Candidate number
236636

Exploring the impact of the number of tournaments and crossover probability on the fitness of Full Microbial Genetic Algorithms

Acquired Intelligence & Adapative Behaviour (G6042)

17-05-2022

Abstract

We investigate the change in performance of a Full Microbial Genetic Algorithm (FMGA) with variations in the number of tournaments and crossover rate. The first two parts investigate the hyper-parameters individually, and the final one aims to find the optimal number of each. In the first part, we generate multiple FMGAs, run them over 10'000 tournaments, and make an average of their best fitness at each tournament to plot its progress. In the second part, we follow the same method but create averaged subsets of FMGAs where each group has a different crossover rate, and investigate the assumptions of this parameter. We found it relies heavily on the initial diversity of the population. In the third part, we create a heat-map of many subsets of FMGAs with varying numbers of tournaments and crossover rates to show which combination performs best. We measure the performance by making our FMGAs solve the knapsack problem. We found that the crossover rate had much less impact than the number of tournaments on the fitness of an FMGA, as long as it is between 0.2 and 0.9.

1 Introduction

Evolution can be summarised as random individuals fighting for their life for the winner's genetic makeup to be passed on. Full Microbial Genetic Algorithms (FMGAs) (see papers [2] and [3]) employ similar rules by defining a random population of individuals to solve a problem and makes them battle. The winner is the individual who is closest to the solution, and the loser's genetic makeup is crossed with some of the winner's. Fights are called tournaments, the probability of one of the loser's gene being crossed is the crossover-rate, and we made these FMGAs solve the knapsack problem. This problem defines a bag and items with benefits that need to be stored without exceeding the bag's weight limit. The closer the best individual in an FMGA's population is to the maximum weight, the fitter the FMGA is. We will explore the effect of the number of tournaments, and of the crossover rate, on the FMGAs fitness. We expect a trend of more tournaments and a crossover probability of 0.5 to maximise the model's performance.

2 Methodology

We will explore the effects of the number of tournaments, crossover rate, and both together on the fitness of the FMGA model. We pre-define a knapsack problem and generate an initial random population of genotypes. The fitness of each genotype is defined by the solution they propose for the knapsack problem. The fitness is 0 if it goes above the maximum weight of the problem. We then run tournaments where we choose a random individual from the population and one of its neighbours, pick a winner and loser according to their fitness, and replace some of the loser's genes with some of the winner's, in a crossover operation. This should improve the loser's fitness, otherwise, it keeps its original genes. With tournaments and crossover operations, the population should, in theory, have an improving overall fitness over time, hopefully reaching the knapsack solution. However, this depends on the number of tournaments, the crossover's parameters and its assumptions.

When measuring the fitness of an FMGA, per tournament or at the end, we take the population's individual with the best fitness at that tournament. This means we are not taking into consideration any under-performing individuals.

2.1 Defining the FMGA

This is a brief section to show how we define an FMGA. When we train this type of model, the information we want to extract from it is the fitness the individuals have at each tournament. This permits us to compare how different models' fitness changes as the number of tournaments or population size changes. When implementing this method, we can define more or less hyper-parameters to input into the algorithm 1 in order to explore the changes in output.

Algorithm 1 Define and train FMGA

Input: number of epochs, population size
Output: fitness of each individual per tournament

```
population = list of random genotypes
populationFitnessPerTournament = []
for t = 1, 2, ..., epochs do
    firstGenotype = getRandomGenotype(population)
    secondGenotype = getRandomGenotypeInNeighbourhood(population,firstGenotype)
    if getFitness(firstGenotype) is less than getFitness(secondGenotype) then
        winner = secondGenotype
        loser = firstGenotype
    else
        winner = firstGenotype
        loser = secondGenotype
        loserReplacement = crossover(winner,loser)
    end if
    if fitness(loserReplacement) is more than fitness(loser) then
        population[loser] = loserReplacement
    end if
    populationFitnessPerTournament.append(getPopulationFitness(population))
end for
return populationFitnessPerTournament
```

