

REPORT on MDL PROJECT (OUR application of GENETIC ALGORITHMS)

TEAM ARTIFICIAL BRILLIANCE
- ANMOL AGARWAL, RAJ MAHESHWARI

Genetic algorithms are search techniques used in computing to find true or approximate solutions to optimization and search. It is inspired by Charles Darwin’s theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation. It uses techniques inspired by evolutionary biology such as **inheritance, mutation, selection, and crossover**.

Evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are selected from the current population (based on their fitness), and modified to form a new population.

The new population is used in the next iteration of the algorithm.

A fundamental difference between applying GA VS the regression based techniques we used in the previous assignment was the switch from calculus based methods to heuristic based methods and the population consisting of multiple points instead of a single member.

Before we proceed further, we would like to define the terminology we used in the report:

- **Individual** - Any possible solution (in our case, a vector having 11 dimensions)
- **Population** - Group of all individuals of a particular generation
- **Fitness** – Target function that we are optimizing (we keep changing the target function throughout the project depending on what parameter we were trying to optimize that day) and how the genetic algorithm performed with the fitness function
- **Gene**: A particular feature (there are 11 genes in the individual in our case) of the individual
- **Trait** - Possible aspect (features) of an individual
- **Genome** - Collection of all chromosomes (traits) for an individual.
- **Run**: The entire set of generations we ran with the same hyper-parameters

Our intuitive interpretation of the problem statement wrt to the genetic algorithms context was as follows:

- We consider each individual of a population as a set of ideas (notions) [not to be considered same as the genes as a particularly useful idea in a given context can be a relation between two genes g1 and g2 rather than their individual values]
- The action of crossover using notions of the individuals (who have been deemed fit for crossover), thus trying to generate new ideas constructed from high performance building blocks (notions) of the past trials.

High level overview of the steps in our implementation

- Initialize a starting population
- Determine fitness of population (based on a self-defined heuristic)
- Until improvement is observed, repeat:
 - Select parents from the population fit for mating
 - Crossover parents in mating pool to generate children vectors
 - Perform mutation on new population
 - Calculate fitness for new population

Our methodology behind running the queries in brief

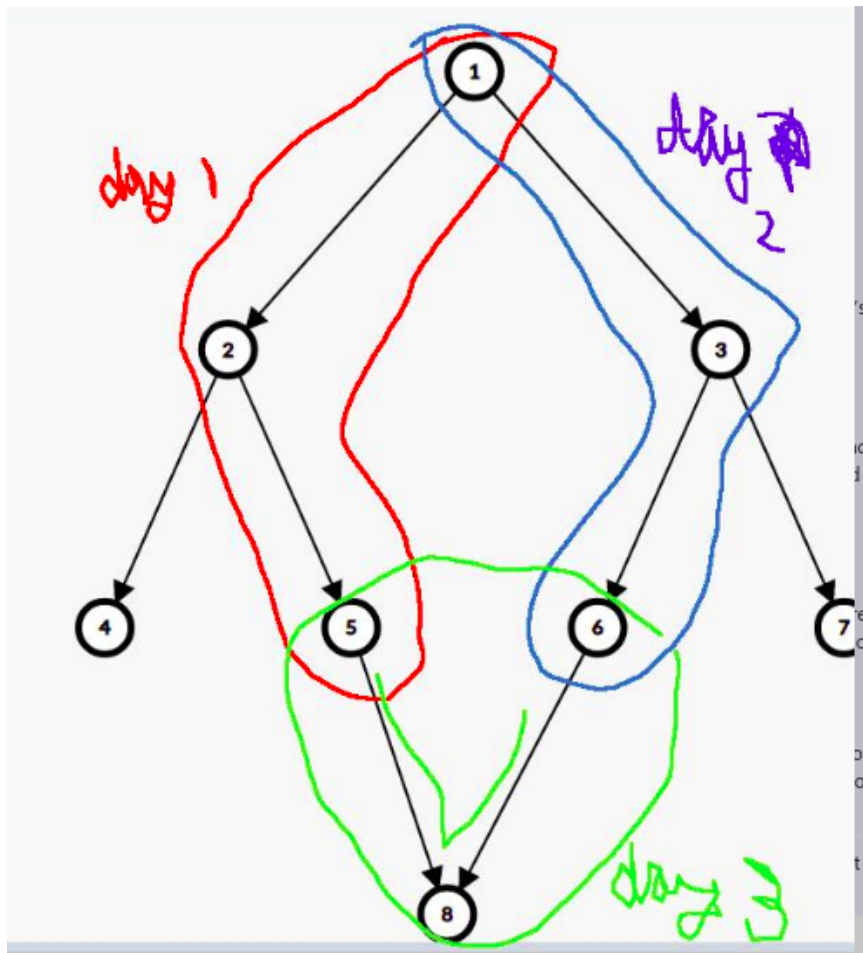
Over the entire span, we tried to experiment with different heuristics, fitness functions and other parameters. We changed metrics like crossover function, mutation function, population size etc in between generations. [So, all generations in generations1.txt would not have the same size.]

On each day, we chose some selected vectors from the previous generations and chose them as the initial population. So, the last generation of day ‘x’ does not coincide with the first generation of day ‘x+1’.

In other words, in this graph, consider node ‘i’ to be part of some ‘ith’ run.

Let v1 be part of the population of node 8. Here, the parents of v1 are not from the same generation but are descendants of the overfit vector (in node 1).

As a result, in the generations.txt's, you will notice that some of our vectors are obtained in generation 3. This implies that this vector was obtained in generation 3 of the run at vertex 8 (on day 3) and the groundwork on this vector began long back in the nodes in the path which lead from node 1 to node 8.



Our methodology for improving performance

The stepwise explanation of our implementation is as follows :

STEP 1: Generation of the initial population

To start the genetic algorithm, we require a population of some **POP_SIZE** number of individuals.

There were two possible options in front of us when it came to deciding the initial population.

- **Random initialization**
- **Heuristic based initialization** (using some vectors which we thought performed well when ranked on the basis of the decided fitness function[heuristic] and using those to generate the initial population)

We thought that if we use random initialization, it is possible that our exploration space is many times better than the case where we used heuristics based initialization.

On running our first few queries of the project by using random initialization, the results were extremely bad (scaling to errors upto 10^{34}). We concluded that despite the exploration space being pretty large, using a random initial population could take several amounts of queries to even come remotely close to the error which the overfit vector produced on the train and validation data sets. With our limited requests per day it did not make sense to randomly initialize.

Learning from the above experience, we decided to use the overfit vector as the initial population for our next set of queries. For most of the future runs of our algorithm, we **picked a set of vectors which were obtained during the previous queries and were known to perform well as per the fitness function chosen by us**. We used this set to be a part of and to generate the other members of the initial population.

To make sure that initial population is diverse, we used to do two things:

1. Apply a good amount of mutation on the vectors which were being used to generate the initial population.
2. Made an attempt to pick vectors which are very different from each other in the first place.

So, typically, if the population size was **POP_SIZE**, then we would pick approximately 4-10 good vectors we had previously obtained and mutated them to generate the remaining members of the initial population.

The mutations would typically vary on 2 factors :

- the number of cells (genes) of the origin vector we were going to mutate AND
- the magnitude by which we were going to change the value of that gene. The magnitude of change allowed was usually (alpha x initial value of the gene) where we used to decide alpha depending on what degree of mutation we wanted to apply.

We have ensured that the first generation of individuals is diverse by keeping the probability of mutation factor high, that is the probability with which a gene is mutated in a chromosome is high (usually 70 % chance of a gene being mutated). This is so that the algorithm does not converge to a local minima in the very beginning itself.

```
CURR_GEN_NUM=1

init_arr = np.array(
[
    [0.1404353012090963, -1.5917637608939356e-12, -1.2461542166753883e-13, 6.945508820550938e-11, -2.83382071568186e-11, 0.07340107154031256, -1.5437615866430792e-12, -1.4641151122621188e-13, 6.645245422169678e-11, -3.7164454801951e-11, 0.03523998918426694, -1.0117081002184981e-12, -1.2994472471499564e-14, 7.893352705113582e-11, -1.1860894152791e-11, 0.10456903003668358, -8.310365465172867e-13, -1.1425053328907633e-14, 9.427139040575664e-11, -1.31322871368850e-11, -0.15312064598796957, -1.2759516587457305e-12, -6.334633169412846e-14, 1.0791176309199842e-10, -4.515511754084e-11, 0.05271366053724377, -1.6029419079568322e-12, -1.1478915535579567e-13, 7.996355167326429e-11, -1.9736060467401e-11]
])

starting_population_mat=create_starting_population(POP_SIZE, NUM_FEATURES, LB_LIMIT, UB_LIMIT)

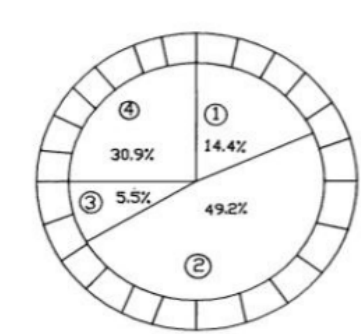
for i in range(0,POP_SIZE):
    if i<6:
        starting_population_mat[i]=get_mutated_version(np.array(init_arr[i%6]), False)
    else:
        starting_population_mat[i]=get_mutated_version(np.array(init_arr[i%6]),True)
```

STEP 2: OBTAIN ERRORS FOR THIS GENERATION

Once the first generation is ready, we obtain the errors of all the individuals (parents for the next generation). This is done to determine the fitness of each individual and to decide their progress into the next generation.

A detailed description of the heuristics and fitness functions used by us and their performance can be found in the latter part of the report.

STEP 3: Determining the parents who would be taking part in reproducing for producing individuals of the next generation and the probability of them doing so



The method we used to select which parents would take part in crossover was the **roulette wheel selection**. In this method, choosing individuals can be depicted as spinning a roulette that has as many pockets as there are individuals in the current generation, with sizes depending on their probability. Over the course of the project, we allotted the fraction of the roulette wheel allotted to a particular individual based on the following metrics:

- **Fitness based Russian Roulette:** This fitness level is used to associate a probability of selection with each individual. If f_i is the fitness value of the individual, then its probability of selection would be [NOTE, we try to minimize error fitness and so we use (-fitness) in the below formula

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j},$$

This method has some **shortcomings** which we encountered:

- Firstly, since there is a chance that **fitness values can be negative**, all fitness values must be **scaled by an appropriate factor**, deciding which is a problem on its own. (Eg: 1, 2,3 are not the same as 5,6,7 even though diff between the values are the same). For our implementation, we considered max(over all fitnesses)-fitness of individuals as the numerator for the individual and sum over all numerators as denominator.

- Secondly, consider the case where the fitness values of 3 individuals are (1, 100, 10000000). In this case, as per our implementation, probability of individual 1 would be $(9999999)/(9999999+9999900)=0.500002$ and probability for second individual would be $(9999900)/(9999999+9999900)=0.499997$. So, **despite individual 1 being 100 times better than individual 2, its probability of being selected is approximately the same as individual 2** as individual 3 is so bad that the difference in quality of individuals 1 and 2 seems insignificant.
- RANK based Russian Roulette:** In order to escape the shortcomings, we switched to rank based russian roulette where we picked rank individuals on basis of fitness value (the best individual having rank 1) and then for an individual having rank 'i', we assign the following probability to it:

$$P_n = \frac{N_{\text{keep}} - n + 1}{\sum_{i=1}^{N_{\text{keep}}} i}$$

Rank-based fitness assignment overcomes the scaling problems of the Fitness based Russian Roulette. The ranking introduced a uniform scaling across the population and provides a simple and effective way of controlling selective pressure.

```
# Probability of getting selected is higher for a higher rank and vice versa
def get_probabilities(n):
    numerator=np.arange(1,n+1)
    numerator*=-1
    denom=(n*(n+1))/2
    numerator+=n+1
    numerator=np.divide(numerator, denom)
    return numerator
```

The spinning of the russian roulette was coded by taking the cumulative sum of the probabilities and choosing the individual in whose bin, a randomly generated number fell.

Due to probabilistic selection of parents, it may happen that the same individual is selected as both parents, in which case the generated offspring will be a copy of the parent. We implemented a check to prevent such unnecessary operations.

```
probs=get_probabilities_russian_roulette(sorted_fitness_vals)
cum_probs=np.cumsum(probs)
cum_probs=np.hstack(([0],cum_probs))
'''Selection of parents who will take part in crossover
'''
pick_1=np.random.uniform(low=0, high=1, size=(matings_pending))
pick_2=np.random.uniform(low=0, high=1, size=(matings_pending))

mother_idx=[]
father_idx=[]
## Making the parents breed
for ic in range(0,matings_pending):
    tot_crosses+=1
    for idx in range(1,POP_SIZE+1):
        if pick_1[ic]<=cum_probs[idx] and pick_1[ic]>cum_probs[idx-1]:
            mother_idx.append(idx-1)
        if pick_2[ic]<=cum_probs[idx] and pick_2[ic]>cum_probs[idx-1]:
            father_idx.append(idx-1)
    # avoid homogeneous crossover
    if mother_idx[-1]==father_idx[-1]:
        father_idx[-1]=((mother_idx[-1])+1)%POP_SIZE
```

STEP 4: CROSSOVER OF PARENT POPULATION

- Once which parents are to mate is decided, the actual mating process takes place. We experimented with several crossover operators and we describe our observations regarding their performance here.

Single point crossover

Initially, we implemented a simple single point crossover where the first parent was copied till a certain index, and the remaining was copied from the second parent. Its disadvantages have been mentioned in the 'swap crossover' section.


```
special_cell_idx=random.randint(0,NUM_FEATURES-1)
beta=random.random()
xy=parents_now[par_one_idx][special_cell_idx]-\
    parents_now[par_two_idx][special_cell_idx]
child_one[special_cell_idx]=parents_now[par_one_idx][special_cell_idx] - beta*xy
child_two[special_cell_idx]=parents_now[par_two_idx][special_cell_idx] + beta*xy
if special_cell_idx!=NUM_FEATURES-1: |

    child_one[:] = np.hstack((child_one[:special_cell_idx] \
        , child_two[special_cell_idx:] ))

    child_two[:] = np.hstack((child_two[:special_cell_idx] \
        , child_one[special_cell_idx:] ))
```

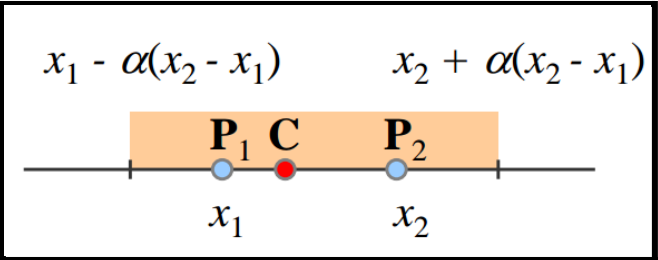
BLEND Crossover

Given the two parent genes x1 and x2 (where x1 < x2) , the blend crossover randomly selects a child gene in the range [x1-α(x2-x1), x2+α(x2-x1)].

```
: def blend_crossover(parent1, parent2):
    ...
    [ x1-α(x2-x1), x2+α(x2-x1)]
    ...
    child1 = np.empty(NUM_FEATURES)
    child2 = np.empty(NUM_FEATURES)
    #print(f"parent 1 is {parent1}")
    #print(f"parent 2 is {parent2}")

    num_genes=len(parent1)
    alpha=0.5

    for i in range(num_genes):
        max_val=max(parent1[i], parent2[i])
        min_val=min(parent1[i], parent2[i])
        lb=min_val-alpha*(max_val-min_val)
        ub=max_val+alpha*(max_val-min_val)
        set_val_1=random.uniform(lb, ub)
        child1[i]=set_val_1
        set_val_2=random.uniform(lb, ub)
        child2[i]=set_val_2 |
```



SBX

The entire idea behind simulated binary crossover is to generate two children from two parents. Here, the value of the offspring gene depends on the parent's genes as follows:

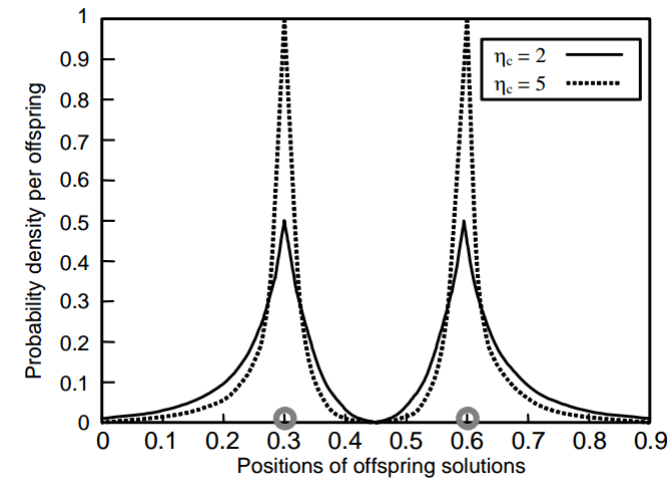
$$x_1^{new} = 0.5[(1 + \beta)x_1 + (1 - \beta)x_2]$$
$$x_2^{new} = 0.5[(1 - \beta)x_1 + (1 + \beta)x_2]$$

Here, Beta is a parameter which is calculated as follows:

$$\beta = \begin{cases} (2u)^{\frac{1}{\eta_c+1}}, & \text{if } u \leq 0.5 \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases}$$

In above, n_c is the distribution index which can be in the range [2,5]. The greater the distribution index, the lesser will be the variation in children as compared to parents. We manipulated the value of n_c based on the extent of exploration we wanted to do in a particular run.

A small simulation to depict the value of the offspring gene depending on distribution index can be seen below:



```
def binary_crossover(parent1, parent2):  
  
    '''  
    Deb and Agrawal [196] developed the simulated binary crossover (SBX) to simulate the  
    behavior of the one-point crossover operator for binary representations.  
    '''  
    child1 = np.empty(NUM_FEATURES)  
    child2 = np.empty(NUM_FEATURES)  
  
    u = random.random()  
    n_c = 2  
    '''where  $r_j \sim U(0, 1)$ , and  $\eta > 0$  is the distribution index. Deb and Agrawal suggested  
    that  $\eta = 1$ .'''  
  
    '''The SBX operator generates offspring symmetrically about the parents, which prevents  
    bias towards any of the parents. For large values of  $\eta$  there is a higher probability that  
    offspring will be created near the parents. For small  $\eta$  values, offspring will be more  
    distant from the parents.'''  
  
    if (u < 0.5):  
        beta = (2 * u)**((n_c + 1)**-1)  
    else:  
        beta = ((2*(1-u))**(-1))**((n_c + 1)**-1)  
  
    parent1 = np.array(parent1)  
    parent2 = np.array(parent2)  
    child1 = 0.5*((1 + beta) * parent1 + (1 - beta) * parent2)  
    child2 = 0.5*((1 - beta) * parent1 + (1 + beta) * parent2)  
  
    for i in range(len(child1)):  
        child1[i]=max(child1[i],LB_LIMIT)  
        child1[i]=min(child1[i],UB_LIMIT)  
  
    for i in range(len(child2)):  
        child2[i]=max(child2[i],LB_LIMIT)  
        child2[i]=min(child2[i],UB_LIMIT)  
  
    return child1.tolist(), child2.tolist()
```

Swap Crossover (designed by us)

We noticed the fact that in all the crossover techniques mentioned above, **either all the genes of two parents intermingle with each other** (SBX, blend crossover) or **always a subset of genes get transferred to the child**. In singlepoint crossover, since from where the child inherits the gene depends on the chosen point, such a case cannot arise where the child inherits genes 1, 4 , 6 from the parent1 and 0,2,3,5,7,8,9 from parent2. We thought **this may be a hindrance as it is possible that the parent1 has its best ideas at features 0 and 11 and child2 has its best features at indexes 4,5 but there is no way that a child inherits 0, 11 from parent1 and 4,5 from parent2**. To take care of this, we implemented a crossover where which indices are to be inherited from a parent are selected randomly in a discrete manner. This helps to **get rid of the role that the permutation ordering** of the genes was playing in other crossovers.

Unsurprisingly, **we got very good results** in terms of the metric we ran this on and we noticed that even though the convergence was slow, there were almost no cases where the child generation had a worse performance than the parent generation.

```
def swap_crossover(parent1, parent2):  
  
    child1 = np.empty(NUM_FEATURES)  
    child2 = np.empty(NUM_FEATURES)  
    num_genes=len(parent1)  
    # Number of distinct random indices at which swapping occurs in the children vectors.  
    num_swap=3  
    swap_indices = random.sample(range(0,num_genes), num_swap)  
  
    for i in range(num_genes):  
        if i in swap_indices:  
            child1[i] = parent1[i]  
            child2[i] = parent2[i]  
        else:  
            child1[i] = parent2[i]  
            child2[i] = parent1[i]  
  
    for i in range(len(child1)):  
        child1[i]=max(child1[i],LB_LIMIT)  
        child1[i]=min(child1[i],UB_LIMIT)  
  
    for i in range(len(child2)):  
        child2[i]=max(child2[i],LB_LIMIT)  
        child2[i]=min(child2[i],UB_LIMIT)
```

STEP 5: Creating the next generation

We retain the few best performing individuals of the previous generation defined by the hyperparameter **RETENTION_RATE** which we varied between the limits [0.2, 0.4]. The parents which are not to be retained were replaced by the newly produced children. The new generation will have a top (RETENTION_RATE x POP_SIZE) number of parents of the previous generation. This is so that the best (best fitness and least error) chromosomes of both populations are advanced to the next generation. The remaining (1-RETENTION_RATE) x POP_SIZE population of the next generation are filled by the children produced in this generation.

STEP 6 : Mutation of the genes of the new generation

The aim of mutation is to introduce new genetic material into an existing individual; that is, to add diversity to the genetic characteristics of the population. Mutation is used in support of crossover to ensure that the full range of allele is accessible for each gene.

