

Evolutionary Approach to the Quadratic Knapsack Problem

Contributors

- 1. Nakul Upadhya
- 2. Maxim Velli
- 3. Joaquin Cervantes

Contents

1	Abstract	1
2	Introduction	1
3	Evolutionary Algorithm Overview	2
3.1	Solution Representation	2
3.2	Individual Generation	2
3.3	Fitness Function	3
3.4	Crossover Function	3
3.5	Mutation Function	3
3.6	Selection Function	3
3.7	Evolutionary Strategy	3
4	Landscape Analysis	4
5	Experimental Design	5
6	Results	6
6.1	Hypervolume analysis	6
6.2	Total Profit Analysis	9
6.3	Total Weight Analysis	11
6.4	Minimum Profit Analysis	13
6.5	Analysis of the Fronts	14
7	Conclusion	15
8	References	16

1 Abstract

Given a set $N = \{1, \dots, n\}$ of items that each has an associated profit p_i , weight w_i , and an associated joint profit p_{ij} for items i and j and $i \neq j$. In addition, $M = \{1, \dots, m\}$ number of knapsacks was provided, each with a capacity constraint C_k for $k = 1, \dots, m$. The goal is to choose a set from N and place these items into M knapsacks such that the overall profit of the objects is maximized, the total weight of items is minimized, and maximize the least profitable knapsack. In this paper, we discuss the methodology of utilizing an evolutionary algorithm to find Pareto-optimal solutions while varying the parameters for different problem sets. An evolutionary algorithm is used to converge to better solutions by evaluating the individual solution's fitness function, then selecting, crossing, and mutating the population of solutions (individuals) to obtain a Pareto-optimal set of solutions. To assess the outcomes of our evolutionary algorithm, different statistical analyses were used. Hypervolume, total profit, total weight, minimum profit obtained for the datasets, as well as Pareto fronts, were all analyzed using different statistical methods. After conducting these analyses, it was found that a population size of 200-250 and a mutation rate of 0.1 yielded the best results when maximizing total profit or maximizing the minimum profit in a given knapsack was prioritized. If minimizing total weight is given preference, the mutation rate should be between .4 and .6 with a population size of 250.

2 Introduction

Consider a problem where n items need to be placed in m knapsacks with different capacities c_k where $k = 1, \dots, m$. Each item i has a profit p_i , weight w_i and a joint profit p_{ij} where i and j are two different items in n . The objective is to select from N and place chosen items into one of the M knapsacks, such that simultaneously we maximize the overall profit, minimize the total weight of items in all knapsacks, and maximize the value of the least profitable knapsack. The objective functions were defined as:

$$\begin{aligned} &\text{maximize } \sum_{k \in M} \sum_{i \in S} p_i + \sum_{k \in M} \sum_{i, j, j \neq i} p_{ij} \\ &\text{minimize } \sum_{k \in M} \sum_{i \in S} w_i \\ &\text{maximize } \min_{k \in M} \left(\sum_{i \in S} p_i + \sum_{i, j, j \neq i} p_{ij} \right) \end{aligned}$$

Figure 1: Objective Functions

The Quadratic Knapsack Problem (QKP) is an NP-hard problem. One of the ways to approach solving such a problem is through Nature Inspired Algorithms, such as Evolutionary Algorithms or Swarm Intelligence algorithms. Genetic algorithms (GAs) are stochastic search algorithm based on the evolution concept, and that is the route we chose over a Swarm Intelligence algorithm. GA uses a heuristic approach to provide good approximate solutions to problems that cannot be solved easily. Because of the random nature of evolutionary algorithms, they are extremely useful to find good solutions close to optimal when problems are too computationally-intensive to find an exact solution.

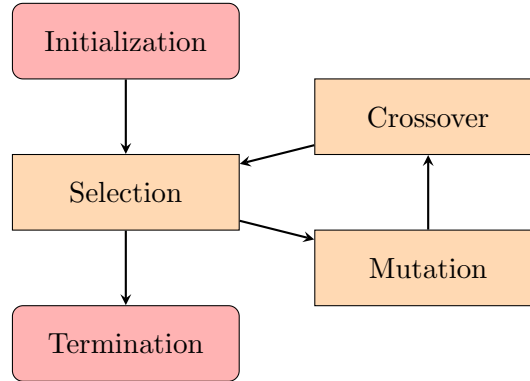
Different analyses were conducted to find optimal parameters for different problem data sets. The first analysis discussed in this paper is landscape analysis. Landscape analysis was performed by evaluating the theoretic entropy of the fitness landscape. Landscape analysis is performed to evaluate the degree to which the fitness landscape contains various rugged, neutral, and smooth shapes. ANOVA tests were conducted to analyze hypervolume, total profit, total weight and minimum profit. The same three parameters: population

size, mutation rate, and interaction of population size + mutation rate, were utilized for every ANOVA test. The ANOVA tests showed if there were significant correlations between the parameters and the magnitude of the hypervolume, total profit, total weight, and minimum profit, respectively. Lastly, Pareto front analysis was conducted. The Pareto fronts for the three objective functions were plotted and compared to get a better feeling for the quality of the multi-objective optimization visually.

Applications of the quadratic knapsack problem include loading items into different ships with certain capacities. The multiple knapsack problem was proposed to be used for deciding how to load m liquids into n tanks, when liquids may not be mixed (Martello, 1990). As a matter of fact, the cutting-stock problem is an NP-hard problem that is reducible to the knapsack problem. When the stocks have different lengths, quadratic knapsack problem is a generalization of the cutting stock problem. The quadratic knapsack problem has several applications in the paper and steel industry.

3 Evolutionary Algorithm Overview

At its core, an evolutionary algorithm performs the following routines at every generation, until the preset number of generations is achieved:



Every subroutine can be adjusted and fine-tuned to fit the specifics of a particular problem. In the following subsections, the choices made for every subroutine are discussed in regards to the variation of the knapsack problem that is being solved.

3.1 Solution Representation

A vector representation was chosen for each individual solution. Initially the use of binary decision variables x_{ij} were discussed, where $x_{ij} = 1$ if item i is in knapsack j and $= 0$ if not. Using binary decision variables would complicate the mutation and crossover functions, to avoid this, an individual integer representation where each individual was a vector containing the location of each item was used. For example, take the individual $x = [0,1,3,2,1]$. The first item on the list is in knapsack 0, in other words, it was not chosen to be part of the solution. Item two was selected to be in knapsack 1, item three in knapsack 3 and so on. Using this kind of integer representation simplified the design of crossover and mutation functions.

