Evolutionary Computation

Assignment 3: PWT Problem

Puzhi Yao (a1205593)
Zhuoying Li (a1675725)
Jingwen Wei (a1671836)
Xueyang Wang (a1690260)

June 4, 2017

1 Operators for Exercise 3

In the Exercise 2, we implemented Greedy-Mutation operator which combines the greedy algorithm and bit-flip mutation to improve both time efficiency and result accuracy. We found out that the greedy algorithm is able to construct a initial good result before conducting bit-flip mutation. This design gives bit-flip mutation a great advantage to avoid useless tries and quickly gets close to the optimal solution.

For the Exercise 3, we reconsidered the multiple points crossover and bit-flip mutation. In the original bit-flip mutation, whatever the length of chromosome is, each bit of chromosome will have a chance to flip. However, this might take long time to approach or exceed the optimal solution. From the experience of previous assignments and experiments, high mutation rate may have a chance to decrease the performance of the entire algorithm. Hence, by modified One-bit-flip mutation, we set a probability on each iteration of the loop, if the random number is higher than the probability, then the mutation process continue, otherwise it will end at current bit position (Operator 1). This process may decrease the overall number of mutation during each generation and it may help the One-bit-flip mutation to quickly reach the optimal solution. On the other hand, if the current packing plan is close to the potential combinations of the packing plan that can generate optimal solution. The One-bit-flip mutation gives a great chance to approach those potential solutions.

For the Crossover operator, we designed a Group-Point-Crossover operator (Operator 2). This operator

Algorithm 1: One-Bit-Flip Mutation

```
while random-Number < probability do
    while index < length of Chromosome do
        Generate a random position and flip the bit
    end
    Flip another bit according to the probability
    random-Number = PseudoRandom.randDouble();
end
```

divides bits of chromosome into several subgroups. Each subgroup has a range of crossover probability, and a random number will decide the bits in which subgroup will be swapped. The Group-Point-Crossover operator treats the bits of chromosome as several group, and each group represents a city. In this format, the high profit packing plan of a single city can be inherited from parents, and cities close to the end of tour will have higher crossover probabilities than cities at start.

2 Dynamic Item Benchmarks for Exercise 3

In order to generate dynamic items for exercise 2, the following class and methods were used:

- PWT.generateDynamicItem: This method uses original items and total generation number to generate dynamic items.

Algorithm 2: Group-Point-Crossover

Compute the total number of bits
Check group-Size and total Number of bits, the total Number of bits must be divided by group-size without remainder
Categorize the bits of the chromosome into N sub-groups, and N = Total Number of Bits / Group Size
The crossover rating R_1 of 1_{st} sub-group is 1
The crossover rating R_2 of 2_{nd} sub-group is $R_2 = R_1+1$
The crossover rating R_3 of 3_{rd} sub-group is $R_3 = R_2+1$
. . .
The crossover rating R_n of n_{th} sub-group is $R_n = R_{n-1}+1$
The total rating tot-R = $R_1 + R_2 + R_3 + ... + R_{n-1} + R_n$
The crossover probability of each sub-group $P_i = R_i$ / tot-R
Normalize all crossover probability and make the sum of all crossover probability of each sub-group equal to 1.
Random a number and check it drops in which probability range of subgroups
Swap all bits in that subgroup

- PWT.readDynamicItem: This method reads previous generated dynamic items and store them in the memory.
- PWT.updateCurrentItems:This method will update item status in instance.items based on the current generation.
- PWT.validateCurrentItems: This method will check the new solution item status in order to void picking unavailable items.

The following operators are grouped and tested in Exercise 3:

- Group-Point-Crossover
- Single-Point-Crossover
- Uniform-Crossover
- Reduced-Surrogate-Crossover
- Arithmetic-Crossover
- HUX-Crossover
- Bit-Flip-Mutation
- One-Bit-Flip-Mutation
- Reversing-Mutation
- Greedy-Mutation
- Interchanging-Mutation
- Dynamic-Bit-Flip-Mutation

Based on above setting, we tested 36 combinations for each algorithm (NSGA2, SPEA2, IBEA) and run 1000, 10000, 50000 and 100000 generations for each combination.

3 Benchmark Results for Exercise 3

The configuration parameters used for each benchmark are listed below:

- Number of Generations = 1000, 10000, 50000 and 100000 times

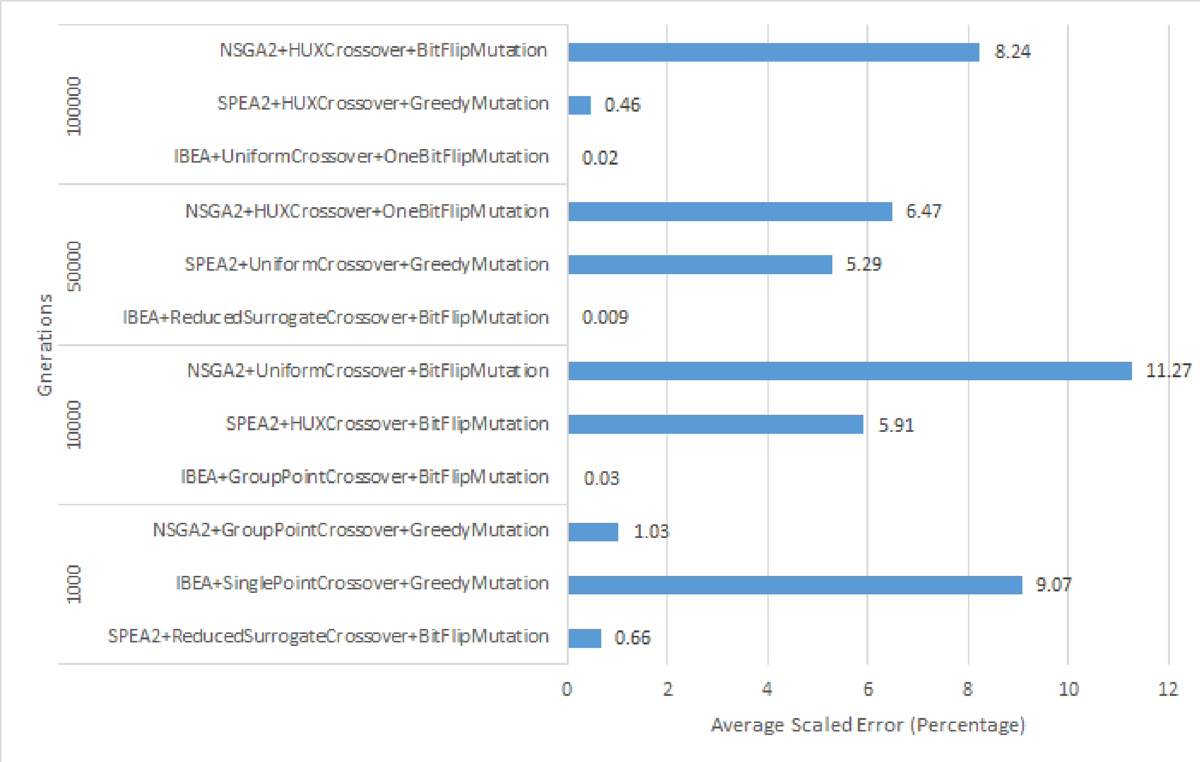

Figure 1: Best three Average scaled Error of NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n100_bounded-strongly-corr_01 instances for different generations. The Error is calculated between Pareto Front and algorithm combination results. IBEA combination performs much better in 10000, 50000 and 100000 generations.

- Total Runs = 10

Figure 1 - 9 show the best three average scaled error of NSGA2, SPEA2 and IBEA + operators combination on benchmark 1 and benchmark 2 of all nine instances for different generations. Figure 1 and 4 illustrate that Greedy Mutation operator performs well for 1000 and 100000 generations. Further, figure 1, 2 and 3 show that the performance of the bit-flip mutation operator is better than other operators in small item set instances. Figure 4 and 7 illustrate that Group-Point-Crossover operator is better used in small or medium size of item set instances. Figure 8 and 9 depict that One-Bit-Flip mutation operator might have better performance in large item set instances.

In conclusion, 1 to 9 show the average scaled errors for different combinations on all 9 instances. Exercise 3 designed Group-Point-Crossover operator and One-Bit-Flip Mutation operator can outperform other operators from Exercise 2 in some situations. All detail performance results are listed in Appendix (Figure 10 - 40).

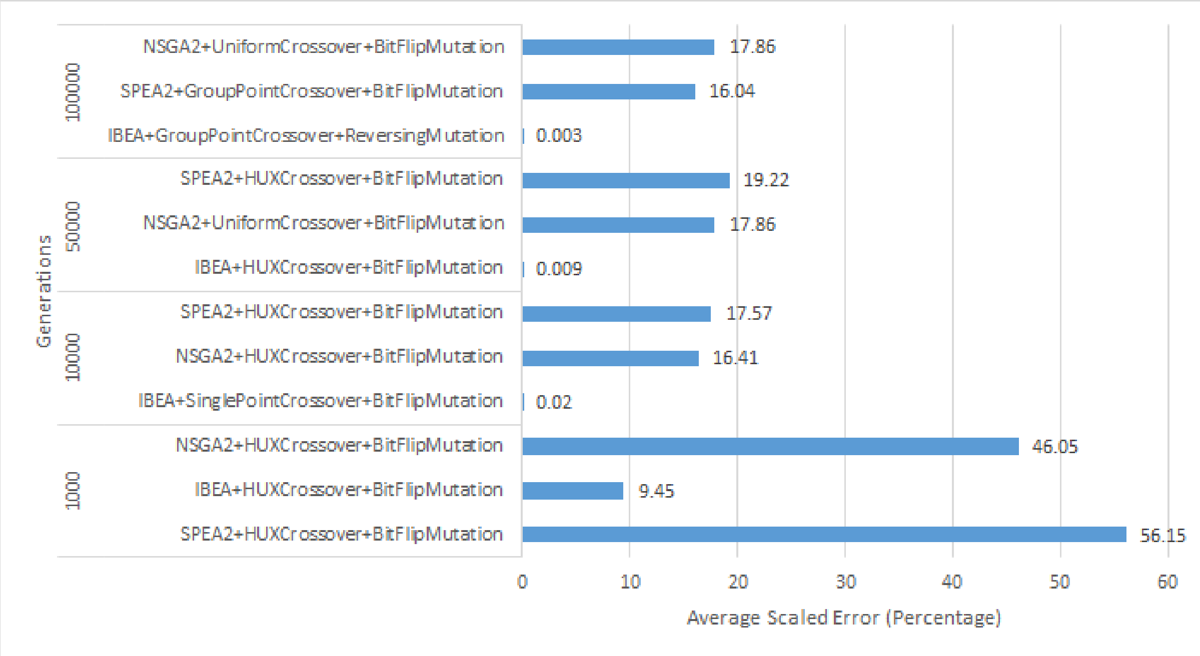

Figure 2: Best three Average scaled Error of NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n100_bounded-strongly-corr_06 instances for different generations. The Error is calculated between Pareto Front and algorithm combination results. IBEA combination performs much better in 10000, 50000 and 100000 generations.

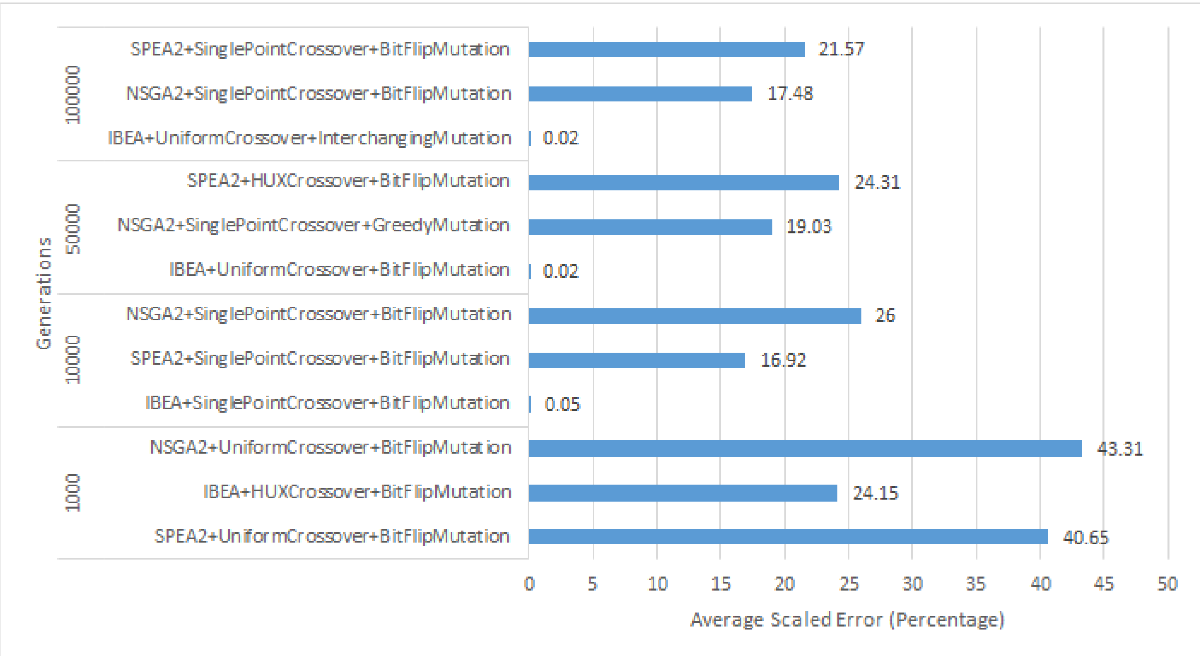

Figure 3: Best three Average scaled Error of NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n100_bounded-strongly-corr_10 instances for different generations. The Error is calculated between Pareto Front and algorithm combination results. IBEA combination performs much better in all four different generations.

