INTRODUCTION

The logistic planning in city transit consist of two stages, in turn: Strategic Planning and Operational Planning. The second one splits into three phases:

- Vehicle Scheduling,
- Duty Scheduling,
- Crew Rostering.

The analyzed problem concerns the third phase that is creating optimal schedule of transportation task assignment to drivers (vehicle operators). A task assigned to a driver forms so called a duty. In practice of city public transit creating the long term roster manually presents a real challenge [8,10]. Commonly met problems apply to data bases of several hundred drivers and duties.

Usually schedule covers monthly period – so the number of decision variables is large as in the example shown in Figure 1. Next problem results from large number of constraints, these which disable assignments as well as those which must be included in the costs function. Moreover relations between duties are very complex along the whole schedule which cause that small modification in one day assignment may change conditions in other days.

The aim of the optimization of transportation task assignment is to achieve such a multiday project of driver roster [1,5,11,12] which is of the lowest operational cost and at the same time in agreement with the law and the internal company rules. The topic is known in literature as CRP - Crew Rostering Problem [11,13]. The paper presents the mathematical model of CRP and practical solutions given by the algorithms elaborated by the author.
1 MATHEMATICAL MODEL

The estimation if the analyzed roster meets requirements is assessed by the value of the cost function. A driver is recognized by the identification number \( k_j \), a day by the date \( d_k \) and a duty by the identification number \( s_i \).

The result given back by the cost function presents the cost of assigning the duty \( s_i \) to the driver \( k_j \) in the day \( d_k \). The higher is the cost value the worse is that assignment.

This kind of estimation is done in two stages. At first stage the potential assignments which are not in conformity with the law are treated as unacceptable and a blocking cost \( T \) of very high value is imposed to them. At the second stage the cost of every permitted assignment is calculated.

The cost function is defined as follows:

\[
P(x_{ij}) = \begin{cases} T & \text{if at least one of the constraint condition is met,} \\ \sum_{l=1}^{10} w_l p_l(x_{ij}) & \text{in all other cases,} \end{cases}
\]

where:

- \( P_{ij} = P(x_{ij}) \) – is the cost of assignment of the single duty \( s_i \) to the driver \( k_j \)
- \( x_{ij} \in X_{n \times m} \) – \( x_{ij} \) is the decision variable representing assignment of the duty \( s_i \) to the driver \( k_j \)
- \( X_{n \times m} \) – is the the matrix of decision variables
- \( T \gg \max \{ w_l p_l(x_{ij}) \} \) – \( T \) is a very high blocking cost
- \( p_l(x_{ij}) \) – is the cost factor calculated separately for each assignment \( x_{ij} \)
- \( w_l \) – is the weight determining influence of the particular factor \( p_l \) on the cost. \( w_l \) has the meaning of the unit cost per minute

- \( n \) – is the number of all duties in the day (or period)
- \( m \) – is the number of all drivers in the day (or period).

The decision variables take the binary values:

\[
x_{ij} = \begin{cases} 0 & \text{if the duty } s_i \text{ is not assigned to the driver } k_j \\ 1 & \text{if the duty } s_i \text{ is assigned to the driver } k_j \end{cases}
\]

Each duty must be assigned to only one driver, which corresponds to the condition (3):

\[
\sum_{j=1}^{m} x_{ij} = 1 \quad \text{for } i=1,2,...,n
\]

The cost of the roster \( P_{\text{roster}} \) equals the sum of the costs of all duty assignments realized in the schedule:

\[
P_{\text{roster}} = \sum_{i=1}^{n} \sum_{j=1}^{m} P_{ij} x_{ij}
\]

The optimization presents the specific set covering problem with minimization of the objective function (4). As \( p_l \) factors in the cost function (1), various costs may be considered depending on the law and company internal regulations, for example:

1) overtime in accounting period (the absolute difference between standard/norm and the driver working time in minutes),
2) undertime in accounting period (the absolute difference between standard and the driver working time in minutes),

\[
P_{ij} = P(x_{ij})
\]
3) lack of duty/driver preference (from 0 of the highest preference to 10 of the lowest preferences, multiplied by 48),
4) incorrect duty (480 when a duty has a planned shift, a driver has a planned shift in that day, and these shifts are different, in other case 0),
5) postponement of work start time (difference between duty start time and duty start time in the they before or midnight),
6) lack of vehicle authorization for the duty (driver default vehicle has no authorization for the duty).
7) overlap of duties of co-drivers,
8) excessive breaks (duty causes excessive brakes at work of a driver in a day),
9) repetition of duties (the same duty for a driver the day before),
10) lack of driver authorization to a duty.

