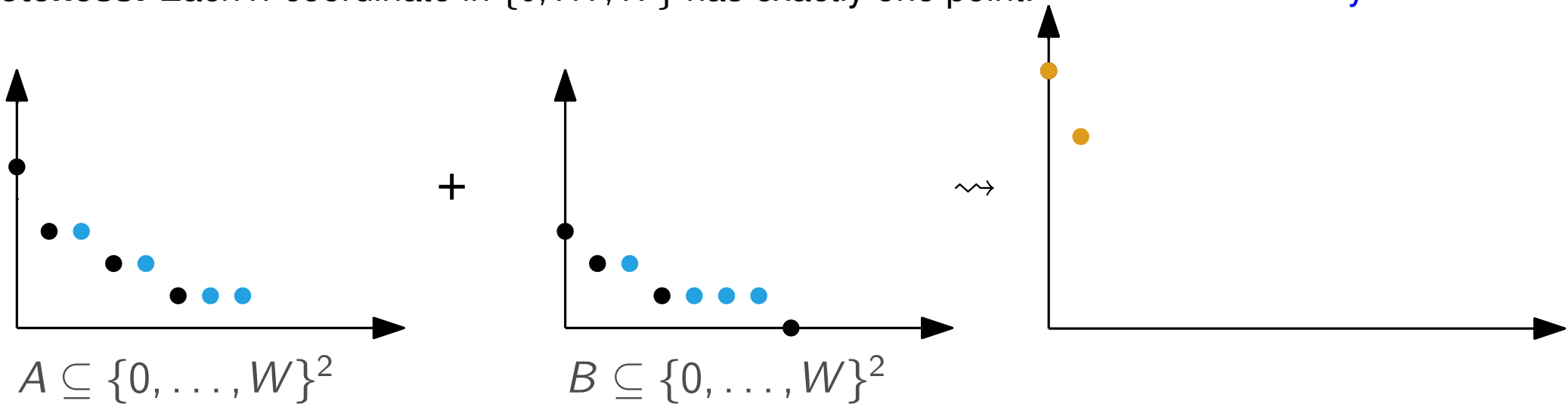


Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing. [Preprocessing](#)

Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point. [Translate + Densify](#)



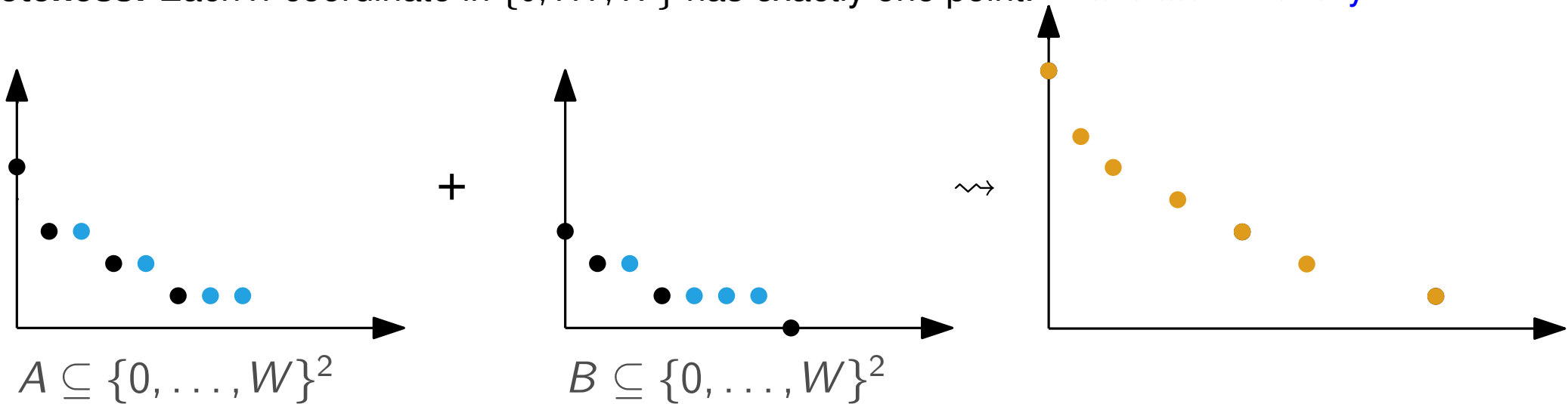
At point with x -coordinate k , we get $\min_{a \in A, b \in B, a_1 + b_1 = k} (a_2 + b_2)$

Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing. [Preprocessing](#)

Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point. [Translate + Densify](#)



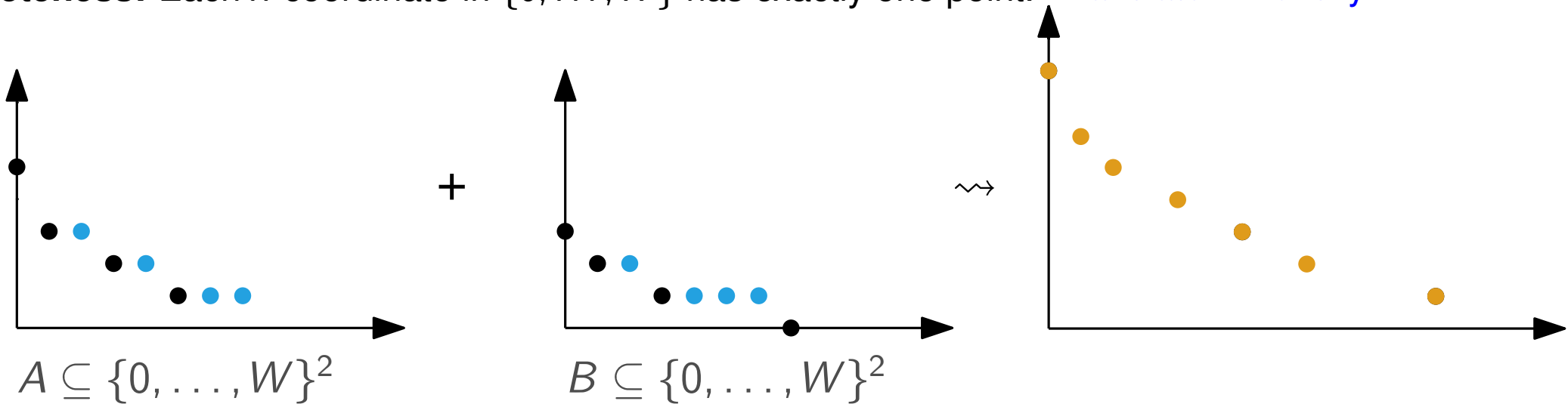
At point with x -coordinate k , we get $\min_{a \in A, b \in B, a_1 + b_1 = k} (a_2 + b_2)$

Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing. [Preprocessing](#)

Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point. [Translate + Densify](#)



At point with x -coordinate k , we get $\min_{a \in A, b \in B, a_1 + b_1 = k} (a_2 + b_2)$

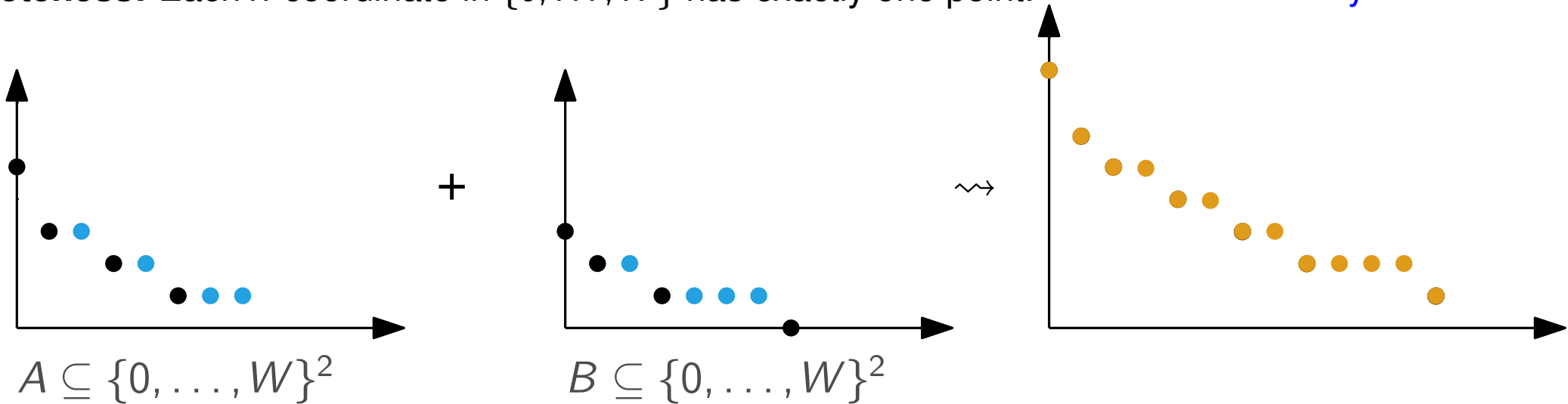
Min-plus Convolution: Given Arrays A and B , compute array C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing. [Preprocessing](#)

Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point. [Translate + Densify](#)



At point with x -coordinate k , we get $\min_{a \in A, b \in B, a_1 + b_1 = k} (a_2 + b_2)$

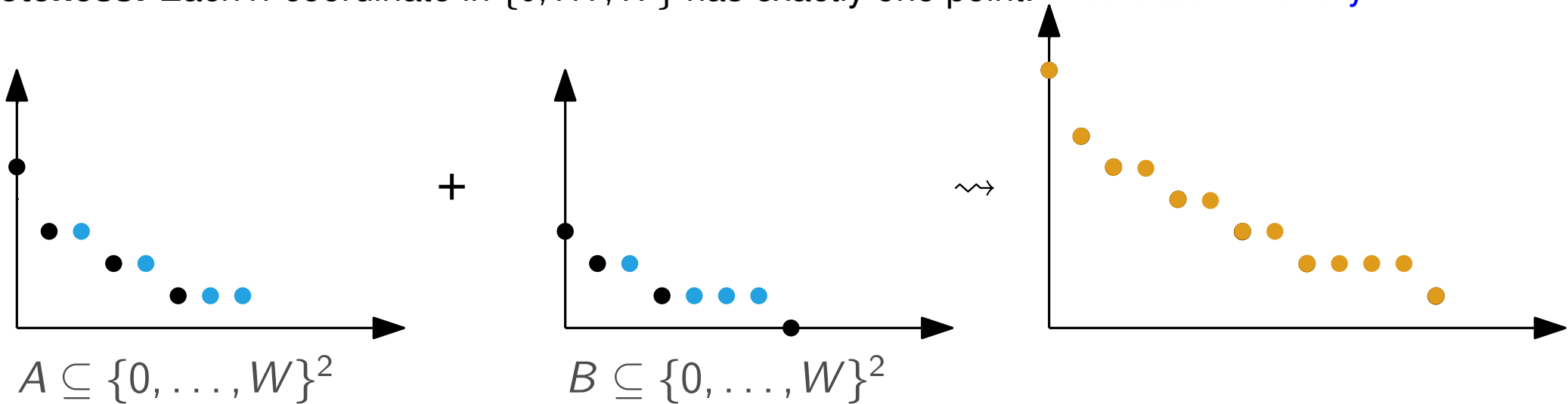
Min-plus Convolution: Given Arrays A and B , compute array C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$.

Monotonicity: A, B are monotone decreasing. [Preprocessing](#)

Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point. [Translate + Densify](#)



At point with x -coordinate k , we get $\min_{a \in A, b \in B, a_1 + b_1 = k} (a_2 + b_2)$

Min-plus Convolution: Given Arrays A and B , compute array C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

