Optimization of duty cycle of Magnetic Resonance Imaging scanners

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

Systems Engineering Group,
Department of Mechanical Engineering
Eindhoven University of Technology
PO Box 513, 5600 MB Eindhoven, The Netherlands
Email: [E.Ivanov, A.Pogromsky, J.E.Rooda]@tue.nl

J.S. Brink

Philips Healthcare
Advanced Concepts Development, BU MR
P.O. Box 10000; 5680 DA BEST; The Netherlands
Email: Johan.van.den.Brink@philips.com

***Abstract*—We describe algorithms for solving specific single machine scheduling problems with additional assumptions motivated by Magnetic Resonance Imaging (MRI) Systems. In the operating model, the temperature dynamics and Radio Frequency field constraints are involved and the objective is to minimize the MRI examination time. The exact and heuristic algorithms are described for different modifications of the problem.**

I. INTRODUCTION

Nowadays Magnetic Resonance Imaging (MRI) has become an essential part in clinical diagnostic imaging. MR image represents the relative response of specific nuclei to absorbed radio frequency (RF) energy. Modern MRI Scanners can produce soft tissue images with different contrasts and with a high spatial resolution.

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. Then a number of Radio Frequency (RF) pulses is then applied through RF coils placed on the patient, the proton nuclei are exited 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.

During the scans processing the amount of the RF pulses and the gradient coils and amplifiers temperature can be the limiting factors. The number of the RF pulses influences the Specific Absorption Rate (SAR) level, that depends on the Radio Frequency field strength. The temperature of amplifies, in turn, depends on the amplitude and frequency of the magnetic gradient waveforms.

The typical MRI examination consists of 5-10 scans that consist of hundreds of segments with different RF and magnetic gradient parameters. During the examination the default order of the scan segments can be changed that can decrease the temperature of the amplifiers and coils and reduce the SAR level. Hence, the problem of duty-cycle optimization can be reformulated in the single machine scheduling problem framework.

Different aspects of the single machine scheduling problems are well covered in literature, see, e.g. [1], [2], [3]. In this paper the theory is adapted for the problem related to MRI systems, therefore some additional assumptions are made and the standard model has been modified.

Special exact and heuristic algorithms has been designed separately for the SAR and temperature limitations. These algorithms are described in papers [4] and [5]. The algorithms were applied to the real examination protocols of Philips Healthcare MRI scanners to calculate the MRI examination time reductions. In the current paper both algorithms are combined.

The outline of the paper is as follows. In Section II the SAR and temperature limitation are described, that are relevant to state the scheduling problem for MRI systems. In Section III the formal scheduling problem is stated separately for both limitations. In section IV the algorithms that solves different modifications of the problem are described.

II. PHYSICAL FACTORS THAT INFLUENCE THE DUTY CYCLE OF THE MRI SCANNERS

An MRI examination consists of different types of scans, and every scan (scanning protocol) consists of hundreds of repetitions of one Imaging Sequence. Some types of scans can overheat the amplifiers or coils, while others can exceed the SAR level limitations. To prevent this harmful behavior an amount of ‘dead time’ is included into every repetition time (TR) to keep the temperature of the amplifiers and the SAR level below the limits. The main constraints that influence the duty cycle of the MRI scanners are described below.

A. *Specific Absorption Rate*

One of the main constraints during the scan execution is the limitations on 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 emitted by MRI scanner during the examination. A scan consists of number of identical Imaging Sequences. So, the SAR can be calculated as a square of the RF field (b_1) averaged over the duration of one Imaging Sequence (TR) with a factor C (depends on a coil):

$$SAR = C \cdot B = C \cdot \frac{1}{TR} \int\limits_{TR} b_1^2(t)dt. \tag{1}$$

The IEC limitations of SAR (IEC 60601-2-33, 2nd edition) are long-term, this means that they bound a maximum average SAR level for the whole duration of scan (4 [W/kg] for the whole body scans). 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 (12 [W/kg]).

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.

