

A Simple Genetic Algorithm for the Geometric Connected Dominating Set Problem

Evan Forbes
Evolutionary Computation CS4063
Dr. Wainwright
Spring 2014
evan-forbes@utulsa.edu

ABSTRACT

For this project, I created a custom genetic algorithm engine from scratch in Java and used it to collect data and compare results from various combinations of standard genetic algorithm operators for selection, crossover, and mutation including the N4N hypermutation proposed by Alkhalifah and Wainwright^[2]. These results were also compared with results from the simulated annealing and foolish hill climbing algorithms, with various perturbation functions.

INTRODUCTION

The geometric connected dominating set problem (GCDS) is formally defined as follows: Given a set of points in a plane, P , and a constant B , find the minimum $|P'|$ such that all points $P - P'$ are within euclidean distance of B of some point in P' and such that the graph $G = (P', E)$ with an edge between two points in P' if and only if they are within distance B of each other is connected^[1].

This problem is also informally referred to as the “radar” problem and can be stated as follows: given a set of cities, we need to select the minimum number of cities in which to install a range limited radar transmitter such that all cities receive radar

coverage and all transmitters are within range of another transmitter.

An example dataset and solution is shown in Figure 1. Each point represents a city. Square points are considered “selected” and will contain a transmitter whose range is represented by the surrounding circle. In order to be a feasible solution, all points must be within a circle and all selected points must have a path to all other selected points.

CHROMOSOME & FITNESS

The chromosome is a simple bit string of length $|P|$. For each point i , $c_i = 1$ if it is selected; Otherwise, $c_i = 0$.

The fitness function is composed of three factors:

- 1) Percent covered points (P_c)
- 2) Percent selected points (P_s)
- 3) Connectedness of selected points (C)

Factors 1 and 3 are to be maximized and factor 2 is to be minimized. The SGA was set up to minimize fitness, so the inverses of factors 1 and 3 were taken. The complete fitness function has the form:

$$Fitness = \alpha * \frac{1}{P_c} + \beta * P_s + \gamma * \frac{1}{C}$$

where alpha, beta, and gamma are variable weights for each respective factor. To select appropriate weights, all other variables were held constant and 10 trials were run for each value of alpha, beta, and gamma shown in Table 1 using datasets E100, E250, and F100. The average number of selected points in each case is given, except when a solution was not always found. Ignoring schemes where a solution was not always found, the best performing weighting schemes based on the average of the three datasets

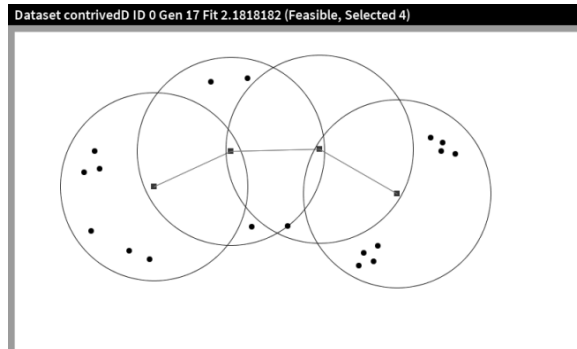


Figure 1- A valid solution to the GCDS

average number of selected points were number 1 and 6. Since both of these schemes had the same average performance in these trials, I went with the simpler choice, number 1, where $\alpha = \beta = \gamma = 1$. It should also be noted that while scheme number 1 was not always the best choice for all datasets, it was always able to find a solution and performed well on average.

id	α	β	γ	Avg. number of selected points		
				E250	E100	F100
1	1	1	1	22.8	12	22.1
2	1	10	10	--	--	--
3	1	100	10	--	--	--
4	3	2	1	22.5	11.6	23
5	5	10	1	--	--	--
6	10	1	10	22.7	11.7	22.5
7	10	5	1	22	12.2	--
8	10	10	1	21.1	11.6	--
9	10	100	1	--	--	--
10	100	1	10	22.8	12.1	23.2
11	100	10	1	21.7	--	--

Table 1- a comparison of various weighting schemes (a "--" indicates a solution was not always found)

OPERATORS

Three standard selection methods were implemented in this SGA: roulette, rank, and tournament. The tournament selection used a tournament parameter of 0.75.

Four crossover methods were implemented: single point, double point, n-point, and uniform. For the n-point crossover, a random number of crossover points were selected between 1 and the number of points divided by 2 and the location of each crossover point was randomly selected.

Three mutation methods were implemented: single invert, double invert, and N4N hypermutation[2]. When the dataset is loaded, the nearest four points are calculated for each point. When a chromosome is selected for N4N hypermutation, the fitness for every combination of selecting the nearest four points is calculated, and the combination with the best fitness is imposed upon the original chromosome.

