Efficient AUTOSAR-Compliant CAN-FD Frame Packing with Observed Optimality

Wenhong Ma[1], Guoqi Xie[1], Renfa Li[1], Weichen Liu[2], Hai (Helen) Li[3], Wanli Chang[1,4]
[1]Key Laboratory for Embedded and Cyber-Physical Systems of Hunan Province, Hunan University, China
[2] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[3] Department of Electrical and Computer Engineering, Duke University, USA
[4] Department of Computer Science, University of York, UK
{wenhongma, xgqman, lirenfa}@hnu.edu.cn, liu@ntu.edu.sg, hai.li@duke.edu, wanli.chang@york.ac.uk

***Abstract*—With the trend towards automated driving, Controller Area Network (CAN) is migrating to CAN with Flexible Data-Rate (CAN-FD), where frame packing (i.e., packing signals of various periods, deadlines, and payloads into frames following the standard CAN-FD format) is critical to address the high bandwidth demand with limited resources. Existing works have applied Integer Linear Programming (ILP), which easily gets intractable as the number of signals to be packed increases, or proposed heuristics, which are not able to obtain the optimal solution. In addition, the security model employed does not meet the AUTOSAR SecOC specification. This paper reports a novel frame packing approach for CAN-FD with an AUTOSAR-compliant security model. We establish the theory that extending the existing frame to pack signals with the same period leads to shorter WCTT (worst-case transmission time) and thus lower bus utilization compared to creating a new frame. Following this principle, the design space is tremendously pruned. As shown in the comprehensive experiments, only 10^{-9} of the original size or even a smaller portion needs to be explored, while the optimality is kept. The computational time is correspondingly reduced, generating solutions within 15 minutes to large-scale problems that are otherwise intractable with ILP.**

I. Introduction

As the automotive industry is moving towards automated driving, the conventional in-vehicle communication protocol Controller Area Network (CAN) with a maximum payload of 8 bytes in each frame and the maximum overall bandwidth of 1 Mbps is insufficient to handle the complexity. Therefore, CAN with Flexible Data-Rate (CAN-FD), having a payload of up to 64 bytes in each frame and the bandwidth of up to 12 Mbps, has been proposed and adopted as the new generation of CAN technology [1].

For CAN-FD, packing signals that may have different periods, deadlines, and payloads into frames with the standard format and various attributes (a.k.a. frame packing or signal packing) is critical to fully utilize the capacity of the bus. Existing works have tackled this NP-hard frame packing problem with ILP (integer linear programming) or MILP (mixed-ILP), depending on variables introduced in constraints linearization, and heuristics [2]–[5]. While ILP is able to compute the optimal solution with the minimum bus utilization, large-scale problems (such as with 200 to 300 signals that are still practically relevant) are intractable due to its exhaustive nature. Heuristics can be fast, yet lose the optimality.

In addition, none of the works in the literature meet the AUTOSAR SecOC specification [6], which require MAC (message authentication code) and FV (freshness value) in each frame to be truncated to different length for satisfying different needs. The symmetric authentication approach is taken in [7], which adds fixed 4 bytes (including MAC and FV) to each frame for combating masquerade and replay attacks. However, this fixed size is too optimistic (short) for security-critical frames and too pessimistic (long) for non-security-critical frames.

In this paper, we propose a novel frame packing approach for CAN-FD with an AUTOSAR-compliant security model to minimize the bus utilization under the timing and security constraints. We prove that extending the existing frame to pack signals with the same period results in shorter WCTT (worst-case transmission time) and lower bus utilization compared to creating a new frame. This principle limits the number of frames to be considered for frame packing and tremendously prunes the design space. Comprehensive experiments show that only 10^{-9} or even a smaller portion of the original size needs to be explored, and that the optimality is kept. The computational time is correspondingly reduced. When there are 100 signals, it takes ILP 1277 seconds and our approach 1.78 seconds on average. When it is 200, which is still practically relevant, ILP starts to fail in some cases. For 300 signals, which is a very large number in real applications, it takes our approach less than 15 minutes, and ILP cannot generate a solution within 5 days.

II. Models and Problem Statement

We consider a common architecture as shown in Figure 1. All electronic control units (ECUs) are connected to a CAN-FD bus, which is half-duplex and non-preemptive. The entire system is modeled by $\{E, S, M\}$. The set of ECUs is denoted as $E = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{|E|}\}$, where $|E|$ is the size of E and thus the number of ECUs. The signals $S = \{S_1, S_2, \ldots, S_{|E|}\}$ are periodically activated from the ECUs, where S_k represents the set of signals generated by the source ECU ε_k. In each $S_k =$

This work is supported by the National Natural Science Foundation of China (Grants 61932010, 61672217, 61702172 and 61972139) and the Open Research Project of Electronic Information and Control of Fujian University Engineering Research Center, Minjiang University.