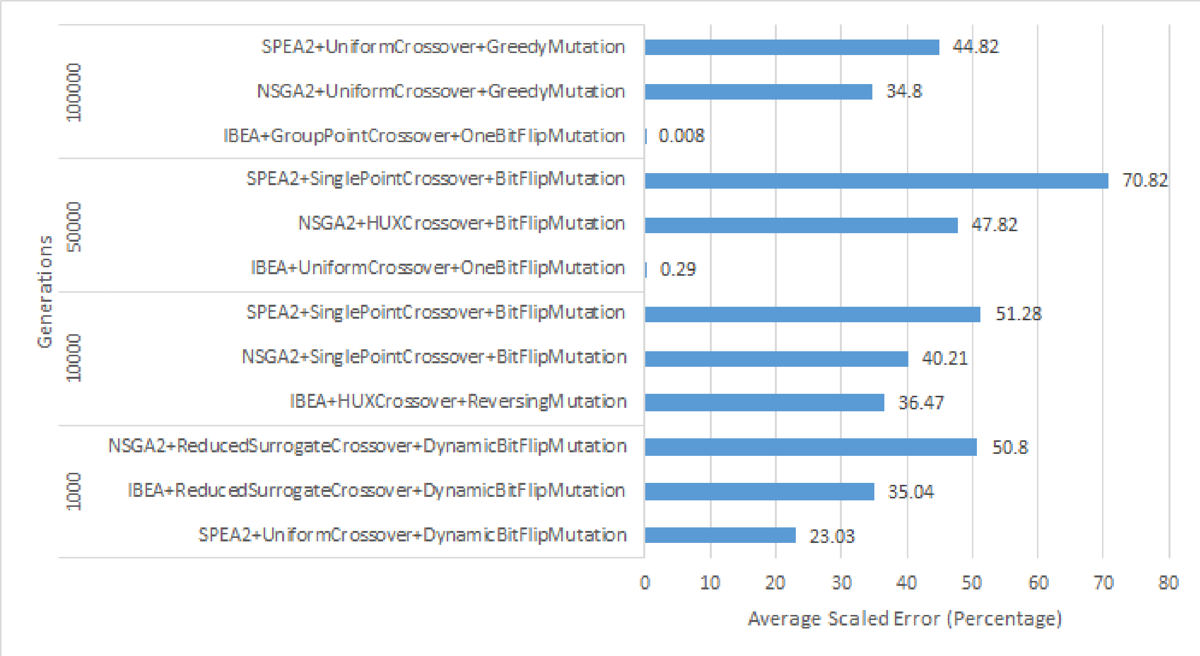

Figure 4: Best three Average scaled Error of NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n500_bounded-strongly-corr_01 instances for different generations. The Error is calculated between Pareto Front and algorithm combination results. IBEA combination performs much better in 50000 and 100000 generations.

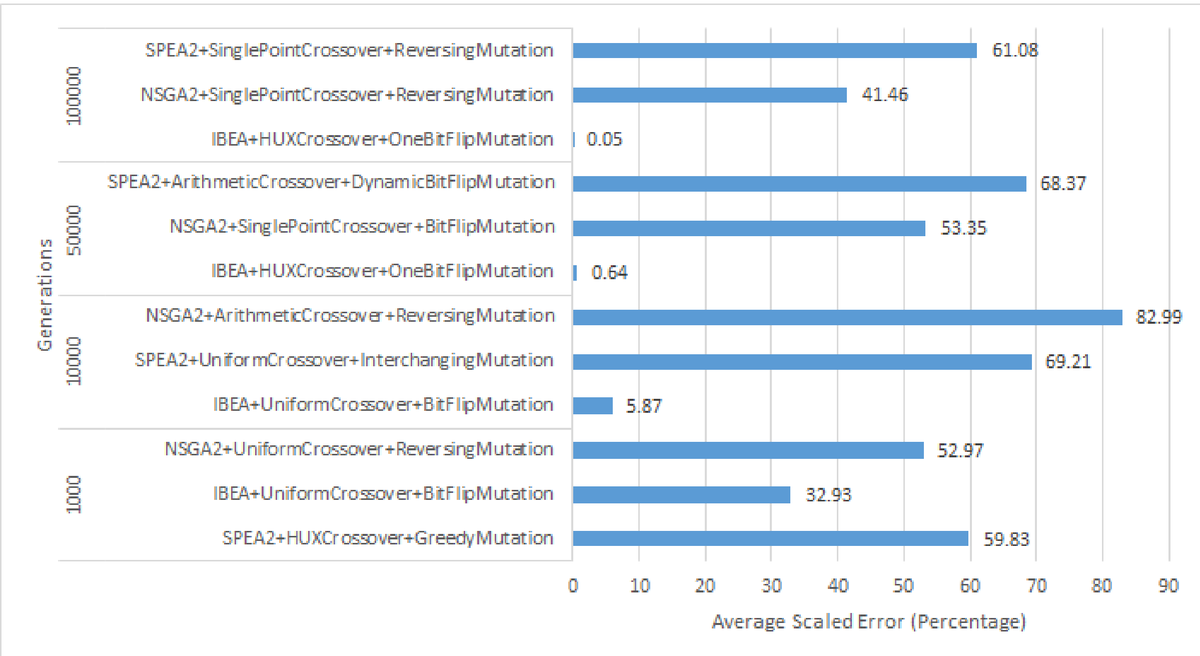

Figure 5: Best three Average scaled Error of NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n500_bounded-strongly-corr_06 instances for different generations. The Error is calculated between Pareto Front and algorithm combination results. IBEA combination performs much better in all four different generations.

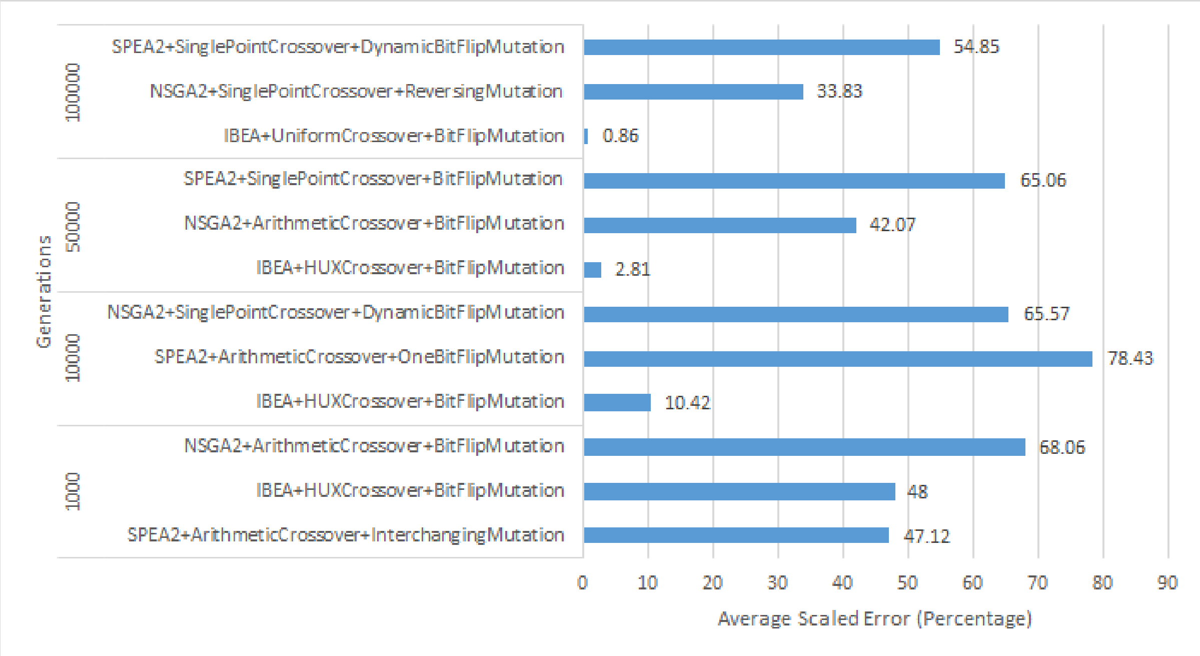

Figure 6: Best three Average scaled Error of NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n500_bounded-strongly-corr_10 instances for different generations. The Error is calculated between Pareto Front and algorithm combination results. IBEA combination performs much better in 10000, 50000 and 100000 generations.

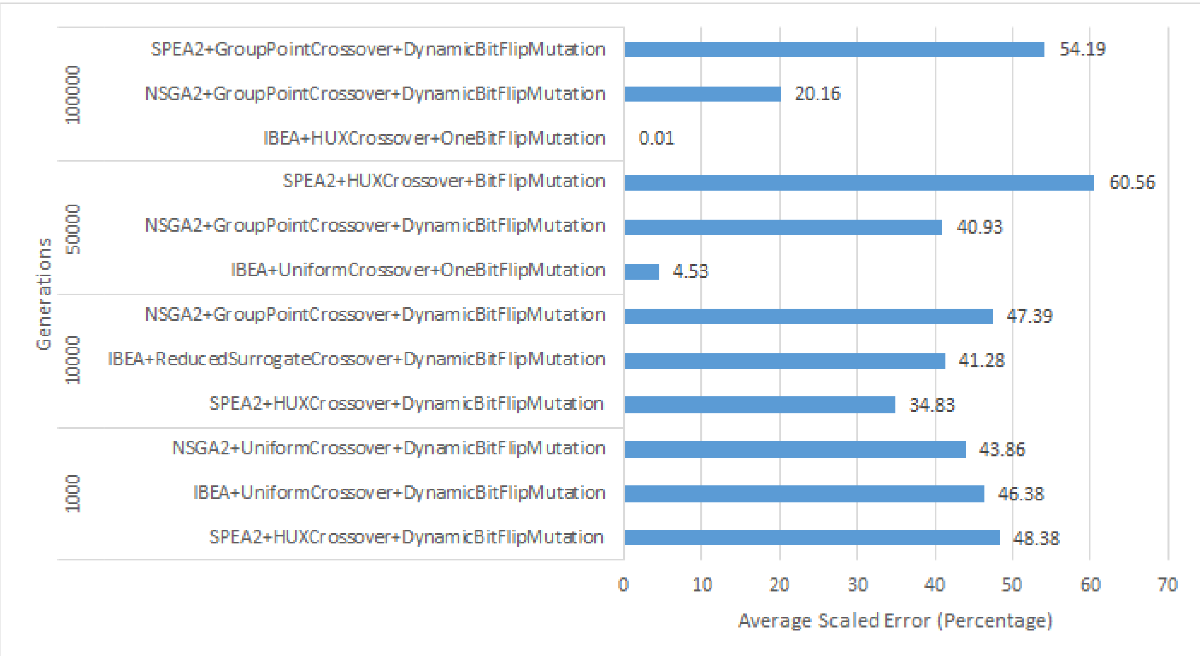

Figure 7: Best three Average scaled Error of NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n1000_bounded-strongly-corr_01 instances for different generations. The Error is calculated between Pareto Front and algorithm combination results. IBEA combination performs much better in 50000 and 100000 generations.

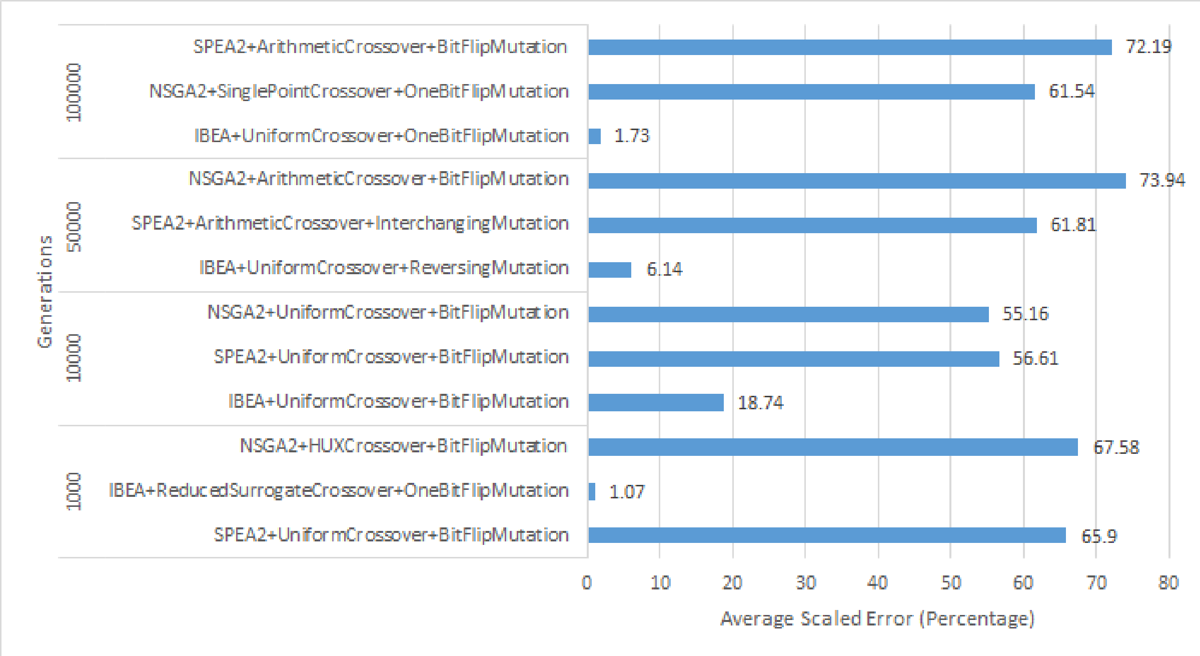

Figure 8: Best three Average scaled Error of NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n1000_bounded-strongly-corr_06 instances for different generations. The Error is calculated between Pareto Front and algorithm combination results. IBEA combination performs much better in all four different generations.

