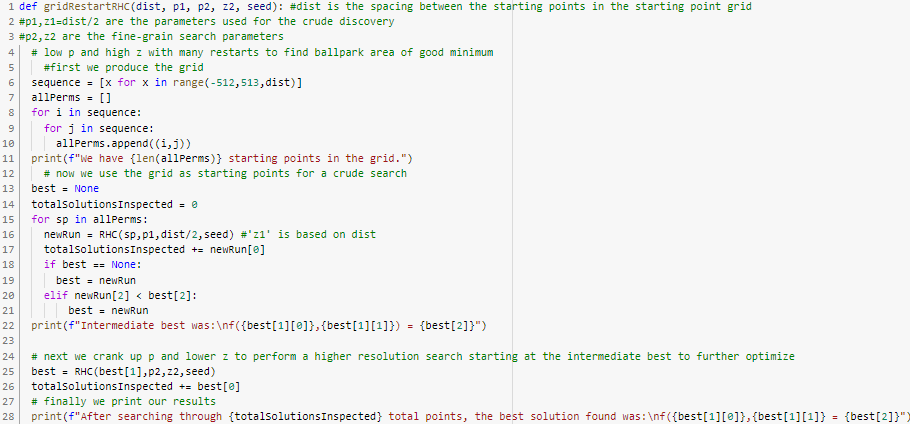
Problem Set 1 Task 1

Name: Arno Dunstatter

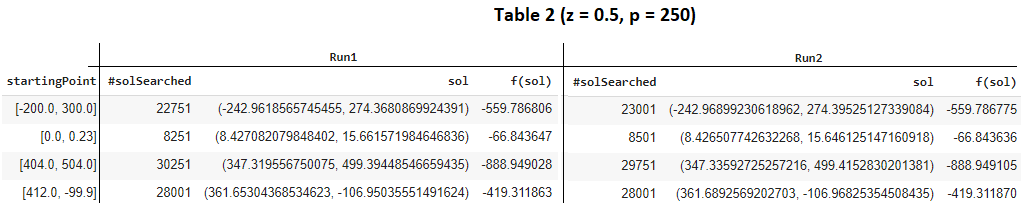
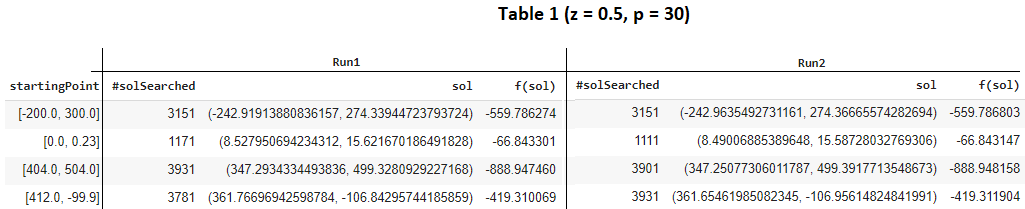
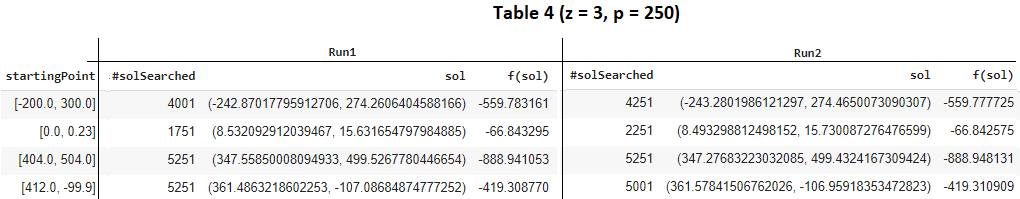
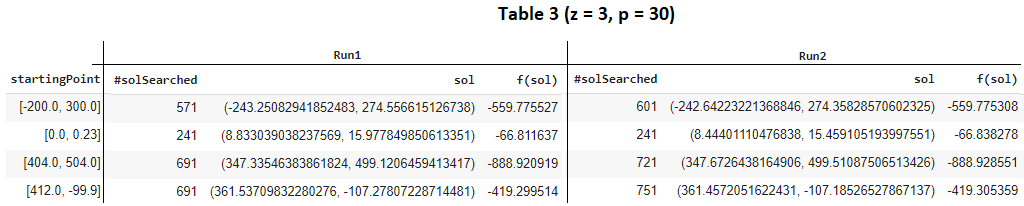
eid: 1926976