2.2 Number of tournaments

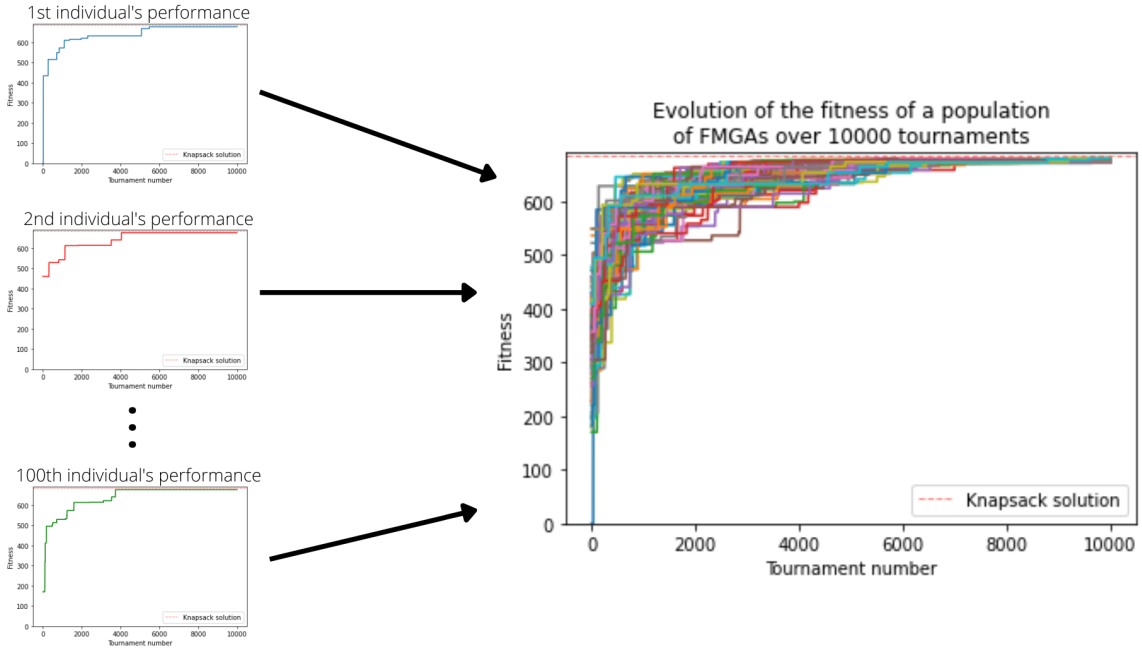


Figure 1: This figure plots the evolution of the fitness of 100 individuals, all part of the same FMGA. On the left, we've plotted three individually to show how we concatenated them into the same plot. The red dotted line at the top is the solution to the knapsack problem we defined. We observe that at the 8'000th tournament, most individuals have arrived at the solution.

Algorithm 2 Get average best fitness of multiple FMGAs at each tournament

Input: number of tournaments T, number of FMGAs N
Output: average best fitness at each tournament

```
listOfFMGAs = generate and train N FMGAs
averageBestFitness = []
for t = 1, 2, ..., T do
    bestFitnessAtThisTournament = []
    for model = 1, 2, ..., N do
        bestFitnessAtThisTournament.add(maxFitness(model[t]))
    end for
    averageBestFitness.add(getAverage(bestFitnessAtThisTournament))
end for
return averageBestFitness
```

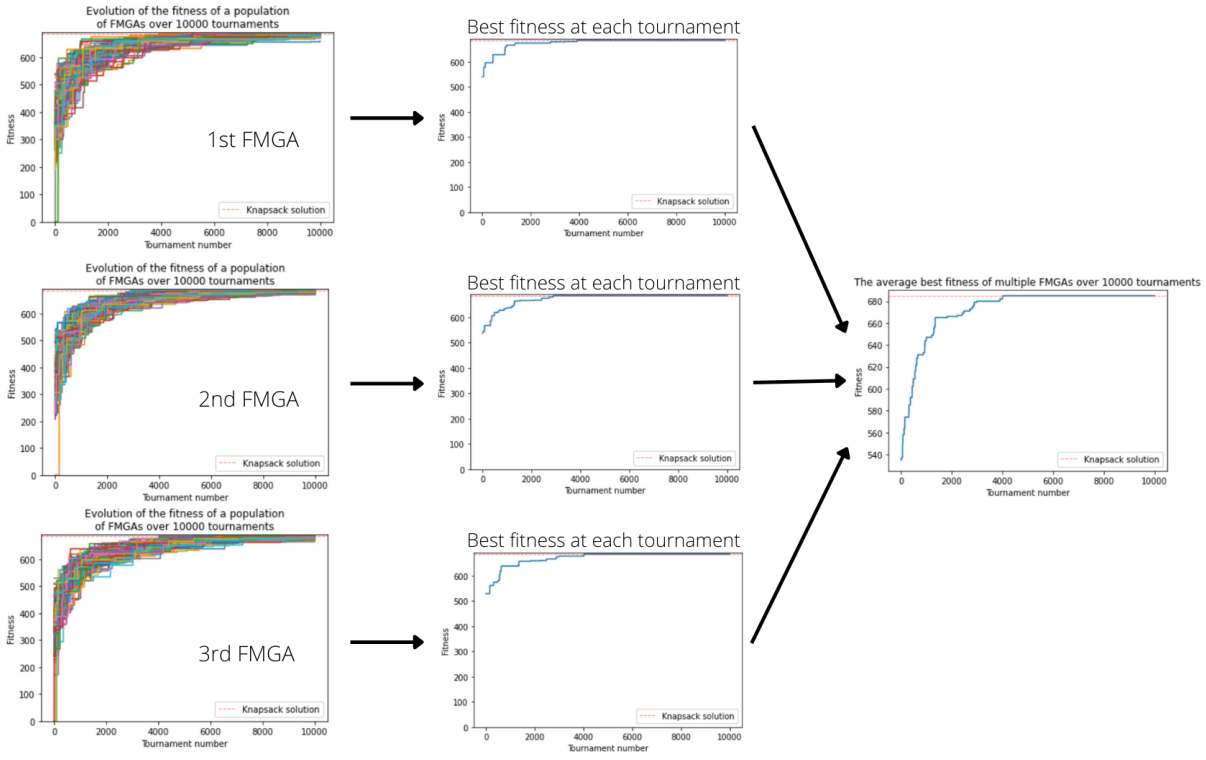


Figure 2: This figure shows the general method for getting the average best fitness per tournament, for 3 FMGAs. For each FMGA, we take their best fitness at each tournament and make an average of them. The plot on the far right shows the final result: the average best fitness of an FMGA model at each tournament.