PARAMETERS

The basic parameters for the generational SGA are population size, crossover rate, and mutation rate. Their values were 100, 1.0, and 0.01 respectively. These parameter values remained constant throughout all runs of the SGA. Elitism was used. Ten trials were run of each combination of selection, crossover, and mutation operators. The initial population was generated randomly.

The algorithm was terminated when any one of the following conditions occurred:

- 1) Out of time- The SGA has run of more than 10 minutes of wall clock time
- 2) Maximum number of generations was reached- The SGA completed over 10,000 generations
- 3) The population has stagnated- no improvement to the best fitness value of the population has occurred in over 2000 generations

DATASETS

The contrived datasets were constructed so that the correct answer is very easily seen by a human, but the search space is still large enough to provide a challenge to the SGA. Three contrived datasets were used to verify the algorithm: "contrived1" with 41 points, "contrived90" with 91 points and "contrived150" with 150 points. These datasets and their respective solutions are shown in Figure 2.

Two types of random datasets were also used, E and F. E type datasets had a larger maximum distance (radar range) than F type datasets, making them generally solvable with fewer selections. Random datasets with 100, 250, and 500 points of each type were used. These datasets and their possible solutions are shown in Figures 3 and 4.

SIMULATED ANNEALING & FOOLISH HILL CLIMBING

The parameters for the SA were as follows: $\alpha = 0.95, \beta = 1.05, T_0 = 10, i = 1000$. The

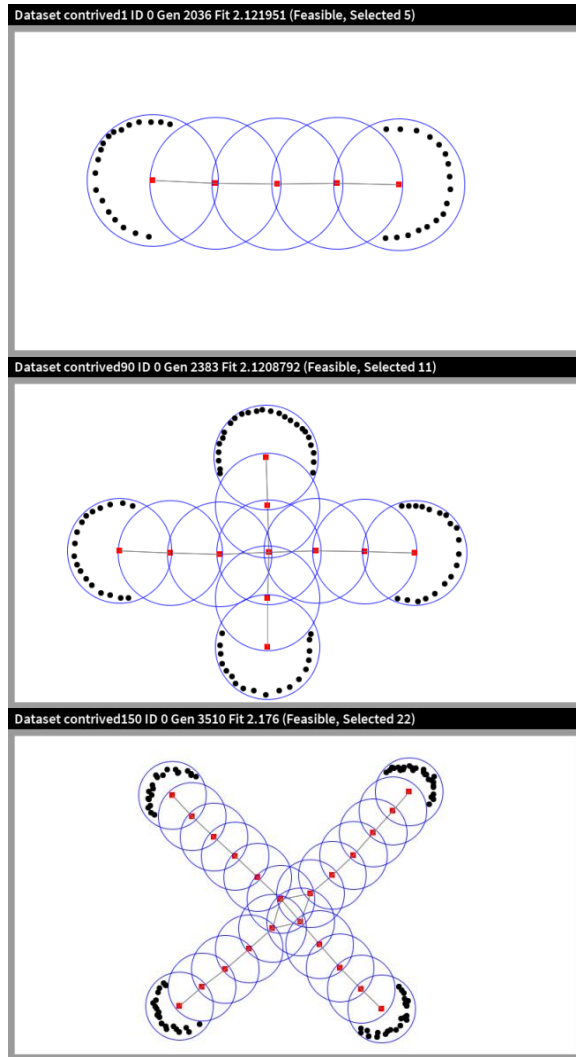


Figure 2: contrived datasets and their solutions

perturbation functions used were the same as the three mutation operators used for the SGA: single invert, double invert, and N4N hypermutation. Elitism was used. The SA and FHC were terminated after 5 minutes of wall clock time.

COMPUTATIONAL RESULTS

The data table is shown in Table 2. For each dataset, the minimum (best) selected number of points, the average of the number of selected points, the standard deviation of the number of selected points, and the average number of generations is shown from all runs. Any run that did not result in a feasible result was discarded. This only occurred in the simulated annealing runs with F-type datasets. For each average selected column, the lowest 10%

values are highlighted in green and the highest 10% values are highlighted in red.

CONCLUSIONS

All combinations of operators for the SGA were able to find feasible and correct (if contrived) solutions in all runs on both the contrived and random datasets. The best performing combination of parameters was (rank, N-point, double invert). This combination was in the top 10% in 5 out of 6 datasets and had the best average number of selected points in the most difficult dataset F500. Another notable combination is (rank, N-point, N4N). This combination performed very well when the number of points was highest (in the E500 and F500 datasets).

