KAUNAS UNIVERSITY OF TECHNOLOGY PANEVEZYS INSTITUTE
PANEVEZYS TECHNOLOGY AND SCIENCE PARK
INTELLIGENT TRANSPORT SYSTEMS, POLAND
TALLINN UNIVERSITY OF TECHNOLOGY
RIGA TECHNICAL UNIVERSITY

INTELLIGENT TECHNOLOGIES IN LOGISTICS AND MECHATRONICS SYSTEMS

ITELMS'2011

PROCEEDINGS OF THE 6TH INTERNATIONAL CONFERENCE

EDITED BY Z. BAZARAS AND V. KLEIZA

May 05-06, 2011
Panevėžys, Lithuania

TECHNOLOGIJA
KAUNAS·2011

Proceedings of 6th International Conference ITELMS'2011

Algorithm for Increasing Traffic Capacity of Level-Crossing Using Scheduling Theory

I. Alps, M. Gorobetz, A. Levchenkov

Riga Technical University, Riga, Latvia, E-mail: ivars.alps@rtu.lv

Abstract

In this paper authors present heuristics algorithm for level-crossing traffic capacity increasing. The genetic algorithm is proposed for this task solution. To create control centre and installed embedded intelligent devices on railway vehicles are proposed to control its motion speed and operate with level-crossing barriers. Algorithm is tested using computer. Results of experiments show big promises for rail transport schedule fulfilment and level-crossing traffic capacity increasing using proposed algorithm.
KEYWORDS: *genetic algorithm, intelligent transport system, level-crossing control, rail transport.*

1. Introduction

Nowadays in cities number of vehicles is increasing day by day. Traffic jams are the main reason for a lot of problems for public transport like delays, inefficient usage of energy etc. Fulfillment of the schedule in such conditions is unforeseen and service level of public transport is going down. In cities, where mixing the road and rail transport, is often to seen conflicts between these types of transport. Road transport is forced to wait at the closed level crossing while the train goes pass level crossing. This downtime negative impact on urban transport and is a one of the source of public transport delays. In Riga such a conflict place for many years is Sarkandaugava's level crossing, between the stations Mangali and Sarkandaugava. Morally and physically outdated equipment and methods are still used to control this level crossing. Since this railway line are busy with passengers and rolling stocks transport, traffic jams in this part of city is significant and the current situation is disaffecting as for Railway Company, for City council as well.

On the one hand the city traffic requests to ensure a continuous traffic flow on the streets, on the other hand rail transport have to ensure passengers and cargo in accordance with the schedule. Since the movement of these various transport systems organized by various organizations, in practice possible often conflicts, which mode of transport should be the priority? Therefore some solutions to fulfill the schedule are necessary. There are some different ways to reach the goal.

The first possible solution is to organize railway transport flow at night, when the road transport unit's amount has fallen to a minimum. This method is not acceptable, since rail transport should be followed the schedule.

Another solution is to build a bridge over railway, but this project implementation will take lot of time and resources. Therefore this solution increase expenses of City council and railway transport companies and this project not to confirm in current economical situation.

In previous papers authors proposed several solutions such as optimal speed and schedule control [8][9] and "green wave" [10], which allows the transport units to switch the traffic lights and transport units moving on the route without any additional braking on the traffic lights, thereby taking into account the schedule.

In this paper new improved algorithm for rail-crossing capacity increasing and schedule completion [3] is proposed.

Optimal train traffic organization on the given rail line is very important to reduce vehicle standby time, i.e. increase the level crossing capacity. The existing control system analyzing shows, that the level crossing is closed before the deadline, required by the security rules. Analysis of timetable shows, that according the existing schedule the trains, going in opposite directions, passing through the rail-crossing one by one. Due to this lack the level-crossing total time in a closed position is longer than might be.

2. Problem formulation

Following hypothesis is run for problem solving: the existing schedule adjustment could reduce the level-crossing time position in a closed position.

If railway transport units moving on the route appropriate optimal schedule this could decrease city transport standby time on railroad crossing, increase railroad crossing traffic capacity, eliminate often braking and acceleration, finally save energy [12].

The purpose of this research is to create algorithm for rail transport moving control on the route according predefined schedule.

Object of research is the city and rail transport system.

Main tasks of research are:

- to define structure of rail transport with built-in intelligent devices;
- to create mathematical model for existing control system;

- to develop algorithm for schedule fulfilment for rail transport;
- to create computer model of proposed system.