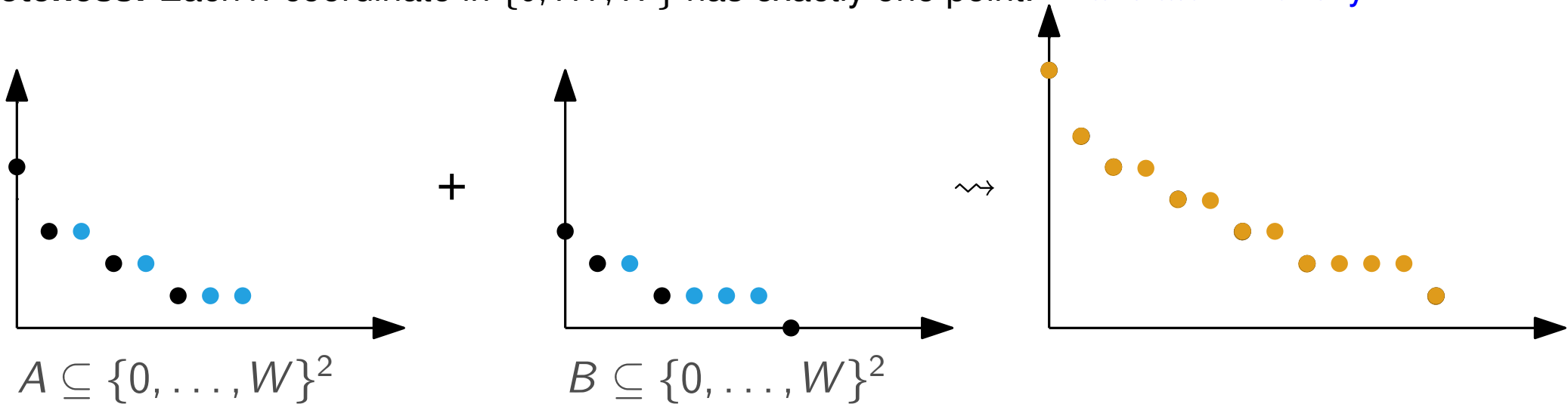
Theorem: Bounded Pareto Sum can be solved in time $\tilde{O}(W^{1.5})$.

Bounded Pareto Sum $\rightarrow$ Bounded Monotone $(\min, +)$ -convolution

Boundedness (Small universe): $A, B \subseteq \{0, \dots, W\}^2$. $\leftarrow ?$

Monotonicity: A, B are monotone decreasing. [Preprocessing](#)

Completeness: Each x -coordinate in $\{0, \dots, W\}$ has exactly one point. [Translate + Densify](#)



At point with x -coordinate k , we get $\min_{a \in A, b \in B, a_1 + b_1 = k} (a_2 + b_2)$

Min-plus Convolution: Given Arrays A and B , compute array C with $C[k] := \min_{i+j=k} (A[i] + B[j])$.

Theorem: Bounded Pareto Sum can be solved in time $\tilde{O}(W^{1.5})$.

Approximation via Scaling (and reduction)

Approximation via Scaling (and reduction)

Pareto Sum: Given sets $A, B \subseteq \mathbb{R}^2$. Compute C such that :

Approximation via Scaling (and reduction)

Pareto Sum: Given sets $A, B \subseteq \mathbb{R}^2$. Compute C such that :

- $\forall a \in A \forall b \in B \exists c \in C : c \leq a + b$
- $C \subseteq A + B$,
- The elements of C are non-dominated.

Approximation via Scaling (and reduction)

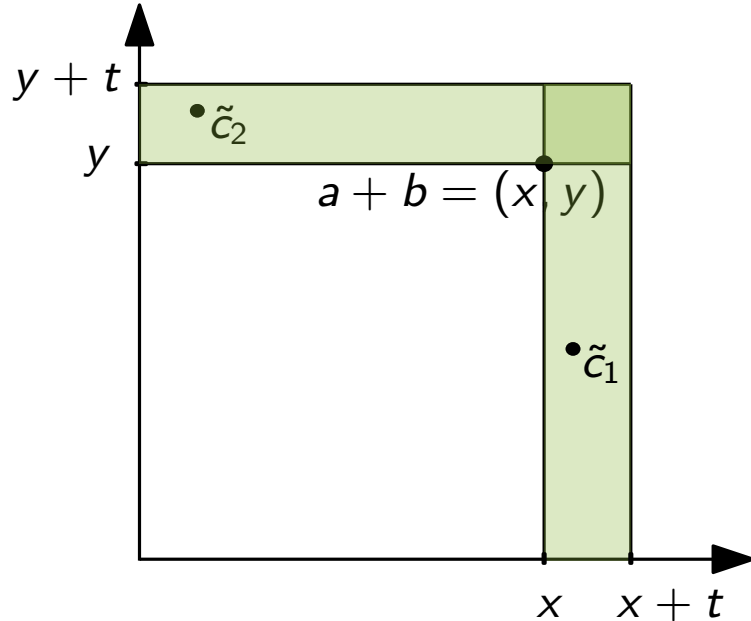
t -Approximate Pareto Sum: Given sets $A, B \subseteq \mathbb{R}^2$. Compute C such that :

- $\forall a \in A \forall b \in B \exists c \in C : c \leq a + b + \begin{pmatrix} t \\ t \end{pmatrix}$
- $C \subseteq A + B$,
- The elements of C are non-dominated.

Approximation via Scaling (and reduction)

t -Approximate Pareto Sum: Given sets $A, B \subseteq \mathbb{R}^2$. Compute C such that :

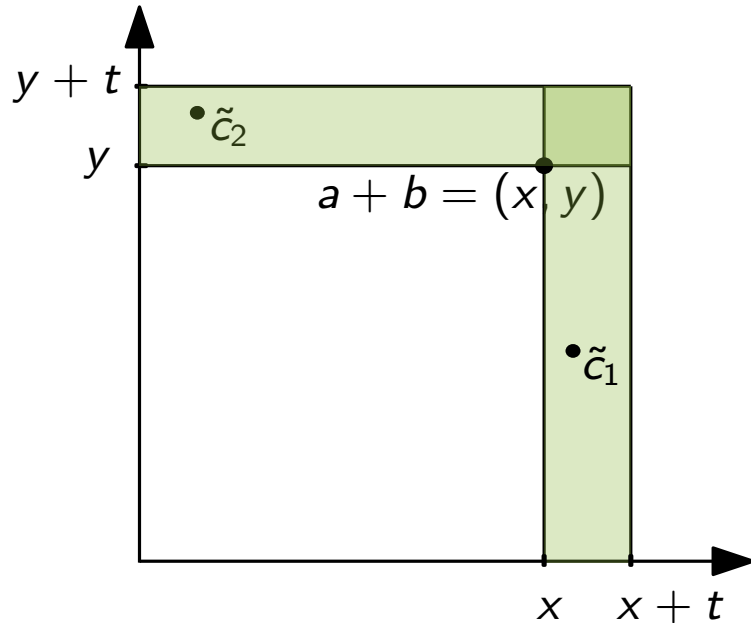
- $\forall a \in A \forall b \in B \exists c \in C : c \leq a + b + \begin{pmatrix} t \\ t \end{pmatrix}$
- $C \subseteq A + B$,
- The elements of C are non-dominated.



