Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2018) 26: 844 – 855
© TÜBİTAK
doi:10.3906/elk-1706-175

Schedulability test for IMA systems based on mixed integer linear programming formulation

Yukai HAO*, **Ming MU, Xiaodi DAI, Xiangdong LI**
Aeronautics Computing Technique Research Institute, Xi'an, P.R. China

Received: 14.06.2017 • **Accepted/Published Online:** 14.11.2017 • **Final Version:** 30.03.2018

Abstract: Integrated modular avionics (IMA) architecture is widely adopted for the design of modern aircraft. It simplifies the system development process and improves the system security and reliability. In IMA systems, avionics applications are packed into various partitions, and integrated into a standard computing platform. How to determine the schedulability of systems is one of the key problems. In this paper, using the characters of avionics systems, a partition model with a strict period is built, and constraints in space, time, and communication are analyzed. Based on the mixed integer linear programming formulation, a solution to determine the schedulability of IMA systems is presented. Experience reveals that this solution not only determines the system schedulability, but also achieves the required minimum number of modules and guides the design of IMA systems.

Key words: Integrated modular avionics (IMA), schedulability test, partition, scheduling analysis

1. Introduction

With the rapid development of microelectronics and computer technology, the complexity of aircraft increases, and the design of avionics systems is developing in the direction of integration, modularization, generalization, and intellectualization [1]. The traditional federated architecture is not suitable for large-scale avionics systems. Integrated modular avionics (IMA) architecture, simplifying the design of avionics software and hardware and improving the system security and reliability, is widely accepted by the avionics industry and adopted in the system design of modern civil and military aircraft, such as B787, Airbus A380, and Lockheed Martin F-22 Raptor.

In IMA architecture, a partition is the basic execution environment of software applications according to the ARINC 653 standard. Avionics tasks are packed into various partitions, and integrated into a standard and shared computing platform. With the segregation of space and time, IMA architecture integrates the system resources, separates hardware operations from user applications, and provides shared computing and communication resources. It not only guarantees that the applications can be designed and verified independently, but also achieves deep system integration and information sharing [2].

Although IMA architecture reduces the weight and power consumption of the whole avionics system, it brings serious partition distribution and scheduling problems. When partitions are integrated through IMA architecture, designers have to determine the system schedulability and allocate proper resources and time windows for each partition, in order to ensure the correctness and reliability of avionics systems. The main problem amounts to finding a method that associates a module and a time window to each partition, such that

*Correspondence: ykhao@stu.xidian.edu.cn

all imposed constraints, i.e. space constraint, time constraint, and communication delay constraint, are verified for all modules and partitions. However, as the number of partitions increases, the schedulability determination and resource allocations of partitions become more and more serious. Designers find it impractical to solve the problems only by hand, and are more dependent on the schedulability algorithms and decision-making tools for resource allocations.

The partitions can be modeled as nonpreemptive tasks with strict periods, and the schedulability problem is classified as a nonpreemptive and strictly periodic scheduling problem. It is very difficult to solve this problem, because not only does the nonpreemptive attribute make it as difficult as NP hard [3], but also the strictly periodic constraint increases the difficulty in obtaining schedulability conditions [4]. The schedulability problem of tasks with strict periods or partitions in IMA systems is also one of the key problems in real-time scheduling theory research [5].

Korst et al. [6] addressed the scheduling problem on two strictly periodic tasks, and presented a necessary and sufficient schedulability condition, which had been proved to be a sufficient condition [7] for more than two tasks. [8,9] solved the problem on a minimum processor platform and [10] gave a scheduling heuristic based on the constraint that the period of new task was a multiple of those of the existing tasks [11]. With the idea of game theory, [12] and [13] proposed best-response algorithms [14] to compute the crucial scaling factor of all partitions and used it to determine the schedulability of partitions on a limit number of modules. However, the partitions involved in these results are independent, and without communication constraints between partitions, which narrows the range of applications. From the perspective of safety and reliability, [15] and [16] presented a distribution strategy of partitions with communication dependency, and adopted graphic theory to reduce the number of variables in the searching process. However, the communication delay is described with data chains, and analyzed under the worst-case situations, without precisely expressing data transmission constraints between partitions. At the same time, this kind of solution does not produce the minimum number of modules required by the system, and cannot reduce the system weight and power consumption.

This paper first analyzes the characters of the partitions in IMA systems, builds partition model with a strict period, and describes the constraints in space, time, and communication delay. Then, using mixed integer linear programming (MILP) [17], the paper proposes an efficient solution to determine the schedulability of IMA systems. Finally, experiments are conducted to show that the proposed solution determines the system schedulability and achieves the minimum number of modules required.

