Classical and quantum algorithms for variants of Subset-Sum via dynamic programming

Yassine Hamoudi
Simons Institute

joint work with

Jon Allcock	Antoine Joux	Felix Klingelhöfer	Miklos Santha
Tencent	CISPA	G-SCOP	CQT, NUS

arXiv:2111.07059

The problems

Subset-Sum

Input: Multiset $\{a_1, \ldots, a_n\}$ and target m

Output: Subset $S \subseteq [n]$ such that $\sum_{i \in S} a_i = m$

The problems

SUBSET-SUM

Input: Multiset $\{a_1, \ldots, a_n\}$ and target m

Output: Subset $S \subseteq [n]$ such that $\sum_{i \in S} a_i = m$

SHIFTED-SUMS

Input: Multiset $\{a_1, \ldots, a_n\}$ and shift s

Output: Subsets $S_1 \neq S_2 \subseteq [n]$ s.t. $\sum_{i \in S_1} a_i + s = \sum_{i \in S_2} a_i$

The problems

SUBSET-SUM

Input: Multiset $\{a_1, \ldots, a_n\}$ and target m

Output: Subset $S \subseteq [n]$ such that $\sum_{i \in S} a_i = m$

SHIFTED-SUMS

Input: Multiset $\{a_1, \ldots, a_n\}$ and shift s

Output: Subsets $S_1 \neq S_2 \subseteq [n]$ s.t. $\sum_{i \in S_1} a_i + s = \sum_{i \in S_2} a_i$

EQUAL-SUMS

Assumption: $s = 0$

The problems

Subset-Sum

Input: Multiset $\{a_1, \ldots, a_n\}$ and target m

Output: Subset $S \subseteq [n]$ such that $\sum_{i \in S} a_i = m$

Shifted-Sums

Input: Multiset $\{a_1, \ldots, a_n\}$ and shift s

Output: Subsets $S_1 \neq S_2 \subseteq [n]$ s.t. $\sum_{i \in S_1} a_i + s = \sum_{i \in S_2} a_i$

Equal-Sums

Assumption: $s = 0$

Pigeonhole Equal-Sums

Assumption: $s = 0$ and $\sum_{i=1}^{n} a_i < 2^n - 1$

Results

Prior work	Classical	Quantum
SUBSET-SUM [HS'74,BJLM'13]	$\widetilde{O}(2^{n/2})$	$\widetilde{O}(2^{n/3})$
EQUAL-SUMS [Woe'08,MNPW'19]	$O(2^{0.773n})$	$O(2^{0.529n})$

Results

Prior work	Classical	Quantum
SUBSET-SUM [HS'74,BJLM'13]	$\widetilde{O}(2^{n/2})$	$\widetilde{O}(2^{n/3})$
EQUAL-SUMS [Woe'08,MNPW'19]	$O(2^{0.773n})$	$O(2^{0.529n})$

Our results	Classical	Quantum
SUBSET-SUM (not Meet-in-the-Middle)	$\widetilde{O}(2^{n/2})$	$\widetilde{O}(2^{n/3})$
SHIFTED-SUMS	$O(2^{0.773n})$	$O(2^{0.504n})$
PIGEONHOLE EQUAL-SUMS	$\widetilde{O}(2^{n/2})$	$\widetilde{O}(2^{2n/5})$

Main idea for Shifted-Sums

Representation technique approach:

– Standard in average-case analysis of Subset-Sum [HJ'10]

– Introduced to worst-case analysis of Equal-Sums [MNPW'19]

Main idea for Shifted-Sums

Representation technique approach:

– Standard in average-case analysis of Subset-Sum [HJ'10]

– Introduced to worst-case analysis of Equal-Sums [MNPW'19]

1/ Select p and hash the subset sum values into bins,

$$T_{p,k} = \{S \subseteq \{1, \ldots, n\} : \sum_{i \in S} a_i \equiv k \pmod{p}\}.$$

Main idea for SHIFTED-SUMS

Representation technique approach:

– Standard in average-case analysis of SUBSET-SUM [HJ'10]

– Introduced to worst-case analysis of EQUAL-SUMS [MNPW'19]

1/ Select p and hash the subset sum values into bins,

$$T_{p,k} = \{S \subseteq \{1, \ldots, n\} : \sum_{i \in S} a_i \equiv k \pmod{p}\}.$$

2/ Select k and search for a collision $(S_1, S_2) \in T_{p,k} \times T_{p,k-s \bmod p}$,

$$\sum_{i \in S_1} a_i = s + \sum_{i \in S_2} a_i.$$

Main idea for SHIFTED-SUMS

Representation technique approach:

– Standard in average-case analysis of SUBSET-SUM [HJ'10]

– Introduced to worst-case analysis of EQUAL-SUMS [MNPW'19]

1/ Select p and hash the subset sum values into bins,

$$T_{p,k} = \{S \subseteq \{1, \ldots, n\} : \sum_{i \in S} a_i \equiv k \pmod{p}\}.$$

2/ Select k and search for a collision $(S_1, S_2) \in T_{p,k} \times T_{p,k-s \bmod p}$,

$$\sum_{i \in S_1} a_i = s + \sum_{i \in S_2} a_i.$$

Good choice for p and k:

– the bins are small to keep the cost of search low.