Approximation via Scaling (and reduction)

t -Approximate Pareto Sum: Given sets $A, B \subseteq \mathbb{R}^2$. Compute C such that :

- $\forall a \in A \forall b \in B \exists c \in C : c \leq a + b + \begin{pmatrix} t \\ t \end{pmatrix}$
- $C \subseteq A + B$,
- The elements of C are non-dominated.

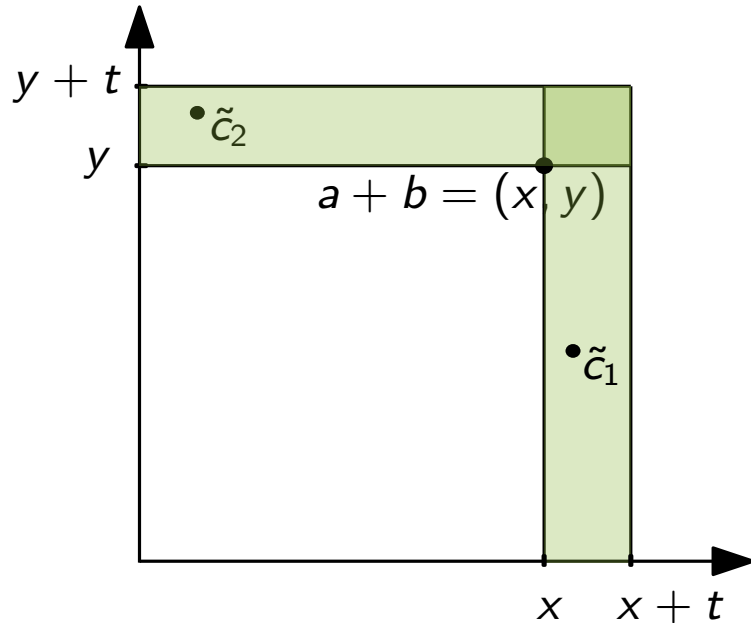


$$\tilde{c}_1 \leq a + b + (t, t),$$

Approximation via Scaling (and reduction)

t -Approximate Pareto Sum: Given sets $A, B \subseteq \mathbb{R}^2$. Compute C such that :

- $\forall a \in A \forall b \in B \exists c \in C : c \leq a + b + \begin{pmatrix} t \\ t \end{pmatrix}$
- $C \subseteq A + B$,
- The elements of C are non-dominated.



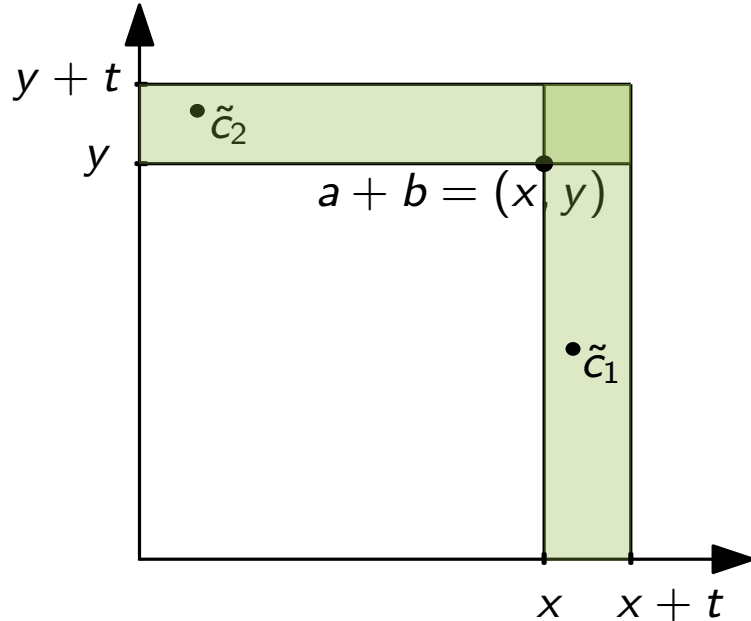
$$\tilde{c}_1 \leq a + b + (t, t),$$

$$\tilde{c}_2 \leq a + b + (t, t).$$

Approximation via Scaling (and reduction)

t -Approximate Pareto Sum: Given sets $A, B \subseteq \mathbb{R}^2$. Compute C such that :

- $\forall a \in A \forall b \in B \exists c \in C : c \leq a + b + \begin{pmatrix} t \\ t \end{pmatrix}$
- $C \subseteq A + B$,
- The elements of C are non-dominated.



$$\tilde{c}_1 \leq a + b + (t, t),$$

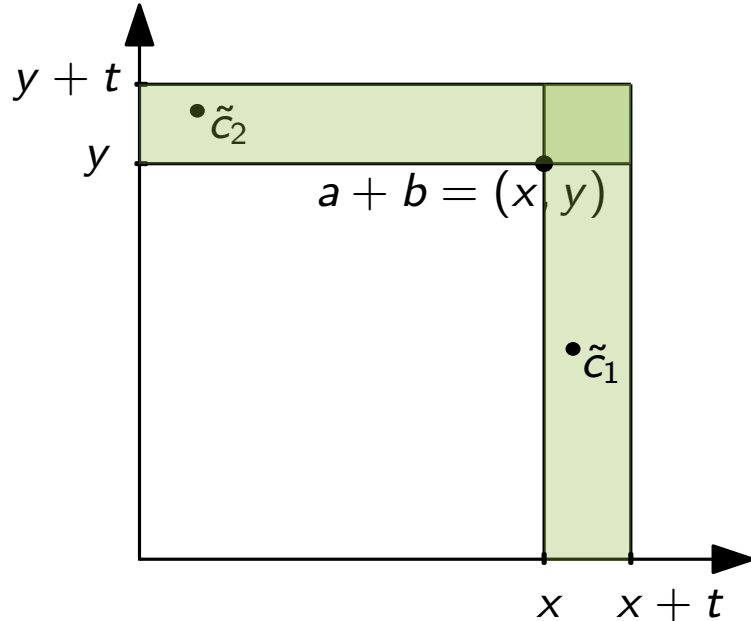
$$\tilde{c}_2 \leq a + b + (t, t).$$



Approximation via Scaling (and reduction)

t -Approximate Pareto Sum: Given sets $A, B \subseteq \mathbb{R}^2$. Compute C such that :

- $\forall a \in A \forall b \in B \exists c \in C : c \leq a + b + \begin{pmatrix} t \\ t \end{pmatrix}$
- $C \subseteq A + B$,
- The elements of C are non-dominated.