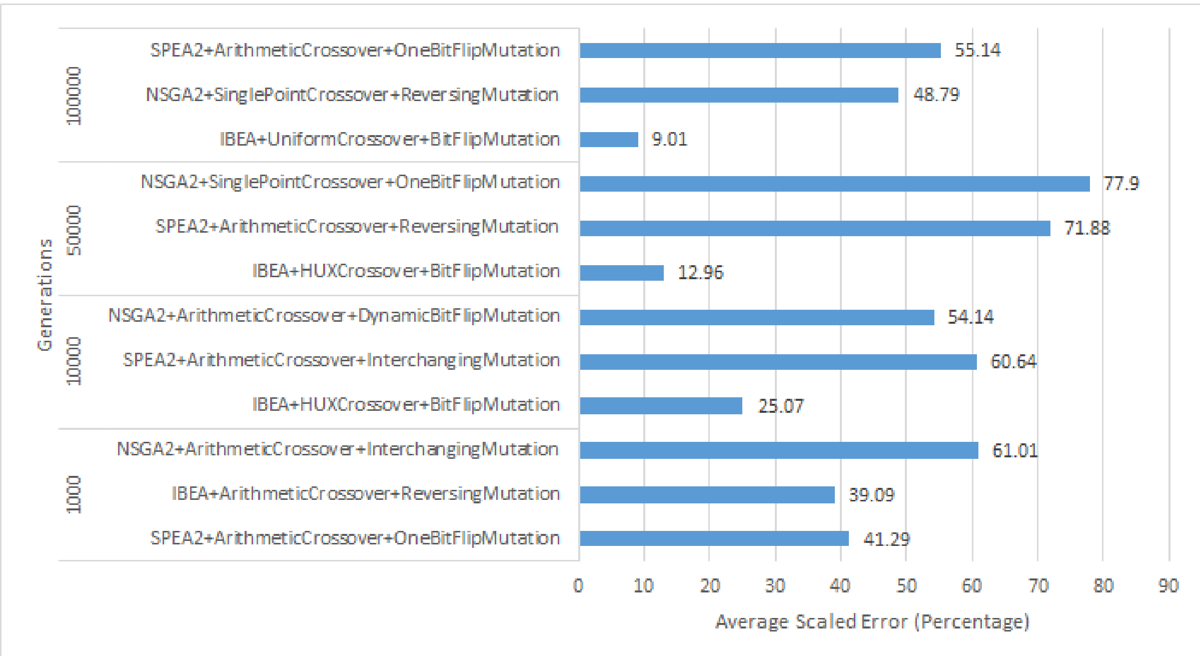

Figure 9: Best three Average scaled Error of NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n1000_bounded-strongly-corr_10 instances for different generations. The Error is calculated between Pareto Front and algorithm combination results. IBEA combination performs much better in 10000, 50000 and 100000 generations.

Appendix

Table 1: Best Combinations Per Algorithm Per instance for 1000 generations

Instance	Algorithm	Crossover Operator	Mutation Operator	Run	Error Percentage
Il101-n100-bounded-strongly-corr-01-1000-generations-ErrorReport	SPEA2	ReducedSurrogateCrossover	BitFlipMutation	RunNum=1	0.66
	NSGA2	GroupPointCrossover	GreedyMutation	RunNum=10	1.03
	IBEA	SinglePointCrossover	GreedyMutation	RunNum=4	9.07
Il101-n100-bounded-strongly-corr-06-1000-generations-ErrorReport	IBEA	HUXCrossover	BitFlipMutation	RunNum=8	9.45
	NSGA2	HUXCrossover	BitFlipMutation	RunNum=2	46.05
	SPEA2	HUXCrossover	BitFlipMutation	RunNum=5	56.15
Il101-n100-bounded-strongly-corr-10-1000-generations-ErrorReport	SPEA2	UniformCrossover	BitFlipMutation	RunNum=3	40.65
	IBEA	HUXCrossover	BitFlipMutation	RunNum=7	24.15
	NSGA2	UniformCrossover	BitFlipMutation	RunNum=10	43.31
Il101-n500-bounded-strongly-corr-01-1000-generations-ErrorReport	SPEA2	UniformCrossover	DynamicBitFlipMutation	RunNum=5	23.03
	IBEA	ReducedSurrogateCrossover	DynamicBitFlipMutation	RunNum=6	35.04
	NSGA2	ReducedSurrogateCrossover	DynamicBitFlipMutation	RunNum=8	50.8
Il101-n500-bounded-strongly-corr-06-1000-generations-ErrorReport	SPEA2	HUXCrossover	GreedyMutation	RunNum=5	59.83
	IBEA	UniformCrossover	BitFlipMutation	RunNum=3	32.93
	NSGA2	UniformCrossover	ReversingMutation	RunNum=7	52.97
Il101-n500-bounded-strongly-corr-10-1000-generations-ErrorReport	SPEA2	ArithmeticCrossover	InterchangingMutation	RunNum=9	47.12
	IBEA	HUXCrossover	BitFlipMutation	RunNum=6	48
	NSGA2	ArithmeticCrossover	BitFlipMutation	RunNum=9	68.06
Il101-n1000-bounded-strongly-corr-01-1000-generations-ErrorReport	SPEA2	HUXCrossover	DynamicBitFlipMutation	RunNum=10	48.38
	IBEA	UniformCrossover	DynamicBitFlipMutation	RunNum=2	46.38
	NSGA2	UniformCrossover	DynamicBitFlipMutation	RunNum=4	43.86
Il101-n1000-bounded-strongly-corr-06-1000-generations-ErrorReport	SPEA2	UniformCrossover	BitFlipMutation	RunNum=10	65.9
	IBEA	ReducedSurrogateCrossover	OneBitFlipMutation	RunNum=8	1.07
	NSGA2	HUXCrossover	BitFlipMutation	RunNum=9	67.58
Il101-n1000-bounded-strongly-corr-10-1000-generations-ErrorReport	SPEA2	ArithmeticCrossover	OneBitFlipMutation	RunNum=5	41.29
	IBEA	ArithmeticCrossover	ReversingMutation	RunNum=9	39.09
	NSGA2	ArithmeticCrossover	InterchangingMutation	RunNum=5	61.01

Table 2: Best Combinations Per Algorithm Per instance for 10000 generations

Instance	Algorithm	Crossover Operator	Mutation Operator	Run	Error Percentage
Il101-n100-bounded-strongly-corr-01-10000-generations-ErrorReport	IBEA	GroupPointCrossover	BitFlipMutation	RunNum=9	0.03
	SPEA2	HUXCrossover	BitFlipMutation	RunNum=4	5.91
	NSGA2	UniformCrossover	BitFlipMutation	RunNum=4	11.27
Il101-n100-bounded-strongly-corr-06-10000-generations-ErrorReport	IBEA	SinglePointCrossover	BitFlipMutation	RunNum=9	0.02
	NSGA2	HUXCrossover	BitFlipMutation	RunNum=8	16.41
	SPEA2	HUXCrossover	BitFlipMutation	RunNum=2	17.57
Il101-n100-bounded-strongly-corr-10-10000-generations-ErrorReport	IBEA	SinglePointCrossover	BitFlipMutation	RunNum=5	0.05
	SPEA2	SinglePointCrossover	BitFlipMutation	RunNum=8	16.92
	NSGA2	SinglePointCrossover	BitFlipMutation	RunNum=6	26
Il101-n500-bounded-strongly-corr-01-10000-generations-ErrorReport	IBEA	HUXCrossover	ReversingMutation	RunNum=4	36.47
	NSGA2	SinglePointCrossover	BitFlipMutation	RunNum=5	40.21
	SPEA2	SinglePointCrossover	BitFlipMutation	RunNum=8	51.28
Il101-n500-bounded-strongly-corr-06-10000-generations-ErrorReport	IBEA	UniformCrossover	BitFlipMutation	RunNum=4	5.87
	SPEA2	UniformCrossover	InterchangingMutation	RunNum=4	69.21
	NSGA2	ArithmeticCrossover	ReversingMutation	RunNum=7	82.99
Il101-n500-bounded-strongly-corr-10-10000-generations-ErrorReport	IBEA	HUXCrossover	BitFlipMutation	RunNum=1	10.42
	SPEA2	ArithmeticCrossover	OneBitFlipMutation	RunNum=1	78.43
	NSGA2	SinglePointCrossover	DynamicBitFlipMutation	RunNum=2	65.57
Il101-n1000-bounded-strongly-corr-01-10000-generations-ErrorReport	SPEA2	HUXCrossover	DynamicBitFlipMutation	RunNum=2	34.83
	IBEA	ReducedSurrogateCrossover	DynamicBitFlipMutation	RunNum=4	41.28
	NSGA2	GroupPointCrossover	DynamicBitFlipMutation	RunNum=4	47.39
Il101-n1000-bounded-strongly-corr-06-10000-generations-ErrorReport	IBEA	UniformCrossover	BitFlipMutation	RunNum=3	18.74
	SPEA2	UniformCrossover	BitFlipMutation	RunNum=10	56.61
	NSGA2	UniformCrossover	BitFlipMutation	RunNum=8	55.16
Il101-n1000-bounded-strongly-corr-10-10000-generations-ErrorReport	IBEA	HUXCrossover	BitFlipMutation	RunNum=8	25.07
	SPEA2	ArithmeticCrossover	InterchangingMutation	RunNum=8	60.64
	NSGA2	ArithmeticCrossover	DynamicBitFlipMutation	RunNum=8	54.14

Figure 10: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n100_bounded-strongly-corr_01 instances compare to Pareto-optimal front for 10000 generations. All three algorithm combinations are close to the Pareto-optimal front, but IBEA combination is slightly better than other two.

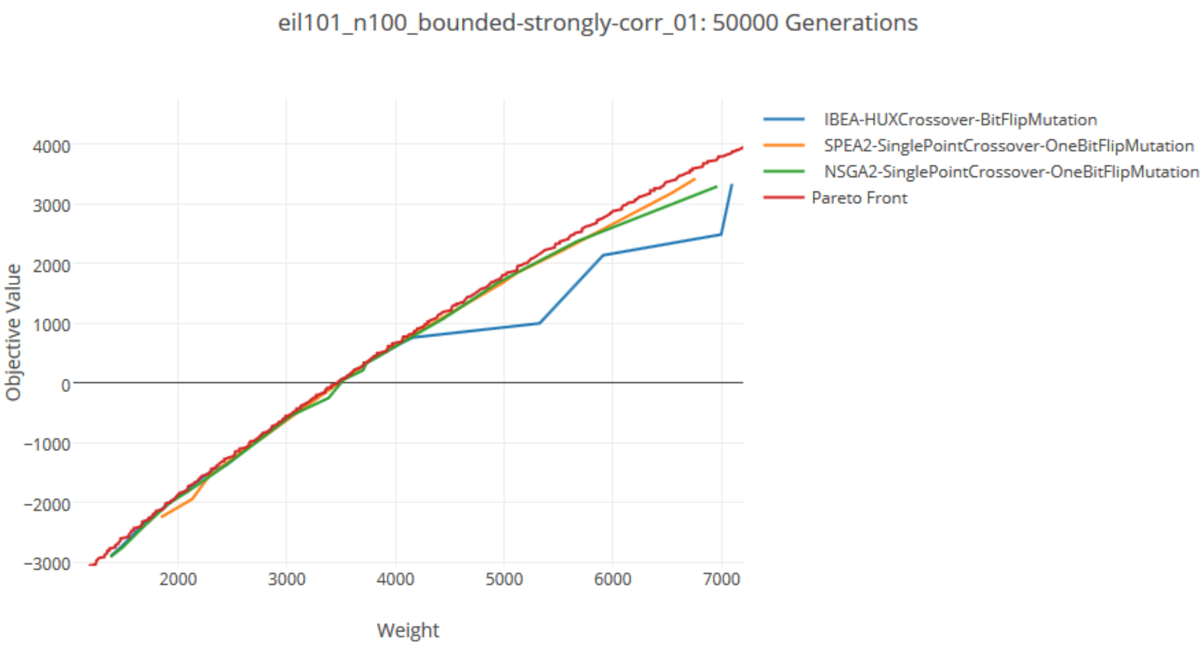

Figure 11: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n100_bounded-strongly-corr_01 instances compare to Pareto-optimal front for 50000 generations. All three algorithm combinations are close to the Pareto-optimal front, but IBEA combination has some variance after certain points.

Figure 12: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n100_bounded-strongly-corr_01 instances compare to Pareto-optimal front for 100000 generations. All three algorithm combinations are close to the Pareto-optimal front.

