EA/MOEA Technical Report

William Riley-Land

6 Oct 2011

Methodology

I. EA

The genotype was composed of an array of booleans, each representing the state of a router: true for enabled, and false for disabled. Each member of the population was assigned a fitness score, calculated as follows: *1 + <number of routers left activated> - (<penalty coefficient> * <number of uncut paths>)*. Initially, a population of size μ was generated with each individual assigned random alleles drawn uniformly from the set of all possible values.

For selecting parents eligible for mating, k-tournament with replacement was used. First, a k-tournament was performed to select a mating pool half the size of the desired number of offspring. For each round of the k-tournament, *k* individuals were selected at random from the population, with no possibility that an individual could be selected twice in the same round. Once the *k* had been selected, the individual with the highest fitness was selected as the winner. A new round was then started, drawing individuals again from the population at large, with nothing to prevent any members of the last tournament from being involved in this next tournament. This process was repeated until the desired number of individuals had been added to the mating pool.

For each member selected via the k-tournament, a mate was selected randomly from the same resultant pool. Having implemented no check to prevent it, there was allowed the possibility that the parent might be chosen to mate with itself, producing two children identical to the parent.

Once two members of the population were chosen to mate, n-point crossover was used to produce their two offspring. One offspring was produced by first picking *n* points in the genotype. Individual alleles were copied from the first parent until one of these points was encountered. When one was encountered, the child's alleles were then taken from the second parent, until another such point was reached, at which point the alleles were again taken from the first parent, and so on. The other child's alleles were transferred in the same manner, using the same *n* points chosen previously, but starting by taking the alleles of the second parent, then the first, and so on.

Once the children were generated, they were potentially mutated. A probability *p* was given that any individual gene would be mutated. Each gene in each child was mutated with probability *p*. When the mutation did occur, a simple bit-flip mutation was used. In other words, when a gene was mutated, if it had a value of true, it would become false, and vice versa.

For selection during the survival step, k-tournament was used again, this time performing tournaments until μ members were selected from the population of parents and offspring.

II. MOEA

The multi-objective evolutionary algorithm was implemented in the same way, except that parent selection was performed using Fitness Proportional selection. Each member of the population was assigned a probability for selection as a parent based on the individual member's fitness compared to the sum of the fitness values of each member of the population. Members were then selected from the population at random using the Stochastic Universal Sampling Algorithm until the desired number of possible parents were selected.

The other difference was that the fitness value used was based on Pareto Fronts (although the measure of the fitness values for this report have been translated to be the same measure as the values for the EA). For each run of the algorithm, a population was generated randomly, then the Pareto fronts were calculated for that population. The fronts were recalculated following the mating step of each generation.

In order to calculate the fronts, all non-dominated members were added to the first front. Then, the members of the first front were removed from consideration for the next front. All non-dominated members of the resulting subset became the second front. The members of the second front were removed from consideration, and the next front was calculated in the same way. This continued until all members of the population had been assigned to a front.

Once the fronts were calculated, a fitness score was assigned to each member of the population equal to the number of solutions in the fronts after the front it belonged to. After this raw fitness value was calculated, fitness sharing was performed. For each front, each solution in that front was assigned a fitness by calculating the Hamming distance between it and each member of the front (including itself). The fitness of the solution was then divided by the sum of *1 - <hamming distance> / 7* for all solutions in the front that were within a distance of 7 (including itself).

Experiment

I. EA

The Evolutionary Algorithm was used in 4 experiments, each consisting of 3 runs. Each run consisted of 10,000 iterations unless the best population fitness did not change for 1,000 iterations, in which case the run was ended. A population size of 100 was used and in each generation 20 offspring were produced, except in experiment 3 in which a population size of 200 was used. A penalty coefficient of 10.5 was used for all experiments. The other variables were set according to Table 1.

Table 1

	Experiment 1	***Experiment 2***	***Experiment 3***	***Experiment 4***
k for parent selection	10	20	10	10
n points for crossover	5	3	9	3
Mutation probability	0.004	0.000667	0.0009	0.0002
k for survival selection	3	7	3	3

II. MOEA

The Multi-Objective Evolutionary Algorithm was used in 9 experiments, each consisting of 30 runs. Experiments were designed to test combinations of 3 values for 2 variables, resulting in 9 combinations. Each run consisted of 10,000 evaluations (not iterations). A population size of 100 was used and each generation 10 offspring were produced. A penalty coefficient of 10.5 was used for all experiments. Survival selection was performed using tournaments of 3 individuals. The other variables were set according to Table 2.

