This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2019.2907921, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

Approaches for Assigning Offsets to Signals for Improving Frame Packing in CAN-FD

Prachi Joshi, S. S. Ravi, Qingyu Liu, Unmesh D. Bordoloi, Soheil Samii, Sandeep Shukla and Haibo Zeng

***Abstract*—Controller Area Network (CAN) is a widely used protocol that allows communication among Electronic Control Units (ECUs) in automotive electronics. It was extended to CAN-FD (CAN with Flexible Data-rate) to meet the increasing demand for bandwidth generated by the growing number of features in modern automobiles. The signal-to-frame packing problem has been studied in literature for both CAN and CAN-FD. In this work, we propose and formulate the *signal offset assignment problem* (SOAP) in CAN-FD to improve the bus utilization during frame packing. We propose two algorithmic themes to solve SOAP and establish their worst case performance guarantees. The first is a general approximation framework (GAF) which can use any approximation algorithm for the makespan minimization problem (MMP) in multiprocessor systems. Its performance guarantee is the product of the performance guarantee of the MMP algorithm and the number of distinct periods in the frame. The second is a 2-dimensional strip packing based framework (2DSPF) which uses the Bottom Left Fill algorithm for 2-D strip packing. The performance guarantee is $2G$, where G is the minimum number of groups into which the set of signals can be partitioned so that the periods of the signals in the same group form a geometric series. The experimental results for GAF and 2DSPF indicate that by carefully assigning offsets for signals in frame packing schemes, one can achieve about 10.83% improvement in bus utilization in CAN-FD systems.**

***Index Terms*—Frame Packing in CAN-FD, Offset Assignment to Signals, 2-D Strip Packing.**

I. Introduction

Modern automobiles are equipped with new and more sophisticated features such as lane keeping and adaptive cruise control. Hence, there is a significant increase in the number of software tasks as well as the number of signals that are transmitted by the Electronic Control Units (ECUs) over the in-vehicle communication network. Often times, there are over 1000 *signals* (the data communicated among software tasks) to be packed into frames and transmitted on one bus. In such a scenario, there is a huge demand for bandwidth regardless of the communication protocol (e.g., Controller Area Network (CAN), FlexRay, Ethernet) used in the automotive system.

CAN, which was standardized in the mid 1990s, is a popular choice among automotive developers for in-vehicle networks. It has attracted a significant amount of research from the real-time systems community ever since its development (e.g., [1], [2], [3]). To meet the growing demand for bandwidth, CAN was extended to CAN-FD in 2012 [4] through two major improvements: (i) increase of bit-rate (up to 8 Mbps), and (ii) increase of payload sizes (up to 64 bytes). The physical layer of CAN was unchanged: the contention resolution mechanism is still a bitwise arbitration based on frame identifiers.

The signal-to-frame packing (or in short, *frame packing*) problem in CAN and CAN-FD has been studied in the literature and shown to be challenging (e.g., [3], [5], [6], [7]). Efficient frame packing enables better bus utilization thereby increasing the amount of data that can be transmitted. In return, this provides better system extensibility (i.e., the ability to accommodate future functionalities). However, all the existing works assume no signal offset assignment even when signals of different periods are packed into the same frame. This, combined with the current practice of using frames with a fixed payload size [8], means that the payload of a frame has to be no smaller than the sum of the payloads of all its signals. Instead, in this work we propose to carefully distribute the signals into frame instances to reduce the frame payload and consequently the bus utilization. We term this problem as *offset assignment* to signals where an *offset* denotes a displacement (in terms of time) of the signal from the first instance of the frame. We now provide a motivational example.

A. Motivational Example

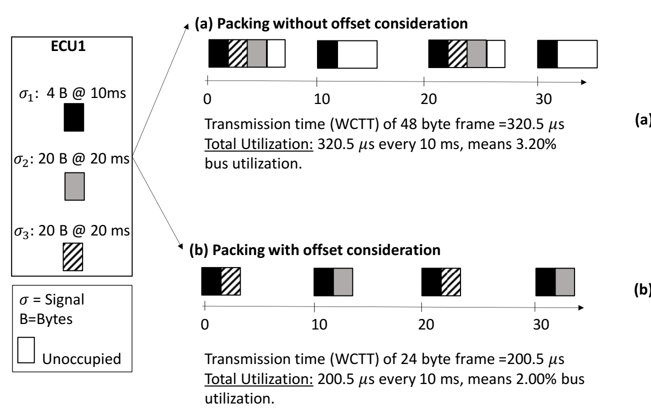


Fig. 1. An example to motivate signal offset assignment for improving bandwidth utilization in frame packing.

Consider a CAN-FD system with a single bus and one ECU transmitting three signals as shown in Figure 1. The signal payload sizes and periods are also shown in the figure.

