Support for Creating Robust Plans and Schedules

Peter Jankovič*
Department of Transportation Networks
Faculty of Management Science and Informatics
University of Žilina
Univerzitná 8215/1, 010 26 Žilina, Slovakia
peter.jankovic@fri.uniza.sk

Abstract

Creation of plans and schedules for systems with random phenomena is a research area that is gaining considerable attention. Standard techniques of scheduling and exact methods cannot be utilised for stochastic systems. The measure of robustness is an important indicator of the schedule, representing the resistance of the schedule to random phenomena that occur during the execution of the schedule. This thesis describes the procedure for evaluating the robustness of the schedules. A complete methodology for creating of robust schedules, based on the use of computer simulation, is presented. Three distinct procedures for increasing the robustness (by changing a plan, increasing of time reserves and by changing the order of allocation of resources) are explained in detail. A new method for changing the order of resource allocation to the activities has been developed. Proposed method provides the possibility to identify the order of allocation that will lead to a system deadlock. The procedures were tested using simulation model of an existing container terminal.

Categories and Subject Descriptors

I.6.4 [**SIMULATION AND MODELING**]: Model Validation and Analysis

Keywords

Simulation, robustness, schedules

1. Introduction

Planning and scheduling, with the development of information technology has become the subject of research by many scientists, but also the companies that are trying to ensure their costs. There are many approaches trying to solve the problem of finding the optimal plan and the subsequent localization of resources for its provision. Standard approaches mainly focus on the mathematical description of the problem and its solution using the various numerical approaches. However, thus obtained plan has large limitations because of stochastic character of many phenomena of the real world and its execution carries the risk of failure.

In addition to the various specific requirements for plans and schedules, the requirement on the robustness of the plan or schedule is important. Today, a large attention is given to the problem of determination of robustness of plan and its increasing. These schedules are needed especially for rail and air transport, but also in manufacturing processes with the aim to reduce downtime of machines in disorders.

1.1 The Planning

Planning is generally understood as the process of creating a plan. The plan is a sequence of actions. These actions must be applied so that the planning system is getting from the initial (default) state to the terminal state (destination). It is a decision about what actions to which conditions should be applied to achieve the desired objectives of the situation.

Plan for practical use is only an approximate picture of how the system should work. In the real conditions, there are the mostly required resources for the fulfilment of activities, without which it is impossible to carry out. Just allocation of resources plays a key role in the practical application of the plan. This issue is solved by scheduling.

The requirements on its schedule may be a part of the plan. In these requirements can be described the maximum duration of the plan, maximum duration for conclusion, or the opening times of the individual activities, and the order of allocation of resources for some activities.

1.2 The Scheduling

In the case that there is already a plan, such as a sequence of actions, it is appropriate to specify in what time will be different actions performed. Scheduling means the assigning of resources to activities in time.

Activity is the basic technological operation, which is not divided on partial technological operations. Activity in some scheduling issues is also called the operation.

*Recommended by thesis supervisor: Assoc. Prof. Valent Klima. Defended at Faculty of Management Science and Informatics, University of Žilina on August 28, 2012.

Jankovič, P. Support for Creating Robust Plans and Schedules. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 4, No. 3 (2012) 23-34

The resource is a device which is able to perform one or several operations. In General, it is available a set of types of resources, while the number of resources of that type is defined for each type. Then, for each activity is defined a set of desired types of resources and their count. For example, for the fulfilment of the task it is necessary 5 staff workers of type 1, 3 staff workers of type 2 and one crane.

2. Measurement and evaluation of robustness

An important part is the determination of the degree of robustness of a plan or a schedule. Robust plan (schedule) is such a plan (schedule), whose quality is not reduced during execution (keeping in advance of known uncertainties). This property is expressed as the probability of which the plan, or schedule will work without distortions (shift of activities, it is necessary the operational solution of the situation.). It is needed to define a measure of robustness as a property of the plan, which is used to compare two plans from the point of view of their resistance to the ever changing realistic conditions. Below we will use even short term "robustness" instead of measure of robustness.

Robust schedule is the schedule, which has the required measure of robustness. The number in the range will express the robustness of the schedule (0, 1>, and the smaller value will mean smaller robustness of the schedule. Extreme value of 0 is not used, because the schedule with zero robustness would have to be unenforceable. The probability of failure of the schedule increases with the decreasing robustness of the schedule. It does not mean, that the probability of failure of the schedule with the robustness of 0, 4 is 0.6. If we allow using zero as a minimum value of robustness, the probability of failure pattern would have to be equal to 1 and thus the schedule would be in 100 % unenforceable. Perfectly robust schedule might not exist, but it may be. Whether the schedule may have the robustness of equal to one depends on the character of uncertainties of real environment. In the case that all the parameters of the model are deterministic, the measure of robustness is always equal to one.

2.1 Methods to obtain the data necessary for the calculation of the quality and robustness of schedule

Evaluation of robustness is not possible to do only on the basis of a single practical execution of the schedule. Up on the basis of a large quantity of its realization in the random conditions it is possible to determine the robustness of the schedule. If possible it is useful to draw from existing actual execution of the schedule. In this way it is possible to measure just the robustness of schedules, which are already deployed and operate long enough. In the face of the process of creating a schedule it is not possible. Also, this is one of the reasons for which, the computer stimulation is used in the conditions with uncertainties. If we can create a sufficiently precise computer model to simulate the work schedule, we can acquire the necessary data using experimental simulation run.

Here it is possible to use the simulation technique of "Monte Carlo". Sometimes it is also referred to as static simulation. The idea is quite simple. On the basis of a large number of replications of the simulation run we can evaluate in detail the simulated system. Replication will be introduced by an execution of a schedule through the computer simulation.

2.2 Calculation of robustness in stochastic environment

Occurrence of random phenomena can cause, that the activity will not be able to start or to end in time. Let we know for each task the time of the latest necessary start and the end of working. In the case that one of these times is not hold on, the task failed. In practical conditions, it is not necessary to use both. The choice depends on the practices as applied in the system. The failure of the task or activity can be defined as a violation of its planned start of working in time or its planned completion about a value that causes the shift (disturbance) of execution of other tasks.

On the basis of data obtained by the procedure described in Chapter 2.1 we can determine for each activity a probability of its failure. It is not only important that a critical time has been exceeded, but especially how long it has been delayed. The weight of failure must therefore be an extension of the execution. First, it is necessary to define the variables that will be used to calculate the robustness.