The simulated annealing algorithm performed well with the smaller datasets (E100 and F100), but did very poorly with increasing dataset size. The foolish hill climbing algorithm did surprisingly well with the E and smaller F datasets. I'm not sure why.

One possible thing to try in the future is to allow the alpha, beta, and gamma factors in the fitness function to vary dynamically. This could produce better results, since different combinations of values performed better for different datasets (as seen in Table 1).

The most difficult part of this project was data collection, so I made a GUI to simplify things (Figure 5). Also to save time, I ran three different runs at a time on different threads.

SOURCES

- [1] Garey, Michael R., and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: W.H. Freeman, 1979.
- [2] Alkhalifah, Yaser and Roger. L. Wainwright. "A Genetic Algorithm Applied to Graph Problems Involving Subsets of Vertices", Proceedings of the 2004 IEEE Congress on Evolutionary Computation (CEC 2004), 2004.

Selection	Crossover	Mutation	E100			E250			E500			F100			F250			F500									
			Minimum selected	Average selected	Stdev selected	Minimum generations	Average selected	Stdev selected	Minimum generations	Average selected	Stdev selected	Minimum generations	Average selected	Stdev selected	Minimum generations	Average selected	Stdev selected	Minimum generations	Average selected	Stdev selected							
Roulette	Single point	Single invert	11	12.1	0.99	2459	22	22.7	0.67	2909	31	33.0	1.49	4292	20	22.7	1.42	3178	42	47.5	3.06	4481	61	63.6	1.58	4523	
		Double invert	12	12.8	1.03	2467	20	22.3	1.06	3092	29	31.8	2.30	4091	20	22.4	1.26	3102	43	46.1	1.85	4175	60	62.5	1.96	4779	
		N4N	11	12.4	1.07	2236	20	21.9	1.20	2638	29	32.9	1.91	3150	22	23.3	0.82	2308	44	46.5	1.51	3079	58	62.5	2.17	3981	
		Single invert	10	11.9	0.99	3028	21	22.5	1.43	2955	31	32.8	1.69	4221	21	23.1	0.99	3272	45	48.0	1.83	3437	59	62.3	1.83	4787	
	Double point	Double invert	11	12.1	1.10	3540	20	21.9	1.10	3126	28	33.8	2.35	4209	21	22.7	1.42	2830	44	47.0	1.56	3639	59	62.4	2.63	4562	
		N4N	12	12.7	0.67	2248	20	22.9	1.52	2645	30	33.1	1.97	3356	23	23.9	1.20	2327	42	45.9	1.79	3091	61	63.0	1.76	3787	
		Single invert	11	12.7	1.16	2431	20	22.1	1.45	2968	32	33.5	1.18	3923	20	22.3	1.57	3606	43	45.7	2.06	3786	60	63.2	1.55	4453	
		Double invert	11	12.1	0.88	2704	19	21.7	1.57	2874	31	33.8	1.81	3917	21	22.7	1.06	2981	44	46.6	2.07	4055	58	62.6	1.96	4467	
	N-point	N4N	12	13.2	0.92	2218	21	22.8	1.32	2497	30	32.4	1.71	3338	22	23.3	0.82	2344	43	46.6	2.63	2994	60	62.1	1.20	4003	
		Single invert	11	12.1	0.88	3212	21	22.2	0.92	3424	30	33.2	1.93	3959	21	22.8	1.40	2849	40	46.1	2.92	3696	60	63.7	2.98	4247	