Corresponding Authors: Guoqi Xie, Wanli Chang

978-3-9819263-5-4/DATE21/©2021 EDAA

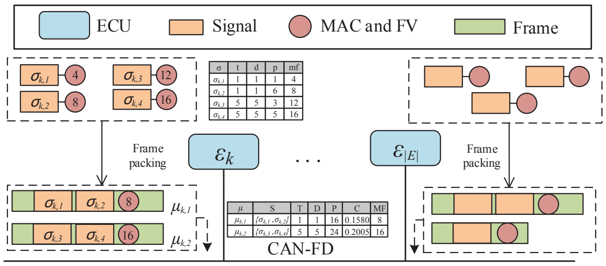

Fig. 1: System architecture of CAN-FD frame packing

$\{\sigma_{k,1}, \sigma_{k,2}, \ldots, \sigma_{k,|S_k|}\}$, $\sigma_{k,i}$ denotes the ith signal in ε_k and has the attributes of $\{t(\sigma_{k,i}), d(\sigma_{k,i}), p(\sigma_{k,i})\}$, representing the period, deadline, and payload of the signal $\sigma_{k,i}$, respectively. We do not consider signals with the same source and destination ECU as they do not use the bus.

The signals transmitted on the bus need to be packed into frames with the format specified by CAN-FD. The set of frames is denoted by $M = \{M_1, M_2, \ldots, M_{|E|}\}$, where M_k is the set of frames generated by the source ECU ε_k. In each $M_k = \{\mu_{k,1}, \mu_{k,2}, \ldots, \mu_{k,|M_k|}\}$, $\mu_{k,j}$ represents the jth frame in ε_k and has the attributes of $\{S(\mu_{k,j}), T(\mu_{k,j}), D(\mu_{k,j}), P(\mu_{k,j}), C(\mu_{k,j})\}$, where $S(\mu_{k,j})$ is the set of signals packed into $\mu_{k,j}$. The other four attributes are period, deadline, payload, and WCTT in order, and can be computed based on $S(\mu_{k,j})$.

The period of $\mu_{k,j}$ is the greatest common divisor of the periods for the signals in $S(\mu_{k,j})$, i.e.,

$$T(\mu_{k,j}) = \gcd\{t(\sigma_{k,i}) | \sigma_{k,i} \in S(\mu_{k,j})\}. \tag{1}$$

The signals packed into one frame must have harmonic periods. The deadline of $\mu_{k,j}$ is the shortest deadline of the signals in $S(\mu_{k,j})$, i.e.,

$$D(\mu_{k,j}) = \min\{d(\sigma_{k,i}) | \sigma_{k,i} \in S(\mu_{k,j})\}. \tag{2}$$

The payload of $\mu_{k,j}$ needs to be large enough to contain all its constituent signals, i.e.,

$$P(\mu_{k,j}) \geq \sum_{\sigma_{k,i} \in S(\mu_{k,j})} p(\sigma_{k,i}), \tag{3}$$

and it must be equal to one of the following integers as specified by CAN-FD: 0 to 8, 12, 16, 20, 24, 32, 48, and 64 bytes. The CAN-FD frame format is shown in Figure 2, where the payload is placed in the data field. The WCTT of the frame $C(\mu_{k,j})$ is equal to [8]

$$C(\mu_{k,j}) = 32\mathrm{t_a} + \left(28 + 5\left\lceil \frac{P(\mu_{k,j}) - 16}{64} \right\rceil + 10P(\mu_{k,j})\right)\mathrm{t_d}, \tag{4}$$

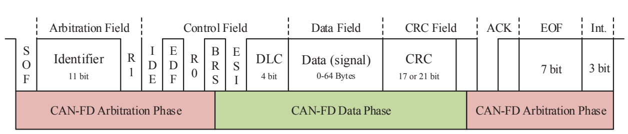

Fig. 2: The standard CAN-FD frame format

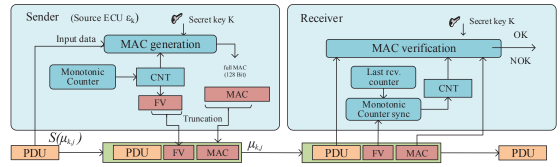

Fig. 3: Message authentication and freshness verification

where $\mathrm{t_a}$ and $\mathrm{t_d}$ are the time required in the arbitration phase and the data phase (both shown in Figure 2) to transmit one bit, respectively. Similar to [3], [7], [8], we have $\mathrm{t_a} = 2\mu$s (i.e., 500 kbps arbitration bit rate) and $\mathrm{t_d} = 0.5\mu$s (i.e., 2Mbps data bit rate).

Security model: In this study, we use the security authentication mechanism provided by the SecOC [6] module in the AUTOSAR specification, which employs symmetric authentication approaches of MAC and FV for identity authentication and integrity protection of sensitive data, against masquerade and replay attacks. As shown in Figure 3, on the sender side (source ECU ε_k), the Protocol Data Unit (PDU, sharing the same concept as signals in a frame and hence also denoted by $S(\mu_{k,j})$) and its FV are used as input data to generate the MAC with 128 bits. AUTOSAR SecOC allows the MAC and FV to be truncated to varying length, which are filled in the data field of the frame together with the PDU. On the receiver side, the ECU uses the symmetric key k to verify whether the MAC and FV are consistent.