Proposed rail transport control system consists of control centre, trains and level-crossings with built-in intelligent devices as shown in Fig.1. Control centre receive signals from each railway transport units about current location, moving speed and other parameters, according predefined schedule, data and algorithms control centre calculate and send through GSM transmitter relevant signals to each transport units about optimal moving speed and for each level-crossing closing units about optimal closing and opening time.

The fragment of existing railway transport movement schedule is shown in Fig.2.

This scheme describes correlations between movement timetable and object location. Level-crossing location in this scheme is market as line "level-crossing". As can be seen in time area A, railway transport units numbered as 6108 and 6109 moving in different directions cross the rail-crossing one by one.

Such a motion algorithm is reason for irrationally waste of time. Judging from the scheme as soon as the passengers train No.6108 has crossed the level-crossing and it could be opened, from the other direction is already approaching passenger train No.6109 and the level-crossing barrier must be closed again. No car or truck can cross the level-crossing and traffic jam on this point only increasing. Level-crossing barrier total time staying in close position graphically could be expressed as shown in Fig.3.

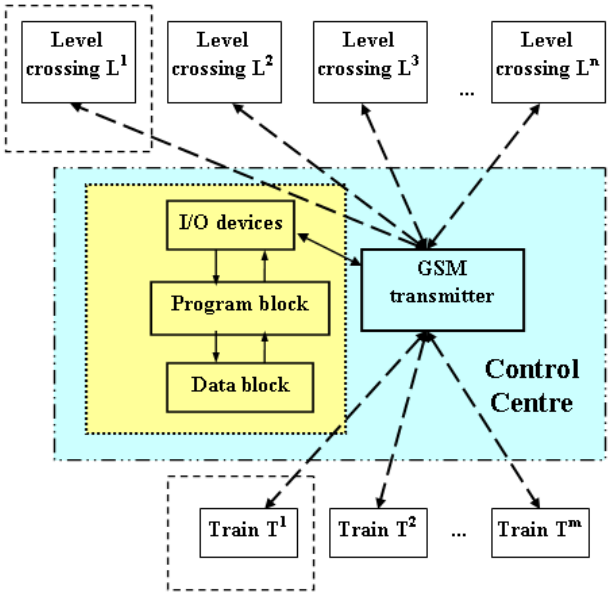

Fig. 1. Rail transport system control scheme

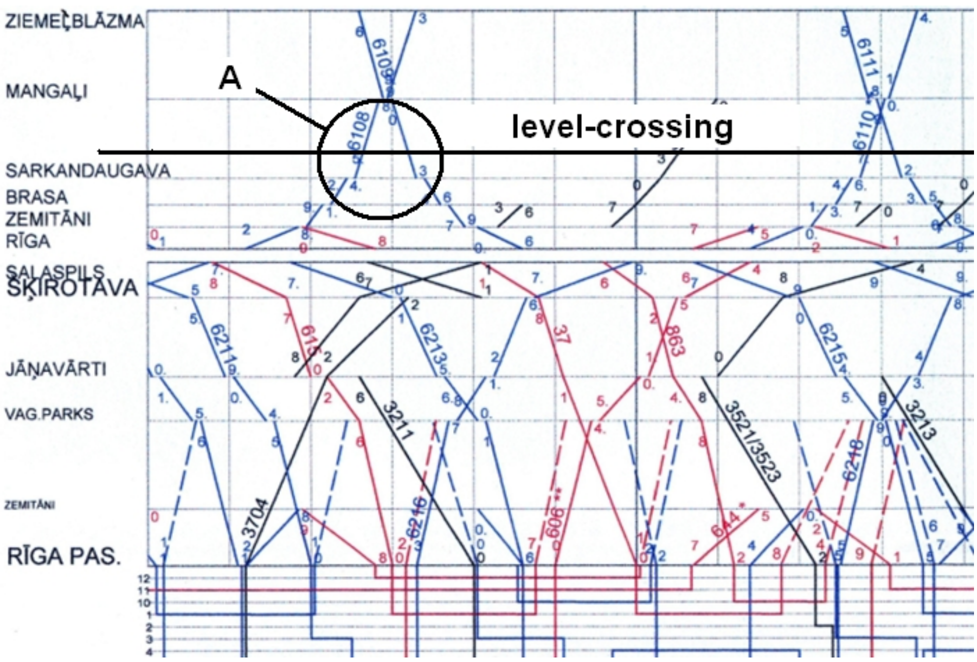

Fig. 2. Existing railway transport movement schedule

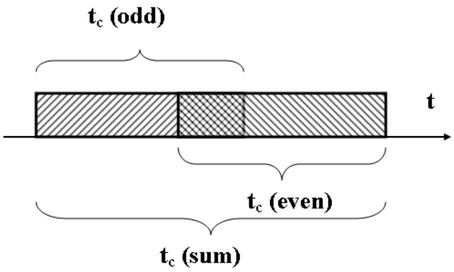

Fig. 3. Level-crossing barrier total time staying in close position

As can be seen from this scheme barrier staying in close position total time $t_c(sum)$ for couple of railway transport units is expressed as following:

$$t_c(sum) = t_c(odd) + t_c(even)$$

One of the scheduling theory methods for total schedule fulfillment decreasing is timetable rearrangement method. In case of railway transport it means to change departure time from previous station. Since train in odd direction arrive in level-crossing area little later or/and train in even direction arrive in level crossing area little earlier, the result of such timetable rearrangement could significantly decrease barrier staying in closed position total time as shown in Fig.4.

