Hartmann, Sönke

Scheduling medical research experiments - an application of project scheduling methods

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 452

Provided in Cooperation with:
Christian-Albrechts-University of Kiel, Institute of Business Administration

Manuskripte
aus den
Instituten für Betriebswirtschaftslehre
der Universität Kiel

No. 452

Scheduling Medical Research Experiments — An Application of Project Scheduling Methods

Sönke Hartmann

August 1997

Sönke Hartmann[1]
Institut für Betriebswirtschaftslehre
Lehrstuhl für Produktion und Logistik
Christian-Albrechts-Universität zu Kiel
Olshausenstraße 40
24118 Kiel, Germany

e-mail: hartmann@bwl.uni-kiel.de
FTP: ftp://ftp.bwl.uni-kiel.de/pub/operations-research/
WWW: http://www.bwl.uni-kiel.de/bwlinstitute/Prod/

[1]supported by the *Studienstiftung des deutschen Volkes*

Abstract

We consider a medical research project that was carried out at the University of Kiel (Germany). This paper deals with the task of scheduling this real-world project. The original problem is shown to be an instance of an extension of the well-known resource-constrained project scheduling problem (RCPSP) with makespan minimization as objective. We obtain a moderately sized problem which consists of 62 activities with time-varying resource request and 27 renewable resources with time-varying availabilities. Subsequently, a genetic algorithm that has recently been suggested for the RCPSP is employed to schedule the real-world project. Within less than one minute, a schedule is obtained which is proved to be optimal. Moreover, we compare the schedule found by the genetic algorithm with the hand-made one according to which the original project was performed. The makespan of the computed schedule is more than 10 % shorter than the hand-made one. We conclude that computer based systems are useful for scheduling projects of moderate size, and that the genetic algorithm considered here is well suited for solving real-world project scheduling problems.

Keywords: Project Management and Scheduling, Resource-Constraints, Real-World Project, Genetic Algorithms, Computational Results.

1 Introduction

Scheduling a project of medium or large size usually is a tedious and time-consuming task. A particular difficulty is to obtain a good schedule in terms of a given objective while observing numerous constraints. Consequently, it is advisable to make use of a computer based scheduling system.

In this paper, we consider a medical research project that was performed in 1994 according to a hand-made schedule. Our goal is to show that scheduling this project using an algorithmic tool would not only have made the scheduling process more convenient, but would also have resulted in a much better schedule than the hand-made one. Additionally, we demonstrate that a computer based procedure allows to evaluate several alternative schedules that are determined for different scenarios. Our results suggest to solve a wide range of project scheduling problems by computer, especially if they have a similar structure like the one considered here, such as many medical, biological, and pharmacological research projects.

In order to provide a basis for a scheduling algorithm, we formalize the original research project as an instance of the well-known resource-constrained project scheduling problem (RCPSP) which additionally covers resource availability and request varying with time. For the classical RCPSP, the currently most powerful exact algorithms of Brucker et al. [2], Demeulemeester and Herroelen [4, 5], Mingozzi et al. [19], and Sprecher [22] are able to solve instances with 30 activities and 4 renewable resources, whereas problems with 60 activities and more are still intractable if the resources are scarce. Resource requirements that vary with time, as to be considered for the research project in this paper, make the scheduling problem even harder to solve, see Sprecher [21]. Consequently, we have to use a heuristic procedure to obtain a near-optimal schedule for the research project at hand. Numerous heuristics have been proposed for the RCPSP, see e.g. Kolisch [11, 12], Kolisch and Drexl [13], Baar et al. [1], Kohlmorgen et al. [10], Lee and Kim [15], Cho and Kim [3], and Hartmann [8]. In order to schedule the medical research project, we employ the

genetic algorithm proposed by Hartmann [8]. This approach was shown to outperform two other genetic algorithms (cf. Lee and Kim [15] and Özdamar [20]) as well as a priority rule based sampling method (cf. Kolisch [12]) on a set of systematically generated standard test instances (cf. Kolisch et al. [14]). Moreover, it can easily be adapted to the RCPSP with time-varying resource availability and request.

The remainder of the paper is organized as follows: Section 2 provides a description of the RCPSP with time-varying resource capacity and requirement. After the description of the medical research project in Section 3, the real-world data are transformed into an instance of the extended RCPSP in Section 4. Section 5 summarizes the basic characteristics of the genetic algorithm (GA) which is applied to schedule the project. Section 6 gives the results of our computational experiments. Finally, Section 7 states some conclusions.