To evaluate how the number of tournaments affects the performance of FMGAs, we first create and train many FMGAs, with one example in the figure 1. Following the algorithm 2, with the method's step shown visually in the figure 2, we take the best fitness at each tournament of every FMGA and make an average. This will result in the average best fitness of the FMGA model at each tournament. We can use these results to visually observe if the model's fitness increases with the number of tournaments. Furthermore, we can use this data to quantitatively prove if there is a correlation between fitness and the number of tournaments by measuring the Pearson Correlation Coefficient (PCC) [1]. A visual change in fitness and the PCC will together help us conclude if there is a correlation between the number of tournaments and the fitness of FMGAs.

Algorithm 3 Get average best fitness at end of tournament of FMGAs with different crossover rates

Input: number of tournaments T, number of FMGAs N
Output: average best fitness at final tournament for different crossover rates

averageBestFitnessAtCrossRate = []
for $crossRate = 0.0, 0.01, \ldots, 1.0$ **do**
    crossRateFitness = 0
    for $i = 1, 2, 3$ **do**
        crossRateFitness += getBestFitnessLastTournament(trainFMGA(crossRate))
    end for
    averageBestFitnessAtCrossRate.add(getAverage(crossRateFitness = 0)
end for
return averageBestFitnessAtCrossRate

2.3 Crossover rate

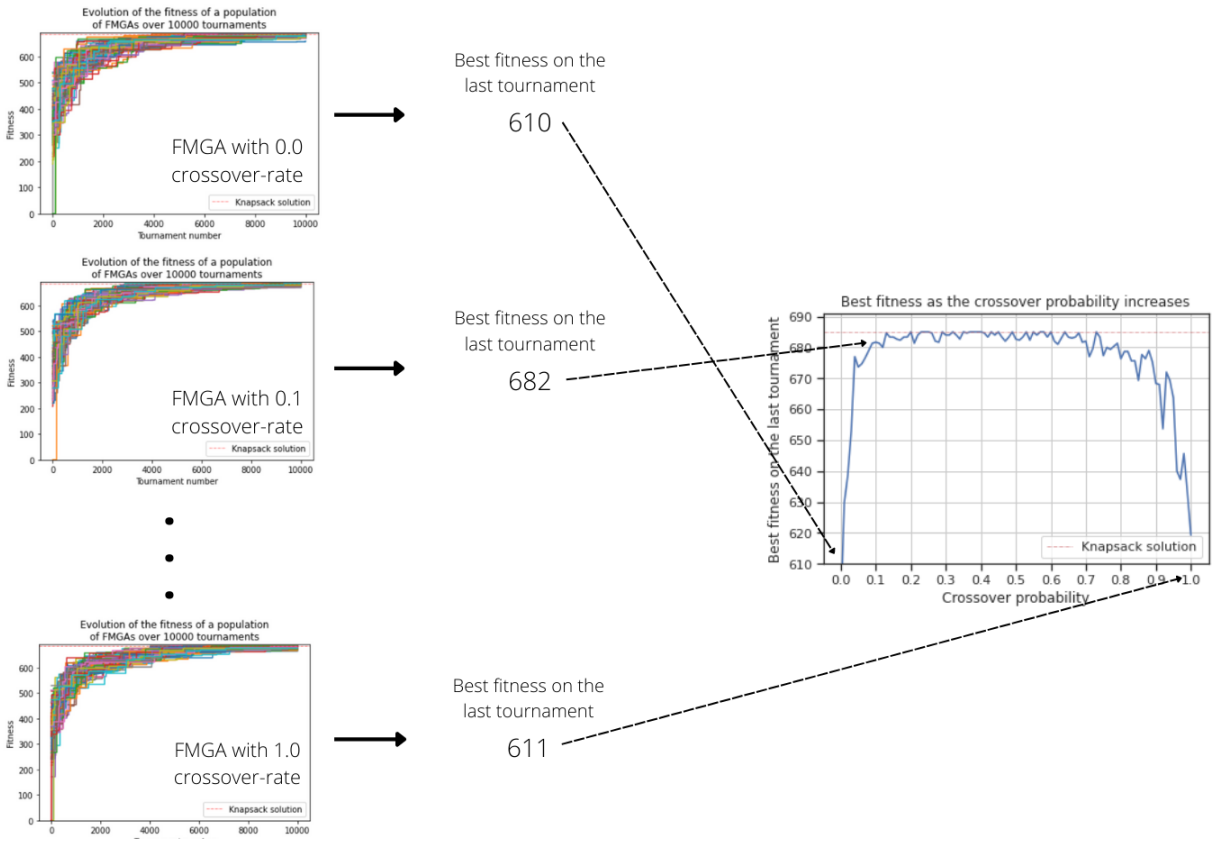


Figure 3: This figure, similarly to figure 2, shows the general method for evaluating the impact of the crossover rate on the fitness of the model described in algorithm 3. On the far left, we have a set of FMGA models, each with a different crossover rate, for which the best fitness in their last tournament is extracted, and then plotted. The result on the far right of the figure lets us visualise the fitness of each model (y-axis) compared to the crossover rate of that model (x-axis).