2. System module

2.1. Partition mechanism in IMA systems

Partition is an important concept in IMA systems. As shown in Figure 1, IMA architecture packets avionics tasks into partitions and allocates partitions to modules. IMA architecture realizes space separation and time separation between applications through partition mechanism.

Space separation: each partition is allocated to a module and gets a series of space resources such as memory. Only the tasks in this partition can access these resources [18].

Time separation: every partition is distributed to a given time window, in which tasks in this partition can be executed according to a certain scheduling algorithm. When the time window of the partition expires, the partition will be hung up. Tasks in the partition are not executed until the next time window arrives.

In IMA architecture, each module can process multiple partitions with different periods according to the given scheduling table. The scheduling table lists the execution order, starting execution time, and end time of all partitions on this module.

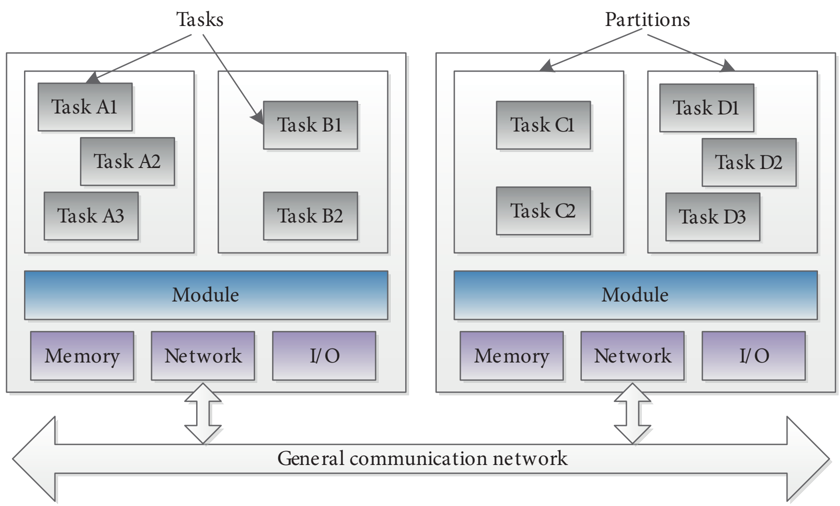

Figure 1. Partition mechanism in IMA system.

2.2. System module

Consider an IMA system constituted of m modules $\Pi = \{\pi_1, \pi_2, ..., \pi_m\}$ and n partitions $\Theta = \{\theta_1, \theta_2, ..., \theta_n\}$. Each module $\pi_k (k \in [1, m])$ has a fixed available memory space M_k and the maximum number of partitions N_k that it can host. Meanwhile, we use m row m column array $\vec{\lambda} = (\lambda_{k,l})$ $(k, l \in [1, m])$ to express communication delay between modules. Each element $\lambda_{k,l}$ is a floating number, representing the maximum data transmission delay between the modules π_k and π_l. When $k = l$, $\lambda_{k,l}$ represents the communication delay inside modules.

Each partition $\theta_i (i \in [1, n])$ is characterized by a triple $\theta_i = \langle c_i, p_i, m_i \rangle (i \in [1, n])$, where c_i, p_i, and m_i respectively represent the worst case execution time, period, and the memory space required by the partition θ_i.

In an IMA system, some partitions may be in exclusion for security reasons, i.e. they cannot run on the same module. An n row n column array $\vec{\eta} = (\eta_{i,j})$ $(i, j \in [1, n])$ represents the exclusive relationship between partitions. Each element $\eta_{i,j}$ is a Boolean variable, expressing whether partitions θ_i and θ_j are exclusive or not. When partitions θ_i and θ_j cannot run on the same module, $\eta_{i,j} = 1$; otherwise $\eta_{i,k} = 0$. That is,

$$\eta_{i,j} = \begin{cases} 1 & \text{if } \theta_i \text{ and } \theta_j \text{ collide} \\ 0 & \text{otherwise} \end{cases}$$

Every partition is made up of infinite jobs and under the strictly periodic constraint the time interval between any two continuous jobs is fixed and equal to the period of the partition. Hence, if the start execution time of partition θ_i is s_i, its r_{th} job starts at $s_i + rp_i$ and ends at $s_i + rp_i + c_i$. Let $B_i^r(s_i)$represent the time units occupied by the r_{th} job of θ_i; then $B_i^r(s_i) = [s_i + rp_i, s_i + rp_i + c_i)$. The partition model used in this paper is illustrated in Figure 2.