2 Problem Description

Within the classical resource-constrained project scheduling problem (RCPSP), the activities of a project have to be scheduled such that the precedence and resource constraints are observed and the makespan of the project is minimized. The problem considered here extends the RCPSP in that it allows the resource availabilities as well as the resource requirements to be varying with time. We will denote this problem as RCPSP/τ for short. It can be described as follows.

We consider a project which consists of J activities (jobs) labeled $j = 1, \ldots, J$. Due to technological requirements, there may be precedence relations between some of the jobs. These precedence relations are given by sets of immediate predecessors P_j indicating that an activity j may not be started before all of its predecessors are completed. The precedence relations can be represented by an activity-on-node network which is assumed to be acyclic. We consider additional activities $j = 0$ representing the single source and $j = J + 1$ representing the single sink activity of the network.

With the exception of the (dummy) source and (dummy) sink activity, each activity requires certain amounts of (renewable) resources to be performed. The set of resources is referred to as K. Denoting the set of periods as T, the availability of resource $k \in K$ in period $t \in T$ is given by $R_k(t)$.

The processing time (duration) of an activity j is denoted as p_j. Its request for resource k in the t-th period it is in process is given by $r_{jk}(t)$, where $t = 1, \ldots, p_j$. Once started, an activity may not be interrupted. W.l.o.g., we assume that the dummy source and the dummy sink activity have a duration of zero periods and no request for any resource.

We assume the parameters to be nonnegative and integer valued. The objective is to determine a schedule with minimal makespan such that both the precedence and resource constraints are fulfilled.

3 The Medical Research Project

We consider a medical research project that took place in 1994 at the University of Kiel (Germany). A detailed description of this project as well as the results can be found in Fitting [6], see also Löser et al. [17]. The research deals with the relationship between polyamine synthesis and cancer. While it has been shown by e.g. Löser et al. [16] that polyamines

cannot be used for cancer diagnosis, the inhibition of polyamine synthesis provides a successful approach to tumor growth inhibition, cf. e.g. Marton and Pegg [18]. The objective of the research project considered here is to examine the impact of certain medicaments and a special diet on the polyamine synthesis and on the stimulated pancreas growth of rats. The medicaments are given to (male) rats. After a prespecified number of days, a rat is sacrificed. Immediately afterwards, several organs (pancreas, liver, prostate, and small gut) are examined.

In the following, we give a description of those details of the medical research project that are relevant for scheduling the project.

- **Experiments:** We have 6 medicaments, which will simply be denoted as $a, \ldots, f$, that are tested in 7 specific combinations $A, \ldots, G$. Each medicament combination is tested over several specific durations on rats which are given a normal diet as well as on rats which are given a special diet. If special food is given, the test duration is 7 days. Otherwise, for the normal diet, we have test durations of 2, 3, and 6 days. Any medicament except for a and b requires an additional preparation time of 2 days when tested in accordance with the normal food, resulting in total durations of 4, 5, and 8 days for a corresponding combination. Testing a medicament combination with a given diet type over a given duration is called an experiment. Once an experiment has been started, it may not be interrupted.

- **Repetitions:** Each experiment is repeated several times in order to allow a statistical evaluation. The number of repetitions varies from experiment to experiment due to the following reasons: First, some of the experiments have already been performed in a similar way, therefore only a few repetitions are sufficient for obtaining reliable results (clearly, the number of rats to be sacrificed must be kept as small as possible). Second, some of the medicaments are scarce and expensive. In fact, three of the experiments are not performed at all, i.e. we have 0 repetitions. Table 1 displays the number of repetitions with respect to medicament combination, duration, and diet type of the experiments of the original project.

- **Temporal arrangement:** Several repetitions should be carried out in parallel, that is, they should start and finish at the same day. This keeps the schedule easier to survey. Moreover, it allows the researcher to dose the medicaments more exactly. However, performing too many repetitions of an experiment in parallel may cause systematical errors. Especially the last day of an experiment (i.e. the day on which the organs are examined) is assumed to be critical in this sense. Therefore, the repetitions of one experiment should finish on at least 2 and at most 3 different days.

