Harvey Mudd College Computer Science

Additive Partition Problems on Sets

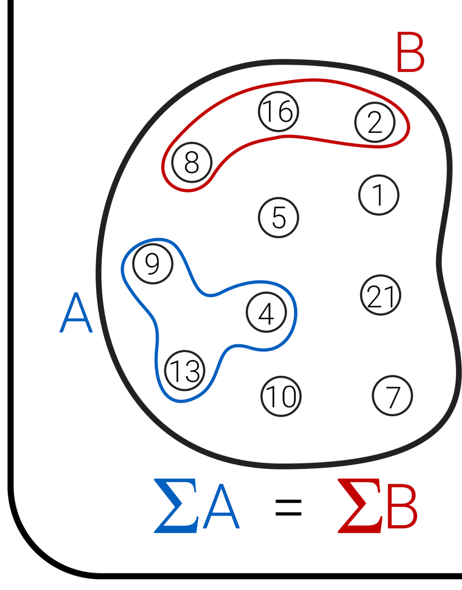
Motivating Questions:

- How hard is it to partition a set into equal subsets in the worst case?
- How well can we do on average?

Why might we want to answer these questions?

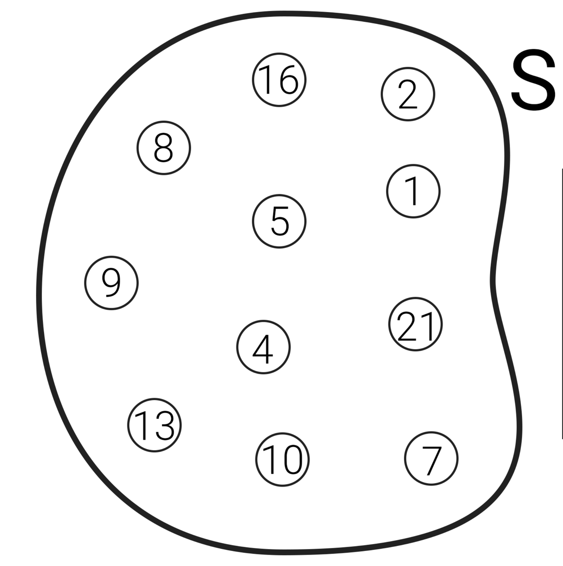
Subset and partition problems have several applications in cryptography, optimization problems, etc. However, more importantly, designing efficient algorithms to solve these problems compels us to look at the additive structure underlying these sets and exposes interesting mathematical properties of integer sets and their partitions.