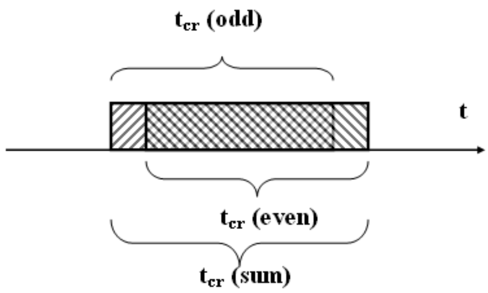

Fig. 4. Level-crossing barrier total time staying in close position for rearranged timetable

In case of rearranged timetable total time t_{cr}(sum) is expressed as following:

$$t_{cr}(odd) + t_{cr}(even) = t_{cr}(sum) < t_c(sum)$$

3. Mathematical model for algorithm

Following objects are given:
$ST = \{ST_1, ST_2, \ldots, ST_k\}$ is set of stations;
$LC = \{LC_1, LC_2, \ldots, LC_m\}$ is set of level crossings, where each level crossing is between two stations ST_i and ST_j;
$RS = \{RS_1, RS_2, \ldots, RS_n\}$ is set of rolling stocks;
$SCH = \{ Td_{ST}^{RS_1},\ Td_{ST}^{RS_2},\ \ldots,\ Td_{ST}^{RS_n} \}$ is set of scheduled departure time for each rolling stock at the station ST before level-crossing (depends on movement direction).
Infrastructure constants:
S_{ST_i, ST_j} is distance between stations ST_i, ST_j (m);
$S_{ST, LC}$ is distance from station ST to level-crossing LC (m);
$t^{calc}{}_{clos}$ is minimal time to close the level-crossing (s);
d is directions: $d \in \{odd, even\}$;
$R^d = \{R1, R2, \ldots\}$ is set of routes in each direction d;
Sc_R^d is distance from level-crossing to closure control point for each route R in each direction d (m);
So_R^d is distance from level-crossing to opening control point for each route in each direction (m);
tc_R^d is time delay between crossing the closure control point and level-crossing closure for each route in each direction (s);
to_R^d is time delay between level-crossing closure and route opening for each route in each direction (s);
Sst_R^d is distance from opening control point to the station for each route in each direction (m).

Rolling stock RS constants:
id^{RS} is id number;
t^{RS} is type;
d^{RS} is direction;
R^{RS} is route;
Lw^{RS} is length of wagon (m);
NwRS is number of wagons;
LRS is length of train (m);
VRS is initial speed (m/s);
VmaxRS is max speed (m/s);
TdRS is scheduled departure time (min);
TaRS is scheduled arrival time (min);
aRS is acceleration (m/s^2);
bRS is deceleration (m/s^2).