- **Examination days:** Due to limited laboratory capacities, the organs of a rat can only be examined on Wednesday, Thursday, and Friday, which will thereafter be called examination days throughout this paper. As examinations must take place on the last day of an experiment, an experiment must finish on an examination day. In addition, the capacity of some equipment in the laboratory is limited: On each examination day, the organs of at most 6 rats can be examined. The calendar showing the examination days is given in Table 2. The days are consecutively numbered, up to the planning horizon of 84 days.

medicament	*diet:*			*normal*				*special*
combination	*duration:*	*2*	*3*	*4*	*5*	*6*	*8*	*7*
$A = \{a\}$		4	2			2		2
$B = \{b\}$		4	4			4		2
$C = \{b, c\}$				6	9		5	3
$D = \{b, d\}$				5	7		5	3
$E = \{b, c, e, f\}$				5	6		0	5
$F = \{b, d, e, f\}$				6	5		0	5
$G = \{c\}$				4	3		3	0

Table 1: Experiments and repetitions

Mo	*Tu*	*We*	*Th*	*Fr*	*Sa*	*Su*	*Mo*	*Tu*	*We*	*Th*	*Fr*	*Sa*	*Su*
1	2	3	4	5	6	7	8	9	10	11	12	13	14
W	W	W	W	W			W	W	W	W	W		
		E	E	E					E	E	E		
15	16	17	18	19	20	21	22	23	24	25	26	27	28
							W	W	W	W	W	W	W
		E	E	E					E	E	E		
29	30	31	32	33	34	35	36	37	38	39	40	41	42
W	W	W	W	W									
		E	E	E					E	E	E		
43	44	45	46	47	48	49	50	51	52	53	54	55	56
W	W	W	W	W									
		E	E	E					E	E	E		
57	58	59	60	61	62	63	64	65	66	67	68	69	70
W	W	W	W	W	W	W	W	W	W	W	W	W	W
		E	E	E					E	E	E		
71	72	73	74	75	76	77	78	79	80	81	82	83	84
W	W	W	W	W			W	W	W	W	W		
		E	E	E					E	E	E		

Table 2: Calendar — working (W) and examination (E) days

- **Working days:** The researcher is allowed to specify some days for vacation and for evaluation of some preliminary results. On the remaining days which are called working days the researcher is in the laboratory. The first working day of the project was June 6, 1994. The original working days are entered into the calendar of Table 2. The tasks of the researcher are examining on the last day of an experiment as well as feeding and giving the medicaments. During the duration of an experiment based on normal food, the researcher must be in the laboratory on each day. However, an experiment related to special food (which always takes 7 days) requires his presence only on days 1, 5, 6, and 7, while on the other days, feeding and giving medicaments are not necessary. Due to the high effort of feeding, giving medicaments, and examining, the researcher can handle only 20 rats at the same time. That is, on each working day at most 20 experiment repetitions requiring his presence in the laboratory can be processed.

- **Objective:** The researcher is responsible for determining a project schedule which observes the restrictions given above. His objective is an early project completion; this also leads to free laboratory capacities for further research projects.

4 Transformation to the RCPSP/τ

4.1 Resources

We start the transformation by defining the set of renewable resources which will be given by $K = \{1, \ldots, 27\}$. The capacities of the first two resources are varying with time. Therefore, we have to specify the set of periods first. As we have a planning horizon of 84 days (see Table 2), we obtain $T = \{1, \ldots, 84\}$, and each period corresponds to one day. Now we are ready to define resource $k = 1$ which reflects the laboratory equipment. On day $t \in T$, its capacity is given by

$$R_1(t) = \begin{cases} 6, & \text{if } t \text{ is an examination day} \\ 0, & \text{otherwise.} \end{cases}$$

The second resource corresponds to the researcher. His capacity on day $t \in T$ is

$$R_2(t) = \begin{cases} 20, & \text{if } t \text{ is a working day} \\ 0, & \text{otherwise.} \end{cases}$$

Finally, we introduce one resource for each experiment in order to control its temporal arrangement. According to Table 1, we have 25 experiments with a number of repetitions greater than zero. Hence, we obtain 25 resources $k = 3, \ldots, 27$ of this type with a constant availability of

$$R_k(t) = 1 \text{ for all } t \in T.$$

The idea behind these latter resources will become clear when we define the activities in the next subsection.

