

Scheduling with sequence dependent setup times in application to Magnetic Resonance Imaging scans processing

E.N. Ivanov, A.Yu. Pogromsky, J.E. Rooda

Abstract—We describe an algorithm for solving a single machine scheduling problem with additional assumptions motivated by Magnetic Resonance Imaging (MRI) Systems. In the operating model, the setup times are sequence dependent and the objective is to minimize the total production time. An application of the algorithm to the real MRI examination results in an increase of the scanner workflow.

I. INTRODUCTION

A single machine scheduling problem with setup times is a classical problem that has been well covered in literature [1], [2], [3]. In this paper the theory has been adapted for a special problem related to MRI systems, therefore some additional assumptions are made and the standard model has been modified.

Scheduling problems with setup times can be divided into sequence-independent and sequence-dependent classes. The setup time is sequence-dependent if it depends on both the current and the next job in the processing sequence. The sequence-independent setup time depends only on the current job to be processed [2].

The objective function of our problem is to minimize the total production time (makespan) that, according to [4], together with maximum tardiness are the most prominent measures among the scheduling objectives in industrial applications.

For the standard scheduling problem of minimizing total production time with sequence-dependent setup times exact solution methods were developed, based on the Traveling Salesman Problem [5], [6], [7]. However the exact solutions are numerically expensive, and different heuristics are used in common practice [4], [8].

In this paper the classical scheduling model is extended to be applicable to MRI scans scheduling problem. Special exact and heuristic algorithms are obtained and applied to the real examination protocols of Philips Healthcare MRI scanners.

The outline of the paper is follows. In Section II some basic information on MRI scanner systems is presented that is used throughout this paper. In Section III the formal scheduling problem is stated. Finally, in Section V the algorithms are applied to the real examination protocols

This work has been carried out as a part of the DARWIN project at Philips Healthcare under the responsibilities of the Embedded Systems Institute (ESI). This project is partially supported by the Dutch Ministry of Economic Affairs under the BSIK program.

All authors are with the systems Engineering Group, Department of Mechanical Engineering, Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands [E.Ivanov, A.Pogromsky, J.E.Rooda]@tue.nl

of MRI systems to calculate the total time reduction per examination.

II. BASIC MRI THEORY

In this section, following [9], some basic facts about MRI systems are introduced that will be used in the remainder of this paper. These facts are relevant to state the scheduling problem for MRI systems.

A. Basic principles of MR Imaging

Magnetic Resonance (MR) image represents the relative response of specific nuclei to absorbed radio frequency (RF) energy. The history of MRI starts from the middle of 20th century. First human MR images were obtained by Sir Peter Mansfield in 1976. Nowadays MRI has become an essential part in clinical diagnostic imaging. Modern MRI Scanners can produce soft tissue images with different contrasts and with a high spatial resolution. MRI offers the unique ability to acquire 2D and 3D images in virtually any orientation, without repositioning the patient.

During the examination the patient is placed in the bore of the MRI scanner. The bore contains a magnet to generate static magnetic field and magnetic gradient coils. The gradient coils together with gradient amplifiers are the parts of the magnetic field gradient system. This system is required to encode spatial information on the nuclei within a tissue sample by local variations of magnetic field. If an RF pulse is then applied through RF coils placed on the patient, the proton nuclei are excited and brought into phase. When the applied pulse is terminated, an MR signal is detected by the RF receiver system and transported to a computer system for digital processing and image visualization. The RF transmit and the RF receiver systems consist both of RF coils and RF amplifiers.

To use images for clinical diagnostics, there must be contrast between the MR signal of different tissue types. The intensity of the magnetic resonance signals is a function of several parameters. The relative contribution of each parameter is controlled by adjusting the RF pulses, the applied gradients and the timing of the data acquisition. These parameters are set in the Imaging Sequence, also called the Pulse Sequence. The completion of an image requires repeated measurements of the signals, thus one Imaging Sequence is repeated hundreds of times. These repetitions take place once every repetition time (TR). In ‘spin echo’ imaging the repetition time (TR) varies between a few hundred milliseconds and a few seconds.