The mutation of vectors in our code was decided by two hyperparameters:

- **MUT_RATE** : this parameter determined how many of the genes were to be manipulated in the individuals of a generation. In the generation, total number of genes is (POP_SIZE * NUM_FEATURES) where NUM_FEATURES=11 . Now, we chose any [MUT_RATE x (POP_SIZE * NUM_FEATURES)] of these genes for mutation. As the iterations of the GA proceed, we manipulate this parameter. [In case the minimum fitness does not change with passing generations or if avg fitness approaches minimum fitness, we increase this parameter to promote diversity of individuals in next generation,](#)
- **MUT_MAGNITUDE** : Once a gene is selected, this decides by how much magnitude the gene is allowed to be varied. We make sure that the new value of the gene is in the range [(1-MUT_MAGNITUDE) x old_value , (1 + MUT_MAGNITUDE) x old_value].When we see that the fitness error is decreasing across generations, we infer that the local minima has still not been reached answer keep this mutation magnitude as low. But as soon as min fitness value does not change across generations, [we increase the MUT_MAGNITUDE with the hopes of the new population members being a part of different hills and thus, giving us the chance to explore possibly better local minimas \(or a global minima\).](#)

HYPERPARAMETERS

NUM_FEATURES=11

LB_LIMIT=-10

UB_LIMIT=10

MAX_ITR=24

POP_SIZE=40

MUT_RATE=0.3

RETENTION_RATE=0.2 # selection rate from prev generation

DESC = "[14th March 0.9 x max(t,v) + 0.1 x min(t,v) SBX RUSSIAN ROULETTE]" # For our reference in MongoDB, describes the 'run'

Other crossover specific parameters have already been described in the crossover section

- **POP_SIZE** :Initially, we started with approximately a population size of 10 with the aim of exploring different crossover, mutation operators and to see which one is best (With the queries restricted to 500, we wanted to get more iterations under the belt). The population size of around 10 did not lead to any real progress.
- **MAX_ITR** : We usually kept this around 20 and these iterations were enough for us to judge the performance of the fitness function and heuristic we were using. Our implementation was such that we could terminate before MAX_ITR if we see horrible results or run it for > MAX_ITR iterations of results were decent.
- **MUT_RATE** : this parameter determined how many of the genes were to be manipulated in the individuals of a generation. In the generation, total number of genes is (POP_SIZE * NUM_FEATURES) where NUM_FEATURES=11 . Now, we chose any [MUT_RATE x (POP_SIZE * NUM_FEATURES)] of these genes for mutation. As the iterations of the GA proceed, we manipulate this parameter. [In case the minimum fitness does not change with passing generations or if avg fitness approaches minimum fitness, we increase this parameter to promote diversity of individuals in next generation,](#)
- **RETENTION_RATE**: We retain the few best performing individuals of the previous generation defined by the hyperparameter RETENTION_RATE which we varied between the limits [0.2, 0.4]. The parents which are not to be retained were replaced by the newly produced children. Typically, initially we kept a very low retention rate as we wanted a greater exploration space and did not want past vectors (which were possibly not very good, given that we were in the initial stages). But as our results were improved and we wanted vectors (who performed good on the leaderboard) to mate so that the offspring can inherit useful genes from both, we increased RETENTION_RATE with simultaneous increase in POPULATION_SIZE, with hopes of increasing the probability of a champion vector which inherited useful genes from both parents.

HEURISTICS APPLIED:

While constructing the Genetic Algorithm, the heuristics that we applied include :

1. **Implementing a diversity factor in the fitness value:** In the initial stages of our querying, there was a certain point where no improvement was being made. Despite several mutations, the fitness converged close to the same value with the best vectors being similar as was obtained a day before. In order to get out of this big hill we were stuck in, we decided to penalize the children based on how close they were to previous generations parents (we penalized as $[\gamma] * (\min \text{ euclidean distance between a parent and the child being tested})$). However, we were not able to find a decent ‘gamma’ value to utilize this idea of ours to get good performance.
2. Variations in mutation value **via simulated annealing** (this did not work for us when we implemented this around 9th March when we were not able to improve our results). This might be due to the fact that due to a high mutation rate initially, we would have been able to reach different hills but due to the limit on the number of queries, we weren’t able to reach close to the local minima of these new hills.
3. **Observation based penalty** in the fitness function (see fitness function 4)

Some of the FITNESS FUNCTIONS WE USED

Over the entire span of the project, we experimented with different fitness functions based on our observations of past queries. Some of them worked and some of them did not. We mention some of them here:

Fitness function 1:

Metric: $\text{val} - 0.3 \times \text{train}$

Comments: Even though we tried several fitness functions before this, this was the first time we made significant progress in reducing the validation error. Logically, this made sense as we were aiming not only to reduce the validation error, but also to remove the overfit on the train error (which would have hindered generalization). So, a negative coefficient of the train error in the initial stages of the project made sense.

Series of fitness functions 2:

Metric: $\text{TRAIN_COEFF} \times \text{train_error} + (1 - \text{TRAIN_COEFF}) \times \text{val_error}$ (TRAIN_COEFF was 0.5, 0.7, 0.9)

Comments: Courtesy of function 1, we were able to reduce validation error quite well but as a result, train error overshoot to high values which we intended to moderate. So, we ran this on vectors similar to the overfit vector and the vectors obtained in the run of fitness function 1 with hopes of inheriting valuable features from both set of vectors. However, results were just decent and we made minor progress in reducing train error while not compromising on validation error by a huge margin.

Series of Fitness functions 3:

Metric: $[\text{COEFF_OF_MAX}] * \max(\text{train}, \text{val}) + [1 - \text{COEFF_OF_MAX}] * \min(\text{train}, \text{val})$ where COEFF_OF_MAX was kept as 0.7, 0.8, 0.9, 0.95

Comments: With an attempt to keep both training and validation error under control, we implemented this. Heere, we attempt to keep both training and validation errors in check. Some of our best vectors were obtained by these metrics (even though the groundwork for these vectors was done by training our algorithm on functions 1 and 2).

Fitness function 4:

Metric:
fitness_val=0
t=its_fitness_val[0]/(1e10)
v=its_fitness_val[1]/(1e10)
punish = 0
punish += 100 if t < 22 else 0
punish += 100 if v > 17 else 0
fitness_val = t + v + punish

Comments: We designed this intuitively keeping in mind that we would want to consider both training and validation errors as useful as far as they do not overfit/underfit. So, the moment validation error went above 17 e10 or training error fell below 22e10, we assigned a penalty to them.

Fitness function 5:

```
fitness_val=0
t=its_fitness_val[0]
v=its_fitness_val[1]
punish = 0
punish += 1e11 if (t/v) < 3 else 0
punish += 1e12 if t < 2e11 else 0
punish += 1e12 if v > 1e11 else 0
fitness_val= max(t,v) + punish
```

Comments: We designed this intuitively keeping in mind that we want to keep in check oth train and validation errors while also making sure that the difference in train and validation errors isn’t huge (to make sure that we do not fail to generalize)

We think our vectors are able as some of them have very low errors (of the order of 8e10) on both train and validation set and they also have errors under (100e10) on the test set which was used for leaderboard ranking. Their variations also perform similarly which seems to indicate they haven’t overfitted to specific features of the datasets.

Diagrams of generations

NOTE: If a child has just one parent, that would mean that it is a parent who actually has been retained from the previous generation.

The vectors mapping can be found here if needed :
<https://drive.google.com/drive/folders/1H0MU3zvBQjkgznFgLwVExRx79rDh3pVV?usp=sharing>
They have also been appended at the end of this doc

Diagram 1 (corresponds to vec1.txt):

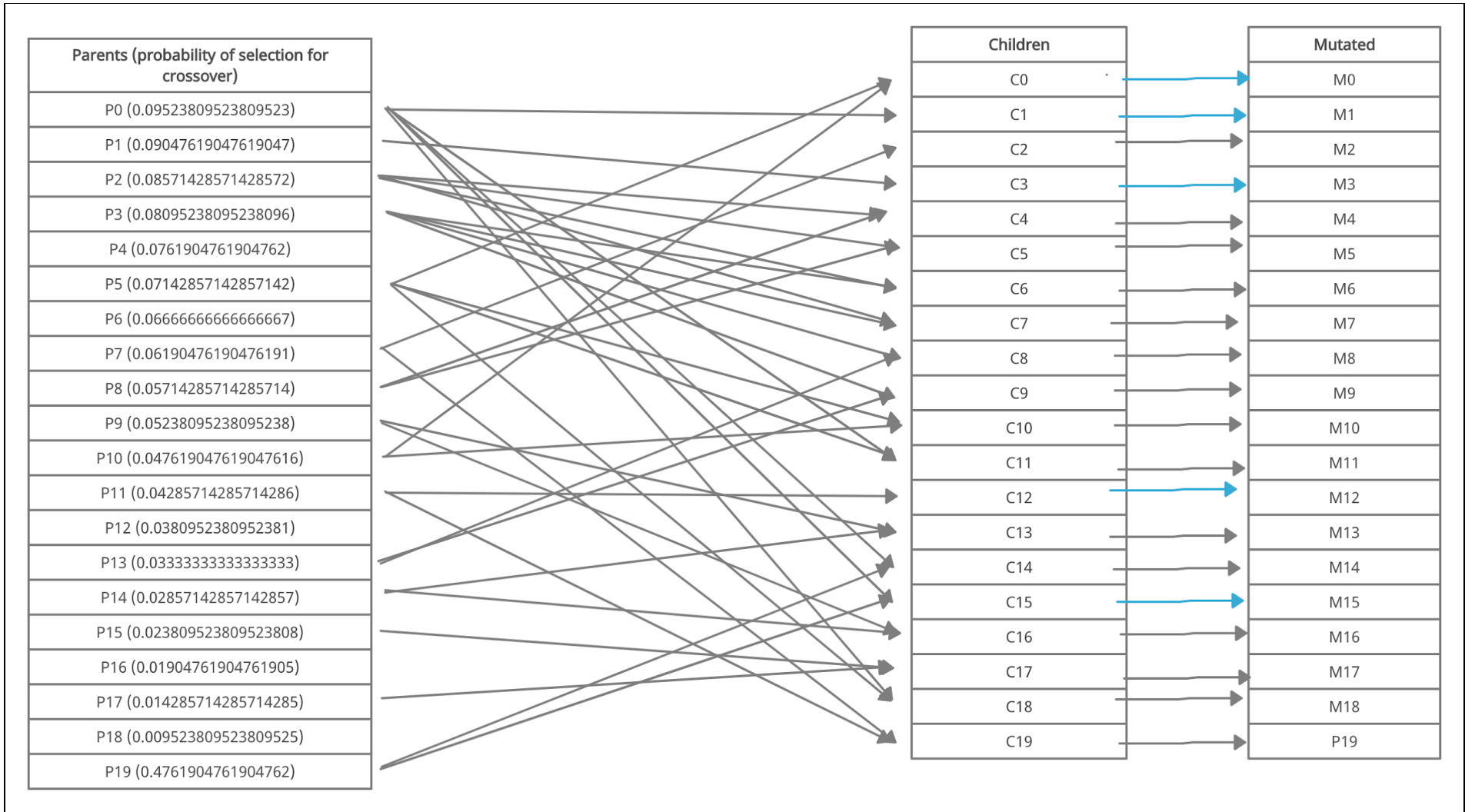


Diagram 2 (corresponds to vec2.txt):

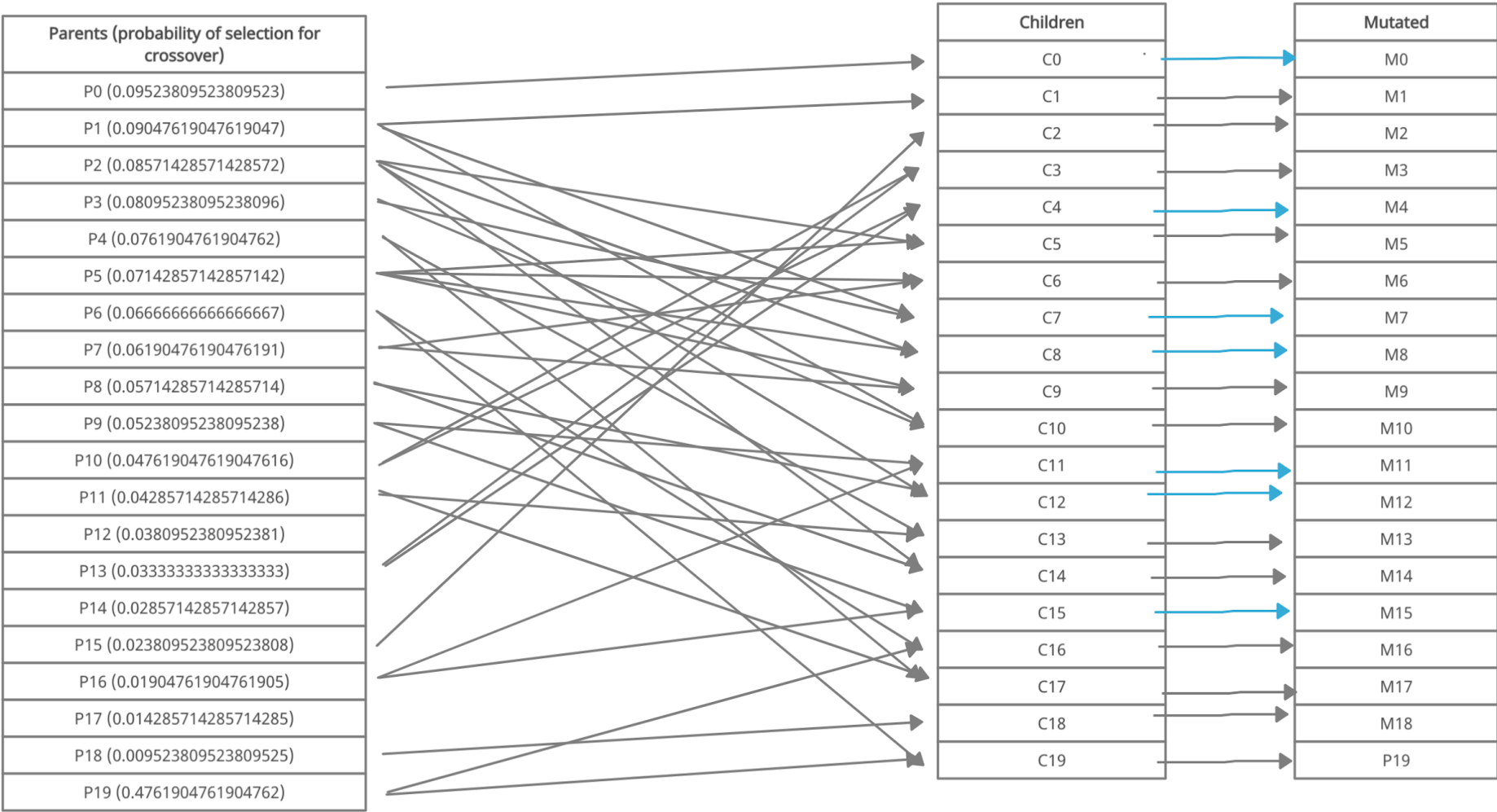
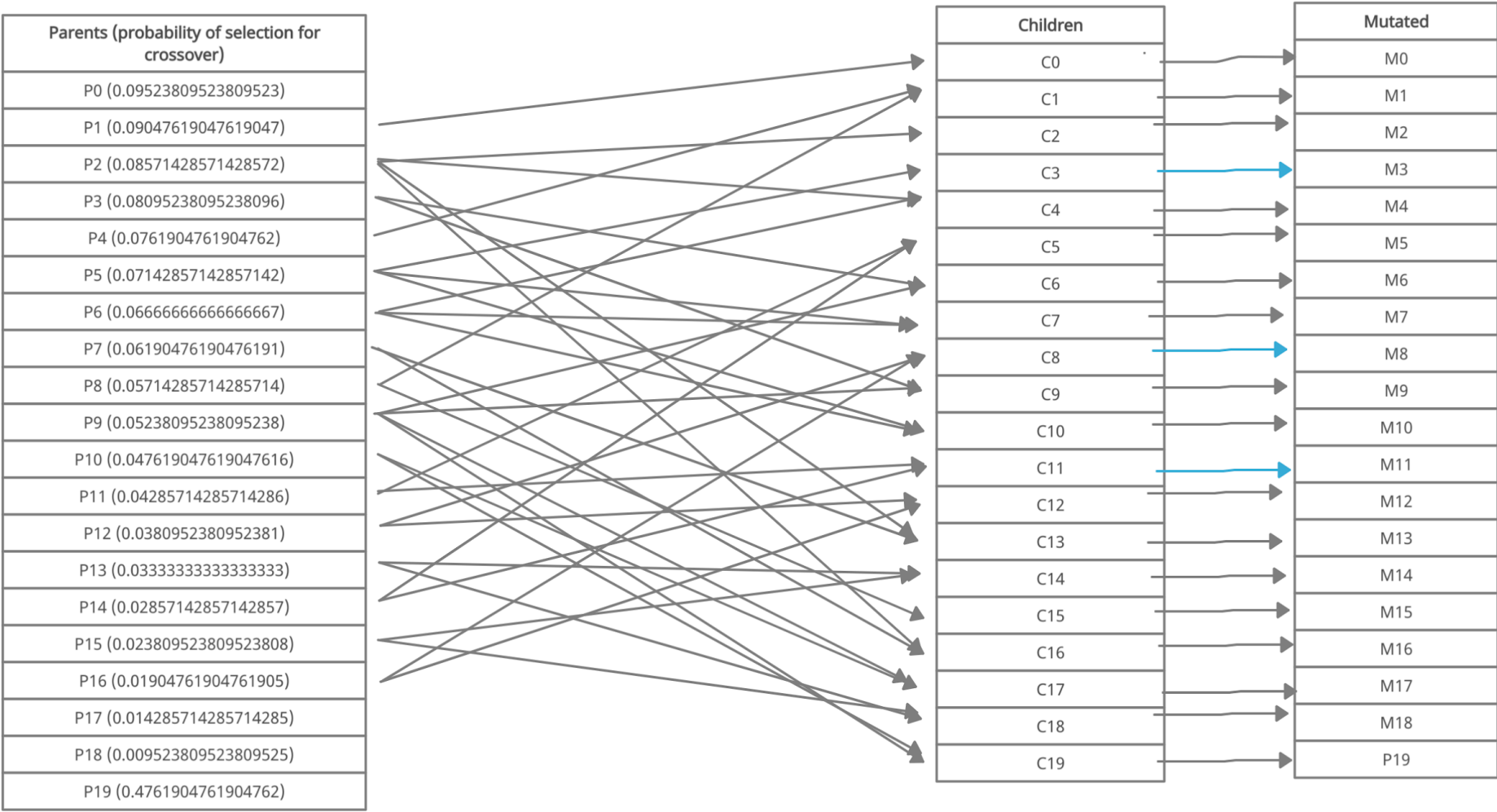


Diagram 3 (corresponds to vec3.txt):



STATISTICAL INFORMATION of the runs made by our algorithm :

The statistical information related to each day’s run can be found in the file ‘records.txt’ (contains details of the runs after 28th Feb only https://docs.google.com/document/d/1tPX1qNchNGkIsrt1NUSff-_pWcawa8RfxrAif7JDb8o/edit#)

ATTACHMENTS IN BELOW PAGES

- 1.Statistical information details
- 2.Diagram mappings

RECORDS (from 1st March onwards)

JSON Name: json_dump_test_1st_MARCH_HASTE.json
ipynb Name:1st_MARCH_generator.ipynb
Date: 1st March
Population Size:20
Initial population description: overfit vector with other pop members with major mutations (max 10% mutation on 70% of the cells) of the overfit vector
Number of best vectors being retained for the next gen: 40 %
Total number of iterations: 23
Fitness function: 1 * val

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: **Single point crossover** (single point position varying)
Crossover parameters if any: Beta (random)

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5%
Was mutation magnitude changed midway ? : NO

Percentage of cells being mutated: 40 %
Was percentage of cells being mutated change midway: NO

Additional comments:

JSON Name: json_dump_test_3rd_MAR_2_actual_.json
ipynb Name:mongo_version_3rd_MARCH_STINT_2.ipynb
Date: 3rd March
DESC: (3rd Mar)
Population Size: 16
Initial population description: Mutations on vector [0.001790041296876042, -1.6089961978533025e-12, -2.171969429718414e-13, 4.492670091954504e-11, -1.822488022721303e-10, -1.762298196359839e-15, 8.458516059706894e-16, 2.4912898505852607e-05, -2.0303461216465176e-06, -1.7407315793329834e-08, 9.943400595965834e-10]) --- (Gen - 18, Desc- 3rd Mar)

Number of best vectors being retained for the next gen: 20%
Total number of iterations: 13
Fitness function: 0.7 x train + 0.3 * val

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: **Single point crossover** (single point position varying)
Crossover parameters if any: Beta (random)

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 0.05
Was mutation magnitude changed midway ? : NO

Percentage of cells being mutated: 20%
Was percentage of cells being mutated change midway: NO

Additional comments:

JSON Name: json_dump_test_4rd_MAR_1_stint_1.json
ipynb Name:4_March_mongo_version.ipynb
Date: 4 March
DESC: (4th Mar - (-0.3*[0]*1*[1])
Population Size: 16
Initial population description:

Based on the vectors after doing the analysis earlier (ratio, negative correlation etc), used variations of
[0.0, -1.378350060791644e-12, -5.674101131657241e-14, 7.145806989243585e-11, -2.0984549655895966e-10, -1.7517579879162334e-15, 8.060220128248057e-16, 2.2757658578306265e-05, -1.989031728863744e-06, -8.908335805469784e-09, 8.161584966021705e-10],
[0.030191655016508116, -1.370451290585018e-12, 1.944684541618345e-14, 8.210907153265034e-11, -2.2255497577166302e-10, -1.713147300890716e-15, 7.847853460987011e-16, 2.174237713450648e-05, -2.0639270090069506e-06, -5.206637844452329e-09, 7.75711723530254e-10],
[0.017657459837896357, -4.99161554821596e-13, -6.579833572841626e-14, 8.593564659863294e-11, -2.152545905225037e-10, -1.835959268234937e-15, 1.6505293587633143e-15, 2.5823340077958138e-05, -2.0197328670331904e-06, -8.971113056810534e-09, 8.029367309649372e-10],
[0.0, -1.378350060791644e-12, -5.674101131657241e-14, 7.145806989243585e-11, -2.0984549655895966e-10, -1.7517579879162334e-15, 8.060220128248057e-16, 2.2757658578306262e-05, -2.032119548051349e-06, -8.238672968960653e-09, 8.161584966021705e-10]
]

