

Scheduling with dynamic constraints 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 temperature dynamics is involved and the objective is to minimize the total production time.

I. INTRODUCTION

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

The objective function of our problem is to minimize the total production time (makespan) that, according to [3], together with maximum tardiness are the most prominent measures among the scheduling objectives in industrial applications.

Special exact and heuristic algorithms are designed and applied to the real examination protocols of Philips Healthcare MRI scanners.

The outline of the paper is as 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 of MRI systems to calculate the total time reduction per examination.

II. BASIC MRI THEORY

In this section, following [4], 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 (see Fig. 1) 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.



Fig. 1. Philips MRI Scanner

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.

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

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. 3). In this figure the thin tall bars are the RF-pulses, the thick bars are the gradient changes.

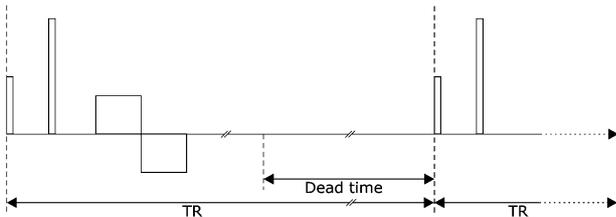


Fig. 2. Imaging sequences

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. Gradient Amplifiers

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

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

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

$$T(t) = e^{-t/\tau} T(0) + \frac{1}{\tau} \int_0^t e^{-\frac{(t-t')}{\tau}} \theta P(t') dt', \quad (1)$$

where the heat resistance θ and the time constant τ are given.

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 periodical trapezoidlike function, where the period is one repetition time (TR). The $G(t)$ function represents the strength and direction of applied magnetic fields of the gradient coils (these fields are additional to the main powerful

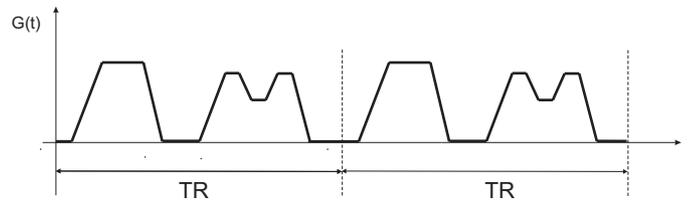


Fig. 3. Gradient waveforms

magnetic field of the scanner, and they play essential role in the process of obtaining the image of a human body).

Typical MRI examination consist of 5-7 scans with different gradient waveforms. Some of them can overheat the amplifiers. To prevent this harmful behavior an amount of ‘dead time’ is included into every repetition time (TR) to keep the temperature of the amplifiers below the limits.

C. Scan Segments Mixing

The temperature of gradient amplifiers, gradient and RF coils, the SAR level 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. Scheduling algorithms related to SAR-limited scans problem are covered in [5].

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’. For each type of scan the minimal size of the segment is given. 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 then the dead time in the scans, and it is neglectable when compared with the segments size.

Therefore the main problem is to find an algorithm of intermixing the segments of different scans to decrease the ‘dead time’, and hence the examination time.

To increase the scanner workflow the ‘dead time’ should be utilized for a useful job.

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 equality is satisfied:

$$N = n_0 + n_1 + n_2 + \dots + n_F, \quad (2)$$

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:

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

Additionally, an 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:

$$(f, 1) \prec (f, 2) \prec \dots \prec (f, n_f), \quad f = 1 \dots F \quad (3)$$

in increasing order.

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

$$\sigma = (\sigma_1, \sigma_2, \dots, \sigma_N), \quad \sigma_k \in \{0, \dots, F\}. \quad (4)$$

The processing of the jobs have additional dynamical constraints. The temperature of the amplifiers depend on the sequence in which the job families are processed and should be kept below the maximal level, T_{\max} . The temperature $T(k)$ after processing of k^{th} job in the sequence σ can be calculated by the recurrent equation:

$$\begin{cases} T(0) = T_0 \\ T(k) = A_{\sigma_k} T(k-1) + B_{\sigma_k}, \end{cases} \quad (5)$$