An MRI examination consists of different types of scans. Typically there are about 5-10 different scans in one examination. Every scan (scanning protocol) consists of hundreds of repetitions of one Imaging Sequence.

To provide a clear picture of morphology images are often acquired for several anatomic ‘slices’. Multiple excitations are often required for each image slice to obtain a satisfactory signal-to-noise ratio. If the acquisitions are performed sequentially (multiple repetitions for each slice are completed before moving to the next slice) then there will be a long period of inactivity (‘dead time’) between the signal and successive excitation pulses. This inefficiency results from the requirement for a relatively long (typically 0.5 - 2 seconds) repetition time (TR) to allow the spins to relax sufficiently between the pulses (see Fig. 1). In this figure the thin tall bars are the RF-pulses, the thick bars are the gradient changes.

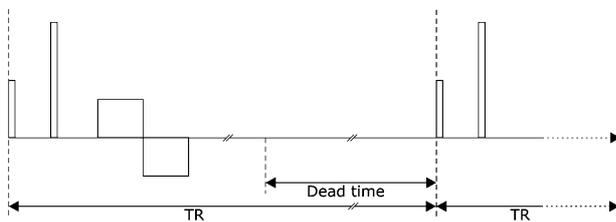


Fig. 1. Imaging sequences

Some special techniques are used to speed up the image acquisition. These are multi-echo and multi-slice acquisitions. The multi-echo acquisition protocols operate by applying a series of 180 degree RF pulses during one TR. The multi-slice image acquisition operates by acquiring data from the other image slices during the ‘dead time’. They both use multiple RF pulses within each repetition. Also there are some combinations and variations of this strategies that help to enhance the scanning duty-cycle.

A disadvantage of these strategies is that it causes high Specific Absorption Rate (SAR) levels during the scan execution. There are also other scanning strategies, that can cause overheating of the gradient amplifier or coils. All these are the duty-cycle limitations of the scanning process.

B. Specific Absorption Rate

One of the main constraints during the scan execution is the limitations of the ‘Specific Absorption Rate’ level (SAR). From a certain point of view, the electromagnetic effects in the bore of MRI-scanner are the same as in a microwave oven. The RF pulses during the scan raise the radiation level that can result in heating the patient body. The SAR indicates how much RF power is being dissipated in the patient body and it is measured in Watts per kilogram [W/kg]. To avoid harmful effects on the patient health the International Electrotechnical Commission (IEC) adopted limits for a safe exposure of the RF energy produced by MRI scanner during the examination. The SAR is calculated as a square of the RF field averaged over the duration of the scan with a factor C which depends on a coil. A scan consists of number

of identical Imaging Sequences. So, the SAR can also be calculated as a square of the RF field averaged over the duration of one Imaging Sequence (TR) with a factor C :

$$SAR = C \cdot B = C \cdot \frac{1}{TR} \int_{TR} b_1^2(t) dt, \quad (1)$$

where b_1 is the RF field.

An example of the RF field strength during a TR is represented in Fig. 2:

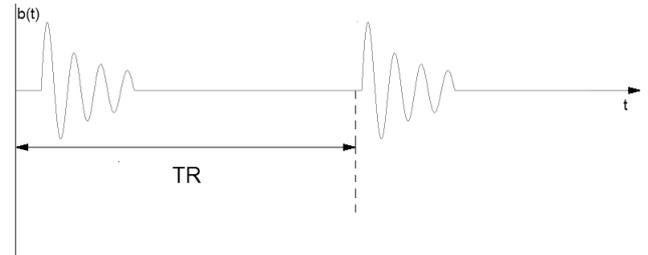


Fig. 2. RF Field

According to the IEC’s classification, there are several types of SAR: the whole body SAR, the head SAR, the local torso SAR and the local extremities SAR. This is in recognition of the fact that RF power dissipation varies across the body for the different types of SAR. Therefore the IEC prescribes different long-term limits for these types of scan:

- head SAR (< 3.2 [W/kg])
- whole body SAR (< 4 [W/kg])
- local torso SAR (< 10 [W/kg])
- local extremities SAR (< 20 [W/kg])

For each particular type of the scan there is only one IEC SAR limit. Those IEC limits are long-term, this means that they bound a maximum average SAR level for the whole duration of scan. On the other hand the IEC also declares short-term SAR limits that the SAR over any 10 seconds period shall not exceed 3 times of its long-term limit.