B. Gradient Amplifiers Temperature

During execution of the MRI examination the temperature T of the gradient amplifiers must not exceed the limit

$$T \leq T_{\max}.$$

The temperature depends on the input power $P(t)$, the dependance is expressed by the following equation:

$$T(t) = \mathrm{e}^{-t/\tau}\ T(0) + \frac{1}{\tau} \int\limits_0^t \mathrm{e}^{\frac{-(t-t')}{\tau}}\ \theta P(t')\ \mathrm{d}\ t', \tag{2}$$

where θ is heat resistance, and τ is time constant.

The input power $P(t)$ depends quadratically on the input circuit $I(t)$ that is scaled gradient waveforms of the scans $G(t)$. Each type of scan, has unique gradient waveform $G(t)$ that is periodic trapezoidlike function, where the period is one repetition time (TR). The function $G(t)$ represents the strength and direction of applied magnetic fields of the gradient coils (these fields are additional to the main powerful magnetic field of the scanner, and they are to make the magnetic field nonhomogeneous, to obtain the spatial information during the MRI scan processing).

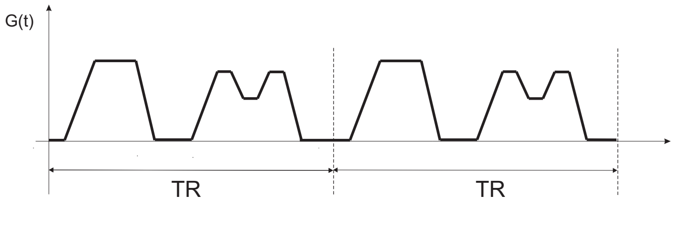

Fig. 1. Gradient waveforms

C. Scan Segments Mixing

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 limitations. 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 1 to 1.5 second. The setup time is much less then the dead time in the scans, thus the time reduction still can be gained.

The segments size depend on the limitation type. The time constants for the SAR limitation is in order of magnitudes smaller than for the gradient amplifier temperature limitation (seconds against minutes), thus the scheduling algorithms for these two types of limitations were designed separately.

III. PROBLEM STATEMENT

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

There are $N < \infty$ jobs $j = 1, \dots, N$ that belong to F different job families (scans). Let n_i denote the number of jobs from family i, the following equality is satisfied:

$$N = n_0 + n_1 + n_2 + \cdots + n_F, \tag{3}$$

where n_0 is a 'dummy' job family that is introduced to deal with the 'dead time' during the execution of the MRI examination.

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

In each job family $g \in \{0, \dots, F\}$ the processing times p_g of the jobs within the family are identical:

$$p_{(gj)} = p_g, \qquad g = 0, \dots, F,\ j = 0, \dots, n_g.$$

Additionally, the following assumption is imposed:

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:

$$(f,1) \prec (f,2) \prec \cdots \prec (f,n_f), \qquad f = 1 \ldots F \tag{4}$$

in increasing order.

Using the last assumption, the job schedule $L = (j_1, \ldots, j_N)$ that specifies the order in which the jobs are processed can be unambiguously described by the switching sequence σ over the job families:

$$\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_N), \quad \sigma_k \in \{0, \ldots, F\}\,. \tag{5}$$

A. SAR specific problem statement

To handle the SAR limitations an extension to the standard scheduling theory ([1], [3]) 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 designed 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, $i \in \{1, \ldots, m\}$, $m \leq F$ (i.e. the jobs (g,j), $j \in 1, \ldots, n_g$ 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. The switching is not immediate, the setup time is s_{gh}. 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.

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, [6], [7]. 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.

B. Amplifiers temperature specific problem statement

For the case of the amplifier temperature limitations the additional dynamical constraints are imposed. The temperature of the amplifiers depends on the sequence in which the job families are processed and should be kept below the maximal level, $T_{\max}$. The setup times between the families were omitted because the processing times of the jobs is much longer than the setup times for the gradient amplifier temperature limitation case. Temperature $T(k)$ after processing of k^{th} job in the sequence σ can be calculated by the recurrent equation:

$$\begin{cases} T(0) = T_0 \\ T(k) = A_{\sigma_k}\; T(k-1) + B_{\sigma_k}, \end{cases} \tag{6}$$