		Double invert	11	12.0	0.94	2578	19	21.9	1.79	3007	29	33.2	2.04	4243	21	22.3	0.82	3641	45	47.0	1.25	3815	59	64.3	2.83	4388	
		N4N	11	12.6	1.17	2237	21	22.5	1.23	2585	31	32.9	1.37	3065	22	22.9	1.29	2341	44	46.7	1.49	3144	57	62.4	2.67	4073	
Rank	Single point	Single invert	12	12.8	0.79	2712	21	22.2	1.18	3651	33	35.4	1.96	4740	21	23.0	1.15	3505	44	46.8	2.10	4378	61	64.6	2.63	4976	
		Double invert	11	12.1	0.99	2839	19	21.4	1.07	3215	30	32.3	1.64	3888	21	23.4	1.51	3122	44	47.0	2.67	3186	59	62.6	2.84	4345	
		N4N	11	12.3	0.95	2073	22	22.9	1.29	2302	32	33.5	1.08	2621	20	23.2	1.99	2153	44	46.8	2.44	2424	57	63.1	3.75	2981	
		Single invert	11	11.9	0.57	2391	21	22.6	0.97	3828	31	33.0	1.83	5413	22	23.1	1.20	3000	44	47.6	2.59	4235	58	64.9	3.96	4911	
	Double point	Double invert	11	11.3	0.48	3128	21	22.7	1.06	2828	29	32.7	2.00	3724	21	22.6	1.07	3188	45	46.9	1.20	3204	60	63.5	2.88	4441	
		N4N	11	12.5	0.97	2083	19	21.9	2.23	2237	30	32.5	1.84	2628	21	22.7	1.16	2166	46	47.9	1.73	2451	59	62.4	2.63	2940	
		Single invert	11	11.7	0.48	2548	20	21.5	1.08	2721	27	32.9	3.00	3348	22	22.7	1.06	2900	43	46.8	1.69	4069	58	61.9	2.38	4914	
		Double invert	10	11.3	0.82	2605	20	21.1	1.29	2821	30	31.1	1.10	3085	21	22.9	1.10	3285	43	45.4	1.51	3501	57	60.5	2.22	4179	
	N-point	Single invert	10	11.7	0.67	2047	20	21.4	1.17	2100	29	31.6	1.51	2573	22	22.6	0.84	2152	43	46.1	2.38	2322	57	61.2	2.20	2821	
		Double invert	10	11.3	0.67	2394	20	21.9	1.37	3225	30	33.1	1.66	3471	21	23.3	0.95	2699	42	46.1	2.23	4091	59	62.5	2.12	4553	
		N4N	10	11.3	0.67	2096	18	21.1	1.52	2464	28	32.0	2.49	3198	21	22.8	1.23	2400	43	45.7	1.83	3559	60	61.7	1.34	3930	
		Double invert	11	12.0	0.67	2052	19	21.6	1.26	2101	28	32.1	2.33	2483	21	22.8	1.23	2118	43	46.2	1.62	2355	59	61.0	1.63	2842	
Tournament	Single point	Single invert	12	12.7	0.67	2906	19	22.4	1.65	2733	31	33.1	1.60	4014	22	23.3	0.95	2633	45	47.4	1.35	3561	61	62.3	0.95	4785	
		Double invert	11	11.9	1.29	2659	20	22.3	1.42	2853	32	33.7	1.64	3365	21	22.7	0.82	2883	43	47.3	2.71	3453	61	63.3	2.31	4141	
		N4N	12	12.9	0.88	2137	20	22.0	1.76	2291	31	33.1	1.29	2549	22	24.0	1.05	2146	45	47.8	2.15	2585	59	62.0	1.89	2975	
		Single invert	11	12.0	0.82	2400	20	22.1	0.99	2909	30	32.6	1.96	4068	22	23.1	0.99	2658	46	47.9	1.20	3438	60	62.8	1.40	4244	
	Double point	Double invert	11	12.0	0.94	3091	20	22.5	1.51	2814	30	32.7	2.16	3511	21	22.5	1.18	3087	44	47.1	2.23	3641	60	62.5	1.65	3962	
		N4N	10	12.2	1.48	2094	20	22.1	1.20	2381	31	32.5	0.85	2608	22	23.2	0.79	2155	44	47.6	1.96	2553	59	63.2	2.62	2994	
		Single invert	11	11.9	0.74	2334	20	21.4	1.07	2773	31	32.1	1.29	3271	21	22.3	1.34	2953	43	46.9	2.33	3214	59	63.3	2.54	3652	