ER_j - end time of replication j, time of completion of the last job,

SR_j - start time of replication j,

LS_i - the time of the latest necessary start of execution of task J_i,

ws_i - the importance of compliance with the LS_i for the task of J_i,

RS_{ij} - the time of the real start of the execution of the task J_i in replication j, if in a given replication the task has started to carry out, otherwise it is equal to ER_j,

LE_i - the time of the latest necessary end of the task J_i,

we_i - the importance of compliance with the LE_i for the task of J_i,

RE_{ij} - the time of the real end of the execution of the task J_i in replication j, if in a given replication the task has completed, otherwise it is equal to ER_j,

WS_{ij} - the delay of the latest necessary start of task J_i in the replication j,

WE_{ij} - the delay of the latest necessary end of task J_i in the replication j,

$\overline{WS_i}$ - the average delay of the latest necessary start of task J_i,

$\overline{WE_i}$ - the average delay of the latest necessary end of task J_i,

R - robustness.

Now we can calculate what would be the average delay of the task in the replication. As far as the time of the actual start (end) is outpacing the time of the latest necessary start (end) will be naturally the delay equal to zero.

$$WS_{ij} = max\{0, RS_{ij} - LS_i\} \tag{1}$$

$$WE_{ij} = max\{0, RE_{ij} - LE_i\} \quad (2)$$

On the basis of the average values of delay for a particular replication we will determine the average delay of the task.

$$\overline{WS_i} = \frac{1}{m}\sum_{j=1}^{m} WS_{ij}, \quad (3)$$

where m is the number of replications.

$$\overline{WE_i} = \frac{1}{m}\sum_{j=1}^{m} WE_{ij}, \quad (4)$$

where m is the number of replications.

Sometimes it may happen that it will not be possible to continue under the applicable schedule. There is, for example, the deadlock. It is necessary to go to the operational management, possibly some tasks will not be practicable. To some extent this is already treated in the face of delays of the task in replication status, when the time of its termination is set on the completion of the entire schedule. Sometimes, however, such a situation is extremely undesirable and robustness must take it into account. Let us say the status of completion of replication j in the form of DR_j, which is equal to one if the replication completed whole and equal to 0, if replication failed. Let TW expresses the proportional weight to delay activities and DW for the weight of the no execution of the plan. The value TW must be of the range $\langle 0, 1\rangle$, the value DW of the range $\langle 0, 1\rangle$. It must naturally apply $DW + TW = 1$.

For the determination of robustness it has been after a careful consideration defined by the following formula:

$$R = TW \times \left(1 - \frac{\frac{\sum_{i=1}^{n}\overline{WS_i} \times ws_i}{n} + \frac{\sum_{i=1}^{n}\overline{WE_i} \times we_i}{n}}{2 \times \frac{\sum_{j=1}^{m} ER_j - SR_j}{m}}\right) + DW \times \frac{\sum_{j=1}^{m} DR_j}{m}, \quad (5)$$

where n is the number of tasks, m is the number of replications and ws_i, we_i are the weight (importance) of the task J_i expressed by the real number of range $\langle 0, 1\rangle$.

Experiments have confirmed that the formulas for calculating the robustness allow creating its fair and useful expression. Using settings of importance for each individual tasks and the constants of TW, DW it is possible to create parameters for calculating of robustness of schedule of the task.

3. The methodology for the establishment of a robust schedule

It can be assumed that by increasing the measure of robustness of a schedule the resources are used in fewer amounts (increasing their number) and operating costs increase. But this is balanced by the robustness, which becomes the dominant feature of the schedule in certain problems. Therefore, our task is to find such a plan or a schedule, which will have the required degree of robustness, but other qualitative parameters will be the best. About which they are, it is necessary to decide for the case and there is not possible any generalization. Robustness of the initial schedule may be too low, or in contrast too high unnecessarily (using of resources is too low).

In existing approaches is most frequently used in the creation of the schedule in the stochastic environment the computer simulation, but which performs mostly the function of the tool to verify the solutions only. The results of the simulations it is possible to use but also to increase the robustness of schedule and, where appropriate, the other required quality criteria. The simulation run provides a large amount of information that can be effectively used. Another reason for using the simulation equipment is just the presence of random phenomena. If it is necessary to assess the robustness of the schedule yet before its commitment to service, the simulation is an ideal way for its calculation. In this chapter a created procedure will be presented that combines mathematical and heuristic procedures with computer simulation.

3.1 A reactive and predictive approach

In the reactive approach is planning progressive in small increments, the individual decisions are made on the basis of the current state. A reactive approach to planning generates, if it is properly used, flexible schedules and plans because all decisions are made without the anticipated consequences. In practice, it is actually about the operational management, where every decision is made on the basis of the current situation. This procedure is also used in practice for the management of the simpler systems, where the high occurrence of random phenomena makes it impossible to use the management according to the plan.

In an environment with the uncertainties, it is good to use techniques based on obtaining predictive solutions (here it is possible to use methods for planning in the static environment described below) and in the course of verification of correctness (most often using simulation techniques) to a finding that the actual state is different from the planned, it is possible to obtain new solution by a reactive way. A prediction approach is based on the fact that uncertainties are taken into account already in the planning of activities and deciding on the routing. This procedure is suitable for cases where the parameters of delays (failure) of individual activities are known.

3.2 Create a robust schedule by using simulation

Let us entitle the "initial schedule" a schedule that does not need to be enforceable, but it includes (planned and staged at the time) all activities of the system. The presumption in the initial schedule is that it will be further improved. As the "initial plan" we will call the plan for the initial schedule.

To obtain the initial schedule, or the plan it is possible to use much methods for planning, which do not calculate with the possibility of random extension of the duration of activities. Duration of the activities, however, it is necessary to appropriately set. If the initial schedule should be as a result, not just a plan, the method is selected, which allows to optimize the order of allocation of resources.