3.2 Individual Generation

As explained in the previous section, a vector representation was chosen to represent the individual. In this case, an individual was generated by creating a vector of a length equal to the number of items (varies per data set) and randomly assigning a number in $[0; m]$ to each vector item, where m is the number of knapsacks.

3.3 Fitness Function

In order to achieve convergence towards better solutions with a genetic algorithm, evaluation of a fitness of an individual is necessary. As described in the introduction section, the three fitness functions were kept separate. The calculation of numerical values for fitness was performed exactly as stated in the mathematical model, without any adjustments.

3.4 Crossover Function

The concept of the used crossover function, that we called “Favorite Child Crossover”, was based on creating a better and a worse individual as a result of the crossover. That was achieved by evaluating which of the two parents had higher a fitness value for objective function f_1 and assigning a higher probability for the “favorite” child to inherit parts of the chromosome of the more fit parent. The opposite methodology applies to the “least-liked” child and less fit parent.

This way of deriving separate lineages for more and less fit individuals has shown to improve the convergence of the individuals towards better solutions and enhance the exploitation/exploration balance on the solution landscape (Results not shown). The choice to use the first objective function to evaluate the probability was deliberate. As mentioned in the mathematical model section, it was possible to combine the objective functions f_1 and f_3 into one, since we hypothesized that these two would be correlated. Thus, choosing for f_1 as an indicator of a more fit individual was expected to yield a better convergence for both f_1 and f_3 and this was preferred over f_2 , which is the trade off objective function in the case of this problem.

3.5 Mutation Function

The used mutation function was essentially assigning a random item to a different, also random, knapsack with a given probability. This rather simplistic function was chosen to not cause too much disturbance for the more elaborate crossover function. This combination was expected to yield a better balance between exploration and exploitation and ultimately help obtaining better overall results.

3.6 Selection Function

As the problem being solved relies on optimizing three objective functions, where one function f_2 is always a trade off of the other two, a selection function that allows for non-dominated solution search was chosen - NSGA-II (Deb et al., 2002). This selection function relies on the: elitism principle, diversity preserving mechanism (Crowding distance) and non-dominated solution set determination (Calle, 2017).

The preference was given to NSGA-II due to the fact that the principles of this algorithm were explained during a student lecture in class, as well as the ease with which the calculation of the hypervolume of the Pareto front could be determined in DEAP. Hypervolume is a measure for the size of the space that is dominated by the obtained Pareto front. This measure was used to compare the Pareto fronts that were obtained by running the evolutionary algorithm using different parameters (population size, crossover and mutation probabilities). By performing hypothesis testing on the difference between hypervolumes obtained from different parameter settings, the optimal combination of parameters for different data set types was determined.

3.7 Evolutionary Strategy

During experimentation, two evolutionary strategies that are provided in the DEAP library were tested: simple and $\mu + \lambda$. However, using NSGA-II as the selection procedure was not compatible with the simple evolutionary algorithm. This is due to the 1:1 replacement of the entire parental generation by the offspring

population at every generation. In NSGA-II, any individual will appear at most once in the produced offspring generation. Using the simple evolutionary algorithm in combination with NSGA-II plainly did not conduct any selection, so the $\mu + \lambda$ algorithm was used.

In $\mu + \lambda$ the selection of individuals for the next generation is performed using the pool of both the parental and the offspring generations. This results in a more directed convergence towards solutions with a better fitness due to the fact that better individuals from two generations are kept rather than one. This nature of the $\mu + \lambda$ algorithm should be taken into account when evaluating the choice of an evolutionary algorithm for a specific problem, as this depends on the empirical hardness of the problem. The quality of the choice for $\mu + \lambda$ in the frame of this experiment is further discussed in the landscape analysis section.

4 Landscape Analysis

Landscape analysis was performed by empirically evaluating the information theoretic entropy of the fitness landscape. In this case the entropy is expressed in the information content of a random walk along the fitness landscape of the three objective functions. The information content is expressed as:

$H(\epsilon) = -\sum_{p \neq q} P_{[pq]} \log_6 P_{[pq]}$, where $p, q \in \{-1, 0, 1\}$ and $P_{[pq]} = \frac{n_{[pq]}}{n}$. n is the length of a string $S_i \in \{-1, 0, 1\}$ which represent an increase (1), decrease (-1) or no change (0) of fitness after applying the mutation operator on an individual. $n_{[pq]}$ is the number of occurrences of a sub-string pq in S_i . Data was obtained by applying the mutation operator 1000 times. This was repeated 30 times for 42 problem instances to handle the various number of knapsacks (3,5, or 10), number of variables (100,200, or 300), and densities (25%, 50%,75%, or 100%). A confidence interval was then generated for each combination, which is presented in the bar plot.

Thus, $H(\epsilon)$ is the degree to which a landscape contains various rugged shapes rather than neutral and smooth. The results were obtained for all three objective functions by applying the mutation operator to an individual for 1000 times. This was repeated 30 times for data sets, that varied in the number of variables, number of knapsacks and density, to gather statistics and display the confidence intervals, as shown in Figure 2. The closer the $H(\epsilon)$ value is to 1, the more rugged the landscape can be considered. A few observations can be made for the plotted results, as seen in Figure 2:

1. The $H(\epsilon)$ generally seems to decrease with an increasing number of variables and an increasing number of knapsacks
2. f_1 is empirically recorded to be the most rugged objective function in this Quadratic Knapsack problem variation.
3. With an increasing density, the difference in $H(\epsilon)$ with decreases as the number of variables increases.

The behavior, as stated in address point #1, can be explained by the nature of the mutation operator. At a lower number of knapsacks the chance of an item to not be included in the solution (not be assigned to any knapsack) as a result of a mutation increases, e.g. for $m = 3$ this chance is 25%, while for $m = 10$ this chance is only 9%. This leads to an increased volatility of the fitness values at lower m , which translates into an increased ruggedness of the fitness landscape for all functions. A similar thought process can be applied to the number of variables. As n increases, the impact that an item has on the solution fitness diminishes. Therefore, the fitness values will experience less volatility, which generally translates into a lower $H(\epsilon)$ value.

With an $H(\epsilon)$ value of around 0.8, f_1 appears to account for a major part of the difficulty in optimizing this objective function. We hypothesize that this phenomenon primarily occurs due to the quadratic aspect of the knapsack profit calculation. Applying a mutation, therefore, assigning an item to another knapsack, changes f_1 according to the way the item interacts with all items in the new knapsack. This makes the fluctuations in f_1 rather drastic, which is represented in the high degree of ruggedness.

