Mind the Gap. Doubling Constant Parametrization of Weighted Problems: TSP, Max-Cut, and More

Mihail Stoian

UTN

@STACS'26
March 10, 2026

Disclaimer

- I'm doing a PhD in *database systems* (3rd year).
- Focus: making databases robust with theory.
  - Query speedups on Microsoft's Fabric via join size lower bounds.
  - Using ℓ_p-norms, reverse inner-product inequalities - happy to discuss!

Disclaimer

- I'm doing a PhD in *database systems* (3rd year).
- Focus: making databases robust with theory.
  - Query speedups on Microsoft's Fabric via join size lower bounds.
  - Using ℓ_p-norms, reverse inner-product inequalities - happy to discuss!
- Submitting in TCS whenever I see an (exciting) result missing.
  - Latest: Approximate min-sum subset convolution [WAOA'24].
- Happy for feedback in terms of TCS terminology!

My Motivation

Algebraic Improvements

- Björklund: Hamiltonicity in time $O^*(1.66^n)$ [FOCS 2010].
- "The 4 Scandinavians": Fast Subset Convolution in time $O^*(2^n)$ [STOC 2007].
- Williams: Max-Cut in time $O^*(2^{\omega n/3})$ [ICALP 2004].
- Nešetřil & Poljak: k-Clique in time $O^*(n^{\omega k/3})$ [CMUC 1985].

My Motivation

Algebraic Improvements

- Björklund: Hamiltonicity in time $O^*(1.66^n)$ [FOCS 2010].
- "The 4 Scandinavians": Fast Subset Convolution in time $O^*(2^n)$ [STOC 2007].
- Williams: Max-Cut in time $O^*(2^{\omega n/3})$ [ICALP 2004].
- Nešetřil & Poljak: k-Clique in time $O^*(n^{\omega k/3})$ [CMUC 1985].
- Limitation: Only applicable for unit weights.
- Still *no speedups* for the weighted counterparts, e.g., TSP.

My Motivation

Algebraic Improvements

- Björklund: Hamiltonicity in time $O^*(1.66^n)$ [FOCS 2010].
- "The 4 Scandinavians": Fast Subset Convolution in time $O^*(2^n)$ [STOC 2007].
- Williams: Max-Cut in time $O^*(2^{\omega n/3})$ [ICALP 2004].
- Nešetřil & Poljak: k-Clique in time $O^*(n^{\omega k/3})$ [CMUC 1985].
- Limitation: Only applicable for unit weights.
- Still *no speedups* for the weighted counterparts, e.g., TSP.

"Small Weights"

- (Almost) every single paper has an extra theorem to support *small weights* instances.

My Motivation

"Small Weights" (cont'ed)

We also note that our algorithm can be used to solve TSP with integer weights via self-reducibility at the cost of a runtime blow-up by roughly a factor of the sum of all edges' weights.

Theorem 3 *There is a Monte Carlo algorithm finding the weight of the lightest TSP tour in a positive integer edge weighted graph on* n *vertices in* $O^*(w1.657^n)$ *time, where* w *is the sum of all weights, with error probability exponentially small in* n*.*

Figure 1: Björklund: How to solve TSP via Hamiltonicity.

My Motivation

"Small Weights" (cont'ed)

THEOREM 3. *The subset convolution over the integer max–sum (min–sum) semiring can be computed in* $\tilde{O}(2^n M)$ *time, provided that the range of the input functions is* $\{-M, -M+1, \ldots, M\}$.

Figure 2: The four Scandinavians: How to solve Min-Sum Subset Convolution.

My Motivation: What Everyone Is Using

The "Small Weights" Case

- Solve the problem over the $(\min, +)$ semiring by embedding the input weights into the polynomial ring.
- Run the new algorithm + extract the coefficients & exponents of the solution polynomial.

My Motivation: What Everyone Is Using

The "Small Weights" Case

- Solve the problem over the $(\min, +)$ semiring by embedding the input weights into the polynomial ring.
- Run the new algorithm + extract the coefficients & exponents of the solution polynomial.

$\rightarrow$ Downside: Pseudo-polynomial factors in the running time.

My Motivation: What Everyone Is Using

The "Small Weights" Case

- Solve the problem over the $(\min, +)$ semiring by embedding the input weights into the polynomial ring.
- Run the new algorithm + extract the coefficients & exponents of the solution polynomial.