The minimum length of the MAC and FV $mf(\sigma_{k,i})$ to meet the security requirement of a signal $\sigma_{k,i}$ depends on its criticality and sensitivity. For a frame $\mu_{k,j}$, the MAC and FV length $MF(\mu_{k,j})$ must be large enough for every internal signal, i.e.,

$$\forall \sigma_{k,i} \in S(\mu_{k,j}), \;\; MF(\mu_{k,j}) \geq mf(\sigma_{k,i}). \tag{5}$$

We can take the equality and get,

$$MF(\mu_{k,j}) = \max\{mf(\sigma_{k,i}) | \sigma_{k,i} \in S(\mu_{k,j})\}. \tag{6}$$

The payload of the frame $P(\mu_{k,j})$ can then be updated from (3) to the following exact form,

$$\begin{cases} Z(\mu_{k,j}), & 0 \leq Z(\mu_{k,j}) \leq 8; \\ \lceil \frac{Z(\mu_{k,j})-8}{4} \rceil \times 4 + 8, & 8 \leq Z(\mu_{k,j}) \leq 24; \\ \lceil \frac{Z(\mu_{k,j})-24}{8} \rceil \times 8 + 24, & 24 \leq Z(\mu_{k,j}) \leq 32; \\ \lceil \frac{Z(\mu_{k,j})-32}{16} \rceil \times 16 + 32, & 32 \leq Z(\mu_{k,j}) \leq 64, \end{cases} \tag{7}$$

where

$$Z(\mu_{k,j}) = MF(\mu_{k,j}) + \sum_{\sigma_{k,i} \in S(\mu_{k,j})} p(\sigma_{k,i}). \tag{8}$$

Problem statement: Frame packing on CAN-FD is creating a certain number of frames and allocating a given set of signals to them, with the aim of minimizing the bus utilization, i.e.,

$$Minimize \sum_{k=1}^{|E|} \sum_{j=1}^{|M|} U(\mu_{k,j}), \tag{9}$$

where $U(\mu_{k,j})$ is the utilization of a frame,

$$U(\mu_{k,j}) = \frac{C(\mu_{k,j})}{T(\mu_{k,j})}. \quad (10)$$

Specifically, it needs to be determined which signals share one frame and what are the attributes of this frame. There are constraints to be respected on timing, security, and CAN-FD specification. We will report the design space pruning in Section III and present the mathematical formulation with its solution approach in Section IV.

III. Design Space Pruning

The existing works applying ILP/MILP to CAN-FD frame packing (such as [3], [4], [7]) initialize the design space with the number of frames equal to the number of signals, and each frame corresponding to a signal of the same period. None of them have provided a proof that such a design space contains the optimal packing solution. We will provide one below. Given a set of signals S_k from an ECU ε_k, we let T_k be the set of periods for S_k in the ascending order, i.e.,

$$T_{k,1} \leq T_{k,2} \leq \ldots \leq T_{k,|T_k|}. \quad (11)$$

The set of frames is M_k as denoted in Section II.

Theorem 1: For a set of signals S_k with its set of periods T_k, the number of frames with any period $T_{k,l}$ in the optimal packing solution is smaller than or equal to the number of signals with the same period $T_{k,l}$ in S_k, i.e.,

$$\left|M^*_{k,l}\right| \leq |S_{k,l}|, \quad (12)$$

where * denotes optimality.

Proof 1: This theorem can be proven by contradiction. If for a certain period $T_{k,l}$, the number of frames $\left|M^*_{k,l}\right|$ is larger than the number of signals $|S_{k,l}|$, there must exist a frame $\mu_{k,j}$ containing only signals with periods larger than $T_{k,l}$ (note that a smaller-period signal cannot be packed into a larger-period frame, referring to (1)). In this case, increasing the period of $\mu_{k,j}$ to equal the smallest period of its signals will decrease the bus utilization according to (9) and (10) without violating any constraints, which contradicts that this solution is the optimal with the minimum bus utilization. Hence, Theorem 1 is proven.

Similarly, in the optimal packing, there is no frame with a period that is different from all periods in T_k. Otherwise, the bus utilization can be reduced when this period is increased to the smallest period of all signals in the frame. Therefore, the design space discussed at the beginning of this section contains the optimal solution.

The main issue with such a design space is the scalability. For a given S_k, the number of design points to explore with ILP is $|S_k|^{|S_k|}$, which easily gets intractable. In this work, we shrink the design space before exploring it. More specifically, we reduce the number of frames for every period to $\left|M'_{k,l}\right|$, which results in a much smaller set of frames denoted as M'_k, while trying to keep the optimal solution. Below we will first lay the theoretical foundations and then present the pruning algorithm.

