	Run1 seed=100			Run2 seed=200		
p=30 & z=0.5	#sol searched	8	f(sol)	#sol searched	sol	f(sol)
(404,504)	3841	347.34246508759276, 499.47401227495544	-888.9483	3811	347.3758675495862, 499.4218121225173	-888.94864
(0,0.23)	1141	8.34021908598721, 15.650548172134584	-66.84267	1141	8.492524537123126, 15.611354040276241	-66.84346
(-200,300)	3211	-242.9827228936233, 274.4307591730589	-559.78623	3331	-242.94421922422924, 274.38205745927945	-559.78672
(412,-99.9)	3751	361.82999442308073, -106.89539661194966	-419.30955	3751	361.6844022065107, -106.94421885460126	-419.31189

	Run1 seed=100			Run2 seed=200		
p=30 & z=3	#sol searched	sol	f(sol)	#sol searched	sol	f(sol)
(404,504)	691	346.63664533518096, 499.3829245205349	-888.84111	691	346.66490464818423, 499.32248136795187	-888.86131
(0,0.23)	211	8.572663327170893, 15.349635023021492	-66.83119	211	8.384851044424005, 15.632136806910626	-66.8432
(-200,300)	631	-243.08532902563908, 274.1949078846229	-559.77513	631	-242.94211911667597, 274.3193113103196	-559.78598
(412,-99.9)	661	362.0253935785934, -107.17501807991692	-419.27701	691	361.24040565425594, -106.81495876499145	-419.2759

	Run1 seed=100			Run2 seed=200		
p=250 & z=0.5	#sol searched	sol	f(sol)	#sol searched	sol	f(sol)
(404,504)	29251	347.3383141677098, 499.456058778601	-888.94873	30001	347.32287949572134, 499.4160336807128	-888.94912
(0,0.23)	8501	8.445228381508091, 15.66494716371552	-66.84369	8501	8.438146976503715, 15.637075867371784	-66.84365
(-200,300)	24251	-242.98496379140906, 274.3680759425498	-559.78681	23751	-242.9827217643548, 274.3874339118634	-559.78684
(412,-99.9)	27501	361.6339926687065, -106.99468404728906	-419.31182	27501	361.682617558574, -106.95004361299785	-419.31192

	Run1 seed=100			Run2 seed=200		
p=250 & z=3	#sol searched	sol	f(sol)	#sol searched	sol	f(sol)
(404,504)	5501	347.26532282957413, 499.33320142673807	-888.94782	5501	347.43348676304015, 499.4950475251346	-888.94729
(0,0.23)	1751	8.532183747103844, 15.594140178623276	-66.84301	2001	8.36928684428422, 15.67360829297440	-66.84315
(-200,300)	4251	-243.01561415485983, 274.5445469004778	-559.78043	4751	-243.0288964085006, 274.407666599518	-559.78653
(412,-99.9)	5001	361.6731014029752, -107.06285564662903	-419.31023	5251	361.5955006262284, -106.99718413107387	-419.3115

		33rd run seed =100	
p=300 & z=350	#sol searched	sol	f(sol)
(404,504)	901	512, 404.4623571528931	-959.58014

Based on the results of the 33rd run, we can conclude that the best starting position is near (512,404) which is the lowest point on the objective function $f(x,y) = -(y+47)\cdot\sin\sqrt{\left|\frac{x}{2}+(y+47)\right|} - x\cdot\sin\sqrt{\left|x-(y+47)\right|}$, which is one of the disadvantages of RHC if the randomized starting location isn't optimal or the neighborhood size isn't large enough to allow for avoiding of plateaus and local optimums. Overall, algorithm speed is very fast depending on when global minimum was reached, it could also fail to reach the global minimum and get stuck in a local minimum. Solution quality seems to be better as both (x,y) gets closer to (512,404). The impact of the starting point has the most effect on quality of solution, and that makes sense, if the algorithm luckily chooses the global optimum upon random selection, then any *p* or *z* that is large enough to where it won't slow down the algorithm speed is good enough. Otherwise, the size of the neighborhood, *z*, has the next most effect on quality of solution, as long as *z* is large enough, you can avoid local minimums and plateaus without effecting too much on speed. Lastly, *p,* the number of neighbors has the largest effect on the algorithm speed because of the amount of solutions being searched, if *z* is too small, *p*, would need to run many more times just to find a local optimum. RHC is very good at finding a local minimum for *f(x,y)*, depending on starting point, neighborhood size, and number of neighbors sampled.