Definitions for our two research problems



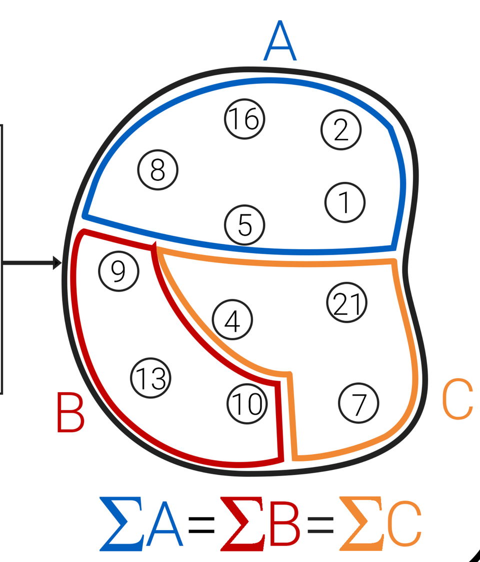
Equal-Subset-Sum

Find two disjoint subsets of S whose elements have the same sum

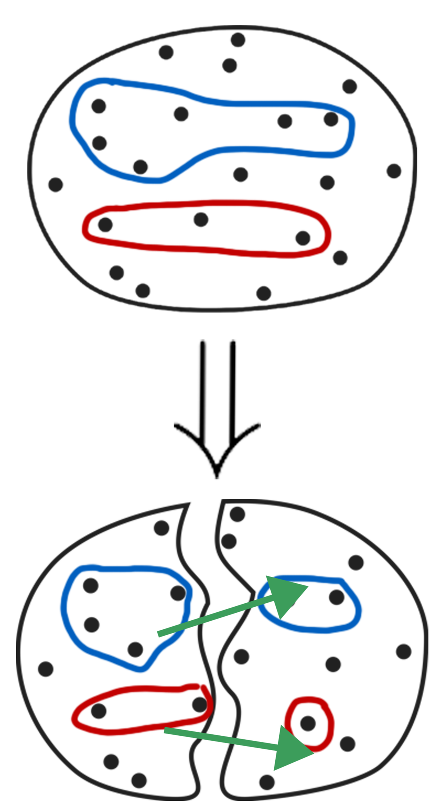


3-Multiway Partition

Partition S into three parts so that each have equal sum



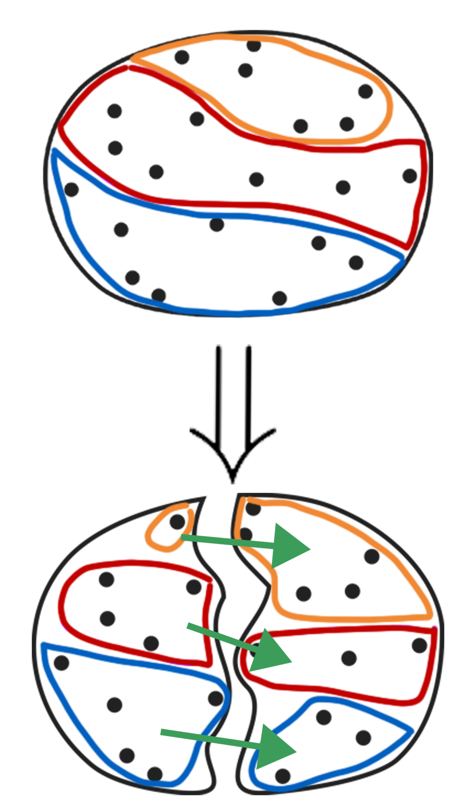
The benchmark algorithm for partition problems



The first method we can apply to these problems is a brute force approach called Meet-in-the-Middle. Developed by Horowitz and Sahni [1], this powerful general technique solves problems that involve enumerating subsets.

Naively, there are 3^n ways to partition the n elements of S in each of the problems. However, Meet-in-the-Middle improves upon this by recognizing that for any partial solution (incomplete piece of a solution), there is a unique sum that can complete this solution.

So, Meet-in-the-Middle divides a set in half and enumerates the partitions for each half. This brings the number of partitions to $2 \cdot 3^{n/2}$ —a massive improvement. Finally, using an efficient sort-and-search, complimentary pieces of solutions are matched. So, the method is guaranteed to produce a solution if one exists.



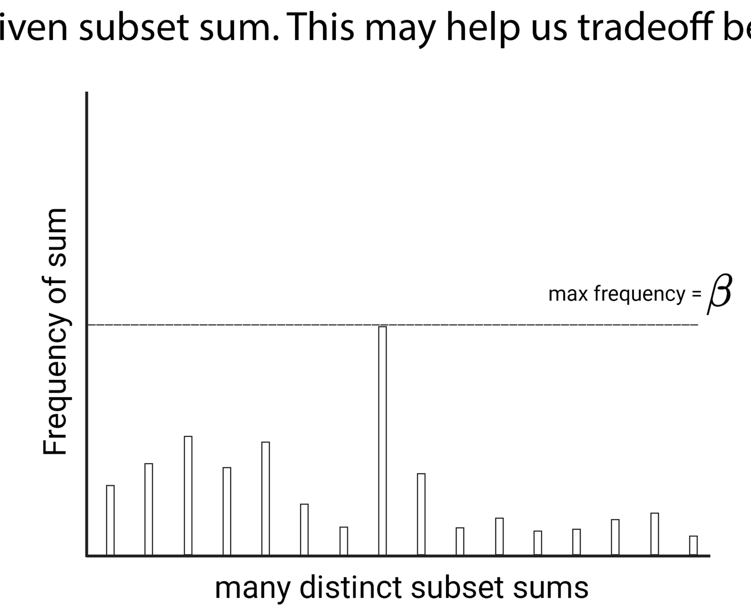
Meet-in-the-Middle runs in time $O^*(3^{n/2})$ and it serves as a strong benchmark to measure new algorithms against. So, to search for improvements, we wanted to consider if we could get around exhaustively enumerating partitions.