$\rightarrow$ Downside: Pseudo-polynomial factors in the running time.

Nevertheless, if the input weights are **polynomially bounded**, the running time for the weighted problem = unweighted problem (under O^*-notation).

My Motivation: Research Question

Q: Are polynomially bounded weights the **only** *case where algebraic algorithms avoid the pseudo-polynomial overhead?*

My Motivation: Research Question

Q: Are polynomially bounded weights the **only** *case where algebraic algorithms avoid the pseudo-polynomial overhead?*

A: Apparently not.

Main Result (informal)

If the input weights have **small doubling***, then we can avoid the pseudo-polynomial overhead.*

Algorithm Design $\bowtie$ Additive Combinatorics

Additive combinatorics got pretty popular in the last decade.

Hype: Some Results

- Clustered 3SUM via the (constructive) BSG theorem [CL, STOC'15].
- 3SUM on Sidon sets [JX, STOC'23].
- Subset Sum & Integer Programs under small doubling via the constructive Freiman's theorem [RW, ESA'24].
- Shaving a $O(\sqrt{n})$-time factor for Subset Sum [BFN, SODA'25].

Small Doubling

Sumset. For a set A, define

$$A + A = \{a + b \mid a, b \in A\}$$

Doubling constant.

$$C(A) = \frac{|A + A|}{|A|}$$

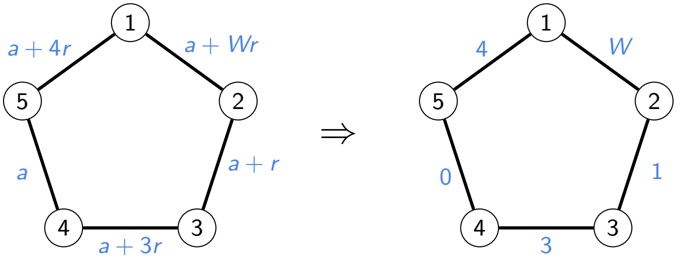
Example.

$$A = \{2, 4, 6, 8\} \qquad A + A = \{4, 6, 8, 10, 12, 14, 16\}$$

$$\Rightarrow C(A) = 7/4 \approx 2.$$

Key intuition: small doubling $\Rightarrow$ strong *additive structure*.

Intuition: TSP with Arithmetic Progressions



Original weights $w \in [0, a + Wr]$ New weights $w' \in [0, W]$

$$w' = \frac{w - a}{r}$$ Recover original cost: $r \cdot \text{cost}' + n \cdot a$.

Beyond Arithmetic Progressions: GAPs

Definition.

$$G = \{x_1\ell_1 + \cdots + x_d\ell_d \mid \ell_i \in [0, L_i]\}$$

- $x_1, \ldots, x_d$: generators (step sizes).
- d: dimension.
- $L_1, \ldots, L_d$: GAP bounds.

Example.

$$G = \{2\ell_1 + 5\ell_2 \mid \ell_1 \in [0, 3],\ \ell_2 \in [0, 2]\}$$

Special case: $d = 1$ gives a simple AP.

Small Doubling $\to$ GAP

Freiman's Theorem [Fre73]. If a finite set A satisfies

$$|A+A| \le C|A|,$$

then A is contained in a GAP

$$G = \{x_1\ell_1 + \cdots + x_d\ell_d \mid \ell_i \in [0, L_i]\}$$

of dimension $d = d(C)$ and size $|G| \le v(C)|A|$.

Both $d(C)$ and $v(C)$ depend only on C.

Constructive Freiman's Theorem

Question

- Can we find the parameters of the GAP?

Constructive Freiman's Theorem

Question

- Can we find the parameters of the GAP?
- Apparently **yes**!

Constructive Freiman's Theorem

Question

- Can we find the parameters of the GAP?
- Apparently **yes**!

Constructive Freiman's Theorem [RW24]. Let A be a set of n integers with $|A+A| \leq C|A|$. Then, there exists an $\tilde{O}_C(n)$-time algorithm that, with probability $1 - n^{-\gamma}$ for an arbitrarily large constant $\gamma > 0$, returns a GAP

$$G = \{x_1\ell_1 + x_2\ell_2 + \cdots + x_{d(C)}\ell_{d(C)} \mid \forall i,\ \ell_i \in [L_i]\} \supseteq A$$

with dimension $d(C)$ and volume $v(C)|A|$.