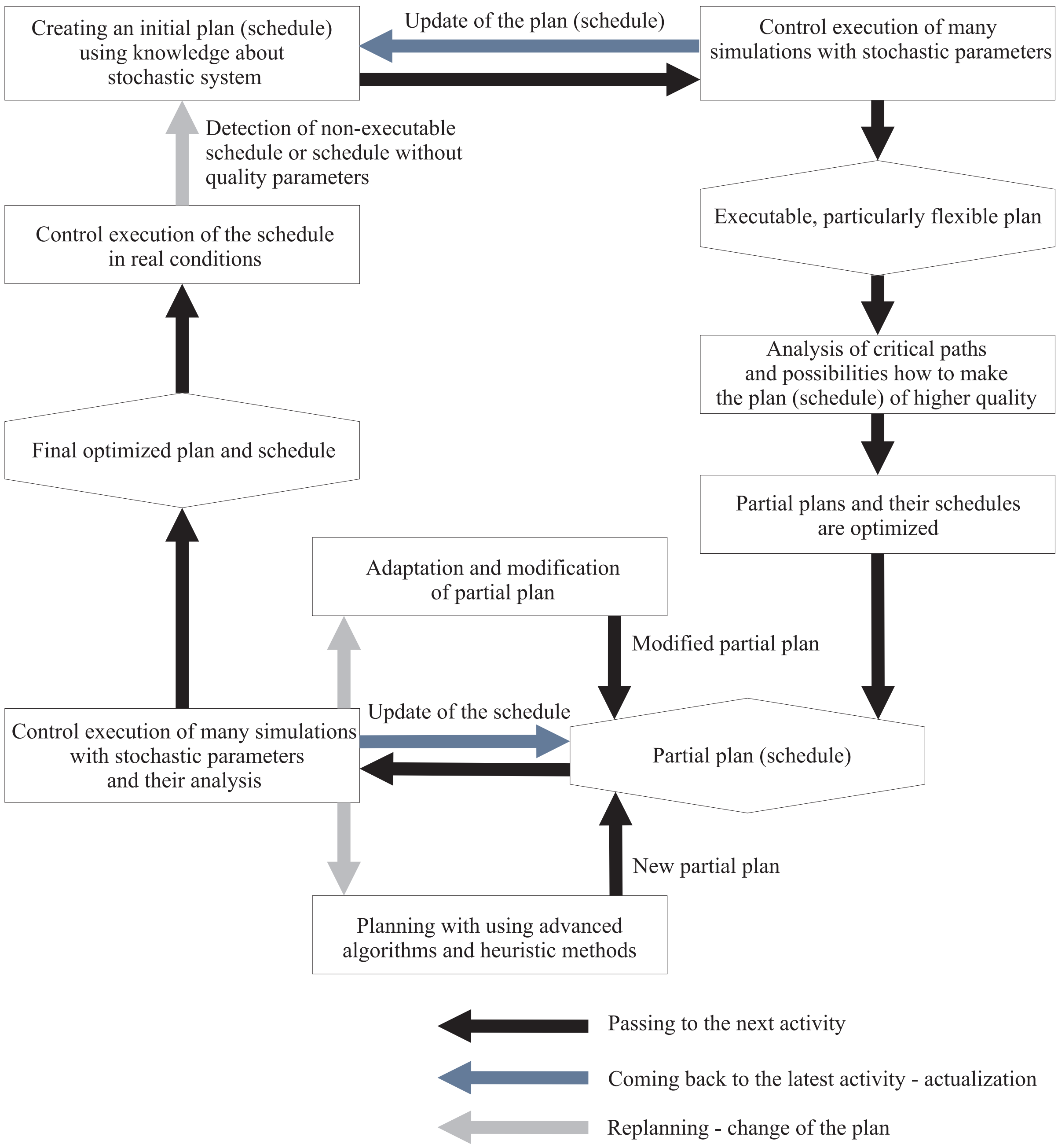

Figure 1: The procedure of making a plan and schedule in the stochastic environment using computer simulation

It is necessary, however, for each activity to determine its deterministic duration. If we have the available statistical information about the probability of extension the activities, we can determine the fixed anticipated durations using the predictive method. They may be used as an estimate of the duration of activities. These fixed durations can be the sum of the duration of the activity, or the time of the assignment of the resource and the average delay.

The obtained initial plan, it is necessary to expose the investigation of its robustness and its further increasing. It is not necessary that the initial schedule fulfils the requirement for robustness, or other criteria. Below we assume that we have a schedule with the lack of robustness, which we will gradually increase. By analogy, but it is possible to do the opposite and from an enforceable plan with high robustness to gradually get a plan with a better use of resources.

Another possibility, how to obtain the initial schedule resistant of random changes in the course of execution, is to obtain the predictive robust solution. It's all decisions are in the process of planning calculated for the worst possible scenario that may arise during execution. It is clear that such a solution would be for the duration of the schedule time-consuming and use of resources would be quite low enough. On the other hand, it is possible to edit the initial robust solution subsequently, what will be shown below. Such a plan is certainly enforceable and maximum robust. If the distribution and the parameters of probability are not known for each random phenomenon, it is not possible to use this technique.

Now it is necessary to realize how it is possible to vary the robustness of the plan or schedule.

The robustness of the schedule may be modified:

- by changing the plan (the order of execution of activities),
- by changing the time reserves of activities,
- by changing the order of allocation of resources to activities.

On the change of the plan itself (order of activities) it is possible to use the standard planning methods. In practical use, however, it is not always possible. For example, the order in which the activities have to be carried out is exactly given.

The first activity to be carried out on the existing initial plan is finding its robustness. The procedure of chapter 2 will be used. From the quantity of replications of simulation run will be but also the amount of information obtained, about the utilization of resources and the behaviour of the entire system. This introductory activity in the initial schedule is on the figure 1 captured in the upper part. After carrying out sufficient quantities of replications or realizations, the current plan will exposure to thorough analysis again. This procedure is repeated until the required criteria are not completed. It should be noted that in this phase of the plan the schedule meeting all the criteria don't have to be found. This may be caused by a lack of resources (computing power, lack of time) and not very suitable initial plan for the examination of a sufficient number of variations of the plan and schedule. From realized options the enforceable plan that best meets the criteria will be chosen (in particular, robustness), or it can be assumed that after the other modifications it will meet. In the next process will this plan and the best schedule to it yet further analysed and altered.

After the end of the first phase of the creation of robust schedule we're getting probably enforceable and with proper application procedures (described in other chapters) the schedule based on the flexible plan. Such a schedule, however, is based on the initial schedule, what can be the reason for its considerable inefficiency. It is possible that the optimal schedule (from the point of view of the robustness, use of resources and other criteria) are dramatically different in the order of the individual activities and the resources that are allocated to individual tasks. Whereas the use of classical optimization methods for the entire schedule is from the design point of view, impossible, it is appropriate to optimize the schedule after the small parts. Now we will accede to the stage of the progressive optimization partial plans.