We consider a basic scenario that a new signal σ_i is to be packed to a frame with the same period. The question is whether packing it to an existing frame μ_{exist} or creating a new frame μ_{new} to accommodate it is more beneficial for the bus utilization. As the period is the same, the WCTT can be evaluated and compared, referring to (10). The WCTT of the extended frame C_{extend} can be computed by (4) with the payload of P_{extend}. If a new frame is created, the WCTT is,

$$\begin{aligned} C_{\text{exist}} + C_{\text{new}} = 64\text{t}_\text{a} + \Bigg(56 + 5\left\lceil\frac{P_{\text{exist}} - 16}{64}\right\rceil \\ + 5\left\lceil\frac{P_{\text{new}} - 16}{64}\right\rceil + 10(P_{\text{exist}} + P_{\text{new}})\Bigg)\text{t}_\text{d}, \end{aligned} \quad (13)$$

where P_{exist} and P_{new} are the payload of the existing and new frame, respectively.

We abuse the floor function $\lfloor\ \rfloor$ and define,

$$\alpha = P_{\text{extend}} - \lfloor P_{\text{extend}} \rfloor - 1, \quad (14)$$

where $\lfloor P_{\text{extend}} \rfloor$ denotes the next smaller payload of P_{extend} according to the frame payload sequence in the CAN-FD specification (i.e., 1-8, 12, 16, 20, 24, 32, 48, 64). For instance, $\lfloor 12 \rfloor = 8$, $\lfloor 64 \rfloor = 48$, and $\lfloor 4 \rfloor = 3$. Below we use Lemma 1 to show the relationship between P_{exist}, P_{new}, and P_{extend}, and apply it to prove Theorem 2.

Lemma 1: There is always

$$P_{\text{exist}} + P_{\text{new}} \geq P_{\text{extend}} - \alpha. \quad (15)$$

Proof 2: From (7), we have

$$P_{\text{exist}} + P_{\text{new}} \geq Z_{\text{exist}} + Z_{\text{new}}, \quad (16)$$

and with (8),

$$\begin{aligned} Z_{\text{exist}} + Z_{\text{new}} = &\boldsymbol{MF}_{\text{exist}} + \sum_{\sigma_{i'} \in S(\mu_{\text{exist}})} p(\sigma_{i'}) \\ &+ \boldsymbol{MF}_{\text{new}} + p(\sigma_i), \end{aligned} \quad (17)$$

For the extended frame,

$$\begin{aligned} Z_{\text{extend}} = &\max\{\boldsymbol{MF}_{\text{exist}}, \boldsymbol{MF}_{\text{new}}\} \\ &+ \sum_{\sigma_{i'} \in S(\mu_{\text{exist}})} p(\sigma_{i'}) + p(\sigma_i). \end{aligned} \quad (18)$$

Therefore,

$$P_{\text{exist}} + P_{\text{new}} \geq Z_{\text{exist}} + Z_{\text{new}} \geq Z_{\text{extend}}. \quad (19)$$

Following (7) and the abused floor function,

$$Z_{\text{extend}} \geq \lfloor P_{\text{extend}} \rfloor + 1. \quad (20)$$

With the above two equations and (14),

$$P_{\text{exist}} + P_{\text{new}} \geq \lfloor P_{\text{extend}} \rfloor + 1 = P_{\text{extend}} - \alpha. \quad (21)$$

Hence, the lemma is proven.

Theorem 2: When packing a signal set $S_{k,l}$ to a frame set $M'_{k,l}$ with the same period $T_{k,l}$, if $\text{t}_\text{a} \geq 4\text{t}_\text{d}$ (which is the case for CAN-FD), then extending an existing frame always leads to better bus utilization than creating a new frame to accommodate a new signal.

Proof 3: Since the period is the same, we just need to consider the WCTT and prove that, when $\mathrm{t_a} \geq 4\mathrm{t_d}$, $C_{\text{exist}} + C_{\text{new}} > C_{\text{extend}}$ is always true. To be convenient, we let $\Delta C = C_{\text{exist}} + C_{\text{new}} - C_{\text{extend}}$. There are in total five cases: (i) $P_{\text{extend}} \leq 16$, which means that $P_{\text{exist}} \leq 16$ and $P_{\text{new}} \leq 16$; (ii) $P_{\text{extend}} > 16$, $P_{\text{exist}} \leq 16$, and $P_{\text{new}} \leq 16$; (iii) $P_{\text{extend}} > 16$, $P_{\text{exist}} > 16$, and $P_{\text{new}} \leq 16$; (iv) $P_{\text{extend}} > 16$, $P_{\text{exist}} \leq 16$, and $P_{\text{new}} > 16$; (v) $P_{\text{extend}} > 16$, $P_{\text{exist}} > 16$, and $P_{\text{new}} > 16$.