$$\tilde{c}_1 \leq a + b + (t, t),$$

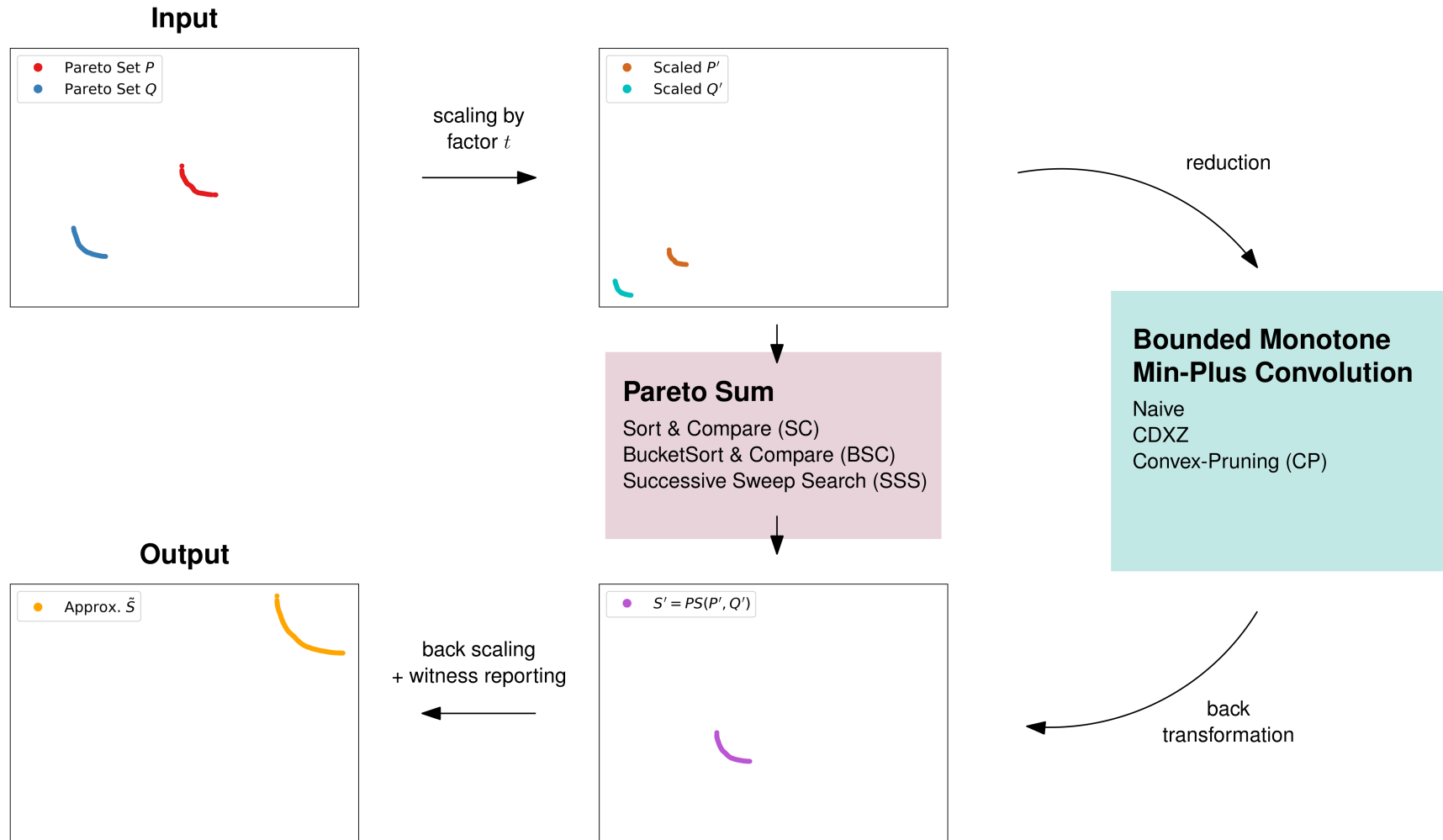
$$\tilde{c}_2 \leq a + b + (t, t).$$



Above approach leads to a $2t$ -approx of Pareto Sum in time $\tilde{O}(n + (\frac{w}{t})^{1.5})$.

Experiments & Implementation

Experiments & Implementation



Algorithms implemented for $(\min, +)$ -convolution

Algorithms implemented for $(\min, +)$ -convolution

1. CDXZ-algorithm based on $\tilde{O}(n^{1.6})$ -algo from [Chi, Duan, Xie, Zhang; 23]

Algorithms implemented for $(\min, +)$ -convolution

1. CDXZ-algorithm based on $\tilde{O}(n^{1.6})$ -algo from [Chi, Duan, Xie, Zhang; 23]
 1. Scale down arrays A and B and compute the $(\min, +)$ -convolution (many same entries).
 2. Scaling back up introduces an error. Possible to correct via FFT.

Algorithms implemented for $(\min, +)$ -convolution

1. CDXZ-algorithm based on $\tilde{O}(n^{1.6})$ -algo from [Chi, Duan, Xie, Zhang; 23]
 1. Scale down arrays A and B and compute the $(\min, +)$ -convolution (many same entries).
 2. Scaling back up introduces an error. Possible to correct via FFT.

Simplifications performed for Step 1. Implemented without witness reporting $\rightsquigarrow$ slightly weaker approx.

Algorithms implemented for $(\min, +)$ -convolution

1. CDXZ-algorithm based on $\tilde{O}(n^{1.6})$ -algo from [Chi, Duan, Xie, Zhang; 23]
 1. Scale down arrays A and B and compute the $(\min, +)$ -convolution (many same entries).
 2. Scaling back up introduces an error. Possible to correct via FFT.

Simplifications performed for Step 1. Implemented without witness reporting $\rightsquigarrow$ slightly weaker approx.

2. Convex Pruning based on Δ -near convex $(\min, +)$ -convolution [Bringmann, Cassis;23]

Algorithms implemented for $(\min, +)$ -convolution

1. CDXZ-algorithm based on $\tilde{O}(n^{1.6})$ -algo from [Chi, Duan, Xie, Zhang; 23]
 1. Scale down arrays A and B and compute the $(\min, +)$ -convolution (many same entries).
 2. Scaling back up introduces an error. Possible to correct via FFT.

Simplifications performed for Step 1. Implemented without witness reporting $\rightsquigarrow$ slightly weaker approx.