Table 2

	Combo 1	***Combo 2***	***Combo 3***	***Combo 4***	***Combo 5***	***Combo 6***	***Combo 7***	***Combo 8***	***Combo 9***
n points for crossover	7	7	7	9	9	9	11	11	11
Mutation probability	0.0001	0.0009	0.0018	0.0001	0.0009	0.0018	0.0001	0.0009	0.0018

Results

I. Random Search

Random search was performed on each network for 10,000 evaluations. The best fitness and the evaluation on which it was found are listed in Table 3.

Table 3

	Evaluation	***Best Fitness***
Network 1	8531	568
Network 2	1410	1612
Network 3	8854	10266
Network 4	9285	−2972.5

II. EA

The best runs for experiments 1-4 are shown in charts 1-4, respectively. Table 4 shows the best solution found for each network.

Table 4

	Best Final Fitness (All runs)
Network 1	965
Network 2	2916
Network 3	16016
Network 4	3135

Chart 1

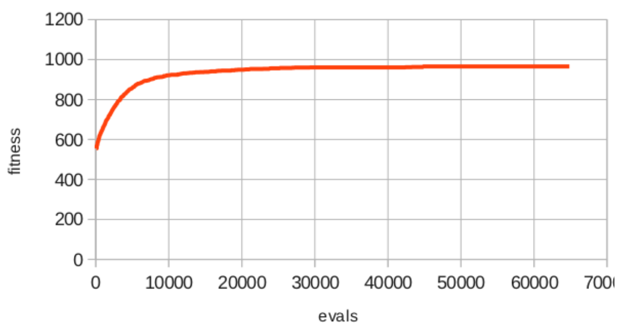


Chart 2

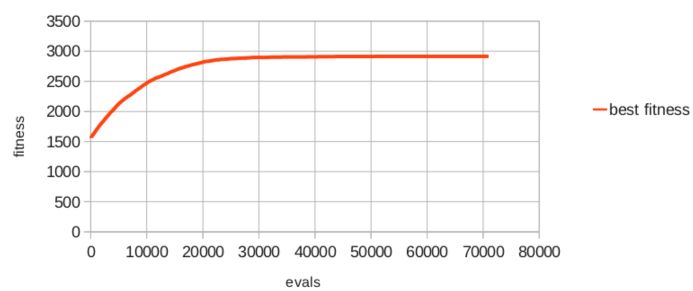


Chart 3

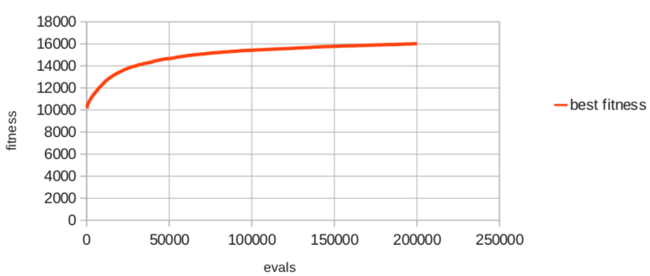
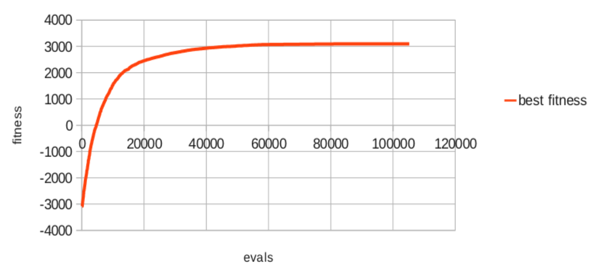