Case I: Since the payload is not larger than 16, there is $P_{\text{extend}} - 3 \leq P_{\text{exist}} + P_{\text{new}}$ according to (14) and (15). In this case,

$$\begin{aligned}\Delta C &= 64\mathrm{t_a} + \Big(56 + 10(P_{\text{exist}} + P_{\text{new}})\Big)\mathrm{t_d} \\ &\quad - \Big(32\mathrm{t_a} + (28 + 10P_{\text{extend}})\mathrm{t_d}\Big) \\ &= 32\mathrm{t_a} + 28\mathrm{t_d} + 10(P_{\text{exist}} + P_{\text{new}} - P_{\text{extend}})\mathrm{t_d} \\ &\geq 32\mathrm{t_a} - 2\mathrm{t_d} > 0.\end{aligned} \tag{22}$$

Case II: Since $P_{\text{exist}} \leq 16$ and $P_{\text{new}} \leq 16$, there is $P_{\text{extend}} \leq 32$, and thus $P_{\text{extend}} - 7 \leq P_{\text{exist}} + P_{\text{new}}$. We have,

$$\begin{aligned}\Delta C &= 64\mathrm{t_a} + \Big(56 + 10(P_{\text{exist}} + P_{\text{new}})\Big)\mathrm{t_d} \\ &\quad - \Big(32\mathrm{t_a} + (33 + 10P_{\text{extend}})\mathrm{t_d}\Big) \\ &= 32\mathrm{t_a} + 23\mathrm{t_d} + 10(P_{\text{exist}} + P_{\text{new}} - P_{\text{extend}})\mathrm{t_d} \\ &\geq 32\mathrm{t_a} - 47\mathrm{t_d} > 0.\end{aligned} \tag{23}$$

Case III: Since $P_{\text{extend}} > 16$, $P_{\text{exist}} + P_{\text{new}} - P_{\text{extend}} \geq -15$. Therefore,

$$\begin{aligned}\Delta C &= 64\mathrm{t_a} + \Big(61 + 10(P_{\text{exist}} + P_{\text{new}})\Big)\mathrm{t_d} \\ &\quad - \Big(32\mathrm{t_a} + (33 + 10P_{\text{extend}})\mathrm{t_d}\Big) \\ &= 32\mathrm{t_a} + 28\mathrm{t_d} + 10(P_{\text{exist}} + P_{\text{new}} - P_{\text{extend}})\mathrm{t_d} \\ &\geq 32\mathrm{t_a} - 122\mathrm{t_d} > 0\end{aligned} \tag{24}$$

Case IV: The calculation is the same as **Case III**.

Case V: Similar to **Case III**, there is $P_{\text{exist}} + P_{\text{new}} - P_{\text{extend}} \geq -15$. Then,

$$\begin{aligned}\Delta C &= 64\mathrm{t_a} + \Big(66 + 10(P_{\text{exist}} + P_{\text{new}})\Big)\mathrm{t_d} \\ &\quad - \Big(32\mathrm{t_a} + (33 + 10P_{\text{extend}})\mathrm{t_d}\Big) \\ &= 32\mathrm{t_a} + 33\mathrm{t_d} + 10(P_{\text{exist}} + P_{\text{new}} - P_{\text{extend}})\mathrm{t_d} \\ &\geq 32\mathrm{t_a} - 117\mathrm{t_d} > 0\end{aligned} \tag{25}$$

In all the above five cases, there is $C_{\text{exist}} + C_{\text{new}} > C_{\text{extend}}$. Hence, the theorem is proven.

Similar to the discussion after Theorem 1, when all the signals in $S_{k,l}$ have been packed to $M'_{k,l}$ with the same period, creating a new frame of the period $T_{k,l}$ to accommodate longer-period signals does not benefit the bus utilization, since this period $T_{k,l}$ can be increased to the smallest period of all signals in this frame to reduce the bus utilization.

These theoretical findings essentially indicate that if there is a frame set $M'_{k,l}$ sufficient to pack all signals in $S_{k,l}$, the optimal solution unlikely requires more frames than $\left|M'_{k,l}\right|$ with the period $T_{k,l}$. Guided by this, we propose a simple pruning algorithm with polynomial time complexity, which estimates the number of frames required for each period by considering only the signals of the same period, and uses it to construct the design space. This tremendously expedites the exploration. It is to be noted that the optimal solution is highly probably kept in the pruned design space, there is no guarantee though. We have performed comprehensive experiments, where in every case, the optimal packing solution is found. In addition, we observe that there are often multiple well separated design points having the minimum (or very close) bus utilization. Therefore, there is strong confidence that the design space pruning does not compromise the optimality.