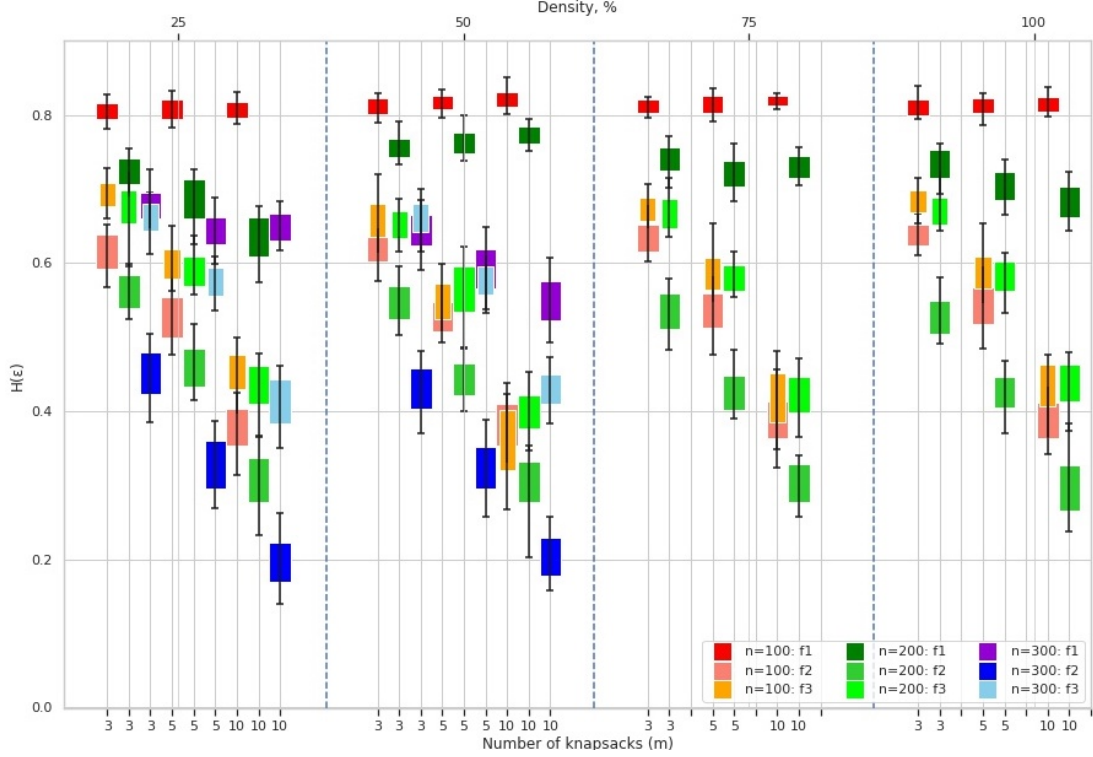


Figure 2: $H(\epsilon)$ confidence interval for all variable number - knapsack number - density combinations

Point #3 can be clarified by the nature of the item value/weight distribution in the data set. As the density decreases, the correlation between value and weight decreases, meaning that the scatter of the weight and value values increases. This is further aggravated by an increasing number of variables, which leads to more significant, yet unpredictable, differences in the ruggedness of the fitness landscape for the objective functions. As density increases, the fitness landscape becomes more predictable and the mutation does not cause as much volatility, which leads to a closer convergence of the fitness functions as the number of variables increases.

To conclude, the difficulty of this variation of the MOO Quadratic Knapsack problem appears to be mostly dependant on the number of knapsacks and number of variables. Therefore, we advise further researchers to account for this by designing deciders that would choose for a more optimal approach to handling the different parameter settings for a problem. Accounting for these differences could be done by changing the mutation or crossover operator or adjusting the probabilities for these operators. At last, the ruggedness of the landscape further supports the use of the $\mu + \lambda$ evolutionary strategy, since it aims to maintain a more directed search in the fitness landscape by focusing on keeping more fit individuals of two, rather than one, populations in the offspring.

5 Experimental Design

The solution obtained to the quadratic knapsack problem using an evolutionary approach depends heavily on the mutation rate and population size used. To identify the optimal parameters to use given a problem instance, a statistical analysis needed to be performed.

Data for this analysis was collected by running the genetic algorithm on data sets referenced from Billion-

net and Soutif (2004b) and Billionnet and Soutif (2004a). For each variable number - density combination, tests were run where the mutation rate was varied between .1 and .9 in increments of .1 ($m = .1, .2, .3, \dots, .9$) and the population size was varied between 100 and 250 in increments of 50 ($p = 100, 150, 200, 250$). Each evolutionary process ran for 200 generations. In order to speed up data collection, a random file for each density-variable number combination was chosen for each different mutation-population-knapsack number combination. This sampling method allowed us to down the number of potential trials from 9720 ($90 \times 36 \times 3$) to 1080 while still utilizing all given data sets for testing. A key assumption that allowed us to do this is that the data for each file with a given density and variable number (ex. `jeu_100_25_2.txt` vs. `jeu_100_25_7.txt`) followed a similar enough distribution that our algorithm would have the same behavior no matter the file.

In each run, the hypervolume of the produced front, the average profit of all solutions on the front, the average total weight of all solutions on the front, and the average minimum weight of all solutions on the front was collected. A visualization of the non-dominated front was saved as well.

The collected data was first averaged across each parameter and graphed. These graphs were used to hypothesize about the potential optimal parameters. Analysis of Variance tests were then performed on all the collected data to determine whether or not the population, the mutation rates, and the interaction between population and mutation had a correlation with the various performance parameters (hypervolume and the values of the various objective functions). For all these tests, a confidence level $\alpha = 99\%$ was used. To ensure the true correlation was recognized by the ANOVA test, each test was blocked by the data file (representing the variable number and density of the problem) and number of knapsacks.

If a variable was determined to have a significant effect on any factor, a Tukey Honestly Significant Difference test was performed to determine the best parameters. Again, this test utilized a confidence level $\alpha = 99\%$. The optimal parameters are determined by the set of parameters that have the largest difference in means from the rest of the parameters.

After an anslysis was done of the whole data set, an examination of how hypervolume changes with respect to mutation and population sizes was conducted with subsets of the data to possibly pinpoint any behavior that may change depending on the problem type. These subsets were runs with 3,5 and 10 knapsacks as well as runs with 100, 200 and 300 variables.