		Double invert	10	11.4	0.84	2427	18	21.2	1.48	2328	28	31.0	1.70	2878	21	22.5	1.08	2783	46	46.7	1.06	3262	57	62.2	2.57	3729	
	N-point	Single invert	11	11.9	0.74	2334	20	21.4	1.07	2773	31	32.1	1.29	3271	21	22.3	1.34	2953	43	46.9	2.33	3214	59	63.3	2.54	3652	
		Double invert	10	11.4	0.84	2427	18	21.2	1.48	2328	28	31.0	1.70	2878	21	22.5	1.08	2783	46	46.7	1.06	3262	57	62.2	2.57	3729	
		N4N	11	12.1	0.74	2079	20	21.3	1.16	2225	30	32.7	1.34	2478	21	22.4	1.35	2205	43	46.4	2.41	2367	58	62.9	2.33	2746	
		Single invert	10	11.6	0.97	2308	20	21.5	0.97	2353	31	32.7	0.95	3379	22	22.9	0.74	2650	44	46.2	1.69	3774	58	62.3	3.27	3730	
Uniform	Double invert	11	12.0	1.49	2295	20	21.2	1.03	2417	30	32.1	1.37	2914	21	22.3	0.82	2788	44	46.0	1.25	3247	58	61.7	2.06	3479		
	N4N	11	12.5	0.85	2049	19	21.4	1.71	2279	30	32.2	1.40	2471	21	23.3	0.95	2128	43	45.6	2.01	2514	59	61.1	1.45	2751		
	Simulated Annealing	Perturbation	Single invert	11	11.8	0.42	--	22	23.7	1.06	--	37	38.1	0.74	--	22	23.1	0.57	--	47	50.0	1.49	--	64	70.8	3.79	--
			Double invert	10	10.9	0.32	--	20	21.5	0.71	--	32	33.5	1.43	--	21	21.8	0.41	--	44	46.3	1.33	--	63	64.9	1.29	--
N4N			13	14.7	0.82	--	21	22.5	0.97	--	32	34.0	1.41	--	22	23.9	1.45	--	45	47.9	2.23	--	59	61.7	2.36	--	
E100			--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--		
E250		Single invert	10	11.2	0.79	--	17	20.5	1.90	--	30	32.8	1.62	--	20	21.3	1.06	--	42	44.6	2.07	--	59	63.6	2.50	--	
		Double invert	10	11.1	0.74	--	18	20.4	1.07	--	30	31.1	1.10	--	20	21.7	0.95	--	42	44.9	1.60	--	57	61.7	2.71	--	
		N4N	12	13.2	1.03	--	21	23.3	1.64	--	31	32.5	1.58	--	22	24.1	1.66	--	46	47.3	1.83	--	57	61.3	2.31	--	
		E100	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--		
E500		Single invert	10	11.2	0.79	--	17	20.5	1.90	--	30	32.8	1.62	--	20	21.3	1.06	--	42	44.6	2.07	--	59	63.6	2.50	--	
		Double invert	10	11.1	0.74	--	18	20.4	1.07	--	30	31.1	1.10	--	20	21.7	0.95	--	42	44.9	1.60	--	57	61.7	2.71	--	
		N4N	12	13.2	1.03	--	21	23.3	1.64	--	31	32.5	1.58	--	22	24.1	1.66	--	46	47.3	1.83	--	57	61.3	2.31	--	
		E100	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--		
F100	Single invert	10	11.2	0.79	--	17	20.5	1.90	--	30	32.8	1.62	--	20	21.3	1.06	--	42	44.6	2.07	--	59	63.6	2.50	--		
	Double invert	10	11.1	0.74	--	18	20.4	1.07	--	30	31.1	1.10	--	20	21.7	0.95	--	42	44.9	1.60	--	57	61.7	2.71	--		
	N4N	12	13.2	1.03	--	21	23.3	1.64	--	31	32.5	1.58	--	22	24.1	1.66	--	46	47.3	1.83	--	57	61.3	2.31	--		
	F100	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--			
F250	Single invert	10	11.2	0.79	--	17	20.5	1.90	--	30	32.8	1.62	--	20	21.3	1.06	--	42	44.6	2.07	--	59	63.6	2.50	--		
	Double invert	10	11.1	0.74	--	18	20.4	1.07	--	30	31.1	1.10	--														