Chart 4



III. MOEA

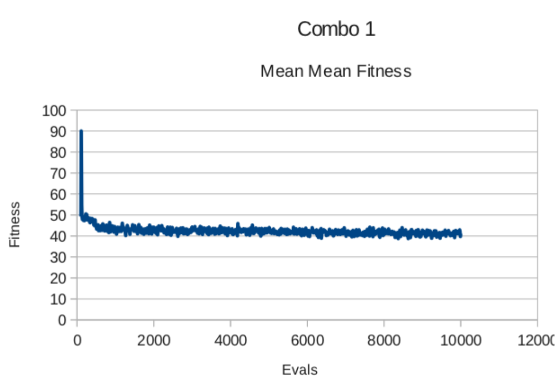
The charts labeled Combo 1-9 show the mean fitness at each evaluation, averaged over all 30 runs for each combination. Table 5 shows the best (multi-objective) fitness across all runs for each combination. Table 6 illustrates the best fitness of each combination averaged over the 30 runs. Tables 7-9 show the F-Test score for each pair of combinations, the type of variance for each pair of combinations, and the two-tailed T-Test score for each pair of combinations, respectively. Each T-Test assumed either equal or inequal variance based on the F-Score for that pair of combinations.

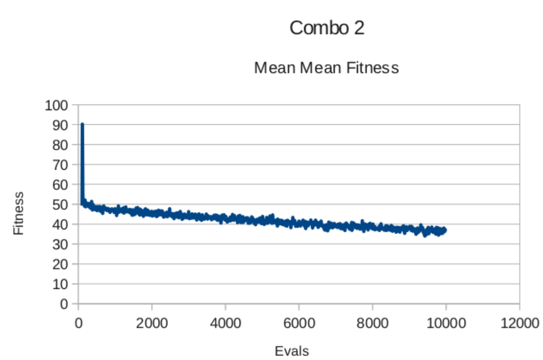
Table 5

	Number of Deactivated Routers	***Number of cut paths***
Combo 1	7916	200
Combo 2	6730	200
Combo 3	6758	200
Combo 4	8013	200
Combo 5	6822	200
Combo 6	6754	200
Combo 7	7784	200
Combo 8	6781	200
Combo 9	6685	200

Table 6

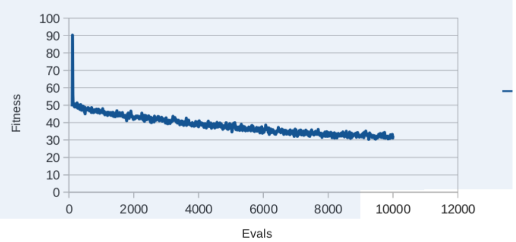
	Combo 1	***Combo 2***	***Combo 3***	***Combo 4***	***Combo 5***	***Combo 6***	***Combo 7***	***Combo 8***	***Combo 9***
Mean Best Fitness Over 30 Runs	12008.8	13167.77	13187.1	12034.37	13236.57	13233.97	12079.03	13255.27	13298.37