6 Results

6.1 Hypervolume analysis

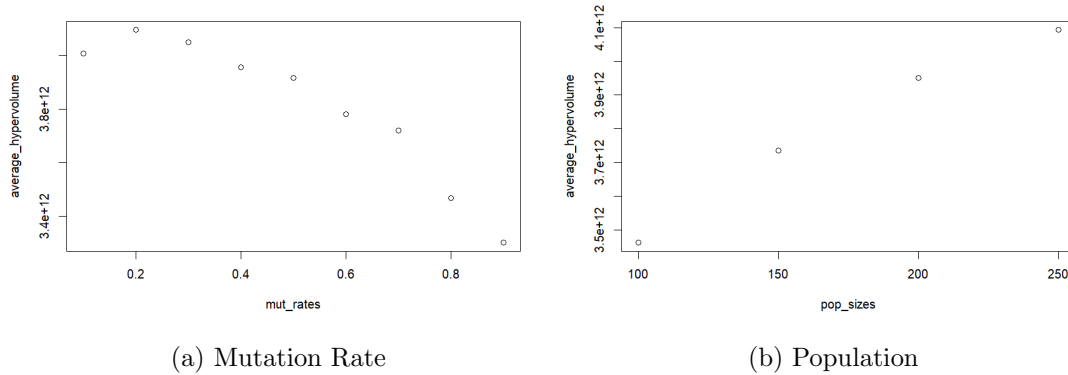


Figure 3: Average Hypervolume Plots

When graphing the relationship between average hypervolume and mutations rates and population sizes, a few trends become apparent. First off, it seems that the optimal mutation rate is around .2. Secondly, it seems that there is a strong linear relationship between the population size and the resultant hypervolumes. However, neither of these relationships can be confirmed as there may be a problem instance with abnormally large or small hypervolume that is skewing our data. Additionally, the differences in the problem being solved- specifically the differences in the profits, weights, number of variables, and number of knapsacks in each problem- may also contribute to bias in the data. To account for this, an ANOVA was performed with the differences in the problem instances as a blocking variable with following null hypotheses:

- $H_{0,1}$ = There is no significant correlation between population size and hypervolume.
- $H_{0,2}$ = There is no significant correlation between mutation rate and hypervolume.
- $H_{0,3}$ = There is no significant correlation between the interaction of population size + mutation rate and hypervolume.

```

              Df    Sum Sq   Mean Sq F value Pr(>F)
block         31 6.538e+28  2.109e+27 3600.61 <2e-16 ***
mut_rate       8 6.103e+25  7.629e+24   13.02 <2e-16 ***
Residuals    1040 6.092e+26  5.858e+23
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

              Df    Sum Sq   Mean Sq F value Pr(>F)
block         31 6.538e+28  2.109e+27 3620.49 <2e-16 ***
pop_size       3 6.146e+25  2.049e+25   35.17 <2e-16 ***
Residuals    1045 6.088e+26  5.826e+23
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

              Df    Sum Sq   Mean Sq F value Pr(>F)
block         31 6.538e+28  2.109e+27 3915.718 <2e-16 ***
pop_size       3 6.146e+25  2.049e+25   38.035 <2e-16 ***
mut_rate       8 6.103e+25  7.629e+24   14.163 <2e-16 ***
pop_size:mut_rate 24 2.106e+24  8.775e+22    0.163      1
Residuals    1013 5.456e+26  5.386e+23
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Figure 4: ANOVA Analysis Results

The ANOVA showed that the differences in the means of different mutation rates and populations are significant even after accounting for the differences in means caused by the different problem instances ($\alpha > 2 \times 10^{-16}$) for both mutation and population size, therefore $H_{0,1}$ and $H_{0,2}$ are rejected. However, we fail to reject $H_{0,3}$ and cannot definitively say that the interaction between mutation rates and crossover impacts the resultant hypervolume.

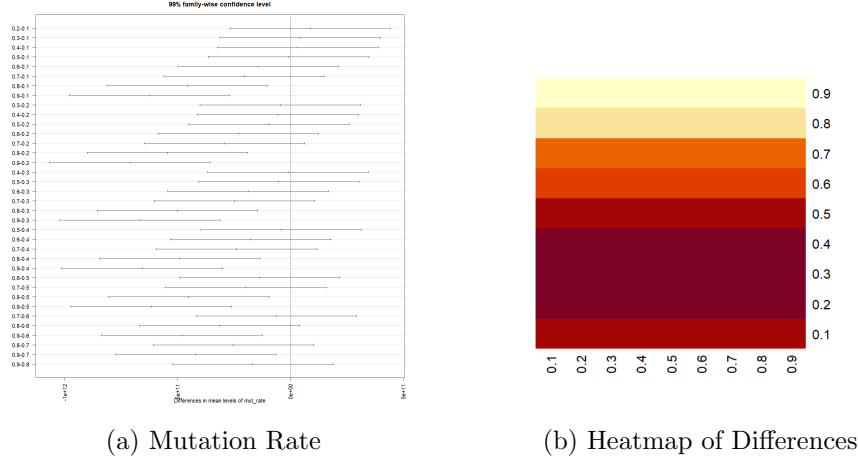


Figure 5: TukeyHSD Mutation Test Hypervolume Results

Looking at the TukeyHSD Test results for the mutation rates, we can see that 0.9 is significantly lower than every other mutation rate except for 0.8. Similarly, 0.8 is significantly lower than every mutation rate except for 0.7 and 0.9. Apart from this, there does not seem to be a statistically significant difference between any of the other mutation rates. To determine the optimal mutation among the ones without a significant difference, we can simply choose the ones that have the highest difference when compared to the rest of the mutation rates as well (as visualized in Figure 5) and revisit the hypothesis drawn from (Figure 3a). From these, we can determine that the best mutation rates is between 0.2-0.4.

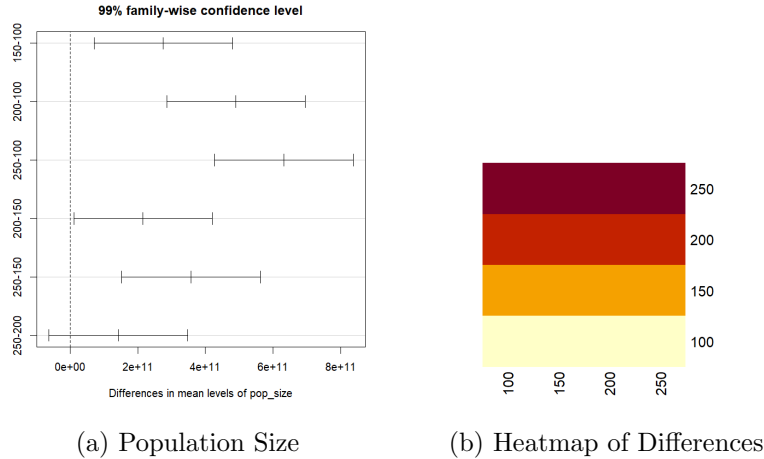


Figure 6: TukeyHSD Population Test Hypervolume Results

The results for the population sizes TukeyHSD tell a clearer story compared to the mutation rate test. The difference between every every population pair is significant apart from the difference between a population size of 200 vs. 250. Additionally, these differences are all positive, supporting the initial hypothesis that the higher the mutation rate, the higher the hypervolume, therefore a population size of 250 should be utilized to ensure the most optimal solution. One consideration that needs to be made is runtime however. Since our evolutionary algorithm utilizes a $\mu + \lambda$ strategy, the number of fitness calls per generation is equivalent to the population. Therefore using a population of 250 would result in $2.5 \times$ the number of fitness calls when compared to utilizing a population of 100 individuals. If runtime is a concern, a population of 200 should be utilized instead.

These optimal parameters do not change when subsetting the data set by knapsack size. Like the results of the ANOVA test for all the data, the tests when looking at the 3, 5, and 10 knapsack cases all result in rejecting $H_{0,1}$ and $H_{0,2}$ while failing to reject $H_{0,3}$. Additionally, the shape of the average hypervolume curves are extremely similar.

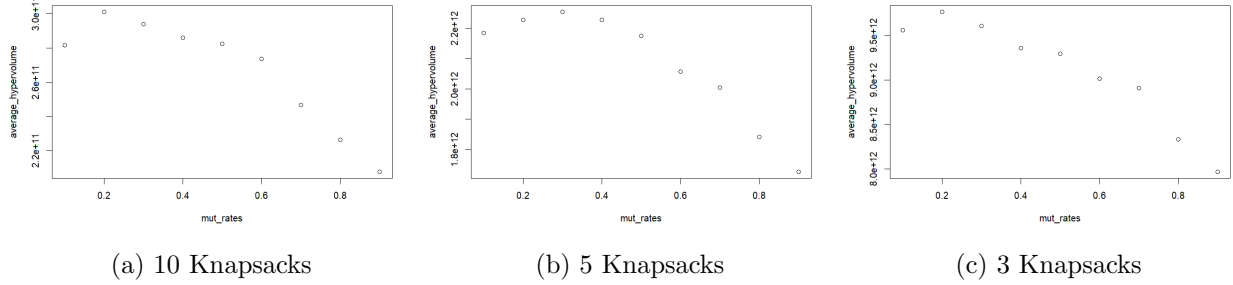


Figure 7: Average Hypervolume by Mutation Rate Curves for Different Knapsack Sizes

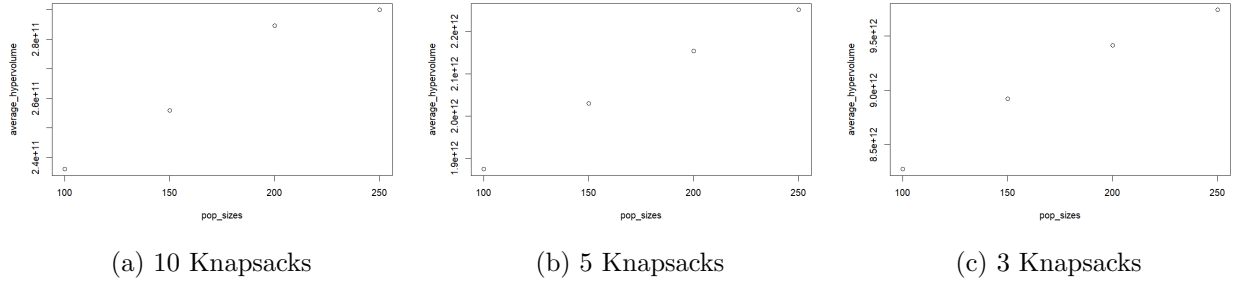


Figure 8: Average Hypervolume by Pop. Size Curves for Different Knapsack Sizes

The TukeyHSD Test results for each different knapsack sizes (not displayed) also confirm that optimal parameters for each knapsack do not vary from the overall optimal parameters.

Similarly, the optimal parameters for maximizing hypervolume do not vary based on the number of variables either. The ANOVAs reject $H_{0,1}$ and $H_{0,2}$ while failing to reject $H_{0,3}$ like the ANOVA performed to examine the whole data set. The shape of the average hypervolume curves (not displayed) and the conclusions from the Tukey HSD tests perform again determine that the best parameters no matter the number of variables are a mutation rate between .2 and .4 as well as a population of 250.

6.2 Total Profit Analysis

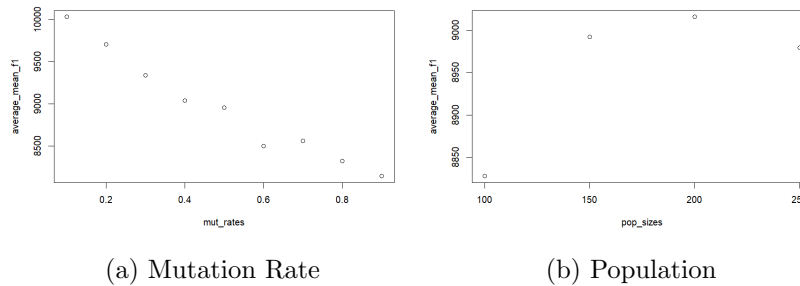


Figure 9: Average Total Profit Plots

When graphing the relationship between the average total profit of a front across different mutation rates and population sizes, a few differences arise when compared to the trending behavior of the hypervolume charts. The first difference is that there is no defined “peak” in the mutation rate graph, instead the entire graph seems to have a linear downtrend. Due to this, the theoretical best mutation rate is .1, but statistical tests still need to be performed to confirm this. The second difference is that, unlike the relationship between hypervolume and population size, the relationship between the average total profit of the produced fronts and the population size seems non-linear with a potential peak at a population size of 200.

Similar to hypervolume, ANOVA tests were run to test 3 main hypotheses.

- $H_{0,1}$ = There is no significant correlation between population size and Average Total Profit.
- $H_{0,2}$ = There is no significant correlation between mutation rate and Average Total Profit.
- $H_{0,3}$ = There is no significant correlation between the interaction of population size + mutation rate and Average Total Profit.