4.2 Activities

The basic idea of the definition of the activities of the project is to comprise those repetitions of an experiment that should be processed in parallel into one activity. Consequently, as the repetitions of an experiment should finish on 2 or 3 different days, each experiment will consist of 2 or 3 activities. An experiment related to 4 repetitions then corresponds to 2 activities, each of which consists of 2 repetitions. An experiment related to 5 repetitions corresponds to 3 activities, two of which consist of 2 repetitions while the third one consists of the remaining repetition. Table 3 defines the transformation of experiment repetitions into activities. For each number of repetitions of one experiment in the first row, the number of activities corresponding to that experiment is shown in the second row. The third row then displays the number of repetitions of each of the resulting activities. In accordance with the number of repetitions stated in Table 1, we obtain $J = 62$ activities. The processing time p_j of activity $j \in \{1, \ldots, 62\}$ is given by the duration of the related experiment. There are no precedence relations between the activities. Clearly, a dummy source and a dummy sink activity can be added.

Repetitions	*2*	*3*	*4*	*5*	*6*	*7*	*9*
activities	2	2	2	3	3	3	3
repetition jobs	1,1	2,1	2,2	2,2,1	2,2,2	3,2,2	3,3,3

Table 3: Transforming experiment repetitions into activities

Now we have to define the resource requirements of the activities. We consider an activity j which corresponds to ρ repetitions of the related experiment. The first resource which reflects the laboratory equipment is requested only on the last day of an activity:

$$r_{j1}(t) = \begin{cases} \rho, & \text{if } t = p_j \\ 0, & \text{otherwise.} \end{cases}$$

The second resource is related to the researcher. If the experiment related to activity j is based on normal food, the researcher must be in the laboratory on each day, that is, we obtain a constant request of

$$r_{j2}(t) = \rho \text{ for all } t \in \{1, \ldots, p_j\}.$$

Otherwise, if the special diet has to be considered, we have a resource requirement varying with time (recall, the duration of such a job is always 7 days):

$$r_{j2}(t) = \begin{cases} \rho, & \text{if } t \in \{1, 5, 6, 7\} \\ 0, & \text{otherwise.} \end{cases}$$

As already mentioned, the remaining resources are responsible for the temporal arrangement of the experiments. Each of these resources is related to a unique experiment and vice versa. First, we consider the resource $k > 2$ that is related to the experiment of activity j. We define the time-varying resource requirement by

$$r_{jk}(t) = \begin{cases} 1, & \text{if } t = p_j \\ 0, & \text{otherwise.} \end{cases}$$

That is, the activities of one experiment request their experiment-specific resource in their last period. As the capacity of this resource is one unit per period, this ensures that two activities of an experiment cannot finish on the same day. Finally, activity j does not request any resource $k > 2$ that is related to another experiment:

$$r_{jk}(t) = 0 \text{ for all } t \in \{1, \ldots, p_j\}.$$

Note that abstract resources in project scheduling models do not necessarily reflect traditional resources such as manpower or machines. They can also be employed to impose desired properties of a schedule. Actually, the use of resources $3, \ldots, 27$ is an example for this technique.

5 Genetic Algorithm

Introduced by Holland [9], genetic algorithms (GAs) serve as a heuristic meta strategy to solve hard optimization problems. For an introduction into GAs, we refer to Goldberg [7]. We apply the GA proposed by Hartmann [8] for the RCPSP to schedule the medical research project discussed in the previous sections. This GA approach can be summarized as follows:

- **Encoding:** An individual I is represented by an activity sequence $j_1^I, \ldots, j_J^I$. This activity sequence is assumed to be a precedence feasible permutation of the set of activities, that is, we have $\{j_1^I, \ldots, j_J^I\} = \{1, \ldots, J\}$ and $P_{j_i^I} \subseteq \{0, j_1^I, \ldots, j_{i-1}^I\}$ for $i = 1, \ldots, J$. Each genotype is related to a uniquely determined schedule (phenotype) which is computed using the following serial scheduling scheme: First, the dummy source activity is started at time 0. Then we schedule the activities in the order that is prescribed by the sequence $j_1^I, \ldots, j_J^I$. Thereby, each activity is assigned the earliest feasible start time.