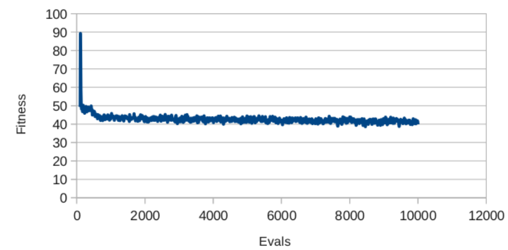
Combo 3

Mean Mean Fitness



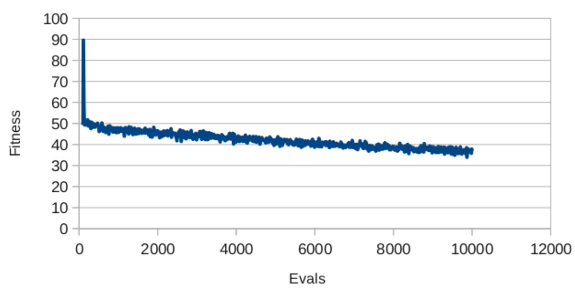
Combo 4

Mean Mean Fitness



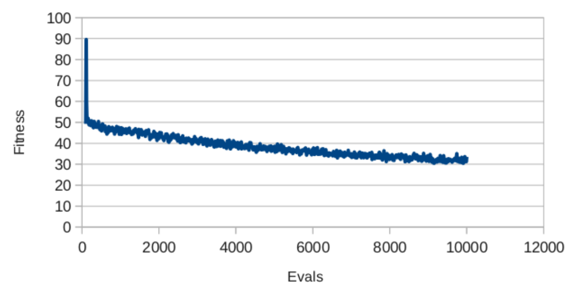
Combo 5

Mean Mean Fitness



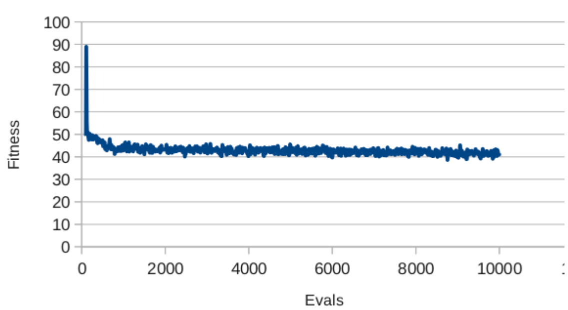
Combo 6

Mean Mean Fitness



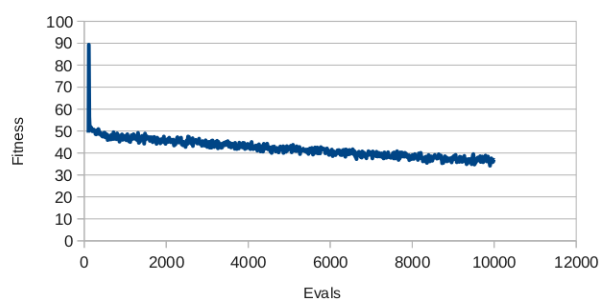
Combo 7

Mean Mean Fitness



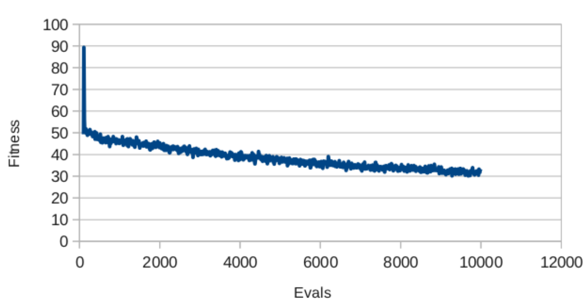
Combo 8

Mean Mean Fitness



Combo 9

Mean Mean Fitness



Tables 7-9

F-Test

	Combo 1	Combo 2	Combo 3	Combo 4	Combo 5	Combo 6	Combo 7	Combo 8	Combo 9
Combo 1	X	X	X	X	X	X	X	X	X
Combo 2	0.0271254494	X	X	X	X	X	X	X	X
Combo 3	0.2036773688	0.3363444649	X	X	X	X	X	X	X
Combo 4	0.6545505585	0.075554867	0.4078119053	X	X	X	X	X	X
Combo 5	0.1459144912	0.4373381837	0.8525096351	0.311384184	X	X	X	X	X
Combo 6	0.0017667234	0.3308671316	0.0550472747	0.0067203314	0.0822383856	X	X	X	X
Combo 7	0.5864423372	0.0924160396	0.4644715275	0.9229220475	0.3594869344	0.0088107394	X	X	X
Combo 8	0.0570066765	0.7507084052	0.5186598521	0.1425304466	0.6455348976	0.1982153917	0.1702671982	X	X
Combo 9	0.149336336	0.4300602061	0.8622883032	0.3173240971	0.9900595907	0.0801217367	0.3660094186	0.6366346968	X

F-Critical 161.4476388 (All F's are less than F-critical)

Variance

	Combo 1	Combo 2	Combo 3	Combo 4	Combo 5	Combo 6	Combo 7	Combo 8	Combo 9
Combo 1	X	X	X	X	X	X	X	X	X
Combo 2	inequal	X	X	X	X	X	X	X	X
Combo 3	inequal	inequal	X	X	X	X	X	X	X
Combo 4	inequal	equal	equal	X	X	X	X	X	X
Combo 5	inequal	inequal	inequal	inequal	X	X	X	X	X
Combo 6	inequal	inequal	inequal	inequal	equal	X	X	X	X
Combo 7	inequal	equal	equal	inequal	equal	equal	X	X	X
Combo 8	inequal	inequal	inequal	inequal	inequal	inequal	inequal	X	X
Combo 9	inequal	inequal	inequal	inequal	inequal	inequal	inequal	inequal	X