The maximal temperature during processing of the k^{th} job is expressed by $T(k-1) + M_{\sigma_k}$ and should not exceed the limits:

$$T(k-1) + M_{\sigma_k} < T_{\max}. \tag{7}$$

Finally, the parameters and initial conditions are given:

$$\begin{array}{l} 0 < A_{\sigma_k} < 1 \\ 0 < B_{\sigma_k} \leq M_{\sigma_k} \\ T_0 < T_{max}, \end{array} \tag{8}$$

for each job family $\sigma_k \in \{1, \ldots, F\}$, and for the 'dummy' job $(\sigma_k = 0)$:

$$\begin{array}{l} A_0 = \varepsilon < 1 \\ B_0 = 0 \\ M_0 = 0. \end{array} \tag{9}$$

For each particular type of scan $f \in \{1, \ldots F\}$ the constants A_f, B_f and M_f can be easily calculated knowing the gradient waveform $G_f(t)$ (hence the input power $P_f(t)$) and the length of the segment t_f of the scan:

$$\begin{aligned} A_f &= \exp\left(\frac{-t_f}{\tau}\right) \\ B_f &= \frac{1}{\tau}\int_0^{t_f} \mathrm{e}^{\frac{-(t_f - t')}{\tau}}\; \theta P_f(t')\; \mathrm{d}\, t', \\ M_f &= \max_{0 \leq t \leq t_f} \frac{1}{\tau}\int_0^{t} \mathrm{e}^{\frac{-(t - t')}{\tau}}\; \theta P_f(t')\; \mathrm{d}\, t', \end{aligned} \tag{10}$$

where θ and the time constant τ are the same as in (2).

The objective of the problem is to find the switching sequence

$$\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_N), \quad \sigma_k \in \{0, \ldots, F\} \tag{11}$$

for which all the jobs from the job families a processed

$$\sum_{n=1}^{N} \chi_g(\sigma_n) = n_g, \qquad g \in \{1, \ldots, F\}$$

and the number of 'dummy' jobs is minimal

$$\sum_{n=1}^{N} \chi_0(\sigma_n) = n_0 \longrightarrow \min.$$

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

$$\chi_g(f) = \begin{cases} 1 & \text{if } f = g \\ 0 & \text{if } f \neq g \end{cases}\;. \tag{12}$$

By decreasing the number of 'dummy' jobs the maximum completion time is decreased, thus our problem is equivalent to the problem of the makespan minimization.

$$C_{\max} \to \min . \tag{13}$$

IV. AN ALGORITHM FOR SOLVING THE SCHEDULING PROBLEM

A. SAR Limitations case

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

$$\sigma^* = \arg \min_{\sigma \in \{\text{All sequences of the jobs}\}} C_{\max}. \tag{14}$$

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

The following assumption slightly simplifies the problem, are based on natural hardware restrictions of the MRI machines.

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 $\{f_1, \ldots, f_k\} \subset \sigma$ that have started processing, but have not processed yet, does not exceed two:

$$\# \left\{ g \,\middle|\, 0 < \sum_{i=1}^{k} \chi_g(f_i) < n_g, g = 1 \ldots F \right\} \leq 2. \tag{15}$$

Here $\chi_g \to \{0, 1\}$ 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.

1) The Algorithm: According to assumption 2, the algorithm that solves our problem 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 $a, b \in \{1 \ldots F\}$ 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 $n_g^* = n_g, g = 1 \ldots F$. Afterwards the n_g^* decreases while processing the jobs from the families.

Our scheduling algorithm consists of two parts. First, 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.

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

$$\#\{\pi\} = F! \tag{16}$$

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 $T(g, h), \quad g, h \in \{1, \ldots, F\}$.