Whole planning (scheduling) problem is divided into a large number of smaller partial goals, which are always planned first and then solved separately. For each of the obtained schedules are then deployed the procedures to change its robustness. After changes and optimization of a goal schedule is the entire schedule using the execution (simulation) authenticated and a new goal is set. The solution obtained using this technique is suboptimal, because it is derived by connecting goals with restricted status. This technique can be used also without the execution of partial goals. The main advantage of using this technique is to save memory and speed up the calculation. There is a compromise between the reactive and predictive methods. This procedure is very suitable for the high number of uncertainties, when it is a big problem to obtain the overall plan, which would be subsequently also enforceable. In this phase of the creation of robust schedule it is important to set suitably the time horizon, to which will be the schedule created. After its creating its correctness by using the execution will be verified and the new goal status will be determined, which is necessary to achieve. Thus, for example, it is appropriate to plan the layout of the execution of the activities for a period of time (for example, two days, a week). In this phase, it is also advantageous to rotate the two activities, as shown in figure 1. Always it will be on the basis of the results of replications, thus robustness, using resources, etc. decided whether will be created a completely new component plan (schedule), or will the current plan modified.

Partial plans and schedules, it is necessary to vote with respect to the critical points identified on the basis of the analysis already realized simulations. There are especially places with a little use of resources or vice versa too small time reserves. Here it is possible to use revision techniques, mathematical programming, programming with restrictive conditions, and the like. In the case of specific requirements, it is appropriate to draw up an ad hoc algorithm, and then subsequently apply to the area.

The result of all the activities is an optimized schedule that meets the demanding requirements that are imposed on it. The last phase is to verify in real practice. For this verification can be used even microscopic simulation model. In the event that the schedule proves unworkable, it is required an in-depth analysis of the causes. Such situations occur most frequently in case of incorrect review of random phenomena. Also it can be a result of changes that have occurred in the system, after obtaining the necessary input data. The creation of the schedule may continue absolutely from the beginning or just by changing any of its parts.

4. Changing the robustness of schedule by changing time reserves

It is obvious that by increasing the time reserves of activities will increase the robustness of the schedule, whereas one can assume a smaller delay of activities and vice versa. If the latest necessary beginning of execution of activities is missed (LS_i) it is possible to move this start to a later time. This will cause the displacement of the latest necessary end of the execution of this activity, but also a move of some subsequent times in the schedule. If the latest necessary end of activity is missed (LE_i), it is possible to move the end on a later time. This will cause the displacement of some subsequent times in the schedule. It should be mentioned that some times may be fixed and their move is not possible. They are a part of requirements on the schedule (arrival of the train in the station, the duration and the latest termination of activities...). Naturally the situation may occur, that only by increasing the time reserves will not be the schedule with the required degree of robustness obtained.

After a sufficient number of simulations, while each replication is the analysis of the plan (mainly its enforceability) in one accidental execution, although there is a wealth of information on delays of the activities, the schedule analyser will be set on, with which we will be able to increase the time reserves. As a result of simulations, we obtain the measure of robustness of the schedule and also we can get much useful information about the real use of the reserves and the like.

With the shift of times (increasing time reserves) it is possible to apply the four basic procedures:

- prolonging the times of activities is going gradually since the beginning of the schedule,
- prolonging the times of the activities is going mass on the selected activities at the same time,
- prolonging the times of the activities going in cooperation with users,
- prolonging the times of activities selected randomly.

In the event that we decide to expand the time reserves gradually from the beginning, it is appropriate to do so by moving one or only a few of the activities. Each move shall be followed by the creation of a new schedule, whereas it is probably that further activities have to be moved. If this change is shown as acceptable (does not cause distortion of the requirements on schedule, the unenforceable of the schedule...) the reserves of another activity can be increased. Always the activity whose execution time is kept to a minimum is selected. Increasing in reserves is not necessary for each delaying activity. It is important to quantify the total robustness of the schedule. Therefore it is appropriate to fix a measure for average delay of action (values $\overline{WS_i}$ and $\overline{WE_i}$) from the beginning and gradually increase the time reserves. If, despite the fact that the average delay of all activities is below the current rate, the schedule does not reach the required measure of robustness, the average delay of activity moves and the whole process is repeated from the beginning of the plan. Where the measure of activities after changing is higher than the required robustness of the plan we can turn the time reserve to be reduced. This shortening is suitable to do always on the only one activity and only after verifying that the overall rate has not increased significantly, we can leave new values of the time reserve.

To speed up the procedure, it is possible to change time reserves for multiple activities at the same time. First, it is necessary to determine which activities they will be. The procedure is essentially identical to what was described in the previous paragraph. The critical period for average delay of action will be determined (value $\overline{WS_i}$ and $\overline{WE_i}$) and in a single step, all the activities lay down such a time reserve that was expected that delays reach below the required level (the proportion of the value of the $\overline{WS_i}$ and $\overline{WE_i}$ for each task J_i). Subsequently, the new schedule will be tested. If it does not meet the required measure of robustness, the entire process is repeated with smaller allowed average delay. The value of that the average delay reduces, must be substantially less than in progressive change in reserves since the beginning of the plan. From the size of its changes, the quality of both approaches depends on. On the other hand, this procedure is faster as the reserves change for many actions at the same time. In both procedures, the automatically execution is supposed, thus the program analyses and modifies the time reserves itself. Let us say, so as in the previous ways, the maximum average delay of activities. From all of the activities, that this delay exceed, we take out at random, just one activity. The probability of selection of an activity may be the same for all activities, or may be taken into account the varying duration of the average delay. Then the probability P_i that any activity J_i will be chosen, can be determined according to the formula:

$$P_i = \frac{\overline{WS_i} + \overline{WE_i}}{\sum_{i=1}^{n} \overline{WS_i} + \overline{WE_i}}, \qquad (6)$$

where n is the number of tasks, of which the average delay exceeds the set limit.

For the selected activity is subsequently increased the time reserve. Adding the element of randomness is in some cases proved to be an appropriate aid to change the time reserves. With the increasing amount of experiments grows also the probability of finding the optimal robust schedule. However, the determination of criteria for the optimal plan in such systems is very difficult and therefore mostly the achievement of criteria is sufficient.

5. Change of the robustness of the schedule by changing the order of allocation of resources

If you change the order in which the resource is assigned to activities and the schedule is changed, so will be also probably changed its robustness. Inappropriate change will immediately translate into a significant deterioration of all the evaluation criteria of the schedule. Therefore, it is necessary to change the order of allocation in the smallest increments. In this way, we create a new schedule.

