Combinatorial optimization project – theoretical part

*Jerzy Pawlik 151964 and Łukasz Andryszewski*

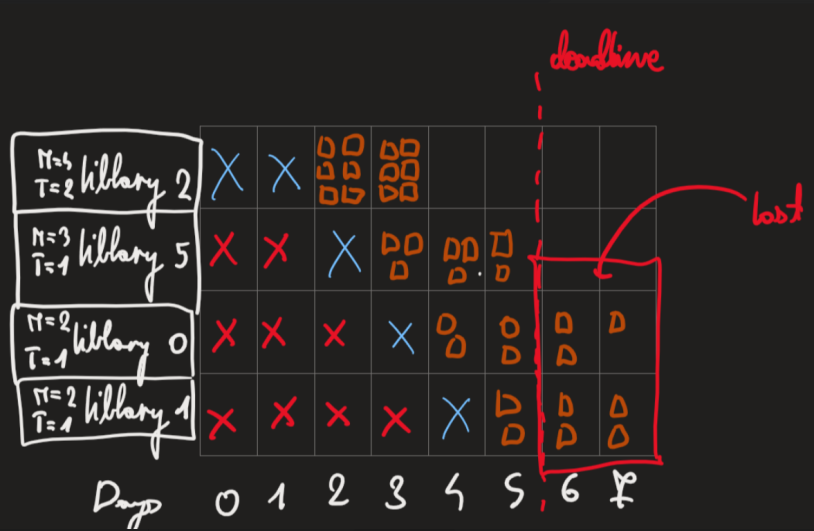
All the symbols B, L, D, N, T, M should be interpreted like they were defined in a problem

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 ofT of all libraries in solution  
**AVG\_SCORE –** average score of books in a library. We calculate it when we read data so it does not take too much time.

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.

1. Genetic algorithm:

The genetic algorithm creates set of 200 encoded solutions where one solution is generated by greedy and the rest 199 is generated randomly. Then it evaluates them and creates population consisting of also 200 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, what increases the size of population, evaluates all the solutions and chooses best 200 of them to create next population. This step is repeated until there is 1000 populations where the best element is the same or too much time passes. Mutation just swaps order of two random libraries in the solution or substitutes one random library outside the solution for some library in the solution. As the crossover we use alternating position crossover, because solutions are encoded as permutations where no library can repeat.

1. Sources:

* https://www.researchgate.net/publication/226665831\_Genetic\_Algorithms\_for\_the\_Travelling\_Salesman\_Problem\_A\_Review\_of\_Representations\_and\_Operators