We measure the effect of the crossover rate on the performance of FMGAs in a very similar way to how we measure the effect of the number of tournaments: average out the best performance at a different hyper-parameter. The difference here is that we only consider the best performing individual of FMGAs on the last tournament as that is indicative of how close the FMGA is to solving the knapsack problem. The number of tournaments for each FMGA is the same, as for all other hyper-parameters, and we choose that number according to section 2.1's results to make sure we maximise that performance. When implementing this method, we would prefer to get an average of many models with the same crossover rate, due to the random nature of the FMGA model. This method will provide the data to measure the correlation, with PCC, between the fitness of the FMGA model and the crossover rate.

2.4 Optimum number of tournaments and crossover rate

We combine the previous two methods to optimise our FMGA model. This will provide two answers: what are the optimal hyper-parameters to maximise the performance of an FMGA model, and do these hyper-parameters affect each other. We will combine the models tested in a heat-map that will outline, with color representing the fitness, which model performs best.

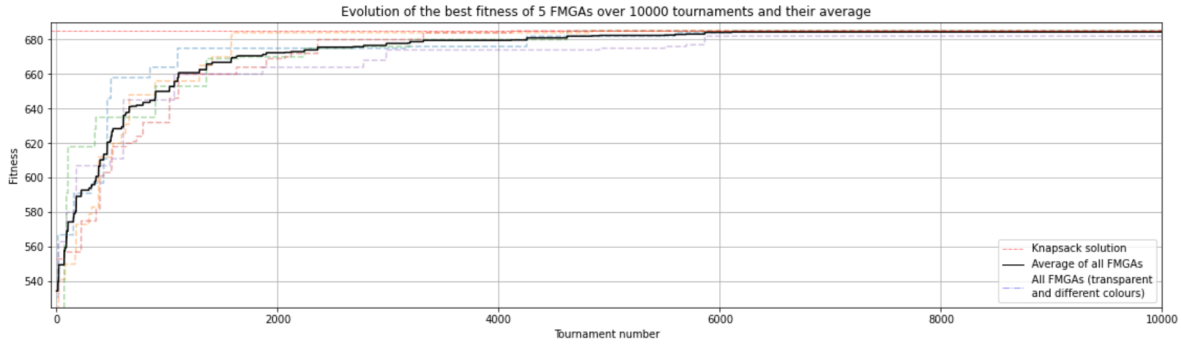
Algorithm 4 Get average best fitness of FMGAs with varying numbers of tournaments and crossover rates

Input: number of FMGAs N
Output: average best fitness of each FMGA with different hyper-parameters

averageBestFitnessAtCrossRate = []=
for $crossRate = 0.0, 0.01, \ldots, 1.0$ **do**
    crossRateFitness = 0
    for $i = 1, 2, 3$ **do**
        crossRateFitness += getBestFitnessLastTournament(trainFMGA(crossRate))
    end for
    averageBestFitnessAtCrossRate.add(getAverage(crossRateFitness = 0)
end for
return averageBestFitnessAtCrossRate

3 Results

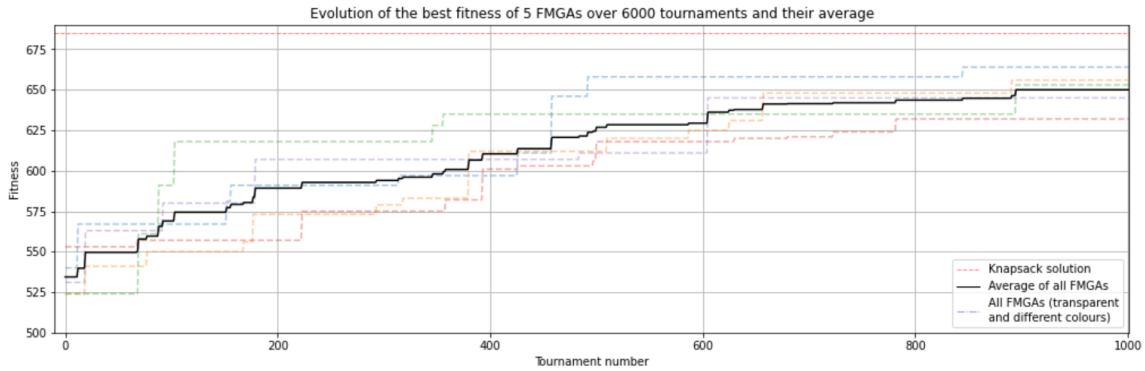
3.1 Effect of number of tournaments on performance



Number of tournaments	0	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000	Knapsack solution
Average best fitness	534	650	672	678	680	682	684	684	684	684	684	685

Figure 4: This figure plots the average best fitness of 5 FMGAs over 10'000 tournaments, as a black line. The table underneath shows the value of the average best fitness every 1'000 tournaments. The faint dotted lines represent each FMGA's average fitness at each tournament. The straight red dotted line at the top represents the solution to the knapsack problem.