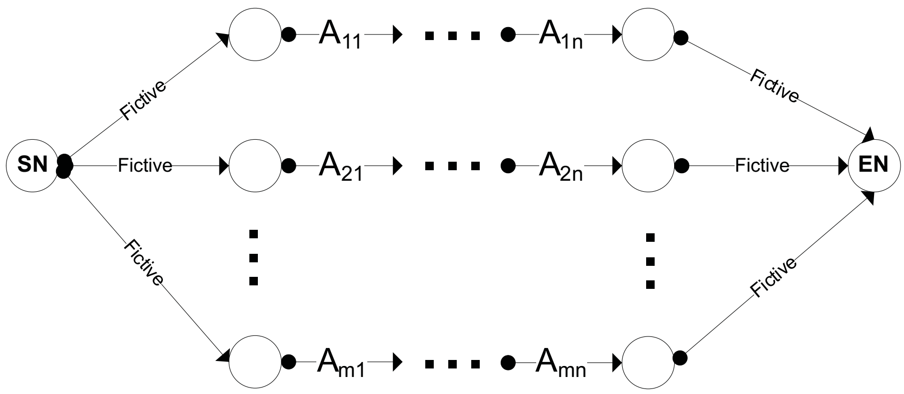

Figure 2: The procedure of creation of a one continuous acyclic digraph

Newly developed schedule will be tested as it has already been mentioned many times. To change the order of allocation of resources, as well as on other activities could also be used the general approaches used in scheduling in the deterministic environment.

In this chapter a new method will be described, which allows you to create a schedule for the plan and to analyse it partially. For the presentation and registration of a plan is sometimes used the net graph. A lot of simulation software use for registration of the plan of activities of individual simulation elements also the net graph. It is well-organized and for the user easy editable form of registration of the plan. The idea to use the net graph describing plans to optimize their schedules came out just from the frequent use of this form of registration in simulation programs. In the existing plans and software, therefore there is no problem to use the existing plans and using this procedure, to improve their properties.

5.1 The procedure for obtaining a single plan written by digraph

The procedure will be clearly explained on the formation of the schedule for the container terminal, on which was also tested, it is, however, fully universal and available to other service and transport systems, or wherever it is possible to use the net graph for writing the plan. Let record the plan in the net graph for each train that comes into the container terminal. This net graph defines in detail the individual activities (landing to the terminal, the train unloading, the train uploading, departure from the terminal ...), which must be carried out on a given train. In order to create a schedule of their work from plans of individual elements of the system, it is needed to create one continuous acyclic digraph. For this it will be subsequently possible to apply the known method for calculation of beginning and supposed end of the activities.

We create the acyclic digraph (figure 2) by defining the one fictional start node (SN) and the one end node (EN). We add the first node of the plan of the concrete train to the start node and the last node of the net graph we add to the end node. Thus, two new edges have been added. Lets have the start time of the work of terminal defined (time of the beginning of the simulation), therefore the time, which will be the beginning of our schedule. The duration of the edge that connects the start node with the first node of the plan will have duration equal to the difference between the arrival time of the train to the terminal (the time of the finding of the train in the simulation model) and time of the start of the schedule. The edge joining the last node of the plan of the train with the end node will have the standard duration of 0.

In this way there was created one coherent acyclic digraph, what allows using multiple methods on its processing. For example, it is convenient to use the method of CPM (Critical path method). This method is commonly known and therefore we do not show the procedure here. Using the method of critical path we can find important information on the chart that has been created in this way, which are available later. The duration of activities can be estimated on the basis of probability of distribution. If they are not available, we can also add the flat rate of constant values. The primary quality of the schedule will suffer, but in the next process will be partly increased.

Using the CPM on the graph, where are the activities on individual edges, we obtain:

- the time of the first possible start of activities emerging from the node,
- the time of the last needed end of activities entering the node.

Thus, we obtained a global plan and the supposed beginnings of individual activities.

5.2 Adding the required resources to the plan

Now it is necessary to add the assign resources to the activities to the plan. Especially the order of allocation is important. Simple idea, that other edges will be added to the plan, which will represent the links between shifts of resources, has proved to be quite complicated.

Lets have a simple net graph shown in figure 3. Let the edges A_{11} and A_{22} require just one resource R_1. In this trivial case, there are two ways to assign a resource, thus two associations of sequence. Either the resource R_1 assigns the first edge A_{11} and then the edge A_{22}, or vice versa. For the registration of the order of the source assignment to edges we will use the following convention. $R_1 = [A_{11},\ A_{22}]$ - the resource is first assigned to the edge (activity) A_{11} and after its termination will be assigned to the edge A_{22}. To the net graph it is necessary

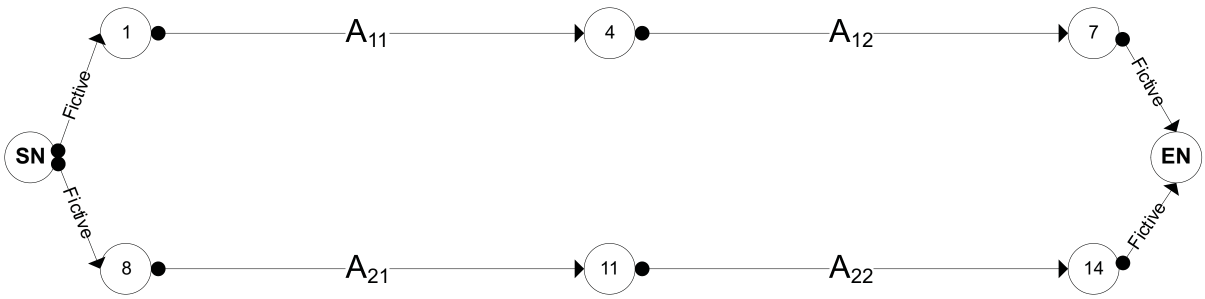

Figure 3: The net graph describing the behavior of the two entities

to add the edge, ensuring that the activity A_{22} will not be executed until the activity A_{11} is not completed and the resource will not move to the place where it may be used for the execution of activity A_{22}.

Figure 4 shows the situation after adding the edges which supplies the movement of the resource. We can see that it was necessary to add to the graph two other edges, which represent waiting for the resource.

It is possible to describe the procedure algorithmically:

1. For each specific resource create the order of allocation of the resource to edges (front Or) that require it. Put all the resources into a set of unprocessed resources *Rn*.
2. Select one resource from the set *Rn* and mark it for the currently processed - *Ra*.
3. From the front, which determines the allocation of the resource to edges (front Or) select the first edge and mark it for the currently processed - *Ac*.
4. If the edge was not yet divided, so divide it by adding two nodes into three parts, the first part presents waiting for the resource *Ra* and its duration, give equal to 0. The second part presents the very edge *Ac* and its duration will remain unchanged. The third part presents a fictional edge with duration of 0. If the edge has already been in the previous work of the algorithm split, so just note that the first part presents waiting also for the resource *Rn*.
5. Add the edge of the last occurrence of the resource to the start of the edge *Ac*. Duration of this edge is equal to the duration of the transfer of resource from its original placement to a place, where it is required to execute the activity *Ac*.
6. If the front Or is not empty, so continue by step 3.
7. If the set *Rn* is not empty, so continue by step 2.