Algorithm 1 The pruning algorithm

Input: S_k
Output: M'_k
1: Define the frame list $M'_k \leftarrow$ NULLSET;
2: Obtain T_k following (11);
3: **for** $T_{k,l} \in T_k$ **do**
4: Sort $S_{k,l}$ in $signal\_list$ by the descending order of payload;
5: $M'_{k,l} \leftarrow \{\mu_{k,l,1}\}$;
6: **for** $\sigma_{k,i} \in signal\_list$ in the descending order of payload **do**
7: $isNewFrame \leftarrow True$;
8: **for** $\mu_{k,l,j} \in M'_{k,l}$ (looping j from 1 onward) **do**
9: Creating a superficial frame μ_s including $\sigma_{k,i}$ and $S(\mu k, l, j)$;
10: Calculate $Z(\mu_s)$ with (6) and (8);
11: **if** $Z(\mu_s) \leq 64$ **then**
12: Pack $\sigma_{k,i}$ into $\mu_{k,l,j}$;
13: $isNewFrame \leftarrow False$;
14: break;
15: **end if**
16: **end for**
17: **if** ($isNewFrame$) **then**
18: Pack $\sigma_{k,i}$ into a new frame and attach the frame at the end of $M'_{k,l}$;
19: **end if**
20: **end for**
21: Add $M'_{k,l}$ to M'_k;
22: **end for**
23: **return** M'_k

Algorithm 1 describes the pruning process. Essentially, for every period $T_{k,l}$ (the outermost loop from Line 3 to Line 22), the signals are packed in the descending order of payload (sorted in Line 4). For each signal $\sigma_{k,i}$ (the loop from Line 6 to Line 20), it tries to enter the existing frames (the innermost loop from Line 8 to Line 16). If the signal $\sigma_{k,i}$ cannot be packed into any existing frame, a new frame is created to hold it (Line 17 to Line 19). Algorithm 1 consists of three loops, where the combination of the outer two loops every signal in S_k, and the innermost one loops the existing frames. Therefore, the time complexity is $\mathcal{O}(n^2)$.

There are alternatives to Algorithm 1 in estimating the number of frames required for each period and construct the design space. As long as they result in feasible packing solutions and sufficiently reduce the number of frames, the performance should be reasonably good. It is not necessary to pursue the minimum possible number of frames, especially if simplicity has to be compromised.

IV. Problem Formulation and Solution Approach

With the design space pruned, we formulate an ILP problem and deploy a Gurobi solver. Similar to (9), the objective function is,

$$Minimize \sum_{j=1}^{|M'_k|} \frac{C(\mu_{k,j})}{T(\mu_{k,j})}, \tag{26}$$

where $T(\mu_{k,j})$ is known and $C(\mu_{k,j})$ depends on $P(\mu_{k,j})$ according to (4). The linearization of $P(\mu_{k,j})$ will be shown below while discussing the constraint on the frame payload specification.

The binary decision variables are $x_{i,j}$, which indicate whether the signal $\sigma_{k,i}$ is packed into the frame $\mu_{k,j}$ (1 for yes and 0 for no). There are multiple constraints to be satisfied. First, all the signals in a frame must come from the same ECU. In this work, we solve the frame packing for one ECU ε_k, which can be trivially extended to cover more ECUs. Therefore, this constraint is directly respected. Second, each signal is only packed into one frame, i.e.,

$$\forall i, \sum_{j=1}^{|M'_k|} x_{i,j} = 1. \tag{27}$$

The third constraint is that every signal must meet its security requirement. Following (5) (not (6) as it is not linear),

$$\forall \sigma_{k,i} \in S_k, \mu_{k,j} \in M'_k, \ \ MF(\mu_{k,j}) \geq mf(\sigma_{k,i}) \cdot x_{i,j}. \tag{28}$$

Fourth, all periods for a frame, including those of the signals in the frame and that of the frame, must be harmonic. We examine all the signals and frames. If the period of a signal $\sigma_{k,i}$ and the period of a frame $\mu_{k,j}$ are not in harmony, a constraint $x_{i,j} = 0$ is added. If the periods of two signals σ_{k,i_1} and σ_{k,i_2} are not in harmony, a constraint $x_{i_1,j} + x_{i_2,j} \leq 1$ is added.

The CAN-FD frame payload specification must be respected. We first use (8) to obtain,

$$Z(\mu_{k,j}) = MF(\mu_{k,j}) + \sum_{i=1}^{|S_k|} x_{i,j} \cdot p(\sigma_{k,i}), \tag{29}$$

and then (7) is linearized to

$$\begin{aligned} P(\mu_{k,j}) &= \lambda_{j,1} \cdot Z(\mu_{k,j}) + 12\lambda_{j,2} + 16\lambda_{j,3} \\ &+ 20\lambda_{j,4} + 24\lambda_{j,5} + 32\lambda_{j,6} + 48\lambda_{j,7} + 64\lambda_{j,8}, \end{aligned} \tag{30}$$

$$\sum_{h=1}^{8} \lambda_{j,h} = 1, \tag{31}$$

$$\begin{cases} Z(\mu_{k,j}) & \leq 8\lambda_{j,1} + 12\lambda_{j,2} + 16\lambda_{j,3} + 20\lambda_{j,4} \\ & +24\lambda_{j,5} + 32\lambda_{j,6} + 48\lambda_{j,7} + 64\lambda_{j,8} \\ Z(\mu_{k,j}) & \geq 1\lambda_{j,1} + 9\lambda_{j,2} + 13\lambda_{j,3} + 17\lambda_{j,4} \\ & +21\lambda_{j,5} + 25\lambda_{j,6} + 33\lambda_{j,7} + 49\lambda_{j,8}, \end{cases} \tag{32}$$