In our domain of 'Exact Exponential Algorithms', randomized techniques have canonically offered improvements over brute force. Randomized algorithms rely on proving a lower-bound on the probability of finding a solution, and then using randomized trials to generate candidate solutions. It is here that being in the NP-Complete space is useful, because candidate solutions can be quickly verified for correctness.

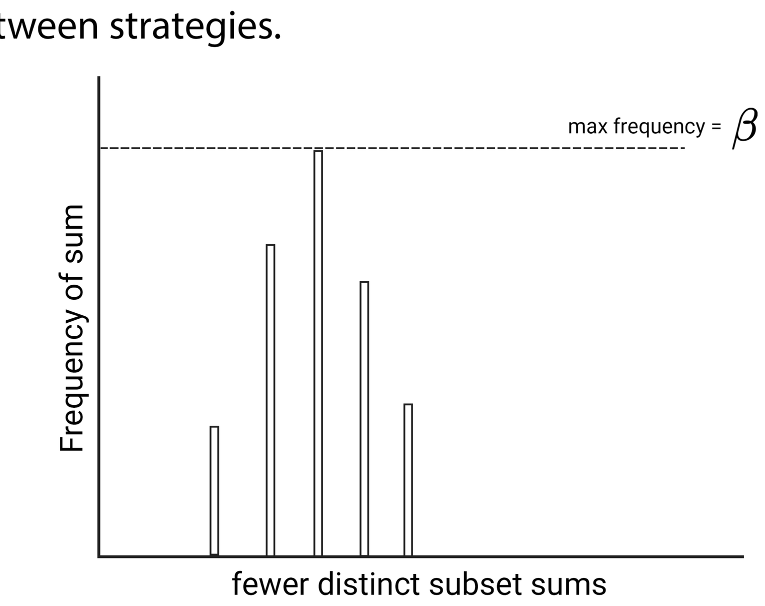
A powerful probabilistic technique we studied was the Representation Method, proposed by Graham and Joux [2] in their paper on hard knapsack problems.

A useful relationship between distinct sums and most frequent sums

A potentially useful result proved by Austrin et. al. [4] creates a bounding relationship between the number of subset sums that can be formed and the maximum frequency of any given subset sum. This may help us tradeoff between strategies.



If there are many distinct sums
⇒ the maximum frequency must be lower.



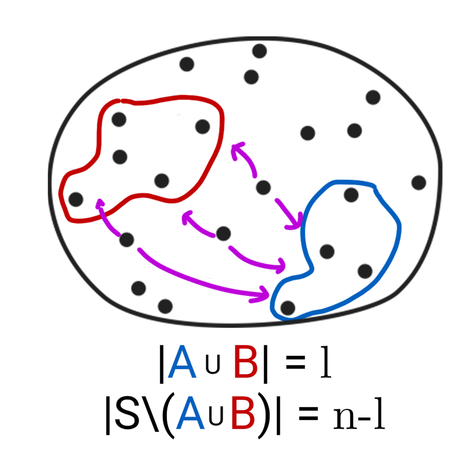
If one sum occurs very frequently
⇒ there must be fewer distinct sums

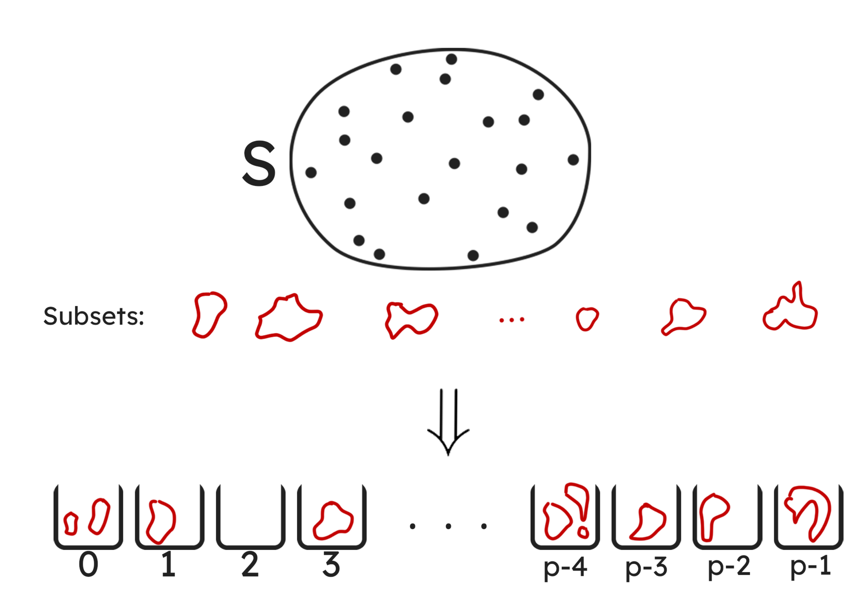
Representation Method: improving on the benchmark for ESS using a randomized approach