Vec.no.	gen	date
1	11	(2nd Mar morning gen 1 - 10)
2	2	(2nd Mar night STINT 2)
3	9	(2nd Mar night STINT 2)
4	2	(2nd Mar night STINT 2)

Number of best vectors being retained for the next gen: 25%
Total number of iterations: 8
Fitness function: val - 0.3 x train

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: **Single point crossover** (single point position varying)
Crossover parameters if any: Beta (random)

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 0.05
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 20%
Was percentage of cells being mutated change midway: NO

Additional comments:
Did analysis on .ipynb and discovered conundrums
Vector which has best val error currently (also improves somewhat on train error)
Validation error has increased by a factor of 10 when compared to overfit vector

```
Out[22]:  _id          603fd1f61f8eb933e69221d3
         generation          3
         try_vector  [0.01946769005157447, -1.656302165775003e-12, ...
         errors      [128932193642.09692, 97398606458.87451]
         desc        (3rd Mar)
         train_err    1.28932e+11
         val_err      9.73986e+10
         Name: 0, dtype: object
```

Best train error (very minor decrease from overfit train error)

```
In [26]: df.iloc[0]

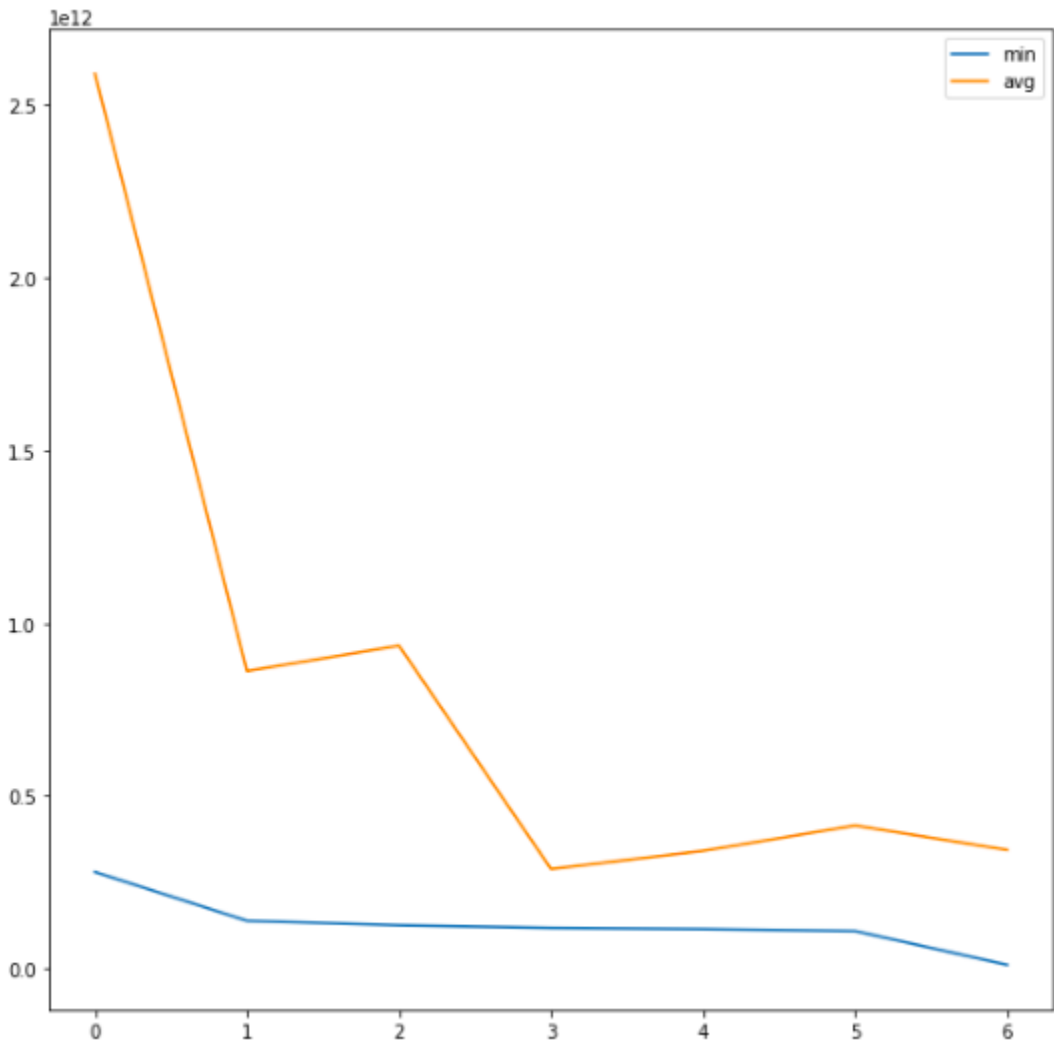
Out[26]:  _id          603e7ac29e229c38b7b5f877
         generation          1
         try_vector  [0.0, -10.0, -10.0, 0.01, 0.0, -1.836698e-15, ...
         errors      [13080071016.835262, 363667169226.52924]
         desc        (2nd Mar night)
         train_err    1.30801e+10
         val_err      3.63667e+11
         Name: 0, dtype: object
```

Has 0.1 * train error + val error

```
In [34]: df.iloc[0]

Out[34]:  _id          603fd1f61f8eb933e69221d3
         generation          3
         try_vector  [0.01946769005157447, -1.656302165775003e-12, ...
         errors      [128932193642.09692, 97398606458.87451]
         desc        (3rd Mar)
         train_err    1.28932e+11
         val_err      9.73986e+10
         Name: 0, dtype: object
```

After analysis, we used our first few 100 requests on weird fitness func -0.1 * train + test



Why did we use this fitness function?
It became apparent to us that since the vector given to us is overfit on the training error, we need to increase the training error (in order to have better generalisation) along with reducing the validation error. More weightage was given to reduction of validation error than increasing training error.

JSON Name: json_dump_test_4rd_MAR_NIGHT_STINT_2.json
ipynb Name:4_March_mongo_version.ipynb
Date: 4th March
DESC: (4th Mar - night 11 pm 0.3 * min, 0.7 MAX
Population Size:28
Initial population description: [0.0, -1.3405312314069806e-12, -6.729218801722296e-14, 6.125174639747928e-11, -2.0984549655895666e-10, -1.7579633528138736e-15, 7.517837850786192e-16, 2.2757658578306265e-05, -1.8625864135343682e-06, -1.080532571374171e-08, 8.144913813202241e-10], [0.0, -9.875299804205226e-13, -5.3944821589819837e-14, 7.930563099592005e-11, -2.0984549655895666e-10, -1.4378337510064146e-15, 8.060220128248057e-16, 2.2757658578306265e-05, -1.989031728863744e-06, -8.908335865469784e-09, 8.161584966021705e-10], [0.0,-1.3783500660791644e-12, -6.424818573634483e-14, 6.580019545565703e-11, -2.0984549655895666e-10, -1.6152790712010518e-15, 7.763006729384186e-16, 2.2757658578306265e-05, -1.9193275505106666e-06, -8.894106807593951e-09, 7.758133607867065e-10],

[0.0,-1.436339392655965e-12,-5.867422008035394e-14,7.485339950833092e-11,-2.0984549655895666e-10,-1.781258849184778e-15,8.019238628879107e-16,2.2757658578306265e-05,-1.989031728863744e-06,-8.908335865469784e-09,8.161584966021705e-10],

[0.0,
-1.457990220064754e-12,
-2.2898007842769645e-13,
4.620107525277624e-11,
-1.7521481289918844e-10,
-1.8366976965696096e-15,
8.529440604118815e-16,
2.2942330256117977e-05,
-2.0472100298772093e-06,
-1.597928341587757e-08,
9.982140340891233e-10]

with major mutations (max 50% mutation on 70% of the cells) of the overfit vector

- 8 (4th Mar - (-0.3*[0]*1*[1]))
- 1 (4th Mar - (-0.3*[0]*1*[1]))
- 6 (4th Mar - (-0.3*[0]*1*[1]))
- 1 PROPER 5th March, STINT_2, Sorted on 0.95 x max + 0.05 x min |
- 5 (4th Mar - (-0.3*[0]*1*[1]))
- 1 (4th Mar - night 11 pm 0.2 * min, 0.8 MAX

vec_no	gen	date
1	8	(4th Mar - (-0.3*[0]*1*[1]))
2	1	(4th Mar - (-0.3*[0]*1*[1]))
3	6	(4th Mar - (-0.3*[0]*1*[1]))
4	5	(4th Mar - (-0.3*[0]*1*[1]))
5	1	(4th Mar - night 11 pm 0.2 * min, 0.8 MAX

Number of best vectors being retained for the next gen: 25 %
Total number of iterations: 9
Fitness function: 0.3min + 0.7 max

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: Single point crossover (single point position varying)
Crossover parameters if any: Beta (random)

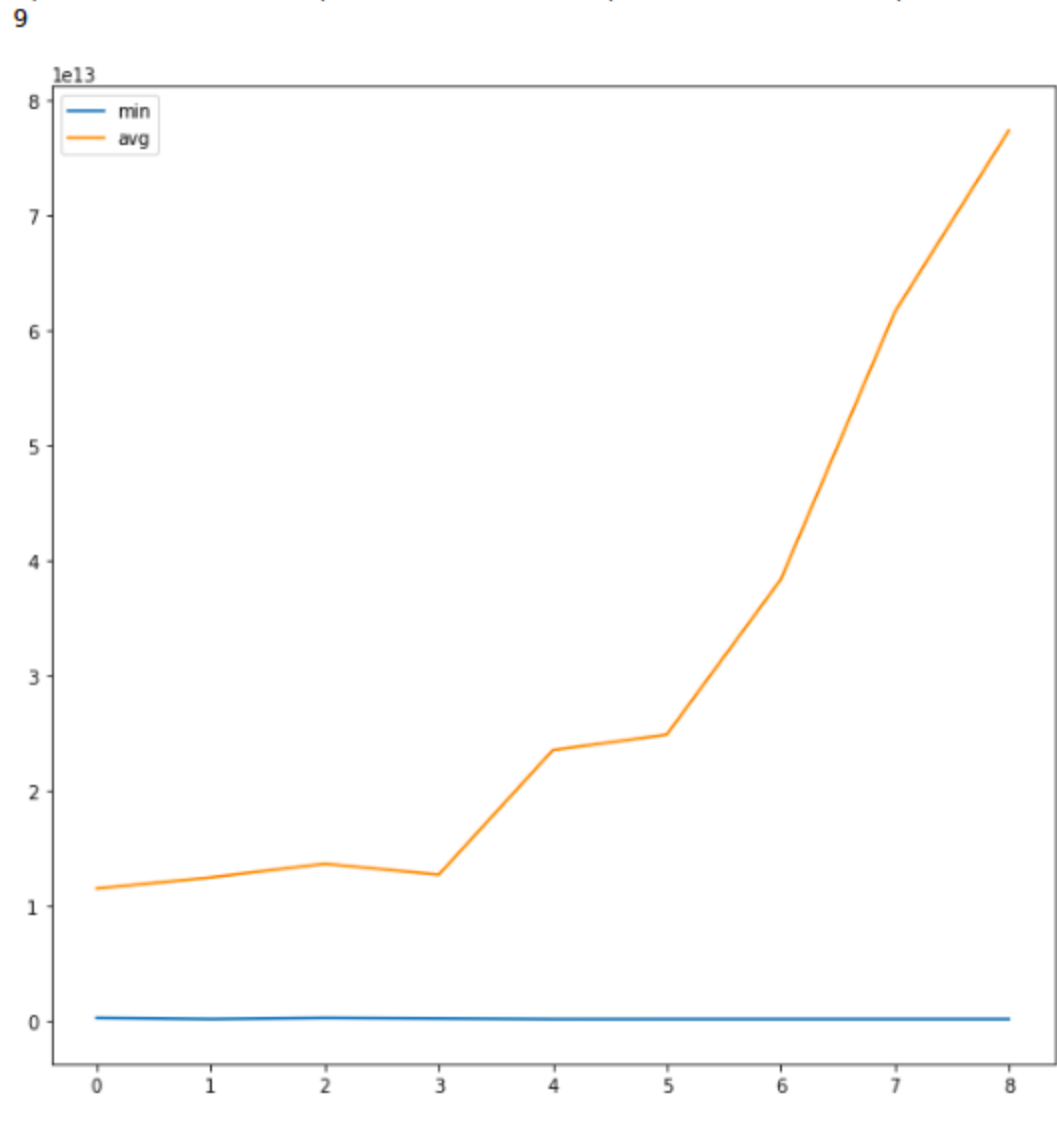
Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5%
Was mutation magnitude changed midway ? : NO

Percentage of cells being mutated: 40 %
Was percentage of cells being mutated change midway: YES

Additional comments:

0, 1757175075712, 1012500074713, 00007100201213, 000010700021-



JSON Name: json_dump_test_4rd_MAR_NIGHT_STINT_2.json
ipynb Name:----
Date: 4th March
DESC: (4th Mar - night 11 pm 0.3 * min, 0.7 MAX
Population Size: 20

Initial population description: -----

Number of best vectors being retained for the next gen: 25 %
Total number of iterations: 19
Fitness function: 0.2min + 0.8 max

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: **Single point crossover** (single point position varying)
Crossover parameters if any: Beta (random)

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5% to 60 %
Was mutation magnitude changed midway ? :YES

Percentage of cells being mutated: 40 %
Was percentage of cells being mutated change midway: YES

Additional comments:

5th MARCH

JSON Name: json_dump_test_5th_MARCH_STINT_1.json
ipynb Name:5_March_mongo_version_backup_for_stint_1.pdf
Date: 5th March
Desc: "5th March, STINT_1, Sorted on 0.9 x max + 0.1 x min"
Population Size:20

Initial population description:

Number of best vectors being retained for the next gen:25 %
Total number of iterations: 14
Fitness function: Sorted on 0.9 x max + 0.1 x min

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: **Single point crossover** (single point position varying)
Crossover parameters if any: Beta (random)

Mutation function: Naive mutation in neighbourhood

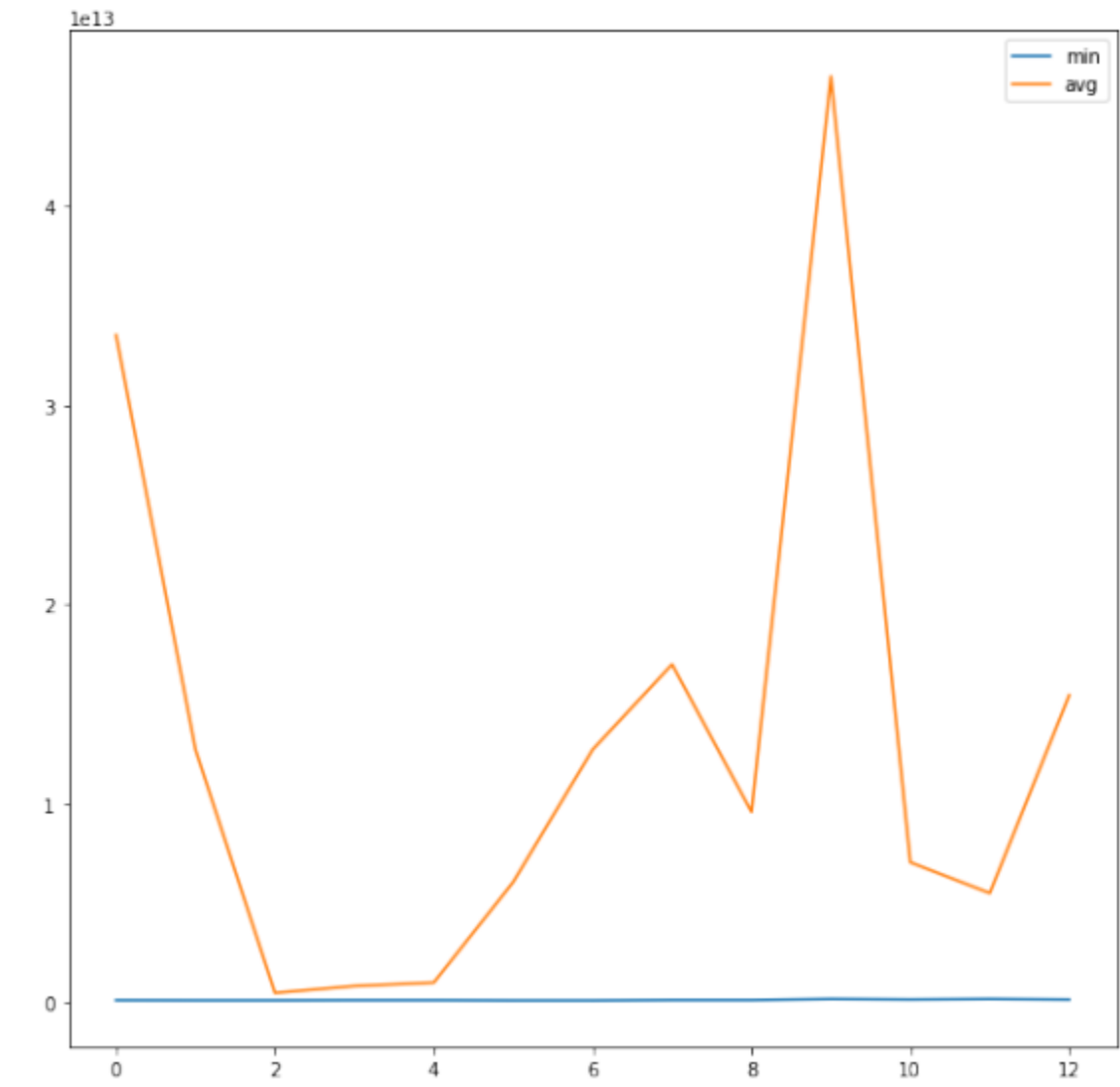
Mutation magnitude: 5%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 25%
Was percentage of cells being mutated change midway: NO

Additional comments:
Details can be found in respective PDFs also from here:
https://gitlab.com/anmolagarwal999/ml-project-genetic/-/tree/trial/5th_March

We selected our initial population on basis of 2 criteria:
(0.1 x min + 0.9 x max) and (0.5 x train + 0.5 x max)
Here, is the initial population: https://gitlab.com/anmolagarwal999/ml-project-genetic/-/blob/trial/5th_March/initial_population_for_stint_1.txt

We trained on basis of **0.9 x max + 0.1 x min**
The intention was that instead of overfitting on one of the 2 datasets, we overfit a little bit on both.



JSON Name: json_dump_test_5th_MARCH_STINT_2.json
ipynb Name:5_March_mongo_version_backup_for_stint_2.pdf
Date: 5th March
Desc: "PROPER 5th March, STINT_2, Sorted on 0.95 x max + 0.05 x min | "
Population Size:26

Initial population description: [
[0.0, -1.3783500660791644e-12, -6.424818573634483e-14, 6.580019545565703e-11, -2.0984549655895666e-10, -1.6152790712010518e-15, 7.763006729384186e-16, 2.2757658578306265e-05, -1.9193275505106666e-06, -8.894106807593951e-09, 7.758133607867065e-10],

[0.0, -1.457990220064754e-12, -9.648767410168089e-14, 7.485339950833092e-11, -1.9881004961860312e-10, -1.5962350118384506e-15, 7.521932081253182e-16, 2.3668681206269528e-05, -1.989031728863744e-06, -7.991914261771e-09, 7.860139677568573e-10],

[-0.030191655016508116, -1.370451290585018e-12, 1.944684541618345e-14, 8.210907153265034e-11, -2.2255497577166302e-10, -1.7133147300890716e-15, 7.847853460987011e-16, 2.174237713450648e-05, -2.0639270090069506e-06, -5.206637844452329e-09, 7.75711723530254e-10],

[-0.016347746048917644, -1.3387827712968235e-12, -6.541567279446175e-14, 8.651454674726037e-11, -2.1494946650019876e-10, -1.8360756358315474e-15, 1.1093211039965001e-15, 2.515778146953262e-05, -2.015649193094959e-06, -8.786550389563894e-09, 8.047779335648933e-10]

Number of best vectors being retained for the next gen:25 %
Total number of iterations: 11
Fitness function: 0.95 max + 0.05 min

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: SBX
Crossover parameters if any: u=3

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 25%
Was percentage of cells being mutated change midway: NO

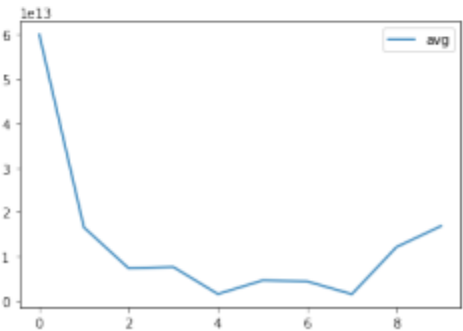
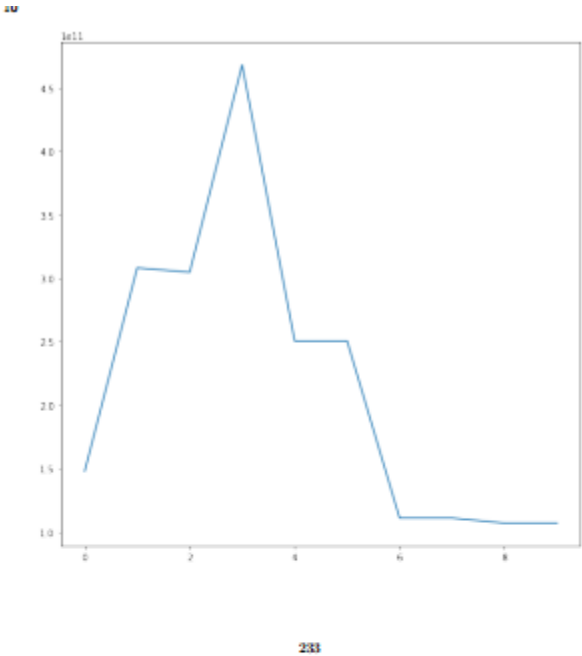
Additional comments:
Next, on this population we trained on the factor **0.95 x max + 0.05 min**.
Due to paucity of time, our 260 requests were left sadly. But the vectors obtained were promising.

```
[17]: def crossover(parent1, parent2):  
  
    '''  
    Deb and Agrawal [196] developed the simulated binary crossover (SBX) to  
    ↪simulate the  
    behavior of the one-point crossover operator for binary representations.  
    '''  
    child1 = np.empty(NUM_FEATURES)  
    child2 = np.empty(NUM_FEATURES)  
    print(f"parent 1 is {parent1}")  
    print(f"parent 2 is {parent2}")  
  
    u = random.random()  
    n_c = 3  
    '''where  $r_j \sim U(0, 1)$ , and  $\eta > 0$  is the distribution index. Deb and Agrawal  
    ↪suggested  
    that  $\eta = 1$ .'''  
  
    '''The SBX operator generates offspring symmetrically about the parents,  
    ↪which prevents  
    bias towards any of the parents. For large values of  $\eta$  there is a higher  
    ↪probability that  
    offspring will be created near the parents. For small  $\eta$  values, offspring  
    ↪will be more  
    distant from the parents.'''  
  
    if (u < 0.5):  
        beta = (2 * u)**((n_c + 1)**-1)  
    else:
```