- **Crossover:** We apply a two-point crossover which combines partial job sequences of the parent individuals to form children. Thereby, the relative positions of the activities in the parents' job sequences are maintained. We consider two individuals selected for crossover, a mother M and a father F. Then we draw two random integers q_1 and q_2 with $1 \leq q_1 < q_2 \leq J$. Now the daughter individual D is determined by taking the job sequence of the positions $i = 1, \ldots, q_1$ from the mother, that is,

  $$j_i^D := j_i^M.$$

  The positions $i = q_1 + 1, \ldots, q_2$ are derived from the father. However, the jobs that have already been taken from the mother may not be considered again. We obtain:

  $$j_i^D := j_k^F \text{ where } k \text{ is the lowest index such that } j_k^F \notin \{j_1^D, \ldots, j_{i-1}^D\}.$$

  The remaining positions $i = q_2 + 1, \ldots, J$ are again taken from the mother, that is,

  $$j_i^D := j_k^M \text{ where } k \text{ is the lowest index such that } j_k^M \notin \{j_1^D, \ldots, j_{i-1}^D\}.$$

  The son individual is computed analogously, taking the first and third part from the father and the second one from the mother. As shown by Hartmann [8], this crossover operator produces precedence feasible offspring if also the parents fulfill the precedence assumption.

- **Mutation:** Given a permutation based individual I, the mutation operator modifies the related job sequence as follows: For all positions $i = 1, \ldots, J-1$, activities j_i^I and j_{i+1}^I are exchanged with a probability of p_{mutation} if the resulting job sequence is precedence feasible.

- **Selection:** The selection operator makes use of a simple survival-of-the-fittest strategy. It repeatedly removes the least fit individual from the population until the original size is reached (ties are broken arbitrarily).

- **Initial population:** The initial population is determined with a sampling procedure which employs a randomized version of the well-known precedence based LFT priority rule, cf. Kolisch [12].

- **Stopping criteria:** In the implementation used here, the GA stops if either a pre-specified maximal number of schedules has been computed or if a given time limit is reached.

6 Computational Results

The following computational results were obtained on a Pentium-based computer with 133 MHz clock-pulse and 32 MB RAM. The genetic algorithm was coded in ANSI C, compiled with the GNU C compiler and tested under Linux.

6.1 Schedule for Original Data

In order to gain computational experience, we have applied the GA to the original project data and varied the maximal number of schedules to be computed. Table 4 summarizes the results. Limiting the number of schedules to 1,000, we obtain a makespan of 68 days. The best schedule with a makespan of 67 days is found within a moderate computation time of less than 12 seconds after 2,000 schedules have been evaluated. Further increasing the number of schedules does not lead to a shorter project duration. In fact, we will show in the next subsection that a makespan of 67 days is already optimal.

Schedules	*makespan*	*CPU-sec*
1,000	68	5.6
2,000	67	11.8
5,000	67	29.5
10,000	67	58.8
100,000	67	581.8
hand-made	75	—

Table 4: Makespan w.r.t. computation time

Before the medical research project was carried out, the researcher made a schedule by hand, i.e. without any computer based support. As listed in Table 4, the resulting makespan of the project as performed in 1994 was 75 days. Thus, a heuristic like our GA would not

only have made the scheduling process much easier and more convenient, it would also have determined a schedule with a makespan 10.7 % shorter than that of the hand-made schedule. Moreover, the GA would have decreased the number of working days the researcher had to spend in the laboratory by 17.4 %.

6.2 Optimality Issues

Before we examine the schedule computed by the GA for the original project data, we study the general capability of the GA to solve instances of the RCPSP/τ. We show that the search space of the GA does not necessarily contain an optimal or even feasible solution when applied to this problem class. This is because the GA employs a serial scheme which schedules each activity as early as possible.

Theorem 1 *There are instances of the RCPSP/τ for which the GA cannot find an (existing) optimal solution.*

Proof. We consider the following counterexample which consists of $J = 2$ non-dummy activities without precedence relations among them. Both activities have a processing time of $p_1 = p_2 = 2$ periods. The set of periods is $T = \{1, 2, 3, 4\}$. We have one resource, that is, $K = \{1\}$, with time-varying capacity

$$R_1(t) = \begin{cases} 2, & \text{if } t \in \{1, 2, 4\} \\ 4, & \text{if } t = 3. \end{cases}$$

The time-variying resource requirements of the activities are given by

$$r_{11}(t) = r_{21}(t) = \begin{cases} 1, & \text{if } t = 1 \\ 2, & \text{if } t = 2. \end{cases}$$

