GENETIC ALGORITHM FOR TRANSITIONS SCHEDULING GUIDANCE OF TRAINS

Maria-Magdalena SANTA Octavian CUIBUS Tiberiu LETIA
Technical University of Cluj-Napoca, 0264 401432, Maria.SANTA@aut.utcluj.ro;
Technical University of Cluj-Napoca, 0264 401432, Octavian.CUIBUS@aut.utcluj.ro;
Technical University of Cluj-Napoca, 0264 401432, Tiberiu.LETIA@aut.utcluj.ro.

Abstract: The railway traffic system is characterized as large and dynamic with uncertain properties related to resource loading, train arrivals and failures. The purpose of this paper is to control system online in order to guarantee that all the trains behave according to their timelines. The solutions are found with distributed expert systems. The control signals are implemented and verified using temporal Rule Nets. Due to the large number of variants involved, extensive check of all possible durations and waiting times on a resource is impossible, therefore an evolutionary approach (genetic algorithm) must be used. The results obtained through simulations show that the proposed distributed controllers solve the control problems and can be used for large scale implementation.

Keywords: *distributed expert systems, genetic algorithm, temporal rule nets and railway applications.*

I. INTRODUCTION

The railway traffic control system is a dynamic one that operates in an environment with uncertain, properties that include transient and resource overloads, arbitrary arrivals, arbitrary failures and decreases of traffic parameters. The railway resources are lines, interlocking (combination of switches and traffic lights), and platforms (a special type of lines). The allocation can be performed off-line before a train starts or on-line, during the system evolution when a train reaches some points. Due to environment circumstances (weather conditions, unexpected maintenances required, faults etc.) some trains are delayed.

The railway network is partitioned in subsystems. Each subsystem has assigned a local controller. It is implemented on the computer of a railway station.

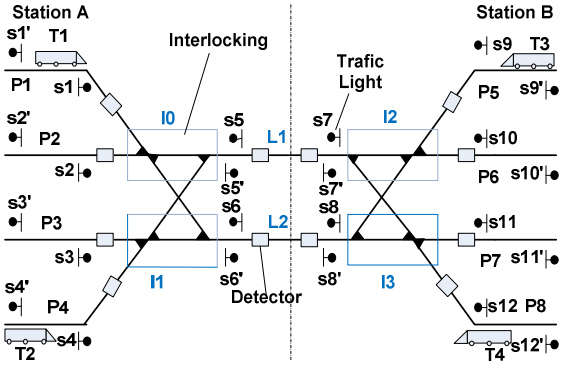

Figure 1. Example of railway structure.

A part of a railway network is present in Figure 1, with the following specified elements: traffic lights, denoted by s1,…, s8; switch points; detectors, represented by squares; interlocking, denoted by I0, …, I3; platforms, denoted by P1, …, P8; lines, denoted by L1, L2; trains, denoted by T1, …, T4.

An interlocking cannot be occupied by a train more than it is necessary to cross it. Each railway route element has associated a minimum-maximum interval. A detector signals the presence of trains in the corresponding zone. The state of a resource can be reserved, occupied or released.

The local controller assigned to each station has the task to implement the train schedules. The controller signals the traffic lights and the interlocking. It handles signals received from detectors to advance its execution and to release the reserved resources. Once entered on a line, the train moves freely towards the next interlocking. The movement on a line takes a specified duration. If the next line is occupied, the control system delays the train entrance using the corresponding traffic light.

II. RELATED WORKS

Formal development and verification of a distributed railway control system are performed applying a series of refinement and verification steps [1]. The distributed train scheduling problem has some similarities with distributed software job scheduling [2, 3]. Both have to fulfill real-time constraints relative to finishing time, communication requests and resource management.

The concept of collaborative scheduling is also applicable. Each node constructs its local schedule using only local information. The lack of global information makes it impossible for a node to make a globally optimal decision. Thus it is possible for a node to make a scheduling decision that is locally optimal in terms of the utility that can be accrued to the node, but compromises global optimality, [4].

Distributed systems offer several advantages when implementing control systems and the advantages can be

Manuscript received Mai 26, 2011; revised October 21, 2011

increased with the application of artificial intelligence techniques. A new layered architecture for the implementation of intelligent distributed control systems is proposed in [5]. This architecture distinguishes four levels of a distributed control system.

Rule Nets (RN) are defines in [6] as a symbiosis between Expert Systems (based on rules) and Petri Nets (PN) in such a way that the facts resemble places and the rules are close to transitions. A fact may be true or false.

A control system that is implemented by means of expert systems based on rule nets are conceives in [7]. They use a formalism that seeks to express an automatism in a similar way to as would make it a human being: "IF antecedents THEN consequents". But at the same time rule nets are a tool for the design, analysis and implementation of rule based systems.

