Michael Kammeyer
mekmb2@mst.edu
COMP SCI 5401 FS2015 Assignment 1d

For this assignment, the author created a multi-objective evolutionary algorithm which is meant to maximize the SAT value when run on a cnf while using a minimum number of variables to do so. The two objectives are represented by the number of correct clauses and the number of "don't care" values in the solution. Fitness for the algorithm was determined by these two values and the solution's level of non-domination, specifically using this equation:

$$Fitness = (Pareto\ Front/Total\ Pereto\ Fronts) * Objective\ 1 * Objective\ 2$$

The algorithm was run five times on the same cnf file, with the following configurations:

1. Parent Selection: K-Tournament
   Survivor Selection: K-Tournament

2. Parent Selection: K-Tournament
   Survivor Selection: Random

3. Parent Selection: K-Tournament
   Survivor Selection: Truncation

4. Parent Selection: Random
   Survivor Selection: Random

5. Parent Selection: Fitness Proportional
   Survivor Selection: Fitness Proportional

All of the experiments used the (μ + λ) Survival Strategy where μ = 500 and λ = 250, a k size of 300 (for the configurations using k-tournament selection), and a flat mutation rate of 1%.

K-Tournament Selection

Using k-tournament parent selection forced the results to converge extremely quickly. This is because the algorithm used far too high of a tournament size. Given a population of 500, 300 were selected for the tournament. This gives a ⅗ chance that best solution was chosen as the parent. It is relatively likely that the same parent was chosen twice twice, which would yield an exact copy of the parents as a child. Because half of the population was replaced with a relatively high probability of copying the top parent, there were many duplicates in the population after a couple generations, and the probability of copying the top parent increased with each generation. The result is a final population that is literally all the same, therefore fitting

into one pareto front. For this reason, the graphs of the pareto fronts were not included for these three experiments.

With regards to the individual objectives, k-tournament selection heavily favored Solution Robustness, or objective two. This is a result of the bias that the fitness function tends to show towards robustness, even though neither objective is weighted in the function. Because of the nature of the MaxSat problem, the total number of clauses will almost always be greater than the total number of boolean variables. This causes a change in the number of “don’t cares” to have a much greater impact on the overall fitness than a change in the number of correct clauses. The most fit individuals in the initial population are the ones that have the highest number of unused variables, so a greedy algorithm using highly selective k-tournament parent selection chooses parents with a high robustness value, therefore favoring that value over the number of correct clauses. The T-Tests in Figures 11 - 16 show that k-tournament parent selection with Truncation survival selection performed the best overall with regards to objective two.

The results of the k-tournament configurations heavily depended on a strong initial population in order to perform well, so while they did provide more consistent results for objective two, they fell far behind in objective one. To improve performance, the diversity of the population needs to drastically increase by lowering the tournament size and/or increasing the mutation rate. That way, the population would not converge so quickly, and obtain a more consistent result.

Random Selection

Using random parent selection and random survival selection yielded a much more stable result than that of the k-tournament configurations. Figures one through four both show that the averages for both objectives did not change much throughout the runs. Being random, the algorithm does not show any sort of bias towards one objective or the other. However, it also does not improve over the course of the runs since it does not try to choose fitter parents or kill off less fit children. It is not viable as an optimization strategy, and really only serves as a control in this experiment. Better results are shown in the configuration using k-tournament parent selection and random survivor selection. Although the k-tournament still shows heavy bias towards objective two and converges quickly, the random survivor selection helps to slow convergence significantly as shown by Figure two. Perhaps it could be even more effective if the roles were switched, using random parent selection and k-tournament survivor selection. This would keep the new children diverse and kill off stragglers, resulting in an overall increase in fitness with more diversity than k-tournament parent selection.

Fitness Proportional Selection

Fitness proportional parent and survivor selection provided strong solutions without converging nearly as fast as k-tournament selection. The T-Tests in figures 7 - 10 show that it performs far better for objective one than k-tournament selection. Although it has a higher mean value, it does not conclusively out-perform random selection because neither of them have

converged at the end of 10,000 evaluations. Despite the fact that neither of them converged, it is relatively apparent that fitness proportional selection has far better averages for objective one over the course of all the evaluations. Figures 5 and 6 show the best pereto fronts over the 30 runs for random selection and fitness proportional selection. Fitness Proportional selection gives far more solutions that dominate the solutions created by random selection, designating it as the better pereto front overall. It should also be noted that fitness proportional selection yielded a front with 39 solutions in it as opposed to random selection's 21. This is because random selection provides a more diverse population overall. The random selection configurations failed to show improvement on average, but it cannot be completely discounted because given a strong starting population and lucky selections it has the potential to perform extremely well.