2. Convex Pruning based on Δ -near convex $(\min, +)$ -convolution [Bringmann, Cassis;23]
 1. Compute convex approximations of A and B .
 2. Use convex approximations to prune the search space for each $C[k]$.

Algorithms implemented for $(\min, +)$ -convolution

1. CDXZ-algorithm based on $\tilde{O}(n^{1.6})$ -algo from [Chi, Duan, Xie, Zhang; 23]
 1. Scale down arrays A and B and compute the $(\min, +)$ -convolution (many same entries).
 2. Scaling back up introduces an error. Possible to correct via FFT.

Simplifications performed for Step 1. Implemented without witness reporting $\rightsquigarrow$ slightly weaker approx.

2. Convex Pruning based on Δ -near convex $(\min, +)$ -convolution [Bringmann, Cassis;23]
 1. Compute convex approximations of A and B .
 2. Use convex approximations to prune the search space for each $C[k]$.

Worst-case still $O(n^2)$, practical runtime dependent on pruning amount.

Algorithms implemented for Pareto Sum

Algorithms implemented for Pareto Sum

1. Bucketsort and compare

Algorithms implemented for Pareto Sum

1. Bucketsort and compare

1. Compute the sumset $P + Q$, by direct addressing store smallest y -coordinate at observed x -coordinate.
2. Perform a single sweep to keep non-dominated elements.

Algorithms implemented for Pareto Sum

1. Bucketsort and compare

1. Compute the sumset $P + Q$, by direct addressing store smallest y -coordinate at observed x -coordinate.
2. Perform a single sweep to keep non-dominated elements.

2. Previous Pareto Sum algorithms [Funke, Hespe, Sanders, Storandt, Truschel; 23]

Algorithms implemented for Pareto Sum

1. Bucketsort and compare

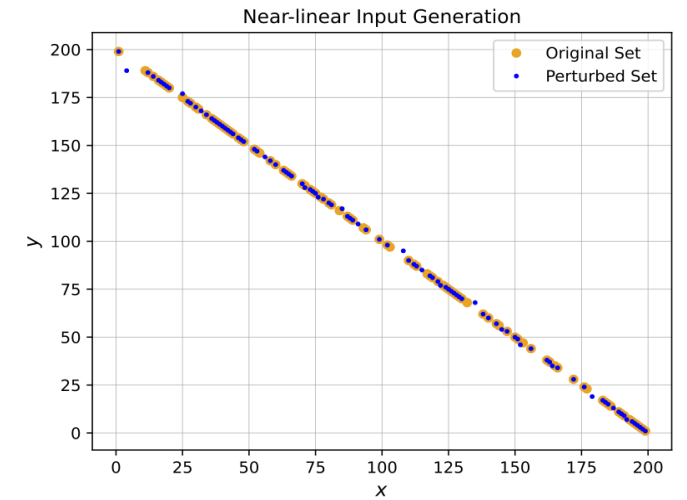
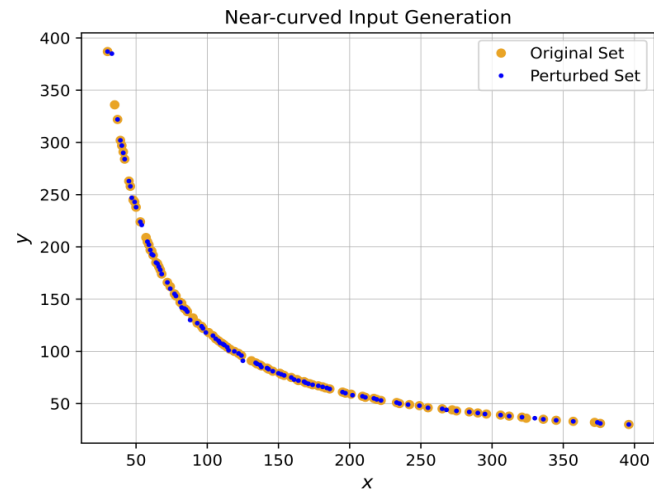
1. Compute the sumset $P + Q$, by direct addressing store smallest y -coordinate at observed x -coordinate.
2. Perform a single sweep to keep non-dominated elements.

2. Previous Pareto Sum algorithms [Funke, Hespe, Sanders, Storandt, Truschel; 23]

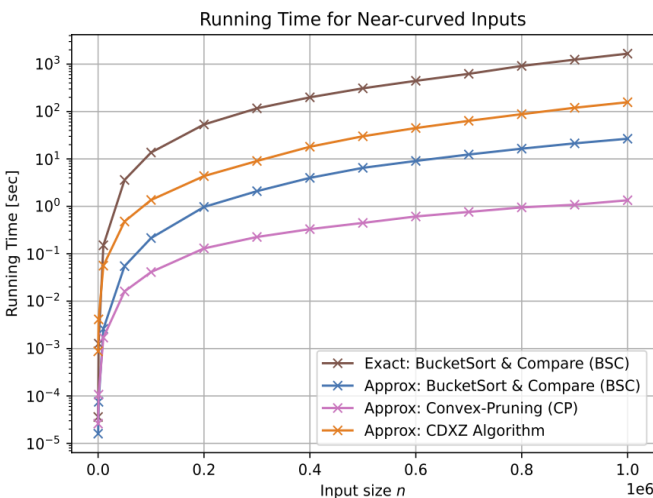
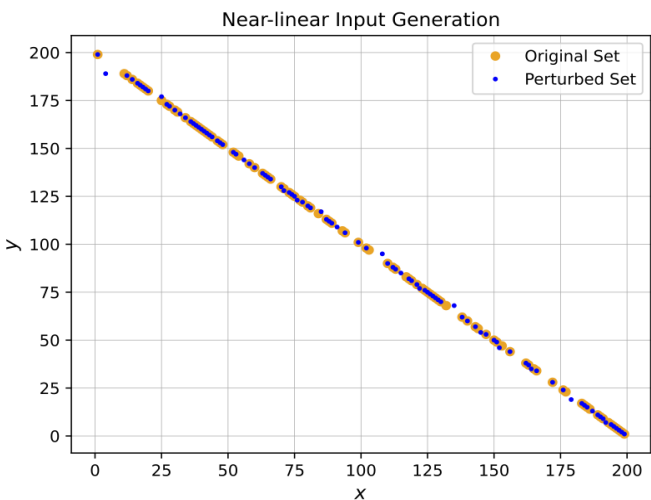
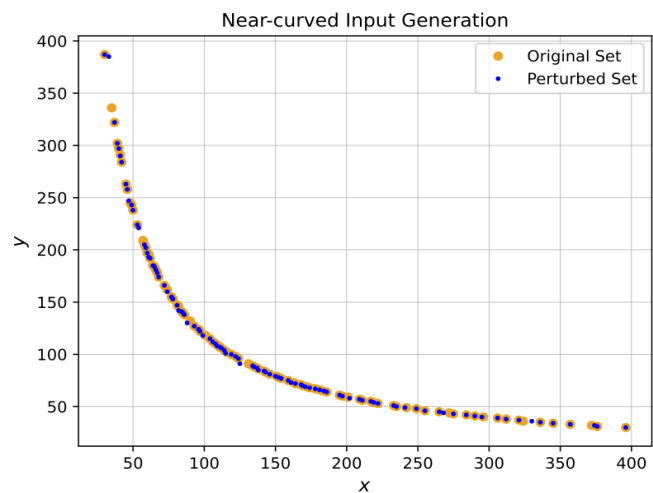
↪ Bucketsort and Compare outperformed all previous Pareto Sum algorithms for Bounded Pareto Sum.