The constraint conditions which are the blocking criteria for assignments may be considered as follows:
1) Duties overlap (in case when a duty overlaps the duty assigned to driver before),
2) Overtime hours in particular day (driver has already assigned duty and assigning him another one causes overtime),
3) Overrun of driver working time in a particular day (addressing duty causes overrun in working time from the last day-off),
4) Overrun of working time in a following day (addressing duty causes overrun in working time till following day-off),
5) Cancellation of day-off (even though a driver has planned day-off),
6) Transgressions in weekly rest (addressing duty to driver will cause transgressions in weekly rest according to law regulations),
7) Transgressions in daily rest (assigning duty to driver causes transgressions in daily rest according to law regulations),
8) Impossible approach (driver cannot approach from a place of duty termination to a starting place of a new duty),
9) Overlap of duties of co-drivers (optional criteria, when a duty overlaps another duty which are assigned to co-drivers),
10) Excessive breaks (optional criteria; assigning duty to a particular driver causes excessive breaks in his daily working time),
11) Repetition of duty (optional criteria; the same duty addressed to the same driver a day before)
12) Lack of authorization of driver to duty (optional criteria; a driver has no authorization to that duty)

2 PROPOSED METHODS OF SOLUTION

Solving the mathematical model the multi-objective function (4) is minimized regarding several constraints as above. The model presents specific set covering problem as shown in Figure 2. The set of duties and drivers are usually not equal in numbers, so more than one duty can be assigned to one driver. Than relations between sets are not one to one. It has to be assumed that there are groups of duties which can be assigned only to specific groups of drivers. Moreover there are duties which are combined together and have to be assigned to combined co-drivers. The above leads to so-called set partitioning. In practice very often different cost function has to be applied for different set partitions. The above requirements lead to the situation that standard methods of linear programming for set covering problems cannot be applied and specific algorithms have to be used. The author formulated and performed the comparative tests of the following three algorithms.

2.1 HLN Algorithm

HLN algorithm was elaborated in accordance with Kuhn – Munkers linear method, called also Hungarian method [3], designed for finding those components of cost matrix, which sum is the lowest.
The dimension of square cost matrix and the assignment matrix must be equal. We assume that from each column and row there is exactly one element chosen. This methods depends on minimal node covering in bi-partite graph that leads to finding independent zeros in cost matrix. In case there is not enough zeros, Gauss elimination method must be applied to the cost matrix. The description of HLN algorithm has been presented in details in [3].

The problem related to Hungarian method, which is a linear method, is the necessity of priory calculation of complete cost matrix, in case when some assignments may influence the value of cost function of other assignments. This problem has been solved in HLN algorithm by making partitions of duties and drivers sets in the following courses of algorithm, so that in each single course there are possible only these assignments, which do not influence one another on their costs.

2.2 GRIT Algorithm

The elaborated GRIT method is based on application of so called “greedy algorithm”. GRIT algorithm relies on choosing the best driver for particular duty and assigning him right away to the duty, if it is in accordance with law regulations. It is equivalent to dynamic creating of cost matrix row each time, when we try to address a duty. Such approach allows to avoid problems connected with influence of already done assignments on cost matrix elements, because all crucial elements of the cost matrix are calculated directly before choosing the smallest of them. The description of GRIT algorithm has been presented in detail in [10].

2.3 GA Algorithm

The third in turn genetic algorithm GA was elaborated in accordance with classical mechanism of genetic methods [6]. Genetic algorithms are methods designed for solving optimization problems based on natural evolution. Those are nondeterministic procedures of searching based on natural selection, crossover and rarely occurred mutations. The detailed description of GA algorithm has been presented in [9].

The course of GA genetic algorithm is as follows:

A. Population creation

Creating initial population of individuals. Usually the first set of individuals is random including only the most important limitations. The population creation is based on the random assignment of
drivers to duties. It is important that the created individuals are allowed – meaning the same duty cannot be assigned to two drivers and each duty must be assigned in the roster.

B. Selection

The selection process is supposed to strengthen good individuals at the cost of weak individuals. In order to estimate all individuals in the population it is necessary to use the cost function. For each individual (vector) the cost of daily schedule is calculated by summing up values returned by the cost function for all duties addressed to drivers in the particular day according to (4). Than using well known roulette wheel selection method it is estimated which individuals will survive, which will be strengthen and which will disappear from the next population.

C. Crossover

The main aim of crossover [6,12] is to create the new individuals using features of already existing individuals in the population. The crossover is done after the selection process, so there is high probability that the new descendent will receive all good features from both parents. Of course, there is also possible that the new descendent will be worse from both parents, however the next selection will exclude such individuals. To receive desired results from crossover it is important to choose the crossover operator $q_c$ appropriate for the problem.

D. Mutation

The mutation in evolution is done in order to diversify population [6,12]. It is necessary for correct algorithm operation as it introduces new individuals, which would not be created in crossover process. The mutation may create individuals which are not better adapted, however worse individuals are than eliminated in the selection process. In case of the roster optimization it is very important to receive in the process of mutation only permitted individuals. To receive desired results from mutation it is important to choose the mutation operator $q_m$ appropriate for the problem.

E. Checking ending condition

If the ending condition is not met, it comes to point B. The ending condition may be defined as: reaching preset number of generations or reaching state when 80% of individuals are the same as the best individual.

3 TESTS RESULTS

The three proposed algorithms have been tested and results subjected to comparison analysis. The lately research has been carried out based on the real data from the city transit companies of certain cities of Poland, among others: Warszawa, Wrocław, Katowice, Gliwice, Tychy, Białystok, Rzeszów, Gdynia.

