Scheduling algorithms for a video-dubbing studio

Attila Álmos

Department of Measurement and Information Systems,

Technical University of Budapest

{aalmos@mit.bme.hu}

ABSTRACT

In this paper the scheduling problem of a video-dubbing studio is presented. In the problem, events must be scheduled, while several constraints and optimization aspects ought to be considered. The author of this paper studied the related problems, especially the classical timetabling and the job-shop scheduling problems, and their solutions.

Because the adapted method proved to be insufficient, a fundamentally different approach was taken. The algorithms based on the new approach can recognize and exploit the inherence of the problem more deeply, and they satisfy all the important constrains of the problem simultaneously. First, an improved backtrack based algorithm was developed, then a genetic algorithm applying the mechanisms of the previous method was improved. The final method shows faster convergence, than the backtrack solution.

Keywords: optimization, scheduling algorithms, timetabling, job-shop scheduling, genetic algorithms, backtrack algorithms.

1. INTRODUCTION

In a video-dubbing studio before the synchronization of a film the producer's assistant's task is to create the schedule of the scenes, the actors for the films. It is a complex task, containing several constraints and optimization aspects; furthermore even contradictory requirements should be fulfilled for the optimal solution. It takes a long time for a human to attempt to create an admissible timing. So it could be very useful to develop an algorithm, which can make proper schedules in acceptable time, and avoids the disadvantages of the manual method, i.e., there is a requirement to create the schedules by computer.

The schedule gives the timing of all the scenes and actors. Meanwhile creating of the schedule several constraints have to be considered. These aspects can be hard or soft depending on whether the schedule has to satisfy the condition or is a matter of as an optimization aspect. Among the hard bounds there are elementary aspects, e.g. there should not be more than one scene scheduled at the same time. Further hard constraints are that, for example an actor has not free time in a given term. Soft aspects are that, for example an actor ought to wait as less as possible between his scenes.

In the focus of this development there are the so-called genetic algorithms and another approach based on the classical backtrack method. The author of this paper examined the related problems, especially the timetabling and the job-shop scheduling tasks, and their solutions mainly based on genetic algorithm. Some of the genetic algorithm based solutions of the studio-problem were derived from these methods, however, the result showed poor performance, and the algorithms could converge weakly or not at all.

Therefore a fundamentally different approach was taken. The algorithms based on the new approach can recognize and exploit the inherence of the problem more deeply, and satisfy all the important constrains of the problem simultaneously. First, an improved backtrack based algorithm was developed. The final version of this method gives feasible solutions with an acceptable scheduling quality. Furthermore, a genetic algorithm applying the mechanisms of the backtrack method was developed. This approach is capable to make appropriate scheduling, too.

The paper is organized as follows: Section 2 gives a short overview about the related problems and some of the possible solutions. The specification and the analysis of the studio-scheduling problem are provided in Section 3, the adapted solutions are described in Section 4. Section 5 presents approaches based on backtrack algorithm, while Section 6 provides the advanced genetic algorithm containing some idea of the previous methods.

2. ALGORITHMS FOR THE SOLUTION OF RELATED PROBLEMS

The analysis of the problem to be solved in the video-dubbing studio shows that there are two types of the related problems with well-developed solutions which may possibly offer a solution for the studio-problem, as well. First the timetabling problems are presented, and then the job-shop scheduling problems follow.

2.1. Timetabling

Among the optimization problems, one of the classical problem fields is the NP-hard timetabling [1]. The following groups can be distinguished within: school timetabling, university timetabling and exam timetabling.

During the creation of a school timetabling lessons have to be scheduled in a one or two-week timetable. Uniquely a class and a teacher are assigned to a lesson, and the rooms are also given where the lesson can be held. The number of hours per week (or two weeks) is also fixed. Usually the lengths of the lessons are the same, but there can be some lessons whereby more than one ought to be held together. The timeslots for the lessons are the same for every day. There may be lessons, which belong to more than one class, and it could also be possible that different groups within a class have different lessons (for example language lessons, physical education) at the same time.

The final timetable has to satisfy several constraints. The most important of them is that the timetable have to be feasible, i.e., in a given timeslot at most one lesson has to belong to every class (or to every student-group) and to every teacher, and simultaneously at most one lesson has to be scheduled in each room. There are some aspects of optimization, as well, which ought to be considered during the creation of the timetable. For example, there can be teachers who cannot hold lessons in some timeslots, there should not exist free holes, the spreading of the hours of the lessons ought to be uniform, every class ought to have nearly the same number of lessons per day.