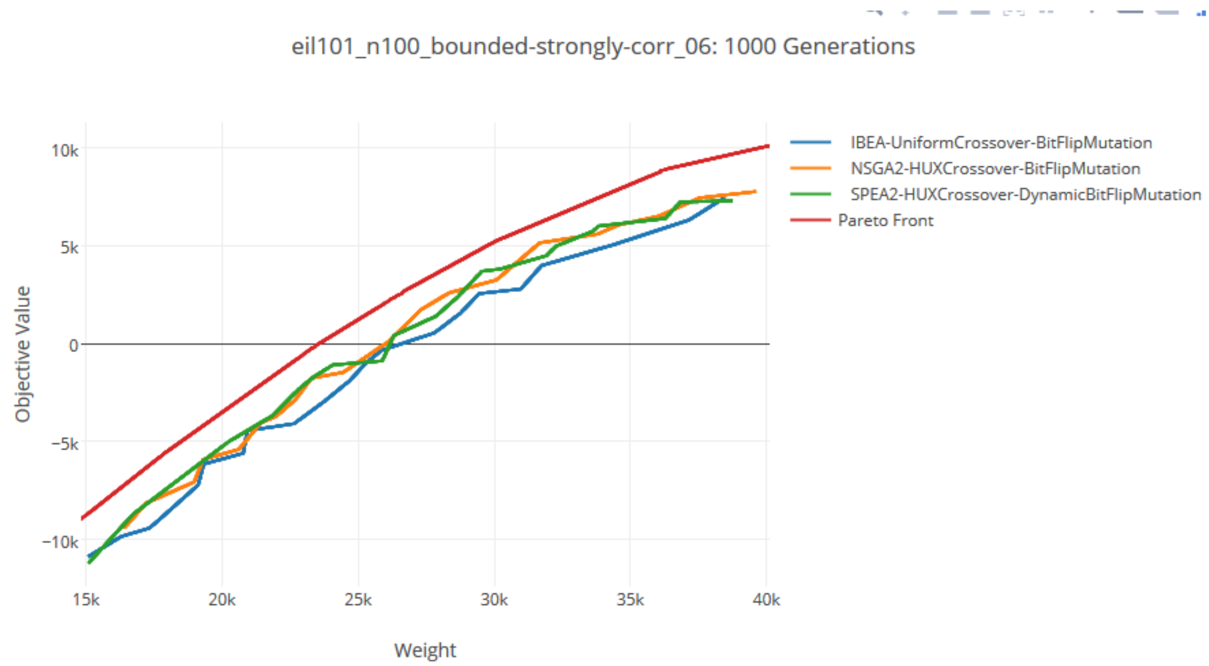

Figure 13: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n100_bounded-strongly-corr_06 instances compare to Pareto-optimal front for 1000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

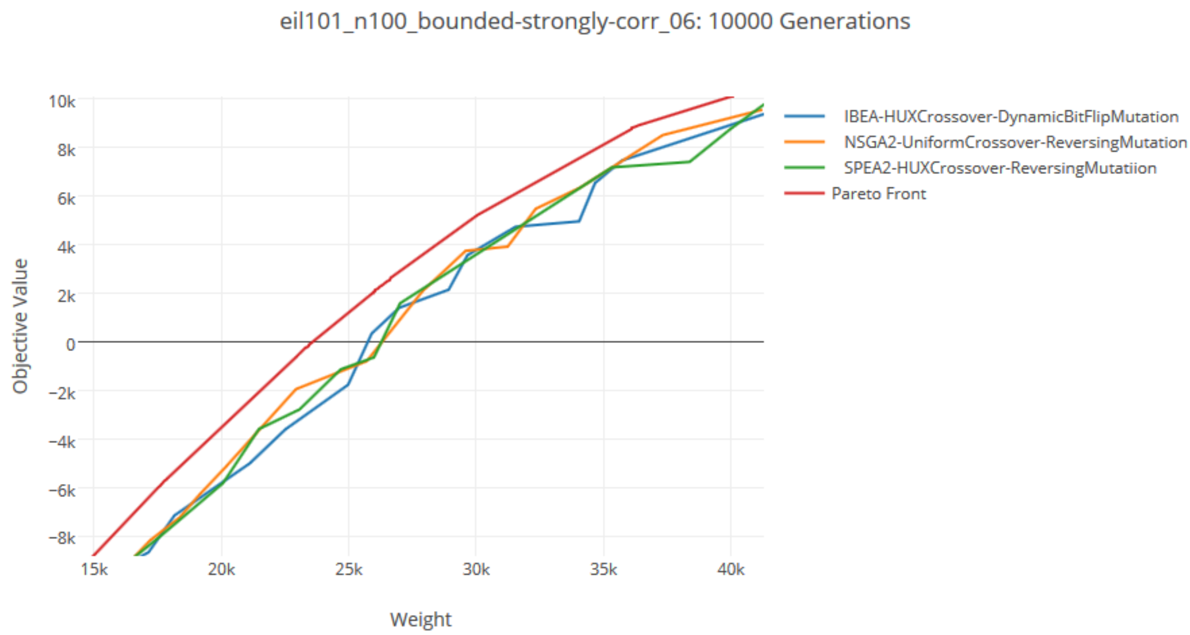

Figure 14: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n100_bounded-strongly-corr_06 instances compare to Pareto-optimal front for 10000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

Figure 15: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n100_bounded-strongly-corr_06 instances compare to Pareto-optimal front for 50000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

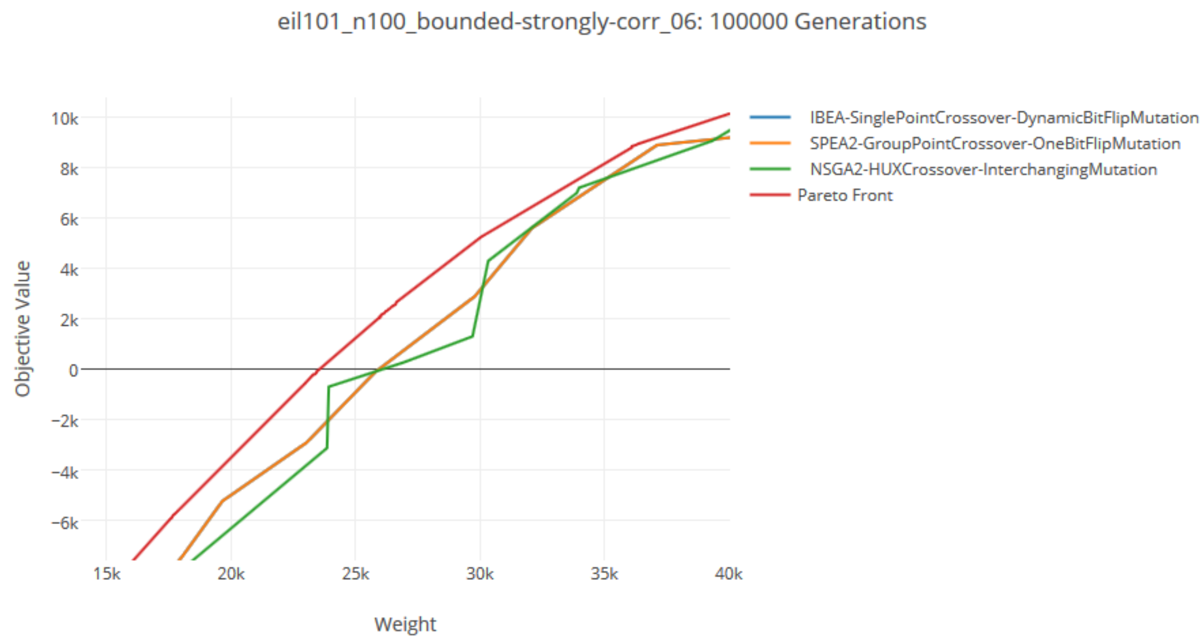

Figure 16: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n100_bounded-strongly-corr_06 instances compare to Pareto-optimal front for 100000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

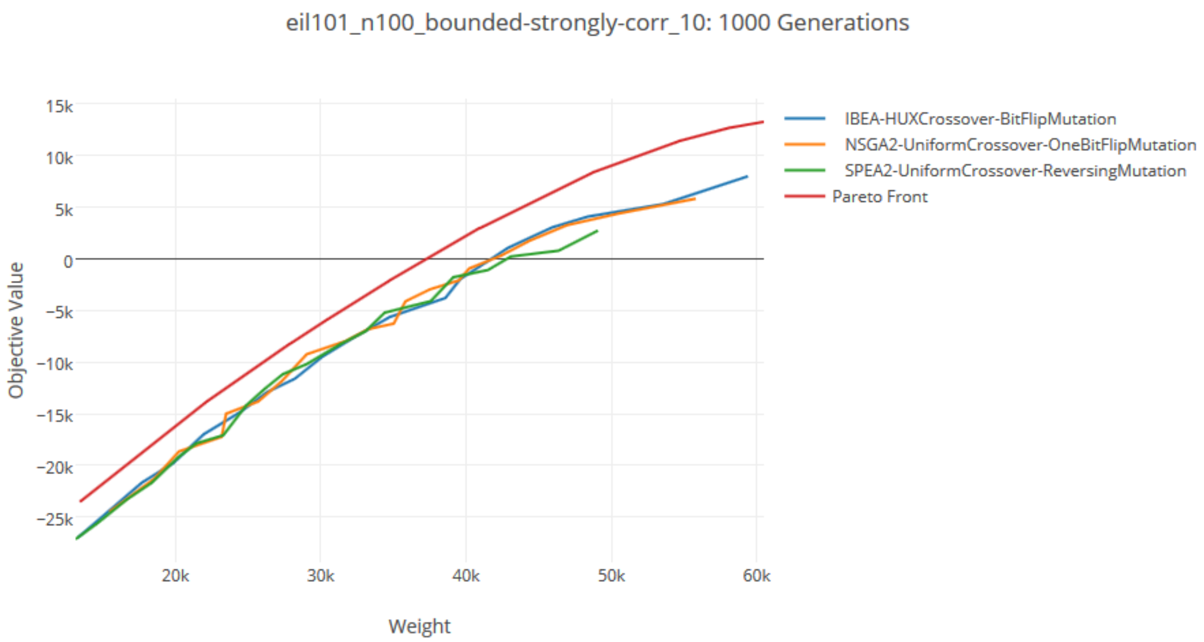

Figure 17: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n100_bounded-strongly-corr_10 instances compare to Pareto-optimal front for 1000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

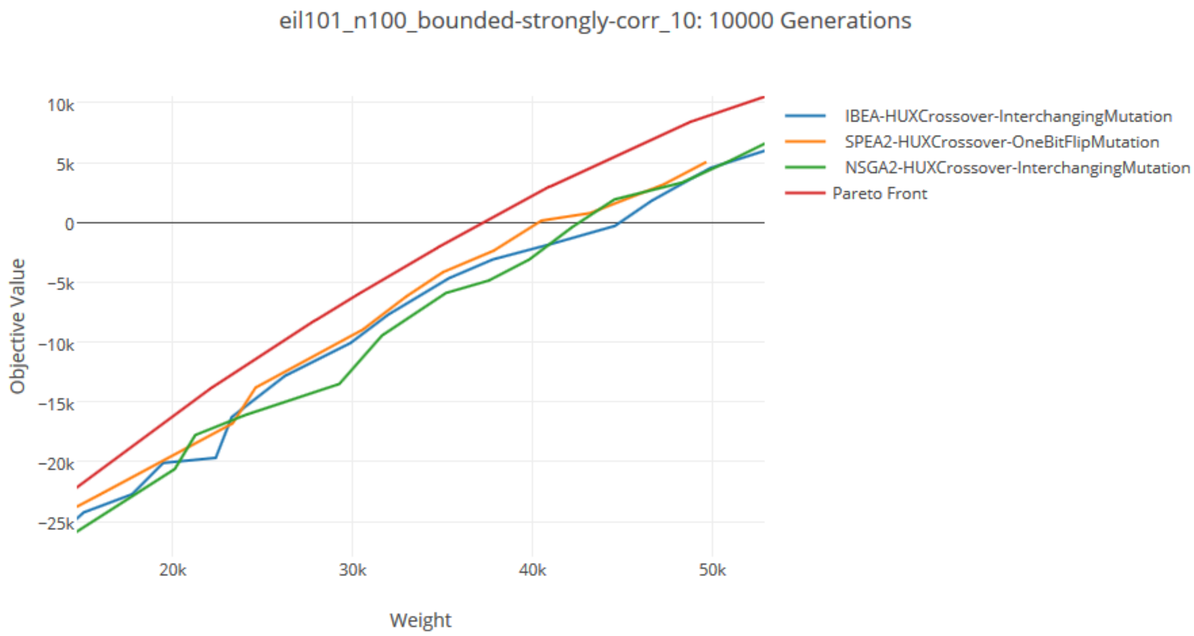

Figure 18: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n100_bounded-strongly-corr_10 instances compare to Pareto-optimal front for 10000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

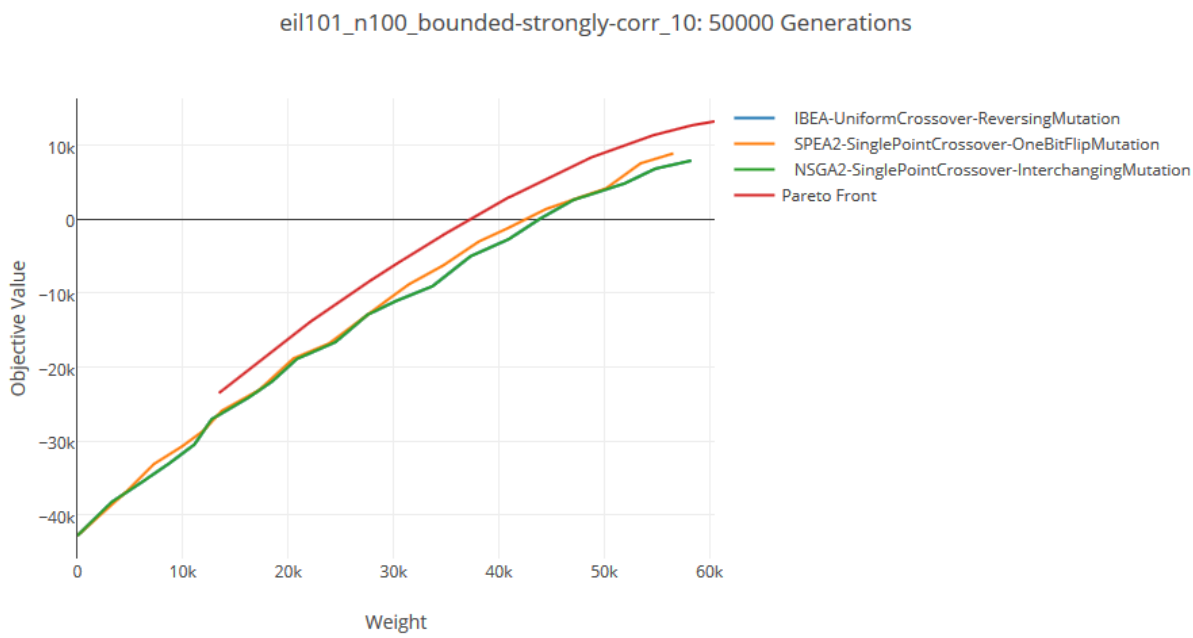

Figure 19: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n100_bounded-strongly-corr_10 instances compare to Pareto-optimal front for 50000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