```

              Df    Sum Sq   Mean Sq F value Pr(>F)
block         31 1.429e+12 4.610e+10 4885.9 <2e-16 ***
mut_rate       8 9.631e+09 1.204e+09  127.6 <2e-16 ***
Residuals    1040 9.813e+09 9.435e+06
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

              Df    Sum Sq   Mean Sq F value   Pr(>F)
block         31 1.429e+12 4.610e+10 2535.603 < 2e-16 ***
pop_size       3 4.446e+08 1.482e+08   16.292 2.4e-10 ***
Residuals    1045 1.900e+10 1.818e+07
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

              Df    Sum Sq   Mean Sq F value   Pr(>F)
block         31 1.429e+12 4.610e+10 5067.372 < 2e-16 ***
pop_size       3 4.446e+08 1.482e+08   16.292 2.4e-10 ***
mut_rate       8 9.631e+09 1.204e+09  132.333 < 2e-16 ***
pop_size:mut_rate 24 1.524e+08 6.351e+06    0.698  0.857
Residuals    1013 9.215e+09 9.097e+06
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Figure 10: Total Profit Anova

The results of the ANOVA tests indicate that we can reject $H_{0,1}$ and $H_{0,2}$, but we fail to reject $H_{0,3}$.

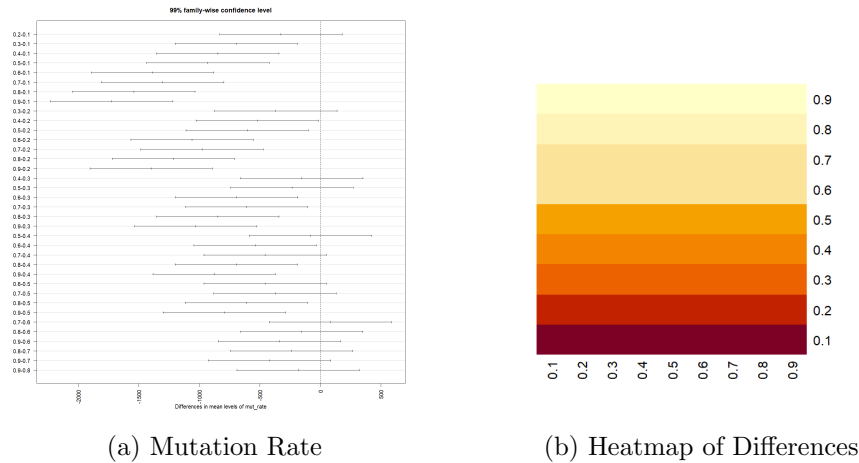


Figure 11: TukeyHSD Mutation Test Total Profit Results

The TukeyHSD test results indicate that a mutation rate of .1 has a statistically higher average total profit than every other mutation rate except for .2. This, combined with the shape of the population-total profit graph (figure 3b) indicate that a mutation rate of .1 is optimal.

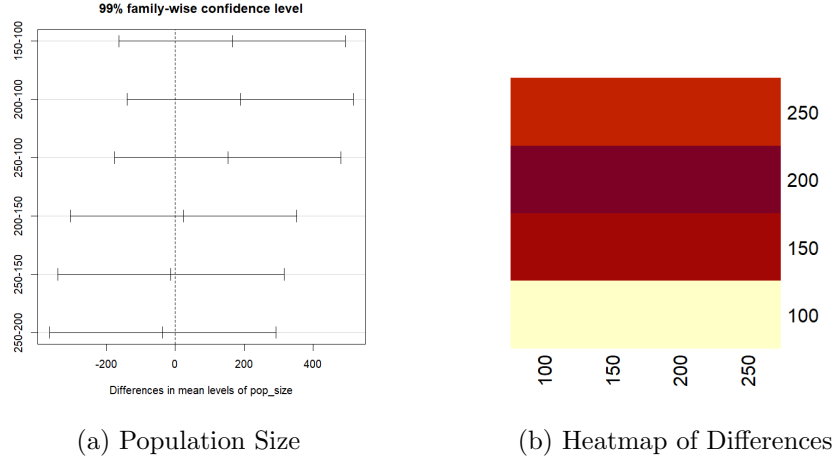


Figure 12: TukeyHSD Population Total Profit Results

Unlike the results suggested by the ANOVA, the TukeyHSD test analyzing population sizes and total profits suggest that there are no significant differences between any of the population sizes. This may be due to the variance of the average total profits across all the runs being too high, resulting in a large confidence interval. Therefore, a true optimal population size cannot be determined, therefore we will default to a population size of 200 as the optimal population due to it having the highest differences compared to the other population sizes and it having the highest mean when all the total profits are averaged across population size (Figure 9b)

6.3 Total Weight Analysis

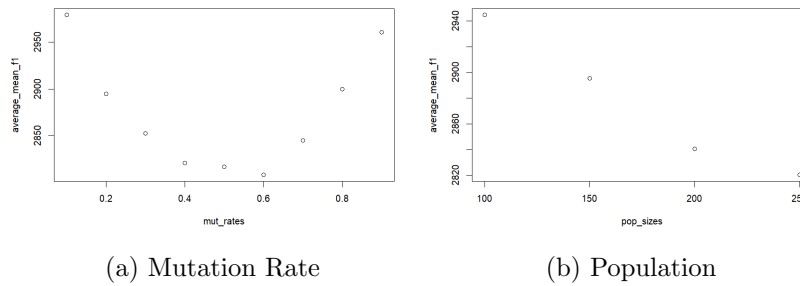


Figure 13: Average Total Profit Plots

When graphing the average knapsack weight with respect to mutation, a clear parabolic pattern can be seen with a minimum at around $m = 0.6$. Additionally, a downward trend can be observed with population sizes and the average total weight.

ANOVA tests were then performed to test 3 main hypotheses.