The university timetables are similar to the school timetables except some important differences. The lectures on a university are not so homogenous like the lessons in a school, i.e., there are different types of events (lectures, labs and language lessons) which should be handled in a different way and students having the same lectures or labs can belong to different student-groups. The number of the students belonging to an event depends on its type. Usually many more students can visit to a lecture than to a lab. So during making a timetable the capacity of the rooms and the number of the proper rooms for an event are also to be considered. The length of the different types of events digresses also.

The main point in this case is again the feasibility. Further aspects of the optimization are for example the ordering constraints (e.g. the lecture should be scheduled earlier than the connected labs), there should be enough time left for the students to get from one place to the other.

The exam timetabling is a different type of problem; they are relevant on such universities, where are not permanent timetables of the exams, but they are built in every semester. Further conditions are that there is only one occasion for passing an exam per semester, and the length of the time of the exams is short. Before making of the timetable it have to be known which students have subscribed for a given exam, which professors have more than one exam, etc. Usually the possible places of exams are irrelevant. In this case the most important aspect is also that students (or professors) should not have more than one exam at the same time. Further aspects are that a student should have only one exam per day or as less as possible and the spreading of his/her exams ought to be as uniform as possible in the exam period.

The basic problem is the same in the three cases, events have to be scheduled in relatively little number of timeslots, meanwhile given constraints and optimization aspects ought to be considered. These aspects can be divided into the following groups [2]:

- Edge-constraints: They are the most important types of aspects. It means that there have not to exist two excluding event in the same time slot or in the same place.
- Capacity-constraints: It means the capacity of the rooms.
- Event-spreading: It means the spreading of the occasions of the lessons or the students' exams.
- Ordering-constraint: It is relevant in the case of university timetabling. It refers to the order of the lecture and their labs.
- Event-excluding: They regard teachers, who cannot hold lessons in a given time or labors, which cannot be used in given times.

Analyzing the solutions of the above timetabling problems it can be seen that the object-function is an important component in every approach. It evaluates the timetables by providing a number-value as their usability. The main properties of the different methods are their models. The solutions are based on genetic algorithm (GA) [3], simulated annealing (SA) [4] and tabu search (TS) [5]. The latter two mechanisms use a neighborhood relation on the elements of the search space, meanwhile the algorithms make a step, so these methods require a well-defined relation.

In [6] a matrix based model is presented. In the rows of the matrix are the teachers, while in the columns the timeslots and the matrix elements are the events (lessons). The model does not contain the rooms. In [6] a genetic, a simulated annealing, and a tabu search based solutions are described using this model. In the genetic algorithm the chromosomes represented a matrix and on them a sophisticated operation was applied. The disadvantage of this approach is that after the genetic operation the timetables can become unfeasible, so it needs a label replacing (filtering) procedure in every step, which deteriorate the efficiency of the algorithm and the natural genetic mechanisms. The first version of the other solutions (SA, TS) has given very poor performance, so there were made some improving steps, for example the relaxation method.

The paper [7] presented a model based on a list-representation. A list element is a tuple of (teacher, subject, room, class) and a list contains the tuples, which ought to be scheduled in a timeslot belonging to the list. The solution is based on a genetic algorithm, the chromosomes contain the lists of the timetable, and the genetic operations manipulate on the lists. After the operations a preparing procedure is also required.

In [2] an event-oriented model is presented, where, similarly to the previous solution, the events contain information about the subject, the class and the teacher, but the variable resources, i.e. the timeslots and the rooms, are assigned to the event according to the given timetable solution. The inadequate timeslots and rooms are excluded from the value range of the event, and the operations of the applied genetic algorithm insure that these values cannot occur at the event any more. The other constraints are handled in the object function of the genetic algorithm.

Summarizing the presented solutions applying genetic algorithm, they use direct representations in the chromosomes, which allow to work the genetic mechanism directly, but some of them need a label replacing procedure to repair the unfeasible timetables, which deteriorate the natural genetic mechanisms.

2.2. Job-shop scheduling

In this problem field, similarly to the timetabling, the main task is to schedule the events in time, meanwhile there are requirements, which ought to be satisfied. But here the number of timeslots is much higher, than in the timetabling problem and the lengths of the events are very various.