Some scans are SAR-limited. The Imaging Sequences of such scans include some amount of not-used time (‘dead time’) in order not to exceed the SAR constraints. The modern generation of MR-scanners use 3 Tesla magnets instead of 1.5 Tesla ones, and it results in rising the number of SAR limited scans. Moreover the future generation of MR-scanners will use 7 Tesla magnets and the number of SAR limited scans is expected to increase accordingly.

C. Scan Segments Mixing

The SAR level, temperature of gradient amplifiers, gradient and RF coils and other duty-cycle limitations of the examination can be called resources. The scan is called resource limited if it includes ‘dead time’ when the system just waits for the expended resource to become available again. During this ‘dead time’ the SAR limited scans wait until the SAR decreases, and similarly the gradient-limited scans wait until the gradient amplifiers temperature decreases. In

some scanning protocols the dead time is up to the half of the examination time that gives a big gap for future optimization.

Different types of scans impose restrictions on different types of resources. The scans that are SAR-limited do not pose severe requirements on the gradient system. On the other hand, the diffusion scans are often gradient duty cycle limited, but pose no SAR constraints.

During the examination the scans can be divided into segments and then mixed in such a way that the ‘dead time’ of one scan would be exploited for the useful job of another one. The segments in which the scans can be divided are the sequences of ‘Imaging Sequences’. In most cases such sequences can be separated in time without much penalty.

Switchings between segments of different scans are not immediate. It takes some time, which is called ‘setup time’. In average the ‘setup time’ is from 1 to 1.5 second. The setup time is much less than the dead time in the scans, thus the time reduction still can be gained.

To increase the scanner workflow the ‘dead time’ should be utilized for a useful job. Therefore the main problem is to find the best algorithm of dividing scans into segments and mixing them with each other to decrease the examination time.

III. PROBLEM STATEMENT

For the formal problem statement the scan segments from here on are called jobs and different scans are called job families.

Let us consider a number of jobs $j = 1, 2, \dots, N$ that belong to F different job families. If the machine (the MRI scanner) switches over from one family to another, e.g. from family g to family h , then a setup time is required. The setup time is sequence dependent and is denoted by $s_{gh} \geq 0$.

Let n_g denote the number of jobs from family g (the number of scan segments in the scan g).

$$N = n_1 + n_2 + \dots + n_F. \quad (2)$$

Let (g, j) refer to job j from family g , $j = 1, \dots, n_g$; it has a processing time $p_{(gj)} = p_g$ identical for all jobs in the family g for all $g = 1 \dots F$. The completion time of job j in the schedule is denoted by C_j .

The objective is to minimize the total production time (makespan) C_{\max} of the jobs in the sequence:

$$C_{\max} = \max(C_1, \dots, C_N) \longrightarrow \min. \quad (3)$$

Our extension to the standard scheduling theory with family dependent setup times ($1|fmls, s_{gh}|C_{\max}$) is introduced below. The main difference is the waiting times within job families that depend on the ‘resources’. This extension was invented to deal with constrains (e.g. SAR level) during the scans execution. The SAR raises during execution of each scan segment, and it needs to be decreased either by adding ‘dead time’ after the segment or by switching to the scans with lower SAR levels.

Each job family g can depend on a resource R_i , $i \in \{1, \dots, m\}$, $m \leq F$ (i.e. the jobs (g, j) , $j \in 1, \dots, n_g$ from the family g can be started if

the resource R_i is available). Different job families can depend on identical resources. This dependence is denoted by $R(g) = R_i$ if job family g depends on the resource R_i , or by $R(g) = \emptyset$ if job family g does not depend on any resources.