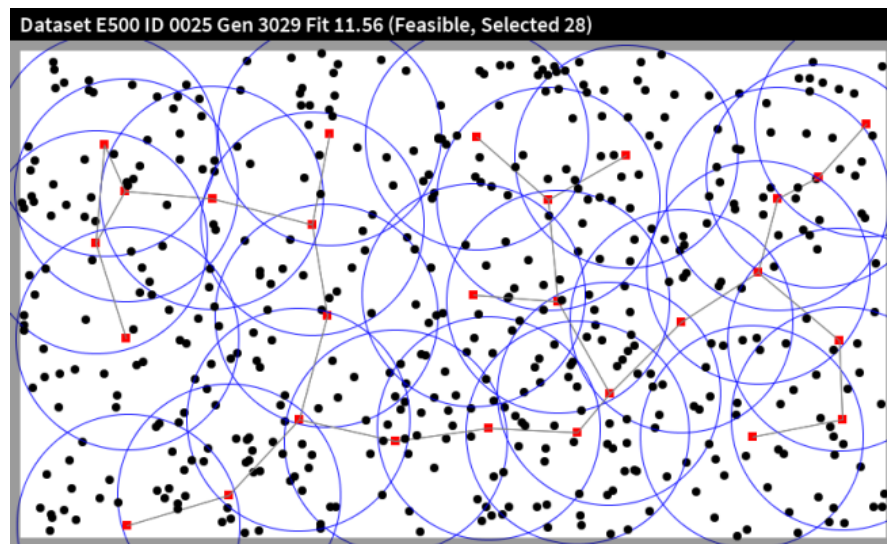
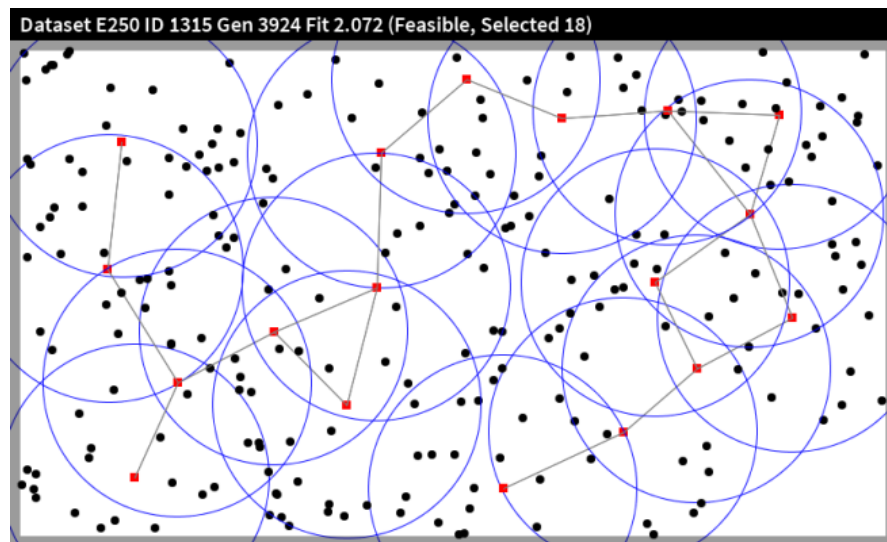
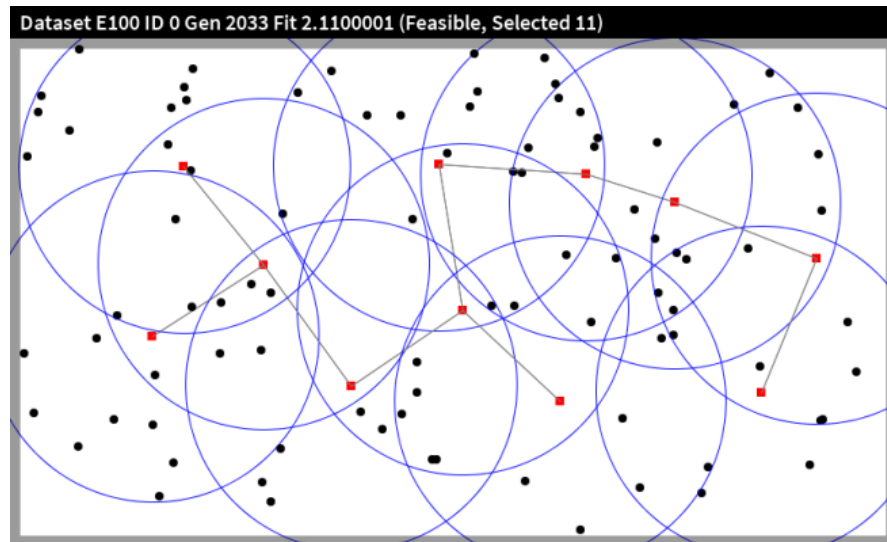