In the basic situation there are different types of machines, jobs, which contain different types of tasks. A machine can work on one type of tasks, and simultaneously only on one task. The tasks are the events, which need schedule. In [8] further general rules are described about the scheduling. There are different types of the job-shop scheduling problems, as follows:

- Open Shop Scheduling Problem (OSSP): There is not any restriction about the order of the tasks.
- Job Shop Scheduling Problem (JSSP): Within a job the order of the tasks is given, but at the other jobs it can be different.

- Flow Shop Scheduling Problem (FSSP): The order of the tasks is given and is the same at every job.

The optimization aspect usually the total competition time of the jobs, but there are other possibilities.

The thesis [2] presents GA based solutions for these problems, too. The methods use indirect similar representations.

For the JSSP the representation is the following: If J denotes the number of jobs and the number of machines is M, the chromosome will contain $J * M$ genes. Every gene can have a value between 1 and J. Genes are read out consecutively. The value J means the following: the first unscheduled task ought to be taken from the job J, and it ought to be scheduled to the first time, where it is allowed. It means that there is a schedule-builder in the algorithm, which solves the problem of scheduling a task to the better period. The values of gene are interpreted as they are from a circular list, i.e., if a job is finished, there will be jobs less, than J, so the gene value is considered with a modulo operation.

For the OSSP the difference in representation is that, because of the looser condition, the genes are doubled, and the new genes describe the task of the given job to be scheduled.

These types of representations cause the loosing of the full meaning of the gene.

3. THE STUDIO-PROBLEM

3.1. Specification of the problem

In a studio-scheduling problem there are various types of events, which must be scheduled. The scenes are the basic events, but the lunchtimes, so-called mass-records and secondary-records also ought to be scheduled.

A scene contains rolls, which are the elementary units of the film and they are placed consecutively. The lengths of each roll are given. It is also known that an actor in which roll, i.e., in which scene plays. There can be some special roll-actor pairs called B-track, which means that the rolls (with the given actor) must be recorded separately from the other actors, otherwise they can be handled as normal scenes.

The lunchtimes are events, which must be scheduled once a day, if the dubbing cannot be finished until a given time-point on the day. Its length is fixed.

The mass-record is a block of records with a usual length of 2-3 hours. It contains roll-actor pairs, where the actor has a few rolls together. There are order-constraints about the schedule of the mass-record and the related rolls. The mass-record must be scheduled in a later term.

The secondary-records are optionally events; it offers a further possibility for the scheduler. By this the scheduler can drag some roll-actor pairs and schedule them in a different term, similarly to the B-track records. Its reason is that the scheduler can decrease the waiting-time of an actor, when he has some role spread widely in the film by collecting these rolls.

The terms of the actors, when they have time for synchronizing, are fixed, and the daytime preferences of the actors can be given optionally.

The hard constraints of the optimization are the followings: only one roll can be scheduled in the same time; all the roll must be scheduled; the events must be in a fixed term of every day; there must be lunchtime, if there are events both in the morning and in the afternoon; ordering constraint of the mass-record (see above); the actors must have time in the term, when their scenes are scheduled.

The (soft) aspects of the optimization are the followings: minimization of the number of the actors' appearances; minimization of the length of the term of the appearances; considering of the actors' daytime preferences; holding of the original order of the scenes (chronology); minimization of the number of the secondary-records; there ought no to be free holes between the events on a day. Some of the soft aspects depend on the director of the dubbing; especially the chronology and the number of the secondary-records are such restrictions.

3.2. Short analysis of the problem

The optimization aspects can be organized as follows:

R0: The most important restrictions are that all the scenes must be scheduled, and only one event can be scheduled at the same time. Further bounds are the terms on the dubbing days, when the scenes can be synchronized, and a soft aspect is that there ought to be as fewer holes between the events on the same day as possible.

R1: These are the absolute restrictions in time. They are derived mainly from the terms when an actor has not got time and from the possible term of the lunchtime.

R2: These are the relative restrictions in time. They are the followings: an actor ought to wait as less as possible between his scenes, he ought to appear in the studio as fewer as possible, and the chronology of the scenes. These aspects are independent from the absolute time, they depend only on, how the events are placed side by side.

The R0 restrictions have to be satisfied anyway. The various solutions attempt to make a compromise between the R1 and R2 aspects.