Experimental results

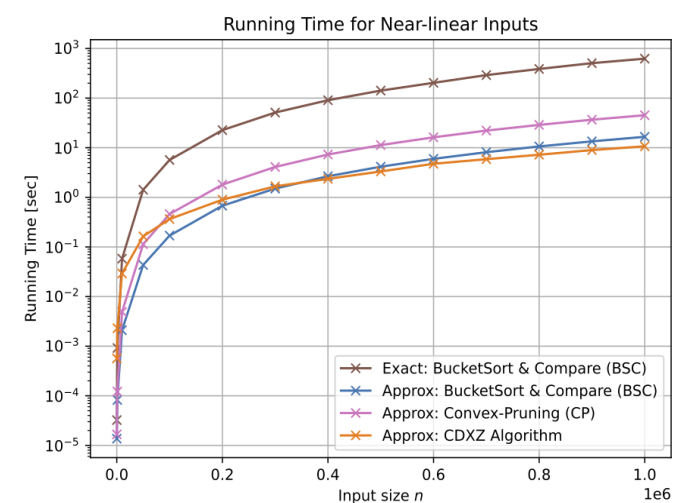
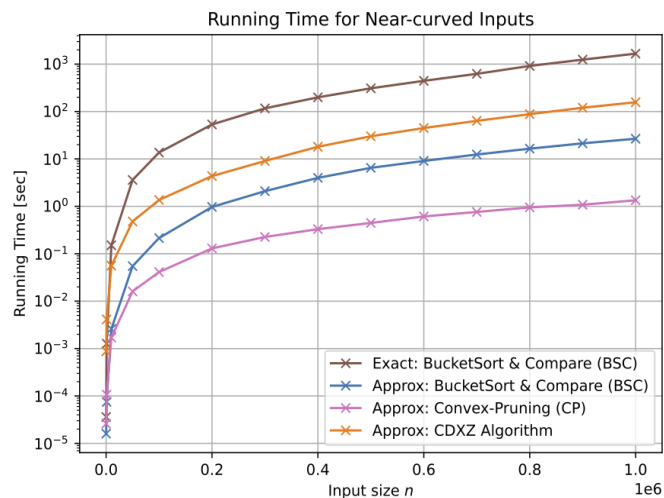
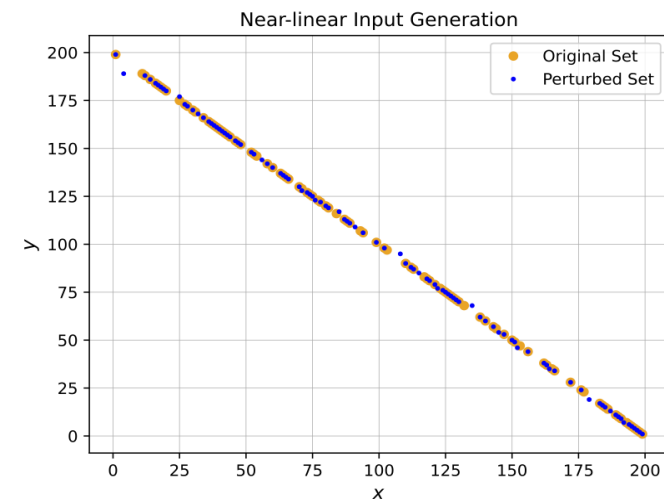
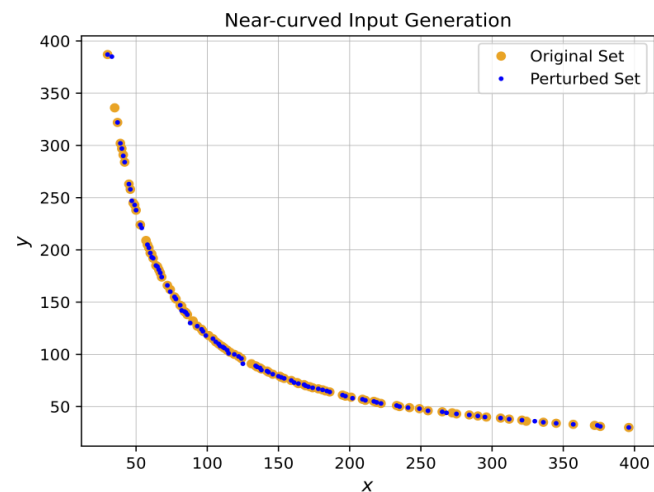
Experimental results