Equations for each RS:

$$t_a = (V_{\max} - V) / a \tag{1}$$

is acceleration time to maximal speed, (s);

$$S_a = (V_{\max} - V) \cdot t_a / 2 \tag{2}$$

is acceleration distance to maximal speed, (m);

$$S_{const} = |(S_{ST,LC} - Sc_R^d)| - S_a \tag{3}$$

is distance of movement with constant speed to the closure control, (m);

$$t_{const} = S_{const} / V_{\max} \tag{4}$$

is time of movement with constant speed to the closure control, (s);

$$t_{total} = t_a + t_{const} \tag{5}$$

is total time from starting movement to the closure control, (s);

$IF \quad S_{ST,LC} >= Sc_R^d \; THEN$

$$t_{clos} = Td \cdot 60 + t_{total} + tc_R^d \tag{6}$$

$ELSE$:

$$t_{clos} = Td \cdot 60 - t_{total} + tc_R^d \tag{7}$$

is level-crossing closing time, (s);

$$S_{clos} = S_a + S_{const} + tc_R^d \cdot V_{\max} \tag{8}$$

is distance from movement starting to closing point, (m);

$$S_{clos,LC} = S_{ST,LC} - S_{clos} \tag{9}$$

is distance from closing point to level-crossing, (m);

$$t_{clos,LC} = S_{clos,LC} / V_{\max} \tag{10}$$

is time of movement from closing point to level-crossing, (s);

$$t_b = V_{\max} / b \tag{11}$$

is braking time from maximal speed, (s);

$$S_b = t_b \cdot V_{\max} / 2 \tag{12}$$

is braking distance from maximal speed, (m);

$IF \quad S_b > Sst_R^d \; AND \; t^{RS} = 'with\ stop' \quad THEN$

$$t_{const}^{last} = (Td - Ta) \cdot 60 - t_a - t_b \tag{13}$$

is last wagon movement time with constant maximal speed, (s);

$$S_{const}^{last} = V_{\max} \cdot t_{const}^{last} \tag{14}$$

is last wagon movement distance with constant maximal speed, (m);

$$So^{last} = S_{ST1,ST2} + L - Sst_R^d - S_a - S_{const}^{last} \tag{15}$$

is distance from last wagon braking point to opening control point, (m);

$$t_{br,op}^{last} = \frac{-V_{\max} + \sqrt{V^2{}_{\max} + 2 \cdot (-b) \cdot So^{last}}}{(-b)} \tag{16}$$

is time from last wagon braking point to opening control point, (s);

$$t_{open} = t_{clos} + t_a + t_{const}^{last} + t_{br,op}^{last} \tag{17}$$

is level-crossing opening time, (s);

ELSE :

$$S_{clos,open} = (S_{ST,LC} - S_{clos}) + So_R^d + L \tag{18}$$

is distance from real closing point to opening point, (m);

$$t_{open} = t_{clos} + S_{clos,open} / V_{\max} \tag{19}$$

is level-crossing opening time, (s);

IF $to_R^d > 0$ *THEN*

$$t_{clos} = 0 \tag{20}$$

is level-crossing closing time, (s);

$$S_{clos} = S_{ST,LC} \tag{21}$$

is distance from movement starting to closing point, (m);

$$S_{clos,open} = S_{clos,LC} + So_R^d + L \tag{22}$$

is distance from closing point to opening point, (m);

IF $S_a <= S_{clos,open}$ *THEN*

$$t_{open} = t_{clos} + t_a + (S_{clos,open} - S_a) / V_{\max} \tag{23}$$

ELSE:

$$t_{open} = t_{clos} + \sqrt{2S_{clos,open} / a} \tag{24}$$

is level-crossing opening time, (s).

4. Algorithm for task solution

Genetic algorithm for task solution is proposed.

General steps of genetic algorithm are following:

Step 1. Randomly generated schedule. According to genetic algorithm operation sequence, the first step is random initialization of possible schedule population.

$$SCH = \{sch_1, sch_2, ..., sch_p\} \qquad sch_i = \{x_1, x_2, ..., x_k\}$$

Step 2. Evaluate schedule according fitness function. Each randomly generated schedule is evaluated by fitness function

$$V^S = \{F(sch_1), F(sch_2), ..., F(sch_p)\}$$

Step 3. Arrange schedules according evaluate. At this step randomly generated schedules are rearrange appropriate to this evaluating value

$$\overline{SCH} = \{\bar{s}ch_1, \bar{s}ch_2, ..., \bar{s}ch_p\} \qquad F(\bar{s}ch_1) = \max(V^S)$$