Meta Algorithm: Key Idea

Sketch (instantiated for TSP)

Let A be the input weights.

$\rightarrow$ All possible solutions are contained in $nA = A + \ldots + A$ (n times).

Meta Algorithm: Key Idea

Sketch (instantiated for TSP)

Let A be the input weights.
$\rightarrow$ All possible solutions are contained in $nA = A + \ldots + A$ (n times).

1. Find parameters GAP $G \supseteq A$.
    - Constructive Freiman's theorem: coefficients x_i and bounds L_i.

Meta Algorithm: Key Idea

Sketch (instantiated for TSP)

Let A be the input weights.
$\rightarrow$ All possible solutions are contained in $nA = A + \ldots + A$ (n times).

1. Find parameters GAP $G \supseteq A$.
   - Constructive Freiman's theorem: coefficients x_i and bounds L_i.
2. Enlarge bounds of G by n. We get $G' := nG \supseteq nA$.

Meta Algorithm: Key Idea

Sketch (instantiated for TSP)

Let A be the input weights.
$\rightarrow$ All possible solutions are contained in $nA = A + \ldots + A$ (n times).

1. Find parameters GAP $G \supseteq A$.
   - Constructive Freiman's theorem: coefficients x_i and bounds L_i.
2. Enlarge bounds of G by n. We get $G' := nG \supseteq nA$.
3. Encode the weights w' via G'.
   - Note: $|G'|$ is polynomial w.r.t. O_C since $|G'| \leq n^{d(C)} v(C)|A|$.

Meta Algorithm: Key Idea

Sketch (instantiated for TSP)

Let A be the input weights.
$\rightarrow$ All possible solutions are contained in $nA = A + \ldots + A$ (n times).

1. Find parameters GAP $G \supseteq A$.
   - Constructive Freiman's theorem: coefficients x_i and bounds L_i.
2. Enlarge bounds of G by n. We get $G' := nG \supseteq nA$.
3. Encode the weights w' via G'.
   - Note: $|G'|$ is polynomial w.r.t. O_C since $|G'| \leq n^{d(C)} v(C)|A|$.
4. Solve the problem on w' with the algebraic algorithm.

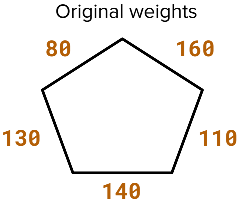
Meta Algorithm: Key Idea

Sketch (instantiated for TSP)

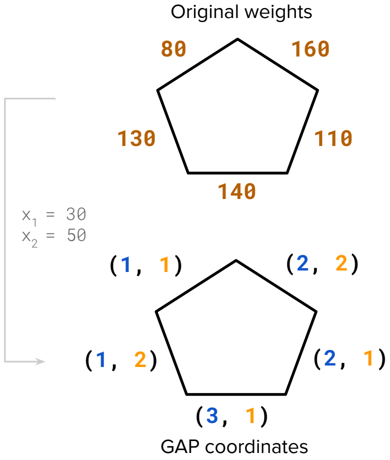
Let A be the input weights.
$\rightarrow$ All possible solutions are contained in $nA = A + \ldots + A$ (n times).

1. Find parameters GAP $G \supseteq A$.
   - Constructive Freiman's theorem: coefficients x_i and bounds L_i.
2. Enlarge bounds of G by n. We get $G' := nG \supseteq nA$.
3. Encode the weights w' via G'.
   - Note: $|G'|$ is polynomial w.r.t. O_C since $|G'| \leq n^{d(C)} v(C)|A|$.
4. Solve the problem on w' with the algebraic algorithm.
5. Decode the solution.