Now consider the schedules shown in Figure 1. Schedule (a) is optimal with a makespan of 3 periods. The search space of the GA consists of two individuals with activity sequences 1,2 and 2,1, respectively. The related schedules are those of Figure 1 (b) and (c), respectively. As the first activity in the sequence is always started as early as possible, i.e. at time 0, the GA can only find these suboptimal schedules with a makespan of 4 periods. □

Theorem 2 *There are instances of the RCPSP/τ for which the GA cannot find an (existing) feasible solution.*

Proof. We consider again the counterexample defined in the proof of Theorem 1. Setting $R_1(4) := 1$ instead of 2 induces that none of the two activity sequences of the search space of the GA is related to a feasible schedule as the first job in the sequence is always started at time 0. The feasible (and optimal) solution for this modified instance still is obtained from starting both activities at time 1, cf. Figure 1 (a). □

Nevertheless, the GA (or any other heuristic based on the serial scheme which schedules an activity at the earliest feasible start time) may be an appropriate approach to the RCPSP/τ. Consider as an example the so-called parallel scheduling scheme which is widely applied to the classical RCPSP although it cannot always find an optimal schedule for this problem class, cf. e.g. Kolisch [12]. In fact, we can prove that the schedule found by the GA for the original data of the medical research project is optimal:

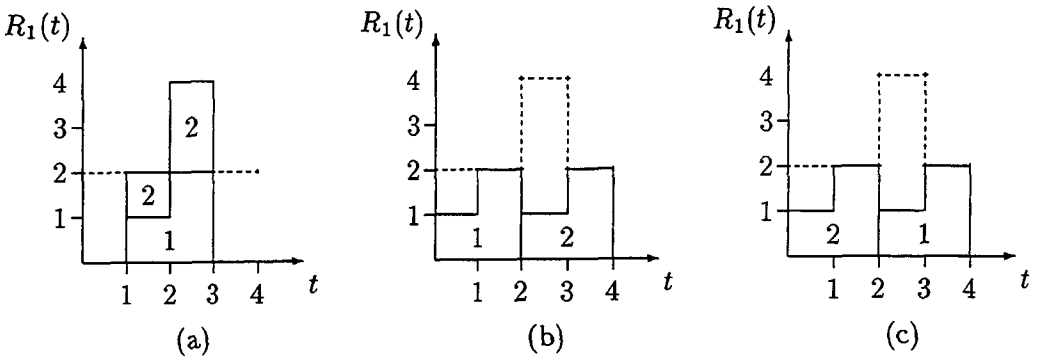

Figure 1: Schedules for the RCPSP/τ Example Instance

Theorem 3 *The optimal duration of the medical research project considered in this paper is 67 days.*

Proof. From Table 1 we know that 109 experiment repetitions have to be scheduled. Given that only 6 rats may be examined per examination day, we need at least 19 examination days which are also working days. Considering the calendar of Table 2, the earliest project completion is on day 66 by the above arguments.

We now show that there is no feasible schedule with a makespan of 66 days. The number of experiment repetitions with a duration of more than 3 days is 95, cf. again Table 1. These repetitions require at least 16 examination days which are also working days. As we consider durations of more than 3 days, however, these repetitions cannot finish on any of the days 3, 24, 45, and 59, cf. the calendar of Table 2. Consequently, the 16th examination and working day for finishing repetitions with a duration of more than 3 days is day 67. That is, 67 days is a lower bound on the project duration. As the upper bound found by the GA is also 67 days, the related schedule is optimal. □

6.3 Impact of Data Variations

Applying a computer-based scheduling system to a real-world project makes it easier to find a good schedule which fulfills all the restrictions. In addition, such a system offers the possibility of varying some of the data and rerun the scheduling procedure in order to find alternative schedules with only a low effort. In particular, this allows to answer the following questions: Can we find a schedule of equal length, if we tighten one of the constraints? Can we find a schedule with a shorter makespan, if we loosen some of the restrictions?

The first approach to finding alternative schedules is by varying the number of repetitions that can be handled by the researcher at the same time (i.e. the capacity of the researcher resource). The resulting project durations can be found in Table 5. Probably the researcher would accept a higher number of simultaneously processed repetitions if this could shorten the project. The results, however, show that this is not the case. In fact, this is not a surprise as the proof of Theorem 3 indicates that a makespan shorter than 67 days can only be achieved when changing the examination or working day contraints. Moreover, the researcher would prefer a schedule with less than 20 repetitions at the same time if this

would not lead to a longer project duration. Table 5 shows that a schedule with equal makespan can be found for only 18 simultaneous repetitions. Note that tightening this constraint makes it harder for the GA to find an optimal solution as it takes more than 10 seconds. Decreasing the maximum number of simultaneous repetitions to 16, however, increases the makespan by 6 days. Observe that the last day of the project must be an examination day, and that days 69, 70, 71, and 72 are no examination days.