– the bins contain a solution with large probability.

Number of collision values

Collision values:

$$V = \left\{ v \in \mathbb{N} : \exists S_1 \neq S_2, v = \sum_{i \in S_1} a_i = \sum_{i \in S_2} a_i + s \right\}$$

Number of collision values

Collision values:

$$V = \left\{ v \in \mathbb{N} : \exists S_1 \neq S_2, v = \sum_{i \in S_1} a_i = \sum_{i \in S_2} a_i + s \right\}$$

Maximum solution size:

$$\gamma = \max\{|S_1| + |S_2| : S_1, S_2 \text{ disjoint solution}\}$$

Number of collision values

Collision values:

$$V = \left\{ v \in \mathbb{N} : \exists S_1 \neq S_2, v = \sum_{i \in S_1} a_i = \sum_{i \in S_2} a_i + s \right\}$$

Maximum solution size:

$$\gamma = \max\{|S_1| + |S_2| : S_1, S_2 \text{ disjoint solution}\}$$

Lemma: At least $|V| \geq 2^{n-\gamma}$ distinct collision values.

How to find a solution in a good bin?

By quantum element distinctness [Ambainis'07].

How to find a solution in a good bin?

By quantum element distinctness [Ambainis'07].

Let's illustrate when $\gamma = 3n/5$,

How to find a solution in a good bin?

By quantum element distinctness [Ambainis'07].

Let's illustrate when $\gamma = 3n/5$,

- at least $2^{2n/5}$ collision values

How to find a solution in a good bin?

By quantum element distinctness [Ambainis'07].

Let's illustrate when $\gamma = 3n/5$,

- at least $2^{2n/5}$ collision values
- choose prime $p \in [2^{2n/5}, 2^{2n/5+1}]$ and $k \in [0, p-1]$ at random

How to find a solution in a good bin?

By quantum element distinctness [Ambainis'07].

Let's illustrate when $\gamma = 3n/5$,

- at least $2^{2n/5}$ collision values
- choose prime $p \in [2^{2n/5}, 2^{2n/5+1}]$ and $k \in [0, p-1]$ at random
- expected bin size $|T_{p,k}| \approx 2^{3n/5}$ and contains a solution w.h.p.

How to find a solution in a good bin?

By quantum element distinctness [Ambainis'07].

Let's illustrate when $\gamma = 3n/5$,

- at least $2^{2n/5}$ collision values
- choose prime $p \in [2^{2n/5}, 2^{2n/5+1}]$ and $k \in [0, p-1]$ at random
- expected bin size $|T_{p,k}| \approx 2^{3n/5}$ and contains a solution w.h.p.

  $\Rightarrow$ Ambainis' algorithm finds a solution in $\approx 2^{2n/5}$ steps.

How to find a solution in a good bin?

By quantum element distinctness [Ambainis'07].

Let's illustrate when $\gamma = 3n/5$,

- at least $2^{2n/5}$ collision values
- choose prime $p \in [2^{2n/5}, 2^{2n/5+1}]$ and $k \in [0, p-1]$ at random
- expected bin size $|T_{p,k}| \approx 2^{3n/5}$ and contains a solution w.h.p.

$\Rightarrow$ Ambainis' algorithm finds a solution in $\approx 2^{2n/5}$ steps.

Difficulty: each step requires one (quantum) query to $T_{p,k}$

$\Rightarrow$ for some indexing of $T_{p,k} = \{S_1, \ldots, S_{|T_{p,k}|}\}$ we need to implement the oracle

$$O_{T_{p,k}} : \ell \mapsto S_\ell, \quad \text{for} \;\; 1 \leq \ell \leq |T_{p,k}|$$

Dynamic programming data structure

Compute the cardinality table

$$t_p[i,j] = |\{S \subseteq \{1,\ldots,i\} : \sum_{i \in S} a_i \equiv j \pmod{p}\}|$$

by dynamic programming in time $O(np)$.

Dynamic programming data structure

Compute the cardinality table

$$t_p[i, j] = |\{S \subseteq \{1, \ldots, i\} : \sum_{i \in S} a_i \equiv j \pmod{p}\}|$$

by dynamic programming in time $O(np)$.

Definition: Denote $\prec$ the strict total order defined as

$S_1 \prec S_2$ if and only if $\max\{i : i \in S_1 \Delta S_2\} \in S_2$.

Dynamic programming data structure

Compute the cardinality table

$$t_p[i,j] = |\{S \subseteq \{1,\ldots,i\} : \sum_{i \in S} a_i \equiv j \pmod{p}\}|$$

by dynamic programming in time $O(np)$.