Figure 3: E type random datasets

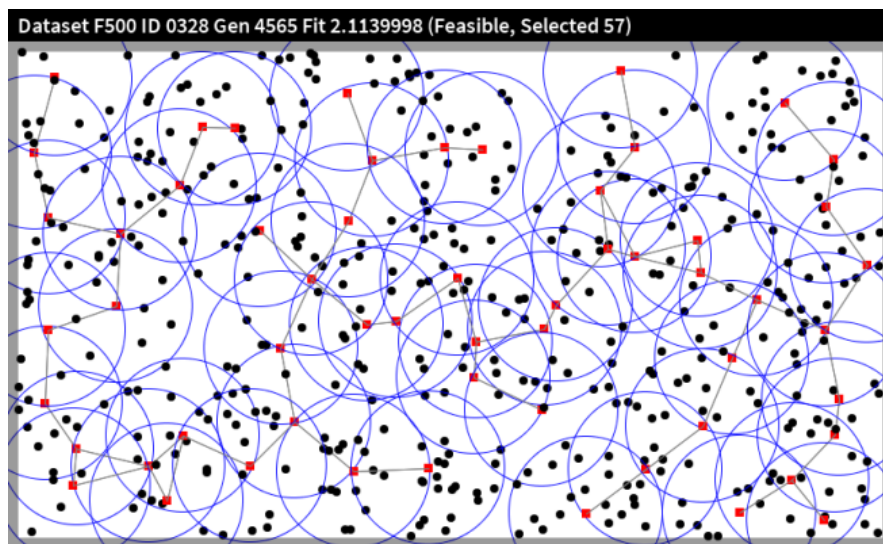
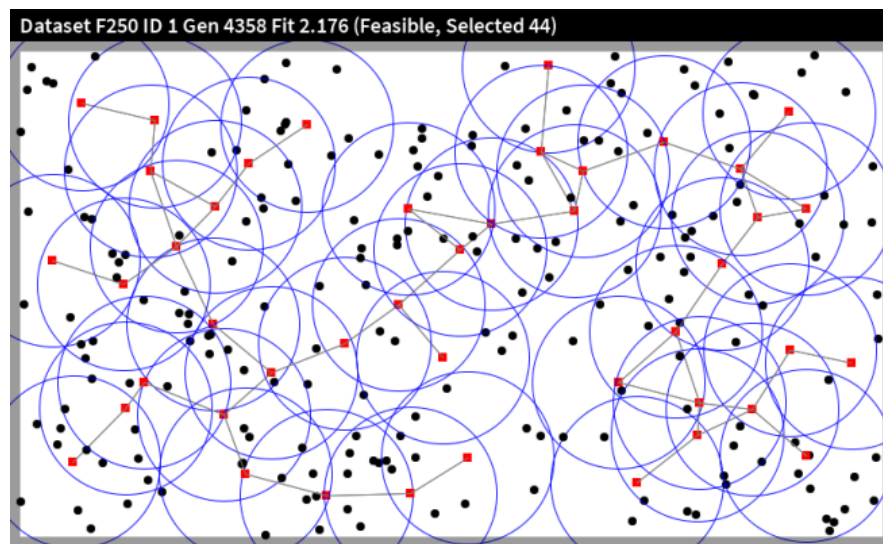
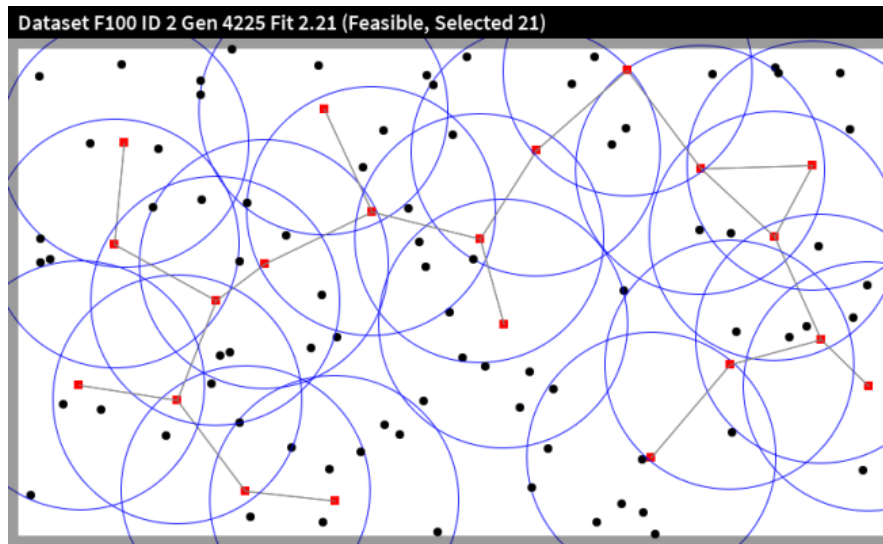