If we let count CPM in the following created graph we obtain quite important information about the length of waiting of the activity for the resource. If the activities in the graph are modelled by individual edges, so the length of waiting for resource is equal to the difference between the first possible start of activities arising from the node, where the waiting ends (node 12) by the first possible beginning of the activities arising from the node, where begins the waiting (node 11). Also, using the method of the CPM we can simply set the schedule. After it is counted we can determine for each activity when it may start and even when it is necessary to terminate it, so as to avoid delays of subsequent activities.

Figure 5 presents the situation with two resources R_1 and R_2. The order of allocation of resources for the figure 5 is: $R_1 = [A_{11}, A_{22}]$, $R_2 = [A_{21}, A_{12}]$.

The table 1 shows a simple plan for a situation of Figure 5 and in the table 2, we can see the schedule that was created by using the method of the CPM. Planned start of execution of the plan is 9:00. Time to move each of the resources in the place of its use has been set for 3 minutes. Assignment of the resource to the first activity does not have the time duration. The obtained schedule fully respects the needs of individual edges on resources and also the time required for the transfer of resources. In real use, it is of course possible to replace the constant time for transfer of resources by the values depending on the concrete place of execution of activities (e.g. the times needed to move locomotives from one terminal to another).

5.3 Methods for optimizing the order of allocation of resources

To obtain the schedule, which use of the resources will be the greatest possible, or will be in as far as possible meet the specified quality parameters it is necessary to determine well the order in which the resource will be assigned to edges.

Inappropriate order for allocation of resources will cause a significant inefficiency of the plan, or the time extension of its execution. The ideal would be to find a way how easily and quickly to find out how to arrange the edges for the resource as so as the schedule as much as possible match the required criteria. Even if there was an exact algorithm, the resulting plan could have the lack of measure of robustness. Here, it is preferable to use experimentation with the order of allocation of resources to the edges. Using the procedure described in chapter 3, it is possible to gradually test the individual variants and to find one that will be the most suitable.

In the definition of the schedule, it was established that every edge may require a specific resource, or a certain amount of resources of that type. But for the work of the algorithm for adding resources to the net graph it is necessary to have defined the order of allocation for a concrete resource. If we assume that it is not very appropriate

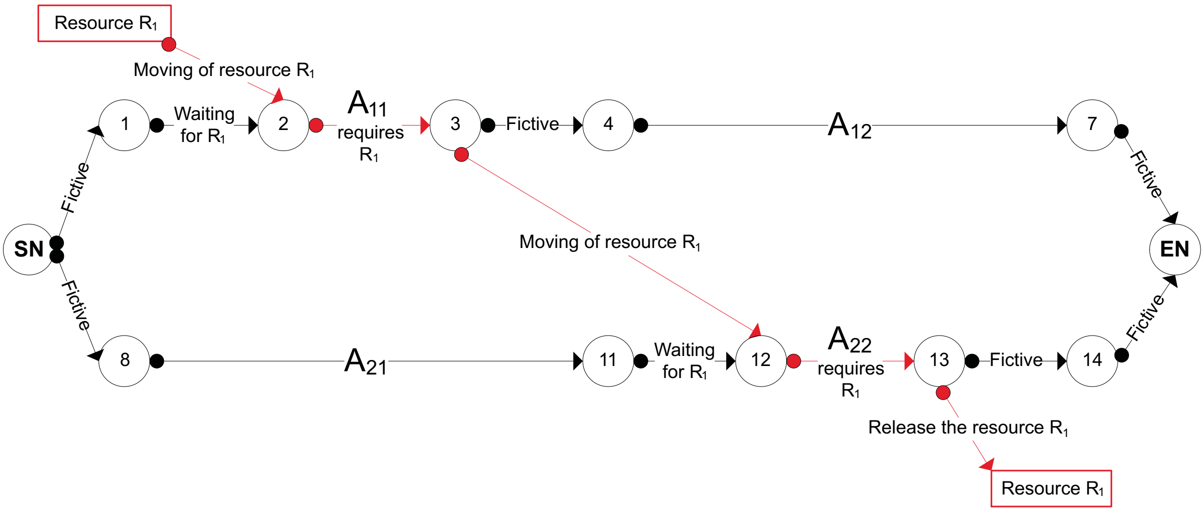

Figure 4: Adding the edge that ensure moving the resource R_1

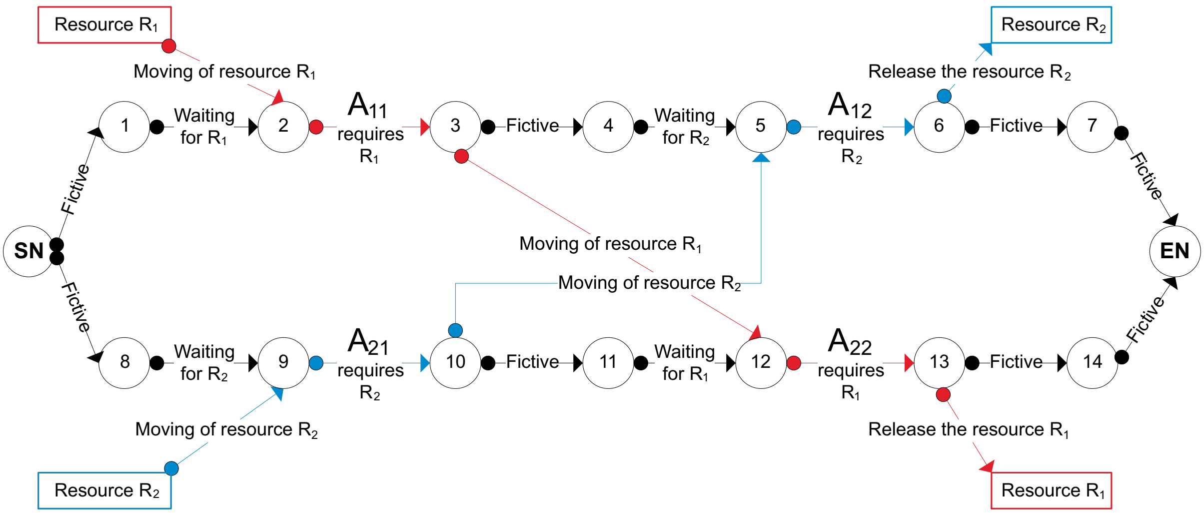

Figure 5: The resulting graph after processing resources

Table 1: The plan of the situation from the figure 5 written in the form of the table