If the job family g depends on resource $R(g) = R_i$, then after each job within the family a waiting time $\tau_g > 0$ (‘dead time’) is needed to restore the resource (if $R(g) = \emptyset$ then $\tau_g = 0$). During the waiting time τ_g the machine can switch over to another job family h if the family does not depend on the same resource, and the resource $R(h)$ is available. When such a switching happens, the machine should process h family jobs for at least τ_{gh} time to restore the resource $R(g)$ of the previous job family g .

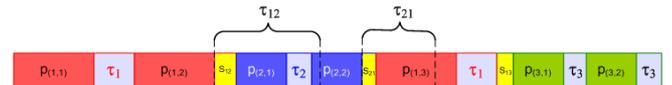


Fig. 3. Example with three job families, $R(1) = R(3) = R_1$, $R(2) = R_2$.

To explain the notation, an example with three job families is presented in Fig. 3. Job families 1 and 3 depend on the same resource $R(1) = R(3) = R_1$ and job family 2 depends on another resource $R(2) = R_2$. Thereby during processing the job family 1 instead of waiting τ_1 the machine switches to the job family 2 and process it for at least τ_{12} time (two jobs here). Afterwards it switches back to the job family 1 and processes it for at least τ_{21} time. Finally the machine switches to the job family 3 which depends on the same resource R_1 , thus the waiting time τ_1 cannot be replaced by processing the 3rd family jobs, and both τ_1 and s_{13} appear.

In general, the problem is close to the problem of scheduling a single machine having sequence dependent setup times where the objective is to minimize the total production time (makespan) for the set of released orders. In this case, the problem could be represented as a Traveling Salesman problem, [5], [6]. However the Traveling Salesman problem is known to be NP-hard and the algorithms that give an exact solution are numerically expensive. Fortunately, our problem has some natural restrictions and an optimization algorithm can be much simpler.

IV. AN ALGORITHM FOR SOLVING THE SCHEDULING PROBLEM

According to Section III, that describes the formal problem statement of the scheduling problem, the objective (3) is to minimize the total production time C_{\max} of the jobs in the sequence.

In this section an algorithm is described, that, for the given set of jobs $j = 1, \dots, N$ (that belong to $F \geq 2$ different job families) and the corresponding set of resources R_1, \dots, R_m , $1 \leq m \leq F$, generates the sequence σ^* with minimal C_{\max} :

$$\sigma^* = \arg \min_{\sigma \in \{\text{All sequences of the jobs}\}} C_{\max}. \quad (4)$$

All the parameters $p_a, p_b, \tau_a, \tau_b, n_a, n_b, s_{ab}, s_{ba}, \tau_{ab}, \tau_{ba}$ (that were defined in the previous section) are given for all pairs $a \neq b$ of job families $a, b \in \{1, \dots, F\}$. Furthermore, additional assumptions are imposed.

A. Assumptions

The following assumptions will slightly simplify the problem. They are based on natural hardware restrictions of the MRI machines.

Assumption 1: Jobs within each family are processed in fixed order, according to increasing of their numbers in the family. Technically, for jobs in each job family a precedence constraints are held in a chain form. Each job (f, j) , $f \in 1, \dots, F$, $j \in 1, \dots, n_f$ has one predecessor and one successor (except the first job $(f, 1)$ and the last job (f, n_f) in the family that has no predecessor and no successor, respectively). The jobs in the chain are sorted according to their numbers in the family:

$$(f, 1) \prec (f, 2) \prec \dots \prec (f, n_f), \quad f = 1 \dots F \quad (5)$$

in increasing order.

With this assumption w.l.o.g. we can consider that our sequences σ have a special form $\sigma = (f_1, \dots, f_N)$, $f_i \in \{1, \dots, F\}$, where each job is represented by it's family.