Figure 4: F type random datasets

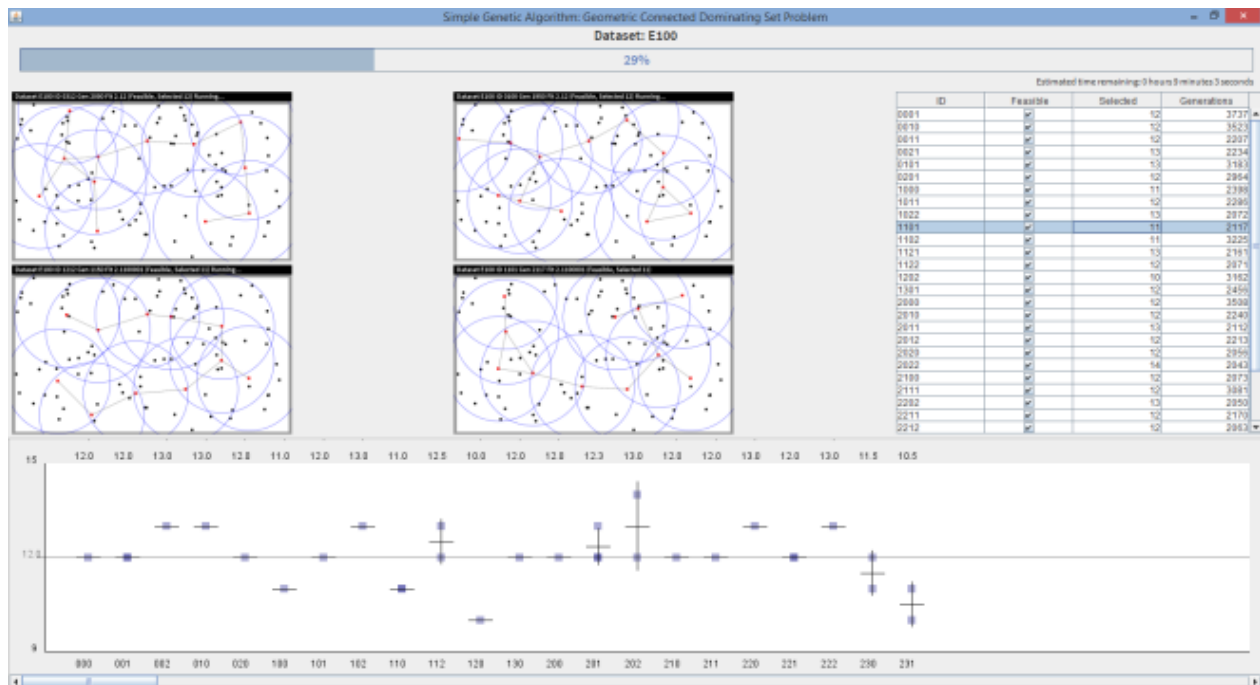
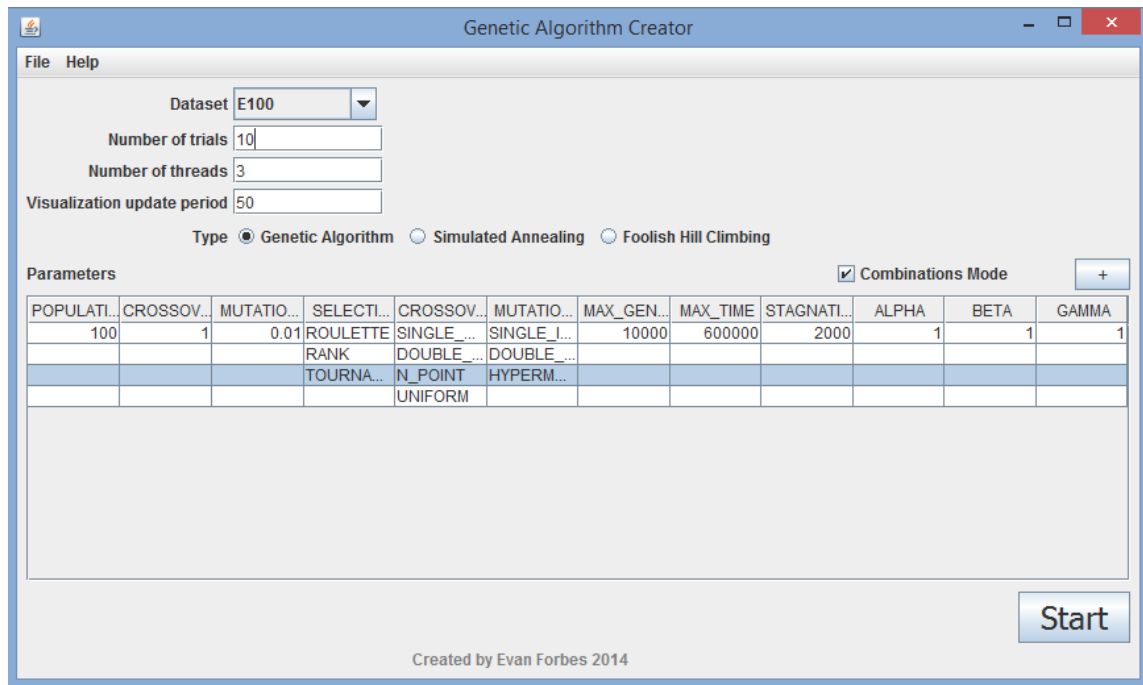


Figure 5: Custom SGA engine GUI