- $H_{0,1}$ = There is no significant correlation between population size and Average Total Weight.
- $H_{0,2}$ = There is no significant correlation between mutation rate and Average Total Weight.
- $H_{0,3}$ = There is no significant correlation between the interaction of population size + mutation rate and Average Total Weight.

```

              Df    Sum Sq Mean Sq F value Pr(>F)
block          31 1.986e+09 64070210 5603.60 <2e-16 ***
mut_rate        8 3.601e+06  450153   39.37 <2e-16 ***
Residuals    1040 1.189e+07   11434
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
              Df    Sum Sq Mean Sq F value Pr(>F)
block          31 1.986e+09 64070210 5177.08 <2e-16 ***
pop_size        3 2.560e+06  853228   68.94 <2e-16 ***
mut_rate        8 3.601e+06  450153   50.112 <2e-16 ***
Residuals    1045 1.293e+07   12376
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
              Df    Sum Sq Mean Sq F value Pr(>F)
block          31 1.986e+09 64070210 7132.407 <2e-16 ***
pop_size        3 2.560e+06  853228   94.983 <2e-16 ***
mut_rate        8 3.601e+06  450153   50.112 <2e-16 ***
pop_size:mut_rate 24 2.317e+05    9653    1.075 0.366
Residuals     1013 9.100e+06    8983
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Figure 14: Total Weight Anova

The results of the ANOVA tests indicate that we can reject $H_{0,1}$ and $H_{0,2}$, but we fail to reject $H_{0,3}$. This is similar to the results of the ANOVA tests performed with hypervolume and total profit as its factors.

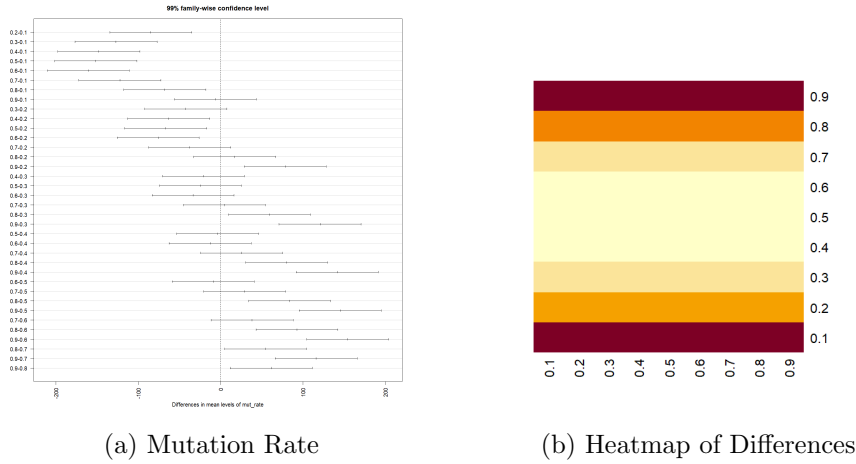


Figure 15: TukeyHSD Mutation Test Total Profit Results

The Tukey HSD test with mutation as a factor shows that many mutation rates do have significant differences between each other. Specifically, the average total weight generated by the “outer” mutation rates (0.1,0.2,0.8,0.9) is significantly larger than the “inner” mutation rates. The heatmap suggests that the optimal mutation rates to minimize the total weight in a problem instance is between 0.4 and 0.6.

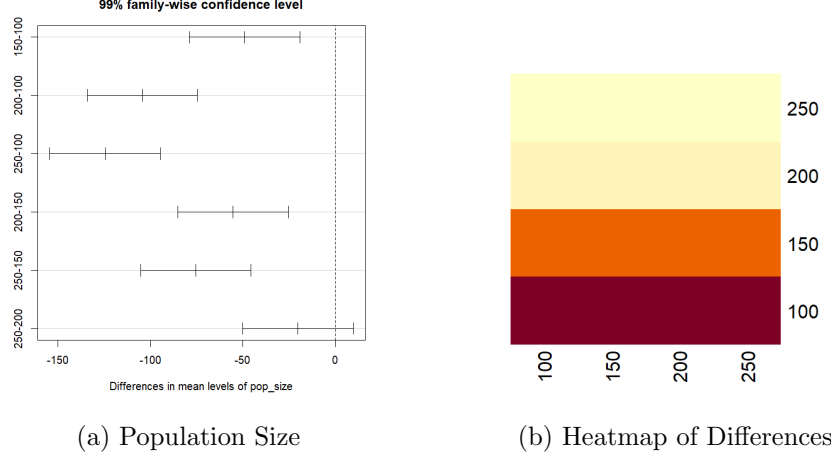


Figure 16: TukeyHSD Population Total Profit Results

The TukeyHSD with population size as a factor shows that a population of 250 individuals produces significantly smaller total weights when compared to all other population sizes except for 200 individuals. This suggests that 250 individuals is the best population size when attempting to minimize the average total weight of a front.

6.4 Minimum Profit Analysis

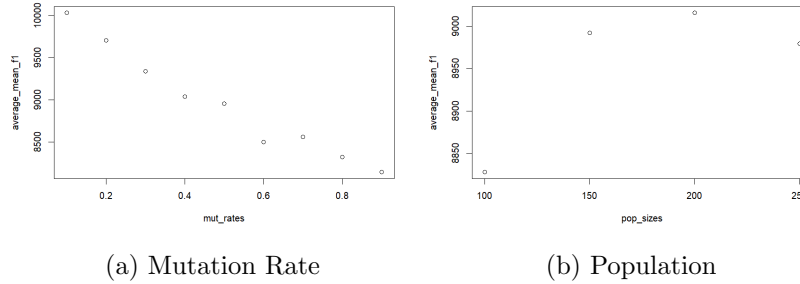


Figure 17: Average Minimum Profit Plots

When the average minimum profit is plotted with respect to mutation rates and population sizes, the shape of the graphs is very similar to the shape of the graphs with the average total profit (figure 9). Due to this, similar conclusions about the optimal mutation rate and population size can be drawn. The theoretical optimal mutation rate is 0.1 and the best population size can be theorized to be around 200 individuals.

Like the rest of the parameters, ANOVA tests were run to test 3 main hypothesis.

- $H_{0,1}$ = There is no significant correlation between population size and Average Minimum Profit.
- $H_{0,2}$ = There is no significant correlation between mutation rate and Average Minimum Profit.
- $H_{0,3}$ = There is no significant correlation between the interaction of population size + mutation rate and Average Minimum Profit.

```

              Df    Sum Sq   Mean Sq F value Pr(>F)
block          31 1.172e+11 3.781e+09 3194.57 <2e-16 ***
mut_rate        8 3.291e+08 4.114e+07   34.76 <2e-16 ***
Residuals     1040 1.231e+09 1.184e+06
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

              Df    Sum Sq   Mean Sq F value Pr(>F)
block          31 1.172e+11 3.781e+09 2542.387 <2e-16 ***
pop_size        3 5.931e+06 1.977e+06    1.329  0.263
Residuals     1045 1.554e+09 1.487e+06
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

              Df    Sum Sq   Mean Sq F value Pr(>F)
block          31 1.172e+11 3.781e+09 3188.241 <2e-16 ***
pop_size        3 5.931e+06 1.977e+06    1.667  0.172
mut_rate        8 3.291e+08 4.114e+07   34.691 <2e-16 ***
pop_size:mut_rate 24 2.365e+07 9.852e+05    0.831  0.699
Residuals      1013 1.201e+09 1.186e+06
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Figure 18: Minimum Profit ANOVA

The results of this ANOVA show that we can reject $H_{0,2}$. However, unlike the other tests perform, we fail to reject $H_{0,1}$ indicating that the correlation between population size and the average minimum profit of the generated front is very weak. As a result, a Tukey Test was not performed to analyze the optimal population size.