The evaluation- or fitness-function $f(x)$ is a general requirement of the methods and it is necessary to compare the results of the different types of approaches. This function is derived from the punishment-function $p(x)$ as (1). As greater this function is, as more violations of the restrictions in the schedule are. The various types of constraints and aspects are considered with different weight, and the punishment function is derived as the sum of the partial punishment values.

$$f(x) = \frac{1}{p(x)} \tag{1}$$

4. ADAPTED ALGORITHMS FOR THE STUDIO-PROBLEM

From the previous section it follows that the detailed related problems are not far from the problem of studio-scheduling, so the idea of exploiting the solutions developed for the previous problems, may offer itself naturally. In the followings as a first step the analysis of the performances of the adapted methods is presented.

4.1. Order based GA

This method makes a scheduling from the chromosome in two steps. In the first step it determines the order of the events, then it places the events in the absolute time. This method is closer to the solution of the job-shop scheduling, because it accents the R2, relative aspects.

The ordering problem is a rather general problem (consider the problem class of the Travelling Salesman Problem [9]). There are several approaches for the representation and genetic operations in the GA based solutions, for example order crossover, cycle crossover and partially mapped crossover (PMX) [10]. Finally, in the presented problem another representation was applied. Here the genes show the required absolute order of the events, i.e. it does not use a circular list, when the events are decreasing. When an order position is busy, the event will be put to the nearest free position. The distribution functions of the various types of events depend on the natural placing tendency.

To solve the second step of the scheduling in the chromosome there are some extra genes which describe the starting time-points within the days and the number of the event on the given days.

The performance of this approach is very poor. Its cause can be summarized as follow: Although, the ordering step insures that the relative order could converge, through the feedback of the object function, the R1 aspects make a bad influence on the genetic mechanism and mess the convergence.

4.2. Absolute time based GA

In this approach, similarly to the solutions of the timetabling problem, the aspects of the absolute time (R1) are in highlight. In the representation of the GA, a starting time-point is assigned to every event, and the algorithm tries to place the events in the given time interval.

There can be two types of problems. The first one is if two or more events clash. In this case a similar repairing procedure is used, what was mentioned at the description of timetabling solutions. This procedure puts the event to the nearest place, where there is enough slots for it. The other problem is the gaps between the events in the final scheduling. There are several methods to solve this problem, but they all have disadvantages. The procedure can pull the events to remove the hole, or place a proper event in the hole. The first solution corrupts the absolute time similarly to the repairing method to solve clashing. The second procedure messes the relative order of the events. However, the various solutions were tried, but they did not offer acceptable results.

5. BACKTRACK ALGORITHMS ON THE CONVENTIONAL SOLUTION

Because the simple approaches cannot work on a respectable performance, a fundamentally different approach was taken. This method can recognize and exploit the inherence of the problem more deeply, and it satisfies all the important restriction of the problem simultaneously.

5.1. The simple backtrack solution

The first presented solution is a simple backtrack based algorithm. Here in one step one event is scheduled. During the scheduling of an event two questions are answered. These are firstly which event ought to be scheduled, and secondly when. The event and the term are selected from a priority-list, which is generated on every level of the backtrack algorithm. If there is a back-step in the algorithm, the next choice follows from the list.

In the method the R1 aspects are the main points, and it cannot handle the secondary-records. In each step the algorithm firstly calculates the possible time period for the event scheduling, then it checks which parts of this period are free term of the studio, i.e., events are not yet scheduled for this period. The free terms of a scene are derived from the spare times of the actors who are playing in the scene, determined by the intersection of the terms of the spare times. The free times of the additional events as lunchtime, mass-record are calculated in a similar way. The free lunchtime is determined depending on its term and the reserved times in it. The free time of the mass-record is derived from the term when it can be scheduled satisfying the ordering constraint.

The priority-list depends on the free times of events. The list is truncated depending on the object-value of the elements in it, and the number of the elements. So there is a limit for weakness of the object-value and there is a limit for the number of them.

The other decision of the step is to determine the term, when the event is scheduled. It depends on only one aspect, that is which term is the least acceptable by the events. The further aspects and restrictions are respected in the punishment function.