Our crossover was this

This was the initial population:

```
[  
[0.0, -1.3783500660791644e-12, -6.424818573634483e-14, 6.580019545565703e-11, -2.0984549655895666e-10, -1.6152790712010518e-15, 7.763006729384186e-16,  
2.2757658578306265e-05, -1.9193275505106666e-06, -8.894106807593951e-09, 7.758133607867065e-10],  
  
[0.0, -1.457990220064754e-12, -9.648767410168089e-14, 7.485339950833092e-11, -1.9881004961860312e-10, -1.5962350118384506e-15, 7.521932081253182e-16,  
2.3668681206269528e-05, -1.989031728863744e-06, -7.991914261771e-09, 7.860139677568573e-10],  
  
[-0.030191655016508116, -1.370451290585018e-12, 1.944684541618345e-14, 8.210907153265034e-11, -2.2255497577166302e-10, -1.7133147300890716e-15,  
7.847853460987011e-16, 2.174237713450648e-05, -2.0639270090069506e-06, -5.206637844452329e-09, 7.75711723530254e-10],  
  
[-0.016347746048917644, -1.3387827712968235e-12, -6.541567279446175e-14, 8.651454674726037e-11, -2.1494946650019876e-10, -1.8360756358315474e-15,  
1.1093211039965001e-15, 2.515778146953262e-05, -2.015649193094959e-06, -8.786550389563894e-09, 8.047779335648933e-10],
```



6th March

Details in git repo

Stint 1 : blend_crossover on population sorted based on 0.95 max + 0.05 min

Stint 2 : SBX crossover with u=3 on population sorted based on 0.95 max + 0.05 min

Stint 3 : SBX crossover with u=2 on population sorted based on MAX of train, val

JSON Name: json_dump_test_6th_MARCH_STINT_1.json

ipynb Name:6_March_mongo_version STINT 1.pdf

DESC: "6th March STINT_1 in 0.95 x max +0.05 x min "

Date: 6th March

Population Size:20

Initial population description:

Number of best vectors being retained for the next gen: 25 %

Total number of iterations: 16

Fitness function: 0.95 max + 0.05 min

Selection for Crossover Criteria: Rank-based Russian Roulette

Crossover function: BLEND

Crossover parameters if any: alpha=0.5

Mutation function: Naive mutation in neighbourhood

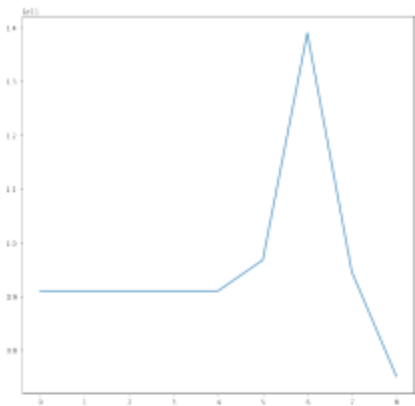
Mutation magnitude: 5%

Was mutation magnitude changed midway ? : YES

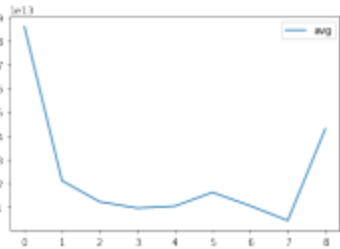
Percentage of cells being mutated: 10 %

Was percentage of cells being mutated change midway: NO

Additional comments:



1.00



JSON Name: json_dump_test_6th_MARCH_STINT_2.json

ipynb Name:6_March_mongo_version STINT 2.pdf

DESC: "PROPER 6th March STINT_2 in 0.95 x max +0.05 x min "

Date: 6th March

Population Size:26

Initial population description: 11 worthy members

Number of best vectors being retained for the next gen: 25 %

Total number of iterations:12

Fitness function: 0.95 max + 0.05 min

Selection for Crossover Criteria: Rank-based Russian Roulette

Crossover function: SBX crossover with u=3

Crossover parameters if any: u=3

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5%

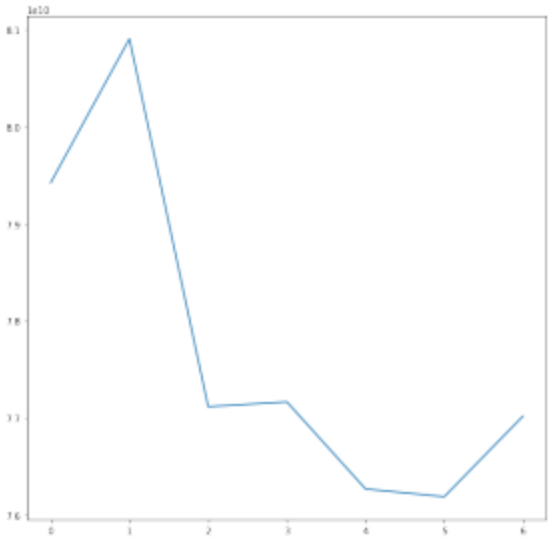
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 20 %

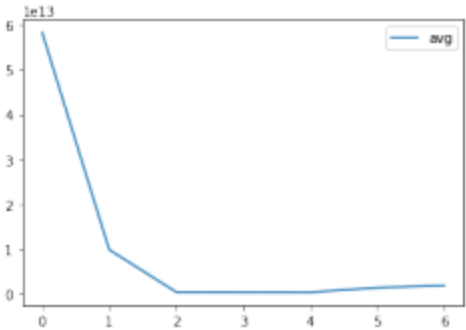
Was percentage of cells being mutated change midway: NO

Additional comments:

3018626557710.867, 4177662323769.0093, 6856798424787.591, 28412248524107.355]]]



136



JSON Name: json_dump_test_6th_MARCH_STINT_3.json
ipynb Name:6_March_mongo_version STINT 3.pdf
DESC: "PROPER 6th March STINT_3 in MAX "
Date: 6th March
Population Size:24
Initial population description: 6 worthy members

Number of best vectors being retained for the next gen: 25 %
Total number of iterations: 9
Fitness function:max(train, val)

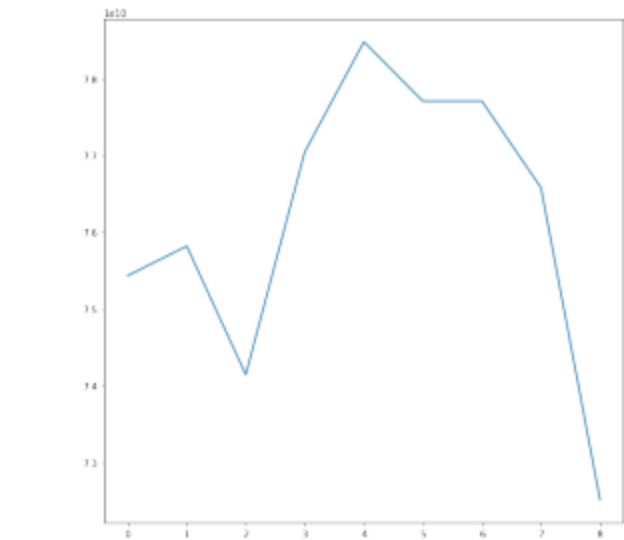
Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: SBX crossover with u=2
Crossover parameters if any: u=2

Mutation function: Naive mutation in neighbourhood

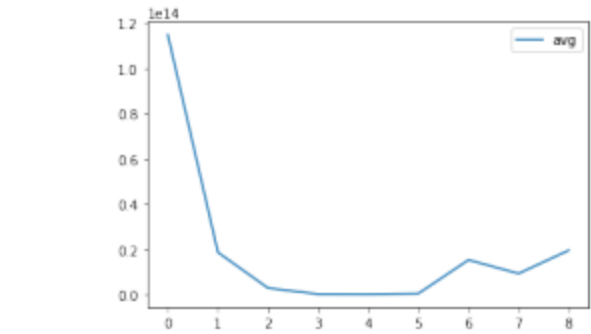
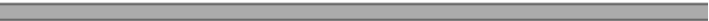
Mutation magnitude: 5%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 15 %
Was percentage of cells being mutated change midway: NO

Additional comments:



174



7th MARCH

JSON Name: json_dump_test_7th_MARCH_STINT_1.json
ipynb Name:7_March_mongo_version_STINT_1.pdf
DESC: "7th March, STINT_1 for outrageous penalty penalisation"
Date: 7th March
Population Size:20
Initial population description: 5 worthy members
[
0.07144470438737868, -1.024341258927964e-12, -2.234254875776415e-14, 7.
,!518054647321033e-11, -1.801621324121818e-10, -1.3903926150219416e-15, 1.
,!0363630046234666e-15, 1.972934634335282e-05, -1.8376799350638052e-06, -7.
,!682037245173995e-09, 7.433548643729261e-10],
[0.04141639993961189, -1.0117081002184981e-12, -1.2994472471499564e-14, 7.
,!893352705113582e-11, -1.604137146808921e-10, -1.485052578582483e-15, 1.
,!0198774026946e-15, 2.0174164573336113e-05, -1.8350397232297526e-06, -7.
,!936364861853528e-09, 7.44044375654244e-10],
[0.05658892175271817, -1.0887038992714193e-12, -8.217466945173303e-14, 7.
,!42364385310126e-11, -1.549785938631631e-10, -1.3658858675157026e-15, 9.
,!474638055926674e-16, 2.049206132808684e-05, -1.8322094202807788e-06, -8.
,!992400344193647e-09, 7.612678224363422e-10],
[0.06660857981986575, -1.0485050121958876e-12, -5.1848667969262964e-14, 7.
,!383933420226818e-11, -1.7053996301357693e-10, -1.3517735508065022e-15, 9.
,!940306295422784e-16, 2.0328653526252607e-05, -1.8372934741650584e-06, -8.
,!424049358979012e-09, 7.532416052416367e-10],
[0.04108911210817906, -1.015583657495513e-12, -1.803672317884344e-14, 7.
,!889503608797325e-11, -1.5928343695820595e-10, -1.4800140859826129e-15, 1.
,!0141756787398243e-15, 2.0253219398914574e-05, -1.8438561327335002e-06, -8.
,!040079184305093e-09, 7.454157721498462e-10]
]

Number of best vectors being retained for the next gen: 25 %
Total number of iterations: 11
Fitness function:

```
fitness_val=0
t=its_fitness_val[0]
v=its_fitness_val[1]
punish = 0
punish += 1e11 if (t/v) < 3 else 0
```



```
punish += 1e12 if t < 2e11 else 0
punish += 1e12 if v > 1e11 else 0
fitness_val= max(t,v) + punish
fitness_arr.append(fitness_val)
```

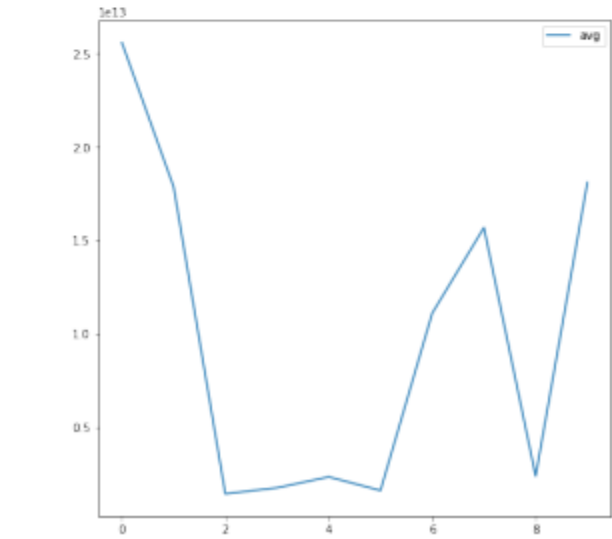
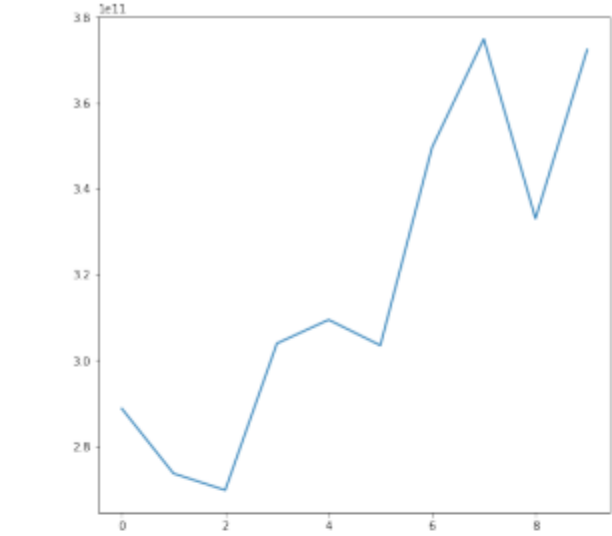
Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: SBX crossover with u=3
Crossover parameters if any: u=3

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 30 %
Was percentage of cells being mutated change midway: YES

Additional comments:



DESC: "7th March, STINT_2 for 0.9 max + 0.1 min"

Date: 7th March

Population Size:20

Initial population description: 3 worthy members

[
0.07144470438737868, -1.024341258927964e-12, -2.234254875776415e-14, 7.
,!518054647321033e-11, -1.801621324121818e-10, -1.3903926150219416e-15, 1.
,!0363630046234666e-15, 1.972934634335282e-05, -1.8376799350638052e-06, -7.
,!682037245173995e-09, 7.433548643729261e-10],
[0.04141639993961189, -1.0117081002184981e-12, -1.2994472471499564e-14, 7.
,!893352705113582e-11, -1.604137146808921e-10, -1.485052578582483e-15, 1.
,!0198774026946e-15, 2.0174164573336113e-05, -1.8350397232297526e-06, -7.
,!936364861853528e-09, 7.44044375654244e-10],
[-0.12826988257084007, -1.3783500660791644e-12, -6.618301505877671e-14, 4.
,!627780406755569e-11, -1.7964979058119056e-10, -1.3330380203323678e-15, 7.
,!763006729384186e-16, 2.2757658578306265e-05, -1.9193275505106666e-06, -8.
,!894106807593951e-09, 7.758133607867065e-10]
]

Number of best vectors being retained for the next gen: 25 %

Total number of iterations: 11

Fitness function: 0.9 max + 0.1 min

Selection for Crossover Criteria: Rank-based Russian Roulette

Crossover function: SBX crossover with u=3

Crossover parameters if any: u=3

Mutation function: Naive mutation in neighbourhood

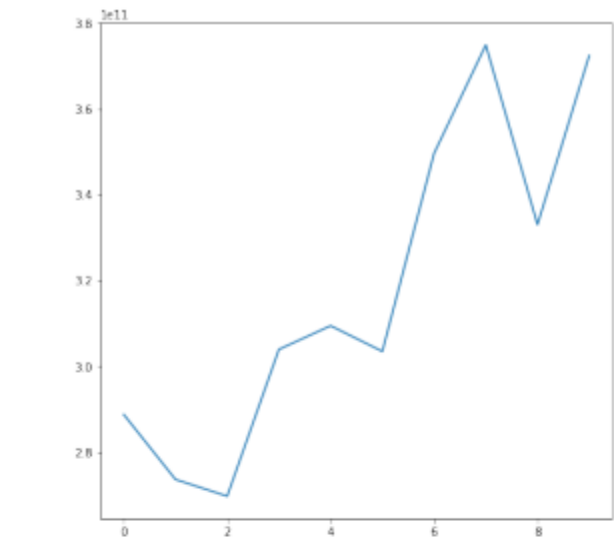
Mutation magnitude: 5%

Was mutation magnitude changed midway ? : YES

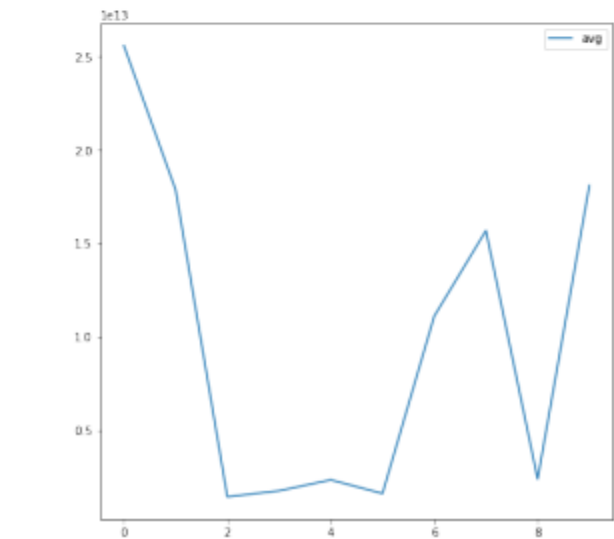
Percentage of cells being mutated: 30 %

Was percentage of cells being mutated change midway: YES

Additional comments:



173



JSON Name: json_dump_test_7th_MARCH_STINT_2.json

ipynb Name:7_March_mongo_version_STINT_2.pdf

DESC: "7th March, STINT_1 for outrageous penalty penalisation"

Date: 7th March

Population Size:20

Initial population description: 6 worthy members

Number of best vectors being retained for the next gen: 25 %
Total number of iterations: 11
Fitness function:

```
fitness_val=0
t=its_fitness_val[0]
v=its_fitness_val[1]
punish = 0
punish += 1e11 if (t/v) < 3 else 0
```

4



```
punish += 1e12 if t < 2e11 else 0
punish += 1e12 if v > 1e11 else 0
fitness_val= max(t,v) + punish
fitness_arr.append(fitness_val)
```

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: SBX crossover with u=3
Crossover parameters if any: u=3

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 30 %
Was percentage of cells being mutated change midway: YES

Additional comments:

JSON Name: json_dump_test_7th_MARCH_STINT_3.json
ipynb Name:7_March_mongo_version_STINT_3.pdf
DESC: "PROPER 3nd 7th March, STINT_3 for 0.95 max + 0.05 min
Date: 7th March
Population Size:36
Initial population description: 4 worthy members

```
[
  0.0,
  -9.875299804205226e+13,
  -5.3944821589819837e-14,
  7.930503099592005e-11,
  -2.0984549658959506e-10,
  -1.4378337510064140e-15,
  8.060201326248057e-16,
  2.2757658578306265e-05,
  -1.988031728863744e-06,
  -8.901336865489784e-09,
  8.161584966021705e-10
]
[
  0.14372121890411854,
  -1.4868311563918534e-12,
  -1.2047278118323332e-13,
  6.863505058677813e-11,
  -3.299473980154729e-10,
  -4.851020561469704e-16,
  5.92344470241633e-16,
  2.4759096674780938e-05,
  -1.4579318080995177e-06,
  -1.705784534447382e-08,
  7.045245746920344e-10
]
[
  0.24025369570399946,
  -1.2348123020955780e-12,
  -6.362901079256059e-14,
  4.719828305704809e-11,
  -3.276889509601229e-10,
  -1.2108269119090016e-15,
  1.0421695923608714e-15,
  1.826514366396874e-05,
  -1.85247721977954e-06,
  -6.53478192798623e-09,
  6.686797453491852e-10
]
9
[
  0.010271807773104644,
  -1.1947187220101849e-12,
  -1.346442133492086e-13,
  6.395955579106194e-11,
  -1.872515032684015e-11,
  -1.0796148320595799e-15,
  6.535201197403842e-16,
  2.5973200089087378e-05,
  -1.8563764687677866e-06,
  -1.524603090943035e-08,
  8.445163800284704e-10
]
```

Number of best vectors being retained for the next gen: 25 %
Total number of iterations: 3
Fitness function: 0.95 max + 0.05 min

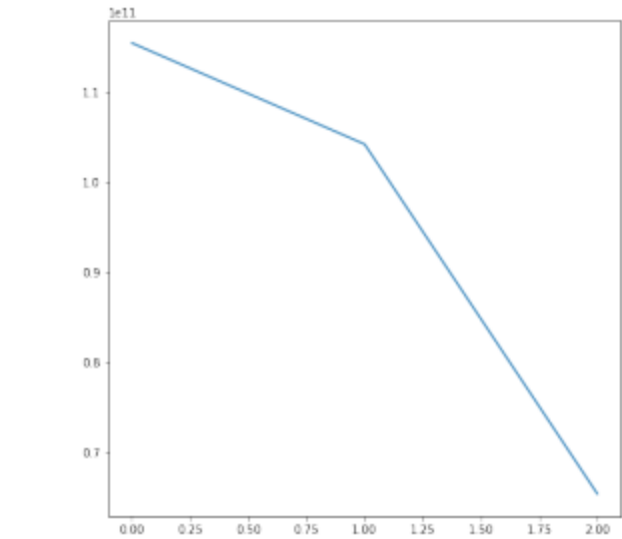
Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: SBX crossover with u=3
Crossover parameters if any: u=3