In IMA systems, data may be received and sent by partitions along a processing chain. We use n row n column array$\vec{\delta} = (\delta_{i,j})$ $(i, j \in [1, n])$ to express the maximum communication delay between partitions. Each element $\delta_{i,j}$ is a floating-point number, representing the maximum available time after the data are sent out from partition θ_i and before the data are received by partition θ_j. When $i = j$, $\delta_{i,j}$ expresses maximum communication delay inside partition θ_i, i.e. $\delta_{i,i} = p_i$.

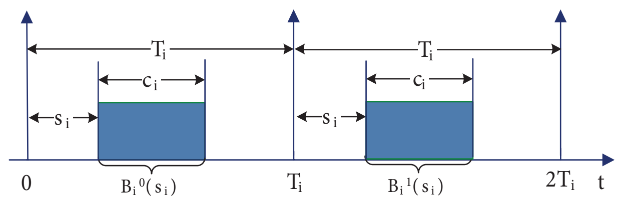

Figure 2. Partition model.

3. Schedulability analysis

When an IMA system is schedulable, all partitions and modules need to meet constraints in space, time, and communication delay [15].

1) Space constraint: (C1) Each partition must be hosted by one and only one module; (C2) Exclusive partitions cannot be hosted by the same module; (C3) The number of partitions running on each module cannot exceed the maximum number that the module supports; (C4) Total memory required by all partitions running on the same module cannot exceed the available maximum memory space of the module.

2) Time constraint: (C5) The first job of each partition shall be completed before the period of the partition ends; (C6) Any two partitions allocated on the same module have no time conflict.

3) Communication delay constraint: (C7) The time of data transmission between any two partitions shall not exceed the predefined maximum communication delay of the two partitions.

When the partition set $\Theta = \{\theta_1, \theta_2, ..., \theta_n\}$ on m modules is schedulable, each partition shall be allocated to a suitable module and an effective start time, such that the whole IMA system meets constraints in all above three aspects.

3.1. Space constraints analysis

Establish an n row$m$ column array $\vec{a} = (a_{i,k})$ $(1 \leq i \leq n \text{and} 1 \leq k \leq m)$ to describe the allocations of partitions. Each element $a_{i,k}$ is a Boolean variable, representing whether partition θ_i is allocated to the module π_k or not. When partition θ_i is allocated to π_k, $a_{i,k} = 1$; otherwise $a_{i,k} = 0$. That is,

$$a_{i,k} = \begin{cases} 1 & if\ \theta_i \text{ is assigned to module } \pi_i \\ 0 & \text{otherwise} \end{cases}$$

Space constraint (C1) shows that one and only one module can be allocated to any partition θ_i; hence, the sum of every row in partition assignment array $\vec{a} = (a_{i,k})$ is equal and only equal to 1, which can be expressed as

$$\forall i \in [1, n], \sum_{1 \leq k \leq m} a_{i,k} = 1$$

Space constraint (C2) requires that exclusive partitions cannot run on the same module, i.e. for partitions θ_i and θ_j, if $\eta_{i,j} = 1$, their allocation $a_{i,k}$, $a_{j,k}$ on any module π_k cannot equal 1 at the same time, which can be expressed as

$$\begin{gathered} \forall i, j \in [1, n], \forall k \in [1, m] \\ \eta_{i,j} = 1 \Rightarrow a_{i,k} + a_{j,k} \leq 1 \end{gathered}$$

In mixed integer linear programming formulation, the following constraint condition can be used to describe space constraint (C2):

$$\forall i,j \in [1,n], \forall k \in [1,m]$$
$$\eta_{i,j} \times (a_{i,k} + a_{j,k}) \le 1$$

Space constraint (C3) restricts the number of partitions hosted by each module. The number of partitions on every module cannot exceed the maximum number that the module supports, which can be expressed as

$$\forall k \in [1,m], \sum_{1\le i\le n} a_{i,k} \le N_k$$

Space constraint (C4) is a limitation of memory resources on modules. For any module π_k, the total memory space required by all partitions running on it is no more than M_k, which can be expressed as

$$\forall k \in [1,m], \sum_{1\le i\le n} a_{i,k} m_i \le M_k$$

3.2. Time constraints analysis

Create an array $\vec{s} = (s_i)$ that contains n elements to represent the offset (i.e. the start time of the first job) of each partition. Time constraint (C5) shows the value range of the start time of each partition, and can be expressed as

$$\forall i \in [1,n], 0 \le s_i \le p_i - c_i$$

Time constraint (C6) restricts the time windows of all partitions on the same module, requiring that no time unit overlaps between any two partitions allocated to the same module.

When two partitions θ_i and θ_j are schedulable on the same module, all their jobs have no overlapping time unit, i.e.

$$\forall k,l \ge 0, B_i^k(s_i) \cap B_j^l(s_j) = \emptyset \tag{1}$$

Although condition (1) is a sufficient and necessary condition to determine the schedulability of two partitions, it cannot be directly used because the jobs of partitions are generated infinitely [19]. [6] proposes a more efficient and convenient determining condition.