The punishment function is a sub-additive function, i.e., its dominant part is additive, which means that in every step the dominant part of the new punishment value is the sum of the old value and the punishments come from the current step. Exploited it as a performance enhancement the algorithm compares the punishment value of the best scheduling with the current sub-value. If the latter one is greater than the best result, it means that there is little chance to create a better timing from the current, incomplete scheduling. So in this case the algorithm takes a back-step and the current scheduling is not continued. Also a back-step is executed when the incomplete schedule becomes unfeasible.

Because of the mechanism of the algorithm the schedules satisfy the R1 and R0 restrictions except the condition about the holes. The R2 constraints get on by the objective function, so these are rather weak aspects, similarly to the simple GAs. They ought to be built into the mechanism of the algorithm. As a summarizing this approach cannot make a compromise.

5.2. The improved backtrack method

In this method there are fundamental changes, owing to this, the algorithm can show better performance than the previous one. For it, the method uses some auxiliary tools, which make the algorithm more efficient. This approach can handle the secondary-records, which increases the performance of the method further.

First of all the scenes are put into groups by that which actors play in them. Those events are in the same group, which contain the same actors. Then these groups are scheduled in the algorithm instead of the scenes. With this idea the numbers of steps are reduced, because there are much fewer groups, than events. Because of it the algorithm becomes faster, however, the order of the scenes is restricted and it messes the aspect of chronology.

Another tool to improve the performance is related to the groups of scenes. On the pairs of groups a distance-function (2) is defined, which has as less value as less the difference is between the actors in the two groups of scenes. The relation of neighborhood means the least value between the scene-groups, which is possible in the given case.

$$d : G \times G \rightarrow \boldsymbol{R}^{+} = \{x \in \boldsymbol{R} : x \geq 0\} \quad (2)$$

The mechanism of the backtrack algorithm is similar to the previous one, but there are some relevant variances. The schedule is creating to add events to it step-by-step, but it happens in the order of the scheduled time, i.e., the first scheduled event is put to the beginning of the dubbing term. It means that in the current step the next event to schedule is chosen depending on which event is the fittest to the current term. By this property the events are sorted in a priority order. The first element of the priority-list is chosen in the first case of the given level, later when back-steps occur, the next events are chosen from the list. The list is truncated similarly to the previous algorithm by the weakness of the elements and by their numbers.

On a level the priority-list of the events contains only such events, which can be scheduled on the given term. These events can be scene-groups, lunchtime and mass-record, but they must not have been scheduled yet. To sign that after an event there is no more events on the day, there is a pseudo-event called event end-of-day.

In the initial step and after an end-of-day event the start time-point of a day is set to a fixed value, and after a back-step this time-point is increased by a given value. The list is generated in different ways if the next event is the first of the day or a latter one. The only aspect, which is the same in both cases, is the possibility of scheduling of an event to the given term, which means that the event has not been scheduled yet and in the scene-group the actors have time or the special events are allowed in the term. It is the most important aspect. The priority-lists are built by an aspect-function, which contains the weighted values of the relevant aspect. A weight is as greater as more important. The value of this function is called the fitness of the given event. But in the followings the aspect-function does not contain the aspect of possibility of scheduling, it is an excluding condition.

First the scenes are scheduled, then the other special events, like the lunchtimes, mass-record, secondary-records and the end-of-day events.

The fitness of an event to be scheduled to the starting time-point of a day is determined mainly by its free time. There are further aspects with less weight: daytime preferences and the number of the neighbors of the events. The second aspect is as less as more preferable, because an event with more neighbors ought to be scheduled into the inner part of a day.

The fitness of an event to be scheduled to a later term of the day is determined mainly by the distance from the last scheduled event. If an event is close to the last one, it means that there are only a few different-actors between them, i.e., it supports the majority of aspects R2. The further aspects with less weight are the followings: the free time of the events and the daytime preferences. The first one is respected with the greatest weight if it is near to zero.

The priority-list contains also the special events like lunchtimes, mass-record and secondary-records. These events are put to the beginning of the list, if they are schedulable to the given term. The secondary-records are derived from the scene-groups, which has actors, who decrease the aspects R2. These actors with their scenes are moved to secondary-records. The possibility of scheduling a secondary-record is that all the scenes of an actor are scheduled. The best solution is that the secondary-records belonging to an actor are placed immediately after his last scene-group. So it has the greatest priority among the special events. Finally, the event end-of-day is put into the priority-list, if there is no more schedulable event for the day.