Mutation function: Naive mutation in neighbourhood

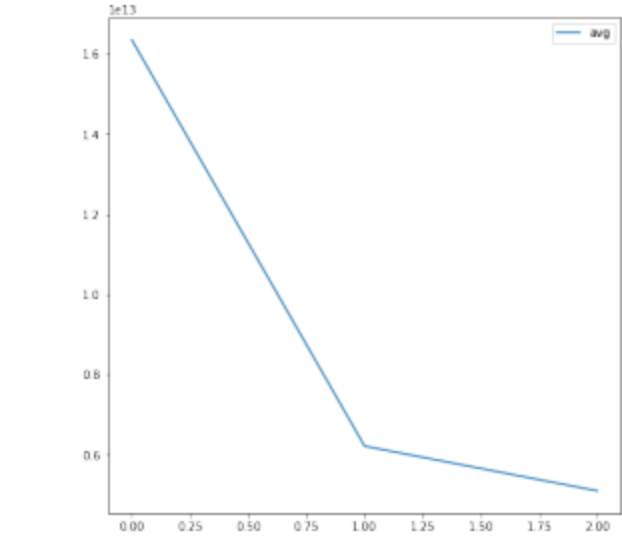
Mutation magnitude: 5%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 20 %
Was percentage of cells being mutated change midway: YES

Additional comments:



76



8th MARCH

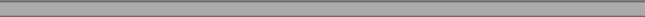
PART 1 PDF MISSING

JSON Name: json_dump_test_8th_STINT_2_check.json
ipynb Name:8_March_mongo_version_STINT_(x+1).pdf
DESC: "PROPER 8th March, STINT_2 CHECK AGAIN"
Date: 8th March
Population Size:36

Initial population description: 4 worthy members
[-0.1427695714633879, -1.4351758102671513e-12, -8.23952024928039e-14, 3.
,!9792353931611e-11, -2.2891737891652097e-10, -1.2741372955243258e-15, 8.
,!498419408139292e-16, 2.2978273844682766e-05, -1.8661056044695021e-06, -9.
,!19468573928787e-09, 7.519936353791697e-10],
[0.07144470438737868, -1.024341258927964e-12, -2.234254875776415e-14, 7.
,!518054647321033e-11, -1.801621324121818e-10, -1.3903926150219416e-15, 1.
,!0363630046234666e-15, 1.972934634335282e-05, -1.8376799350638052e-06, -7.
,!682037245173995e-09, 7.433548643729261e-10],
[-0.12826988257084007, -1.3783500660791644e-12, -6.618301505877671e-14, 4.
,!627780406755569e-11, -1.7964979058119056e-10, -1.3330380203323678e-15, 7.
,!763006729384186e-16, 2.2757658578306265e-05, -1.9193275505106666e-06, -8.
,!894106807593951e-09, 7.758133607867065e-10],
[0.0, -9.875299804205226e-13, -5.3944821589819837e-14, 7.
,!930563099592005e-11, -2.0984549655895666e-10, -1.4378337510064146e-15, 8.
,!060220128248057e-16, 2.2757658578306265e-05, -1.989031728863744e-06, -8.
,!908335865469784e-09, 8.161584966021705e-10]
]

Number of best vectors being retained for the next gen: 25 %
Total number of iterations: 7
Fitness function:

```
fitness_val=0
t=its_fitness_val[0]/(1e10)
v=its_fitness_val[1]/(1e10)
```



```
punish = 0

punish += 100 if t < 28 else 0

punish += 100 if v > 10 else 0

fitness_val = t + v + punish
```

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: SBX crossover with u=3
Crossover parameters if any: u=3

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 40 %
Was percentage of cells being mutated change midway: YES

Additional comments:

JSON Name:json_dump_test_8th_STINT_3_check_check.json'
ipynb Name:8_March_mongo_version_STINT_(x+2).pdf
DESC: "PROPER 8th March, STINT_3 CHECK AGAIN"
Date: 8th March
Population Size:20
Initial population description: 4 worthy members with 50% variation in 70% cells
[-0.1427695714633879, -1.4351758102671513e-12, -8.23952024928039e-14, 3.
,!9792353931611e-11, -2.2891737891652097e-10, -1.2741372955243258e-15, 8.
,!498419408139292e-16, 2.2978273844682766e-05, -1.8661056044695021e-06, -9.
,!19468573928787e-09, 7.519936353791697e-10],
[0.07144470438737868, -1.024341258927964e-12, -2.234254875776415e-14, 7.
,!518054647321033e-11, -1.801621324121818e-10, -1.3903926150219416e-15, 1.
,!0363630046234666e-15, 1.972934634335282e-05, -1.8376799350638052e-06, -7.
,!682037245173995e-09, 7.433548643729261e-10],
[-0.12826988257084007, -1.3783500660791644e-12, -6.618301505877671e-14, 4.
,!627780406755569e-11, -1.7964979058119056e-10, -1.3330380203323678e-15, 7.
,!763006729384186e-16, 2.2757658578306265e-05, -1.9193275505106666e-06, -8.
,!894106807593951e-09, 7.758133607867065e-10],
[0.0, -9.875299804205226e-13, -5.3944821589819837e-14, 7.
,!930563099592005e-11, -2.0984549655895666e-10, -1.4378337510064146e-15, 8.
,!060220128248057e-16, 2.2757658578306265e-05, -1.989031728863744e-06, -8.
,!908335865469784e-09, 8.161584966021705e-10]
]

Number of best vectors being retained for the next gen: 20 %
Total number of iterations: 11
Fitness function:

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: SBX crossover with u=1
Crossover parameters if any: u=1

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 40 %
Was percentage of cells being mutated change midway: YES

Additional comments:

JSON Name:json_dump_test_8th_STINT_4_check_check.json
ipynb Name:8_March_mongo_version stint_x+3.pdf
DESC: "PROPER 8th March, STINT_4 CHECK AGAIN"
Date: 8th March

Population Size:10
Initial population description: 4 worthy members with 50% variation in 70% cells
[-0.1427695714633879, -1.4351758102671513e-12, -8.23952024928039e-14, 3.
,!9792353931611e-11, -2.2891737891652097e-10, -1.2741372955243258e-15, 8.
,!498419408139292e-16, 2.2978273844682766e-05, -1.8661056044695021e-06, -9.
,!19468573928787e-09, 7.519936353791697e-10],
[0.07144470438737868, -1.024341258927964e-12, -2.234254875776415e-14, 7.
,!518054647321033e-11, -1.801621324121818e-10, -1.3903926150219416e-15, 1.
,!0363630046234666e-15, 1.972934634335282e-05, -1.8376799350638052e-06, -7.
,!682037245173995e-09, 7.433548643729261e-10],
[-0.12826988257084007, -1.3783500660791644e-12, -6.618301505877671e-14, 4.
,!627780406755569e-11, -1.7964979058119056e-10, -1.3330380203323678e-15, 7.
,!763006729384186e-16, 2.2757658578306265e-05, -1.9193275505106666e-06, -8.
,!894106807593951e-09, 7.758133607867065e-10],
[0.0, -9.875299804205226e-13, -5.3944821589819837e-14, 7.
,!930563099592005e-11, -2.0984549655895666e-10, -1.4378337510064146e-15, 8.
,!060220128248057e-16, 2.2757658578306265e-05, -1.989031728863744e-06, -8.
,!908335865469784e-09, 8.161584966021705e-10]
]

Number of best vectors being retained for the next gen: 20 %
Total number of iterations: 5
Fitness function:

```
fitness_val=0  
t=its_fitness_val[0]/(1e10)  
v=its_fitness_val[1]/(1e10)
```

4



```
punish = 0  
  
punish += 100 if t < 30 else 0  
  
punish += 100 if v > 17 else 0  
  
fitness_val = t + v + punish
```

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: SBX crossover with u=1
Crossover parameters if any: u=1

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 40 %
Was percentage of cells being mutated change midway: YES

Additional comments:

9th MARCH

PDF FOR FIRST STINT MISSING

JSON Name: json_dump_test_9th_STINT_2.json
ipynb Name:9_March_mongo_version_STINT_2.pdf
DESC: "PROPER 9th MARCH STINT_2"
Date: 9th March
Population Size:36

Initial population description: 2 worthy members with 80% variation in 70% cells
[0.1405910770983203, -1.7273777435221077e-12, -1.2047276118323332e-13, 7.
,!23118202530563e-11, -3.179423918833098e-10, -4.602893729601193e-16, 5.
,!922444470241633e-16, 2.5198183633776467e-05, -1.4579318080095177e-06, -1.
,!705784324447362e-08, 7.045245746926344e-10],
[0.43542751227864757, -1.330308667202201e-12, -1.2030721426306613e-13, 8.
,!419862855418934e-11, -6.246699146633278e-10, -1.108612018153589e-15, 1.
,!0989612241367958e-15, 2.5660207848837518e-05, -1.6448092463278343e-06, -1.
,!3573296156732487e-08, 7.21852191916886e-10]

Number of best vectors being retained for the next gen: 20 %
Total number of iterations: 10
Fitness function: 0.9 max + 0.1 min - 0.3 train

```
fitness_val=0
t=its_fitness_val[0]/(1e10)
v=its_fitness_val[1]/(1e10)
```

```
punish = 0

punish += 100 if t < 30 else 0

punish += 100 if v > 17 else 0

fitness_val = t + v + punish
```

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: SBX crossover with u=3
Crossover parameters if any: u=3

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 40 %
Was percentage of cells being mutated change midway: YES

Additional comments:

10th MARCH

JSON Name: json_dump_test_10th_STINT_1.json
ipynb Name:10_March_mongo_version_1.pdf
DESC: "PROPER 10th March STINT_1"
Date: 10th March
Population Size:50
Initial population description: 10 worthy members with 80% variation in 70% cells

```
[
0.07037735484644053,
-1.024341258927964e-12,
9
-2.234254875776415e-14,
7.716405737305545e-11,
-1.801621324121818e-10,
-1.3241653448018192e-15,
1.0363630046234666e-15,
1.972934634335282e-05,
-1.8376799350638052e-06,
-7.682037245173995e-09,
7.408557333642838e-10
],
[
0.03523998918426694,
-1.0117081002184981e-12,
-1.2994472471499564e-14,
7.893352705113582e-11,
-1.1860894152791612e-10,
-1.485052578582483e-15,
1.3056718502199423e-15,
2.0174164573336113e-05,
-1.812224643455608e-06,
-8.514787778320894e-09,
7.44044375654244e-10
],
[
-0.13993226827508395,
-1.4159770228070609e-12,
-8.274997712098415e-14,
4.049738114285228e-11,
-2.466974201620993e-10,
-1.2791979175143146e-15,
8.726629929919916e-16,
2.1906866100263204e-05,
-1.8638551631024331e-06,
-8.709922374940335e-09,
7.531818351308259e-10
],
[
```

0.26755941285701934,
-1.2459351551872372e-12,
-7.452707840711382e-14,
5.846061801710225e-11,
-2.971383448241105e-10,
-1.1940879271263902e-15,
1.0451380399518052e-15,
1.7221287415322333e-05,
-1.7029386807814643e-06,
-7.920633893732619e-09,
10
7.118911148183581e-10
],
[
-0.12826988257084007,
-1.3783500660791644e-12,
-6.618301505877671e-14,
4.627780406755569e-11,
-1.7964979058119056e-10,
-1.3330380203323678e-15,
7.763006729384186e-16,
2.2757658578306265e-05,
-1.9193275505106666e-06,
-8.894106807593951e-09,
7.758133607867065e-10
],
[
-0.010469555008325542,
-1.3756110090619466e-12,
-3.2098970500073777e-14,
7.515151589614182e-11,
-2.1425276047581854e-10,
-1.738427026022934e-15,
7.68636029308922e-16,
2.2405589606963618e-05,
-2.0150031522552243e-06,
-7.624698680490636e-09,
8.021327760726349e-10
],
[
0.0,
-9.875299804205226e-13,
-5.3944821589819837e-14,
7.930563099592005e-11,
-2.0984549655895666e-10,
-1.4378337510064146e-15,
8.060220128248057e-16,
2.2757658578306265e-05,
-1.989031728863744e-06,
-8.908335865469784e-09,
8.161584966021705e-10
],
[
0.14316662146238254,
-9.701610706365794e-13,
-1.2574367246859343e-14,
7.987909155153459e-11,
-9.614047328760541e-11,
11
-9.48951067043736e-16,
9.659819886715143e-16,
1.948878926694676e-05,
-1.5930369340321572e-06,
-6.180963548988106e-09,
6.020852300525158e-10
],
[
0.1404353012090963,
-1.616987929427746e-12,
-1.1961210548669883e-13,
7.011318547456527e-11,
-3.0352467806629654e-10,
-4.88498840328072e-16,
5.952174040049254e-16,
2.497979714709046e-05,
-1.5030814308516114e-06,
-1.3950416058207657e-08,
6.710158549311463e-10
],
[
0.06788979793629188,
-1.0346591606568448e-12,
-4.831324359219157e-14,
7.414504158932174e-11,
-1.7602557643371059e-10,
-1.408958442916462e-15,
9.881336787703099e-16,
2.058590506217254e-05,
-1.8375919680861094e-06,
-8.546005416410758e-09,
7.535922916172745e-10
]
]

Total number of iterations: 7
Fitness function: t+v

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: SBX crossover with u=3
Crossover parameters if any: u=3

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 30 %
Was percentage of cells being mutated change midway: YES

Additional comments:

JSON Name: json_dump_test_10th_STINT_2.json
ipynb Name:10_March_mongo_version_2.pdf
DESC: "PROPER 10th March STINT_2"
Date: 10th March
Population Size:60
Initial population description: 10 worthy members with 80% variation in 70% cells

[
0.07037735484644053,
-1.024341258927964e-12,
9
-2.234254875776415e-14,
7.716405737305545e-11,
-1.801621324121818e-10,
-1.3241653448018192e-15,
1.0363630046234666e-15,
1.972934634335282e-05,
-1.8376799350638052e-06,
-7.682037245173995e-09,
7.408557333642838e-10
],
[
0.03523998918426694,
-1.0117081002184981e-12,
-1.2994472471499564e-14,
7.893352705113582e-11,
-1.1860894152791612e-10,
-1.485052578582483e-15,
1.3056718502199423e-15,
2.0174164573336113e-05,
-1.812224643455608e-06,
-8.514787778320894e-09,
7.44044375654244e-10
],
[
-0.13993226827508395,
-1.4159770228070609e-12,
-8.274997712098415e-14,
4.049738114285228e-11,
-2.466974201620993e-10,
-1.2791979175143146e-15,
8.726629929919916e-16,
2.1906866100263204e-05,
-1.8638551631024331e-06,
-8.709922374940335e-09,
7.531818351308259e-10
],
[
0.26755941285701934,
-1.2459351551872372e-12,
-7.452707840711382e-14,
5.846061801710225e-11,
-2.971383448241105e-10,
-1.1940879271263902e-15,
1.0451380399518052e-15,
1.7221287415322333e-05,
-1.7029386807814643e-06,
-7.920633893732619e-09,
10
7.118911148183581e-10
],
[
-0.12826988257084007,
-1.3783500660791644e-12,
-6.618301505877671e-14,
4.627780406755569e-11,
-1.7964979058119056e-10,
-1.3330380203323678e-15,
7.763006729384186e-16,
2.2757658578306265e-05,
-1.9193275505106666e-06,
-8.894106807593951e-09,
7.758133607867065e-10
],
[
-0.010469555008325542,
-1.3756110090619466e-12,

-3.2098970500073777e-14,
7.515151589614182e-11,
-2.1425276047581854e-10,
-1.738427026022934e-15,
7.68636029308922e-16,
2.2405589606963618e-05,
-2.0150031522552243e-06,
-7.624698680490636e-09,
8.021327760726349e-10
],
[
0.0,
-9.875299804205226e-13,
-5.3944821589819837e-14,
7.930563099592005e-11,
-2.0984549655895666e-10,
-1.4378337510064146e-15,
8.060220128248057e-16,
2.2757658578306265e-05,
-1.989031728863744e-06,
-8.908335865469784e-09,
8.161584966021705e-10
],
[
0.14316662146238254,
-9.701610706365794e-13,
-1.2574367246859343e-14,
7.987909155153459e-11,
-9.614047328760541e-11,
11
-9.48951067043736e-16,
9.659819886715143e-16,
1.948878926694676e-05,
-1.5930369340321572e-06,
-6.180963548988106e-09,
6.020852300525158e-10
],
[
0.1404353012090963,
-1.616987929427746e-12,
-1.1961210548669883e-13,
7.011318547456527e-11,
-3.0352467806629654e-10,
-4.88498840328072e-16,
5.952174040049254e-16,
2.497979714709046e-05,
-1.5030814308516114e-06,
-1.3950416058207657e-08,
6.710158549311463e-10
],
[
0.06788979793629188,
-1.0346591606568448e-12,
-4.831324359219157e-14,
7.414504158932174e-11,
-1.7602557643371059e-10,
-1.408958442916462e-15,
9.881336787703099e-16,
2.058590506217254e-05,
-1.8375919680861094e-06,
-8.546005416410758e-09,
7.535922916172745e-10
]
]

Number of best vectors being retained for the next gen: 30 %
Total number of iterations:5
Fitness function:0.8max + 0.2 min

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: SBX crossover with u=3
Crossover parameters if any: u=3

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 40 %
Was percentage of cells being mutated change midway: YES

Additional comments:

11th March

JSON Name: json_dump_test_11th_STINT_1.json
ipynb Name:11_March_mongo_version STINT_1.pdf
DESC: "PROPER 11th MARCH STINT_1 0.9 * max(t,v) + 0.3 * min(t,v)"
Date: 11th March
Population Size:46
Initial population description: 10 worthy members with 80% variation in 70% cells
[
0.07037735484644053,

-1.024341258927964e-12,
9
-2.234254875776415e-14,
7.716405737305545e-11,
-1.801621324121818e-10,
-1.3241653448018192e-15,
1.0363630046234666e-15,
1.972934634335282e-05,
-1.8376799350638052e-06,
-7.682037245173995e-09,
7.408557333642838e-10
],
[
0.03523998918426694,
-1.0117081002184981e-12,
-1.2994472471499564e-14,
7.893352705113582e-11,
-1.1860894152791612e-10,
-1.485052578582483e-15,
1.3056718502199423e-15,
2.0174164573336113e-05,
-1.812224643455608e-06,
-8.514787778320894e-09,
7.44044375654244e-10
],
[
-0.13993226827508395,
-1.4159770228070609e-12,
-8.274997712098415e-14,
4.049738114285228e-11,
-2.466974201620993e-10,
-1.2791979175143146e-15,
8.726629929919916e-16,
2.1906866100263204e-05,
-1.8638551631024331e-06,
-8.709922374940335e-09,
7.531818351308259e-10
],
[
0.26755941285701934,
-1.2459351551872372e-12,
-7.452707840711382e-14,
5.846061801710225e-11,
-2.971383448241105e-10,
-1.1940879271263902e-15,
1.0451380399518052e-15,
1.7221287415322333e-05,
-1.7029386807814643e-06,
-7.920633893732619e-09,
10
7.118911148183581e-10
],
[
-0.12826988257084007,
-1.3783500660791644e-12,
-6.618301505877671e-14,
4.627780406755569e-11,
-1.7964979058119056e-10,
-1.3330380203323678e-15,
7.763006729384186e-16,
2.2757658578306265e-05,
-1.9193275505106666e-06,
-8.894106807593951e-09,
7.758133607867065e-10
],
[
-0.010469555008325542,
-1.3756110090619466e-12,
-3.2098970500073777e-14,
7.515151589614182e-11,
-2.1425276047581854e-10,
-1.738427026022934e-15,
7.68636029308922e-16,
2.2405589606963618e-05,
-2.0150031522552243e-06,
-7.624698680490636e-09,
8.021327760726349e-10
],
[
0.0,
-9.875299804205226e-13,
-5.3944821589819837e-14,
7.930563099592005e-11,
-2.0984549655895666e-10,
-1.4378337510064146e-15,
8.060220128248057e-16,
2.2757658578306265e-05,
-1.989031728863744e-06,
-8.908335865469784e-09,
8.161584966021705e-10
],
[
0.14316662146238254,
-9.701610706365794e-13,
-1.2574367246859343e-14,
7.987909155153459e-11,
-9.614047328760541e-11,
11

-9.48951067043736e-16,
9.659819886715143e-16,
1.948878926694676e-05,
-1.5930369340321572e-06,
-6.180963548988106e-09,
6.020852300525158e-10
],
[
0.1404353012090963,
-1.616987929427746e-12,
-1.1961210548669883e-13,
7.011318547456527e-11,
-3.0352467806629654e-10,
-4.88498840328072e-16,
5.952174040049254e-16,
2.497979714709046e-05,
-1.5030814308516114e-06,
-1.3950416058207657e-08,
6.710158549311463e-10
],
[
0.06788979793629188,
-1.0346591606568448e-12,
-4.831324359219157e-14,
7.414504158932174e-11,
-1.7602557643371059e-10,
-1.408958442916462e-15,
9.881336787703099e-16,
2.058590506217254e-05,
-1.8375919680861094e-06,
-8.546005416410758e-09,
7.535922916172745e-10
]
]

Number of best vectors being retained for the next gen: 20 %
Total number of iterations:14
Fitness function: 0.9 * max(t,v) + 0.3 * min(t,v)

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: SBX crossover with u=2
Crossover parameters if any: u=2

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 40 %
Was percentage of cells being mutated change midway: YES

Additional comments:

JSON Name: json_dump_test_11th_STINT_3.json
ipynb Name: 11_March_mongo_version STINT_3.pdf
DESC:"PROPER 11th MARCH STINT_3 val v"
Date: 11th March
Population Size:26
Initial population description: 1 worthy members with 80% variation in 70% cells
[
0.07144470438737868, -1.024341258927964e-12, -2.234254875776415e-14, 7.
,518054647321033e-11, -1.801621324121818e-10, -1.3903926150219416e-15, 1.
,0363630046234666e-15, 1.972934634335282e-05, -1.8376799350638052e-06, -7.
,!682037245173995e-09, 7.433548643729261e-10
]
]

Number of best vectors being retained for the next gen: 20 %
Total number of iterations:13
Fitness function: val

Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: SBX crossover with u=1
Crossover parameters if any: u=1

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 5%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 40 %
Was percentage of cells being mutated change midway: YES

Additional comments:

13th March

JSON Name: json_dump_test_13th_STINT_3.json
ipynb Name: 13_March_mongo_version.ipynb
DESC:"PROPER 13th MARCH STINT 1 [0.7*max(t,v)+0.3*min(t,v)]"

