

Partition problem

In number theory and computer science, the **partition problem**, or **number partitioning**,^[1] is the task of deciding whether a given multiset S of positive integers can be partitioned into two subsets S_1 and S_2 such that the sum of the numbers in S_1 equals the sum of the numbers in S_2 . Although the partition problem is NP-complete, there is a pseudo-polynomial time dynamic programming solution, and there are heuristics that solve the problem in many instances, either optimally or approximately. For this reason, it has been called "the easiest hard problem".^{[2][3]}

There is an optimization version of the partition problem, which is to partition the multiset S into two subsets S_1 , S_2 such that the difference between the sum of elements in S_1 and the sum of elements in S_2 is minimized. The optimization version is NP-hard, but can be solved efficiently in practice.^[4]

The partition problem is a special case of two related problems:

- In the subset sum problem, the goal is to find a subset of S whose sum is a certain target number T given as input (the partition problem is the special case in which T is half the sum of S).
- In multiway number partitioning, there is an integer parameter k , and the goal is to decide whether S can be partitioned into k subsets of equal sum (the partition problem is the special case in which $k = 2$).
- However, it is quite different than the 3-partition problem: in that problem, the number of subsets is not fixed in advance - it should be $|S|/3$, where each subset must have exactly 3 elements. 3-partition is much harder than partition - it has no pseudo-polynomial time algorithm unless **P = NP**.^[5]

Contents

Examples

Computational hardness

Approximation algorithms

Exact algorithms

Hard instances and phase-transition

Probabilistic version

Variants and generalizations

Applications

Notes

References

Examples

Given $S = \{3,1,1,2,2,1\}$, a valid solution to the partition problem is the two sets $S_1 = \{1,1,1,2\}$ and $S_2 = \{2,3\}$. Both sets sum to 5, and they partition S . Note that this solution is not unique. $S_1 = \{3,1,1\}$ and $S_2 = \{2,2,1\}$ is another solution.

Not every multiset of positive integers has a partition into two subsets with equal sum. An example of such a set is $S = \{2,5\}$.

Computational hardness

The partition problem is NP hard. This can be proved by reduction from the subset sum problem.^[6] An instance of SubsetSum consists of a set S of positive integers and a target sum $T < S$; the goal is to decide if there is a subset of S with sum exactly T .

Given such an instance, construct an instance of Partition in which the input set contains the original set plus two elements: z_1 and z_2 , with $z_1 = \text{sum}(S)$ and $z_2 = 2T$. The sum of this input set is $\text{sum}(S) + z_1 + z_2 = 2\text{sum}(S) + 2T$, so the target sum for Partition is $\text{sum}(S) + T$.

- Suppose there exists a solution S' to the SubsetSum instance. Then $\text{sum}(S') = T$, so $\text{sum}(S' \cup \{z_1\}) = \text{sum}(S) + T$, so $S' \cup \{z_1\}$ is a solution to the Partition instance.
- Conversely, suppose there exists a solution S'' to the Partition instance. Then, S'' must contain either z_1 or z_2 , but not both, since their sum is more than $\text{sum}(S) + T$. If S'' contains z_1 , then it must contain elements from S with a sum of exactly T , so S'' minus z_1 is a solution to the SubsetSum instance. If S'' contains z_2 , then it must contain elements from S with a sum of exactly $\text{sum}(S) - T$, so the other objects in S are a solution to the SubsetSum instance.

Approximation algorithms

As mentioned above, the partition problem is a special case of multiway-partitioning and of subset-sum. Therefore, it can be solved by algorithms developed for each of these problems. Algorithms developed for multiway number partitioning include:

