Sphere Packing

Tom Magerlein & Ben Nissan

Overview

This project consisted of implementation of and experimentation with a sphere-packing algorithm in Mathematica. The algorithm went as follows: a given number of point-like spheres were dispersed randomly on the surface of a unit sphere. The algorithm then attempted to move each one by a small amount along the surface of the sphere, and recorded the ratio of successful to unsuccessful shifts. If the ratio was zero, the spheres were considered fully packed, and the algorithm finished. Otherwise, all of the spheres were expanded by a given amount, and the whole process was repeated until the spheres were packed. We tested the effects of changes in the convergence criterion, number of attempted moves, inflation strategy, and number of particles on the algorithm's performance.

Variations on the Algorithm

We tried two significant variations on this general algorithm.

Multiple Move Attempts

We realized early on that only allowing a single move attempt per particle in each iteration would likely create a significant number of zero acceptance ratios when there were still many possible moves, which would in turn result in a loose final pack. With this in mind, we adapted the algorithm to make multiple move attempts per particle before calling it quits. This became especially relevant near the end of the algorithm; tightly packed particles would very often only be able to move in a few specific directions, and making those moves was necessary to get a near-optimal final pack.

Adaptive Inflation

The second variation of the algorithm that we tried was implementing adaptive inflation rate, where the particles would increase at some fixed rate up until the minimum distance between two particles became less than twice that inflation per iteration; for all iterations where two particles were closer than that, it would use half the distance between their surfaces as the inflation for the given step, and scale the average magnitude of the particles' motion accordingly. This added a new termination condition for the algorithm as well. If the inflation rate was too small (we used below an inflation of 10^{-15}), the algorithm would stop. This avoided a situation we encountered early on where the spheres would inflate basically by machine epsilon at every iteration, and the corresponding particle movements would be so small that the algorithm would run all the way out to our cap of 2000 iterations without really doing anything once it got into that state.

Results

For each variation on our algorithm, we performed 50 runs each with 12, 20, and 50 particles. We tested 1, 2, 5, and 10 move attempts per particle per iteration; the adaptive inflation method was tested with both 1 and 10 move attempts per particle per iteration.

Basic Algorithm

Our most basic packing algorithm worked essentially as we would have expected; it generally got reasonably close to ideal packs for numbers of particles where we knew what those would be, and its other packs looked decent as well. It sometimes ended up with rings or other structures that couldn't be repacked, but over a few runs it would generally hit a good pack. One problem that this method had, however, was that it would often terminate itself before reaching the best possible pack because the moves that happened to be randomly generated for that iteration happened to all be rejected. This is shown by Figure 1, as the general curvature of the acceptance ratio curve doesn't appear to be at zero when the algorithm terminates. This is shown particularly vividly on the moving-average graph, where the average acceptance ratio for the last few iterations is actually above 20%. This became somewhat less of a problem for larger numbers of particles, in large part because there were simply more particles moving, so every one of them happening to make a move that was rejected when acceptable moves were available was lower.

Multiple-Move-Attempt Algorithm

As we expected, this algorithm tended to produce somewhat denser packs at the cost of only slightly increased run time. However, there was in general no improvement in the pack as a result of making more move attempts beyond two or three, as shown by the fact that in Figure 2 the multi-attempt packs' results are all clustered together, with the single-attempt pack somewhat below and to the left of these clusters.

This method of making multiple move attempts did have a downside, however. For larger numbers of particles, inflation was rapid enough that a single particle could keep the algorithm from stopping by making tiny moves, with the rest of the particles failing to make any moves, and thus causing the other particles to inflate such that they overlapped one another. Needless to say this makes the pack invalid. Looking at the individual run data in Figure 3, we see that the packing ratios for some of the 50-point, 10-attempt packs even go above one, which is very obviously wrong.

Adaptive Inflation

As expected, the adaptive inflation algorithm took massively longer than anything else we tried for comparable numbers of particles — at least for multiple-attempt packs. As shown in Figure 2, for smaller numbers of particles, it sometimes converged as quickly (time-wise) as our other methods when run taking only single move attempts. It also tended to produce lower packing ratios than any of the other methods. This can probably be attributed to some combination of the fact that the adaptive inflation algorithm was the only one that actually enforced non-overlapping spheres specifically (since having the spheres overlap would inflate the packing ratio) and that it could terminate if two spheres happened to end up right next to one another by chance, even if they have plenty of room to move in other directions.

Conclusions

Our general assessment our our results had several major points. First off, the basic, single-move-attempt algorithm actually worked quite well, and didn't suffer nearly as much from the overlapping problem that the multi-attempt variant did. It thus appeared to be pretty much universally better than the adaptive inflation method, at least the way we implemented it. That said, the adaptive method was in fact the only one that completely ensured non-overlapping spheres, and finding some way to enforce that requirement on the other methods would certainly make them at least as slow if not slower.

