All the symbols B, L, D, N, T, M should be interpreted like they were defined in a problem

Our approach combines two methods, initial solution is selected greedily and then optimized by genetic algorithm. Solution is encoded as permutation of libraries in order of registering. When the solution needs to be printed or evaluated, every library in permutation scans the best books it can scan, that were not scanned by other libraries

1. Greedy preselecting:

In order to select the best libraries to be registered we needed somehow evaluate libraries. However, the contribution of the books scanned in library to the final score depends on the day in which library was registered and started scanning books. That's why we decided to add libraries to the solution one by one, every time revaluating their evaluation values and choosing the new library with the best value up to the moment when the sum of registration times of libraries in the solution will be greater or equal D (**REMAINING_DAYS_FOR_SCANNING** <= 0) or all the libraries will be jet added to the solution.

In evaluation function we wanted to include the approximated gain which library will generate by scanning books. However it also very important to remember that, the more time it takes to register the library the less time is left for other later registered libraries to scan books and some of them may not finish scanning for example on a picture below, if we decide to register library 2 first it will scan 12 books (gain) but also will disable scanning of 7 other books by libraries 0 and 1 (what is basically its cost)
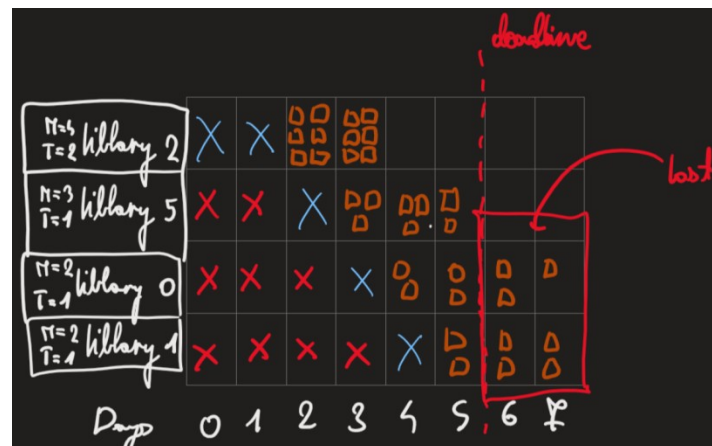
So the evaluation formula of library should look like that:

EVALUATION = GAIN – COST

To estimate gain of library we use this formula:

GAIN = (REMAINING_DAYS_FOR_SCANNING – T) *AVG_SCORE*M

Where:

REMAINING_DAYS_FOR_SCANNING = D – sum of T of all libraries in solution
AVG_SCORE – average score of books in a library, it is weighted by the probability of a book not being scanned = (1 – frequency / B)

And to estimate cost:

COST = LOSS*T

Where **LOSS** can be interpreted as average expected score that can be gained in one day of scanning. That's how we estimate it:

LOSS = (D/AVG_T)*AVG_M*AVG_GLOBAL_BOOKS_SCORE

Where again values:
- **AVG_T**
- **AVG_M**
- **AVG_GLOBAL_BOOKS_SCORE**
Can be easily and quickly computed and are respectably:
- average value of T in all libraries
- average value of M in all libraries
- average score of book in library (considering all books in all libraries)

This mathematical formulas of course do not precisely describe values which we wanted to calculate. In order to increase precision more computations is needed what also increases execution time. The aim was to create a greedy algorithm to create any reasonable solution for genetic algorithm to optimise it.

2. Genetic algorithm:

The genetic algorithm creates set of **POPSIZE** encoded solutions where one solution is generated by greedy and the rest **POPSIZE**-1 is generated randomly. Then it evaluates them and creates population consisting of also POPSIZE solutions selecting them from that set using Stochastic Universal

Sampling. However, the best element of the set is always included in the population to maintain elitism. After creating initial population it performs mutations and crossovers, which increases the size of population, evaluates all the solutions and chooses best **POPSIZE** of them to create next population. This step is repeated until there is 1000 iterations where the best element did not change or too much time passes. Mutation swaps places of two random libraries in the solution or substitutes one random library outside the solution for some library in the solution. Before crossover the population is shuffled so the solutions can be mutated in pairs. As the crossover operator we use Alternating Position Crossover, because solutions are encoded as permutations where no library can repeat, and are of different lenghts.
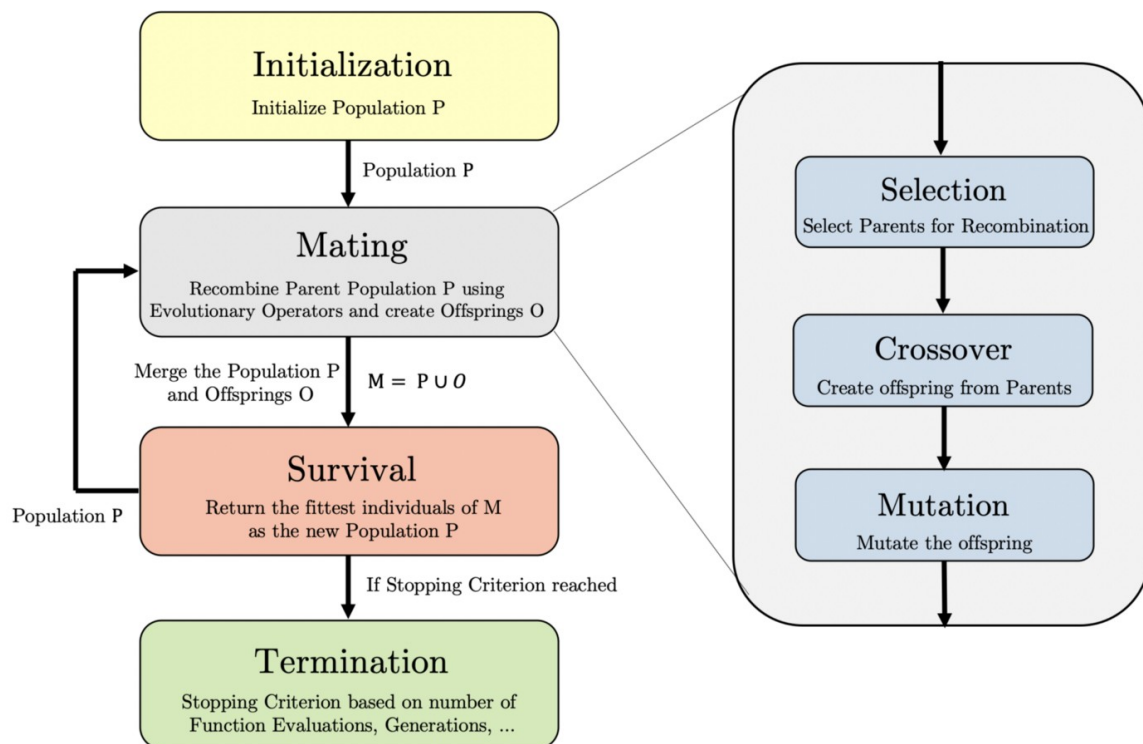
Encoding:

Combination of libraries up, whose time does not exceede the **DAYS**. There is also a set of libraries that are not included in this combination. One library from that set is sometimes swapped with a library in the proper solution during mutation.

3. Sources:

- https://www.researchgate.net/publication/226665831_Genetic_Algorithms_for_the_Travelling_Salesman_Problem_A_Review_of_Representations_and_Operators
- https://towardsdatascience.com/an-introduction-to-genetic-algorithms-the-concept-of-biological-evolution-in-optimization-fc96e78fa6db