Date: 13th March
Population Size:40
Initial population description: 3 worthy members with 120% variation in 70% cells

[
0.07144470438737868, -1.024341258927964e-12, -2.234254875776415e-14, 7.
,!518054647321033e-11, -1.801621324121818e-10, -1.3903926150219416e-15, 1.
,!0363630046234666e-15, 1.972934634335282e-05, -1.8376799350638052e-06, -7.
,!682037245173995e-09, 7.433548643729261e-10],
[-0.09095988142406682, -1.4203303157343897e-12, -5.4620034600944574e-14, 4.
,!932216035549199e-11, -1.785324884664207e-10, -1.2943745560596486e-15, 8.
,!810038558693486e-16, 2.094915047366797e-05, -1.8608796305562087e-06, -8.
,!950362748811072e-09, 7.67677796428203e-10],
[0.5193216903108466, -8.033761650681735e-13, -7.51094294751401e-14, 7.
,!331399868909055e-11, -2.2766028073711876e-10, -1.956527142092565e-15, 4.
,!816791522433075e-16, 3.344995598237033e-05, -2.264872375573763e-06, -1.
,!1412871360508586e-08, 8.8968826660451e-10],
[0.0, -1.0217731653111337e-12, -1.2829164164652538e-13, 7.
,!485339950833092e-11, -1.4882280387987952e-10, -1.5914537149580294e-15, 7.
,!810723105904916e-16, 2.9258687441711774e-05, -1.8976677814090546e-06, -1.
,!7340868726097614e-08, 8.920882344120692e-10]
]

Number of best vectors being retained for the next gen: 20 %
Total number of iterations:25
Fitness function: 0.7 * max(t,v) + 0.3 * min(t,v)

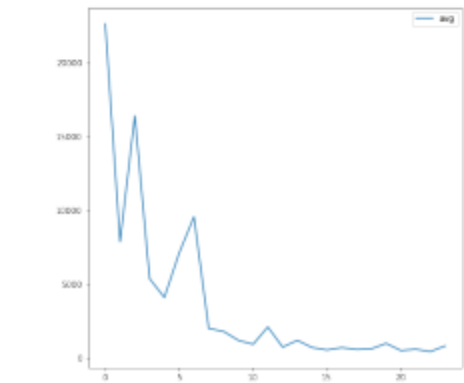
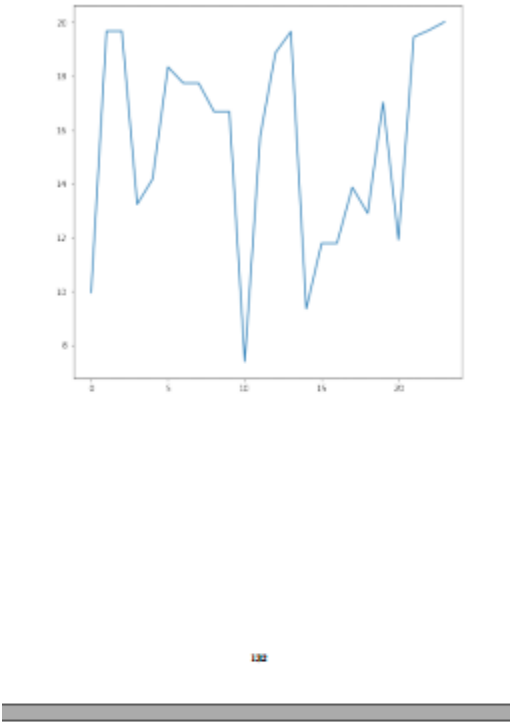
Selection for Crossover Criteria: Rank-based Russian Roulette
Crossover function: BLEND
Crossover parameters if any: alpha=0.5

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 20%
Was mutation magnitude changed midway ? : NO

Percentage of cells being mutated: 30 %
Was percentage of cells being mutated change midway: NO

Additional comments:



14th March

JSON Name: json_dump_test_14th_STINT_1.json
ipynb Name:14_March_mongo_version.ipynb
DESC:"[14th March 0.9 x max(t,v) + 0.1 x min(t,v) SBX RUSSIAN ROULETTE]"

Date: 14th March
Population Size:40
Initial population description: 6 worthy members with 400% variation in 70% cells
"
0-> cri
1-> cri
2-> good on ranklist
3-> good on ranklist (less than 75 e 10)
4-> test -> 210 e 10
5th -> 4510 e 10

'''

[0.1404353012090963, -1.5917637608939356e-12, -1.2461542166753883e-13, 6.
,!945508820550938e-11, -2.833820715681869e-10, -4.995728196797225e-16, 6.
,!007192185810843e-16, 2.3151128263733076e-05, -1.5030814308516114e-06, -1.
,!3260366172169755e-08, 6.710158549311463e-10],
[0.07340107154031256, -1.5437615866430792e-12, -1.4641151122621188e-13, 6.
,!645245422169678e-11, -3.7164454801951106e-10, -3.6675104747060576e-16, 4.
,!671199506291923e-16, 2.37846748093054e-05, -1.4689166025106545e-06, -1.
,!4156223932628392e-08, 6.63936470940267e-10],
[0.03523998918426694, -1.0117081002184981e-12, -1.2994472471499564e-14, 7.
,!893352705113582e-11, -1.1860894152791612e-10, -1.485052578582483e-15, 1.
,!3056718502199423e-15, 2.0174164573336113e-05, -1.812224643455608e-06, -8.
,!514787778320894e-09, 7.44044375654244e-10],
[0.10456903003668358, -8.310365465172867e-13, -1.1425053328907633e-14, 9.
,!427139040575664e-11, -1.313228713688508e-10, -1.1727056219971407e-15, 1.
,!0779150203669628e-15, 1.643454994947712e-05, -1.6601745706684455e-06, -6.
,!180195958180512e-09, 6.592835767247373e-10],
10
[-0.15312064598796957, -1.2759516587457305e-12, -6.334633169412846e-14, 1.
,!0791176309199842e-10, -4.5155117540841926e-10, -1.6054819562735365e-15, 8.
,!06656115843023e-16, 3.113179944203683e-05, -1.9525844629653752e-06, -1.
,!1537654091036487e-08, 7.67663074742261e-10],
[0.05271366053724377, -1.6029419079568322e-12, -1.1478915535579567e-13, 7.
,!996355167326429e-11, -1.9736060467401116e-10, -2.5951233583070186e-16, 6.
,!595567762286758e-16, 2.6845001767140418e-05, -1.2827130152295797e-06, -1.
,!939661532122104e-08, 6.55003088671148e-10]

Number of best vectors being retained for the next gen: 20 %
Total number of iterations:25
Fitness function: 0.9 * max(t,v) + 0.1 * min(t,v)

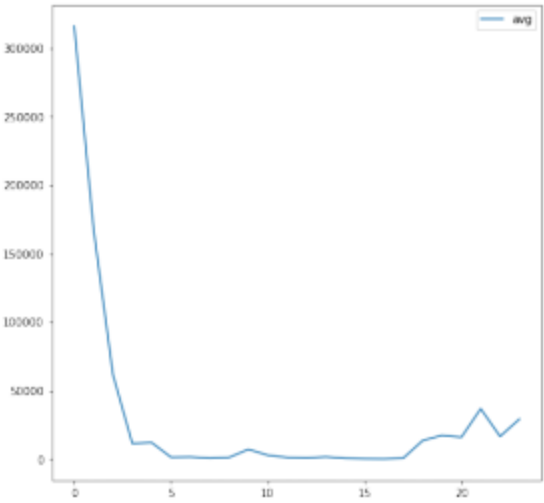
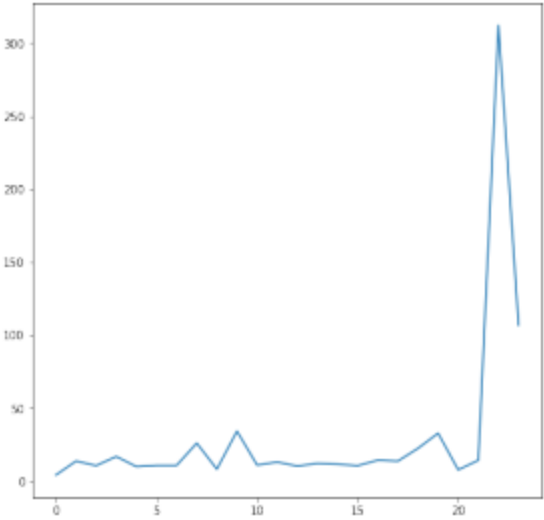
Selection for Crossover Criteria: FITNESS-based Russian Roulette
Crossover function: SBX with u=2
Crossover parameters if any: u=2

Mutation function: Naive mutation in neighbourhood

Mutation magnitude: 20%
Was mutation magnitude changed midway ? : YES

Percentage of cells being mutated: 30 %
Was percentage of cells being mutated change midway: YES

Additional comments:



+	
id	vec
+=====+	
=====+	
C0	[0.07144470438737868, -1.119955952479052e-12, -2.234254875776415e-14, 7.518054647321033e-11, -1.363149393789821e-10, -1.171865621470783e-15, 1.0363630046234666e-15, 1.7517805652149366e-05, -1.643820608253183e-06, -6.216662030472747e-09, 6.581384284971805e-10]
+-----+	
+	
C1	[0.07144470438737868, -1.024341258927964e-12, -2.234254875776415e-14, 7.518054647321033e-11, -1.801621324121818e-10, -1.3903926150219416e-15, 1.0363630046234666e-15, 1.972934634335282e-05, -1.8376799350638052e-06, -7.682037245173995e-09, 7.433548643729261e-10]
+-----+	
+	
C2	[0.03523998918426694, -1.0117081002184981e-12, -1.2994472471499564e-14, 7.893352705113582e-11, -1.1860894152791612e-10, -1.485052578582483e-15, 1.3056718502199423e-15, 2.0174164573336113e-05, -1.812224643455608e-06, -8.514787778320894e-09, 7.44044375654244e-10]
+-----+	
+	
C3	[0.04141639993961189, -1.0117081002184981e-12, -1.2994472471499564e-14, 7.893352705113582e-11, -1.604137146808921e-10, -1.485052578582483e-15, 1.0198774026946e-15, 2.0174164573336113e-05, -1.8350397232297526e-06, -7.936364861853528e-09, 7.44044375654244e-10]
+-----+	
+	
C4	[-0.1335339772661754, -1.3548655849568657e-12, -6.277613466694457e-14, 3.548832743701572e-11, -1.757399471896529e-10, -1.0293490966452213e-15, 8.102084319435116e-16, 2.2592178295527207e-05, -1.9234350522324353e-06, -9.482413997057998e-09, 8.389760499769913e-10]
+-----+	
+	
C5	[0.02368969753883534, -1.0351925813407968e-12, -1.6401352863331698e-14, 7.596029168379363e-11, -1.225187849194538e-10, -1.4538658485973253e-15, 1.2717640912148494e-15, 2.033964485611517e-05, -1.8198354893840126e-06, -8.58100868469191e-09, 7.505411627350811e-10]
+-----+	
+	
C6	[-0.16880571130193178, -1.4502028065050865e-12, -7.508126160418715e-14, 3.6129543320854e-11, -1.874048884974767e-10, -1.3922524907734511e-15, 7.235161352781883e-16, 2.3372311458548926e-05, -1.9358994684985523e-06, -9.457769665931136e-09, 7.824014226380575e-10]
+-----+	
+	
C7	[0.1119805331184704, -9.524885185020424e-13, -1.3444302212353717e-14, 1.0642518329329508e-10, -1.3368636621261531e-10, -9.820818024968418e-16, 1.0891475422836973e-15, 1.9114693463110164e-05, -1.82110801707592e-06, -5.553352707745898e-09, 7.367668025215753e-10]
+-----+	
+	
C8	[0.0715513448584818, -1.0241139480095258e-12, -2.237852435596594e-14, 9.63264755238362e-11, -1.4138725724074663e-10, -1.0400283870662846e-15, 1.036277056824697e-15, 1.972545393364221e-05, -1.8376068282132191e-06, -6.1181850076155814e-09, 7.433528943240485e-10]
+-----+	
+	
C9	[0.03401421439053448, -1.1041267190893494e-12, -9.715220177036874e-15, 7.888397407389302e-11, -1.6046792156904696e-10, -1.4863204644541236e-15, 1.0665304278242705e-15, 2.1095570641029852e-05, -1.8633402234258788e-06, -5.7065450464819686e-09, 7.440463457031218e-10]
+-----+	
+	
C10	[0.011437389025567138, -1.1138284981670104e-12, -3.551509026813513e-14, 6.649627355768133e-11, -1.8000819143924842e-10, -1.4089748097708505e-15, 8.902636627015963e-16, 1.8887774811975983e-05, -1.8622122152706824e-06, -7.437408281119532e-09, 7.22096963312873e-10]
+-----+	
+	
C11	[-0.06826256720902853, -1.2326828246421066e-12, -5.301047354840573e-14, 5.49620769830847e-11, -1.7980373155412395e-10, -1.4336551358161194e-15, 6.962188018897332e-16, 1.7770024187496272e-05, -1.8947952703037897e-06, -7.1124992660886254e-09, 6.938628426242507e-10]
+-----+	
+	
C12	[-0.13183162199859885, -1.5750763295631825e-12, -4.96682154842757e-14, 4.6697086717266865e-11, -1.697173755726951e-10, -1.5713189470508816e-15, 8.57788763945963e-16, 2.2782946197003435e-05, -2.1707227084973147e-06, -8.942011956960339e-09, 7.764601230308917e-10]
+-----+	
+	
C13	[0.05024462805451616, -8.827028312206423e-13, -1.2262754271144239e-14, 6.372037341451015e-11, -1.6022808710526955e-10, -1.7179657020171347e-15, 1.0231114114785379e-15, 2.1490242202637587e

+-----+		
+-----+		
M3		[0.04141639993961189, -1.0117081002184981e-12, -1.2994472471499564e-14, 7.893352705113582e-11, -1.604137146808921e-10, -1.485052578582483e-15, 1.0198774026946e-15, 2.0174164573336113e-05, -1.8350397232297526e-06, -7.936364861853528e-09, 7.44044375654244e-10] False
+-----+		
+-----+		
M4		[-0.1335339772661754, -1.3548655849568657e-12, -6.277613466694457e-14, 3.548832743701572e-11, -1.757399471896529e-10, -1.0293490966452213e-15, 8.102084319435116e-16, 2.2592178295527207e-05, -1.9234350522324353e-06, -9.482413997057998e-09, 8.390845746248741e-10] True
+-----+		
+-----+		
M5		[0.02368969753883534, -1.0351925813407968e-12, -1.6401352863331698e-14, 7.596029168379363e-11, -1.225187849194538e-10, -1.4538658485973253e-15, 1.2231518221918635e-15, 2.033964485611517e-05, -1.8198354893840126e-06, -8.58100868469191e-09, 7.505411627350811e-10] True
+-----+		
+-----+		
M6		[-0.16880571130193178, -1.4502028065050865e-12, -7.508126160418715e-14, 3.790635067362551e-11, -1.874048884974767e-10, -1.3922524907734511e-15, 7.235161352781883e-16, 2.3372311458548926e-05, -1.9358994684985523e-06, -9.457769665931136e-09, 7.824014226380575e-10] True
+-----+		
+-----+		
M7		[0.1119805331184704, -9.524885185020424e-13, -1.3972774469701094e-14, 1.0642518329329508e-10, -1.3368636621261531e-10, -9.820818024968418e-16, 1.0891475422836973e-15, 1.9114693463110164e-05, -1.82110801707592e-06, -5.553352707745898e-09, 7.367668025215753e-10] True
+-----+		
+-----+		
M8		[0.0715513448584818, -1.0241139480095258e-12, -2.237852435596594e-14, 9.63264755238362e-11, -1.4138725724074663e-10, -1.0173853539454782e-15, 1.036277056824697e-15, 1.9725453933642221e-05, -1.8376068282132191e-06, -6.1181850076155814e-09, 7.433528943240485e-10] True
+-----+		
+-----+		
M9		[0.03401421439053448, -1.1041267190893494e-12, -9.715220177036874e-15, 7.888397407389302e-11, -1.6046792156904696e-10, -1.4216861692911745e-15, 1.0665304278242705e-15, 2.1095570641029852e-05, -1.8633402234258788e-06, -5.7065450464819686e-09, 7.440463457031218e-10] True
+-----+		
+-----+		
M10		[0.011437389025567138, -1.1138284981670104e-12, -3.551509026813513e-14, 6.649627355768133e-11, -1.8000819143924842e-10, -1.4524819813594942e-15, 8.902636627015963e-16, 1.8887774811975983e-05, -1.8622122152706824e-06, -7.437408281119532e-09, 7.22096963312873e-10] True
+-----+		
+-----+		
M11		[-0.06826256720902853, -1.2326828246421066e-12, -5.4517894834626315e-14, 5.49620769830847e-11, -1.7980373155412395e-10, -1.4336551358161194e-15, 6.962188018897332e-16, 1.7770024187496272e-05, -1.8947952703037897e-06, -7.1124992660886254e-09, 6.938628426242507e-10] True
+-----+		
+-----+		
M12		[-0.13183162199859885, -1.5750763295631825e-12, -4.96682154842757e-14, 4.6697086717266865e-11, -1.697173755726951e-10, -1.5269335493517621e-15, 8.57788763945963e-16, 2.2782946197003435e-05, -2.1707227084973147e-06, -8.942011956960339e-09, 7.764601230308917e-10] True
+-----+		
+-----+		
M13		[0.05024462805451616, -8.827028312206423e-13, -1.2262754271144239e-14, 6.372037341451015e-11, -1.664949013881089e-10, -1.7179657020171347e-15, 1.0231114114785379e-15, 2.1490242202637587e-05, -1.828342142625986e-06, -6.493098966052742e-09, 7.433976134100588e-10] True
+-----+		
+-----+		
M14		[0.04670203575763512, -1.013931811393589e-12, -1.5132342249898894e-14, 7.827292070216484e-11, -1.349664179194054e-10, -1.4683903626122578e-15, 9.462519962902995e-16, 2.078934527036708e-05, -1.9396571258281e-06, -9.153537069524623e-09, 7.439230066461536e-10] True
+-----+		
+-----+		
M15		[0.06615906856935545, -1.0221175477528735e-12, -2.0802273278849015e-14, 7.584115282218133e-11, -1.7050722853261406e-10, -1.4070548309921671e-15, 1.0079887425227441e-15, 1.9955787895779288e-05, -1.8346863865063723e-06, -7.996385397921462e-09, 7.434762333810168e-10] True
+-----+		
+-----+		
M16		[0.0447321844449099, -1.0168328925332057e-12, -1.373792319362818e-14, 7.847707685241235e-11, -1.9229749538289532e-10, -1.5208171054723566e-15, 1.0164727729224303e-15, 2.0210275742963637e-05, -2.371286809094156e-06, -1.0005129990015494e-08, 7.444884313434796e-10] True
+-----+		
+-----+		
M17		[-0.12581744122545513, -1.373225273764457e-12, -6.54395643366481e-14, 4.673425426627917e-11, -1.798290818157693e-10, -1.3351628263076335e-15, 7.797053027105884e-16, 2.272154740867874e-05, -1.9257344310254454e-06, -8.82438829270469e-09, 7.753693050974711e-10] True
+-----+		
+-----+		
M18		[0.08046122949040725, -1.1000200835321397e-12, -2.1141910346084682e-14, 7.645145650260026e-11, -1.3481529834970585e-10, -1.1537019686794007e-15, 1.02110463800243e-15, 1.7281241732786006e-05, -1.620334907075895e-06, -6.0957832884010104e-09, 6.528257520565375e-10] True
+-----+		
+-----+		
M19		[-0.13728640767386865, -1.5814681256995965e-12, -5.0137135695599857e-14, 4.57591828399804e-11, -1.7103138902634885e-10, -1.5923512745264357e-15, 8.672413596163337e-16, 2.299422249766963e-05, -2.1875108290708365e-06, -9.014985549665688e-09, 7.811260372273497e-10] True
+-----+		
+-----+		

CONTENTS OF vec2.txt

20		
+-----+		
-----+-----+		
id		vec
	fitness	
+=====+		
=====		
=====		
=====+=====+		
P0		[0.07144470438737868, -1.119955952479052e-12, -2.234254875776415e-14, 7.518054647321033e-11, -1.363149393789821e-10, -1.171865621470783e-15, 1.0363630046234666e-15, 1.7517805652149366e-05, -1.643820608253183e-06, -6.2166620304727474e-09, 6.581384284971805e-10] 2.27462e+11
+-----+		
-----+-----+		
P1		[0.07144470438737868, -1.024341258927964e-12, -2.234254875776415e-14, 7.518054647321033e-11, -1.801621324121818e-10, -1.3903926150219416e-15, 1.0363630046234666e-15, 1.972934634335282e-05, -1.8376799350638052e-06, -7.682037245173995e-09, 7.433548643729261e-10] 2.92509e+11
+-----+		
-----+-----+		

P2 | [0.03523998918426694, -1.0117081002184981e-12, -1.2994472471499564e-14, 7.893352705113582e-11, -1.1860894152791612e-10, -1.485052578582483e-15, 1.3056718502199423e-15, 2.0174164573336113e-05, -1.812224643455608e-06, -8.514787778320894e-09, 7.44044375654244e-10] | 2.99394e+11 |

+-----+

| P3 | [0.04141639993961189, -1.0117081002184981e-12, -1.2994472471499564e-14, 7.893352705113582e-11, -1.604137146808921e-10, -1.485052578582483e-15, 1.0198774026946e-15, 2.0174164573336113e-05, -1.8350397232297526e-06, -7.936364861853528e-09, 7.44044375654244e-10] | 3.73188e+11 |

+-----+

| P4 | [-0.1335339772661754, -1.3548655849568657e-12, -6.277613466694457e-14, 3.548832743701572e-11, -1.757399471896529e-10, -1.0293490966452213e-15, 8.102084319435116e-16, 2.2592178295527207e-05, -1.9234350522324353e-06, -9.482413997057998e-09, 8.390845746248741e-10] | 4.47248e+11 |