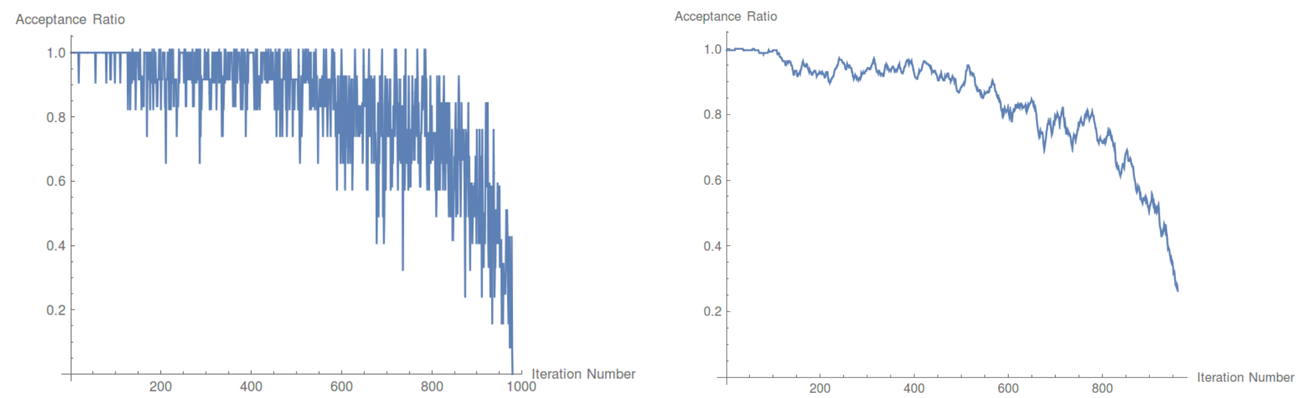

Figure 1: The acceptance ratios for each iteration of a 12-particle 1-attempt pack. The left plot is the original data, and the right one is a moving average over 20 iterations of that data. Note the gap between the end of the moving average plot and zero acceptance.

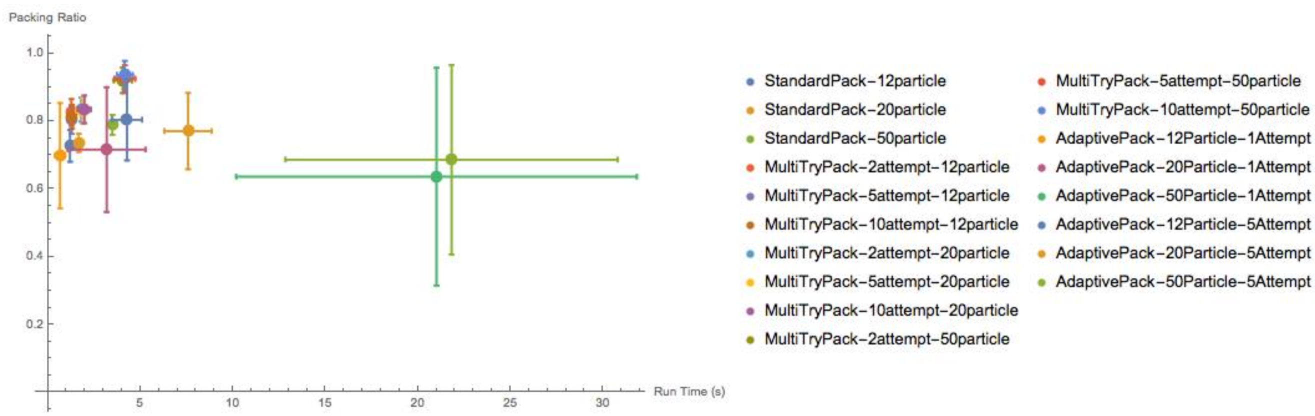

Figure 2: A plot of the mean packing ratios and run times for 50 runs each of the various packs indicated. Error bars show one standard deviation in the corresponding quantity for that pack.

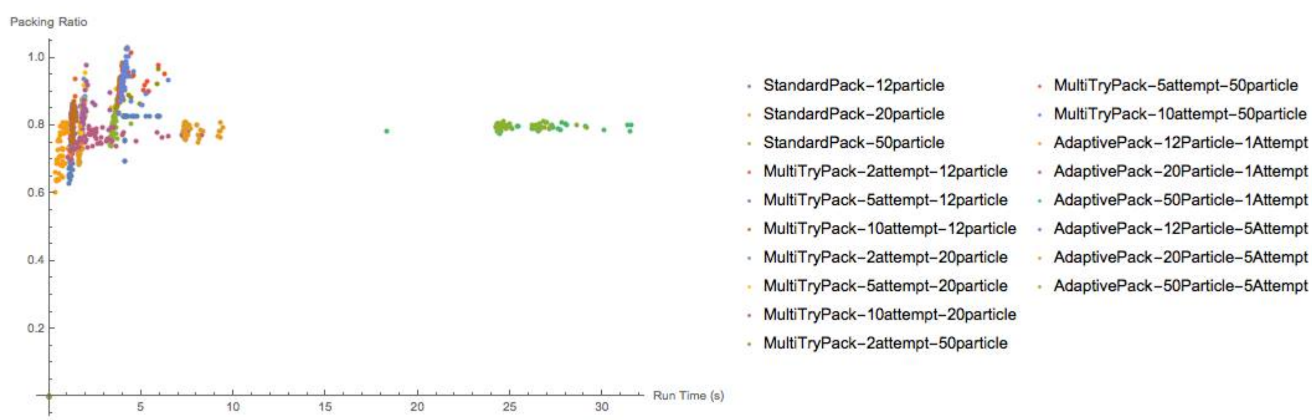

Figure 3: A plot of all the data collected from the 50 runs of each packing configuration.