where $\lambda_{j,h}$ is a binary variable indicating the range of $Z(m_{k,j})$ and the value of $P(\mu_{k,j})$ in (7). For example, $\lambda_{j,8} = 1$ means that $49 \leq Z(m_{k,j}) \leq 64$ and $P(\mu_{k,j}) = 64$. There is only one $\lambda_{j,h}$ equal to 1. As $\lambda_{j,1} \cdot Z(\mu_{k,j})$ is still not linear and needs further operation, we replace $\lambda_{j,1} \cdot Z(\mu_{k,j})$ with $Z'(\mu_{k,j})$ and let,

$$\begin{aligned} P(\mu_{k,j}) =& Z'(\mu_{k,j}) + 12\lambda_{j,2} + 16\lambda_{j,3} + 20\lambda_{j,4} \\ &+ 24\lambda_{j,5} + 32\lambda_{j,6} + 48\lambda_{j,7} + 64\lambda_{j,8}, \end{aligned} \tag{33}$$

$$\begin{cases} Z'(\mu_{k,j}) \leq Z(\mu_{k,j}) \\ Z'(\mu_{k,j}) \geq Z(\mu_{k,j}) - \mathrm{M} \cdot (1 - \lambda_{j,1}) \\ 0 \leq Z'(\mu_{k,j}) \leq \mathrm{M} \cdot \lambda_{j,1}, \end{cases} \tag{34}$$

where M is a large integer constant. In addition, the transmission of a frame must be completed before its deadline, i.e.,

$$C(\mu_{k,j}) \leq D(\mu_{k,j}), \tag{35}$$

where $C(\mu_{k,j})$ is computed from (4) and the ceiling function can be easily linearized with negligible loss of accuracy. There are several methods [3], [7] to extend this constraint for the worst-case response time (WCRT), where interference and blocking from other signals are taken into account.

V. Experimental Results

We generate a set of signals following the real-world automotive benchmarks [9], including the periods with their shares and the payloads with their shares, as illustrated in Table I. Those signals larger than 64 bytes are placed in the bin of 33-64 bytes [3]. The length of MAC and FV is distributed in the range of 0-16 bytes with an average of 4 bytes [7]. In the experiments, we generate a total of 800 signal sets with different sizes of 50, 80, 100, 120, 150, 200, 250, and 300 signals. There are 100 sets for each size. Every signal set is evenly distributed among three ECUs across a CAN-FD bus. Our experiments are conducted on a Windows 10 PC (AMD Ryzen 5 1500X C, 8 GB main memory), and the algorithm is implemented using Gurobi (9.0.0) and Python (3.7.1).

TABLE I: Signal parameters and their distribution

Period [ms]	Share [%]
1	4
2	3
5	3
10	31
20	31
50	3
100	20
200	1
1000	4

Size [Byte]	Share [%]
1	35
2	49
4	13
5-8	0.8
9-16	1.3
17-32	0.5
33-64	0.4

The design space pruning performance of Algorithm 1 is reported in Table II. When the number of signals is 50, on average across the 100 sets, the size of the design space is reduced from 50^{50} to 19.4^{50}, the latter number being 10^{-9} of the former. As the number of signals increases, the reduction is even more significant.

TABLE II: Design space pruning performance

Size	50	80	100	120	150	200	250	300
Original	50^{50}	80^{80}	100^{100}	120^{120}	150^{150}	200^{200}	250^{250}	300^{300}
Pruned	19.4^{50}	22.1^{80}	23.5^{100}	25^{120}	26.3^{150}	29^{200}	31.9^{250}	34.4^{300}

We compare our work with the state-of-the-art methods, including:

- **Best Fit** [10]: Sorting is performed in the descending order of the signal period. Best Fit is applied in the frame selection. This is essentially a greedy algorithm.

- **Next Fit** [11]: Similar to Best Fit, sorting is performed in the descending order of the signal period. Next Fit is applied in the frame selection. This is a lighter version of Best Fit, with potentially worse performance and faster execution.
- **SA** [5]: Simulate annealing (SA) can also be employed for CAN-FD frame packing. We modify [5], which is originally for CAN, to suit CAN-FD. These three methods are heuristics.
- **ILP** [3], [4], [7]: Depending on the variables introduced in constraints linearization, ILP [3] or MILP [4], [7] have been deployed under different scenarios, with very similar performances and computation time. For comparison, in this work, we apply ILP to our problem formulation with the unpruned design space.

Security constraints are added to all these state-of-the-art methods for fair comparison. The bus utilization and computation time are shown in Table III and IV, respectively. For each size of the signal set, the average results across the 100 sets are reported.