We found that, as the number of tournaments increases, the average performance of the FMGA model also increases 4. Taking in consideration the parameters of our knapsack problem, there is a quick increase in fitness between 0 and 3'000 tournaments, and then a plateau that slowly increments to the solution. This preliminary plotting visualises the positive change in fitness as the number of tournaments increases.



Number of tournaments	0	100	200	300	400	500	600	700	800	900	1000	Knapsack solution
Average best fitness	534	569	589	594	610	627	629	641	644	650	650	685

Figure 5: This figure plots the same data as 4 but zoomed in to show the quick increase in average fitness between 0 and 1'000 tournaments.

For the data shown in 5 that shows the quick increase in fitness, we found a PCC of 0.95. This indicates a very strong correlation between the number of tournaments and the average fitness of the FMGA model. In contrast, the PCC between the fitness change and the 10'000 tournaments is 0.62. We can attribute this decrease to the plateau that starts at 3'000 tournaments 4. We can confirm from the data that an increase in the number of tournaments equals an increase in the fitness of the FMGA model.

3.2 Effect of crossover rate on performance

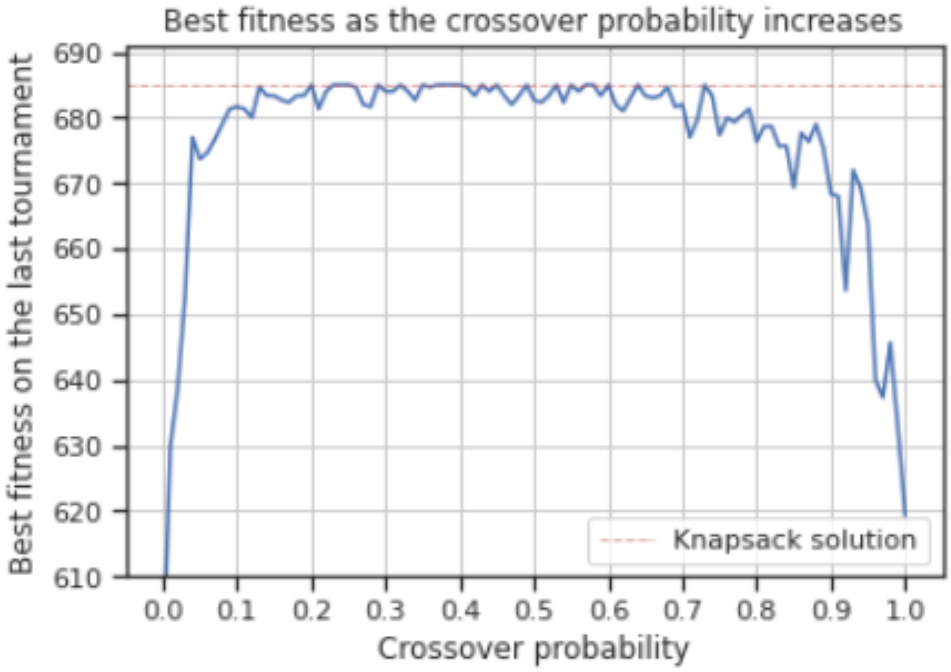


Figure 6: This figure plots the average fitness of FMGAs at different crossover probabilities. At each crossover probability we wanted to test, we trained 3 FMGAs with that hyper-parameter, averaged out the best fitness of each at the last tournament, and plotted it.

We found that the best crossover rate is at 0.5 and any less or more, between 0 and 1, only decreases the best fitness of the FMGA model (see figure 6). This is likely due to the randomness effect of the 0.5 crossover rate. We also found a PCC of -0.19 between the fitness of the FMGA model and the crossover probability, meaning that there is no correlation between them. We can deduct that there

is an optimal crossover rate boundary, which is between 0.2 and 0.6 (see figure 6), but no overall improvement or decrease in fitness as the crossover rate changes.