Theorem 1 [6] Partitions θ_i and θ_j are schedulable on the same module, if and only if

$$c_i \le (s_j - s_i) mod(g_{i,j}) \le g_{i,j} - c_j, \tag{2}$$

where $g_{i,j}$ is the greatest common divisor of periods of θ_i and θ_j, i.e. $g_{i,j} = GCD(p_i, p_j)$.

For any two partitions, if they can be allocated to the same module, the start times of the two partitions should meet condition (2). Then time constraint (C6) can be expressed as

$$\begin{gathered} \forall i,j \in [1,n], \forall k \in [1,m], a_{i,k} = a_{j,k} = 1, \\ c_i \le (s_j - s_i) mod(g_{i,j}) \le g_{i,j} - c_j \end{gathered} \tag{3}$$

In conditions (2) and (3), mod is not a linear operation; in MILP formulation, $(s_j - s_i) mod(g_{i,j})$ should be replaced by the following equation:

$$(s_j - s_i) mod(g_{i,j}) = (s_j - s_i) - g_{i,j} \times e_{i,j}$$

$e_{i,j}$ is a new integer variable, representing the quotient from the modulo operation mod, and its value ranges from $(c_i - p_i)/g_{i,j}$ to $(p_j - c_j)/g_{i,j}$. Then condition (3) should be updated to

$$\forall i, j \in [1, n], \forall k \in [1, m], a_{i,k} = a_{j,k} = 1,$$
$$c_i \leq (s_j - s_i) - g_{i,j} \times e_{i,j} \leq g_{i,j} - c_j$$
$$\frac{c_i - p_i}{g_{i,j}} \leq e_{i,j} \leq \frac{p_j - c_j}{g_{i,j}}$$

3.3. Communication delay constraint analysis

Constraint (C7) is a limitation of communication delays between partitions. The date transmission between any two partitions must be completed in a predefined time interval. As shown in Figure 3, for any two partitions θ_i and θ_j, there are two time delays after data are sent out from partition θ_i and before partition θ_j receives them: 1) Communication delay between modules where partitions θ_i and θ_j are hosted, which is expressed with $d_{i,j}$; (2) Time delay after the data reach partition θ_j and before the job of θ_j begins, which is represented by $b_{i,j}$. Communication delay constraint (C7) can be expressed as

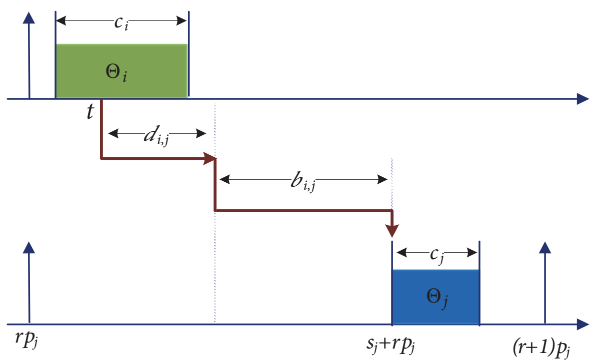

Figure 3. Data transfer time between two partitions.

$$\forall i, j \in [1, n], d_{i,j} + b_{i,j} \leq \delta_{i,j} \tag{4}$$

The modules on which partitions θ_i and θ_j run are determined by the i_{th} row and j_{th} column in the partition attribution vector $\vec{a}$. $d_{i,j}$ can be expressed as

$$d_{i,j} = \sum_{1 \leq k \leq m} \sum_{1 \leq l \leq m} a_{i,k} a_{j,l} \lambda_{k,l} \tag{5}$$

In Eq. (5), $a_{i,k}$ and $a_{j,l}$ are Boolean variables and their product is a quadratic constrain [20]. Then Eq. (5) should be transformed in the MILP formulation. Introduce a Boolean variable $w_{i,j,k,l}$ to express the product of $a_{i,k}$ and $a_{j,l}$, i.e. $w_{i,j,k,l} = a_{i,k} \times a_{j,l}$. When and only when $a_{i,k}$ and $a_{j,l}$ are equal to 1, $w_{i,j,k,l} = 1$; otherwise, $w_{i,j,k,l} = 0$. $w_{i,j,k,l}$ can be expressed as

$$w_{i,j,k,l} \in [0, 1]$$
$$w_{i,j,k,l} \leq a_{i,k}$$
$$w_{i,j,k,l} \leq a_{j,l}$$
$$w_{i,j,k,l} \geq a_{i,k} + a_{j,l} - 1$$

