Stochastic Optimization of the Number Partitioning Problem

Lana Awad, Erik Bauch, Tamara Dordevic, Gregory Phelps, Trevor Rhone

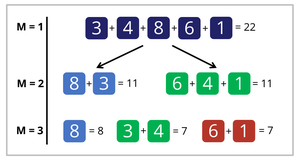
Introduction

Number Partitioning Problem (NPP): Suppose we have a set of numbers, $A = \{a_1, a_2, \cdots, a_N\}$. We want to place these N numbers into M partitions such that each partition is as close to equal (in total value) as possible. Formally, this amounts to minimization of the following energy functional:

$$E^2[\boldsymbol{P}] = \sum_{m=1}^{M} \left| \sum_{i \in \boldsymbol{P}_m} a_i - \frac{1}{M} \sum_{i=1}^{N} a_i \right|^2 ,$$

where $\boldsymbol{P}_m$ is the partition containing the corresponding indices in $\boldsymbol{A}$. Interestingly, this is one of the famous NP hard problems. The complexity is understood by recognizing that there are M^N combinations of indices in $\boldsymbol{P}$.

Applications: Dividing tasks between processors, scheduling attendants on flights, cryptography

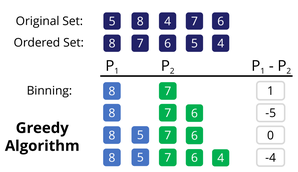


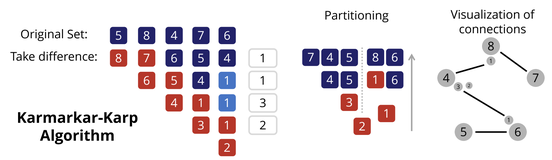
Examplea partitioning for M = 1, 2, 3 partitions

Deterministic Algorithms

Greedy Algorithm: This algorithm sorts each element into the partition with smaller total value (see right). While it is extremely efficient and easy to implement, the optimal value is not always found.

Complete Algorithms: A class of algorithms that create subsolutions, and combines them together in all possible ways. While they are usually a lot more complex in nature, they find optimal solutions if they exist.





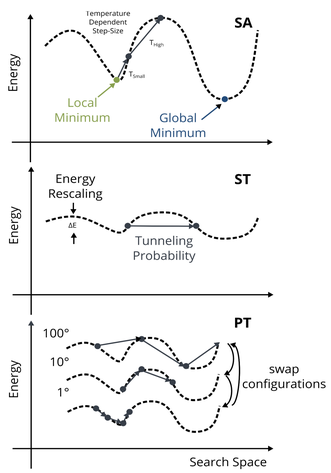
Karmarkar-Karp Algorithm: Is the most efficient and versatile of all algorithms. It provides solutions very close to the optimal.

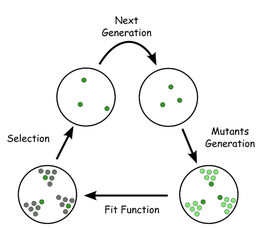
Stochastic Algorithms

Simulated Annealing (SA): As an extension to MCMC, the step size in SA is intimately linked to a temperature. A cooling schedule and multiple reannealing cycles allow to find an approximate global minimum.

Stochastic Tunneling (ST): Introduces an adaptive cooling schedule to SA controlled by a tunneling parameter γ. This is achieved by rescaling the energy landscape in each step.

Parallel Tempering (PT): In PT multiple copies of SA at different but fixed temperature are run in parallel. At regular intervals, the configurations of any two simulations are swapped propabilistically to make configurations at high T available to simulations at low T and vice versa. The overhead in runtime is usually compensated by a much faster convergence.

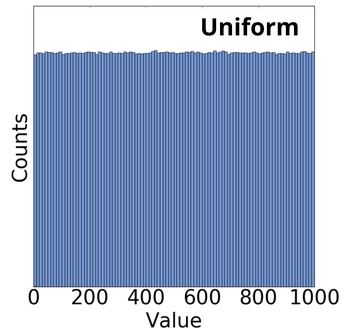


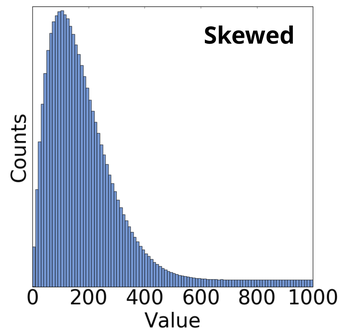


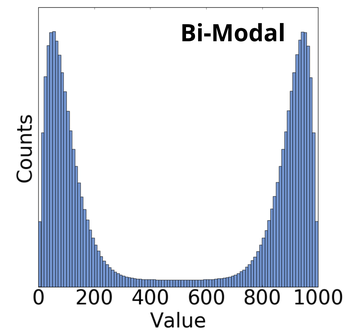
Genetic Algorithm: Uses concepts from Darwinian evolution to breed optimal (most fit) solutions. (1) Crossover: two solutions in one generation exchanging subsets in the next generation (2) Mutation: subsets randomly undergoing insertion and deletion of values in them and (3) Elitism: a fitness metric is used to give the top solutions preference to move directly to the next generation of solutions.