1) All pairs of the families are tried to find the most 'gainful' one ($T(\cdot, \cdot) \to \max$); the number of pairs is $\binom{F}{2}$ - the number of 2-element subsets of an F-element set, where $(\cdot)$ stands for binomial coefficients.
2) 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:

$$\#\{\pi\} = \binom{F}{2} + \binom{F-2}{1} + \cdots + \binom{1}{1} = (F-1)^2 \tag{17}$$

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

For a small number of jobs 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:

$$\begin{gathered}(l, \phi(g, h)) = (l, \underbrace{g, h, h, h}_{k^* \leq k_g + k_h}, \overbrace{g, g}^{k_g}, \underbrace{h, h, h}_{k_h}, \overbrace{g, g}^{k_g}, \underbrace{h, h, h}_{k_h}, \ldots \\ \ldots, \overbrace{g, g}^{k_g}, \underbrace{h, h, h}_{k_h}, \overbrace{g, g}^{k_g}, \underbrace{h, h, h}_{k_h})\end{gathered} \tag{18}$$

where

$$\begin{aligned} k_g &= \arg \min_{k \in N} \left(s_{hg} + k \ (p_g + \tau_g) - \tau_{hg} > 0 \right) \\ k_h &= \arg \min_{k \in N} \left(s_{gh} + k \ (p_h + \tau_h) - \tau_{gh} > 0 \right) \end{aligned} \tag{19}$$

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

$$k_g < n_g^*, \qquad k_h < n_h^* \tag{20}$$

$$\tau_g > 0 \quad \text{or} \quad \tau_h > 0 \tag{21}$$

$$\tau_g + \tau_h - (s_{gh} + s_{hg}) > 0, \tag{22}$$

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

The inequality (22) guarantees that there will be a time reduction for every switching from one family to another and a way around. 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 inequalities (20) - (22) hold true then intermixing occurs and one of the job families (g or h) will be completely processed. Suppose, it is family g, i.e.

$$\left\lfloor \frac{n_g^*}{k_g} \right\rfloor \le \left\lfloor \frac{n_h^*}{k_h} \right\rfloor, \tag{23}$$

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

$$\sum_{i=1}^{n} \chi_g(\phi_i) = n_g^*. \tag{24}$$

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

$$\begin{aligned} T(g,h) &= M_{gh}(\tau_g - s_{gh}) + M_{hg}(\tau_h - s_{hg}) = \\ &= M_{hg}(\underbrace{\tau_g + \tau_h - (s_{gh} + s_{hg})}_{>0}) + (\underbrace{M_{gh} - M_{hg}}_{0 \le \cdot \le 2})(\tau_g - s_{gh}). \end{aligned} \tag{25}$$

The first $k^* \le k_g + k_h$ jobs in switching sequence $\phi(g,h)$ (see (18)) 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 $a_h, (0 \le a_h \le k_h)$ available jobs of family h with the objective to maximize $T(g,h)$.

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

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

The new number of jobs that needs to be processed in the sequel intermixing is specified as:

$$\begin{aligned} n_h^* &:= n_h - (k_h \left\lfloor \frac{n_g^*}{k_g} \right\rfloor + a_h) \\ n_g^* &:= 0 \end{aligned} \tag{26}$$

and

$$l := h. \tag{27}$$

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

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

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:

$$O(\left\lfloor \frac{n_g^*}{k_g} \right\rfloor) \sim O(\frac{N}{k^* F}). \tag{28}$$

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

$$O\left(\frac{2^{k^*}}{\sqrt{k^*}} + \frac{n}{k^*} \right), \tag{29}$$

where $n = \frac{N}{F}$ is an average number of jobs in a family and k^* is an 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 are to 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:

$$\sigma^* = \phi(\pi_1, \pi_2), \phi(\pi_{r_1}, \pi_3), \phi(\pi_{r_2}, \pi_4), \dots, \phi(\pi_{r_{F-1}}, \pi_F), \tag{30}$$

where $\pi_{r_i}, \; i = 1, \dots, F-1$ 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}.

B. The Amplifiers Temperature limitations case

The problem can be solved with backtracking algorithm, see e.g. [8], [9], which systematically searches for a solution to the problem among all available cases. Solution σ is built incrementally, and partial candidates σ' are rejected if they cannot be completed to a valid solution. If a partial candidate σ' is rejected, the algorithm backtracks by removing the trailing value from the σ', and then proceeds by trying to extend σ' with alternative values. The solution space, Θ, for the problem is

$$\Theta = \underbrace{F \times F \times \cdot \times F}_{N \text{ times}} = F^N, \tag{31}$$

so each solution $\sigma \in \Theta$. The traversal of the solution space can be represented by a depth-first traversal of a tree. Not all the branches of the searching tree are passed and a large number of candidates is eliminated a with a single test.