- **Greedy number partitioning** - loops over the numbers, and puts each number in the set whose current sum is smallest. If the numbers are not sorted, then the runtime is $O(n)$ and the approximation ratio is at most **3/2** ("approximation ratio" means the larger sum in the algorithm output, divided by the larger sum in an optimal partition). Sorting the numbers increases the runtime to $O(n \log n)$ and improves the approximation ratio to **7/6**. If the numbers are distributed uniformly in $[0,1]$, then the approximation ratio is at most $1 + O(\log \log n / n)$ almost surely, and $1 + O(1/n)$ in expectation.
- **Largest Differencing Method** (also called the **Karmarkar-Karp algorithm**) sorts the numbers in descending order and repeatedly replaces numbers by their differences. The runtime complexity is $O(n \log n)$. In the worst case, its approximation ratio is similar - at most **7/6**. However, in the average case it performs much better than the greedy algorithm: when numbers are distributed uniformly in $[0,1]$, its approximation ratio is at most $1 + 1/n^{\Theta(\log n)}$ in expectation. It also performs better in simulation experiments.
- The **Multifit algorithm** uses binary search combined with an algorithm for bin packing. In the worst case, its approximation ratio is **8/7**.
- The subset sum problem has an **FPTAS** which can be used for the partition problem as well, by setting the target sum to $\text{sum}(S)/2$.

Exact algorithms

There are exact algorithms, that always find the optimal partition. Since the problem is NP-hard, such algorithms might take exponential time in general, but may be practically usable in certain cases. Algorithms developed for multiway number partitioning include:

- The pseudopolynomial time number partitioning takes $O(nm)$ memory, where m is the largest number in the input.
- The **Complete Greedy Algorithm (CGA)** considers all partitions by constructing a binary tree. Each level in the tree corresponds to an input number, where the root corresponds to the largest number, the level below to the next-largest number, etc. Each branch corresponds to a different set in which the current number can be put. Traversing the tree in depth-first order requires only $O(n)$ space, but might take $O(2^n)$ time. The runtime can be improved by using a greedy heuristic: in each level, develop first the branch in which the current number is put in the set with the smallest sum. This algorithm finds first the solution found by greedy number partitioning, but then proceeds to look for better solutions. Some variations of this idea are fully polynomial-time approximation schemes for the subset-sum problem, and hence for the partition problem as well.^{[7][8]}
- The **Complete Karmarkar-Karp algorithm (CKK)** considers all partitions by constructing a binary tree. Each level corresponds to a pair of numbers. The left branch corresponds to putting them in different subsets (i.e., replacing them by their difference), and the right branch corresponds to putting them in the same subset (i.e., replacing them by their sum). This algorithm finds first the solution found by the largest differencing method, but then proceeds to find better solutions. It runs substantially faster than CGA on random instances. Its advantage is much larger when an equal partition exists, and can be of several orders of magnitude. In practice, problems of arbitrary size can be solved by CKK if the numbers have at most 12 significant digits.^[9] CKK can also run as an anytime algorithm: it finds the KK solution first, and then finds progressively better solutions as time allows (possibly requiring exponential time to reach optimality, for the worst instances).^[1] It requires $O(n)$ space, but in the worst case might take $O(2^n)$ time.

Algorithms developed for subset sum include:

- **Horowitz and Sanhi** - runs in time $O(2^{n/2} \cdot (n/2))$, but requires $O(2^{n/2})$ space.
- **Schroeppel and Shamir** - runs in time $O(2^{n/2} \cdot (n/4))$, and requires much less space - $O(2^{n/4})$.
- **Howgrave-Graham and Joux** - runs in time $O(2^{n/3})$, but it is a randomized algorithm that only solves the decision problem (not the optimization problem).

Hard instances and phase-transition

Sets with only one, or no partitions tend to be hardest (or most expensive) to solve compared to their input sizes. When the values are small compared to the size of the set, perfect partitions are more likely. The problem is known to undergo a "phase transition"; being likely for some sets and unlikely for others. If m is the number of bits needed to express any number in the set and n is the size of the set then $m/n < 1$ tends to have many solutions and $m/n > 1$ tends to have few or no solutions. As n and m get larger, the probability of a perfect partition goes to 1 or 0 respectively. This was originally argued based on empirical evidence by Gent and Walsh,^[10] then using methods from statistical physics by Mertens,^{[11][12]} and later proved by Borgs, Chayes, and Pittel.^[13]

Probabilistic version

A related problem, somewhat similar to the Birthday paradox, is that of determining the size of the input set so that we have a probability of one half that there is a solution, under the assumption that each element in the set is randomly selected with uniform distribution between 1 and some given value. The solution to this problem can be counter-intuitive, like the birthday paradox.