The fitness proportional selection keeps the population more diverse than k-tournament selection because the selections are still made based on a random chance. The weighted values of higher fitnesses allow it to consistently improve while staying diverse. As noted in the k-tournament section, k-tournament selection could stay more diverse by choosing a smaller tournament size, therefore lowering the chances of using the same parent for recombination.

Conclusion

As a whole, the algorithm behaved about as expected given the configurations for the experiments. Tweaking the different experiments could yield better results, but they still show how the multi-objective algorithm behaves. The biggest flaw with the algorithm in its current form is that it does not account for the natural bias towards objective two - solution robustness. This could be remedied by changing the fitness function to include a weight for each objective depending on how large the maximum value for that objective is. This would help the algorithm to perform similarly for all input files.

Appendix

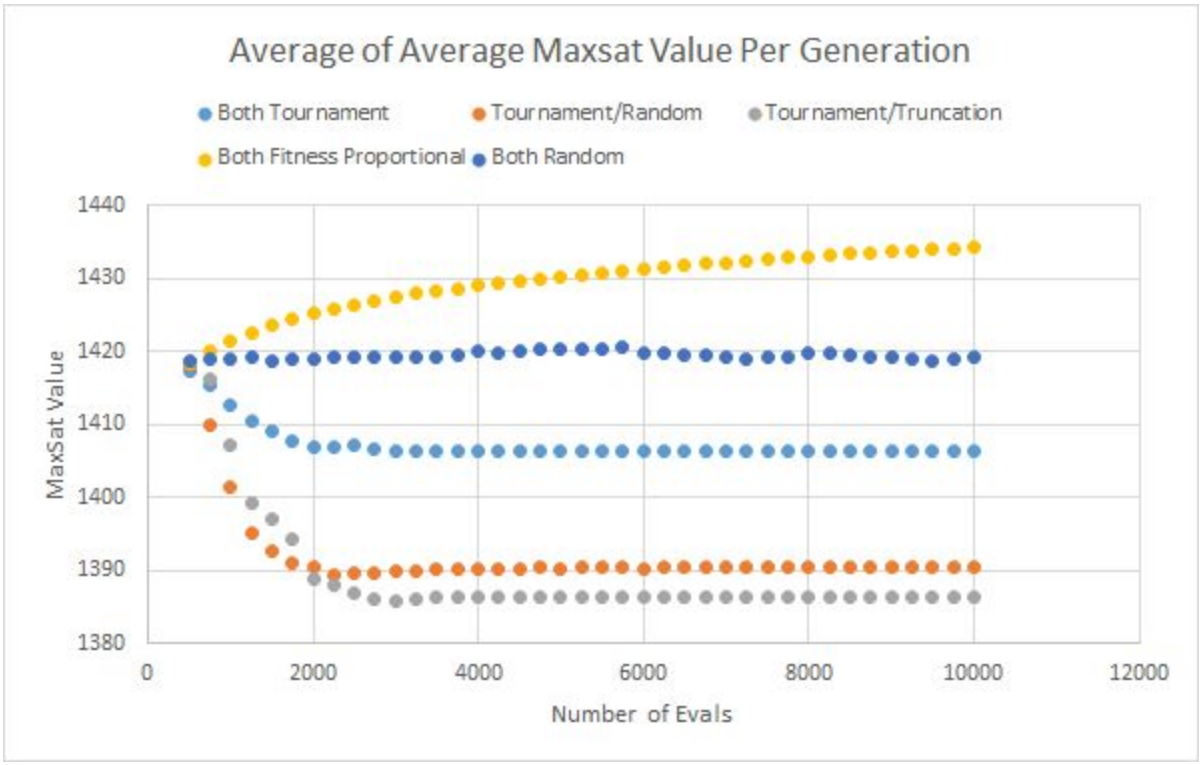

Figure 1

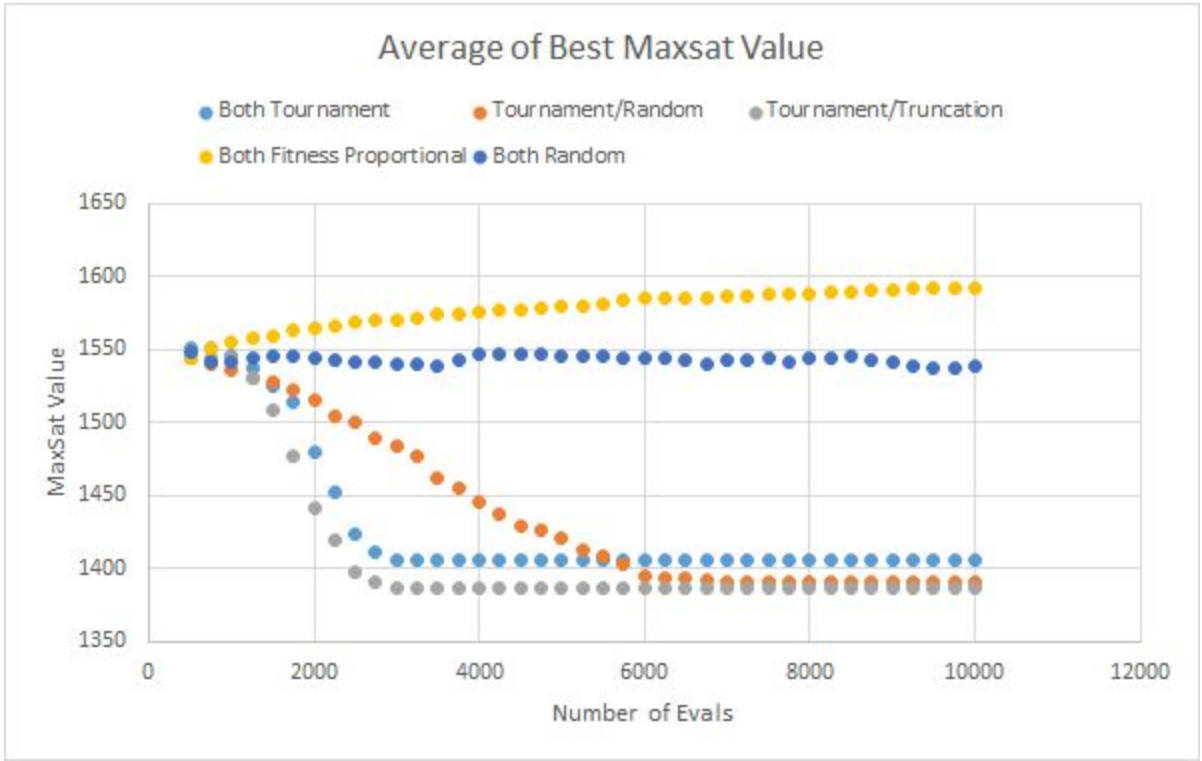

Figure 2

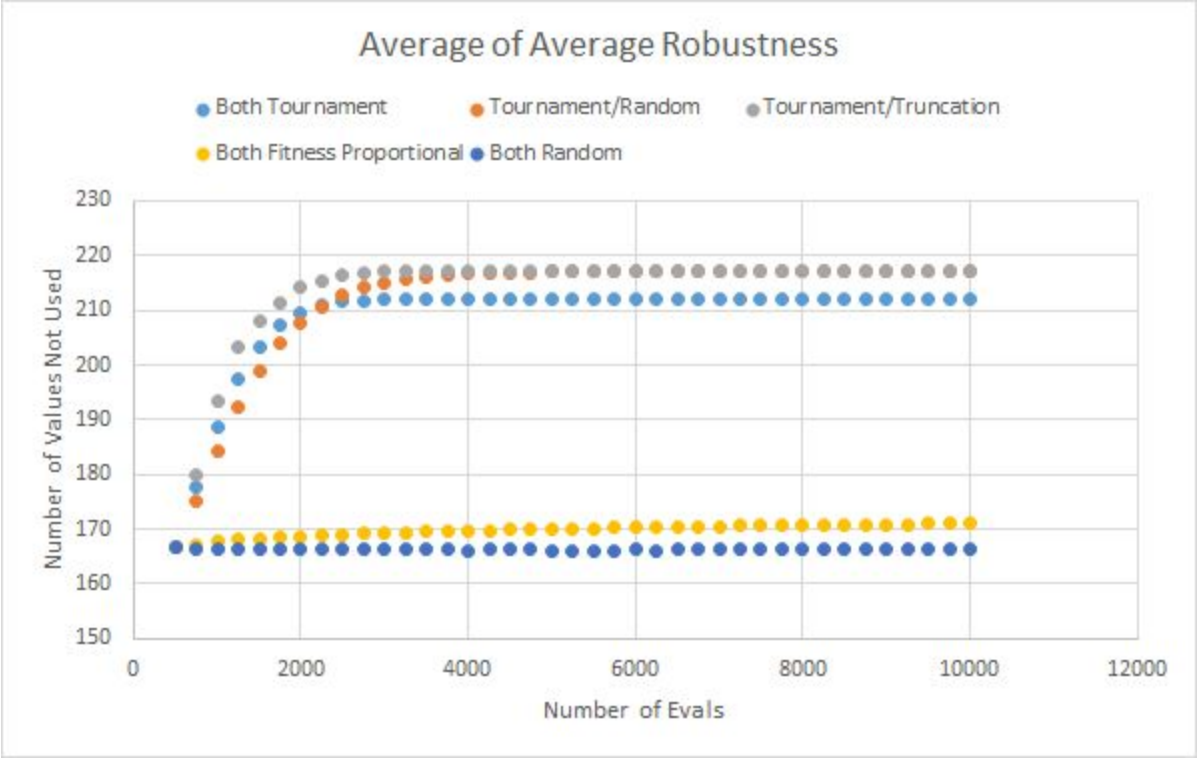

Figure 3