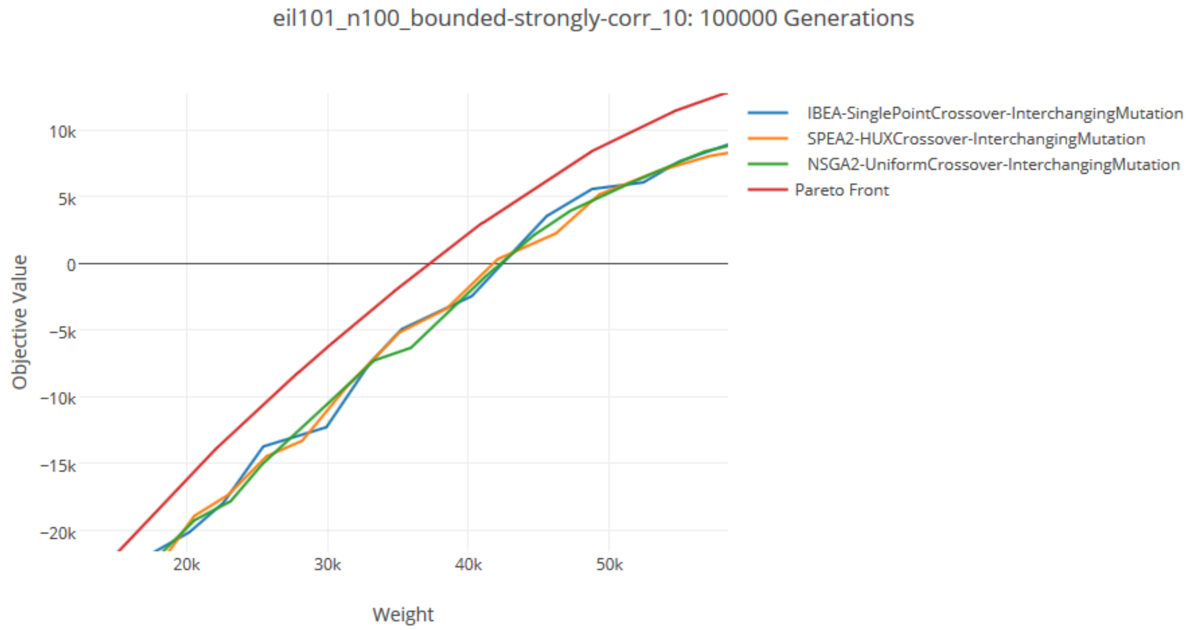

Figure 20: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n100_bounded-strongly-corr_10 instances compare to Pareto-optimal front for 100000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

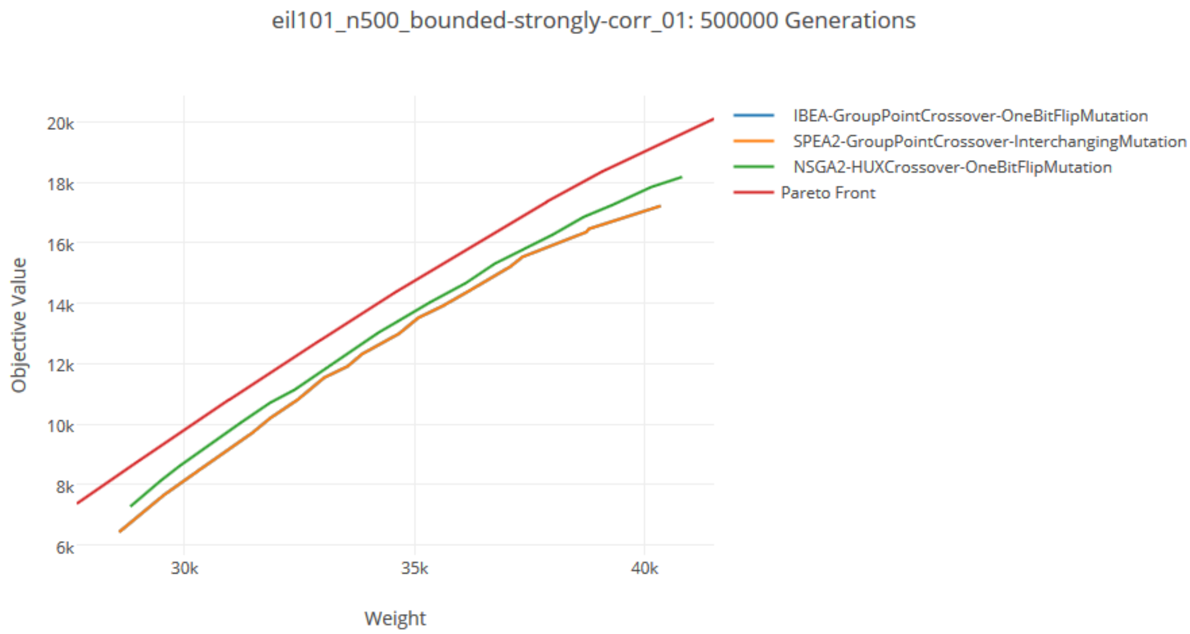

Figure 21: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n500_bounded-strongly-corr_01 instances compare to Pareto-optimal front for 50000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

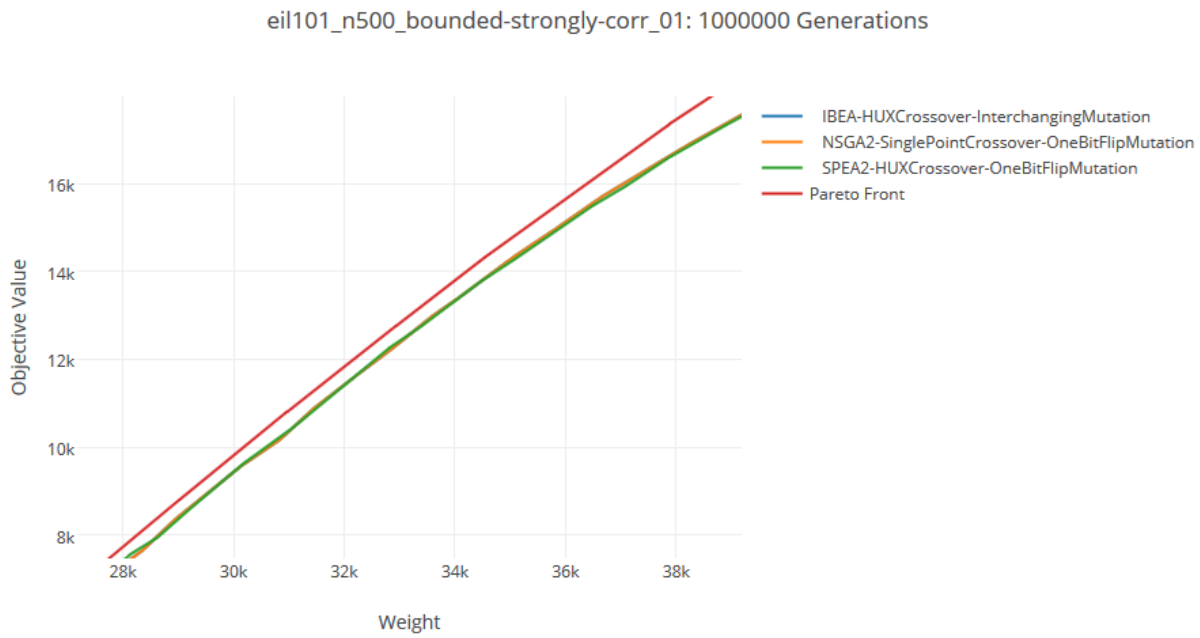

Figure 22: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n500_bounded-strongly-corr_01 instances compare to Pareto-optimal front for 100000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

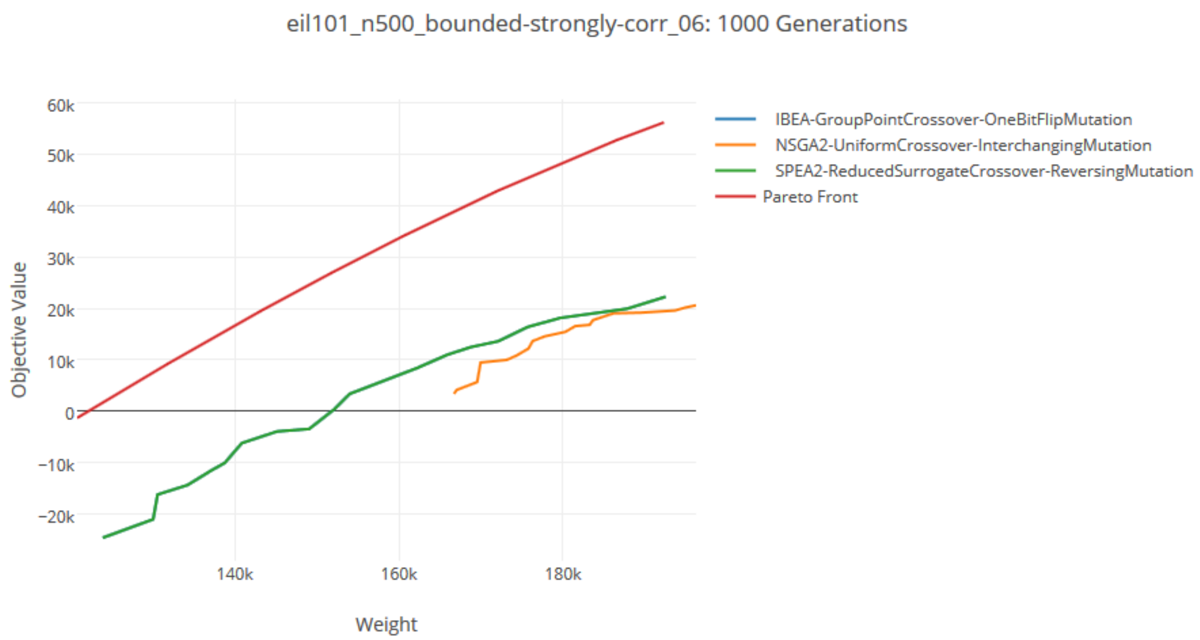

Figure 23: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n500_bounded-strongly-corr_06 instances compare to Pareto-optimal front for 1000 generations. All three algorithm combinations has a big difference between the Pareto-optimal front.

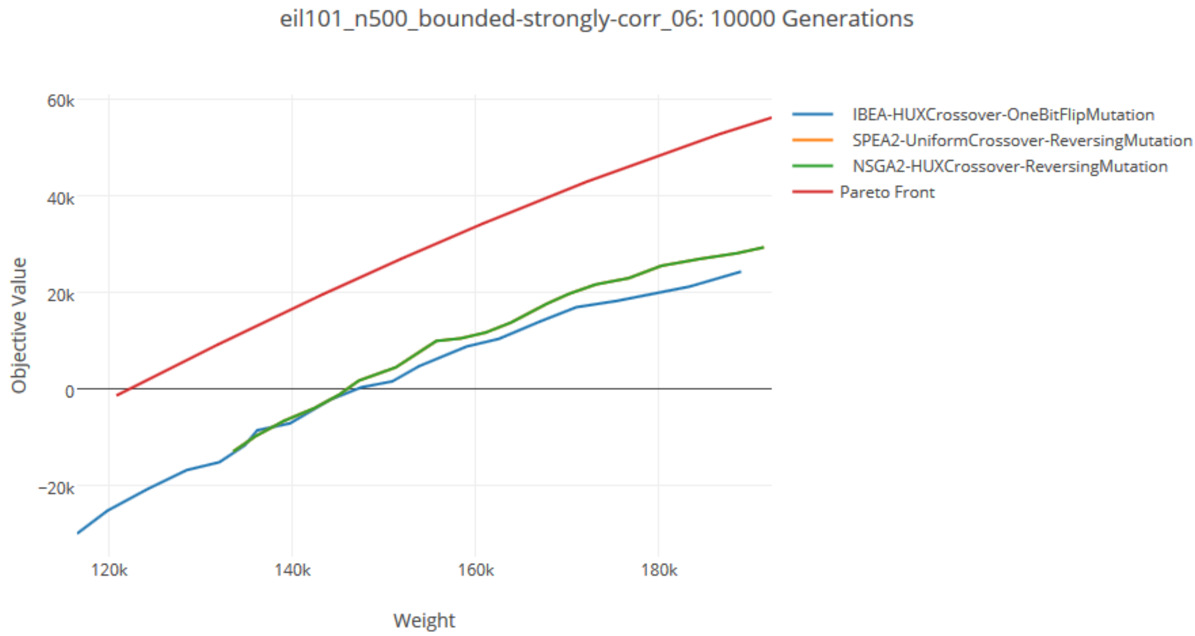

Figure 24: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n500_bounded-strongly-corr_06 instances compare to Pareto-optimal front for 10000 generations. All three algorithm combinations has a big difference between the Pareto-optimal front.

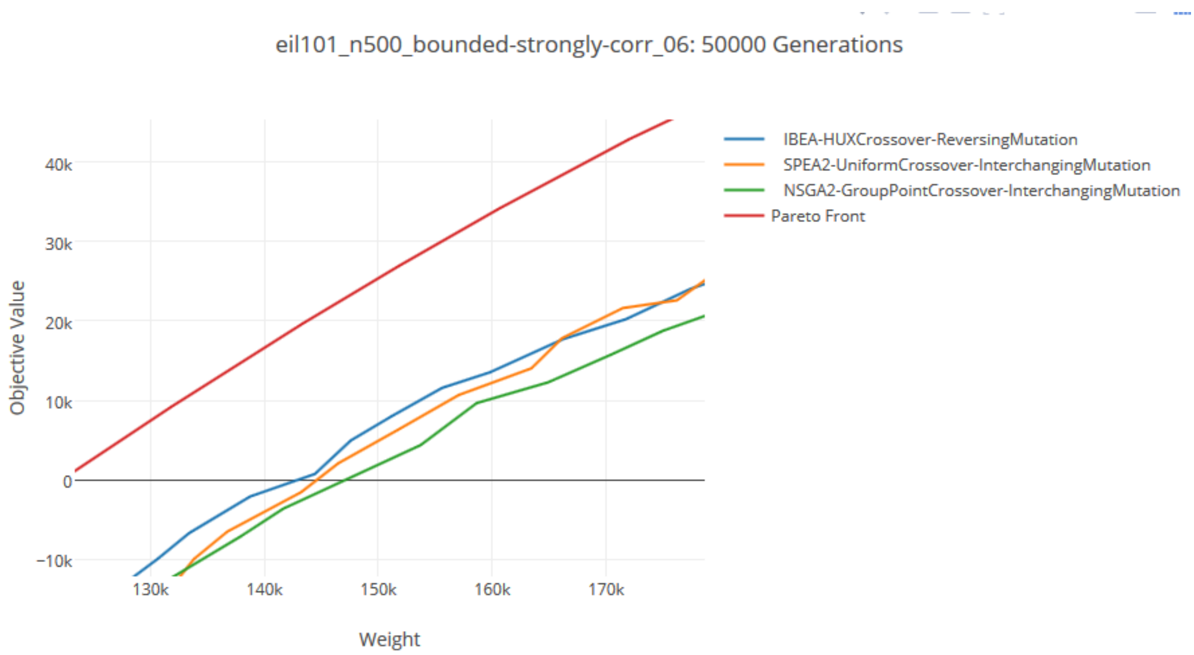

Figure 25: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n500_bounded-strongly-corr_06 instances compare to Pareto-optimal front for 50000 generations. All three algorithm combinations has a big difference between the Pareto-optimal front.