Step 4. Select the best schedules for elite set. From rearranged schedules predefined amount of best schedules are selected for elite set

$$Sch_E \subset \bar{S}ch$$

Step 5. Select schedules for crossover. Predefined amount of schedules are selected for crossover from elite set

$$Sc = \bar{S}ch$$

Step 6. Use crossover for new schedules generation. Appropriate predefined crossover conditions create the new generation of schedules

$$i, j = \overline{1, p} \qquad \bar{s}ch_i \; \Pi \; \bar{s}ch_j \rightarrow sch'_i = sch_{ij}; \quad sch'_j = sch_{ji}$$

Step 7. Mutation, random changes in schedules

$$x_j^{s'_i} = x_j^{s'_i} + 1; \quad sch'_i \in SCH'; \quad j = rand(\overline{1, k}); \quad i = rand(\overline{1, p})$$

Step 8. Evaluate new schedules population according fitness function

$$V^{S'} = \{F(sch'_1), F(sch'_2), ..., F(sch'_p)\}$$

Step 9. Combine result of new population and elite set

$$SCH = SCH_E \cap \bar{S}CH'$$

Step 10. Arrange new populated schedules according evaluation

$$\overline{SCH}' = \{\bar{s}ch'_1, \bar{s}ch'_2, ..., \bar{s}ch'_p\}, \qquad F(\bar{s}ch'_1) = \max(V^{S'})$$

Step 11. Deleting the worst schedules

$$SCH = SCH / \{sch_{p+1}, sch_{p+2}, ...\}$$

Step 12. Stop criterion. The whole process will stop when the generation equals to the-defined time, numbers of population and other criterions.

Fitness function for Genetic Algorithm:

Fitness function is expressed as following

$$F(X) = f(\Delta t, T_\Sigma) \rightarrow \min$$

where: $X = \{x_1, x_2, \ldots, x_n\}$ is deviation of the original schedule SCH (s); $\Delta t = \sum_{i=1}^{n} x_i^2 \Big/ n \rightarrow \min$ is average deviation from original schedule SCH (s); $T_\Sigma = \sum_{j=1}^{u} \Delta t_{closed} \rightarrow \min$ is total summary time, when level-crossing is in close position (s); u is number of intervals between level-crossing closing and opening.

Restrictions related with railway safety

Check of the new schedule to avoid time crossings on the same railway track.

Compare each train $i = \overline{1, n}$ with other $j = \overline{1, n}$ trains in the same direction: $d_i = d_j$, $i \neq j$.

Step 1.

$$Td_{check} = Tdi + xi; \qquad Ta_{check} = Tai + xi$$

Step 2.

IF ($Tdj + xj < Td_{check}$ AND $Taj + xj > Ta_{check}$) OR ($Tdj + xj > Td_{check}$ AND $Taj + xj < Ta_{check}$) OR
($Tdj + xj > Td_{check}$ AND $Taj + xj > Ta_{check}$ AND $Tdj + xj < Ta_{check}$) OR
($Tdj + xj < Td_{check}$ AND $Taj + xj < Ta_{check}$ AND $Td_{check} < Taj + xj$) THEN
Schedule failed
ELSE
Schedule successful

5. Computer Experiment in Developing of algorithm

Existing system structure investigation

The level-crossing of Sarkandaugava in Riga is selected for the computer experiment with the 24-hour train schedule.

The structure of level-crossing and elements location on the route are shown in Fig. 5.

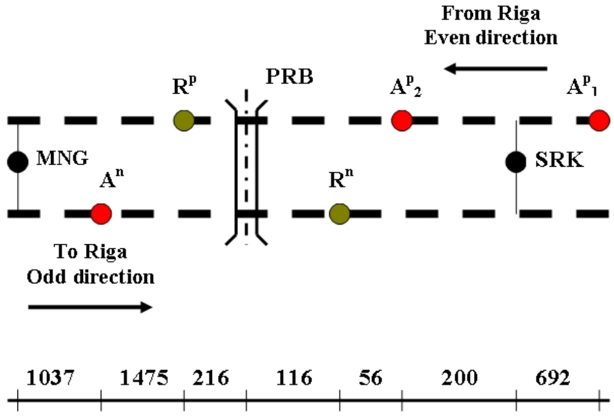

Fig. 5. Structure of the level-crossing. Where: $ST = \{MNG, SRK\}$ are two stations Mangali (MNG) and Sarkandaugava (SRK); $LC = \{PRB\}$ is one level-crossing PRB; $RS = \{RS_1, RS_2, \ldots, RS_{26}\}$ are 26 trains between two stations in time period between 3pm and 7pm; $R^{odd} = \{R^{odd}{}_1\}$ is one route in odd direction; $R^{even} = \{R^{even}{}_1, R^{even}{}_2\}$ are two routes in even direction; $S_{MNG,SRK} = 3100$ m; $S_{MNG,\ LC} = 2728$ m; $S_{SRK,\ LC} = 372$ m