Assumption 2: No more than two job families can be intermixed simultaneously. Technically, for each $k \leq N$ the number of different job families in the head subsequence $\{f_1, \dots, f_k\} \subset \sigma$ that have started processing, but have not processed yet, does not exceed two:

$$\# \left\{ g \left| 0 < \sum_{i=1}^k \chi_g(f_i) < n_g, g = 1 \dots F \right. \right\} \leq 2. \quad (6)$$

Here $\chi_g \rightarrow \{0, 1\}$ is the characteristic function of subset of the jobs that belong to the family g .

Last assumption is often satisfied for the MRI applications, where the hardware can manage with no more then two types of scans simultaneously.

B. The Algorithm

According to the assumption 2, the algorithm that solves our problem (3) should work with pairs of the job families. On general level, the sequence should be specified in which the pairs of the job families will be processed. And for each particular pair the optimal intermixing should be calculated.

After intermixing of a pair of job families $a, b \in \{1 \dots F\}$ typically several jobs from one of the families will be left over. And they can be intermixed with the next job family c . Let n_g^* denote a number of jobs from family g that still needs to be processed. When the machine starts working $n_g^* = n_g, g = 1 \dots F$. Afterwards the n_g^* decreases while processing the jobs from the families.

Our scheduling algorithm consists of two parts. Firstly, a sequence π in which the pairs of the job families will be processed is selected. Secondly, the jobs in the pairs are intermixed to minimize the C_{max} of each pair in the sequence π .

1) *Selection of the sequence of the job families:* The selection of the sequence π can be done either in brute-force way or by heuristical greedy algorithm. In the brute-force case all the permutations $\pi = (1, \dots, F)$ are tied out, thus the number of sequences is

$$\#\{\pi\} = F! \quad (7)$$

the factorial of the number of job families; the complexity is $O(F!)$.

The greedy algorithm on each step searches for a pair of job families which after intermixing will gain maximal time reduction $T(g, h)$, $g, h \in \{1, \dots, F\}$.

- 1) First step all pairs of the families are tried to find the most 'gainful' one ($T(\cdot, \cdot) \rightarrow \max$); the number of pairs is $\binom{F}{2}$ - the number of 2-element subsets of an F -element set, where $\binom{\cdot}{\cdot}$ states for binomial coefficients.
- 2) Next step the remaining job families are tried to find one, that will gain maximal time reduction $T(g, \cdot)$ after intermixing with n_g^* residual jobs of the previous family.
- 3) Algorithm finishes after $F - 2$ repetitions of step 2), when all F families are processed.

The total number of job family sequences is:

$$\begin{aligned} \#\{\pi\} &= \binom{F}{2} + \binom{F-2}{1} + \binom{F-3}{1} + \dots + \binom{1}{1} = \\ &= \binom{F}{2} + F - 2 + \dots + 1 = \frac{F!}{2!(F-2)!} + (F-2+1) \frac{F-2}{2} = \\ &= \frac{F(F-1)}{2} + \frac{(F-1)(F-2)}{2} = (F-1)^2, \quad (8) \end{aligned}$$

and the complexity is $O(F^2)$.

For a small number of job families $F \leq 5$ both algorithms are comparable.

2) *Intermixing of a pair of job families:* For a given pair of job families (g, h) and the last job l in the previous pair (for the first pair $l = 0$), the optimal sequence $\phi(g, h)$ will be designed in such a form:

$$\begin{aligned} (l, \phi(g, h)) &= (l, \underbrace{g, h, h, h}_{k^* \leq k_g + k_h}, \underbrace{g, g}_{k_h}, \underbrace{h, h, h}_{k_h}, \underbrace{g, g}_{k_g}, \underbrace{h, h, h}_{k_h}, \dots \\ &\dots, \underbrace{g, g}_{k_g}, \underbrace{h, h, h}_{k_h}, \underbrace{g, g}_{k_g}, \underbrace{h, h, h}_{k_h}) \end{aligned} \quad (9)$$

where

$$\begin{aligned} k_g &= \arg \min_{k \in N} (s_{hg} + k(p_g + \tau_g) - \tau_{hg} > 0) \\ k_h &= \arg \min_{k \in N} (s_{gh} + k(p_h + \tau_h) - \tau_{gh} > 0) \end{aligned} \quad (10)$$

are the minimal numbers of jobs to restore the resources $R(h)$ and $R(g)$, respectively. Following inequalities should

hold:

$$k_g < n_g^*, \quad k_h < n_h^* \quad (11)$$

$$\tau_g > 0 \quad \text{or} \quad \tau_h > 0 \quad (12)$$

$$\tau_g + \tau_h - (s_{gh} + s_{hg}) > 0, \quad (13)$$

otherwise the job families cannot be intermixed and must be processed sequentially.