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:

$$T(k-1) + M_{\sigma_k} < T_{\max}. \quad (6)$$

Finally, the parameters and initial conditions are given:

$$\begin{aligned} 0 < A_{\sigma_k} < 1 \\ 0 < B_{\sigma_k} &\leq M_{\sigma_k} \\ T_0 < T_{\max}, \end{aligned} \quad (7)$$

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

$$\begin{aligned} A_0 &= \varepsilon < 1 \\ B_0 &= 0 \\ M_0 &= 0. \end{aligned} \quad (8)$$

For each particular type of scan $f \in \{1, \dots, F\}$ 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:

$$\begin{aligned} A_f &= \exp\left(\frac{-t_f}{\tau}\right) \\ B_f &= \frac{1}{\tau} \int_0^{t_f} e^{\frac{-(t_f-t')}{\tau}} \theta P_f(t') \, dt', \\ M_f &= \max_{0 \leq t \leq t_f} \frac{1}{\tau} \int_0^t e^{\frac{-(t-t')}{\tau}} \theta P_f(t') \, dt', \end{aligned} \quad (9)$$

where θ and the time constant τ are the same as in (1).

The objective of the problem is to find the switching sequence

$$\sigma = (\sigma_1, \sigma_2, \dots, \sigma_N), \quad \sigma_k \in \{0, \dots, F\} \quad (10)$$

for which all the jobs from the job families a processed

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

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 :

$$\chi_g(f) = \begin{cases} 1 & \text{if } f = g \\ 0 & \text{if } f \neq g \end{cases}. \quad (11)$$

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.

$$C_{\max} \rightarrow \min. \quad (12)$$

IV. AN ALGORITHM FOR SOLVING THE PROBLEM

The problem can be solved with backtracking algorithm, see e.g. [6], [7], which systematically searches for a solution to the problem among all available cases. The solution σ is build 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

$$\Theta = \underbrace{F \times F \times \dots \times F}_{N \text{ times}} = F^N, \quad (13)$$

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.

Typical backtracking algorithm consists of five procedural parameters: *reject*, *accept*, *first*, *next*, and *output*. Where *reject(s)* returns *TRUE* if the partial candidate *s* is rejected; *accept(s)* returns *TRUE* if valid solution *s* is found; *first(s)* generates a first extension of a candidate *s*; *next(s)* generates a next alternative extension of a candidate; finally, *output(s)* uses solutions *s* as valid for the problem:

Procedure IV.1 BACKTRACKING(*S*, *Param*) (General form of the backtracking procedure)

Require: Vector *S*, Parameters *Param*

```

1: if REJECT(S, Param) then
2:   return
3: else if ACCEPT(S, Param) then
4:   OUTPUT(S, Param)
5:   return
6: V ← FIRST(S, Param)
7: while V ≠ [ ] do
8:   BACKTRACKING(V, Param)
9:   V ← NEXT(V, Param)

```

Where *Param* is data specific for particular instance of the problem (in our case it is the temperature *T* of the amplifiers). 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.

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} , same as in (6), represents limitation for the temperature of the amplifiers. The constant $n = [n_1, n_2, \dots, n_F, n_{F+1}]$ is the vector of size $F + 1$ where $n_i, i \in \{1, \dots, F\}$ 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 $k_i, i \in \{1, \dots, F + 1\}$ 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 have not exceed the family size, and still can be processed:

Procedure IV.2 FIRST(*S*)

```

1: j ← { first i :  $k[i] < n[i]$  }
2: if j = ∅ then
3:   return [ ]
4: else
5:    $k[j] = k[j] + 1$ 
6:   return  $S \cup j$ 

```

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

Procedure IV.3 NEXT(*S*)

```

1: j ← { first i >  $S[end]$  :  $k[i] < n[i]$  }
2: if j = ∅ or  $S = [ ]$  then
3:   return [ ]
4: else
5:    $k[j] = k[j] + 1$ 
6:    $S[end] \leftarrow j$ 
7:   return S,

```

The procedure *Output* saves the solution to the global variable *Solution*.

Procedure IV.4 OUTPUT(*S*)

```

1: Solution ← S
2: Display Solution
3: return

```

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

Procedure IV.5 REJECT(*S*, *T*)

Require: Vector *S*, Scalar *T* (the temperature)

```

1: if  $T + M_{S[end]} > T_{\max}$  then
2:   return TRUE
3: else
4:   return FALSE

```

The procedure *Accept* checks if the solution is found:

Procedure IV.6 ACCEPT(*S*)

```

1: if  $k = n$  then
2:   return TRUE
3: else
4:   return FALSE

```

The version of the backtracking procedure specific to our problem is listed in (IV.7).