A typical backtracking algorithm consists of five procedural parameters: *reject*, *accept*, *first*, *next*. Where *reject(s)* returns *TRUE* if the partial candidate s is rejected; *accept*(s) returns *TRUE* if a valid solution s is found; *first*(s) generates a first extension of candidate s; finally, *next*(s) generates a next alternative extension of the candidate.

When initialized, the algorithm starts with an empty vector $S = [\ ]$ and proceeds backtracking until a solution is found (accepted) or all the partial candidates are rejected. The second parameter of the algorithm is data specific for particular

Procedure IV.1 BACKTRACK1(S, T)

```
Require: Vector $S$, Scalar $T$ (the temperature)
1: if REJECT$(S,T)$ or $Solution \neq [\ ]$ then
2:     return
3: else if $k = n$ then
4:     $Solution \leftarrow S$
5:     return
6: $V \leftarrow$ FIRST(S)
7: while $V \neq [\ ]$ do
8:     BACKTRACK1$(V,\ T * A_{V[end]} + B_{V[end]})$
9:     $k[V[end]] = k[V[end]] - 1$
10:    $V \leftarrow$ NEXT$(V)$
```

instance of the problem (in our case it is the temperature T of the amplifiers). For our problem the procedures *reject — output* are listed below.

In the algorithms the global constants $T_{\max}, n$ and variables $k, Solution$ are used. The constant $T_{\max}$, the same as in (7), represents the limitation for the temperature of the amplifiers. The constant $n = [n_1, n_2, \ldots, n_F, n_{F+1}]$ is the vector of size $F+1$ where $n_i, i \in \{1, \ldots, F\}$ denotes a number of job from family i to be processed and $n_{F+1} = n_0$ represents a number of 'dummy' jobs to cool down the amplifiers. The variable k is the vector of size $F+1$ where $k_i, i \in \{1, \ldots, F+1\}$ denotes the number of jobs in each family that has been already processed on the current step of the algorithm. In the variable $Solution$ the first found valide solution is stored.

The procedure *First* searches for the first job that has not exceeded the family size, and still can be processed:

Procedure IV.2 FIRST(S)

```
$j \leftarrow \{$ first $i : k[i] < n[i]\}$
if $j = \emptyset$ then
    return [ ]
else
    $k[j] = k[j] + 1$
    return $S \cup j$
```

The procedure *Next* searches for the next job that has not exceeded the family size, where $S[end]$ denotes the last element of the partial candidate vector S:

Procedure IV.3 NEXT(S)

```
$j \leftarrow \{$ first $i > S[end] : k[i] < n[i]\}$
if $j = \emptyset$ or $S = [\ ]$ then
    return [ ]
else
    $k[j] = k[j] + 1$
    $S[end] \leftarrow j$
    return $S$,
```

The procedure *Reject* abandons partial candidates if they do not satisfy the restrictions.

Procedure IV.4 REJECT(S, T)

```
Require: Vector $S$, Scalar $T$ (the temperature)
return $T + M_{S[end]} > T_{\max}$
```

The procedure *Accept* checks if the solution is found. In our case it only checks if $k = n$, so it is written inline: (IV.1), lines 3-5.

This algorithm traverses all possible solutions. However, a large number of job families, F, and a large number of jobs, N, can result in large computational efforts. For that case a slight modification of the backtracking procedure is designed (IV.5), (IV.6) which backtracks no more than one step back. The complexity of this heuristical backtracking procedure is only $O(F \cdot N)$. In most cases it provides the same solution as the exact algorithm, but there is a chance that some solutions are lost.