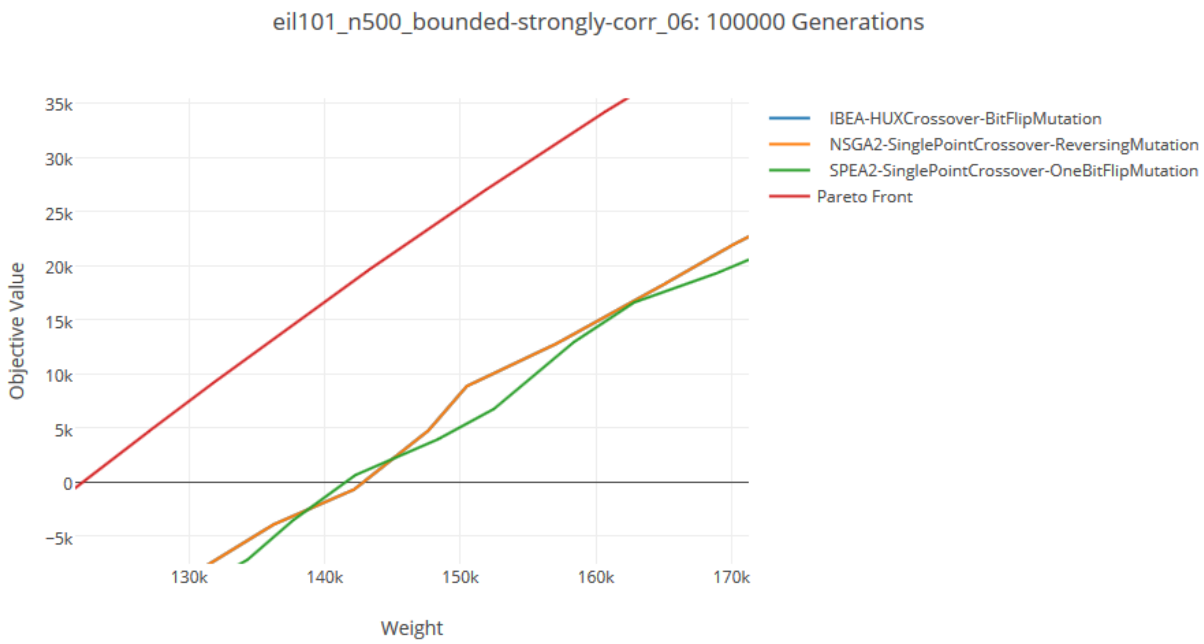

Figure 26: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n500_bounded-strongly-corr_06 instances compare to Pareto-optimal front for 100000 generations. All three algorithm combinations has a big difference between the Pareto-optimal front.

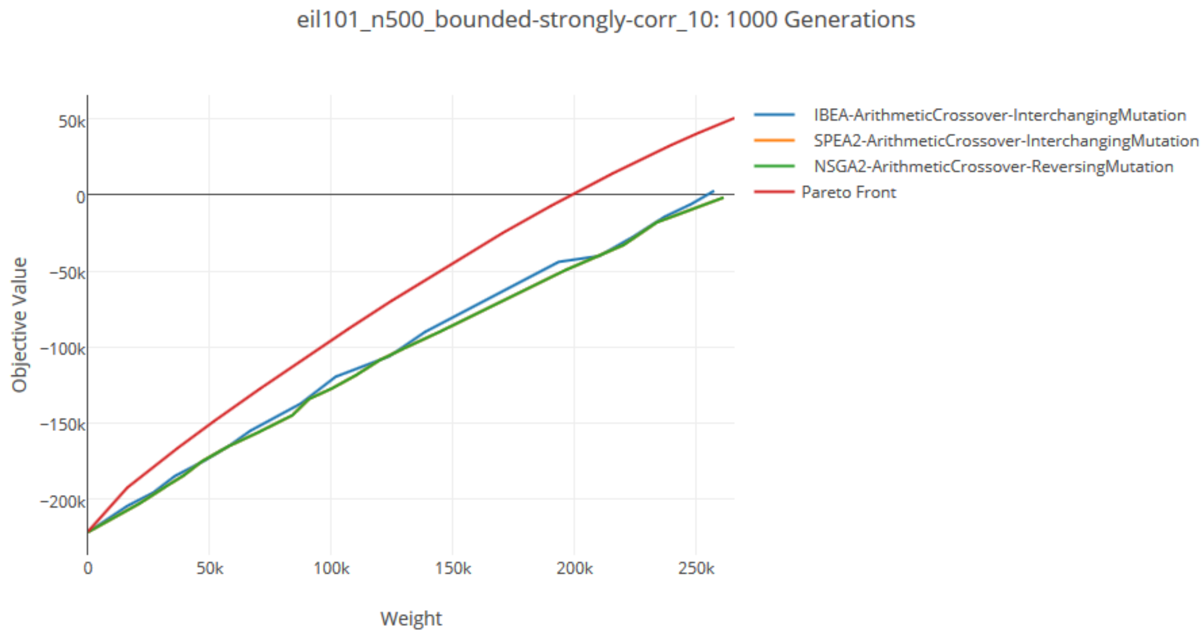

Figure 27: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n500_bounded-strongly-corr_10 instances compare to Pareto-optimal front for 1000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

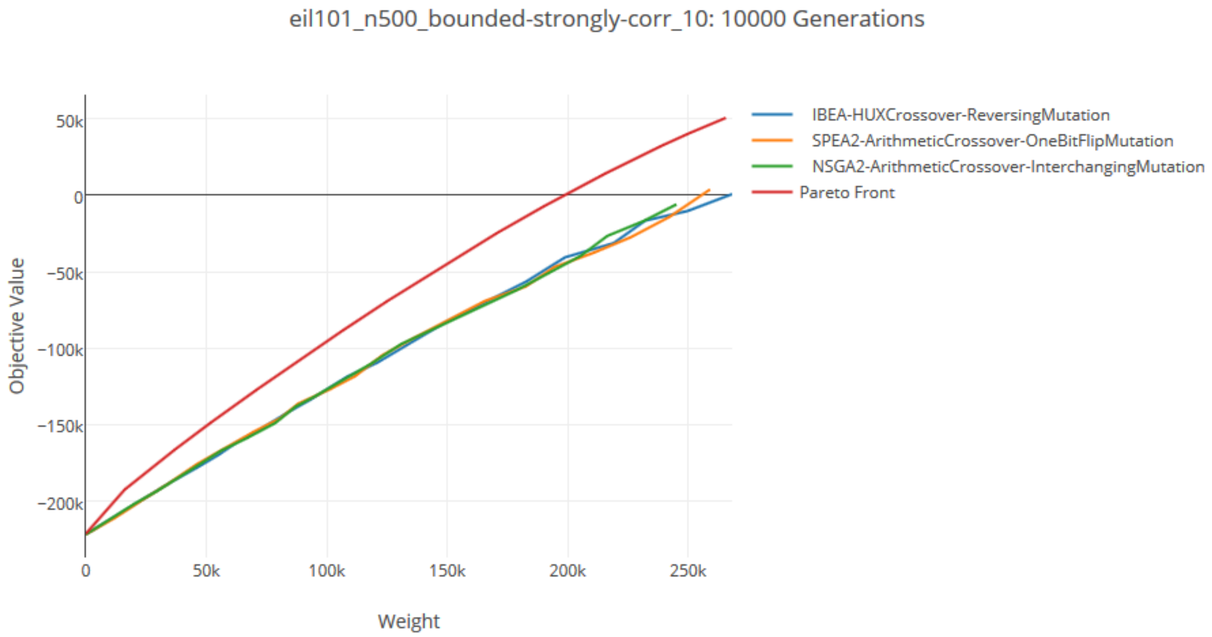

Figure 28: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n500_bounded-strongly-corr_10 instances compare to Pareto-optimal front for 10000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

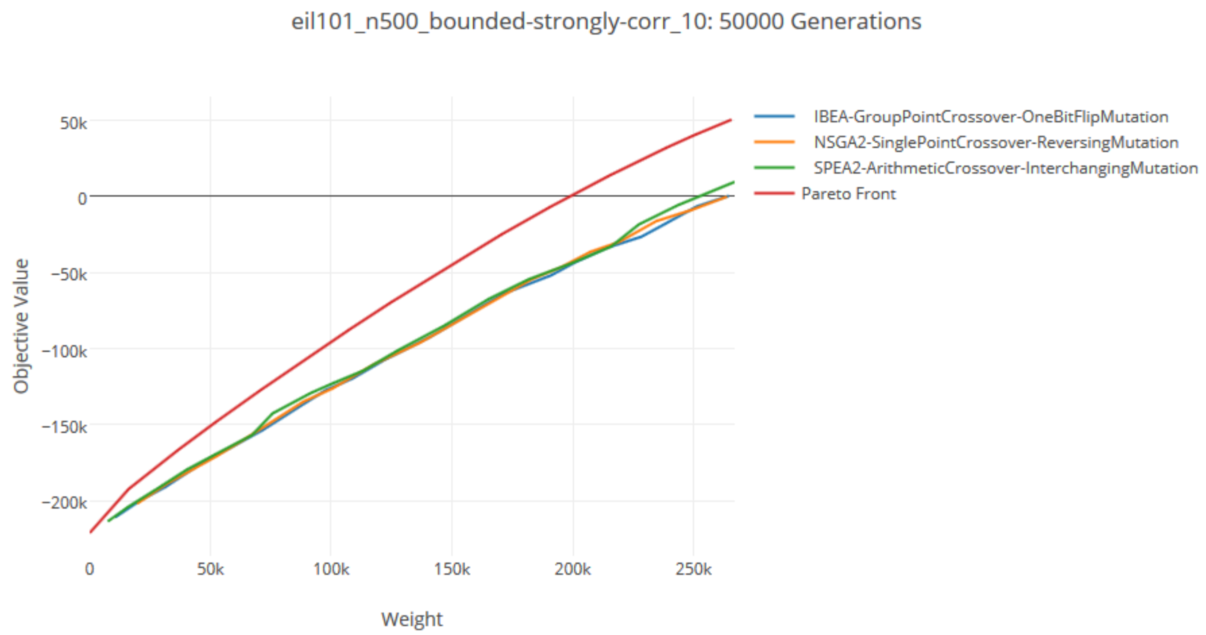

Figure 29: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n500_bounded-strongly-corr_10 instances compare to Pareto-optimal front for 50000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

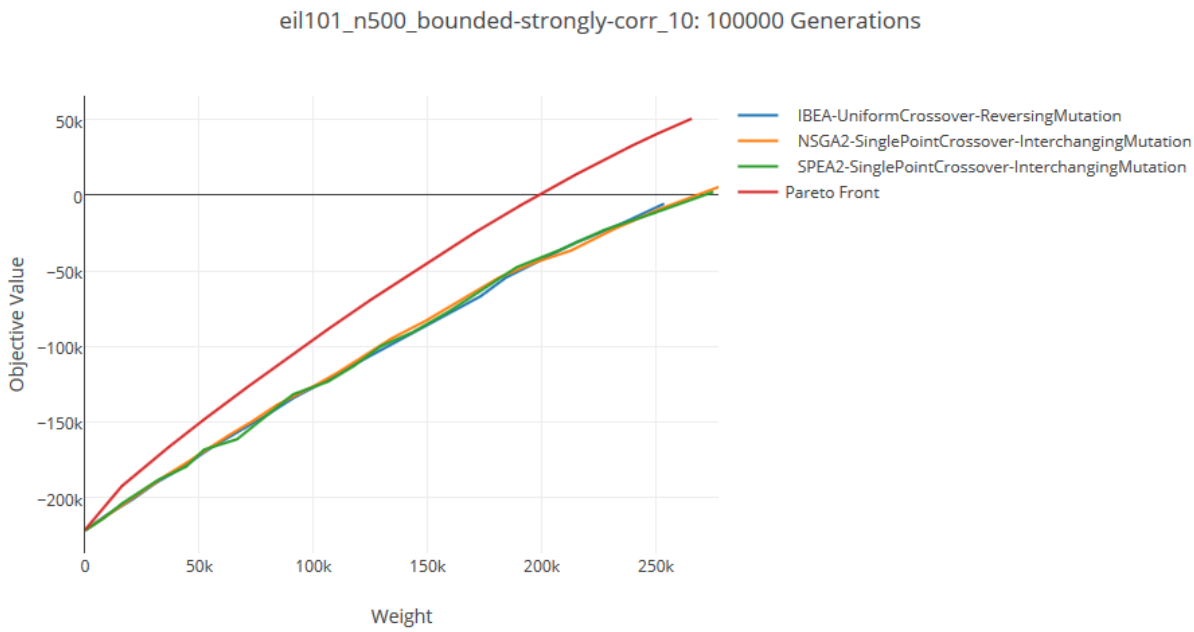

Figure 30: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n500_bounded-strongly-corr_10 instances compare to Pareto-optimal front for 100000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