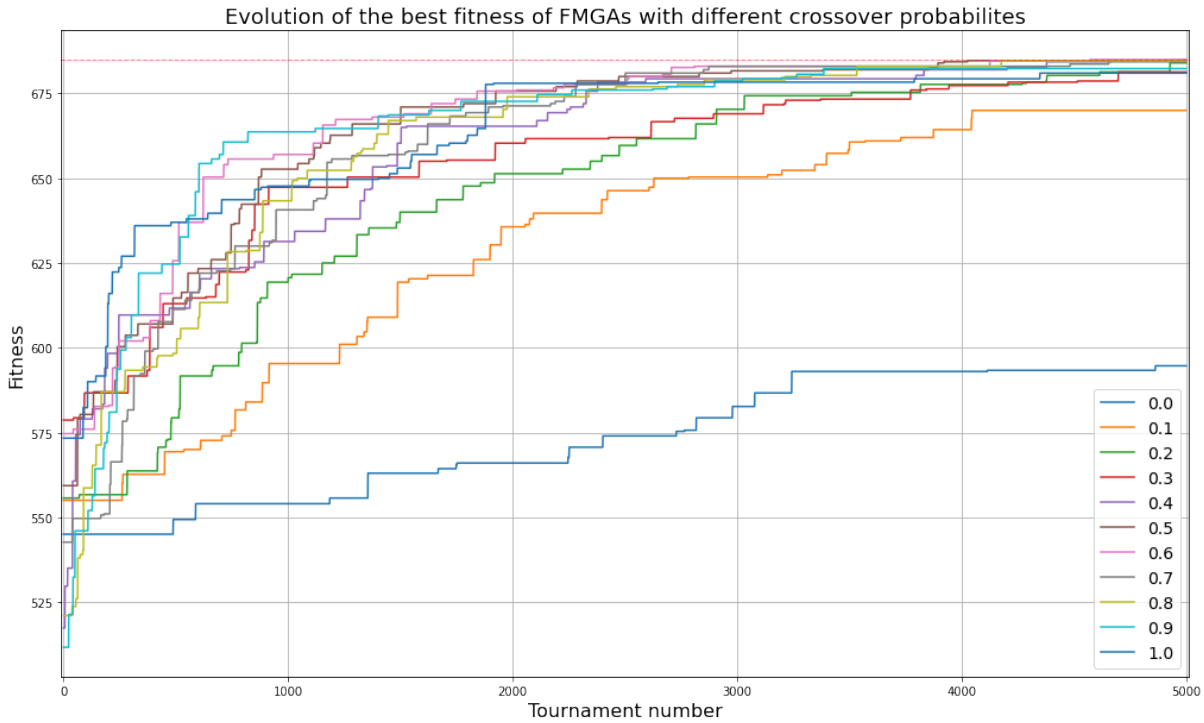


Figure 7: Similarly to figure 6, this figure plots the progress of the average best fitness of FMGAs with different crossover probabilities. Each coloured line represents the best fitness of a model trained with a different crossover probability. The red dotted line at the top represents the solution to the knapsack problem.

Here we outline the theory of what happens in the extreme cases of crossover rates that figure 7 displays:

1. At a crossover rate of 1, on the search space, the winner brings the loser to his exact position, so the population moves by converging to where the winners already are. This means that the diversity of the population now depends on the mutation rate.
2. At a crossover rate of 0.5, the winner brings the loser halfway closer to him, giving him a new position. This increases the diversity as the losers are spread around the search space.
3. At a crossover rate of 0, the loser does not move at all, there is no point in running the tournament. The diversity of the population depends on the mutation rate.

The crossover operation randomly switches some of the loser's genes with some of the winner's genes, which largely depends on how different the winner and the loser are, implying the success of the crossover operation in improving the population's fitness depends on the initial diversity. We can expect that if individuals in a population all had the same genotype, the crossover rate wouldn't matter, the tournaments would never find a winner; the change in the fitness would completely depend on the mutation rate. Therefore, we will explore the assumption the crossover rate makes about the initial diversity of the population by plotting models with varying hyper-parameters