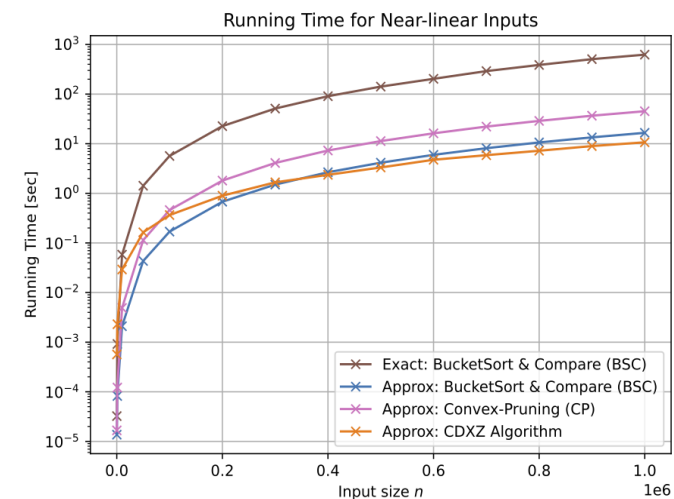
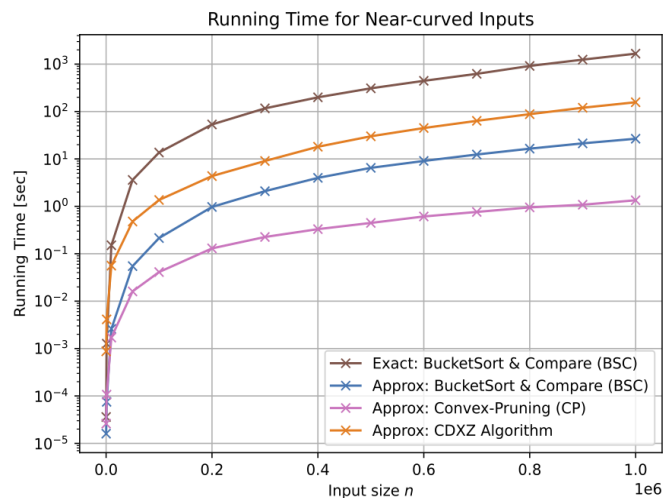
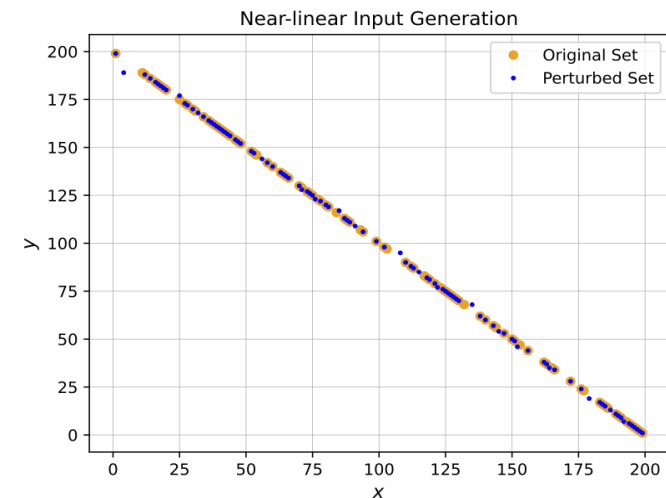
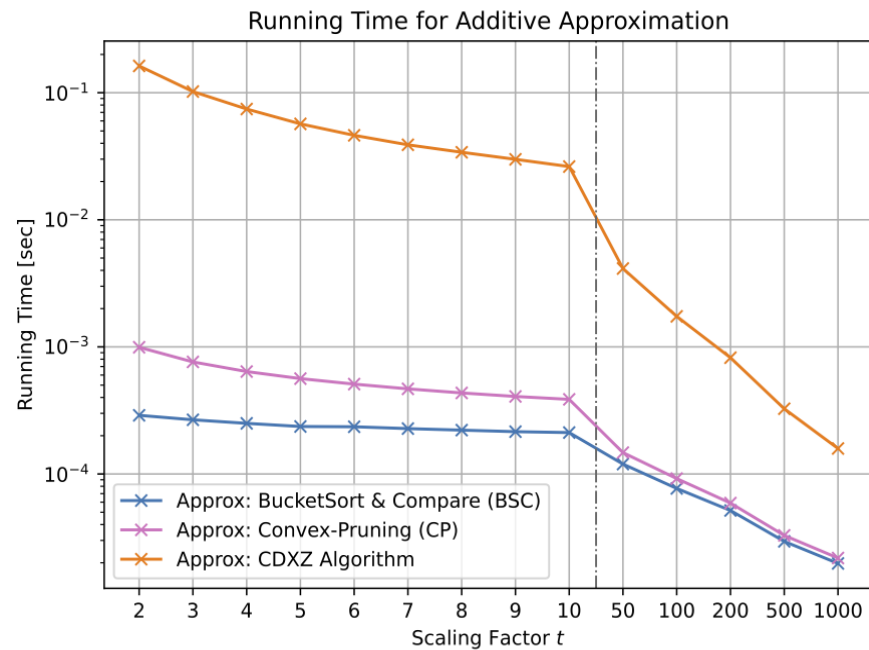
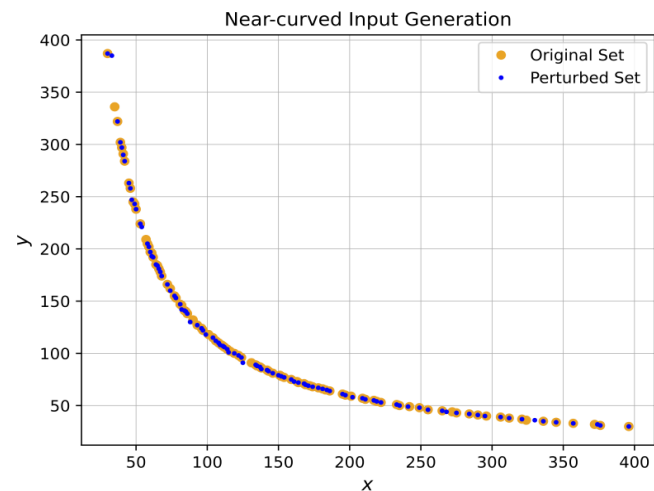
Experimental results



Experimental results



Experimental results



Open questions:

Open questions:

Bounded Pareto Sum.

Can we achieve subquadratic runtime for $d \geq 3$?

Open questions:

Bounded Pareto Sum.

Can we achieve subquadratic runtime for $d \geq 3$?

Further $(\min, +)$ -convolution algorithms.

Can implementation of further $(\min, +)$ -convolution algorithms improve practical runtime?

Open questions:

Bounded Pareto Sum.

Can we achieve subquadratic runtime for $d \geq 3$?

Further $(\min, +)$ -convolution algorithms.

Can implementation of further $(\min, +)$ -convolution algorithms improve practical runtime?

Thanks for listening!

Open questions:

Bounded Pareto Sum.

Can we achieve subquadratic runtime for $d \geq 3$?

Further $(\min, +)$ -convolution algorithms.

Can implementation of further $(\min, +)$ -convolution algorithms improve practical runtime?

Thanks for listening!



Full version of the paper.