Suppose that out of the n elements in an ESS instance, l are included in the solution (A or B). Then, there are n-l elements that are not in the solution (non-solution).

Observation: If ΣA=ΣB, you can add any non-solution element to both A and B, and their sums will still be the same. So, there are at least 2^{n-l} subsets where ΣA=ΣB.

Under a scheme where we assume $l > n/2$ and that this solution size is miminum, it can be shown that these new sums are all unique.





Knowing that there exist these exponentially many subsets with equal sum, we construct a filter that has sufficient probability of finding one such pair.

We do so by choosing a large prime, and sorting the subset sums into buckets, based on their resiude class of this prime. It was proved by Mucha et. al [3] that this has sufficient success probability.

Then, by only enumerating the subsets of a bucket chosen at random, we expect to eventually find two subsets with a collision of sums.

3-Multiway Partition: Improvements in the average case setting

Why is this problem interesting?

- The general case of MP3 may have specific pathological instances which are hard to solve.
- In both Subset-Sum and Equal-Subset-Sum, past research has shown improvements in the average-case, so we want to explore it in the MP3 setting [2,5].

Characterizing the average-case input

- Let our input set S be n elements chosen uniformly at random from the integers $\{1, 2, \ldots, M\}$.
- Similar to above, let β here be defined as the number of subsets of S such that their sum is ∑S/3.

Tradeoff strategy based on input parameters

- We can choose optimal strategies based on different values of our input parameters, M and β.
- Because we only care about the exponential term and not the polynomial, we can run our algorithm repeatedly over different guessed parameter choices (assuming they are in *poly(n)*).

Input: $S = [\, s_1\ s_2\ s_3 \ldots s_n \,]$

$M < 2^{0.38n}$ ⇓ Conjecture: can be solved with dynamic programming in time: $O^*(M^2) < O^*(2^{0.76n})$

$M > 3^{n/2}$ ⇓ Theorem: probability of a solution exponentially approaches 0 for large n

β is small ⇓ Conjecture: can be solved by enumerating subsets with sum ∑S/3 and finding disjoint pairs in time: $O^*((2^{0.45n}) + \min(2^{0.33n}\beta, \beta^2))$

β is big ⇓ Conjecture: For $2^{0.38n} < M < 3^{n/2}$, β is "big" with exponentially small probability over the input

Works Cited

[1] Horowitz, Ellis, and Sartaj Sahni, Computing Partitions with Applications to the Knapsack Problem, J. ACM, 21 (1974), no. 2, 277–292.

[2] Howgrave-Graham, Nick, and Antoine Joux, New Generic Algorithms for Hard Knapsacks, in Advances in Cryptology -- EUROCRYPT 2010, ed. Henri Gilbert, Springer, Berlin, Heidelberg, 2010, pp. 235–256.

[3] Mucha, Marcin, Jesper Nederlof, Jakub Pawlewicz, and Karol Węgrzycki, Equal-subset-sum faster than the meet-in-the-middle, arXiv preprint arXiv:1905.02424, 2019.

[4] Austrin, Per, Mikko Koivisto, Petteri Kaski, and Jesper Nederlof, Dense subset sum may be the hardest, arXiv preprint arXiv:1508.06019, 2015.

[5] Chen, Xi, Yaonan Jin, Tim Randolph, and Rocco A. Servedio, Average-case subset balancing problems, in Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, 2022, pp. 743–778.

Contributors

Team Members: Timothy Randolph, Uddant Patodia, Xiaofeng Li

Acknowledgements: Riley Brown, Theodore Julien, Mia Alexander, HMC CS Department, Chelsey Calingo, Kevin Herera, Landon Tu, Selene Ye, Bill Lenhart, Karol Węgrzycki