Procedure IV.5 BACKTRACK2(S, T)

```
Require: Vector $S$, Scalar $T$ (the temperature)
if REJECT2$(S,T)$ or $Solution \neq [\ ]$ then
    return
else if $k = n$ then
    $Solution \leftarrow S$
    return
$V \leftarrow$ FIRST(S)
while $V \neq [\ ]$ do
    BACKTRACK2$(V,\ T * A_{V[end]} + B_{V[end]})$
    $V \leftarrow$ NEXT$(V)$
```

Procedure IV.6 REJECT2(S, T)

```
Require: Vector $S$, Scalar $T$ (the temperature)
if $T + M_{S[end]} > T_{\max}$ then
    $k[S[end]] \leftarrow k[S[end]] - 1$
    return TRUE
else
    return FALSE
```

Finally, the main body of the algorithm with initialization of all the parameters is presented in (IV.7). The input of the algorithm is vector $n' = [n'_1, n'_2, \ldots, n'_F]$ of size F where $n'_i, i \in \{1, \ldots, F\}$ denotes a number of job from family i to be processed, and thermal parameters of the gradient amplifiers, such as the initial temperature T_0, the maximum available temperature $T_{\max}$ and thermal constants A, B, M of the dynamical system (6),(7).

The complexity of the algorithm in heuristical case is $O(N \cdot F \cdot n_0)$ where N is the number of jobs, F is the number of job families and n_0 is the minimum size of the 'dummy' jobs family. If the 'while' loop in $(IV.7)$ is replaced by the binary search algorithm (see e.g. [9]) the overall complexity will reduce to $O(N \cdot F \cdot 2 \log n_0)$.

Algorithm IV.7 ALGORITHM($T_{\max}, T_0, A, B, M, n'$)

Require: Vectors A, B, M, n; Scalars $T_{\max}, T_0, F$

```
$Solution \leftarrow [\ ]$
$i = 0$
while $Solution = [\ ]$ do
    $n = n' \cup i$
    $k \leftarrow [\ \underbrace{0, 0, \ldots, 0}_{F \text{ times}}\ ]$
    $\left.\begin{array}{l}\textsc{Backtrack1}([\ ], T_0)\\ \textsc{Backtrack2}([\ ], T_0)\end{array}\right\}$ either exact or heuristical
    $i = i + 1$
return $Solution$
```

Two algorithms were designed that solve our scheduling problem for te case of gradient amplifiers temperature limitations. The first (IV.1) is the exact backtracking algorithm, where all the candidates for the solutions are checked. And the second (IV.5) is the heuristical one, where some solutions can be lost, but the efficiency is better. In typical MRI application the number of jobs is about 100 and a number of job families is less then 10. For these input parameters both algorithms find the correct solution in a reasonable time, and can be implemented in the Philips MRI software for 'online' scheduling of the scans segments.

V. CONCLUSIONS

In this paper scheduling algorithms for the MRI Systems are studied. Classical scheduling model was extended to the case of dynamical constraints and family-dependent setup times. Two types of limitations were investigated and several algorithms are described that can be utilized to schedule the scan segments during the MRI examinations on different time scale to reduce the examination time.

In future work these algorithms will be combined and extended to solve the multivariate resource-management problem, to cover all possible limitation on the MRI scanners workflow.

ACKNOWLEDGMENT

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.

REFERENCES

[1] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 3rd ed. Berlin: Springer, 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] A. Allahverdi, C. T. Ng, T. C. E. Cheng, and M. Y. Kovalyov, "A survey of scheduling problems with setup times or costs," *European Jornal of Operational Research*, vol. 187, pp. 985–1032, 2008.

[4] E. Ivanov, A. Pogromsky, and J. Rooda, "Scheduling with sequence dependent setup times in application to magnetic resonance imaging scans processing." Submited to IEEE Multi-conference on Systems and Control, 2009.

[5] E. Ivanov, J. Rooda, and A. Pogromsky, "Scheduling with dynamic constraints in application to magnetic resonance imaging scans processing." Submited to IEEE Multi-conference on Systems and Control, 2009.

[6] L. Bianco, A. Mingozzi, and S. Ricciardelli, "The travelling salesman problem with cumulative costs," *Networks*, vol. 23, pp. 81–91, 1993.

[7] M. Fischetti, G. Laporte, and S. Martello, "The delivery man problem and cumulative matroids, operations research," *Operations Research*, vol. 41, pp. 1055–1064, 1993.

[8] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. The MIT Press and McGraw-Hill Book Company, 1989.

[9] D. Knuth, *The Art of Computer Programming, 3rd Edition*. Addison-Wesley, 1997, vol. 3,4.