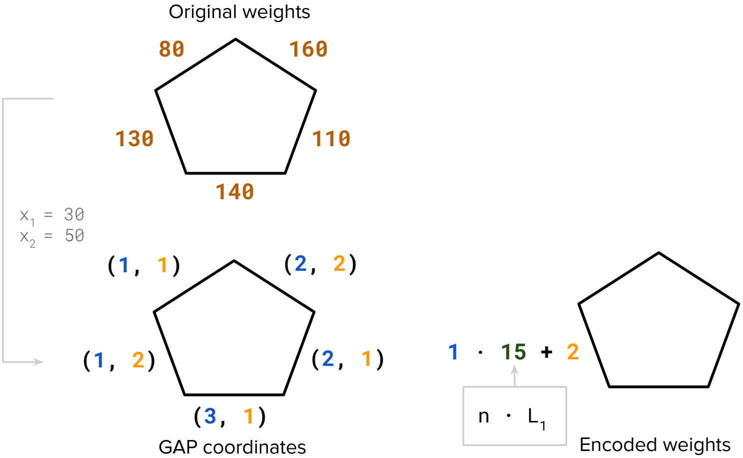
Key Idea: Visualization



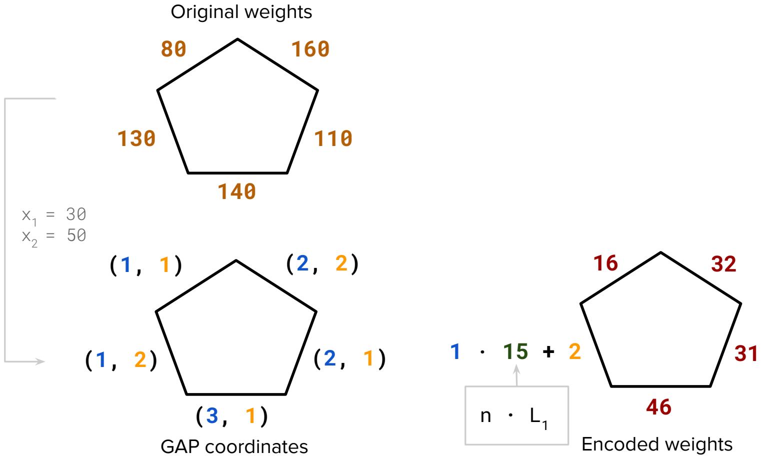
Key Idea: Visualization



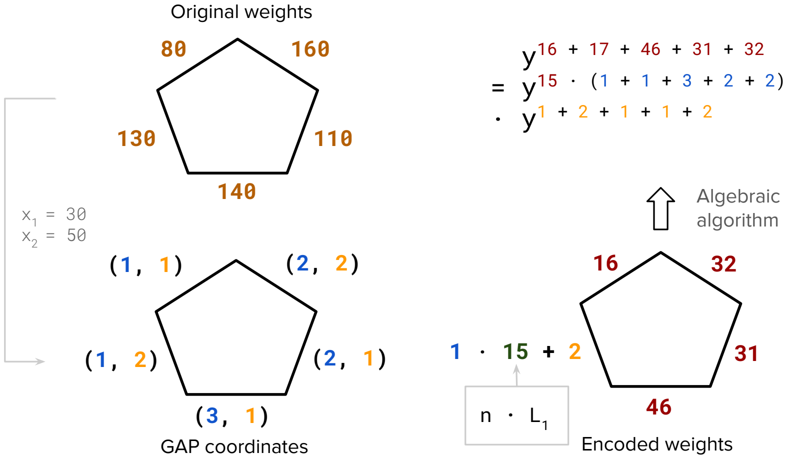
Key Idea: Visualization



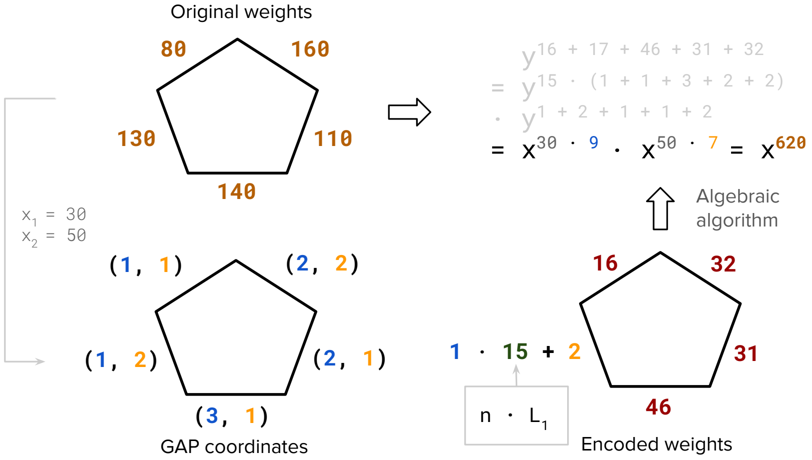
Key Idea: Visualization



Key Idea: Visualization



Key Idea: Visualization



Meta Algorithm: Formalization

Let C-$\mathbf{P}_w$ be the problem where the input weights have doubling constant C. Example:

- C-TSP is TSP with $|w(E) + w(E)| \leq C|w(E)|$.