Variants and generalizations

Equal-cardinality partition is a variant in which both parts should have an equal number of items, in addition to having an equal sum. This variant is NP-hard too; it is problem [SP12] in.^[5]

Distinct partition is a variant in which all input integers are distinct. This variant is NP-hard too.

Product partition is the problem of partitioning a set of integers into two sets with the same *product* (rather than the same sum). This problem is strongly NP-hard.^[14]

Kovalyov and Pesch^[15] discuss a generic approach to proving NP-hardness of partition-type problems.

Applications

One application of the partition problem is for manipulation of elections. Suppose there are three candidates (A, B and C). A single candidate should be elected using a voting rule based on scoring, e.g. the veto rule (each voter vetos a single candidate and the candidate with the fewest vetos wins). If a coalition wants to ensure that C is elected, they should partition their votes among A and B so as to maximize the smallest number of vetoes each of them gets. If the votes are weighted, then the problem can be reduced to the partition problem, and thus it can be solved efficiently using CKK. The same is true for any other voting rule that is based on scoring.^[16]

Notes

1. Korf 1998.
2. Hayes, Brian (March–April 2002), "The Easiest Hard Problem" (<http://bit-player.org/bph-publications/AmSci-2002-03-Hayes-NPP.pdf>) (PDF), *American Scientist*, Sigma Xi, The Scientific Research Society, vol. 90 no. 2, pp. 113–117, JSTOR 27857621 (<https://www.jstor.org/stable/27857621>)
3. Mertens 2006, p. 125 (<https://books.google.com/books?id=4YD6AxV95zEC&pg=PA125>).
4. Korf, Richard E. (2009). *Multi-Way Number Partitioning* (<http://ijcai.org/papers09/Papers/IJCAI09-096.pdf>) (PDF). IJCAI.
5. Garey, Michael; Johnson, David (1979). *Computers and Intractability; A Guide to the Theory of NP-Completeness* (<https://archive.org/details/computersintract0000gare>). pp. 96–105 (<https://archive.org/details/computersintract0000gare/page/96>). ISBN 978-0-7167-1045-5.
6. Goodrich, Michael. "More NP complete and NP hard problems" (<https://www.ics.uci.edu/~goodrich/teach/cs162/notes/pnp3.pdf>) (PDF).
7. Hans Kellerer; Ulrich Pferschy; David Pisinger (2004). *Knapsack problems* (<https://books.google.com/books?id=u5DB7gck08YC&pg=PA97>). Springer. p. 97. ISBN 9783540402862.

8. Martello, Silvano; Toth, Paolo (1990). "4 Subset-sum problem" (<https://archive.org/details/knapsackproblems0000mart/page/105>). *Knapsack problems: Algorithms and computer interpretations*. Wiley-Interscience. pp. 105–136 (<https://archive.org/details/knapsackproblems0000mart/page/105>). ISBN 978-0-471-92420-3. MR 1086874 (<https://www.ams.org/mathscinet-getitem?mr=1086874>).
9. Korf, Richard E. (1995-08-20). "From approximate to optimal solutions: a case study of number partitioning" (<https://dl.acm.org/doi/abs/10.5555/1625855.1625890>). *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. IJCAI'95. 1. Montreal, Quebec, Canada: Morgan Kaufmann Publishers. pp. 266–272. ISBN 978-1-55860-363-9.
10. Gent & Walsh 1996.
11. Mertens 1998.
12. Mertens 2001, p. 130.
13. Borgs, Chayes & Pittel 2001.
14. Ng, C. T.; Barketau, M. S.; Cheng, T. C. E.; Kovalyov, Mikhail Y. (2010-12-01). "'Product Partition' and related problems of scheduling and systems reliability: Computational complexity and approximation" (<http://www.sciencedirect.com/science/article/pii/S0377221710003905>). *European Journal of Operational Research*. **207** (2): 601–604. doi:10.1016/j.ejor.2010.05.034 (<https://doi.org/10.1016%2Fj.ejor.2010.05.034>). ISSN 0377-2217 (<https://www.worldcat.org/issn/0377-2217>).
15. Kovalyov, Mikhail Y.; Pesch, Erwin (2010-10-28). "A generic approach to proving NP-hardness of partition type problems" (<https://www.sciencedirect.com/science/article/pii/S0166218X10002647>). *Discrete Applied Mathematics*. **158** (17): 1908–1912. doi:10.1016/j.dam.2010.08.001 (<https://doi.org/10.1016%2Fj.dam.2010.08.001>). ISSN 0166-218X (<https://www.worldcat.org/issn/0166-218X>).
16. Walsh, Toby (2009-07-11). "Where are the really hard manipulation problems? the phase transition in manipulating the veto rule" (<https://dl.acm.org/doi/abs/10.5555/1661445.1661497>). *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. IJCAI'09. Pasadena, California, USA: Morgan Kaufmann Publishers. pp. 324–329.