To increase the performance of the backtrack algorithm, one of the best way is the decreasing of the number of the possible tracks on the levels. For this purpose the quality of the uncompleted schedule is traced. If this becomes unfeasible, then a back-step will be performed. Another way to make the method more efficient is to exploit the sub-additive property of the punishment-function, so the algorithm can take a back-step if the value of this function transcends the ever-best result.

This method provides acceptable results. It is owing to that fact that the approach can make compromise among the aspects, especially between the aspects R1 and R2. Usually derived from these aspects contradictory effects will occur, but this method can keep it well in hand. The aspects R0 and R1 are handled in a hard way, i.e., it is impossible to violate these restrictions. The scene-groups and distance-function support the most of the aspects R2. If these tools are built into the mechanism of the algorithm, then these restrictions can get on, too. Only the chronology of the scenes cannot get on, but this is a soft aspect and the most dubbing directors do not take interest in it. Another disadvantage of this method is that the backtrack algorithm does not handle each step in a uniform way. It means that the beginning steps are deeper in the algorithm-tree, so they can be changed rarely, while the later steps, for example those scenes, which can be scheduled only on the last day of dubbing, can be changed easily. This asymmetry can also decrease the performance.

A further disadvantage of the backtrack algorithm is that despite the performance increasing steps, it may take a long time to find the first acceptable result. It is particularly likely if the restrictions are too strict. To eliminate this problem a GA based solution was developed.

6. ADVANCED GENETIC ALGORITHM

The advanced genetic algorithm applies the idea of the last backtrack algorithm. The method takes the same steps during the creation of schedule, but the decisions in the steps are not determined directly by the order of an event in the priority-list, but it is modified by a permutation, which is in the assigned gene of the given chromosome. So this method uses the scene-groups and distance-function, too. The main differences are derived from the different mechanism.

The previous GAs show that the representation is the main point of a GA based method. In this approach there are lists in the genes, i.e., the chromosome is a 1D-array of lists. A list represents a permutation. This permutation modifies the original priority-list assigned to the scene-groups. The priority-list describes, which scene-groups ought to follow after the assigned group according to the decreasing priority, containing all the groups. These lists are used globally, i.e., they are generated before starting of the GA, then they are fixed. So the order in the lists ought to be determined by global aspects. These are the followings: the most important aspect is the distance between the event assigned to the list and the given event. The nearer one has the greater priority. If there are some events with the same distance, they are compared by their free-times, and the less one has the greater priority.

There is a further priority-list to determine the first scene-group on the day. This list is similar to the previous ones, but the aspects are like at the last backtrack, i.e., its free time, with less weight: daytime preferences and the number of the neighbors of the events. The list also has an assigned gene in the chromosome.

In the chromosomes the lengths of the lists are much shorter, than the priority-list, because usually the first elements are relevant, so the genetic algorithm ought to converge by these ones. The represented permutation lists modify the beginning of the priority-list, its tail is fixed. The result order is respected at the choice of the next scene-group. The first group is selected, which can be placed after the current event. This method is modified by the special events. If a special event can be scheduled, similarly to the backtrack method, it is attempted to schedule before any scene-group.

The starting time of the days is fixed to a given time-point. This time-point is modified in only some case. If there is not any event to be scheduled, the time-point is postponed. If there were not enough time for the full dubbing, the time-point would be placed earlier.

In the chromosomes the genes contain permutation lists, i.e., the lists contain the first natural numbers uniquely. The genetic operators work on the genes separately. One of the used crossovers is a uniform type, which swaps the genes in the children's chromosomes independently from each other. Another, advanced type of the used crossovers can derive the children's chromosome as a composition of the parent chromosome. It determines the positions of the numbers in the two parents, and the final order is organized by the average position of the numbers. The mutation swaps elements in the permutation list individually with a probability with geometric-distribution, so those elements are mutated often, which are used in the scheduling.

The fitness-function is the same as the evaluate-function (1). There three types of selections are used: Roulette Wheel, Rank Based and Tournament, and there is two types of reproduction: Simple and Steady State.

In the algorithm there are more decision making possibilities, where local or global optimization aspects can be considered, for example the priority-list of the events. The latter one can contain only that information, which is available at the beginning of the algorithm. The advantage of the first choice could be that the algorithm can choose better steps, so the complete schedule may be better than the global method. However, from another point of view as more aspects are considered locally, as more the genetic mechanisms decrease, because the local aspects make it more indirect, i.e., the representation provides less direct information about the schedule. Finally the effect of the advantage of the local method is much less then this disadvantage, so the algorithm contains the global approach anywhere it is possible.