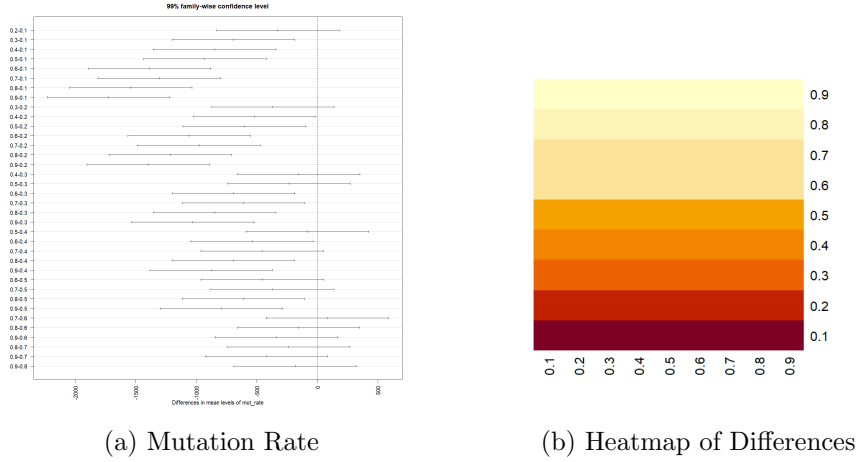


Figure 19: TukeyHSD Mutation Test Minimum Profit Results

The results of the TukeyHSD test indicate that the average minimum profits generated with a mutation rate of .1 is significantly higher than all other mutation rates except for .2. This, combined with the heatmap, indicate that the optimal mutation rate is .1 if one desires to maximize the minimum profit.

6.5 Analysis of the Fronts

Using the parameters derived, the algorithm was run on a few instances and the resultant non-dominated Pareto fronts were plotted. In each front, $f_1(x)$ is the total profit of the solution, $f_2(x)$ is the total weight of the solution, and $f_3(x)$ is the minimum profit in a knapsack of a solution.

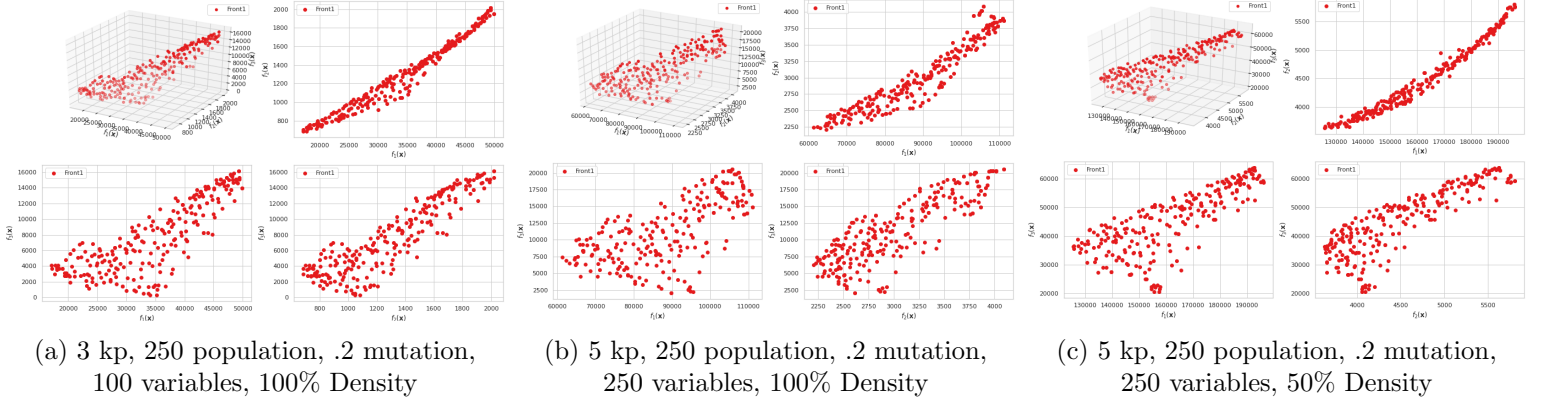


Figure 20: Non-Dominated Pareto Fronts

One common trend between all of the fronts is a strong relationship between f_1 and f_2 . As f_1 increases, f_2 also increases. However, no strong relationship seems to be found between f_2 and f_3 or f_1 and f_3 .

7 Conclusion

The best parameters to maximize the hypervolume of the front for the QKP problem are a mutation rate of .2-.4 and a population size of 250. These parameters do not significantly change when changing the number of knapsacks or the number of variables in the problem. If one wishes to generate a Pareto front that skews more towards maximizing the total profit or maximizing the minimum profit in a given knapsack, a population size of 200-250 and a mutation rate of .1 should be utilized.

If one wishes to create a Pareto front that skews more towards minimizing the total weight of the items in each knapsack, then a mutation rate between .4 and .6 should be utilized with a population size of 250.

This optimal mutation rate to minimize total weight is relatively much higher than the mutation rate needed to maximize the profit of the knapsacks (both total and minimum). This phenomena can be attributed to the nature of the mutation function. If both of an individual's parents have item i in knapsack k , then the only way that an individual can "lose" item i is through mutation. Therefore, a higher mutation rate allows more items to not be in knapsacks. However, since our algorithm utilizes a $\mu + \lambda$ evolutionary strategy, a higher mutation rate results in a lower crossover. Therefore, as the mutation rate increases, the algorithm loses the benefits of crossover resulting in a decrease in performance. This additionally explains why the optimal mutation parameters for maximizing profit is very low. A lower mutation rate allows more items to remain in knapsacks, effectively pushing the profits of the knapsacks to increase.

References

- Alain Billionnet and Eric Soutif. An exact method based on lagrangian decomposition for the 0-1 quadratic knapsack problem. *European Journal of Operational Research*, 157:565–575, 2004a. URL <https://hal.archives-ouvertes.fr/hal-01124918>.
- Alain Billionnet and Éric Soutif. Using a mixed integer programming tool for solving the 0–1 quadratic knapsack problem. *INFORMS Journal on Computing*, 16(2):188–197, 2004b. doi: 10.1287/ijoc.1030.0029. URL <https://doi.org/10.1287/ijoc.1030.0029>.
- Paul Calle. Nsga-ii explained! <http://oklahoaaanalytics.com/data-science-techniques/nsga-ii-explained/>, October 2017.
- K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi: 10.1109/4235.996017.
- S. Martello. Knapsack problems: Algorithms and computer implementations. *Wiley-Interscience series in discrete mathematics and optimization*, 1990. URL <https://ci.nii.ac.jp/naid/20000416220/en/>.