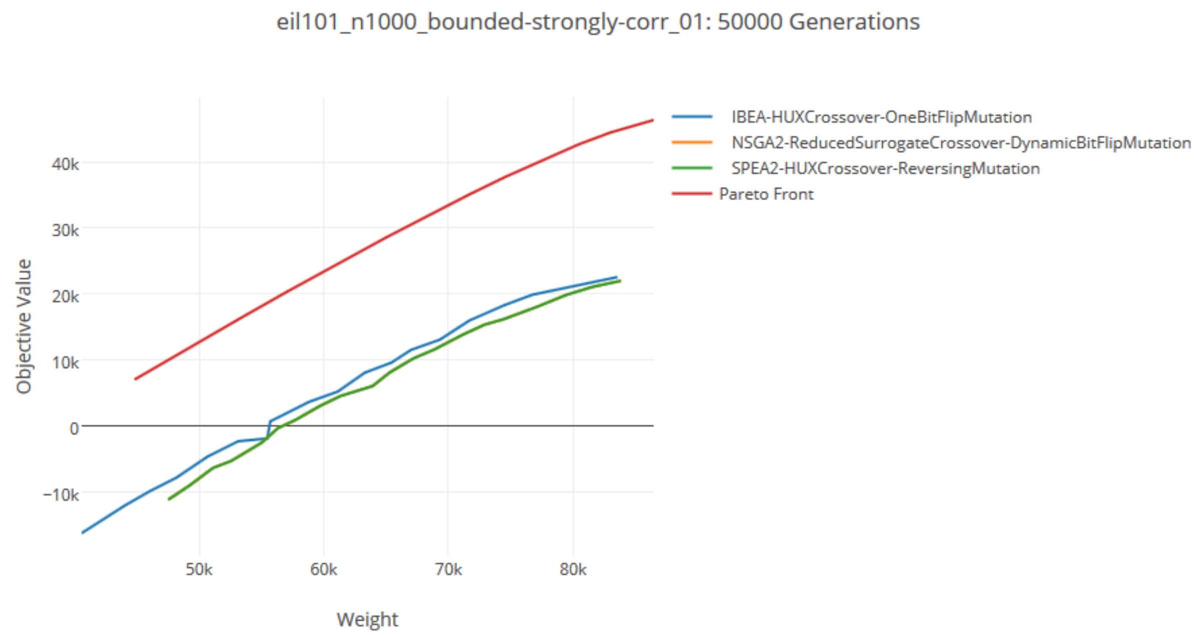

Figure 31: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n1000_bounded-strongly-corr_01 instances compare to Pareto-optimal front for 50000 generations. All three algorithm combinations has a big difference between the Pareto-optimal front.

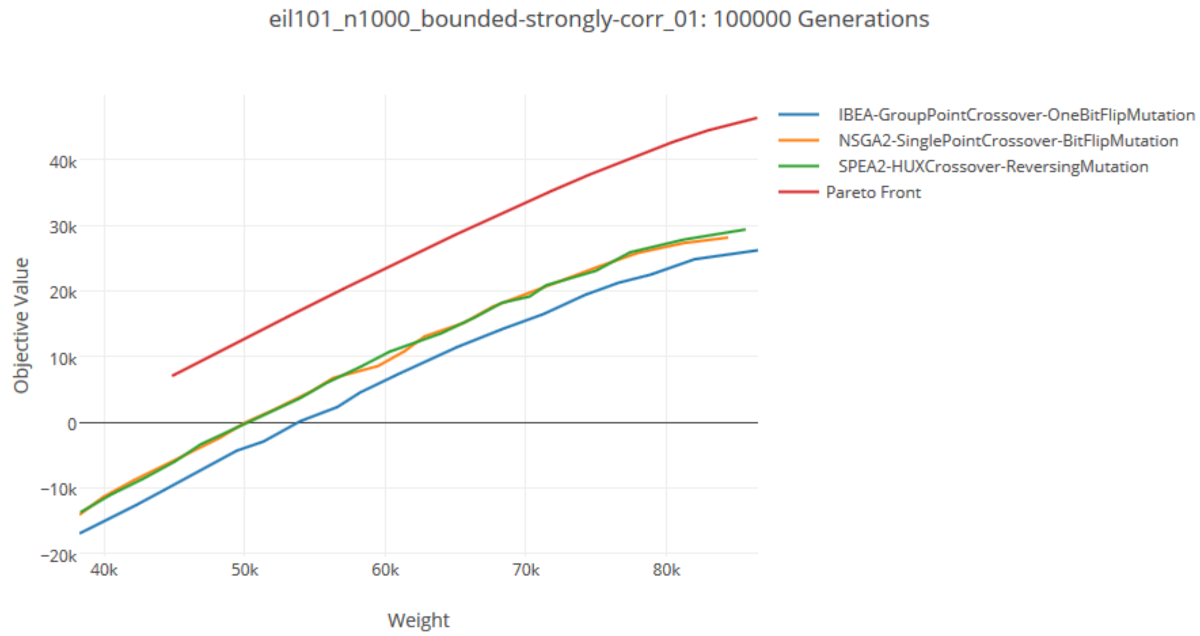

Figure 32: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n1000_bounded-strongly-corr_01 instances compare to Pareto-optimal front for 100000 generations. All three algorithm combinations has a big difference between the Pareto-optimal front.

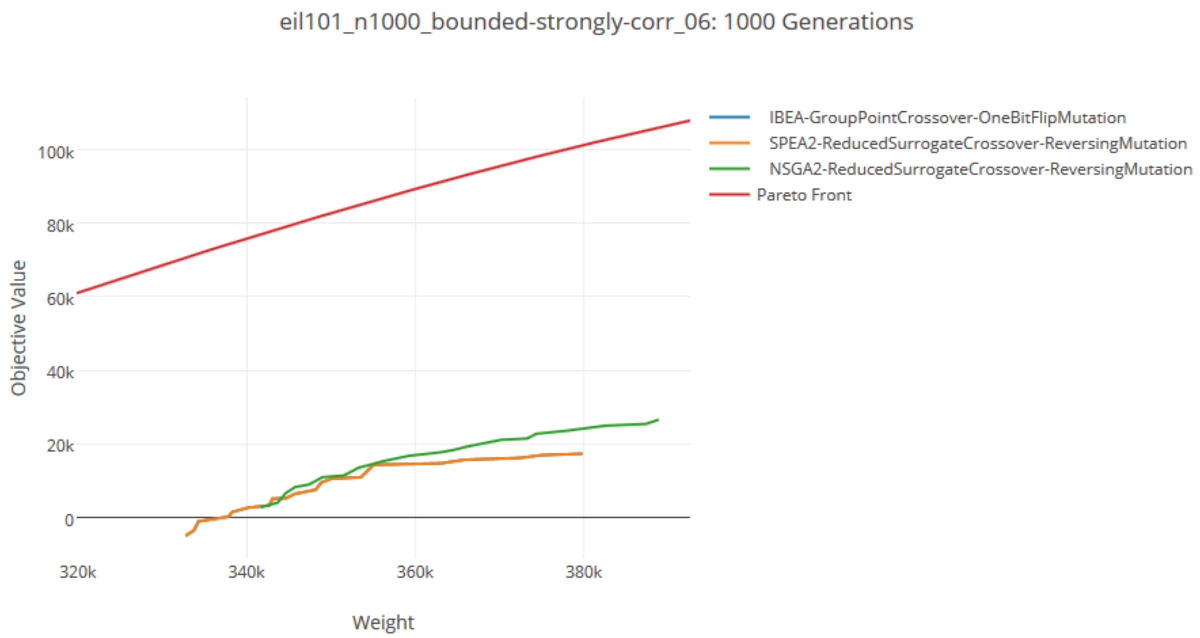

Figure 33: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n1000_bounded-strongly-corr_06 instances compare to Pareto-optimal front for 1000 generations. All three algorithm combinations has a big difference between the Pareto-optimal front.

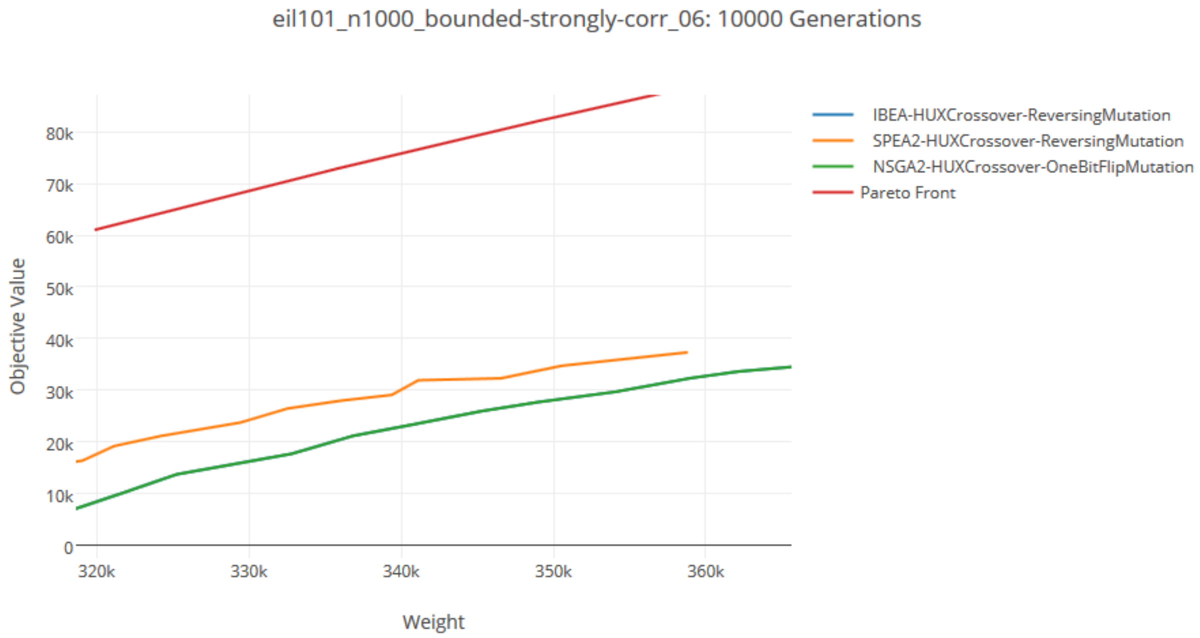

Figure 34: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n1000_bounded-strongly-corr_06 instances compare to Pareto-optimal front for 10000 generations. All three algorithm combinations has a big difference between the Pareto-optimal front.

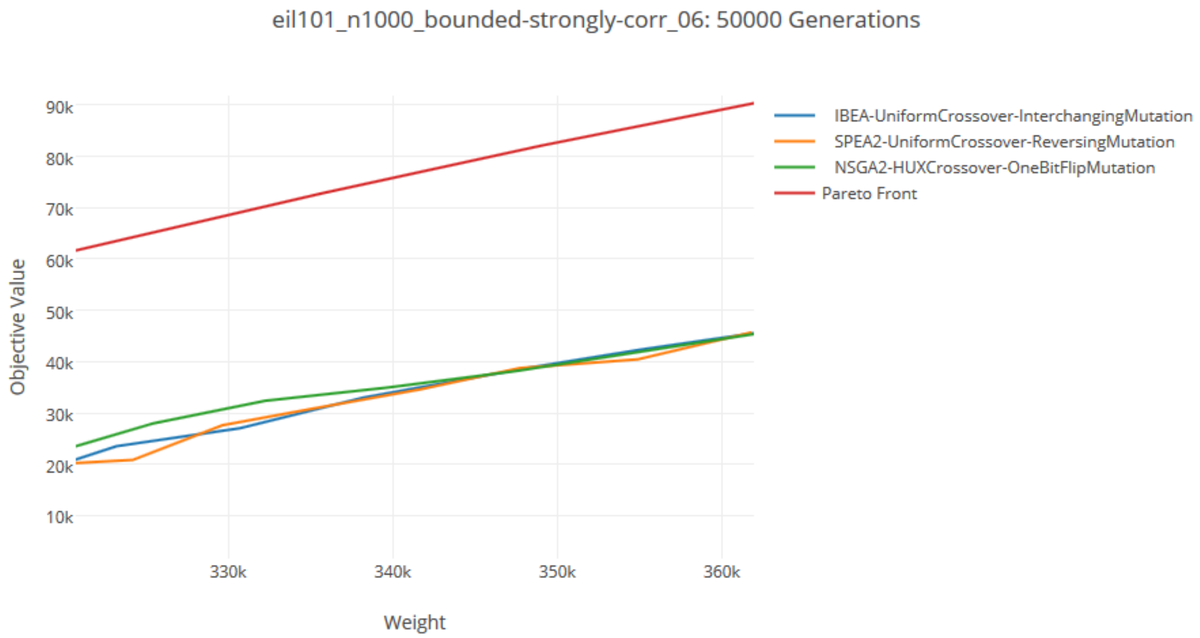

Figure 35: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n1000_bounded-strongly-corr_06 instances compare to Pareto-optimal front for 50000 generations. All three algorithm combinations has a big difference between the Pareto-optimal front.