An application of evolutionary algorithms to solve very complex real-world problems in railway traffic is presented in [8]. For this purpose a Genetic Algorithm is designed to solve the Train Timetabling Problem. The railway scheduling problem considered in this work implies the optimization of trains on a railway line that is occupied by other trains with fixed timetables. The timetable for the new trains is obtained with a Genetic Algorithm.

III. CONTROL RAILWAY SYSTEM WITH DISTRIBUTED EXPERT SYSTEM

The main control problems of the train traffic are analyzed and solved using distributed expert system (DES) and modeled by Temporal Rule Nets (TRN).

A DES can be implemented through interconnected nodes (which are expert systems) physically dispersed, but collaborating for the development of a common task.

Each train has assigned a corresponding train agent that has global information about the route and times for every station. The agent asks the local expert system a path from current platform (line) of station to the next platform (line) of the next station.

The control system for railway system is composed by a DES composed of a set of local expert systems (ES), a number of train agents (Ag), local controller (C), resource table (RT), and resource scheduler (RS). An ES is composed by a RS and a rule base (RB). The architecture of a DES is given in Figure 2.

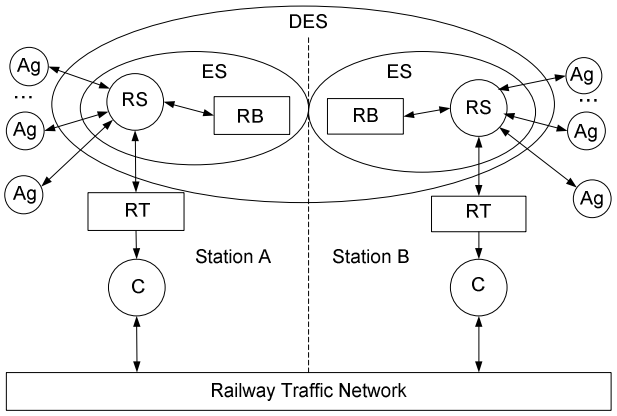

Figure 2. System Architecture.

An ES attempts to provide an answer to a problem or to clarify uncertainties where normally one or more human experts would need to be consulted. The ES of the TRN has rules of the form: IF (conditions) THEN (actions). For every transition there is a rule.

The rule set is distributed onto the local traffic control subsystems. The ES uses: global information, local information (state and reservations), local rule set and an inference machine to find a path that is not in conflict with any other current active path, cooperation with neighbor ES to obtain the neighbor resource allocations and a path evaluator to assign the set of proposed solutions.

The term *task* is introduced to define the control problem. It consists of the moves of a train from a given platform (line) of a railway station, to a specified platform (line) of another railway station.

A train control problem consists of a set of tasks which have to be controlled, acting on traffic lights and switches, such that all the trains fulfill the departure and arrival times. A train specification contains to the shortest duration needed by a train to cross a resource. A train T_n needs t(R) minute(s) to cross a resource R, where t is time. For the given example, the train specifications are presented in a tabular form in Table 1.

Table 1. Train specification in station A

Resource	P1	P2	P3	P4	I0	I1	L1	L2
Time	4	3	3	5	1	1	6	5

A train schedule solution specifies the necessary resources, the occupancy order and the durations required for the train to spend on each resource.

The online scheduling problem considers the case when a train set is already scheduled. They are in movement when one or more trains demand to be scheduled. The previously scheduled trains move according to the allocated resources and the algorithm searches to allocate the available resources to the new unscheduled trains without disturbing the rout of previously scheduled trains. A relevant requirement for any solution is to avoid the train deadlocks. Any solution that leads to a deadlock has to be rejected.

IV. TEMPORAL RULE NETS FOR TRAIN SCHEDULING

In general RN is a tool for the design, analysis and implementation. RN offers the user a simple and intuitive interface for the specification of the desired behavior of the ES. RN consists of a mathematic-logical structure which analytically reflects the set of rules that allow a human expert to take its own decisions. Like Petri Nets, RN admits a graphical representation as well as a matrix form.

The TRN is well suited to describe the dynamic behavior of complex concurrent systems based on graph theoretical concepts and it is possible to analyze properties of such systems. Input places represent the

availability of resources, the transition their utilization and output places the release of the resources. Timed transitions are used to represent resource activities.

Every transition (t_0, …, t_n) has associated a rule (R_1,…, R_n). Rules can be used to verify, for a train on a location in station A, if the next resources are free and so, creating a path to the next station B.

For each train there is a time associated with the period of resource occupation. When a resource is released the time is incremented with corresponding duration from the rule. Table 2 represents the transitions, the rule assigned to each transition and the timing, corresponding to each resource of the station A.