Theorem. If problem $\mathbf{P}_w$ satisfies property ϕ and the unweighted version can be solved in time $O(T(n))$ by an algebraic algorithm $\mathcal{A}$, then C-$\mathbf{P}_w$ can be solved in time $O^*(T(n))$.

Meta Algorithm: Property ϕ

Property ϕ. Let $\mathbf{P}_w$ be a weighted problem and $\mathbf{P}$ its unweighted counterpart. Let I be an instance of $\mathbf{P}_w$ of size n, $w(I)$ its set of weights, and $W = \max w(I)$. Then, we say that $\mathbf{P}_w$ has property ϕ if the following hold:

Meta Algorithm: Property ϕ

Property ϕ. Let $\mathbf{P}_w$ be a weighted problem and $\mathbf{P}$ its unweighted counterpart. Let I be an instance of $\mathbf{P}_w$ of size n, $w(I)$ its set of weights, and $W = \max w(I)$. Then, we say that $\mathbf{P}_w$ has property ϕ if the following hold:

1. Any feasible solution to I is the total weight of a set $S \subseteq w(I)$,

Meta Algorithm: Property ϕ

Property ϕ. Let $\mathbf{P}_w$ be a weighted problem and $\mathbf{P}$ its unweighted counterpart. Let I be an instance of $\mathbf{P}_w$ of size n, $w(I)$ its set of weights, and $W = \max w(I)$. Then, we say that $\mathbf{P}_w$ has property ϕ if the following hold:

1. Any feasible solution to I is the total weight of a set $S \subseteq w(I)$,
2. There is an algebraic algorithm $\mathcal{A}$ that solves $\mathbf{P}$ in $O(T(n))$-time and $\mathbf{P}_w$ in $O^*(T(n) \cdot W)$-time, and any intermediate solution to I produced by $\mathcal{A}$ is the total weight of a polynomial-size multi-set with support in $w(I)$.

Meta Algorithm: Instantiations

P	$\mathbf{P}_w$	$T(n)$
Hamiltonicity	TSP	1.66^n
Max-Cut	Weighted Max-Cut	$2^{\omega n/3}$
k-Clique	Edge-weighted k-Clique	$n^{\omega k/3}$

Your favorite problem?

Summary & Future Work

- When the set of input weights has **small doubling**, TSP can be solved in the same time as Hamiltonicity.
- Meta algorithm $\to$ application to other weighted problems, e.g., weighted Max-Cut, edge-weighted k-Clique, weighted Steiner tree.

Summary & Future Work

- When the set of input weights has **small doubling**, TSP can be solved in the same time as Hamiltonicity.
- Meta algorithm $\rightarrow$ application to other weighted problems, e.g., weighted Max-Cut, edge-weighted k-Clique, weighted Steiner tree.

Poly-time problems?

- Min-plus convolution—unweighted: $O(n \log n)$, weighted: $O(n^2)$.
- APSP—unweighted: $O(n^\omega)$, weighted: $O(n^3)$.

Summary & Future Work

- When the set of input weights has **small doubling**, TSP can be solved in the same time as Hamiltonicity.
- Meta algorithm $\rightarrow$ application to other weighted problems, e.g., weighted Max-Cut, edge-weighted k-Clique, weighted Steiner tree.

Poly-time problems?

- Min-plus convolution—unweighted: $O(n \log n)$, weighted: $O(n^2)$.
- APSP—unweighted: $O(n^\omega)$, weighted: $O(n^3)$.

Beyond Freiman's theorem

- Currently, Freiman's theorem limited to $C = O(1)$.
  What if $C = O(\alpha(n))$?

Gregory A Freiman.
Foundations of a structural theory of set addition.
Translation of Math. Monographs, 37, 1973.

Tim Randolph and Karol Wegrzycki.
Parameterized Algorithms on Integer Sets with Small Doubling: Integer Programming, Subset Sum and k-SUM.
In Timothy M. Chan, Johannes Fischer, John Iacono, and Grzegorz Herman, editors, *32nd Annual European Symposium on Algorithms, ESA 2024, September 2-4, 2024, Royal Holloway, London, United Kingdom*, volume 308 of *LIPIcs*, pages 96:1–96:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.
doi:10.4230/LIPIcs.ESA.2024.96.