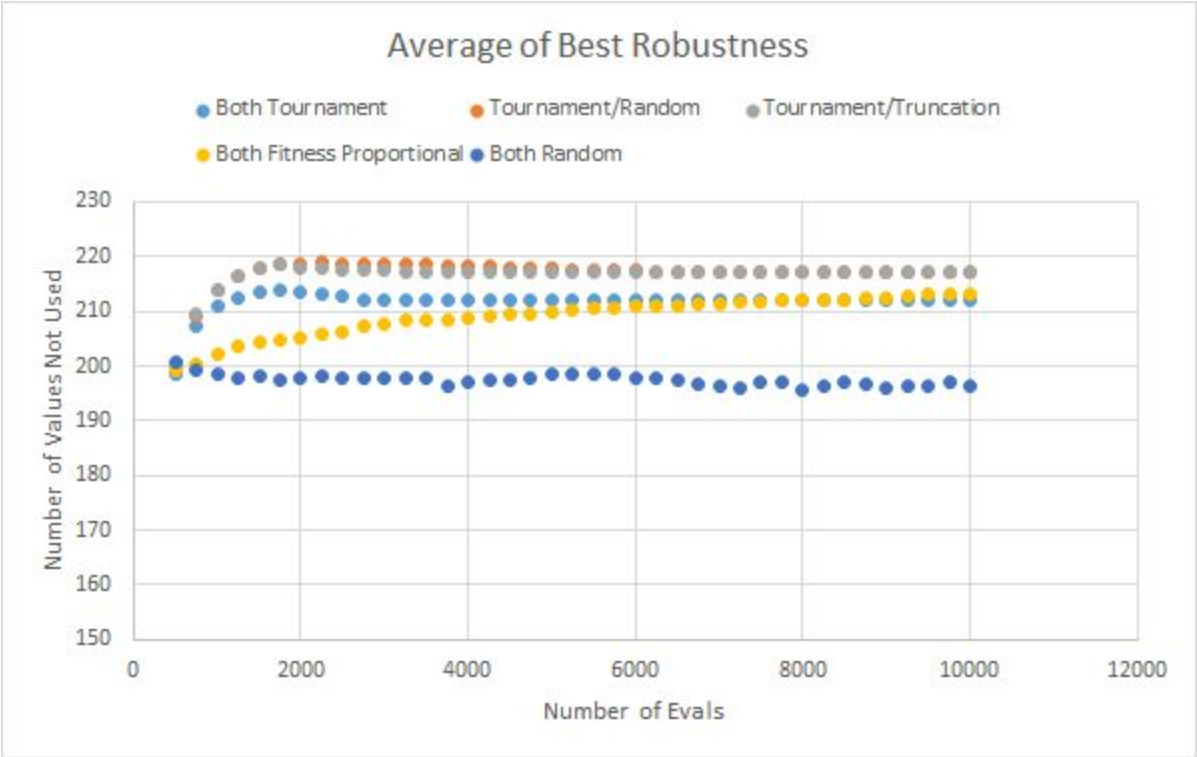

Figure 4

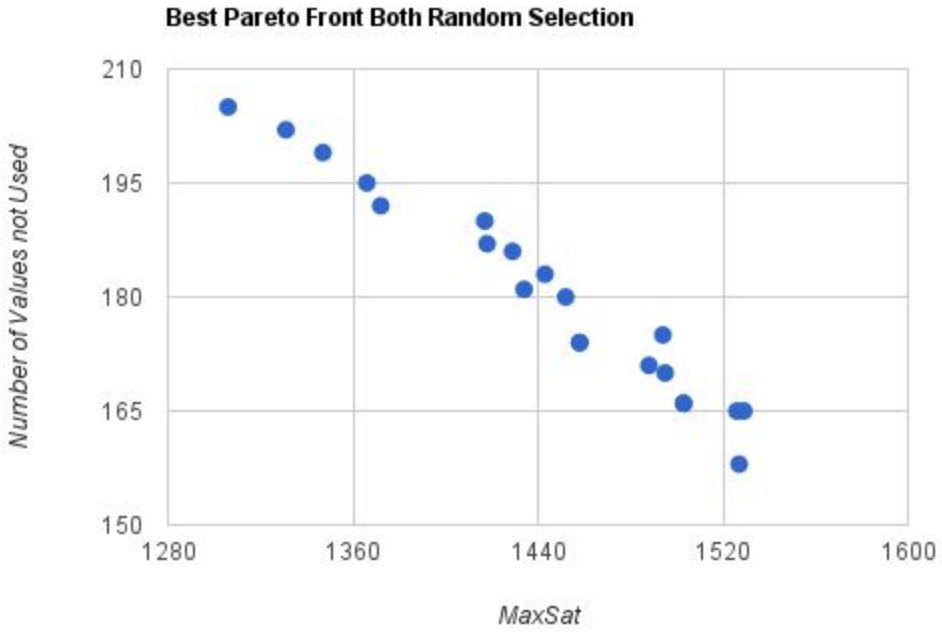

Figure 5

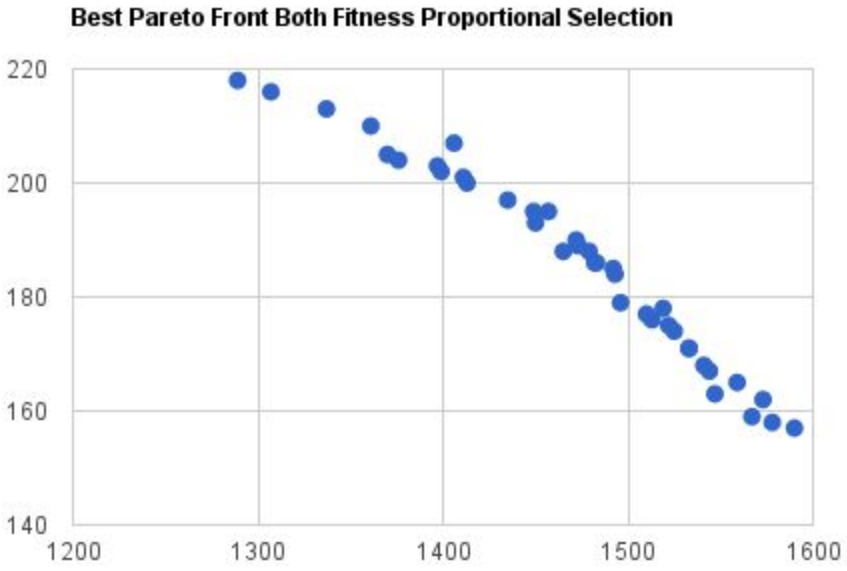

Figure 6

T-Tests for Objective 1

t-Test: Two-Sample Assuming Equal Variances		
	Both Tourney	*Both Fitness*
Mean	1407.406872	1429.486373
Variance	5.995423777	17.43795615
Observations	39	39
Pooled Variance	11.71668996	
Hypothesized Mean Difference	0	
df	76	
t Stat	-28.48420426	
P(T<=t) one-tail	1.32223E-42	
t Critical one-tail	1.665151353	
P(T<=t) two-tail	2.64447E-42	
t Critical two-tail	1.99167261	

Figure 7

t-Test: Two-Sample Assuming Unequal Variances		
	Both Random	*Both Fitness*
Mean	1419.49655	1429.486373
Variance	0.250739085	17.43795615
Observations	39	39
Hypothesized Mean Difference	0	
df	39	
t Stat	-14.83344998	
P(T<=t) one-tail	6.33231E-18	
t Critical one-tail	1.684875122	
P(T<=t) two-tail	1.26646E-17	
t Critical two-tail	2.02269092	

t-Test: Two-Sample Assuming Unequal Variances		
	Variable 1	*Variable 2*
Mean	1407.406872	1429.486373
Variance	5.995423777	17.43795615
Observations	39	39
Hypothesized Mean Difference	0	
df	61	
t Stat	-28.48420426	
P(T<=t) one-tail	3.04217E-37	
t Critical one-tail	1.670219484	
P(T<=t) two-tail	6.08433E-37	
t Critical two-tail	1.999623585	