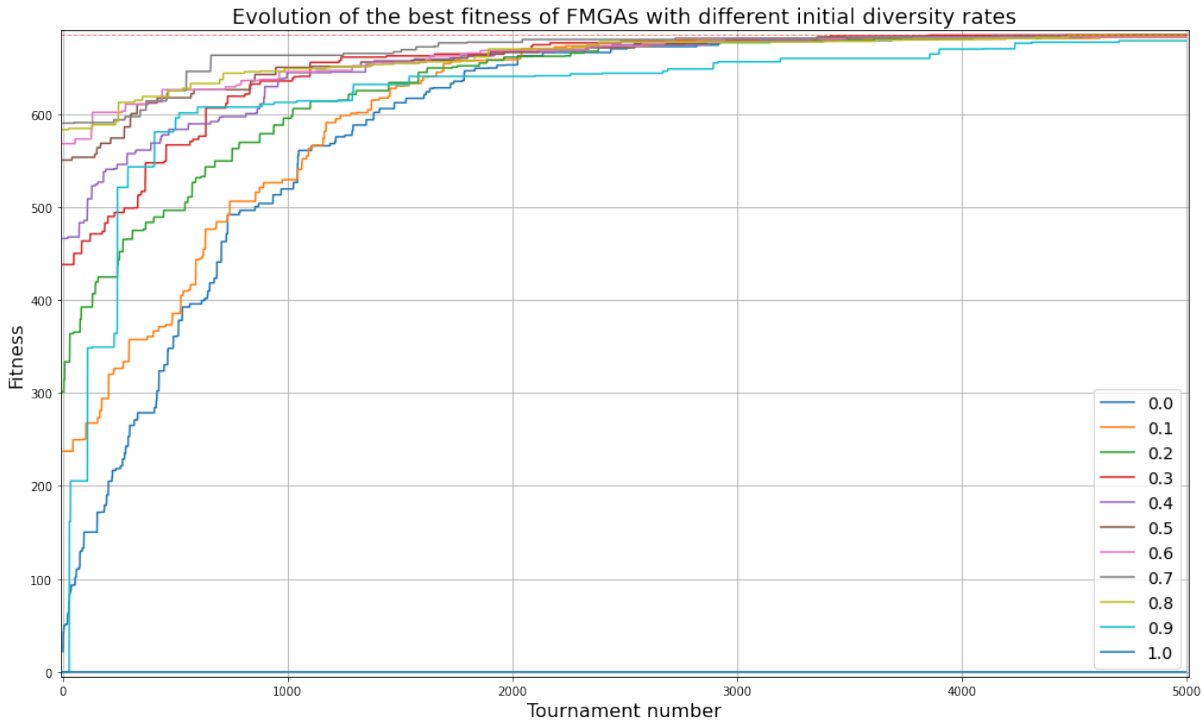


Figure 8: In this figure, we plotted FMGAs with a crossover rate of 0.5 and a varying initial diversity rate. This means that with an initial diversity rate of 0, all genotypes will be made of 0s at the start; with 0.5 they'll be made of an equal mix of 0s and 1s; with 1 they will all be made of 1s. Each coloured line represents the best fitness of an FMGA model per tournament with varying initial diversity rates.

We plotted multiple FMGAs with varying initial diversity rates (in figure 8) and found that the more diverse we get, meaning the more we have an equilibrium of 0s and 1s, the better the model gets. We can observe that with a diversity rate of 0, 0.1, 0.9 or 1, the model struggles to get to the knapsack solution compared to the other initial diversity rates. We can even see that the model with an initial diversity rate of 1 is stuck with a fitness of 0, which is due to how our knapsack problem is made: going over the maximum weight returns a fitness of 0, and since a genotype of all 1s selects all items it always gets a 0; even if the mutation rate makes it deselect an item, choosing all items but 1 still exceeds the weight limit, so the genotype is never replaced: the FMGA can never improve.

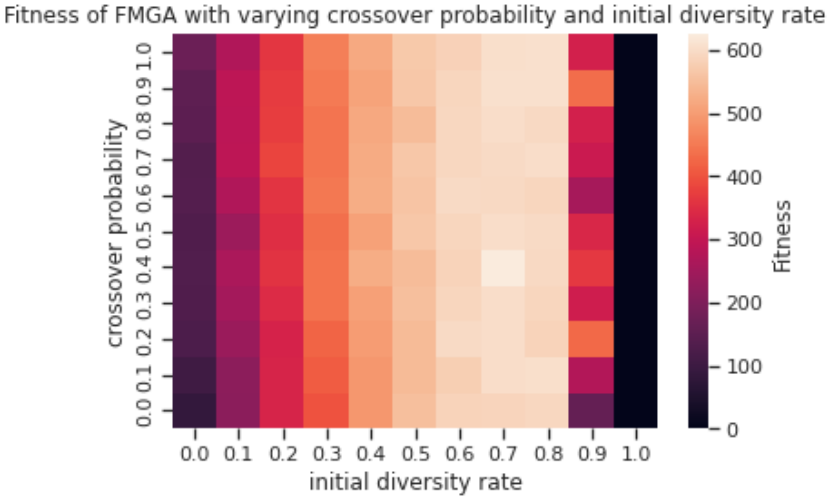


Figure 9: This figure plots the averaged out fitness of FMGAs (colour) with different initial diversity rates for their population (x-axis) and different crossover probabilities played out in their tournaments (y-axis).

The crossover probability hyper-parameter is dependent on the initial diversity rate of the pop-

ulation of an FMGA. In figure 9, we show how diverse the initial population is the decider of the overall fitness of the FMGA. We can observe that if the initial diversity rate is low (e.g 0 or 0.9), it does not matter what the crossover rate may be: the fitness will be low. In contrast, we find that an initial diversity rate of 0.7 with a crossover rate of 0.4 maximises the fitness of our FMGA model. It is also important to remember that the crossover rate and the initial diversity rate both also depend on the mutation rate for the FMGA to improve, which is a hyper-parameter we have kept constant in our experimentation. We now recognise that the crossover probability is fully dependent on the initial diversity rate of the population of the FMGA.

3.3 Optimising tournament number and crossover rate

Having found the best initial diversity rate to maximise the fitness of our FMGA model, which the crossover probability hyper-parameter depends on, we can now find the best combination of number of tournaments and crossover probability.