Table 2. Transition, Rules and Timing for station A.

Transition	Rule	Timing
t0	R1	1
t1	R2	1
t2	R3	1
t3	R4	1
t4	R5	4
t5	R6	3
t6	R7	3
t7	R8	5
t8	R9	1
t9	R10	1
t10	R11	1
t11	R12	1
t12	R13	1
t13	R14	1

A path for a train is constructed using the transitions with associated rules. All the possible paths for a train are calculated in parallel until there are n paths.

In our case, when the train T is on platform P1 and wait for a path toward a platform in station B, the applicable rules are:

R1: IF (T on P1) THEN (T on P1) AND (t (T) = t (t) + 1)

R5: IF (T on P1) AND !(I0 reserved) THEN

(I0 reserved for T) AND (t (T) = t(T)+ t (P1)).

The ES computes two paths for the train, (a path for each rule) so, there is a tree representation for the possible paths. For rule R1 is executed the transition t0, that means the train T wait on platform P1 and the time is incremented, t(T) = t(T) + 1, from Table 2.

For rule R5 is executed transition t4, that means: IF the train T is on the platform P1 and interlocking I0 is free THEN the train T is moving on interlocking I0, the time is incremented with time necessary for the train to cross resource P1, t (T) = t (T) + t(P1), t (P1) = 4, from Table 2.

The next execution can continue the existent path or create new paths. The train path is complete when the train has arrived at destination. When there are n = 20 paths, the finite set of paths calculated is Π_n, $\Pi(i)$ is a vector that contains all needed resource for a train. Total time for a path is the sum of the minimum time needed for a train to cross all the resources.

A train T_j movement can be specified by the sequence of transitions t_j^k that have to be fired at the moment τ^k: $T_j = t_j^1(\tau^1)\cdot t_j^2(\tau^2)\cdot \cdots \cdot t_j^k(\tau^k)\cdot \cdots \cdot t_j^n(\tau^n)$, for the aim to meet the demanded state. The argument τ^k specifies the absolute time when the transition is executed. This sequence of transitions corresponds to the relation between resources states and train schedule.

Let $S = \{T_1, T_2, \ldots, T_n\}$ be the set of already scheduled trains. At the moment τ^k, for all the trains, the sequence of transitions is $t_w^k(\tau^k)$. The set S involves a sequence of transitions:

$$\Sigma = t_u^1(\tau^1)\cdot t_v^2(\tau^2)\cdot \cdots \cdot t_w^k(\tau^k)\cdots t_z^n(\tau^n)$$

that have to be fired such that all the scheduled trains reach the destination at the scheduled times.

The set of transitions from Σ that have to be fired at the moments τ^k is denoted by $\sigma(\tau^k)$.

The sequence of transitions that allows the trains from the set S and the new train T_x to fulfill the requirements is denoted by Σ'. This new sequence of transitions has to be executed to make possible the correct movements of the trains of the set S plus the train T_x. The new set is: $S' = \{T_1, T_2, \ldots, T_n, T_x\}$.

A transition could be enabled, but the scheduler could not elect it for execution. The scheduling of a transition involves the reservation of a corresponding resource.

The finite set $\Pi(n)$ of paths for a train moving from departure to destination specifies only the necessary resources but not the times when they are allocated or released. For scheduling of a train T_x at the moment i the input date are the set of paths $\Pi(n)$ for the new train, the pending transitions at the time i from the current state, the longest period δ to schedule a train on a resource and the largest attempt number is γ.

If the resources cannot be reserved even if the train is delayed on some resources, the path is rejected. If more than one path is scheduled correctly, the best conform time criteria, is chosen. If no valid schedule can be found, the train is not schedulable during the specified time interval.

V. GENETIC ALGORITHM FOR TRANSITIONS SCHEDULING GUIDANCE

Once the path is known, the next step is to schedule the train, i.e. reserve the resources for specific amounts of time. Within the previous algorithm, for each path one has to determine the durations for all the transitions. Due to the large number of variants involved, extensive check of all possible durations and waiting times is impossible, therefore some heuristic approach must be used. Because of simplicity, the genetic algorithm shall be used to choose the best durations for each transition for train T_x. In the following description, the genetic algorithm is used for station A. The algorithm should produce a set of best possible paths for T_x, ordered according to the travel duration of train T_x. It is considered that train T_x is, for example, of the first type (see Table 1), thus the normal travelling times for each resource correspond to the first

line in Table 1. The minimum timing of each transition is listed again in Table 3, first column. On top of that, further delay can be added, representing waiting times on each resource.