+-----+

| P5 | [0.02368969753883534, -1.0351925813407968e-12, -1.6401352863331698e-14, 7.596029168379363e-11, -1.225187849194538e-10, -1.4538658485973253e-15, 1.2231518221918635e-15, 2.033964485611517e-05, -1.8198354893840126e-06, -8.58100868469191e-09, 7.505411627350811e-10] | 4.77819e+11 |

+-----+

| P6 | [-0.16880571130193178, -1.4502028065050865e-12, -7.508126160418715e-14, 3.790635067362551e-11, -1.874048884974767e-10, -1.3922524907734511e-15, 7.235161352781883e-16, 2.3372311458548926e-05, -1.9358994684985523e-06, -9.457769665931136e-09, 7.824014226380575e-10] | 5.27117e+11 |

+-----+

| P7 | [0.1119805331184704, -9.524885185020424e-13, -1.3972774469701094e-14, 1.0642518329329508e-10, -1.3368636621261531e-10, -9.820818024968418e-16, 1.0891475422836973e-15, 1.9114693463110164e-05, -1.82110801707592e-06, -5.553352707745898e-09, 7.367668025215753e-10] | 1.15007e+12 |

+-----+

| P8 | [0.0715513448584818, -1.0241139480095258e-12, -2.237852435596594e-14, 9.63264755238362e-11, -1.4138725724074663e-10, -1.0173853539454782e-15, 1.036277056824697e-15, 1.972545393364221e-05, -1.8376068282132191e-06, -6.1181850076155814e-09, 7.433528943240485e-10] | 1.1973e+12 |

+-----+

| P9 | [0.03401421439053448, -1.1041267190893494e-12, -9.715220177036874e-15, 7.888397407389302e-11, -1.6046792156904696e-10, -1.4216861692911745e-15, 1.0665304278242705e-15, 2.1095570641029852e-05, -1.8633402234258788e-06, -5.7065450464819686e-09, 7.440463457031218e-10] | 1.37517e+12 |

+-----+

| P10 | [0.011437389025567138, -1.1138284981670104e-12, -3.551509026813513e-14, 6.649627355768133e-11, -1.8000819143924842e-10, -1.4524819813594942e-15, 8.902636627015963e-16, 1.8887774811975983e-05, -1.8622122152706824e-06, -7.437408281119532e-09, 7.22096963312873e-10] | 1.44804e+12 |

+-----+

| P11 | [-0.06826256720902853, -1.2326828246421066e-12, -5.4517894834626315e-14, 5.49620769830847e-11, -1.7980373155412395e-10, -1.4336551358161194e-15, 6.962188018897332e-16, 1.7770024187496272e-05, -1.8947952703037897e-06, -7.1124992660886254e-09, 6.938628426242507e-10] | 1.63375e+12 |

+-----+

| P12 | [-0.13183162199859885, -1.5750763295631825e-12, -4.96682154842757e-14, 4.6697086717266865e-11, -1.697173755726951e-10, -1.5269335493517621e-15, 8.57788763945963e-16, 2.2782946197003435e-05, -2.1707227084973147e-06, -8.942011956960339e-09, 7.764601230308917e-10] | 1.68777e+12 |

+-----+

| P13 | [0.05024462805451616, -8.827028312206423e-13, -1.2262754271144239e-14, 6.372037341451015e-11, -1.664949013881089e-10, -1.7179657020171347e-15, 1.0231114114785379e-15, 2.1490242202637587e-05, -1.828342142625986e-06, -6.493098966052742e-09, 7.433976134100588e-10] | 2.55501e+12 |

+-----+

| P14 | [0.04670203575763512, -1.013931811393589e-12, -1.5132342249898894e-14, 7.827292070216484e-11, -1.349664179194054e-10, -1.4683903626122578e-15, 9.462519962902995e-16, 2.078934527036708e-05, -1.9396571258281e-06, -9.153537069524623e-09, 7.439230066461536e-10] | 3.61149e+12 |

+-----+

| P15 | [0.06615906856935545, -1.0221175477528735e-12, -2.0802273278849015e-14, 7.584115282218133e-11, -1.7050722853261406e-10, -1.4070548309921671e-15, 1.0079887425227441e-15, 1.9955787895779288e-05, -1.8346863865063723e-06, -7.996385397921462e-09, 7.434762333810168e-10] | 3.96911e+12 |

+-----+

| P16 | [0.0447321844449099, -1.0168328925332057e-12, -1.373792319362818e-14, 7.847707685241235e-11, -1.9229749538289532e-10, -1.5208171

+-----+-----	
-----+-----+	
P17	[-0.12581744122545513, -1.373225273764457e-12, -6.54395643366481e-14, 4.673425426627917e-11, -1.798290818157693e-10, -1.3351628263076335e-15, 7.797053027105884e-16, 2.272154740867874e-05, -1.9257344310254454e-06, -8.82438829270469e-09, 7.753693050974711e-10] 1.07018e+13
+-----+-----	
-----+-----+	
P18	[0.08046122949040725, -1.1000200835321397e-12, -2.1141910346084682e-14, 7.645145650260026e-11, -1.3481529834970585e-10, -1.1537019686794007e-15, 1.02110463800243e-15, 1.7281241732786006e-05, -1.620334907075895e-06, -6.0957832884010104e-09, 6.528257520565375e-10] 1.09511e+13
+-----+-----	
-----+-----+	
P19	[-0.13728640767386865, -1.5814681256995965e-12, -5.0137135695599857e-14, 4.57591828399804e-11, -1.7103138902634885e-10, -1.5923512745264357e-15, 8.672413596163337e-16, 2.299422249766963e-05, -2.1875108290708365e-06, -9.014985549665688e-09, 7.811260372273497e-10] 5.32965e+13
+-----+-----	
-----+-----+	
+-----+-----	
-----+-----	
id	vec
+=====+	
=====	
=====+	
C0	[0.07144470438737868, -1.119955952479052e-12, -2.234254875776415e-14, 7.518054647321033e-11, -1.363149393789821e-10, -1.171865621470783e-15, 1.0363630046234666e-15, 1.7517805652149366e-05, -1.643820608253183e-06, -6.2166620304727474e-09, 6.581384284971805e-10]
+-----+-----	
-----+-----	
C1	[0.04141639993961189, -1.0117081002184981e-12, -1.2994472471499564e-14, 7.937378184676173e-11, -1.604137146808921e-10, -1.485052578582483e-15, 1.0198774026946e-15, 2.0174164573336113e-05, -1.8350397232297526e-06, -7.936364861853528e-09, 7.44044375654244e-10]
+-----+-----	
-----+-----	
C2	[0.06615906856935545, -1.0221175477528735e-12, -2.0802273278849015e-14, 7.584115282218133e-11, -1.7050722853261406e-10, -1.4070548309921671e-15, 1.0079887425227441e-15, 1.9955787895779288e-05, -1.8346863865063723e-06, -7.996385397921462e-09, 7.434762333810168e-10]
+-----+-----	
-----+-----	
C3	[0.08046122949040725, -1.1136075369327564e-12, -2.1141910346084682e-14, 7.125326979584275e-11, -1.3481529834970585e-10, -1.1537019686794007e-15, 1.02110463800243e-15, 1.7281241732786006e-05, -1.620334907075895e-06, -6.0957832884010104e-09, 6.528257520565375e-10]
+-----+-----	
-----+-----	
C4	[-0.11817970161890119, -1.3592565367143253e-12, -6.366368036935533e-14, 4.789226527586557e-11, -1.7945821410855167e-10, -1.338023032549327e-15, 8.239738233208918e-16, 2.1618171612668106e-05, -1.922112106312563e-06, -8.928023763069466e-09, 7.74100446853909e-10]
+-----+-----	
-----+-----	
C5	[0.0585213289628015, -1.0360862848030052e-12, -2.2578157246141794e-14, 7.468314181259493e-11, -1.7087809623983168e-10, -1.4041946247504736e-15, 1.0003644210891711e-15, 2.002466572296022e-05, -1.8383087112192547e-06, -8.03498612870213e-09, 7.447450916245789e-10]
+-----+-----	
-----+-----	
C6	[0.07361502801468481, -1.0522935423454853e-12, -1.9130152288104575e-14, 6.954653495897632e-11, -1.4316024902642308e-10, -1.2815476899154405e-15, 1.0215593144015187e-15, 1.8234878603204773e-05, -1.667463286023651e-06, -6.185803444520753e-09, 6.733466968812883e-10]
+-----+-----	
-----+-----	
C7	[0.05709082953023861, -9.043042346940603e-13, -1.4274512329124348e-14, 6.542710825137658e-11, -1.6330187381120014e-10, -1.5901199807810949e-15, 1.022656735079449e-15, 2.053660533221882e-05, -1.7812137636782303e-06, -6.403078809933e-09, 7.228766685853081e-10]
+-----+-----	
-----+-----	
C8	[0.02324156296553284, -1.0614581814434209e-12, -1.6471313124840162e-14, 7.604213783598347e-11, -1.2230497195372427e-10, -1.4590851271948054e-15, 1.2266650381871037e-15, 2.0392824665652627e-05, -1.8233044248192498e-06, -8.624222024432592e-09, 7.167644117506565e-10]
+-----+-----	
-----+-----	
C9	[0.0814391661333448, -1.1144988173042776e-12, -2.1221735141377253e-14, 7.117142364365294e-11, -1.350291113154354e-10, -1.1484826900819208e-15, 1.0175914220071899e-15, 1.7228061923248557e-05, -1.616865971640658e-06, -6.052569948660328e-09, 6.517329816506141e-10]

+-----+-----+
-----+
| C10 | [0.1079861028152531, -9.537554252631213e-13, -1.4245170290355798e-14, 1.0691928249886176e-10, -1.3324715966117508e-10, -9.73567547283569e-16, 1.0899115873611054e-15, 1.974553316935059e-05, -1.8433109471473784e-06, -5.54247974329572e-09, 7.366900833322951e-10] |
+-----+-----+
-----+
| C11 | [0.049584194321066574, -8.417233333457106e-13, -1.2240079851790676e-14, 6.322627420894349e-11, -1.7208603103935765e-10, -1.726479957230408e-15, 1.0223473664011302e-15, 2.1510197784053054e-05, -1.8281709328760395e-06, -6.503971930502921e-09, 7.43474332599339e-10] |
+-----+-----+
-----+
| C12 | [0.039347367016421685, -1.0251701189776464e-12, -1.1987914998237531e-14, 7.98581840555701e-11, -1.6042886640000432e-10, -1.4790440766084544e-15, 1.032917692527883e-15, 2.0459319469898664e-05, -1.8429501797105192e-06, -7.313093453673185e-09, 7.440449263153915e-10] |
+-----+-----+
-----+
| C13 | [0.036083247313724684, -1.0464078867179776e-12, -1.0399963427459427e-14, 8.062238010875007e-11, -1.6045276984993474e-10, -1.4695650250876697e-15, 1.0534901379909875e-15, 2.0909181676419384e-05, -1.8554297669451122e-06, -6.329816454662312e-09, 7.440457950419744e-10] |
+-----+-----+
-----+
| C14 | [0.0503503260471544, -8.409702267016169e-13, -1.2253993606609094e-14, 6.353295937992691e-11, -1.7178131548002063e-10, -1.7207543079849144e-15, 1.0231501313929065e-15, 2.1505999244907303e-05, -1.8282619543009747e-06, -6.4758191329940755e-09, 7.433898699002589e-10] |
+-----+-----+
-----+
| C15 | [0.04131070194697365, -1.0137281136236704e-12, -1.3003233136034709e-14, 7.956119588134497e-11, -1.6027922368878884e-10, -1.4822639726147034e-15, 1.0198386827802313e-15, 2.0158407531066398e-05, -1.835119911554764e-06, -7.953644694912195e-09, 7.44052119164044e-10] |
+-----+-----+
-----+
| C16 | [-0.13651054430864523, -1.3963410521029743e-12, -6.879295722420292e-14, 4.4647249443731137e-11, -1.810705202753829e-10, -1.3255787220317128e-15, 8.033357194586213e-16, 2.1783784847694232e-05, -1.9315335435304432e-06, -9.032500349900794e-09, 7.773722531687259e-10] |
+-----+-----+
-----+
| C17 | [0.052109503022802, -9.88592321879981e-13, -9.641079583944745e-15, 8.146078666930978e-11, -1.5917227622127854e-10, -1.4946366828584041e-15, 1.0328911851232977e-15, 2.0077429165490923e-05, -1.829240610724755e-06, -7.870489005802866e-09, 7.420414275829894e-10] |
+-----+-----+
-----+
| C18 | [0.06891531504574777, -1.0287766298900422e-12, -2.295364156375044e-14, 9.566412877535375e-11, -1.4190068085572775e-10, -1.021629545276754e-15, 1.033339735260536e-15, 1.9751652727977843e-05, -1.8387838481105752e-06, -6.366621706604675e-09, 7.437805010138905e-10] |
+-----+-----+
-----+
| C19 | [-0.1231814114127211, -1.3685625918839406e-12, -6.486444712886359e-14, 4.739660101476161e-11, -1.7931565820078816e-10, -1.3309186349763576e-15, 8.192868234514797e-16, 2.1660850645513402e-05, -1.924557411128089e-06, -8.931429157474235e-09, 7.749416984076291e-10] |
+-----+-----+
-----+

+-----+-----+-----+
-----+-----+
| id | vec
| mut? |
+=====+=====+
=====+
=====+
| M0 | [0.07144470438737868, -1.119955952479052e-12, -2.234254875776415e-14, 7.518054647321033e-11, -1.363149393789821e-10, -1.171865621470783e-15, 1.0363630046234666e-15, 1.7517805652149366e-05, -1.643820608253183e-06, -6.2166620304727474e-09, 6.581384284971805e-10] | False |
+-----+-----+
-----+-----+
| M1 | [0.04089673378121227, -1.0117081002184981e-12, -1.2994472471499564e-14, 7.937378184676173e-11, -1.604137146808921e-10, -1.485052578582483e-15, 1.0198774026946e-15, 2.0174164573336113e-05, -1.8350397232297526e-06, -7.936364861853528e-09, 7.44044375654244e-10] | True |
+-----+-----+
-----+-----+
| M2 | [0.06823780246382927, -1.0221175477528735e-12, -2.0802273278849015e-14, 7.584115282218133e-11, -1.7050722853261406e-10, -1.4070548309921671e-15, 1.0079887425227441e-15, 1.9955787895779288e-05, -1.8346863865063723e-06, -7.996385397921462e-09, 7.434762333810168e-10] | True |

+-----+	
-----+-----+	
M3	[0.08046122949040725, -1.1136075369327564e-12, -2.1141910346084682e-14, 7.125326979584275e-11, -1.3481529834970585e-10, -1.1537019686794007e-15, 1.02110463800243e-15, 1.6932113369328485e-05, -1.620334907075895e-06, -6.0957832884010104e-09, 6.528257520565375e-10] True
+-----+	
-----+-----+	
M4	[-0.11817970161890119, -1.3592565367143253e-12, -6.366368036935533e-14, 4.789226527586557e-11, -1.7945821410855167e-10, -1.338023032549327e-15, 8.239738233208918e-16, 2.1618171612668106e-05, -1.922112106312563e-06, -8.928023763069466e-09, 7.74100446853909e-10] False
+-----+	
-----+-----+	
M5	[0.0585213289628015, -1.0360862848030052e-12, -2.330685720537812e-14, 7.468314181259493e-11, -1.7087809623983168e-10, -1.4041946247504736e-15, 1.0003644210891711e-15, 2.002466572296022e-05, -1.8383087112192547e-06, -8.03498612870213e-09, 7.447450916245789e-10] True
+-----+	
-----+-----+	
M6	[0.07361502801468481, -1.0522935423454853e-12, -1.9130152288104575e-14, 6.954653495897632e-11, -1.4742239343450159e-10, -1.2815476899154405e-15, 1.0215593144015187e-15, 1.8234878603204773e-05, -1.667463286023651e-06, -6.185803444520753e-09, 6.733466968812883e-10] True
+-----+	
-----+-----+	
M7	[0.05709082953023861, -9.043042346940603e-13, -1.4274512329124348e-14, 6.542710825137658e-11, -1.6330187381120014e-10, -1.5901199807810949e-15, 1.022656735079449e-15, 2.053660533221882e-05, -1.7137041233595913e-06, -6.403078809933e-09, 7.228766685853081e-10] True
+-----+	
-----+-----+	
M8	[0.02324156296553284, -1.0614581814434209e-12, -1.6471313124840162e-14, 7.604213783598347e-11, -1.2230497195372427e-10, -1.4590851271948054e-15, 1.2266650381871037e-15, 2.0392824665652627e-05, -1.8233044248192498e-06, -8.624222024432592e-09, 7.167644117506565e-10] False
+-----+	
-----+-----+	
M9	[0.08182819594004698, -1.1144988173042776e-12, -2.1221735141377253e-14, 7.117142364365294e-11, -1.350291113154354e-10, -1.1484826900819208e-15, 1.0175914220071899e-15, 1.7228061923248557e-05, -1.616865971640658e-06, -6.052569948660328e-09, 6.517329816506141e-10] True
+-----+	
-----+-----+	
M10	[0.1079861028152531, -9.537554252631213e-13, -1.4245170290355798e-14, 1.0691928249886176e-10, -1.3324715966117508e-10, -9.73567547283569e-16, 1.1063352638434942e-15, 1.974553316935059e-05, -1.8433109471473784e-06, -5.54247974329572e-09, 7.366900833322951e-10] True
+-----+	
-----+-----+	
M11	[0.049584194321066574, -8.417233333457106e-13, -1.2240079851790676e-14, 6.322627420894349e-11, -1.7208603103935765e-10, -1.726479957230408e-15, 1.0223473664011302e-15, 2.1510197784053054e-05, -1.8281709328760395e-06, -6.503971930502921e-09, 7.43474332599339e-10] False
+-----+	
-----+-----+	
M12	[0.039347367016421685, -1.0251701189776464e-12, -1.1987914998237531e-14, 7.98581840555701e-11, -1.6042886640000432e-10, -1.4790440766084544e-15, 1.032917692527883e-15, 2.0459319469898664e-05, -1.8429501797105192e-06, -7.313093453673185e-09, 7.676339552141659e-10] True
+-----+	
-----+-----+	
M13	[0.036083247313724684, -1.0464078867179776e-12, -1.0399963427459427e-14, 8.062238010875007e-11, -1.6045276984993474e-10, -1.4695650250876697e-15, 1.0038833934025524e-15, 2.0909181676419384e-05, -1.8554297669451122e-06, -6.329816454662312e-09, 7.440457950419744e-10] True
+-----+	
-----+-----+	
M14	[0.0503503260471544, -8.409702267016169e-13, -1.2253993606609094e-14, 6.353295937992691e-11, -1.7178131548002063e-10, -1.7526421786717257e-15, 1.0231501313929065e-15, 2.1505999244907303e-05, -1.8282619543009747e-06, -6.4758191329940755e-09, 7.433898699002589e-10] True
+-----+	
-----+-----+	
M15	[0.04131070194697365, -1.0137281136236704e-12, -1.2537619253106434e-14, 7.956119588134497e-11, -1.6027922368878884e-10, -1.4822639726147034e-15, 1.0198386827802313e-15, 2.0158407531066398e-05, -1.835119911554764e-06, -7.953644694912195e-09, 7.44052119164044e-10] True
+-----+	
-----+-----+	
M16	[-0.13651054430864523, -1.3963410521029743e-12, -6.879295722420292e-14, 4.4647249443731137e-11, -1.810705202753829e-10, -1.3255787220317128e-15, 8.033357194586213e-16, 2.1783784847694232e-05, -1.867985038783987e-06, -9.032500349900794e-09, 7.773722531687259e-10] True
+-----+	
-----+-----+	

| M17 | [0.052109503022802, -9.88592321879981e-13, -9.641079583944745e-15, 8.146078666930978e-11, -1.5917227622127854e-10, -1.4946366828584041e-15, 1.0328911851232977e-15, 2.0077429165490923e-05, -1.8605135924128038e-06, -7.870489005802866e-09, 7.420414275829894e-10] | True |

+-----+-----+

+-----+

| M18 | [0.06891531504574777, -1.0287766298900422e-12, -2.295364156375044e-14, 9.566412877535375e-11, -1.4190068085572775e-10, -1.021629545276754e-15, 1.033339735260536e-15, 1.9751652727977843e-05, -1.8387838481105752e-06, -6.366621706604675e-09, 7.608922359732318e-10] | True |

+-----+-----+

+-----+

| M19 | [-0.1231814114127211, -1.3685625918839406e-12, -6.486444712886359e-14, 4.739660101476161e-11, -1.7931565820078816e-10, -1.3309186349763576e-15, 8.192868234514797e-16, 2.1738265132259e-05, -1.924557411128089e-06, -8.931429157474235e-09, 7.749416984076291e-10] | True |

+-----+-----+

+-----+

CONTENTS OF vec3.txt

20

+-----+-----+

+-----+

| id | vec

| fitness |

+=====+

=====

=====+

| P0 | [0.07144470438737868, -1.119955952479052e-12, -2.234254875776415e-14, 7.518054647321033e-11, -1.363149393789821e-10, -1.171865621470783e-15, 1.0363630046234666e-15, 1.7517805652149366e-05, -1.643820608253183e-06, -6.2166620304727474e-09, 6.581384284971805e-10] | 2.27462e+11 |

+-----+-----+

+-----+

| P1 | [0.04089673378121227, -1.0117081002184981e-12, -1.2994472471499564e-14, 7.937378184676173e-11, -1.604137146808921e-10, -1.485052578582483e-15, 1.0198774026946e-15, 2.0174164573336113e-05, -1.8350397232297526e-06, -7.936364861853528e-09, 7.44044375654244e-10] | 2.92509e+11 |

+-----+-----+

+-----+

| P2 | [0.06823780246382927, -1.0221175477528735e-12, -2.0802273278849015e-14, 7.584115282218133e-11, -1.7050722853261406e-10, -1.4070548309921671e-15, 1.0079887425227441e-15, 1.9955787895779288e-05, -1.8346863865063723e-06, -7.996385397921462e-09, 7.434762333810168e-10] | 2.99394e+11 |