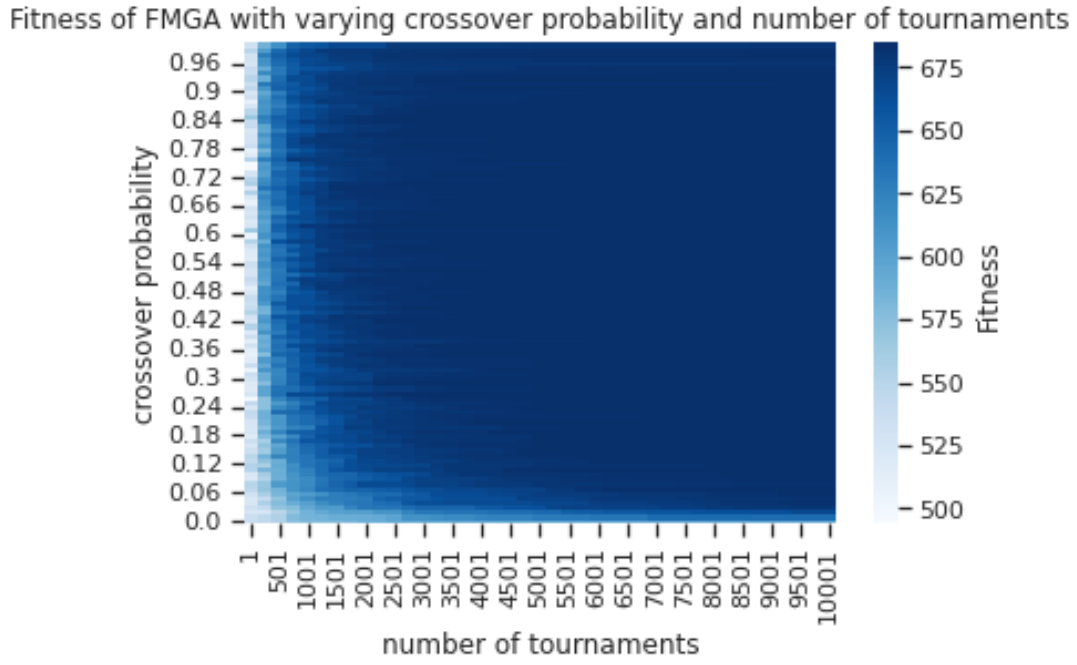


Figure 10: This figure plots the averaged out fitness of FMGAs (colour) at an increasing number of tournaments with varying crossover rates. This shows how fit an FMGA gets after a certain number of tournaments at a specific crossover rate.

We can observe in figure 10 that after 4'500 tournaments, on average, the FMGA's fitness has been maximised, no matter what crossover rate it started with, as long as it is not 0. The crossover rate has a noticeable impact on the fitness of the model before the 4'500 tournament mark has been reached, and it is the crossover probability of 0.5 that seems to reach the solution first, although the difference is not massive. Therefore, we have found that the impact of the crossover rate is small compared to the number of tournaments, even if previous data (see figure 6) has shown that a crossover probability between 0.2 and 0.6 maximises performance.

4 Discussion

In our introduction, we made the hypothesis that more tournaments would increase the fitness of the FMGA, and that a crossover probability of 0.5 would optimise it. Our experimentation have revealed that our first idea was overwhelmingly true, but did hit a plateau before reaching the solution in most cases 4. In the case of the crossover probability, we found its impact on perfomance to not be as significant, as long as its value is not extreme (e.g 0 or 1). We even found in the figure 10, when searching for the optimal number of tournaments and crossover probability, that the number of tournaments is the major decider in the FMGA's fitness.

We can deduce that the number of tournaments gives the population more chances to evolve and, when given enough tournaments, most individuals will have had the chance to increase their fitness.

Furthermore, the crossover rate should increase the search space the algorithm covers. With the mutation rate also increasing the covered search space, the crossover probability needs to be at a minimum of 0.02 for the FMGA to reach the knapsack solution (see figures 6 and 10).

Most importantly about the crossover rate, we explored the assumption it makes about the initial diversity of the population. We found that if the initial diversity of the population is low (genotypes close to only 0s or only 1s), no matter what crossover rate the FMGA is given, the fitness won't increase as rapidly towards the solution than a more diverse initial population (see figure 9).

To improve our experimentation and solidify the significance of our findings, we could have tested the varying hyper-parameters of the FMGA model on different knapsacks problem. The size of our problem meant that selecting all items, as the initial diversity rate of 1 showed in figure 9, prevented the algorithm from getting a better fitness than 0. Therefore, varying knapsacks problem, defined over a scale of the percentage of items needed to solve the problem, could have revealed a different impact of the assumptions the crossover probability makes.

A second way to improve our experimentation would have been by varying other parameters of our FMGA, such as the mutation rate. We often found that the randomness of the mutation helped improve our FMGA model, such as when it had an initial diversity of 0 in figures 8 and 9. If there were no mutation rate, the fitness would have stayed at 0 for the entire population for each tournament as no genotype would have ever won a tournament.

5 Conclusion

We found that, when implementing FMGAs, the number of tournaments is the main defining factor of their fitness, relying on a crossover rate of 0.2, an initially diverse population, and a mutation rate that modifies at least one genotype. Moreover, we found that the crossover rate's impact on the fitness of an FMGA is minimal compared to the number of tournaments, and is only detrimental to the fitness if it does not increase the search space, by replacing all genes of the tournaments' losers or replacing none. We learned that the crossover probability hyper-parameter of FMGAs is dependent on the initial diversity of the population, as it permits a wider coverage of the search space. Overall, we found that diversity is the key to fitter FMGAs. For further experimentation, we could deepen our findings with varying knapsack problems, in terms of the percentage of items needed for the solution, and by modifying more hyper-parameters of the FMGA model, such as the mutation rate.

References

[1] David Freedman, Robert Pisani, and Roger Purves. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.

[2] Inman Harvey. The microbial genetic algorithm. In *ECAL*, 2009.

[3] Melanie Mitchell, Stephanie Forrest, and John H. Holland. The royal road for genetic algorithms: Fitness landscapes and ga performance. 1991.