The broad tests have been performed to select appropriate parameters for genetic algorithm, including number of individuals in population, probability of crossover and mutation, number of generations, as well as to compare the results of the GA algorithm with the results of HLN and GRIT algorithms.

In each of chosen exemplary comparisons bellow, the cost of roster and the calculation time has been analyzed on the real data from city of Białystok transit. The results shown in figures below are average values obtained from 6 independent runs of each algorithm.

- **Test 1**
  - Monthly roster, 4 calculation threads.
  - Working day – 177 tasks, Saturday – 102 tasks, holiday – 79 tasks, 229 drivers.

Fig. 3. Test 1 results
Results of that test show that GA, even though it takes more time, gives the roster with the lower costs. HLN is the fastest calculating algorithm, however the roster made by this algorithm is slightly worse than others. Results from GRIT algorithm are between GA and HLN in relation to the cost and calculating time.

- Test 2
  Monthly roster, 2 calculation threads.
  Working day – 168 tasks, Saturday – 97 tasks, holiday – 63 tasks, 229 drivers.

![Fig. 4. Test 2 results](image)

Test 2 is another proof of the fact that GA algorithm creates the best roster with the lowest (comparing to other algorithms) cost. It also has assigned much more duties, however the calculating time is the longest. The GA algorithm has been stopped when it met „maximum number of generations”.

- Test 3
  Monthly roster, 4 calculation threads.
  Working day – 179 tasks, Saturday – 120 tasks, holiday – 80 tasks, 229 drivers.

![Fig. 5. Test 3 results](image)

Test 3 shows that GA is still the best in creating roster. However it is important to pay attention that comparing to earlier tests in that case GA algorithm is better about less than 1% from GRIT. In that case GRIT has created very good roster and the calculating period was 10 times less than GA. HLN has found the best roster 120 times faster than GA.

- Test 4
  Weekly roster, 4 calculation threads.
  Working day – 179 tasks, Saturday – 120 tasks, holiday – 80 tasks, 229 drivers.

![Fig. 6. Test 4 results](image)
According to that test the best roster have been found by GRIT algorithm, and much faster than GA.

The tests were performed on the computer of the following parameters: 2*2 core processor Intel Xeon 1,6GHz, 4 GB RAM, 4 calculation threads.

**SUMMARY**

In most tests the genetic algorithm allowed to find the best roster. However all compared algorithms gives the results similar in cost. In all performed tests the difference between algorithms in cost function is not higher that 10%. Considering time of calculation hundreds times better results are achieved by the HLN algorithm.

Genetic algorithms are of high sensitivity to value of crossover and mutation probability. The recommended range of these probabilities for the presented algorithm have been estimated in the tests as:

\[ q_c = 0.9 \quad \text{– crossover probability}, \]
\[ q_m = 0.005 \quad \text{– mutation probability} \quad [9]. \]

It is necessary to save the best individuals during the process, as not always following generations are as good as previous and loss of good results may happen. It is also important to remember that the genetic algorithm is based on accidental choosing and probability of receiving good result with little population and low number of iterations is small. The tests proved that recommended number of individuals in population is in the range of 20 to 30. The calculations were stopped while the maximum number of generations was reached - in the tests usually 20000, or while maximum number of generations without the change of the best individual was reached - in the tests usually 1000 [9].

Genetic algorithm has typical adaptation features, which adjust to initial data set and allow to find rare solutions, that cannot be found by deterministic HLM algorithm. The disadvantage of genetic algorithm is very long calculation time.

Modified linear HLM Hungarian algorithm allows to determine each roster in much shorter time than other algorithms. HLN is deterministic method and it may be performed just once, however GRIT and GA methods depend on initial data, so their results of roster may differ each time on the same data.

The results of optimization have been so good that all three proposed algorithms have been included in commercial software system OptiGraph®, designed for public transport roster optimization [7,8].

The application of OptiGraph in many large cities of Poland and other (ex. Kuala Lumpur, Kuantan, Malaysia in 2014) has proved the usefulness of elaborated algorithms for large scale roster optimization in city public transport.

**Abstract**

In the work the mathematical model of crew rostering problem has been formulated. Specific demands of city public transit have been considered in formulating the multi-objective function and the constraints. The complexity of the mathematical model results from large number of decision variables, complex relations and number of constraints. Three original algorithms have been proposed and elaborated for optimization of crew roster: modified deterministic Hungarian algorithm, stochastic greedy algorithm and classical genetic algorithm. Comparison tests and analysis of the algorithms have been performed on large data sets from several cities of Poland. The results have been good and proved the usefulness of the algorithms to application in operational planning of city public transit.
Optymalizacji harmonogramów służb kierowców: zmodyfikowany deterministyczny algorytm węgierski, stochastyczny algorytm zachłanny i klasyczny algorytm genetyczny. Testy porównawcze i analiza algorytmów została przeprowadzona na wielkich zbiorach danych z kilkunastu miast polskich. Rezultaty były pozytywne i wykazały użyteczność algorytmów w planowaniu operacyjnym publicznej komunikacji miejskiej.

REFERENCES