Figure 8

t-Test: Two-Sample Assuming Equal Variances		
	Both Tourney	*Tourney/Random*
Mean	1407.406872	1391.983138
Variance	5.995423777	31.79406966
Observations	39	39
Pooled Variance	18.89474672	
Hypothesized Mean Difference	0	
df	76	
t Stat	15.66881945	
P(T<=t) one-tail	8.1914E-26	
t Critical one-tail	1.665151353	
P(T<=t) two-tail	1.63828E-25	
t Critical two-tail	1.99167261	

Figure 9

t-Test: Two-Sample Assuming Equal Variances		
	Both Tourney	*Tourney Truncation*
Mean	1407.406872	1389.372937
Variance	5.995423777	62.3977564
Observations	39	39
Pooled Variance	34.19659009	
Hypothesized Mean Difference	0	
df	76	
t Stat	13.61809644	
P(T<=t) one-tail	2.19477E-22	
t Critical one-tail	1.665151353	
P(T<=t) two-tail	4.38953E-22	
t Critical two-tail	1.99167261	

Figure 10

T-Tests for Objective 2

t-Test: Two-Sample Assuming Equal Variances		
	Both Tourney	*Tourney Random*
Mean	208.492947	211.7109795
Variance	94.7899841	145.7978071
Observations	39	39
Pooled Variance	120.2938956	
Hypothesized Mean Difference	0	
df	76	
t Stat	-1.295644733	
P(T<=t) one-tail	0.099509655	
t Critical one-tail	1.665151353	
P(T<=t) two-tail	0.199019311	
t Critical two-tail	1.99167261	

Figure 11

t-Test: Two-Sample Assuming Equal Variances		
	Variable 1	*Variable 2*
Mean	211.7109795	213.5141333
Variance	145.7978071	113.6230578
Observations	39	39
Pooled Variance	129.7104324	
Hypothesized Mean Difference	0	
df	76	
t Stat	-0.69913756	
P(T<=t) one-tail	0.243300128	
t Critical one-tail	1.665151353	
P(T<=t) two-tail	0.486600255	
t Critical two-tail	1.99167261	

Figure 12

t-Test: Two-Sample Assuming Unequal Variances		
	Tourney Truncation	*Both Fitness*
Mean	213.5141333	169.8025431
Variance	113.6230578	1.232328845
Observations	39	39
Hypothesized Mean Difference	0	
df	39	
t Stat	25.47142404	
P(T<=t) one-tail	3.23567E-26	
t Critical one-tail	1.684875122	
P(T<=t) two-tail	6.47133E-26	
t Critical two-tail	2.02269092	

Figure 13

t-Test: Two-Sample Assuming Unequal Variances		
	Tourney Truncation	*Both Random*
Mean	213.5141333	166.3201846
Variance	113.6230578	0.015007276
Observations	39	39
Hypothesized Mean Difference	0	
df	38	
t Stat	27.64755334	
P(T<=t) one-tail	4.47457E-27	
t Critical one-tail	1.68595446	
P(T<=t) two-tail	8.94913E-27	
t Critical two-tail	2.024394164	

Figure 14

T-Tests for Pareto Fronts

t-Test: Two-Sample Assuming Equal Variances		
	Both Random	*Both Fitness*
Mean	1442.666667	1470.410256
Variance	4473.833333	5874.932524
Observations	21	39
Pooled Variance	5391.794872	
Hypothesized Mean Difference	0	
df	58	
t Stat	-1.395925686	
P(T<=t) one-tail	0.084028792	
t Critical one-tail	1.671552762	
P(T<=t) two-tail	0.168057583	
t Critical two-tail	2.001717484	

Figure 15

t-Test: Two-Sample Assuming Equal Variances		
	Both Random	*Both Fitness*
Mean	180.1904762	186.025641
Variance	181.8619048	296.9730094
Observations	21	39
Pooled Variance	257.2795251	
Hypothesized Mean Difference	0	
df	58	
t Stat	-1.344054384	
P(T<=t) one-tail	0.092082413	
t Critical one-tail	1.671552762	
P(T<=t) two-tail	0.184164825	
t Critical two-tail	2.001717484	

Figure 16