Definition: Denote $\prec$ the strict total order defined as

$$S_1 \prec S_2 \text{ if and only if } \max\{i : i \in S_1 \Delta S_2\} \in S_2.$$

Theorem: Given $1 \leq \ell \leq t_{p,k}$ and random access to the elements of the table t_p, the ℓ-th set in $T_{p,k}$ can be computed in time $O(n)$.

Computing the oracle

Input: Table t_p, integers $0 \leq k \leq p-1$ and $1 \leq \ell \leq t_{p,k}$
Output: ℓ-th set $S \in T_{p,k}$ for $\prec$.

❶ $j = k, S = \varnothing$

❷ **for** $i = n, \ldots, 1$ **do**

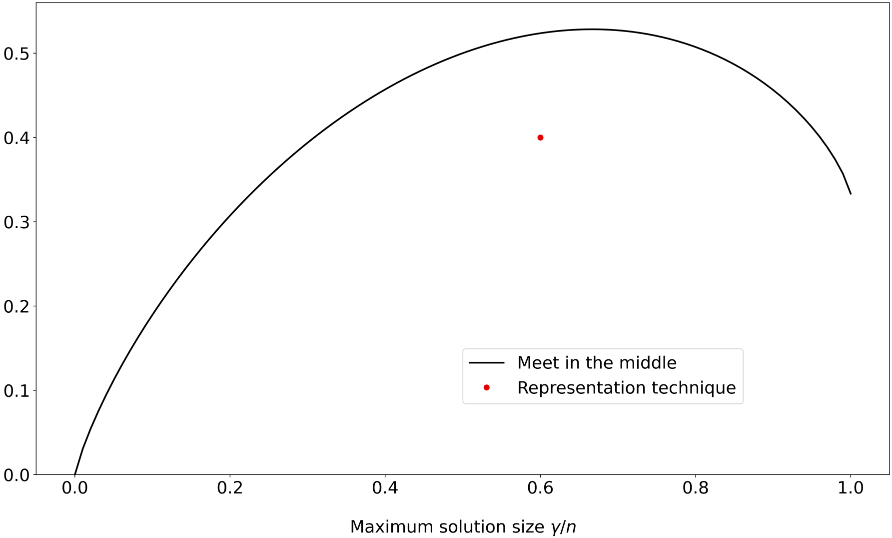
❸ **if** $\ell \leq t_p[i-1, j]$ **then** Do nothing

❹ **else**

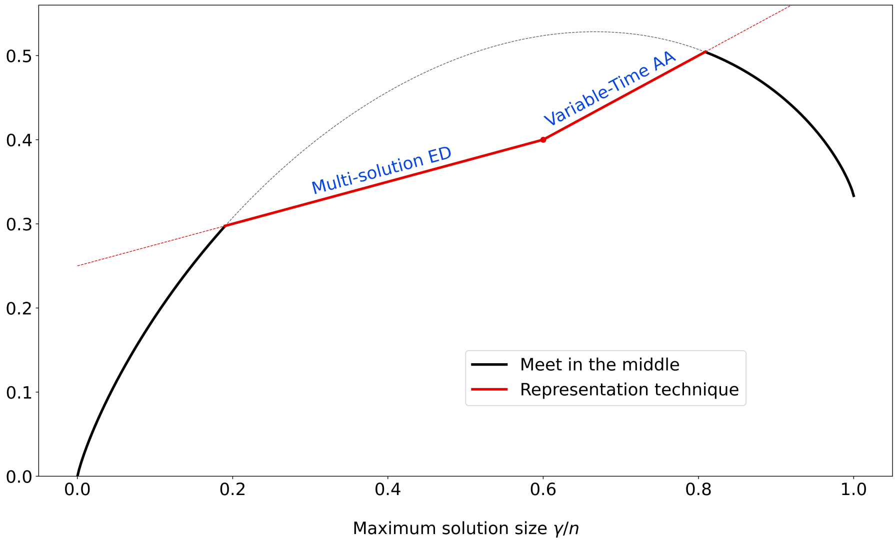
❺ $S = S \cup \{i\}, \quad \ell = \ell - t_p[i-1, j], \quad j = j - a_i \mod p$

❻ Return S

Running time exponent



Running time exponent



Open problem

PIGEONHOLE MODULAR EQUAL-SUMS
Input: Set $\{a_1, \ldots, a_n\}$ and a modulus q s.t. $q \leq 2^n - 1$
Output: Subsets $S_1 \neq S_2 \subseteq [n]$ s.t. $\sum_{i \in S_1} a_i \equiv_q \sum_{i \in S_2} a_i$

Generalizes PIGEONHOLE EQUAL-SUMS

Theorem: Can be solved deterministically in time $\widetilde{O}(2^{n/2})$

Question: Can be solved quantumly in time $\widetilde{O}(2^{2n/5})$?