Comparison of Methods

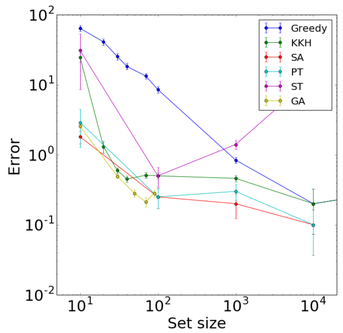
Synthetic Data: To see the advantage of one algorithm over another we explore the numbers from three distributions. We expect the skewed distribution to more likely have perfect solutions for lower set sizes when compared to the uniform distribution. The bi-modal distribution will have solutions with larger values for small set sizes.

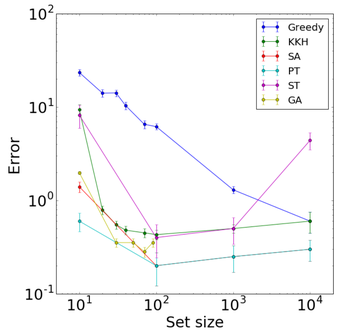


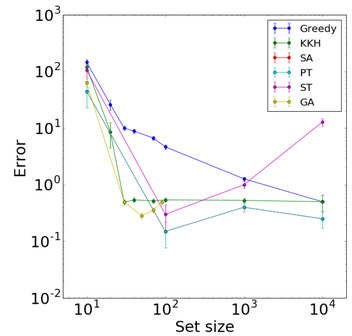




Results: The evaluation of the energy functional is plotted against set size. This is done for data taken from all three distributions. One thing to notice is that the Greedy algorithm is initially bad at finding a solution and gets better as the set size increases. KKH, on the other hand, is almost immediately as good as the stochastic methods.



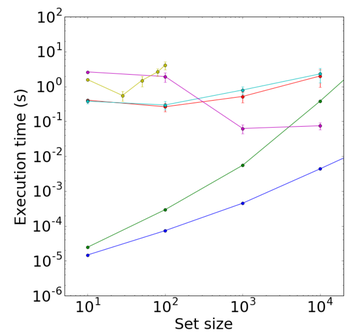




Discussion: We find slight differences in the behavior of the error for low set sizes between the various distributions. We see the error increase and decrease in the skewed and bi-modal distribution data sets, as we had intended. We find that simulated annealing and parallel tempering perform the best for nearly all data set sizes. Depending on the application, it may be beneficial to sacrifice finding an optimal solution for speed or vice versa. The Complete Karmarkar-Karp algorithm is capable of finding an exact solution, but it can (in worst case) take order 2^N operations to do so.

Future Directions: We believe that these stochastic algorithms will begin to distinguish themselves more when the number of partitions is increased - initial findings suggest this. With more time we would like to try increasing the number of partitions as well as trying more engineered data sets to further distinguish the various algorithms.

Timing: All code was executed on the same machine using an Intel I7-4790K processor. We find, as expected, that the Greedy and KKH perform the best in execution time. The scaling of the two deterministic algorithms are nowhere near the optimal – due to inefficient implementation. Ideally these two should always scale more favorably than the other algorithms.



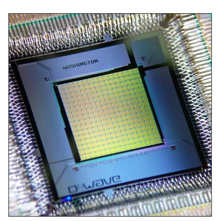
Beyond Classical Algorithms

Quantum Annealing: The analog of SA, but now allows quantum tunneling (a quantum effect where a particle can move through a potential wall that classically forbids the particle). Tunneling can occur even at low temperatures, which allows more of the state space to be searched. In the 2-partition problem, we represent the occupation of a number by a bit (1 or 0), which extends to the quantum case by introducing the qubit. Speed-up occurs due to the simultaneous evolution of the global quantum state for all qubits. This provides an advantage over classical computers since many clock cycles are necessary to search the parameter space.

We can map the 2-partition problem to a spin basis, $s_i \in \{-1,1\}$, by the following:

$$F[\boldsymbol{P}] = \sum_{i \in \boldsymbol{P}} a_i - \sum_{i \notin \boldsymbol{P}} a_i \rightarrow F[P] = \sum_i \boldsymbol{s_i} a_i \rightarrow H \propto \sum_{i,j} a_i a_j \boldsymbol{s_i s_j}$$

This is similar to the Ising Hamiltonian, but to write down the quantum mechanical Hamiltonian we need to replace the spins by operators.



D-Wave 2X Quantum Annealer