Repetitions	*makespan after 10 sec*	*makespan after 60 sec*
22	67	67
20 (original)	67	67
18	68	67
16	73	73

Table 5: Varying the maximal number of repetitions in process each day

Next, we examine the impact of calendar changes concerning the working and examination days. Table 6 shows that additional vacations of the researcher result in a longer project duration. On the other hand, additionally working on one weekend does not shorten the makespan (this is again due to the fact that the examination constraints are critical). Finally, we have examined the impact of using the laboratory equipment on a Monday and/or Tuesday as well, e.g. if there is some flexibility concerning other research projects which use this laboratory. The last two rows of Table 6 indicate that this may result in a reduction of the makespan (as well as a reduction of the number of days the researcher has to be in the laboratory).

Calendar change	*makespan*
none (original)	67
no work on day 57	68
no work on days 57, 58	73
no work on days 62, 63	73
additionally work on days 6, 7	67
additionally examine on day 30	67
additionally examine on day 65	66
additionally examine on days 64, 65	65

Table 6: Impact of calendar changes — time limit: 60 sec

The last data variation to be tested affects the temporal arrangement of activities belonging to the same experiment. Up to this point, we have tried to exclude systematical errors by distributing the examinations related to one experiment on different days (as is common practice). One can argue, however, that other potential sources for errors (in addition to the examination process) occur during the total experiment duration. An example is the dosing of the medicaments. In order to avoid these errors, we have to ensure that different activities of one experiment do not overlap at all. This can easily be achieved

by adapting the request of activity $j = 1, \ldots, J$ for the resource k that is related to its experiment as follows:

$$r_{jk}(t) = 1 \text{ for all } t \in \{1, \ldots, p_j\}.$$

This adaption allows further changes of the project data which improve the behaviour of the genetic algorithm. So far, we have enforced the activities of one experiment to be performed in sequence (i.e. they do not overlap). Now consider two activities of one experiment that correspond to the same number of repetitions. As these activities are equal, the order in which they are executed is irrelevant. Consequently, we may introduce a precedence relation between these two jobs. Doing so, we reduce the search space of the GA, of course without excluding all optimal solutions. Notice that the GA exploits the added precedence relations when computing the initial population.

Finally, consider those experiments for which all (two or three) activities correspond to the same number of repetitions. As the order of these equal activities is irrelevant, we may impose an arbitrary order by adding precedence relations. In this case, we may omit the resource which is responsible for the temporal arrangement of the experiment. Reducing the number of resource constraints results in lower computation times for each schedule. Consequently, the GA can evaluate more schedules within the same time limit.

Table 7 shows that forcing the activities of one experiment to be performed in sequence (without overlapping) increases the project makespan by one week. This is still less than the duration of the project according to the original hand-made schedule, see again Table 4.

Repetition jobs of one experiment	*makespan*
overlapping allowed (original)	67
always in sequence	74

Table 7: Changing the temporal arrangement — time limit: 60 sec

7 Conclusions

In this paper, we have scheduled a real-world medical research project with a genetic algorithm that had been proposed for the classical resource-constrained project scheduling problem (RCPSP). After a description of the medical research project, we have transformed the real-world data into an instance of the RCPSP with time-varying resource availability and request. Thereby, we have shown how to use the modeling concepts (resources, activities) in order to formalize the real-world situation. Our computational experiments revealed that the genetic algorithm is capable of finding good schedules within moderate computation times of up to one minute. We were able to prove that the schedule computed for the original project data is optimal. Moreover, the makespan of the computed schedule is more than 10 % shorter than the duration of the project according to the original hand-made schedule. Finally, we demonstrated how to obtain alternative schedules for different project scenarios.

It is noteworthy that there are no precedence relations to be observed in the original medical research project. Consequently, a heuristic which exclusively makes use of precedence informations (such as many widely used priority rule based procedures) would be

doomed to failure. The genetic algorithm employed here takes all characteristics of the RCPSP into account, which makes it a widely applicable tool for scheduling real-world projects.