The genes for the genetic algorithm contain the total durations spent by the train T_x on each resource, therefore the genotype is set accordingly, using the minimum and maximum execution times of each transition. These correspond to the execution times of the transitions, which have to be between the minimum and the maximum values specified for the train category of T_x, as shown in Table 3. The minimum value T_{xmin} corresponds to the minimum time the train needs to travel throughout the given resource, whereas T_{xmax} is the maximum time which also includes the largest waiting time that the train is permitted to wait on the resource. The last row represents the maximum allowed waiting times on each resource.

Table 3. Total possible durations spent on each resource

	P1	P2	P3	P4	I0	I1	L1	L2
T_{xmin}	4	3	3	5	1	1	6	5
T_{xmax}	12	11	11	13	1	1	9	8
Max.waiting	8	8	8	8	0	0	3	3

The exact form of the chromosome and the genotype is shown below.

Table 4. The chromosome and the genotype.

g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
4÷12	3÷11	3÷11	5÷13	1÷1	1÷1	6÷9	5÷8

The fitness function calculates the total travel time for each chromosome, by summing up the durations for transitions that are fired when the train T_x travels along the pre-established path. Collision with other travelling trains is drastically penalized. In order to check for collisions, the genetic algorithm has to simulate the travel of all trains on the Petri Net, as described in detail in paper [9].

For a specific path, not all genes are relevant for calculating the total travel duration. For the path for train T_x, that is: P1-I0-L1, only genes g_1, g_5 and g_7 are relevant. The chromosome can thus be resized, according to the path used. Considering that the genetic algorithm has to be ran for each path α from the set $\Pi(n)$, it is easier to use a general purpose chromosome, constructed as shown above, for station A.

Table 5. The best 5 solutions for train T_x

Nr.	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
1	11	3	3	5	1	1	6	5
2	11	3	3	5	1	1	7	5
3	12	3	3	5	1	1	6	5
4	12	3	3	5	1	1	7	5
5	13	3	3	5	1	1	6	5

Using the algorithm for station B implies building another chromosome with a different genotype. For example, for the path P1-I0-L1, using time criteria, the best 5 solutions calculated by the genetic algorithm are shown in Table 5.

V. CONCLUSION

The online scheduling (resource allocation) methods obtain better performances for timelines fulfilling compared to offline scheduling methods. The simulations show that the trains do not collide and do not enter in deadlocks. The execution durations are significantly dependent on the rule list. The rule list is dependent on the number of constrains. The execution durations are also dependent on the number of the new trains that have to be scheduled and on the interlocking number. The number of trains that are already scheduled can slightly influence the execution durations

However, if the train does not follow the schedule, the path has to be rescheduled in order to guarantee deadlock avoidance. In that situation, the priority is deadlock avoiding rather than deadline fulfilling.

REFERENCES

[1] A. Hauxthausen, and J. Peleska, "Formal development and verification of a distributed railway control system", *IEEE Trans. on Software Engineering,* vol. 26, 2000.

[2] S.F. Fahmy, B. Ravindran and E. D. Jensen, "On collaborative scheduling of distributable real-time threads in dynamic, Networked Embedded Systems", *Proceedings of the 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, pp. 485-491, 2008.

[3] S.F.Fahmy,B.Ravindran and E.D.Jensen,"Scheduling distributable real-time threads in the presence of crash failures and message losses", *ACM SAC, Track on Real-Time Systems*, 2008.

[4] T. Leția, M. Hulea, R. Miron, "Distributed Scheduling for Real-Time Railway Traffic Control", *IMCSIT 2008, Real-Time Workshop RTS'08, Wisla, ISSN 1896-7094*, pp. 679-685, Polonia, 2008.

[5] J.V.Capella, A. Bonastre and R. Ors, " Advanced Architecture for Distributed Systems with a Network Infrastructure Based on CAN and Internet for Content Distribution", *AWCC 2004*, LNCS 3309, pp. 58-69, 2004.

[6] M. Silva, "The Petri Nets in Automation and in Compute science", *Editorial AC*, Madrid, 1985.

[7] J.V.Capella, A. Bonastre and R. Ors, "Analysis and validation in design time of distributed control systems implemented by means of rule based expert systems", *The Fourth International Conference on Control and Automation (ICCA'03),* Montreal, Canada, 2003.

[8] J.Tornquist and J.A. Persson, "Train traffic deviation handling using tabu search and simulated annealing", *Proceeding of the 38th Annual Hawaii International Conference on System Science*, pp. 73a, 2005.

[9] Santa, M.M., Cuibus, O., Leția, T., 2010. Train Scheduling with Delay Time Petri Nets and Genetic Algorithms. *Proceedings of the 14th International Conference on System Theory and Control (Joint conference of SINTES14, SACCS10, SIMSIS14),* 17-19 octombrie, Sinaia, pp. 479-484.