This algorithm traverses all possible solutions. However, a big number of job families, F , and a big number of jobs, N , can result in large computational effort. For that case a slight modification of the backtracking procedure is designed (IV.8), (IV.9) 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 probability that some solutions will be lost.

Finally, the main body of the algorithm with initialization of all the parameters is presented in (IV.10). The input of the algorithm is vector $n' = [n'_1, n'_2, \dots, n'_F]$ of size F where $n'_i, i \in \{1, \dots, F\}$ 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 (5),(6).

Procedure IV.7 BACKTRACK1(S, T)

Require: Vector S , Scalar T (the temperature)

```
1: if REJECT( $S, T$ ) or  $Solution \neq []$  then
2:   return
3: else if ACCEPT( $S$ ) then
4:   OUTPUT( $S$ )
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:   $V \leftarrow NEXT(V)$ 
```

Procedure IV.8 BACKTRACK2(S, T)

Require: Vector S , Scalar T (the temperature)

```
1: if REJECT2( $S, T$ ) or  $Solution \neq []$  then
2:   return
3: else if ACCEPT( $S$ ) then
4:   OUTPUT( $S$ )
5:   return
6:  $V \leftarrow FIRST(S)$ 
7: while  $V \neq []$  do
8:   BACKTRACK2( $V, T * A_{V[end]} + B_{V[end]}$ )
9:    $V \leftarrow NEXT(V)$ 
```

Procedure IV.9 REJECT2(S, T)

Require: Vector S , Scalar T (the temperature)

```
1: if  $T + M_{S[end]} > T_{max}$  then
2:    $k[S[end]] \leftarrow k[S[end]] - 1$ 
3:   return TRUE
4: else
5:   return FALSE
```

Algorithm IV.10 ALGORITHM($T_{max}, T_0, A, B, M, n'$)

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

```
1:  $Solution \leftarrow []$ 
2:  $i = 0$ 
3: while  $Solution = []$  do
4:    $n = n' \cup i$ 
5:    $k \leftarrow [ \underbrace{0, 0, \dots, 0}_{F \text{ times}} ]$ 
6:    $\left. \begin{array}{l} \text{BACKTRACK1}([ ], T_0) \\ \text{BACKTRACK2}([ ], T_0) \end{array} \right\}$  either exact or heuristical
7:    $i = i + 1$ 
8: return  $Solution$ 
```

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.10) is replaced by the binary search algorithm (see e.g. [7]) the overall complexity will reduce to $O(N \cdot F \cdot 2 \log n_0)$.

Two algorithms were designed that solve our scheduling problem. The first (IV.7) is the exact backtracking algorithm, where all the candidates for the solutions are checked. And the second (IV.8) 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. APPLICATION TO MRI SYSTEMS

The results of application of the algorithm to the real MRI system will be added in the final version of the paper, after approval of Philips Healthcare.

VI. CONCLUSIONS

In this paper scheduling algorithms for the MRI Systems are studied. Classical scheduling model was extended to the case of dynamical constraints. Two modifications of the algorithm are described that can be utilized to schedule the scan segments during the MRI examinations to reduce the examination time.

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

REFERENCES

- [1] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 3rd ed. Berlin: Springer, 2008.
- [2] 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.
- [3] S. S. Panwalker and W. Iskander, “A survey of scheduling rule,” *Operations Research*, vol. 25, pp. 45–61, 1977.
- [4] PMS, *Basic principles of MR Imaging*. NL: Philips Medical Systems, 2005.
- [5] E. Ivanov, A. Pogromsky, and J. Rooda, “Scheduling with sequence dependent setup times in application to magnetic resonance imaging scans processing.” IEEE Multi-conference on Systems and Control, 2009.
- [6] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. The MIT Press and McGraw-Hill Book Company, 1989.
- [7] D. Knuth, *The Art of Computer Programming, 3rd Edition*. Addison-Wesley, 1997, vol. 3,4.