Train types $t^{RS} \in$ {Cargo, Pas}. “Cargo” is rail transport units without stop at *SRK* station. “Pas” is rail transport units with stop at *SRK* station.

Routes’ characterized parameters are shown it Table1.

Table 1

Routes’ parameters

	Sc	*So*	*Sst*	*tc*	*to*
R^1_{odd}	1691	116	256	40	0
R^1_{even}	1064	216	2512	0	0
R^2_{even}	172	216	2512	0	13,5

Since in existing structure of level-crossing in Sarkandaugava the approaching points *An*, *Ap*1 and *Ap*2 are located too far or too close from the level-crossing, some time constant are necessary for secure level-crossing operate in close mode. Table 1 with *tc* marked delaying time constant for closed in odd direction, and with t0 marked the untimely time constant for closing in even direction.

All necessary numerical values of all 90 rolling stocks on Sarkandaugava-Mangali stage for calculations are combined in the Table 2.

Parameters of genetic algorithm

Following parameters are selected to implement genetic algorithm:

- Random selection
- Uniform crossover
- Crossover rate = 0.8
- Mutation rate = 0.02
- Bits for one variable = 5
- Population size = 100
- Number of generations = 1000
- Limits for x: -5 < X < 5

In order to avoid too big changes in schedule is set to permissible summary average deviation from original schedule is not over by 10 minutes.

Table 2

List of rolling stocks

	id	*t*	*d*	*R*	*Lw*	*Nw*	*L*	*V*	*Vmax*	*a*	*b*	*STd*	*Td*	*Sta*	*Ta*
RS_1	3648	cargo	even	Reven1	20	45	900	8.61	8.61	-	-	SRK	902	MNG	908
RS_2	3650	cargo	even	Reven1	20	40	800	7.38	7.38	-	-	SRK	938	MNG	945
RS_3	3654	cargo	even	Reven1	20	40	800	7.38	7.38	-	-	SRK	1015	MNG	1022
RS_4	3649	cargo	odd	Rodd1	20	57	1140	0	5.17	0.1	-	MNG	904	SRK	914
RS_5	3651	cargo	odd	Rodd1	20	35	700	0	12.92	0.1	-	MNG	929	SRK	933
RS_6	3655	cargo	odd	Rodd1	20	35	700	0	12.92	0.1	-	MNG	999	SRK	1003
RS_7	3657	cargo	odd	Rodd1	20	57	1140	0	5.17	0.1	-	MNG	1030	SRK	1040
RS_8	3663	cargo	odd	Rodd1	20	57	1140	0	5.17	0.1	-	MNG	1137	SRK	1147
RS_9	6136	pas	even	Reven2	-	-	120	0	20.7	0.6	0.8	SRK	928	MNG	931
RS_{10}	6138	pas	even	Reven2	-	-	120	0	20.7	0.6	0.8	SRK	966	MNG	969
RS_{11}	6140	pas	even	Reven2	-	-	120	0	20.7	0.6	0.8	SRK	980	MNG	983
RS_{12}	6142	pas	even	Reven2	-	-	120	0	20.7	0.6	0.8	SRK	1003	MNG	1007
RS_{13}	6144	pas	even	Reven2	-	-	120	0	20.7	0.6	0.8	SRK	1045	MNG	1048
RS_{14}	6146	pas	even	Reven2	-	-	120	0	20.7	0.6	0.8	SRK	1066	MNG	1069
RS_{15}	6148	pas	even	Reven2	-	-	120	0	20.7	0.6	0.8	SRK	1085	MNG	1088
RS_{16}	6150	pas	even	Reven2	-	-	120	0	20.7	0.6	0.8	SRK	1116	MNG	1119
RS_{17}	6133	pas	odd	Rodd1	-	-	120	0	20.7	0.6	0.8	MNG	919	SRK	922
RS_{18}	6135	pas	odd	Rodd1	-	-	120	0	20.7	0.6	0.8	MNG	947	SRK	950
RS_{19}	6137	pas	odd	Rodd1	-	-	120	0	20.7	0.6	0.8	MNG	981	SRK	984
RS_{20}	6139	pas	odd	Rodd1	-	-	120	0	20.7	0.6	0.8	MNG	1008	SRK	1011
RS_{21}	6141	pas	odd	Rodd1	-	-	120	0	20.7	0.6	0.8	MNG	1024	SRK	1027
RS_{22}	6143	pas	odd	Rodd1	-	-	120	0	20.7	0.6	0.8	MNG	1045	SRK	1048
RS_{23}	6145	pas	odd	Rodd1	-	-	120	0	20.7	0.6	0.8	MNG	1065	SRK	1068
RS_{24}	6147	pas	odd	Rodd1	-	-	120	0	20.7	0.6	0.8	MNG	1090	SRK	1093
RS_{25}	6149	pas	odd	Rodd1	-	-	120	0	20.7	0.6	0.8	MNG	1120	SRK	1123
...															
RS_{89}	3668	cargo	even	Reven1	20	40	800	7.38	7.38	-	-	SRK	1388	MNG	1394
RS_{90}	3671	cargo	odd	Rodd1	20	57	1140	0	5.17	0.1	-	MNG	1397	SRK	1407