During running this algorithm the quality of the results was nearly the same like the improved backtrack method. There are some differences, but they are not relevant (Table 6.1.). Depending on the reproducing, selecting and crossover method applied in the genetic algorithm the results showed a little variety (Table 6.2.). The advanced crossover method showed better results.

Table 6.1. The results of the backtrack and the genetic algorithm

	Best results[1]	Average results[1]
Backtrack algorithm	10,97	-
Genetic algorithm	11,28	8,15

Table 6.2. The results of the different method of the GA

	Best results[1]	Average results[1]
Simple crossover	9,32	7,43
Advanced crossover	11,28	8,87
Steady state reproduction	10,63	8,03
Simple reproduction	11,28	8,27
Roulette Wheel selection	11,28	8,13
Rank Based selection	10,93	8,20
Tournament selection	9,89	8,12

The greatest difference between the two approaches, namely the backtrack and the genetic algorithm, is that the speed of the latter one is much better. The GA can converge after about 1000 step, while the backtrack must run 5-10 times more long to find an acceptable result.

7. CONCLUSIONS

In this paper the scheduling problem of a video-dubbing studio was presented. The author of this paper studied the timetabling and the job-shop scheduling problem, which are related problems to the studio-scheduling problem, and some of their solutions mainly based on genetic algorithm.

[1] The results are derived from scaled fitness values.

These problems are rather similar to the studio-problem, but there are special requirements in it, which cannot be satisfied by the adapted solutions. These GA based methods use direct representation, but they are not able to handle all the types of the constrains of the problem together. There are two important types of the restriction, namely one of them concerns the scheduling of the events in the absolute time, and the other one concerns the relative placement of the events. They cannot be considered simultaneously.

The mechanism of the new approach is based on a backtrack method, where the steps are taken so that they insure that the schedule, which is built event by event, will satisfy the constraints of the absolute time. An event can be scheduled to the next term, if it does not cause any violation. Besides, the method applies tools, which help to consider the aspects of relative location of the events. So the relations between the last event and the others determine the choice of the next event.

The method of the advanced genetic algorithm exploits the idea of the previous approach. The chromosomes code the steps of the backtrack algorithm in a general way, so the local effects of the schedule building cannot get on. Consequently they cannot decrease the genetic mechanism. The advantage of the GA based method is the high speed of convergence, while the backtrack algorithm may require 5-10 times more.

8. ACKNOWLEDGMENT

I would like to thank to my advisors, Annamária R. Várkonyi-Kóczy, associate professor at the Department of Measurement and Information Systems and Márk Fék, PhD student for their help.

9. REFERENCES

[1] Even, S., Itai, A., Shamir, A.: On the complexity of timetable and multicommodity flow problems, SIAM Journal of Computing 5:4, 691-703, 1976.

[2] Fang, Hsiao-Lan: Genetic Algorithms in Timetabling and Scheduling, Ph.D., Department of Artifical Intelligence, University of Edinburgh, 1994.

[3] Whitley, Darell: A Genetic Algorithm tutorial. Technical report, Colorado State University, March 1993.

[4] Kirkpatrick, S., Gelatt, C. D., Jr., and Vecchi, M. P.: Optimization by simualted annealing. Science, 220, 671-680, 1983.

[5] Glover, F.: Tabu search - Part I, II. ORSA Journal on Computing, 1, 190-206, 1989, Journal on Computing 2, 4-32, 1990.

[6] Colorni, Alberto; Dorigo, Marco; Maniezzo, Vittorio: Metaheuristics for High-School Timetabling, Computational Optimization and Applications Journal, 1997.

[7] Abela, J., Abramson, D.: A Parallel Genetic Algorithm for Solving the School Timetabling Problem, 15 Australian Computer Science Conference, Hobart, Feb 1992.

[8] Conway, Richard W., Maxwell, William L., Miller, Louis W.: Theory of Scheduling, Assison-Wesley Publishing Company, 1967.

[9] Grefenstette, J. J., editor. Proceedings of the First International Conference on Genetic Algorithms and their Applications. San Mateo, Morgan Kaufmann, 1985.

[10] Goldberg, David, E.: Genetic Algorithms in Search, Optimization and Machine Learning, Reading, MA: Addison-Wesley, 1989.