Num.	Activity	Next activity	Required resources
1.	A_{11}	A_{12}	Resource 1
2.	A_{12}	-	Resource 2
3.	A_{21}	A_{22}	Resource 2
4.	A_{22}	-	Resource 1

Table 2: The schedule of the situation from figure 5 written in the form of the table

Num.	Activity	Supposed duration	Time of the first possible start	Time of supposed ending	Time of the latest necessary end
1.	A_{11}	0:10	9:00	9:10	9:12
2.	A_{12}	0:05	9:18	9:23	9:24
3.	A_{21}	0:15	9:00	9:15	9:15
4.	A_{22}	0:10	9:15	9:25	9:25

to allocate activities to one of the two required resources of identical type and to let it wait for the allocation of further, we can deal with the situation by a simple procedure. If the edge requires the resource type, the order of allocation of types of resources will be optimized to edges, not the specific types of resources. On the basis of this order are then allocated the specific resources to the edge.

It is possible to describe the procedure algorithmically:

1. For each specific resource prepare the order of assignment of types of resources to edges (front *Ogr*) which require it. Put all the resources into a set of unprocessed types of resources *Rgn*.
2. Select from the *Rgn* set one type of resource, and mark it for the actually processed - *Rga*. Let the *Rga* represent the front of specific resources.
3. From the front, which determines the allocation of the type of the resource to edges (front *Ogr*) select the first edge and mark it for the actually processed - *Ac*.
4. To the order for the allocation of resources in the edge *Ac* gradually add the required amount of resources from the *Rga* front. If the first *Ogr* edge has a start node identical to end node of the *Ac* edge, so select each added particular resource to the edge and allocate it back to the end of the *Rga* front. This will ensure a balanced use of all resources of the type. If two edges with one common node require the same resource, the same will be assigned to them.
5. If the *Ogr* front is not empty, so continue by step 3.
6. If the *Rgn* set is not empty, so continue by step 2.

A selection of resources from the set of resources of the same type is appropriate to do by cyclic way. The aim is to verify the definitive amount of orders of allocation of the types of resources to the edge and to determine such, that they will meet the required criteria. If it is a small project, it is possible to test all the possible combinations. In the case of large plans, it may not be possible for reasons of time. Here it is necessary to establish an appropriate procedure, which allows changing the order of the types of resources to the edge. For each order, a new schedule will be drawn up and it will be subsequently tested. Very important is that for the order of allocation, it is possible on the net graph immediately to verify whether the solution is realizable at all. If it turns out that some of the requirements of the schedule are not met (beginning or the end of some activities, duration of the schedule etc.), it is clear that the schedule is inappropriate (does not meet the requirements, or is unenforceable) and there is no need to further review using simulation.

Testing the schedule on the net graph is not far from such precision (does not contain some of the detailed characteristics of the system) as its testing using simulation, but it is incomparably faster. On the basis of information about the probability distribution of delays of individual activities (this could be obtained from the real system), for each activity will be created a generator of its delay. Using the "Monte Carlo" method of experiments is always generated the duration of activities with delays. On the basis of such duration of activities, will be on the net graph using CPM calculated the expected behaviour of the schedule. The procedure is the same as for testing of schedule using computer simulation; it is only incomparably faster and less accurate. Using procedures of chapter 2 is set the robustness of the schedule and its other properties. If the schedule meets the required criteria, it is moved to the next review and testing using the simulation.

To change the order in which the types of resources will be allocated to the activities, several algorithms were created and tested. Two methods are presented.

5.3.1 Method using genetic algorithm

This approach is based on the genetic algorithm. Let's title gene, the sequence in which a particular type of resource will be assigned to the edge of the gene. On the gene, only the operation of mutation is allowed. Of the gene are randomly selected two consecutive edges. These will exchange their positions in the gene and the new gene will be created. This new gene will be the subject of testing. The procedure is constantly repeated. It is more than good, when a list of already investigated genes is created and every newly developed gene is reviewed at first that it has not been already tested. This technique is quite sensitive to the existing schedule and in small increments it is trying to experiment with the order of allocation of the resource types to edges.

5.3.2 Method using randomness

As well as in other processes, using of the element of randomness may noticeable change the schedule. However, this may bring its significant improvement. Let it be randomly selected from the sequence of allocation of the edges two edges. These edges can exchange the position in the sequence. A new sequence and the new schedule is created, which is necessary to test. As in the previous method, well here it is convenient to create a list of already tested variants. The new sequence will be investigated only if it has never been investigated.

5.4 Detection of the order of allocation of resources causing the deadlock

The procedure, using which are also the requirements for resources incorporated into the net graph, has further advantage. It allows very easily detecting such allocation of resources, which would have caused the deadlock in real terms. This is a condition where two or more activities are waiting for the allocation of a resource. At least one activity is waiting for the allocation of a resource, but which is bound in another waiting activity. The deadlock is permanent, and if no one from activities will voluntarily release the resource, this state will continue.

To determine, if deadlocks can occur in operating systems, the resource allocation graph is used. If in the graph appears the cycle, it is possible that the deadlock will occur. If the cycle does not appear, it is clear that the deadlock will not occur. The procedure created by myself is significantly better in this respect.

If the cycle appears in the graph while the resources are incorporated into the net graph, it is clear, that the order of allocation of resources will cause the deadlock of the system. This is a very important knowledge that can