Result of genetic algorithm implement

As shown in Fig. 6 all selected genetic algorithm parameters are entered in appropriate machine code. According to the genetic algorithm steps the computer experiment results are summarized in Table 3.

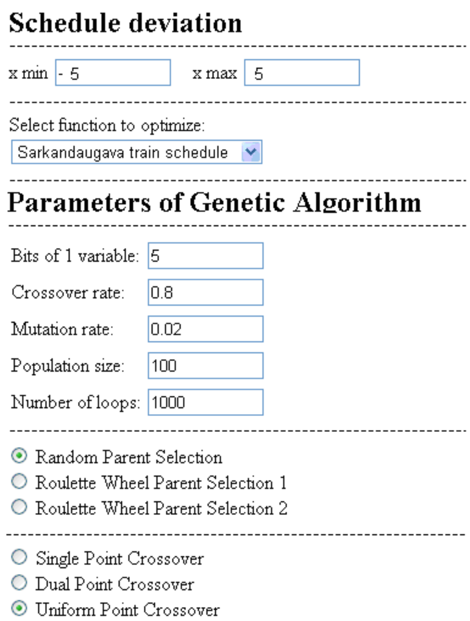

Fig. 6. Entering genetic algorithm parameters

Table 3

Computer experiment results (calculated in 33422 seconds)

Train	Direction	Route	Original depart time	Original arrival time	Deviation	Result depart time	Result arrival time	Level-crossing closed
3605	odd	R1	1:09	1:19	-3	1:06	1:16	1:10:52 - 1:19:16
3607	odd	R1	1:31	1:35	3	1:34	1:38	1:38:09 - 1:39:39
3611	odd	R1	2:38	2:48	3	2:41	2:51	2:45:52 - 2:54:16
3615	odd	R1	3:08	3:12	2	3:10	3:14	3:14:09 - 3:15:39
3619	odd	R1	4:12	4:16	2	4:14	4:18	4:18:09 - 4:19:39
3621	odd	R1	4:37	4:47	-2	4:35	4:45	4:39:52 - 4:48:16
6101	odd	R1	6:00	6:03	-2	5:58	6:01	6:00:05 - 6:02:34
3627	odd	R1	6:32	6:42	-4	6:28	6:38	6:32:52 - 6:41:16
3629	odd	R1	6:50	6:54	4	6:54	6:58	6:58:09 - 6:59:39
6103	odd	R1	6:58	7:01	4	7:02	7:05	7:04:05 - 7:06:34
6105	odd	R1	7:30	7:33	-2	7:28	7:31	7:30:05 - 7:32:34
6107	odd	R1	7:45	7:48	1	7:46	7:49	7:48:05 - 7:50:34
6109	odd	R1	8:30	8:33	3	8:33	8:36	8:35:05 - 8:37:34
6111	odd	R1	9:29	9:32	-1	9:28	9:31	9:30:05 - 9:32:34
6150	even	R2	18:36	18:39	3	18:39	18:42	18:38:47 - 18:39:38
3658	even	R1	19:20	19:26	3	19:23	19:29	19:21:40 - 19:25:41
6152	even	R2	19:42	19:45	-1	19:41	19:44	19:40:47 - 19:41:38
6154	even	R2	20:21	20:24	2	20:23	20:26	20:22:47 - 20:23:38
6156	even	R2	20:58	21:01	-5	20:53	20:56	20:52:47 - 20:53:38
3662	even	R1	22:00	22:07	-5	21:55	22:02	21:53:26 - 21:58:22
6158	even	R2	22:37	22:40	1	22:38	22:41	22:37:47 - 22:38:38
3668	even	R1	23:08	23:14	3	23:11	23:17	23:09:40 - 23:13:41
6160	even	R2	23:36	23:39	-2	23:34	23:37	23:33:47 - 23:34:38