Acknowledgement: I would like to thank Thies Fitting for providing the relevant data of the medical research project (and for noticing that he had to deal with a difficult OR problem).

References

[1] BAAR, T., P. BRUCKER, AND S. KNUST (1997): Tabu-search algorithms for the resource-constrained project scheduling problem. Working Paper, University of Osnabrück, Germany.

[2] BRUCKER, P., S. KNUST, A. SCHOO, AND O. THIELE (1996): A branch-and-bound algorithm for the resource-constrained project scheduling problem. European Journal of Operational Research, to appear.

[3] CHO, J.H. AND Y.D. KIM (1997): A simulated annealing algorithm for resource-constrained project scheduling problems. Journal of the Operational Research Society, Vol. 48, pp. 736-744.

[4] DEMEULEMEESTER, E. AND W. HERROELEN (1992): A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. Management Science, Vol. 38, pp. 1803-1818.

[5] DEMEULEMEESTER, E. AND W. HERROELEN (1995): New benchmark results for the resource-constrained project scheduling problem. Management Science, to appear.

[6] FITTING, T. (1997): Bedeutung der intrazellulären S-Adenosylmethionindecarboxylase-Aktivität für die Regulation des Polyaminstoffwechsels während des Camostat-stimulierten Pankreaswachstums von Ratten. MD Dissertation, University of Kiel, Germany.

[7] GOLDBERG, D.E. (1989): Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading, Massachusetts.

[8] HARTMANN, S. (1997): A competitive genetic algorithm for resource-constrained project scheduling. Manuskripte aus den Instituten für Betriebswirtschaftslehre, No. 451, University of Kiel, Germany.

[9] HOLLAND, H.J. (1975): Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor.

[10] KOHLMORGEN, U., H. SCHMECK, AND K. HAASE (1996): Experiences with fine-grained parallel genetic algorithms. Working Paper, University of Karlsruhe.

[11] KOLISCH, R. (1996): Efficient priority rules for the resource-constrained project scheduling problem. Journal of Operations Management, Vol. 14, pp. 179-192.

[12] KOLISCH, R. (1996): Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. European Journal of Operational Research, Vol. 90, pp. 320-333.

[13] KOLISCH, R. AND A. DREXL (1996): Adaptive search for solving hard project scheduling problems. Naval Research Logistics, Vol. 43, pp. 23-40.

[14] KOLISCH, R., A. SPRECHER, AND A. DREXL (1995): Characterization and generation of a general class of resource-constrained project scheduling problems. Management Science, Vol. 41, pp. 1693-1703.

[15] LEE, J.-K. AND Y.-D. KIM (1996): Search heuristics for resource-constrained project scheduling. Journal of the Operational Research Society, Vol. 47, pp. 678-689.

[16] LÖSER, C., U.R. FÖLSCH, C. PAPROTNY, AND W. CREUTZFELD (1990): Polyamines in colorectal cancer: Evaluation of Polyamine concentrations in colon tissue, serum, and urine of 50 patients with colorectal cancer. Cancer, Vol. 65, pp. 958-966.

[17] LÖSER, C., T. FITTING, AND U.R. FÖLSCH (1997): Importance of intracellular S-Adenosylmethionine Decarboxylase activity for the regulation of Camostate-induced pancreatic Polyamine metabolism and growth: In vivo effect of two novel S-Adenosylmethionine Decarboxylase inhibitors. Digestion, Vol. 58, pp. 258-265.

[18] MARTON, L.J. AND A.E. PEGG (1995): Polyamines as targets for therapeutic intervention. Annu. Rev. Pharmacol. Toxicol., Vol. 35, pp. 55-91.

[19] MINGOZZI, A., V. MANIEZZO, S. RICCIARDELLI, AND L. BIANCO (1995): An exact algorithm for project scheduling with resource constraints based on a new mathematical formulation. Management Science, to appear.

[20] ÖZDAMAR, L. (1996): A genetic algorithm approach to a general category project scheduling problem. Research Report, Marmara University, Istanbul.

[21] SPRECHER, A. (1994): Resource-constrained project scheduling: Exact methods for the multi-mode case. Lecture Notes in Economics and Mathematical Systems, Vol. 409, Springer, Berlin et al.

[22] SPRECHER, A. (1996): Solving the RCPSP efficiently at modest memory requirements. Manuskripte aus den Instituten für Betriebswirtschaftslehre, No. 425, University of Kiel, Germany.