Putting $w_{i,j,k,l}$ into Eq. (5), $d_{i,j}$ can be expressed as

$$d_{i,j} = \sum_{1 \le k \le m} \sum_{1 \le l \le m} w_{i,j,k,l} \lambda_{k,l}$$

If an item of data is sent out from partition θ_i at time t, it reaches the module that partition θ_j runs on at the time $t + d_{i,j}$. Assume the next job of partition θ_j is its r_{th} job, then

$$b_{i,j} = s_j + rp_j - t - d_{i,j} \tag{6}$$

In the worst case, when an item of data arrives at the module that partition θ_j runs on, it happens to slightly miss the start execution of a job of partition θ_j. In this situation, the waiting time is p_j and so $b_{i,j} \le p_j$. Take a modular operation on both sides of Eq. (6) with p_j:

$$\begin{aligned} b_{i,j} &= (b_{i,j}) mod(p_j) \\ &= (s_j + rp_j - t - d_{i,j}) mod(p_j) \\ &= (s_j - t - d_{i,j}) mod(p_j) \end{aligned} \tag{7}$$

In Eq. (7), t is a float variable, representing a time that any job of partition θ_i runs, $t \in \{y \mid s_i + xp_i \le y < s_i + c_i + xp_i, \forall x \ge 0\}$. Since partitions θ_i and θ_j are with strict periods, $t \in \{y | s_i + xp_i \le y \le s_i + c_i + xp_i, 0 \le x \le \frac{lcm(p_i, p_j)}{p_i}\}$. Putting Eq. (7) into Condition (4), Condition (4) changes to

$$\forall i, j \in [1, n], d_{i,j} + (s_j - t - d_{i,j}) mod(p_j) \le \delta_{i,j} \tag{8}$$

Similarly, the mod operation in Condition (8) is not linear; in the MILP formulation, $(s_j - t - d_{i,j}) mod(p_j)$ should be replaced with the following equation:

$$(s_j - t - d_{i,j}) mod(p_j) = (s_j - t - d_{i,j}) - p_j \times q_{i,j}$$

$q_{i,j}$ is an integer variable, representing the integer quotient from the modulo operation mod, i.e. $q_{i,j} = (s_j - t - d_{i,j})/p_j$. The value range of $q_{i,j}$ is $[(-t - d_{i,j})/p_j, (p_j - c_j - t - d_{i,j})/p_j]$. Hence, Condition (4) should be changed to

$$\forall i, j \in [1, n], s_j - t - p_j \times q_{i,j} \le \delta_{i,j}$$

$$\frac{-t - d_{i,j}}{p_j} \le q_{i,j} \le \frac{p_j - c_j - t - d_{i,j}}{p_j}$$

3.4. MILP solution

In this section, we determine the schedulability of IMA systems based on a MILP formulation. MILP is an exact framework for linear programs in which some or all variables are required to take integer value, and can completely search the resolution space to find a feasible solution for a periodic scheduling problem under limited number of modules and partitions in IMA systems. We first use linear conditions to describe all space, time, and communication delay constraints when the system is schedulable; we then list all time windows and module allocations for the partitions and judge whether all constraint conditions are met. If there exists a valid allocation to meet all constraint conditions, the IMA system is schedulable; otherwise, the system is unschedulable.

In practice, there may be many allocations that satisfy the constraint conditions and ensure the system is schedulable. In order to achieve better system performance, an optimum object is often set in the process of solving the schedulability problem. A strategy that is the most frequently used is to get the minimum number of modules required by partitions, which reduces the weight and power consumption of the system as much as possible.

When the number of partitions allocated to a given module is zero, i.e. all values of the column in partition attribution vector $\vec{a}$ are zero, this module is not used. Use an array $\vec{z_k}$ that contains m elements to record the use situations of modules. Each element z_k is a Boolean value, representing whether the module π_k is used. If and only if $\forall i \in [1,n], a_{i,k} = 0$, $z_k = 0$; otherwise, $z_k = 1$. z_k can be expressed as follows:

$$\forall i \in [1,n], a_{i,k} \leq z_k \leq 1$$

$$z_k \leq \sum_{1 \leq i \leq n} a_{i,k}$$

Let m' denote the number of models used in an IMA system; then $m' = \sum_{1 \leq k \leq m} z_k$. In the searching process of solving the schedulability problem by MILP formulation, the optimum object is to minimize the number of modules used, and the constraints are the limitations of schedulability. The whole programming can be written as follows:
minimum m'
subject to

$$\forall i \in [1,n], \forall k \in [1,m], a_{i,k} \in \{0,1\} \tag{9}$$

$$\begin{aligned} &m' = \sum_{1 \leq k \leq m} z_k \\ &\forall i \in [1,n], \forall k \in [1,m], a_{i,k} \leq z_k \leq 1 \\ &\forall k \in [1,m], z_k \leq \sum_{1 \leq i \leq n} a_{i,k} \end{aligned} \tag{10}$$