Original total time of closed level-crossing
T_{orig} = 14892.070033683 s

Total time of closed level-crossing
T_{sum} = 10447.344239586 s

6. Conclusions

Analyzing the data from algorithm conclusions are:

- compared with the original schedule changed schedule total level-crossing time in the closed position has been decreased from 248 min (14892 sec) to 174 min (10447 sec);
- changed schedule average deviation is not over 10 minutes;
- genetic algorithm can be used to solve public and cargo transport flow organization tasks.

Schedule changes in ratio + / - 5 min give possibility to significantly reduce the level-crossing being in closed position, thereby significantly increasing the capacity of the level-crossing.

To reduce level-crossing being in closed position give several possibilities:

- increase city transport flow through level-crossing;
- increase rail passengers and cargo transport flow on the route Mangali – Sarkandaugava without negative effect on current city transport flow.

To define saving up energy and take in notice all restrictions and meets requirements is necessary to implement additional researches.

References

1. **Marek Horyński.** Intelligent Electric Systems in Urban Traffic Control. Department of Computer and Electrical Engineering, Lublin University of Technology, TEKA Kom. Mot. Energ. Roln. OL PAN, Poland, 2007.
2. **Маркгвард К.М.** Электроснабжение электрифицированных железных дорог. Москва, Транспорт, 1982.
3. **Танаев В.С.** Теория расписаний. – М.: Знание, 1988.
4. **Матвеев С.И., Коугия В.А.** Высокоточные цифровые модели пути и спутниковая навигация железнодорожного транспорта. 2005. – 290 с. ISBN 5-89035-227-X.
5. **Рутковская Д., Пилинский М., Рутковский Л.** Нейронные сети. Генетические алгоритмы и нечеткие системы. – Москва: Горячая линия – Телеком, 2006.
6. **Marco Wiering, Jelle van Veenen, Jilles Vreeken, Arne Koopman.** Intelligent Traffic Light Control. Institute of Information and Computing Sciences, Utrecht University. Padualaan 14, The Netherlands, 2004.

7. **Kuei-Hsiang Chao, Ren-Hao Lee, Meng-Hui Wang.** An Intelligent Traffic Light Control Based on Extension Neural Network // Department of Electrical Engineering, National Chin-Yi University of Technology.
8. **Ribickis L., Gorobetz M., Levchenkov A.** Intelligent Electric Vehicle Motion And Crossroad Control. // In Proceedings of 12th International Power Electronics and Motion Control Conference. Portoroz, Slovenia, 2006. 1239 - 1244 p.
9. **Rankis I., Gorobetz M., Levchenkov A.** Optimal Electric Vehicle Speed Control By Intelligent Devices. Rīgas Tehniskās universitātes raksti. Enerģētika un Elektrotehnika. Sērija 4, sējums 16. 2006. 127-137 lpp.
10. **Gorobetz M., Levchenkov A.** Evolutionary Algorithms and Dynamic Parameters for Public Electric Transport Modelling //In Proceedings of 16th Mediterranean Conference on Control and Automation, Ajaccio, France, 25-27 June 2008. 215-219 p.
11. Artificial Neural Nets and Genetic Algorithms: Proceedings of the International Conference in Portoroz, Slovenia, 1999.
12. **Alps I., Potapov A., Gorobetz M., Levchenkov A.** Algorithm for Public Electric Transport Schedule Control for Intelligent Embedded Devices // In proceedings of RTU 51st International Scientific Conference "Power and Electrical Engineering" Latvia, Riga 2010, 155 – 160 p.