The inequality (13) guarantees that there will be a time reduction on each switching from one family to another and back. The obvious strategy for designing the $\phi(g, h)$ is to maximize the number of such switchings. In our case we should switch as soon as the resource of the previous job family is restored.

If the inequalities (11) - (13) are held then intermixing happens and one of the job families (g or h) will be completely processed. Let consider that it is family g , i.e.

$$\left\lfloor \frac{n_g^*}{k_g} \right\rfloor \leq \left\lfloor \frac{n_h^*}{k_h} \right\rfloor, \quad (14)$$

where $\lfloor \cdot \rfloor$ is the integer part, and the following equality holds for $\phi(g, h) = (\phi_1, \dots, \phi_n)$:

$$\sum_{i=1}^n \chi_g(\phi_i) = n_g^*. \quad (15)$$

The sequence $\phi(g, h)$ consists of $\left\lfloor \frac{n_g^*}{k_g} \right\rfloor \leq M_{gh} \leq \left\lfloor \frac{n_g^*}{k_g} \right\rfloor + 1$ switchings from family g to h and of $\left\lfloor \frac{n_g^*}{k_g} \right\rfloor - 1 \leq M_{hg} \leq \left\lfloor \frac{n_g^*}{k_g} \right\rfloor$ switchings from family h to g . The time reduction $T(g, h)$ on these switchings is

$$\begin{aligned} T(g, h) &= M_{gh}(\tau_g - s_{gh}) + M_{hg}(\tau_h - s_{hg}) = \\ &= M_{hg} \underbrace{(\tau_g + \tau_h - (s_{gh} + s_{hg}))}_{>0} + \underbrace{(M_{gh} - M_{hg})}_{0 \leq \leq 2} (\tau_g - s_{gh}). \end{aligned} \quad (16)$$

The first $k^* \leq k_g + k_h$ jobs in the switching sequence $\phi(g, h)$ (see (9)) are selected by brute-force from the rest $a_g = n_g^* - k_g \left\lfloor \frac{n_g^*}{k_g} \right\rfloor$ jobs of family g and a_h , ($0 \leq a_h \leq k_h$) available jobs of family h with the objective to maximize $T(g, h)$. During this brute-force the setup times s_{lg}, s_{lh} and τ_l are taken into account.

The designed $\phi(g, h)$ gains maximal time reduction for intermixing of the pair of the job families (g, h):

$$T(g, h) \rightarrow \max.$$

The new number of jobs that needs to be processes in the further intermixing is specified:

$$n_h^* := n_h - (k_h \left\lfloor \frac{n_g^*}{k_g} \right\rfloor + a_h) \quad (17)$$

$$n_g^* := 0$$

and

$$l := h. \quad (18)$$

The estimation of the algorithm time follows. The brute-force for the first k^* jobs takes:

$$\begin{aligned} &\binom{k_g + k_h}{k_g} + \binom{k_g + k_h - 1}{k_g} + \dots + \binom{k_g}{k_g} = \\ &= \frac{k_h + 1}{k_g + 1} \binom{k_g + k_h + 1}{k_g} \sim \left[\left\{ \begin{array}{l} k_h \sim \frac{k^*}{2} \\ k_g \sim \frac{k^*}{2} \end{array} \right\} \right] \sim \\ &\sim \binom{k^* + 1}{k^*/2} = (k^* + 1) \mathbb{C}_{\frac{k^*}{2}} \sim O\left(\frac{2^{k^*}}{\sqrt{k^*}}\right), \end{aligned}$$

where $k^* = k_g + k_h$ is a minimal number of jobs in both families to restore the resources of each other; and \mathbb{C}_n is n^{th} Catalan number. The calculation of the rest switching sequence takes:

$$O\left(\left\lfloor \frac{n_g^*}{k_g} \right\rfloor\right) \sim O\left(\frac{N}{k^* F}\right). \quad (19)$$