TABLE III: Comparison of the bus utilization under different sizes of signal sets

Size	Best Fit	Next Fit	SA	ILP	Ours
50	0.354263	0.354092	0.350058	0.345462	0.345462
80	0.471806	0.472982	0.474177	0.460002	0.460002
100	0.545058	0.547298	0.555685	0.531427	0.531427
120	0.619552	0.622150	0.641329	0.602861	0.602861
150	0.688118	0.693282	0.730258	0.670638	0.670638
200	0.787528	0.793914	0.866864	0.768096	0.768096
250	0.889953	0.902449	1.010229	—	0.869102
300	1.016660	1.031601	1.181216	—	0.985946

TABLE IV: Comparison of the computational time in seconds under different sizes of signal sets

Size	Best Fit	Next Fit	SA	ILP	Ours
50	0.34	0.2	501.03	3.98	0.52
80	0.45	0.34	798.39	550.29	1.17
100	0.55	0.37	990.89	1276.52	1.78
120	0.67	0.43	1198.16	2550.77	2.69
150	0.86	0.53	1514.01	6783.95	4.22
200	1.27	0.7	1956.9	6 hours	45.72
250	1.66	0.88	2433.1	>5 days	263.63
300	2.08	1.02	3038.6	>5 days	888.92

While applying ILP on the original design space produces the optimal solution, our approach performing pruning keeps the optimality in every case experimented on. When the size of the signal set is 200, which is still a practically relevant number, ILP fails to generate a solution in about 10% of the cases. When there are 300 signals (100 signals for one ECU), which is a very large-scale problem in real applications, ILP cannot generate any solution within 5 days, while our approach takes less than 15 minutes. For smaller sizes, we are also significantly advantageous in computational time, such as 4 seconds against nearly 2 hours when there are 150 signals. Among the heuristics, SA has worse performance than our approach and longer computational time. Best Fit and Next Fit are faster, yet lose the optimality, which is expected. The distribution of computational time for our approach under different sizes of signal sets (100 sets per size) is illustrated in Figure 4.

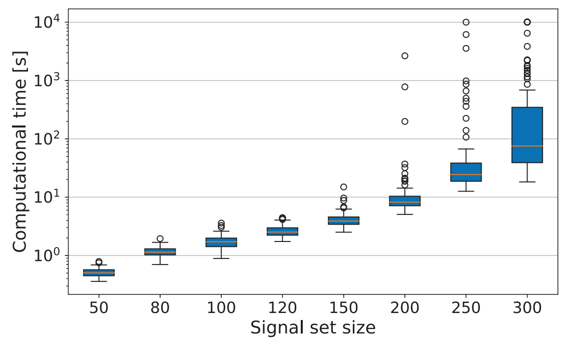

Fig. 4: The computational time distribution of our approach for different sizes of signal sets

VI. Conclusion

CAN-FD as the new generation of CAN technology with larger bandwidth is gaining ground in the revolution towards autonomous vehicles. Frame packing in CAN-FD is an NP-hard problem, and has been extensively investigated due to its significant impact on the bus utilization. However, the existing works are not satisfactory, being either easily intractable or away from the optimality. In this work, we propose a new approach, which is able to generate the optimal solution for all the cases in the comprehensive experiments. It takes less than 15 minutes for practically very large-scale problems on average. This paper is an important contribution to the automotive domain.

References

[1] *CAN with Flexible Data-Rate*, Robert Bosch GmbH, 2012, Specification Version 1.0.

[2] K. Sandstrom, C. Norstom, and M. Ahlmark, "Frame packing in real-time communication," in *RTCSA*, 2000.

[3] P. Joshi, H. Zeng, U. D. Bordoloi, S. Samii, S. S. Ravi, and S. K. Shukla, "The multi-domain frame packing problem for CAN-FD," in *ECRTS*, 2017.

[4] M. Di Natale, C. L. M. da Silva, and M. M. D. Santos, "On the applicability of an MILP solution for signal packing in CAN-FD," in *INDIN*, 2016.

[5] P. Pop, P. Eles, and Z. Peng, "Schedulability-driven frame packing for multicluster distributed embedded systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 4, no. 1, pp. 112–140, 2005.

[6] *Specification of Module Secure Onboard Communication*, AUTOSAR, 2015, Release 4.2.2.

[7] X. Yong, G. Zeng, R. Kurachi, H. Takada, and G. Xie, "Security/timing-aware design space exploration of CAN FD for automotive cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1094–1104, 2018.

[8] U. D. Bordoloi and S. Samii, "The frame packing problem for CAN-FD," in *RTSS*, 2014.

[9] S. Kramer, D. Ziegenbein, and A. Hamann, "Real world automotive benchmarks for free," in *WATERS*, 2015.

[10] R. Saket and N. Navet, "Frame packing algorithms for automotive applications," *Journal of Embedded Computing*, vol. 2, no. 1, pp. 93–102, 2006.

[11] F. Polzlbauer, I. Bate, and E. Brenner, "Optimized frame packing for embedded systems," *IEEE Embedded Systems Letters*, vol. 4, no. 3, pp. 65–68, 2012.