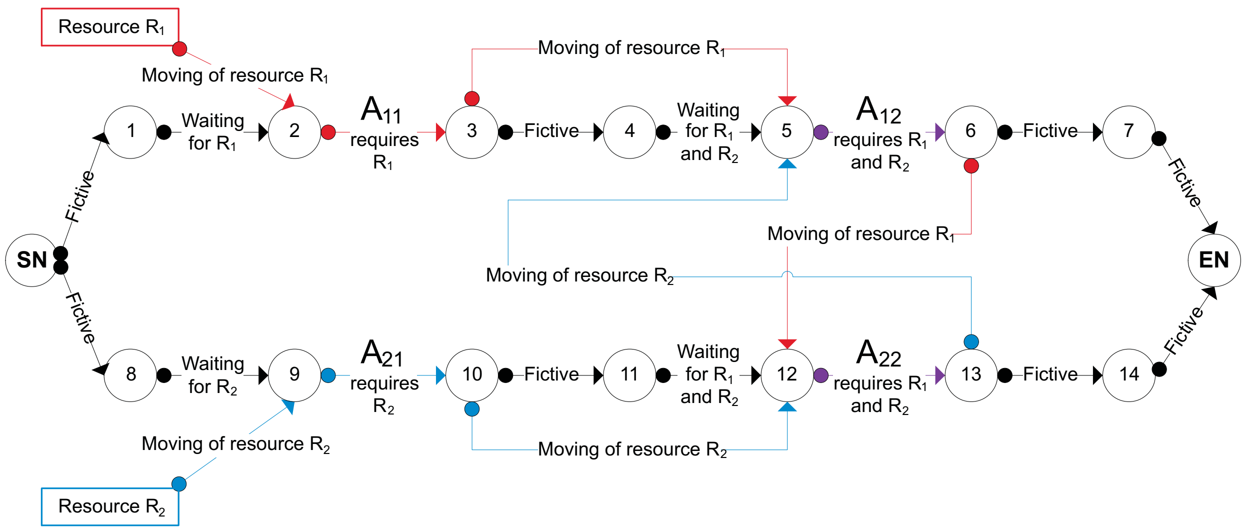

Figure 6: Preview of the order of allocation of resources causing the deadlock

Table 3: The plan of the situation of figure 6 written in the form of a table

Num.	Activity	Next activity	Required resources
1.	A_{11}	A_{12}	Resource 1
2.	A_{12}	-	Resource 1 and 2
3.	A_{21}	A_{22}	Resource 2
4.	A_{22}	-	Resource 1 and 2

significantly speed up the whole process of optimizing the allocation of resources. So are the inappropriate variants of the schedule immediately identified. Unlike the graph of allocation of resources used in operating systems, so we can clearly determine whether the procedure for allocation of resources causes the cycle. This procedure should in principle be used also for the allocation of resources in operating systems, but as the solution of this problem is not the aim of the thesis, we don't give it closer.

The order of allocation of resources: $R_1 = [A_{11}, A_{12}, A_{22}]$, $R_2 = [A_{21}, A_{22}, A_{12}]$.

The figure 6 shows the status after processing allocated resources. As we can see, we discovered a cycle in the digraph. We can confidently state that such allocation of resources is not applicable in practice. The procedure has been shown in the trivial example, where it is possible even without the use of any means to state, that the system will not work. In the net graphs containing a large amount of edges and resources is the detection of unrealisable order of significant benefits. In the realization of the schedule in the simulation model, the deadlock of course can still occur. There are several reasons for this phenomenon (there are not included all the resources in the net graph, some details of a real environment it is possible to capture only in the simulation model...).

6. Conclusions

The article discussed about the way in which it is possible to quantify the robustness of the schedule. This evaluation criterion is applicable when comparing schedules with the occurrence of random phenomena. The prepared method of calculation is sufficiently parametrisable to be usable for a wide scale of problems. The method is beneficial for the whole area of making schedules presented in such environment.

The methodology for the gradual creation of a robust schedule is widely described in chapter 3. By successive steps, it is possible to improve the features and robustness of initial schedule. The procedure uses the ways to increase robustness and to change of the plan described in the following chapters. The methodology provides a new way how to create schedules using computer simulation. Computer simulation in this process does not fulfil only the function to verify the existing schedule, as was hitherto, but its results are actively used in the process of making the robust schedule.

One of the major benefits is the introduction of a completely new way of incorporating the required resources for the activities in the plan written by the net digraph. In this way is a schedule of the work of the resources obtained. This technique of presentation allows extremely fast verification of many combinations and variants of the schedule. The procedure can automatically determine the sequence of the assignment of resources, which would cause the deadlock in the practical execution. It makes easy to work with the plan and using the graphical interface gives the user an overview of the procedure of the optimization of the schedule.

Published procedures were tested on the detailed computer model of the container terminal of the company Hupac near Italian Milan.

References

[1] N. Adamko and Ľ. Sadloň. Aplikácia sieťových grafov v simulačných modeloch železničných uzlov. 2003.

[2] R. Barták. Conceptual models for combined planning and scheduling. Proceedings of CP99 Workshop on Large Scale Combinatorial Optimisation and Constraints, pp. 2-14, Alexandria, VA, USA, October 1999

[3] R. Barták. Integrated modelling for planning, scheduling. Proceedings of PLANSIG 2001, Edinburgh, UK, December 2001

[4] R. Barták. Towards mixed planning and scheduling. Proceedings of CPDC2000 Workshop (invited talk), Gliwice, June 2000

[5] J. Bidot. A general framework integrating techniques for scheduling under uncertainty. Dissetation thesis. 2005.

[6] M. P. J. Fromherz. Constraint-based scheduling. Invited tutorial paper at the American Control Conference (ACCŠ01), Arlington, VA, June 2001.

[7] J. Janáček. *Optimalizace na dopravních sítích*. Edis, Žilina, 2003.

[8] P. Martincová and K. Grondžak. *Operačné systémy*. Edis, Žilina, 2003.

[9] K. Musselman, J. O. Reilly, and S. Duket. The role of simulation in advanced planning and scheduling. *Winter Simulation Conference*, 2002.

[10] S. Palúch. *Teória rozvrhov*. Žilina, 2003.

[11] L. Peeters. Cyclic railway timetable optimization. Dissertation thesis. 2003.

[12] P. Xuan. Techniques for robust planning in degradable multiagent systems. In P. Scerri, et al, editors, Challenges in the Coordination of Large Scale Multiagent Systems, Springer, 2005.

Selected Papers by the Author

P. Jankovič. Plánovanie dopravných systémov v stochastickom postredí. In *Infotrans 2007*, Univerzita Pardubice, 2007.

P. Jankovič, V. Klima. Robust Plans of Processes in the Railway Transport. In *The Present and Future of Modern Transport*, Czech Technical University in Prague, 2008.

P. Jankovič. Robust Planning Using The Computer Simulation. In *Transcom 2009*, University of Žilina, 2009.

P. Jankovič, V. Lendel. Use of Data Mining for Management Decision Making with Applications in Transportation. In *Information Technology in Management and Marketing*, Edis, 2010.

P. Jankovič. Creating a Robust Plan Using CPM Method and Simulation. In *Transcom 2011*, University of Žilina, 2011.

P. Jankovič, O. Šuch. Functional Programming and Discrete Simulation. In *Itat 2011*, PONT s. r. o., 2011.

P. Jankovič, V. Lendel. Modelovanie dopravného systému. In *Journal of Information, Control and Management Systems*, Faculty of Management Science and Informatics, University of Žilina, 2011.

P. Jankovič, V. Lendel. Simulation as a support tool of managerial decision-making with application in conditions of transport. In *Scientific Journal on Transport and Logistics*, Institut Jana Pernera, 2011.