$$\forall i \in [1,n], \sum_{1 \leq k \leq m} a_{i,k} = 1 \tag{11}$$

$$\begin{aligned} &\forall i,j \in [1,n], \forall k \in [1,m] \\ &\eta_{i,j} \times (a_{i,k} + a_{j,k}) \leq 1 \end{aligned} \tag{12}$$

$$\forall i \in [1,n], 0 \leq s_i \leq p_i - c_i \tag{13}$$

$$\forall k \in [1,m], \sum_{1 \leq i \leq n} a_{i,k} \leq N_k \tag{14}$$

$$\forall k \in [1,m], \sum_{1 \leq i \leq n} a_{i,k} m_i \leq M_k \tag{15}$$

$$
\begin{aligned}
&\forall i,j \in [1,n], \forall k \in [1,m], a_{i,k} = a_{j,k} = 1, \\
&c_i \le (s_j - s_i) - g_{i,j} \times e_{i,j} \le g_{i,j} - c_j \\
&\tfrac{c_i - p_i}{g_{i,j}} \le e_{i,j} \le \tfrac{p_j - c_j}{g_{i,j}}
\end{aligned} \tag{16}
$$

$$
\begin{aligned}
&\forall i,j \in [1,n], w_{i,j,k,l} \in [0,1] \\
&w_{i,j,k,l} \le a_{i,k}, w_{i,j,k,l} \le a_{j,l} \\
&w_{i,j,k,l} \ge a_{i,k} + a_{j,l} - 1 \\
&d_{i,j} = \sum_{1 \le k \le m} \sum_{1 \le l \le m} w_{i,j,k,l} \lambda_{k,l} \\
&t \in \{y | s_i + x p_i \le y \le s_i + c_i + x p_i, \\
&\qquad 0 \le x \le \tfrac{lcm(p_i,p_j)}{p_i}\} \\
&\forall i,j \in [1,n], s_j - t - p_j \times q_{i,j} \le \delta_{i,j} \\
&\tfrac{-t - d_{i,j}}{p_j} \le q_{i,j} \le \tfrac{p_j - c_j - t - d_{i,j}}{p_j}
\end{aligned} \tag{17}
$$

In the linear programming solution, Condition (9) gives the value range of each element in partition attribution vector $\vec{a}$; Condition (10) shows the use situations of the modules; Conditions (11), (12), (13), (14), (15), (16), and (17) are derived from schedulability constraints (C1) to (C7). This formulation not only gets the minimum number of modules required by partitions, but also provides a feasible module and time window allocation for each partition while respecting the space constraint, time constraint, and communication delay constraint between them. This approach not only determines the schedulability of an IMA system, but also guides the resource allocations in IMA systems.

4. Case analysis

In this section, we illustrate the effectiveness of the proposed solution with a practical example. The central maintenance system (CMS) of a transport aircraft is composed of five partitions including flying data acquisition, configuration information management, data upload and download, fault monitoring, and data record. The parameters of each partition are shown in Figure 4. Flying data acquisition partition must receive control commands from other partitions within 300 ms, while data record partition needs get state information of other partitions within 500 ms. There is no communication time limit between other partitions. Due to system safety, the flying data acquisition partition and data record partition cannot run on the same module.

There are 3 homogeneous modules in the system. The available memory space of each module is 10 MB and the maximum number of partitions can be hosted by each module is 3. The maximum communication delay between any two modules is 1 ms. With the models proposed in section 2, parameters of the system are described as follows:

(1) partition set: $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5\}$; parameters of partitions: $\theta_1 = \langle 30, 100, 4\rangle$, $\theta_2 = \langle 10, 100, 2\rangle$, $\theta_3 = \langle 20, 50, 3\rangle$, $\theta_4 = \langle 40, 200, 1\rangle$ and $\theta_5 = \langle 30, 150, 5\rangle$; Only partitions θ_1 and θ_5 are exclusive, then in the exclusive relationship array $\vec{\eta} = (\eta_{i,j})$, all elements except $\eta_{1,5}$ and $\eta_{5,1}$ are equal to 0. The maximum communication delays between partitions are shown in Table 1.