+-----+-----+

+-----+

| P3 | [0.08046122949040725, -1.1136075369327564e-12, -2.1141910346084682e-14, 7.125326979584275e-11, -1.3481529834970585e-10, -1.1537019686794007e-15, 1.02110463800243e-15, 1.6932113369328485e-05, -1.620334907075895e-06, -6.0957832884010104e-09, 6.528257520565375e-10] | 3.73188e+11 |

+-----+-----+

+-----+

| P4 | [-0.11817970161890119, -1.3592565367143253e-12, -6.366368036935533e-14, 4.789226527586557e-11, -1.7945821410855167e-10, -1.338023032549327e-15, 8.239738233208918e-16, 2.1618171612668106e-05, -1.922112106312563e-06, -8.928023763069466e-09, 7.74100446853909e-10] | 4.47248e+11 |

+-----+-----+

+-----+

| P5 | [0.0585213289628015, -1.0360862848030052e-12, -2.330685720537812e-14, 7.468314181259493e-11, -1.7087809623983168e-10, -1.4041946247504736e-15, 1.0003644210891711e-15, 2.002466572296022e-05, -1.8383087112192547e-06, -8.03498612870213e-09, 7.447450916245789e-10] | 4.77819e+11 |

+-----+-----+

+-----+

| P6 | [0.07361502801468481, -1.0522935423454853e-12, -1.9130152288104575e-14, 6.954653495897632e-11, -1.4742239343450159e-10, -1.2815476899154405e-15, 1.0215593144015187e-15, 1.8234878603204773e-05, -1.667463286023651e-06, -6.185803444520753e-09, 6.733466968812883e-10] | 5.27117e+11 |

+-----+-----+

+-----+

| P7 | [0.05709082953023861, -9.043042346940603e-13, -1.4274512329124348e-14, 6.542710825137658e-11, -1.6330187381120014e-10, -1.5901199807810949e-15, 1.022656735079449e-15, 2.053660533221882e-05, -1.7137041233595913e-06, -6.403078809933e-09, 7.228766685853081e-10] | 1.15007e+12 |

+-----+-----+

+-----+

| P8 | [0.02324156296553284, -1.0614581814434209e-12, -1.6471313124840162e-14, 7.604213783598347e-11, -1.2230497195372427e-10, -1.4590851271948054e-15, 1.2266650381871037e-15, 2.0392824665652627e-05, -1.8233044248192498e-06, -8.624222024432592e-09, 7.167644117506565e-10] | 1.1973e+12 |

+-----+-----+

+-----+

| P9 | [0.08182819594004698, -1.1144988173042776e-12, -2.1221735141377253e-14, 7.117142364365294e-11, -1.350291113154354e-10, -1.1484826900819208e-15, 1.0175914220071899e-15, 1.7228061923248557e-05, -1.616865971640658e-06, -6.052569948660328e-09, 6.517329816506141e-10] | 1.37517e+12 |

+-----+-----+

| P10 | [0.1079861028152531, -9.537554252631213e-13, -1.4245170290355798e-14, 1.0691928249886176e-10, -1.3324715966117508e-10, -9.73567547283569e-16, 1.1063352638434942e-15, 1.974553316935059e-05, -1.8433109471473784e-06, -5.54247974329572e-09, 7.366900833322951e-10] | 1.44804e+12 |

+-----+-----+

| P11 | [0.049584194321066574, -8.417233333457106e-13, -1.2240079851790676e-14, 6.322627420894349e-11, -1.7208603103935765e-10, -1.726479957230408e-15, 1.0223473664011302e-15, 2.1510197784053054e-05, -1.8281709328760395e-06, -6.503971930502921e-09, 7.43474332599339e-10] | 1.63375e+12 |

+-----+-----+

| P12 | [0.039347367016421685, -1.0251701189776464e-12, -1.1987914998237531e-14, 7.98581840555701e-11, -1.6042886640000432e-10, -1.4790440766084544e-15, 1.032917692527883e-15, 2.0459319469898664e-05, -1.8429501797105192e-06, -7.313093453673185e-09, 7.676339552141659e-10] | 1.68777e+12 |

+-----+-----+

| P13 | [0.036083247313724684, -1.0464078867179776e-12, -1.0399963427459427e-14, 8.062238010875007e-11, -1.6045276984993474e-10, -1.4695650250876697e-15, 1.0038833934025524e-15, 2.0909181676419384e-05, -1.8554297669451122e-06, -6.329816454662312e-09, 7.440457950419744e-10] | 2.55501e+12 |

+-----+-----+

| P14 | [0.0503503260471544, -8.409702267016169e-13, -1.2253993606609094e-14, 6.353295937992691e-11, -1.7178131548002063e-10, -1.7526421786717257e-15, 1.0231501313929065e-15, 2.1505999244907303e-05, -1.8282619543009747e-06, -6.4758191329940755e-09, 7.433898699002589e-10] | 3.61149e+12 |

+-----+-----+

| P15 | [0.04131070194697365, -1.0137281136236704e-12, -1.2537619253106434e-14, 7.956119588134497e-11, -1.6027922368878884e-10, -1.4822639726147034e-15, 1.0198386827802313e-15, 2.0158407531066398e-05, -1.835119911554764e-06, -7.953644694912195e-09, 7.44052119164044e-10] | 3.96911e+12 |

+-----+-----+

| P16 | [-0.13651054430864523, -1.3963410521029743e-12, -6.879295722420292e-14, 4.4647249443731137e-11, -1.810705202753829e-10, -1.3255787220317128e-15, 8.033357194586213e-16, 2.1783784847694232e-05, -1.867985038783987e-06, -9.032500349900794e-09, 7.773722531687259e-10] | 5.37481e+12 |

+-----+-----+

| P17 | [0.052109503022802, -9.88592321879981e-13, -9.641079583944745e-15, 8.146078666930978e-11, -1.5917227622127854e-10, -1.4946366828584041e-15, 1.0328911851232977e-15, 2.0077429165490923e-05, -1.8605135924128038e-06, -7.870489005802866e-09, 7.420414275829894e-10] | 1.07018e+13 |

+-----+-----+

| P18 | [0.06891531504574777, -1.0287766298900422e-12, -2.295364156375044e-14, 9.566412877535375e-11, -1.4190068085572775e-10, -1.021629545276754e-15, 1.033339735260536e-15, 1.9751652727977843e-05, -1.8387838481105752e-06, -6.366621706604675e-09, 7.608922359732318e-10] | 1.09511e+13 |

+-----+-----+

| P19 | [-0.1231814114127211, -1.3685625918839406e-12, -6.486444712886359e-14, 4.739660101476161e-11, -1.7931565820078816e-10, -1.3309186349763576e-15, 8.192868234514797e-16, 2.1738265132259e-05, -1.924557411128089e-06, -8.931429157474235e-09, 7.749416984076291e-10] | 5.32965e+13 |

+-----+-----+

+-----+-----+

+-----+-----+

| id | vec

|

+=====+=====

=====

=====+

| C0 | [0.08046122949040725, -1.1136075369327564e-12, -2.1141910346084682e-14, 7.125326979584275e-11, -1.3481529834970585e-10, -1.1537019686794007e-15, 1.02110463800243e-15, 1.6932113369328485e-05, -1.620334907075895e-06, -6.0957832884010104e-09, 6.528257520565375e-10] |

+-----+-----+

+-----+

| C1 | [0.04131070194697365, -1.0137281136236704e-12, -1.2537619253106434e-14, 7.956119588134497e-11, -1.6027922368878884e-10, -1.4822639726147034e-15, 1.0198386827802313e-15, 2.0158407531066398e-05, -1.8330095043826336e-06, -7.953644694912195e-09, 7.44052119164044e-10] |

+-----+-----+

+-----+

| C2 | [0.04141639993961189, -1.0117081002184981e-12, -1.2994472471499564e-14, 7.937378184676173e-11, -1.604137146808921e-10, -1.485052578582483e-15, 1.0198774026946e-15, 2.0174164573336113e-05, -1.8350397232297526e-06, -7.936364861853528e-09, 7.44044375654244e-10] |
+-----+-----+
-----+
| C3 | [0.06823780246382927, -1.0221175477528735e-12, -2.1290781917046326e-14, 7.584115282218133e-11, -1.7050722853261406e-10, -1.3877442966350558e-15, 1.0292409473099452e-15, 1.9955787895779288e-05, -1.8346863865063723e-06, -7.996385397921462e-09, 7.434762333810168e-10] |
+-----+-----+
-----+
| C4 | [0.047307171576636914, -1.028641880607479e-12, -1.3822842796136873e-14, 7.831953433061548e-11, -1.565422815171362e-10, -1.4328653270814911e-15, 1.0195060948127282e-15, 1.9724725022352982e-05, -1.8010208989186096e-06, -7.672291238512152e-09, 7.303891590503059e-10] |
+-----+-----+
-----+
| C5 | [0.0758317263103837, -1.099585050320469e-12, -1.9936511598346814e-14, 7.241308519438242e-11, -1.3876605348708806e-10, -1.197881335615133e-15, 1.017924009974693e-15, 1.7661744431961973e-05, -1.6488545771046821e-06, -6.3339234050603695e-09, 6.653959417643524e-10] |
+-----+-----+
-----+
| C6 | [-0.16435576860492573, -1.4472344418657165e-12, -7.525312526384461e-14, 4.040496250983625e-11, -1.825070995324125e-10, -1.3171243711389027e-15, 7.726132162571052e-16, 2.2032387515023957e-05, -1.8568035289374908e-06, -9.173409195756603e-09, 7.819820206643764e-10] |
+-----+-----+
-----+
| C7 | [0.09608302676010975, -9.71224157990131e-13, -1.4830613877404643e-14, 8.00834397560762e-11, -1.6907064927558445e-10, -1.3961986475278659e-15, 1.0599634505114613e-15, 1.9707185228449562e-05, -1.8320386057059081e-06, -7.855476552065654e-09, 7.388664658853663e-10] |
+-----+-----+
-----+
| C8 | [0.058948722056903415, -1.0352914299754738e-12, -2.2970620158658256e-14, 7.483608092348098e-11, -1.7053690331094799e-10, -1.4068310119976416e-15, 1.001000645151502e-15, 2.0029540158237374e-05, -1.8474624644817727e-06, -8.03177056563261e-09, 7.44722244661967e-10] |
+-----+-----+
-----+
| C9 | [0.04200731102029361, -1.0125029550460296e-12, -1.3330709518219429e-14, 7.922084273587569e-11, -1.6075490760977582e-10, -1.4824161913353151e-15, 1.019241178632269e-15, 2.016929013805896e-05, -1.8354584205026609e-06, -7.939580424923048e-09, 7.440672226168559e-10] |
+-----+-----+
-----+
| C10 | [0.06017089736830655, -1.0350725123967248e-12, -2.3160541794832443e-14, 7.476718374161647e-11, -1.7085118074656924e-10, -1.4030007524155717e-15, 1.0024601170255176e-15, 2.0019666957477703e-05, -1.8469235591449532e-06, -8.032184704665922e-09, 7.446530050264714e-10] |
+-----+-----+
-----+
| C11 | [0.06760653823310786, -1.023131320159154e-12, -2.1437097327592004e-14, 7.575711089315979e-11, -1.705341440258765e-10, -1.3889381689699577e-15, 1.0271452513735987e-15, 1.9960786661261805e-05, -1.8356439891161001e-06, -7.99918682195767e-09, 7.435683199791242e-10] |
+-----+-----+
-----+
| C12 | [0.04675959193503972, -8.06215227861274e-13, -1.0273942628424747e-14, 5.7586842867826295e-11, -1.773109534693819e-10, -1.88759349014419e-15, 1.0212644679405922e-15, 2.183065436915389e-05, -1.826314798075508e-06, -6.496026944671956e-09, 7.471645266659913e-10] |
+-----+-----+
-----+
| C13 | [0.0733443943817825, -1.0635316287303852e-12, -2.4933692541934787e-14, 1.0161024528745437e-10, -1.3637104286636652e-10, -8.884512970979168e-16, 1.0352253987128505e-15, 1.9426997603731255e-05, -1.840731004336042e-06, -6.3464138949267936e-09, 7.192180117506443e-10] |
+-----+-----+
-----+
| C14 | [0.07212255165639006, -1.0475442139120153e-12, -2.25528228169268e-14, 9.026358712234719e-11, -1.4033314472832754e-10, -1.0522294156163558e-15, 1.030632900878901e-15, 1.9127871329647785e-05, -1.7904552495101258e-06, -6.306702710719647e-09, 7.074692756147811e-10] |
+-----+-----+
-----+
| C15 | [0.07809233810368503, -1.0948399529107835e-12, -2.1542729092908323e-14, 7.665381144884932e-11, -1.3638283447710606e-10, -1.124875161633426e-15, 1.0238114723840652e-15, 1.7555894767658546e-05, -1.6686635056763444e-06, -6.155702284286038e-09, 6.683491449581332e-10] |
+-----+-----+
-----+
| C16 | [-0.10609006940494461, -1.3378938745441785e-12, -6.112191706133838e-14, 5.2481843031550575e-11, -1.721065488564285e-10, -1.3044979258338735e-15, 8.405097149287851e-16, 2.159093273730013e-05, -1.973721796819564e-06, -8.68086759531781e-09, 7.721135699061579e-10] |

+-----+-----+
-----+
| C17 | [0.09089476080747663, -9.844241426028832e-13, -1.7987700357881004e-14, 1.0257312437059314e-10, -1.3634960260968881e-10, -9.99988256426053e-16, 1.0851123723661886e-15, 1.9892865564309455e-05, -1.85372264504984e-06, -5.793041305452144e-09, 7.395182118337662e-10] |
+-----+-----+
-----+
| C18 | [0.04006435372888396, -1.0093345094905732e-12, -1.213208545649933e-14, 7.974238039524092e-11, -1.6116104134649535e-10, -1.495102108956243e-15, 1.0210239605860028e-15, 2.0229538594422983e-05, -1.8408343940248116e-06, -8.02548663668406e-09, 7.476055261661148e-10] |
+-----+-----+
-----+
| C19 | [0.07269105260546838, -1.124349556612149e-12, -2.2748082554371252e-14, 7.499936195931438e-11, -1.3807696358679556e-10, -1.1590274851292435e-15, 9.899959529291384e-16, 1.836748130479676e-05, -1.635995718611005e-06, -6.1448200887008836e-09, 6.545850214951098e-10] |
+-----+-----+
-----+

+-----+-----+
-----+-----+
| id | vec
| mut? |
+=====+=====+
=====+
=====+
| M0 | [0.08046122949040725, -1.1136075369327564e-12, -2.1141910346084682e-14, 7.125326979584275e-11, -1.277938941622218e-10, -1.1537019686794007e-15, 1.02110463800243e-15, 1.6932113369328485e-05, -1.620334907075895e-06, -6.0957832884010104e-09, 6.528257520565375e-10] | True |
+-----+-----+
-----+-----+
| M1 | [0.04131070194697365, -1.0137281136236704e-12, -1.2537619253106434e-14, 7.583106437505115e-11, -1.6027922368878884e-10, -1.4822639726147034e-15, 1.0198386827802313e-15, 2.0158407531066398e-05, -1.8330095043826336e-06, -7.953644694912195e-09, 7.44052119164044e-10] | True |
+-----+-----+
-----+-----+
| M2 | [0.04141639993961189, -1.0117081002184981e-12, -1.2994472471499564e-14, 7.937378184676173e-11, -1.604137146808921e-10, -1.485052578582483e-15, 1.0198774026946e-15, 2.0174164573336113e-05, -1.8350397232297526e-06, -7.936364861853528e-09, 7.44044375654244e-10] | False |
+-----+-----+
-----+-----+
| M3 | [0.06823780246382927, -1.0221175477528735e-12, -2.1290781917046326e-14, 7.584115282218133e-11, -1.7050722853261406e-10, -1.3877442966350558e-15, 1.0334693655839296e-15, 1.9955787895779288e-05, -1.8346863865063723e-06, -7.996385397921462e-09, 7.434762333810168e-10] | True |
+-----+-----+
-----+-----+
| M4 | [0.047307171576636914, -1.028641880607479e-12, -1.3822842796136873e-14, 7.831953433061548e-11, -1.565422815171362e-10, -1.4328653270814911e-15, 1.0195060948127282e-15, 1.9724725022352982e-05, -1.8010208989186096e-06, -7.672291238512152e-09, 6.806651566443516e-10] | True |
+-----+-----+
-----+-----+
| M5 | [0.0758317263103837, -1.099585050320469e-12, -2.131149086354666e-14, 7.241308519438242e-11, -1.3876605348708806e-10, -1.197881335615133e-15, 1.017924009974693e-15, 1.7661744431961973e-05, -1.6488545771046821e-06, -6.3339234050603695e-09, 6.653959417643524e-10] | True |
+-----+-----+
-----+-----+
| M6 | [-0.16847719707035366, -1.4472344418657165e-12, -7.525312526384461e-14, 4.040496250983625e-11, -1.825070995324125e-10, -1.3171243711389027e-15, 7.726132162571052e-16, 2.2032387515023957e-05, -1.8568035289374908e-06, -9.173409195756603e-09, 7.819820206643764e-10] | True |
+-----+-----+
-----+-----+
| M7 | [0.09608302676010975, -9.71224157990131e-13, -1.4280548456574188e-14, 8.00834397560762e-11, -1.6907064927558445e-10, -1.3961986475278659e-15, 1.0599634505114613e-15, 1.9707185228449562e-05, -1.8320386057059081e-06, -7.855476552065654e-09, 7.388664658853663e-10] | True |
+-----+-----+
-----+-----+
| M8 | [0.058948722056903415, -1.0352914299754738e-12, -2.2970620158658256e-14, 7.483608092348098e-11, -1.7053690331094799e-10, -1.4068310119976416e-15, 1.001000645151502e-15, 2.0029540158237374e-05, -1.8474624644817727e-06, -8.03177056563261e-09, 7.796305507412573e-10] | True |
+-----+-----+
-----+-----+
| M9 | [0.04200731102029361, -1.0125029550460296e-12, -1.3330709518219429e-14, 7.922084273587569e-11, -1.6075490760977582e-10, -1.4663360178858256e-15, 1.019241178632269e-15, 2.016929013805896e-05, -1.8354584205026609e-06, -7.939580424923048e-09, 7.440672226168559e-10] | True |

+-----+-----+
-----+-----+
| M10 | [0.06017089736830655, -1.0350725123967248e-12, -2.3160541794832443e-14, 7.476718374161647e-11, -1.7085118074656924e-10, -1.4795284563207368e-15, 1.0024601170255176e-15, 2.0019666957477703e-05, -1.8469235591449532e-06, -8.032184704665922e-09, 7.446530050264714e-10] | True |
+-----+-----+
-----+-----+
| M11 | [0.06760653823310786, -1.023131320159154e-12, -2.008187650622384e-14, 7.575711089315979e-11, -1.705341440258765e-10, -1.3889381689699577e-15, 1.0271452513735987e-15, 1.9960786661261805e-05, -1.8356439891161001e-06, -7.99918682195767e-09, 7.435683199791242e-10] | True |
+-----+-----+
-----+-----+
| M12 | [0.04675959193503972, -8.06215227861274e-13, -1.0273942628424747e-14, 5.7586842867826295e-11, -1.773109534693819e-10, -1.88759349014419e-15, 1.0212644679405922e-15, 2.183065436915389e-05, -1.826314798075508e-06, -6.496026944671956e-09, 7.781981385240979e-10] | True |
+-----+-----+
-----+-----+
| M13 | [0.0733443943817825, -1.0635316287303852e-12, -2.4933692541934787e-14, 9.740520457102684e-11, -1.3637104286636652e-10, -8.884512970979168e-16, 1.0352253987128505e-15, 1.9426997603731255e-05, -1.840731004336042e-06, -6.3464138949267936e-09, 7.192180117506443e-10] | True |
+-----+-----+
-----+-----+
| M14 | [0.07212255165639006, -1.0570273304718467e-12, -2.25528228169268e-14, 9.026358712234719e-11, -1.4033314472832754e-10, -1.0522294156163558e-15, 1.030632900878901e-15, 1.9127871329647785e-05, -1.7904552495101258e-06, -6.306702710719647e-09, 7.074692756147811e-10] | True |
+-----+-----+
-----+-----+
| M15 | [0.07809233810368503, -1.0948399529107835e-12, -2.1542729092908323e-14, 7.665381144884932e-11, -1.3638283447710606e-10, -1.083566542682868e-15, 1.0238114723840652e-15, 1.7555894767658546e-05, -1.6686635056763444e-06, -6.155702284286038e-09, 6.683491449581332e-10] | True |
+-----+-----+
-----+-----+
| M16 | [-0.10609006940494461, -1.3378938745441785e-12, -6.112191706133838e-14, 5.2481843031550575e-11, -1.721065488564285e-10, -1.3044979258338735e-15, 8.405097149287851e-16, 2.159093273730013e-05, -1.973721796819564e-06, -8.68086759531781e-09, 8.189069353528707e-10] | True |
+-----+-----+
-----+-----+
| M17 | [0.09089476080747663, -9.844241426028832e-13, -1.7987700357881004e-14, 1.0257312437059314e-10, -1.3634960260968881e-10, -9.99988256426053e-16, 1.0653460550547298e-15, 1.9892865564309455e-05, -1.85372264504984e-06, -5.793041305452144e-09, 7.395182118337662e-10] | True |
+-----+-----+
-----+-----+
| M18 | [0.04006435372888396, -1.0093345094905732e-12, -1.213208545649933e-14, 7.974238039524092e-11, -1.6116104134649535e-10, -1.495102108956243e-15, 1.0210239605860028e-15, 2.0229538594422983e-05, -1.8408343940248116e-06, -8.474234099126071e-09, 7.476055261661148e-10] | True |
+-----+-----+
-----+-----+
| M19 | [0.07269105260546838, -1.124349556612149e-12, -2.3451345780343743e-14, 7.499936195931438e-11, -1.3807696358679556e-10, -1.1590274851292435e-15, 9.899959529291384e-16, 1.836748130479676e-05, -1.635995718611005e-06, -6.1448200887008836e-09, 6.545850214951098e-10] | True |
+-----+-----+
-----+-----+

.==.