The overall complexity of the algorithm for mixing a pair of jobs is:

$$O\left(\frac{2^{k^*}}{\sqrt{k^*}} + \frac{n}{k^*}\right), \quad (20)$$

where $n = \frac{N}{F}$ is an average number of jobs in a family and k^* is order of magnitudes smaller then n .

Finally, both algorithms are designed:

- The general one, that specifies the sequence π in which the pairs of the job families will be processed
- The local one, that intermixes each pair in the sequence in an optimal way.

The resulting sequence σ^* in which the jobs in the schedule are processed follows:

$$\sigma^* = \phi(\pi_1, \pi_2), \phi(\pi_{r_1}, \pi_3), \phi(\pi_{r_2}, \pi_4), \dots, \phi(\pi_{r_{F-1}}, \pi_F), \quad (21)$$

where π_{r_i} , $i = 1, \dots, F - 1$ is the job family that remains underprocessed after intermixing of the i^{th} pair of job families. The sequence σ^* satisfies all the assumptions and guarantees the minimal C_{max} .

V. APPLICATION TO MRI SYSTEMS

The scheduling algorithm derived in Section IV can be utilized to schedule the scan segments during the MRI examinations to increase the workflow by reducing the 'dead time' in resource-limited scans.

In Section II it is mentioned that the different types of scans impose restrictions on different types of resources (SAR-limited scans are not gradient-limited and vice-versa). The segments of different scans (Imaging sequences) are denoted as jobs (g, j) , $j = 1, \dots, n_g$, $g = 1, \dots, F$, that belong to F different families, where F is the number of scans in the MRI examination. All the segments of each specific scan are identical. The 'dead time' in each segment (g, j) is denoted as τ_g . The processing time (excluding 'dead time') is denoted by p_g .

While performing an MRI examination, the MRI system can switch during the 'dead time' from the segment of one scan to the segment of another one, if all the necessary resources are available to start processing the next scan. But

switching time s_{gh} between the segments of different scans is not zero. In average it takes from 1 to 1.5 seconds.

In some cases the time required to restore the resource $R(g), g = 1, \dots, F$ depends on the type of the scan h that is executed during the waiting time τ_g of the g scan, and it is denoted as τ_{gh} . For example, if during the waiting time of a SAR-limited scan g the low-SAR scan segment h is performed, then the SAR restores quicker in contrast to performing a segment of medium-SAR scan f :

$$\tau_{gh} < \tau_{gf}, \quad f, g, h \in \{1, \dots, F\}$$

Therefore our model describes the scan segments of MRI examinations and possible switching strategies. Distributing such parts of the scans during the examination does not affect the image quality. Also it does not cause any harm to the health or any uncomfortable feelings for the patient.

The greedy modification of the algorithm has been applied to the real examination protocols of MRI examinations of Philips Healthcare. The resource was the SAR. The relative examination time reduction due to the algorithm was calculated for different examination protocols ('ExamCards'). The results for 3.0 Tesla examination protocols are presented on Fig. 4.