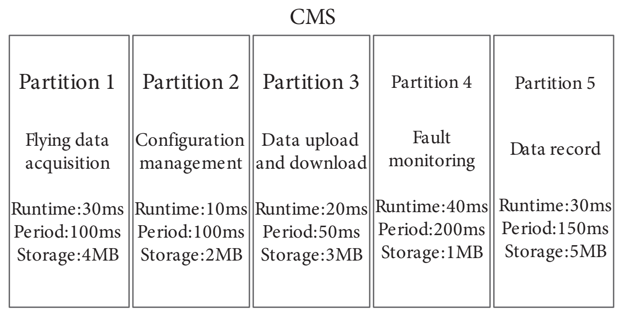

Figure 4. Partitions and its parameters of CMS.

Table 1. Maximum communication delay between partitions.

Partition no.	1	2	3	4	5
1	100	∞	∞	∞	500
2	300	100	∞	∞	500
3	300	∞	50	∞	500
4	300	∞	∞	200	500
5	300	∞	∞	∞	150

(2) module set: $\Pi = \{\pi_1, \pi_2, \pi_3\}$; attributes of module: $\forall k \in [1,3], N_k = 3, M_k = 10$; communication delay between modules $\vec{\lambda} = (\lambda_{k,l})$ is described in two different cases: communication delay between different modules is 1 ms, i.e. $\forall k,l \in [1,3], \lambda_{k,l} = 1$; communication delay on the same module is zero, i.e. $\forall k \in [1,3], \lambda_{k,k} = 0$.

Putting the above parameters of modules and partitions into the solution presented in Section 2.4 and solving it with Cplex Optimizer programming solver we get that the minimum number of modules required is 2, i.e. $m' = 2$, and the values of partition offset vector $\vec{s} = [5, 8, 0, 0, 2]$. The partition attribution vector $\vec{a}$ is shown in Table 2.

Table 2. Partition attribution vector form MILP.

Module no. / Partition no.	1	2	3
1	0	1	0
2	0	1	0
3	1	0	0
4	0	1	0
5	1	0	0

From Table 2 we can find that partitions θ_3 and θ_5 are allocated to the module π_1, while partitions θ_1, θ_2, and θ_4 are assigned to module π_2. As this IMA system only needs two modules to run all partitions, this system is schedulable. According to partition offset vector $\vec{s}$ and partition assignment vector $\vec{a}$, obtained from the MILP solution, the main time frames on modules π_1 and π_2 are shown in (a) and (b) of Figure 5.

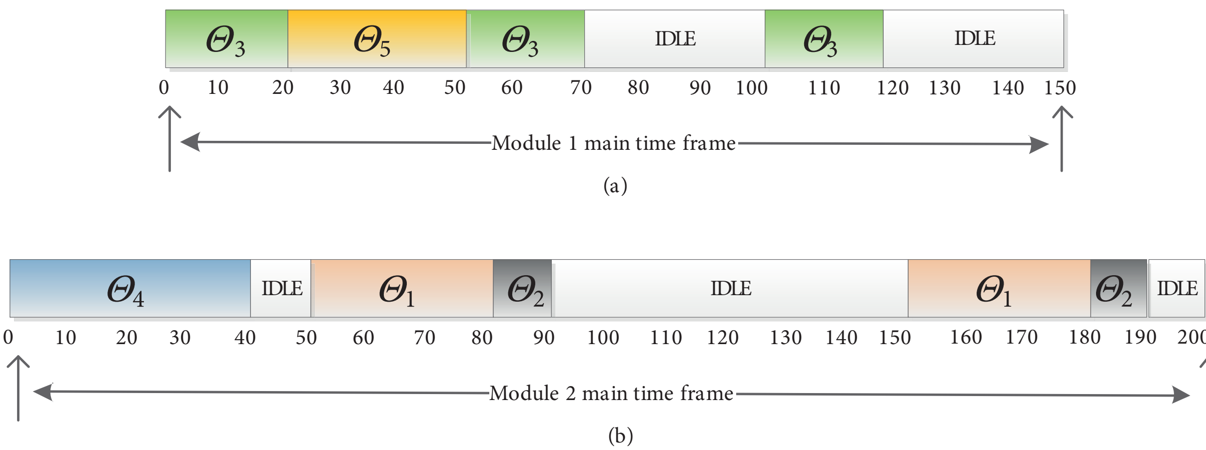

Figure 5. The main time frames of CMS.

5. Conclusion

In this paper, we first build a module model and partition model with a strict period, and analyze the constraints in space, time, and communication delay when IMA systems are schedulable. Then based on MILP formulation, we propose a solution to search all available space and determine whether all partitions are schedulable on a limited number of modules. Besides dealing with the space constraint, which has been the sole concern of many previous solutions, our approach handles the time and commutation requirements of the partitions of IMA systems. Our approach determines the allocation of modules and time windows to partitions and the schedule for communication delay between each two partitions. A practical example shows that the solution proposed not only provides a determination of the schedulability of an IMA system, but also achieves the minimum number of modules required by all partitions, which reduces the system weight and power consumption of the system, and provides a way of allocating resources for IMA systems.