T-Test (All two-tailed, with either equal or inequal variance assumed)

	Combo 1	Combo 2	Combo 3	Combo 4	Combo 5	Combo 6	Combo 7	Combo 8	Combo 9
Combo 1	X	X	X	X	X	X	X	X	X
Combo 2	2.84669E-056	X	X	X	X	X	X	X	X
Combo 3	2.02696E-059	0.0970320345	X	X	X	X	X	X	X
Combo 4	0.0974686166	6.15038E-064	1.20522E-062	X	X	X	X	X	X
Combo 5	5.61032E-060	8.95876E-008	0.0001562289	2.42309E-062	X	X	X	X	X
Combo 6	1.06218E-053	4.49003E-009	0.000059025	9.75172E-057	0.8046933257	X	X	X	X
Combo 7	1.88396E-005	3.13495E-063	6.44515E-062	0.0029649135	2.23480E-063	4.15138E-066	X	X	X
Combo 8	5.68455E-059	3.22032E-011	3.00825E-007	9.00051E-062	0.1101848733	0.0359533971	1.44478E-061	X	X
Combo 9	3.71848E-061	1.23607E-016	3.00825E-007	1.42849E-063	3.38889E-006	1.09464E-007	2.94693E-063	0.0004334234	X

Discussion

One interesting thing to note is that Combination 9, the best combination in the MOEA experiments, had an average fitness that decreased the most over the 10,000 evaluations and 30 runs. Perhaps the lower average indicates a higher diversity of solutions, which allowed Combination 9 to find better solutions.

Combination 5 and 6 appear likely to be equivalent according to the T-Tests (roughly an 80% chance). Combinations 2 and 3, as well as 5 and 8, and 1 and 4, all had a > 10% chance of being from the same distribution. While some other pairs of combinations had a > 1% chance of being from the same distribution, most pairs had a very small chance. This seems to indicate that the variables in question have a large impact on the overall performance of the algorithm.

Combination 9 did the best out of the MOEA experiments and, according to the T-Tests, it appears to have a very low probability that it was by random chance that it performed better. This might lead one to imagine that both a higher number of crossover points and a higher rate of mutation would lead to better results even than thos found with Combination 9.

Furthermore, the fact that Combinations 3, 6, and 9 performed very well relative to the other experiments seems to indicate that the high mutation rate increased the effectiveness of the MOEA more than the number of crossover points. Interestingly, according to the T-Tests the higher mutation rate was highly probable to be more effective in combination with the higher number of crossover points. This seems to suggest the importance of genetic diversity when solving the router problem with this MOEA.

For some reason, the EA performed better than the MOEA. But, as expected, the EA and the MOEA both performed better than random search for network 3. The MOEA was also much slower than the EA, while the EA was, to the naked eye, roughly equivalent in speed to the random search. This was probably due to the way the Pareto fronts were calculated. A quick Google Scholar search seemed to indicate that there might already be literature explaining faster implementations than the naïve implementation used in these experiments. This would certainly facilitate further experimentation if it made the MOEA faster.

Since the MOEA populations' average fitness was roughly constant or decreased over the experiments, when one would assume it to be increasing, one might suspect that there was an implementation error in the MOEA. Another apparent problem with the MOEA was that its best Pareto fronts only contained one solution. This may have been due to implementation error, or as suggested by an another reviewer of the results, it may have not been a problem if the MOEA was run for more evaluations. However, due to time constraints and lack of background knowledge, the nature of these problems remain conjecture to the author.

Conclusion

Results seem to indicate that a higher mutation rate and larger number of crossover points increased the effectiveness of the MOEA. To determine that this is the case, more experiments need to be run over a spectrum of larger values for these two variables. Before this could be done, the implementation of the MOEA should be checked in order to determine if the MOEA would have yielded more than single results for its best Pareto front if allowed to run longer, and to determine why the average fitness of the MOEA population was roughly constant or decreasing. Even with these unknowns, it is clear that even simple EAs perform much better than random search for the router problem.