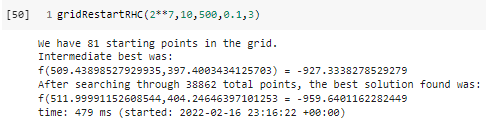
# Methods

RHC was implemented as described in the lecture notes and was called 32 times to generate the first four tables below. Additionally, for the “33rd Run”, in order to find as optimal a solution as possible, the RHC function was also utilized in a function called “gridRestartRHC” which essentially produced an evenly spaced grid of starting points across the space of interest, performed a crude search across all these starting points to produce a new, more optimal starting point, and then performed a more granular search from that more optimal starting point, ultimately printing out the results and some relevant metrics. This algorithm is shown in full below:



# Results

The best run of gridRestartRHC was as follows:  


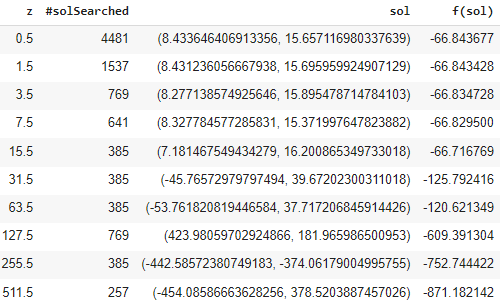
Note that this method explored ~8800 more solutions (~28% more) than the sp = (404,504) run in table 2, and this resulted in a ~7% lower minimum achieved, which is much better than any of the other runs using the same starting point with different p and z values.

# **Discussion**

As the tables above illustrates above here are the effects of changing p and z:

* Effects on time:
  + The parameter p has a strong positive correlation with the number of samples searched, which is to say it has a strong negative correlation with the speed of the algorithm. I.e. increasing p strongly increases the number of solutions searched and slows the algorithm’s convergence.
  + The parameter z has a middling negative correlation with the number of samples searched, which is to say it has a middling positive correlation with the speed of the algorithm. I.e. increasing z moderately decreases the number of solutions searched and quickens the algorithm’s convergence.
* Effects on optimization:
  + The parameter p has a weak positive correlation with optimization performance. I.e. increasing p marginally improves the optimization performance.
  + The parameter z has a weak negative correlation with optimization performance. I.e. increasing z marginally decreases the optimization performance.

Generally, reasonable p and z values mostly affect the rate at which a particular optimum is converged upon and the degree to which it that optimum is realized, however, with large z values the algorithm can jump from one particular optimum region to a different optimum region, as shown for different RHC calls with sp=(0,0), p = 128, seed =1:



It should be noted here that while the largest z value resulted in the best solution of all z values, this large z value also prevents that algorithm-run from fully exploring that optimal region, as it likely overshoots the region into a less optimal region with the next iteration of sampling. For this reason large z values are generally not advisable for optimization, other than perhaps for crude exploration of the space.

Unlike reasonable p and z values, which generally only affect the rate at which a particular optimum is converged upon and the degree to which that optimum is realized, changing the starting point of the algorithm can have very strong effects on both the speed of the algorithms convergence on a solution as well as the optimality of said solution, and this is because changing starting points essentially allows the algorithm to explore different regions of local optimum while maintaining reasonable values for p and z. For instance, if one selects a starting point that is already near a strong local optimum this will result in both faster convergence and a better overall optimization, provided p and z are held constant, relative to some other sub-optimal starting point.

# Conclusion

There are two competing motivations in choosing p and z - one is the desire to find the lowest point in the space space of f, the other is to obtain results quickly. Unfortunately, as with so many things, there is a tradeoff between these two motivations. Selecting a low p and high z will result in an algorithm that terminates relatively quickly, however it may not actually realize its local optimum. On the other hand, choosing a high p and a low z will, in general, realize its local optimum better but it will take longer to converge thereupon. It is therefore necessary to find a balance such that a good enough solution is found in a contextually reasonable amount of time. Ultimately, however, the best value of f found is mostly determined by the starting point, rather than by p or z, which instead primarily affect the speed at which the algorithm terminates and the degree to which a local optimum is realized. This is precisely why restarts are so widely used alongside hill climbing - to expose the algorithm to different regions of local-optimality to explore, rather than simply exploring the same hill or hole at different rates. Overall, random hill climbing is a straightforward algorithm to implement, and for reasonable values of p and z, combined with a sensible restart strategy, can find very good solutions in reasonable amounts of time.