References

[1] Xiong HG, Zhou GR, Li Q. A survey on avionics bus and network interconnections and their progress. Acta Aeronautica et Astronautica Sinica 2006; 27: 1135-1144.

[2] Chen J, Du C. Schedulability analysis for independent partitions in integrated modular avionics systems. In: Proceedings of the IEEE International Conference on Progress in Informatics and Computing; 18–20 Dec 2015; Nangjing, China. New York, NY, USA: IEEE. pp. 521-525.

[3] Baruah SK, Chakraborty S. Schedulability analysis of non-preemptive recurring real-time tasks. In: Proceedings of the Parallel and Distributed Processing Symposium; 25–29 April 2006; Rhodes Island, Greece. New York, NY, USA: IEEE. pp. 8-15.

[4] Chen J, Du C, Xie F, Yang Z. Schedulability analysis of non-preemptive strictly periodic tasks in multi-core real-time systems. Real-Time Systems 2016; 52: 239-271.

[5] Kim JE, Yoon MK, Im S, Bradford R, Sha L. Multi-IMA partition scheduling with synchronized solo-partitions for multi-core avionics systems. Department of Computer Science, University of Illinois at Urbana-Champaign, Technical report, May 2012.

[6] Korst J, Aarts E, Lenstra JK, Wessels J. Periodic Multiprocessor Scheduling. Berlin, Germany: Springer, 1991.

[7] Kermia O, Sorel Y. Schedulability analysis for non-preemptive tasks under strict periodicity constraints. In: Embedded and Real-Time Computing Systems and Applications; 25–27 August 2008; Kao-hsiung shih, Taiwan. IEEE. pp. 25-32.

[8] Eisenbrand F, Hähnle N, Niemeier M, Skutella M, Verschae J, Wiese A. Scheduling periodic tasks in a hard real-time environment. In: Proceedings of the 37th International Colloquium Conference on Automata, Languages and Programming. ICALP'10; 6–10 July 2010; Bordeaux, France. IEEE. pp. 299-311.

[9] Eisenbrand F, Kesavan K, Mattikalli R. Solving an avionics real-time scheduling problem by advanced IP-methods. In: de Berg M, Meyer U, editors. Algorithms. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2010; 21: 11-22.

[10] Marouf M, Sorel Y. Scheduling non-preemptive hard real-time tasks with strict periods. In: 2011 IEEE Emerging Technologies Factory Automation (ETFA); 5–9 Sep 2011; Toulouse, France. IEEE. pp. 1-8.

[11] Chen J, Du C, Han P. Scheduling independent partitions in integrated modular avionics systems. Plos One 2016; 11: 20-36.

[12] Sheikh AA, Brun O, Hladik PE. Strictly periodic scheduling in IMA-based architectures. Real-Time Systems 2012; 48: 359-386.

[13] Pira C, Artigues C. An efficient best response heuristic for a non-preemptive strictly periodic scheduling problem. Learning and Intelligent Optimization. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2013; 20: 281-287.

[14] Pira C, Artigues C. Line search method for solving a non-preemptive strictly periodic scheduling problem. Journal of Scheduling 2016; 19: 1-17.

[15] Sheikh AA, Brun O, Hladik PE. Partition scheduling on an IMA platform with strict periodicity and communication delays. In: Proceedings of the 18th International Conference on Real-Time and Network Systems; 4–5 Sep 2011; Toulouse, France. IEEE. pp. 179-188.

[16] Al Sheikh A, Brun O, Hladik PE, Prabhu BJ. A best-response algorithm for multiprocessor periodic scheduling. In: 23rd Euromicro Conference on Real-Time Systems (ECRTS); 6–8 July 2011; Porto, Portugal. IEEE. pp. 228-237.

[17] Linderoth JT, Lodi A, Cochran J. MILP Software. Wiley Encyclopedia of Operations Research and Management Science. Hoboken, NJ, USA: Wiley, 2010.

[18] Hann M, Deredempt MH, Cortier A. Requirements baseline for integrated modular avionics for space separation kernel qualification. In: Proceedings of the ESA Special Publication; 23–27 March 2015; Bergen, Norway. IEEE. pp. 732-739.

[19] Chen J, Du C, Xie F, Lin B. Allocation and scheduling of strictly periodic tasks in multi-core real-time systems. In: Proceedings of the IEEE International Conference on Embedded and Real-Time Computing Systems and Applications; 17–19 Aug 2016; Daegu, South Korea. IEEE. pp. 130-138.

[20] Burer S, Saxena A. The MILP Road to MIQCP. New York, NY, USA: Springer, 2012.