References

- Borgs, Christian; Chayes, Jennifer; Pittel, Boris (2001), "Phase transition and finite-size scaling for the integer partitioning problem", *Random Structures and Algorithms*, **19** (3–4): 247–288, CiteSeerX 10.1.1.89.9577 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.89.9577>), doi:10.1002/rsa.10004 (<https://doi.org/10.1002%2Frsa.10004>)
- Gent, Ian; Walsh, Toby (August 1996). "Phase Transitions and Annealed Theories: Number Partitioning as a Case Study". In Wolfgang Wahlster (ed.). *Proceedings of 12th European Conference on Artificial Intelligence*. ECAI-96. John Wiley and Sons. pp. 170–174. CiteSeerX 10.1.1.2.4475 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.4475>).
- Gent, Ian; Walsh, Toby (1998), "Analysis of Heuristics for Number Partitioning", *Computational Intelligence*, **14** (3): 430–451, CiteSeerX 10.1.1.149.4980 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.149.4980>), doi:10.1111/0824-7935.00069 (<https://doi.org/10.1111%2F0824-7935.00069>), S2CID 15344203 (<https://api.semanticscholar.org/CorpusID:15344203>)
- Korf, Richard E. (1998), "A complete anytime algorithm for number partitioning", *Artificial Intelligence*, **106** (2): 181–203, CiteSeerX 10.1.1.90.993 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.90.993>), doi:10.1016/S0004-3702(98)00086-1 (<https://doi.org/10.1016%2FS0004-3702%2898%2900086-1>), ISSN 0004-3702 (<https://www.worldcat.org/issn/0004-3702>)
- Mertens, Stephan (November 1998), "Phase Transition in the Number Partitioning Problem", *Physical Review Letters*, **81** (20): 4281–4284, arXiv:cond-mat/9807077 (<https://arxiv.org/abs/>)

cond-mat/9807077), Bibcode:1998PhRvL..81.4281M (<https://ui.adsabs.harvard.edu/abs/1998PhRvL..81.4281M>), doi:10.1103/PhysRevLett.81.4281 (<https://doi.org/10.1103%2FPhysRevLett.81.4281>), S2CID 119541289 (<https://api.semanticscholar.org/CorpusID:119541289>)

- Mertens, Stephan (2001), "A physicist's approach to number partitioning", *Theoretical Computer Science*, **265** (1–2): 79–108, arXiv:cond-mat/0009230 (<https://arxiv.org/abs/cond-mat/0009230>), doi:10.1016/S0304-3975(01)00153-0 (<https://doi.org/10.1016%2FS0304-3975%2801%2900153-0>), S2CID 16534837 (<https://api.semanticscholar.org/CorpusID:16534837>)
- Mertens, Stephan (2006). "The Easiest Hard Problem: Number Partitioning" (<https://books.google.com/books?id=4YD6AxV95zEC&pg=PA125>). In Allon Percus; Gabriel Istrate; Cristopher Moore (eds.). *Computational complexity and statistical physics*. USA: Oxford University Press. pp. 125–140. arXiv:cond-mat/0310317 (<https://arxiv.org/abs/cond-mat/0310317>). Bibcode:2003cond.mat.10317M (<https://ui.adsabs.harvard.edu/abs/2003cond.mat.10317M>). ISBN 9780195177374.
- Mertens, Stephan (1999), "A complete anytime algorithm for balanced number partitioning", arXiv:cs/9903011 (<https://arxiv.org/abs/cs/9903011>)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Partition_problem&oldid=1038464132"

This page was last edited on 12 August 2021, at 18:17 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.