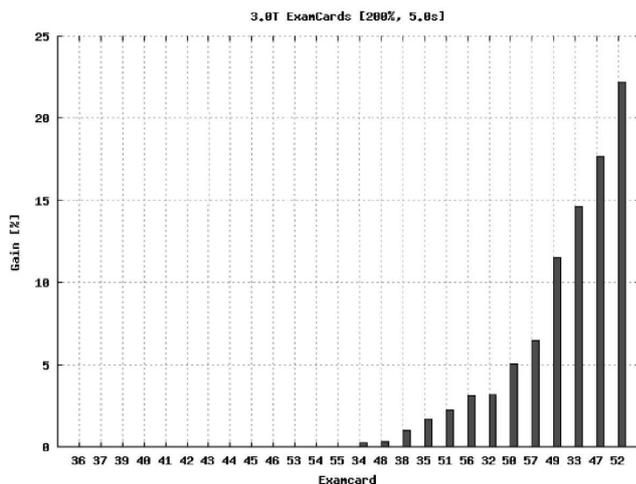


Fig. 4. Relative examination time reduction in % on real examination protocols

The sensitivity analysis was performed with respect to the total relative amount of resource SAR "maxSarLevel" (proportional to the τ_g and τ_{gh}) and the processing time "maxSarDuration" (equal to p_g) for each specific examination protocol. The sensitivity plot for one examination protocol is presented on Fig. 5.

The application of the algorithm results in time reduction up to 22% of the total examination time.

VI. CONCLUSIONS

This paper studies a new method of the scan segments scheduling algorithm for MRI Systems. An extension of the classical scheduling model with sequence dependent setup

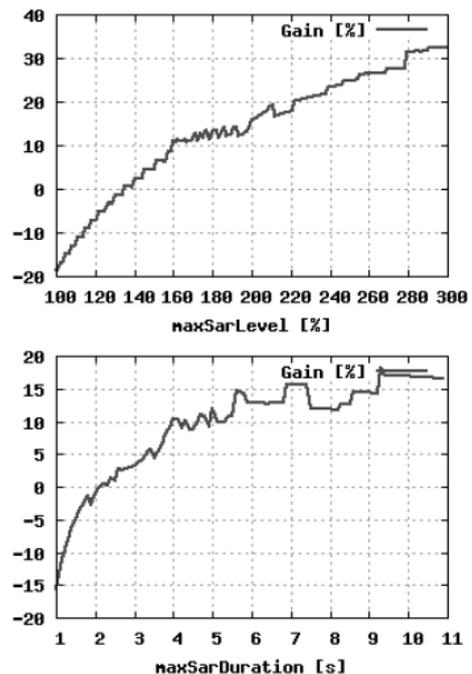


Fig. 5. Sensitivity plot for the relative examination time reduction in % on a particular examination protocol

times is used with additional assumptions related to MRI Systems. Two modifications of the algorithm are described. Preliminary results on the application of the algorithm are promising for the clinical implementation.

In future work the scheduling algorithm to solve the multivariate resource-management problem will be applied to maximum possible number of examination protocols to get more precise statistics for the average time reduction. Also the algorithm is to be extended to the case of dynamical resources.

REFERENCES

- [1] A. Allahverdi, C. T. Ng, T. C. E. Cheng, and M. Y. Kovalyov, "A survey of scheduling problems with setup times or costs," *European Journal of Operational Research*, vol. 187, pp. 985–1032, 2008.
- [2] A. Allahverdi, J. N. D. Gupta, and T. Aldowaisan, "A review of scheduling research involving setup considerations," *Omega*, vol. 27, pp. 219–239, 1999.
- [3] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 3rd ed. Berlin: Springer, 2008.
- [4] S. S. Panwalker and W. Iskander, "A survey of scheduling rule," *Operations Research*, vol. 25, pp. 45–61, 1977.
- [5] L. Bianco, A. Mingozzi, and S. Ricciardelli, "The travelling salesman problem with cumulative costs," *Networks*, vol. 23, pp. 81–91, 1993.
- [6] M. Fischetti, G. Laporte, and S. Martello, "The delivery man problem and cumulative matroids, operations research," *Operations Research*, vol. 41, pp. 1055–1064, 1993.
- [7] G. Stecco, J.-F. Cordeau, and E. Moretti, "A branch-and-cut algorithm for a production scheduling problem with sequence-dependent and time-dependent setup times," *Computers and Operations Research*, vol. 35, pp. 2635–2655, 2008.
- [8] C. Malandraki and R. B. Dial, "A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem," *European Journal of Operational Research*, vol. 90, pp. 45–55, 1996.
- [9] PMS, *Basic principles of MR Imaging*. NL: Philips Medical Systems, 2005.