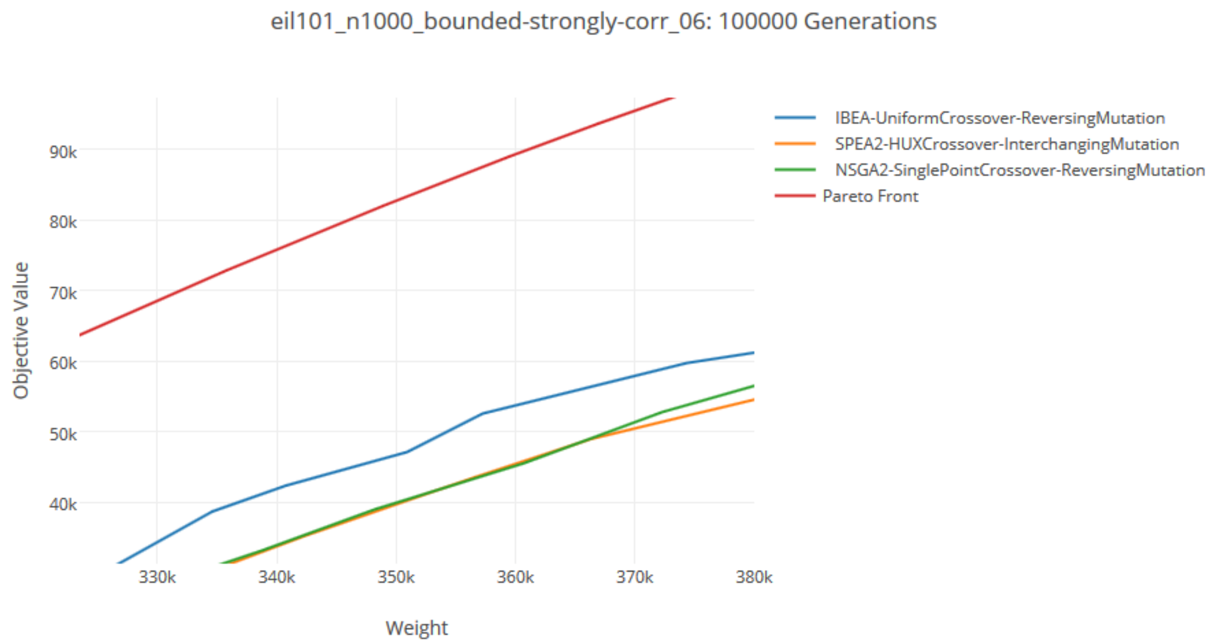

Figure 36: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n1000_bounded-strongly-corr_06 instances compare to Pareto-optimal front for 100000 generations. All three algorithm combinations has a big difference between the Pareto-optimal front.

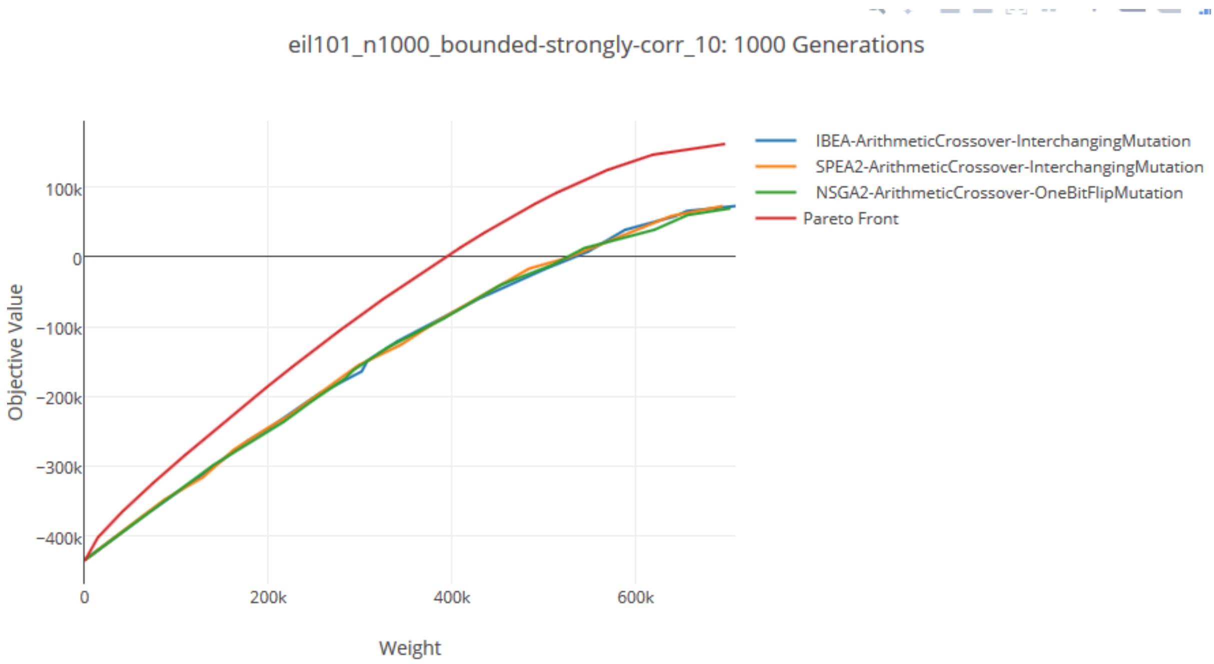

Figure 37: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n1000_bounded-strongly-corr_10 instances compare to Pareto-optimal front for 1000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

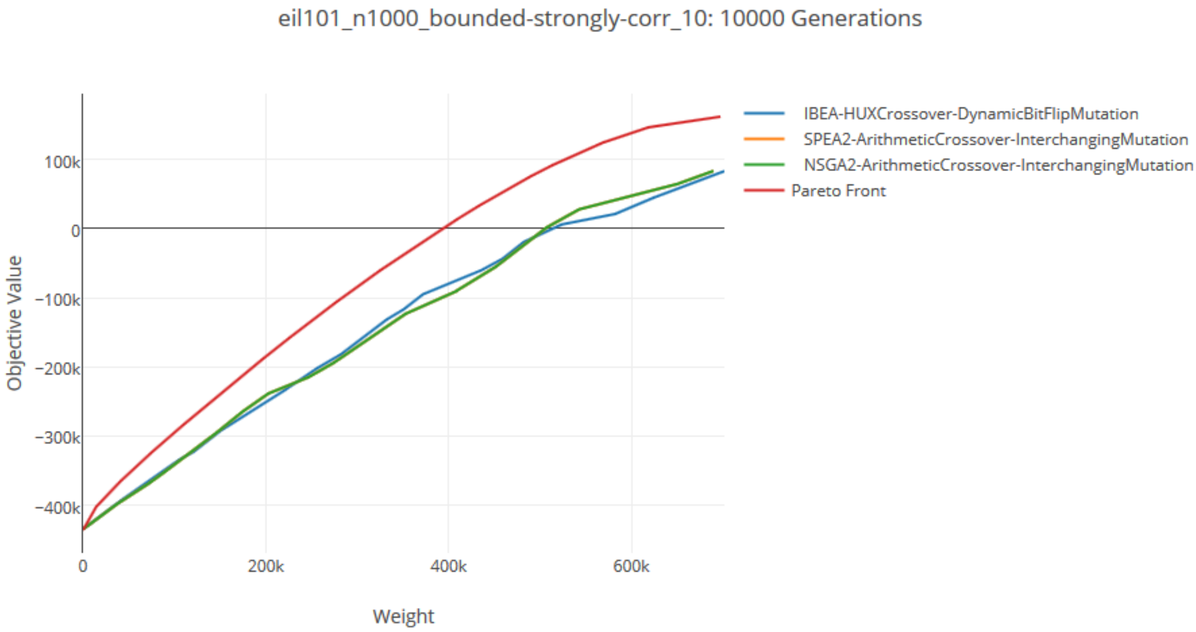

Figure 38: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n1000_bounded-strongly-corr_10 instances compare to Pareto-optimal front for 10000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

Figure 39: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n1000_bounded-strongly-corr_10 instances compare to Pareto-optimal front for 50000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

Figure 40: Performance of best NSGA2, SPEA2 and IBEA algorithm operator combination on benchmark 1 and benchmark 2 of eil101_n1000_bounded-strongly-corr_10 instances compare to Pareto-optimal front for 100000 generations. All three algorithm combinations has a difference between the Pareto-optimal front.

Table 3: Best Combinations Per Algorithm Per instance for 50000 generations

Instance	Algorithm	Crossover Operator	Mutation Operator	Run	Error Percentage
Il101-n100-bounded-strongly-corr-01-50000-generations-ErrorReport	IBEA	ReducedSurrogateCrossover	BitFlipMutation	RunNum=4	0.005
	SPEA2	UniformCrossover	GreedyMutation	RunNum=8	5.29
	NSGA2	HUXCrossover	OneBitFlipMutation	RunNum=3	6.47
Il101-n100-bounded-strongly-corr-06-50000-generations-ErrorReport	IBEA	HUXCrossover	BitFlipMutation	RunNum=3	0.003
	NSGA2	UniformCrossover	BitFlipMutation	RunNum=1	17.86
	SPEA2	HUXCrossover	BitFlipMutation	RunNum=1	19.22
Il101-n100-bounded-strongly-corr-10-50000-generations-ErrorReport	IBEA	UniformCrossover	BitFlipMutation	RunNum=4	0.02
	NSGA2	SinglePointCrossover	GreedyMutation	RunNum=10	19.03
	SPEA2	HUXCrossover	BitFlipMutation	RunNum=6	24.31
Il101-n500-bounded-strongly-corr-01-50000-generations-ErrorReport	IBEA	UniformCrossover	OneBitFlipMutation	RunNum=4	0.29
	NSGA2	HUXCrossover	BitFlipMutation	RunNum=4	47.82
	SPEA2	SinglePointCrossover	BitFlipMutation	RunNum=9	70.82
Il101-n500-bounded-strongly-corr-06-50000-generations-ErrorReport	IBEA	HUXCrossover	OneBitFlipMutation	RunNum=9	0.64
	NSGA2	SinglePointCrossover	BitFlipMutation	RunNum=10	53.35
	SPEA2	ArithmeticCrossover	DynamicBitFlipMutation	RunNum=9	68.37
Il101-n500-bounded-strongly-corr-10-50000-generations-ErrorReport	IBEA	HUXCrossover	BitFlipMutation	RunNum=6	2.81
	NSGA2	ArithmeticCrossover	BitFlipMutation	RunNum=6	42.07
	SPEA2	SinglePointCrossover	BitFlipMutation	RunNum=10	65.06
Il101-n1000-bounded-strongly-corr-01-50000-generations-ErrorReport	IBEA	UniformCrossover	OneBitFlipMutation	RunNum=8	4.53
	NSGA2	GroupPointCrossover	DynamicBitFlipMutation	RunNum=9	40.93
	SPEA2	HUXCrossover	BitFlipMutation	RunNum=10	60.56
Il101-n1000-bounded-strongly-corr-06-50000-generations-ErrorReport	IBEA	UniformCrossover	ReversingMutation	RunNum=10	6.14
	SPEA2	ArithmeticCrossover	InterchangingMutation	RunNum=2	61.81
	NSGA2	ArithmeticCrossover	BitFlipMutation	RunNum=8	73.94
Il101-n1000-bounded-strongly-corr-10-50000-generations-ErrorReport	IBEA	HUXCrossover	BitFlipMutation	RunNum=5	12.96
	SPEA2	ArithmeticCrossover	ReversingMutation	RunNum=7	71.88
	NSGA2	SinglePointCrossover	OneBitFlipMutation	RunNum=6	77.9

Table 4: Best Combinations Per Algorithms Per instance for 100000 generations

Instance	Algorithm	Crossover Operator	Mutation Operator	Run	Error Percentage
Il101-n100-bounded-strongly-corr-01-100000-generations-ErrorReport	IBEA	UniformCrossover	OneBitFlipMutation	RunNum=10	0.02
	SPEA2	HUXCrossover	GreedyMutation	RunNum=4	0.46
	NSGA2	HUXCrossover	BitFlipMutation	RunNum=7	8.24
Il101-n100-bounded-strongly-corr-06-100000-generations-ErrorReport	IBEA	GroupPointCrossover	ReversingMutation	RunNum=9	0.006
	SPEA2	GroupPointCrossover	BitFlipMutation	RunNum=10	16.04
	NSGA2	UniformCrossover	BitFlipMutation	RunNum=2	17.86
Il101-n100-bounded-strongly-corr-10-100000-generations-ErrorReport	IBEA	UniformCrossover	InterchangingMutation	RunNum=5	0.02
	NSGA2	SinglePointCrossover	BitFlipMutation	RunNum=2	17.48
	SPEA2	SinglePointCrossover	BitFlipMutation	RunNum=8	21.57
Il101-n500-bounded-strongly-corr-01-100000-generations-ErrorReport	IBEA	GroupPointCrossover	OneBitFlipMutation	RunNum=9	0.008
	NSGA2	UniformCrossover	GreedyMutation	RunNum=2	34.8
	SPEA2	UniformCrossover	GreedyMutation	RunNum=2	44.82
Il101-n500-bounded-strongly-corr-06-100000-generations-ErrorReport	IBEA	HUXCrossover	OneBitFlipMutation	RunNum=9	0.05
	NSGA2	SinglePointCrossover	ReversingMutation	RunNum=6	41.46
	SPEA2	SinglePointCrossover	ReversingMutation	RunNum=9	61.08
Il101-n500-bounded-strongly-corr-10-100000-generations-ErrorReport	IBEA	UniformCrossover	BitFlipMutation	RunNum=2	0.86
	NSGA2	SinglePointCrossover	ReversingMutation	RunNum=3	33.83
	SPEA2	SinglePointCrossover	DynamicBitFlipMutation	RunNum=1	54.85
Il101-n1000-bounded-strongly-corr-01-100000-generations-ErrorReport	IBEA	HUXCrossover	OneBitFlipMutation	RunNum=9	0.01
	NSGA2	GroupPointCrossover	DynamicBitFlipMutation	RunNum=5	20.16
	SPEA2	GroupPointCrossover	DynamicBitFlipMutation	RunNum=8	54.19
Il101-n1000-bounded-strongly-corr-06-100000-generations-ErrorReport	IBEA	UniformCrossover	OneBitFlipMutation	RunNum=10	1.73
	NSGA2	SinglePointCrossover	OneBitFlipMutation	RunNum=3	61.54
	SPEA2	ArithmeticCrossover	BitFlipMutation	RunNum=1	72.19
Il101-n1000-bounded-strongly-corr-10-100000-generations-ErrorReport	IBEA	UniformCrossover	BitFlipMutation	RunNum=9	9.01
	NSGA2	SinglePointCrossover	ReversingMutation	RunNum=5	48.79
	SPEA2	ArithmeticCrossover	OneBitFlipMutation	RunNum=4	55.14