We compare two frame packing solutions. In the first solution, signal offsets are not considered; i.e., all the signals

Prachi Joshi, Qingyu Liu and Haibo Zeng are with the ECE Department, Virginia Tech, VA 24060, USA. Email: {prachi,qyliu14,hbzeng}@vt.edu.
S. S. Ravi is with the Biocomplexity Institute & Initiative of the University of Virginia and the Computer Science Department of the University at Albany – State University of New York. Email: ssravi0@gmail.com
Soheil Samii and Unmesh D. Bordoloi are with General Motors, MI, USA. Email: {soheil.samii,unmesh.bordoloi}@gm.com
Sandeep Shukla is with IIT Kanpur, India. Email: sandeeps@cse.iitk.ac.in
Manuscript received xx xx, 2018; revised xx xx, 2018.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2019.2907921, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

have an implicit offset of 0 ms. The resulting frame payload size is 48 bytes since the signal payloads $(4, 20, 20$ bytes) sum up to 44 bytes and the nearest available CAN-FD payload size is 48 bytes (see Section III). The frame is transmitted every 10 ms, thus incurring a utilization of 3.2% (as explained in Figure 1). For this solution, there is a payload wastage (as denoted by the white boxes in the figure) of 4 bytes for frame instances at $20 \cdot k$ ms (when all the signals are transmitted), and of 44 bytes for instances at $(20k+10)$ ms (when only σ_1 is transmitted), where k is any non-negative integer.

In the second solution, signal σ_2 with a period of 20 ms is assigned an offset of 10 ms. The remaining two signals (σ_1 and σ_3) are assigned an offset of 0 ms. Thus, the two signals σ_2 and σ_3 with period 20 ms are transmitted alternately in the frame. The frame payload size is reduced to 24 bytes. This arrangement leads to zero payload wastage for the frame and a lower utilization (as compared to the first solution) of 2%. Thus, by assigning appropriate offsets to signals, we can improve the utilization of a frame.

The above example shows that signal offset assignment can lead to an improvement in bandwidth utilization even with just one ECU and three signals. We deliberately choose a simple example to quickly convey the usefulness of offset assignment. In real systems, there are multiple CAN-FD buses and many ECUs, with each ECU generating signals that result in many frames. Hence, offset assignment can lead to a considerable improvement in bandwidth utilization in practice.

B. Contributions

In this paper, we motivate and formulate the signal offset assignment problem (SOAP) in the context of CAN-FD frame packing, where the goal is to minimize the bus utilization by appropriately assigning signals into frame instances. To the best of our knowledge, this is the first attempt to formulate, solve, and apply the *offset assignment problem* for frame packing in CAN-FD. Our contributions are as follows:

- We are the first to motivate, solve and apply SOAP for signal-to-frame packing in CAN-FD.
- We prove that SOAP is, in general, strongly NP-complete by a reduction from the 3-Partition problem. This rules out the existence of pseudo-polynomial algorithms for SOAP, under the standard assumption that **P** $\neq$ **NP** [9].
- We propose a **general approximation framework** (GAF) for SOAP that can use any approximation algorithm for the makespan minimization problem (MMP) for multiprocessor systems. We prove that GAF provides a worst-case performance guarantee of ρK, where K is the number of distinct signal periods in the frame and ρ is the performance guarantee of the algorithm for MMP used in GAF. We have presented the GAF for SOAP in [10]. Since several approximation algorithms with ρ close to 1 are known for MMP (e.g., [11], [12]), the performance guarantee provided by GAF is close to K.
- We further propose a **2-D Strip Packing Framework (2DSPF)** that uses the Bottom Left Fill 2-D strip packing algorithm for SOAP. This approach improves the previous performance bound to $2G$, where G is the minimum number of groups into which the set of signals must be partitioned so that the periods of the signals in each group form a geometric series.
- Our experimental results show that assigning signal offsets in the frame packing step achieves up to 10.83% improvement in bus utilization over the baseline approach with no offset assignment. We also show improved schedulability from offset assignment compared to the case without offset assignment.

The rest of the paper is organized as follows. Section II summarizes the related work. Section III gives a brief overview of the CAN-FD protocol. Section IV defines the signal offset assignment problem and briefly discusses its complexity. Section V describes the proposed general approximation framework. Section VI presents the 2D Strip Packing approach for offset assignment. Section VII explains how the offset assignment step can be integrated into frame packing. Section VIII presents the experimental results on synthetic systems and an industrial case study. Finally, Section IX concludes the paper and discusses directions for future work.

II. Related Work

The frame packing problem in CAN and CAN-FD has been addressed before. Its main difficulty comes from the fact that signals have different periods, deadlines and sizes. For the single bus CAN-FD frame packing problem, Bordoloi and Samii [3] present a heuristic based on dynamic programming for packing the signals, followed by a priority assignment step. Di Natale et al. [5] present a single-step Integer Linear Programming (ILP) formulation to achieve optimal bus utilization while respecting the schedulability of frames. Joshi et al. [13] address the frame packing problem for a multi-bus CAN-FD system with an ILP formulation and a greedy heuristic. For standard CAN, Pölzlbauer et al. [14] and Sandstrom et al. [7] present frame packing approaches inspired by the *next fit decreasing* heuristic for the well-known bin packing problem. Saket and Navet [15] present a bi-directional frequency fit (BDFF) frame packing heuristic which sorts the signals by their bandwidth utilization and then packs this list of sorted signals alternately from both sides of the list. However, *none of these papers considers offset assignment* for signals in a frame for the purpose of improving bus utilization or schedulability.

The frame packing problem has also been considered under other communication protocols that are time-triggered (such as FlexRay static segment [16], [17], [18], [19], [20]) or mixed event/time-triggered [21]. However, due to the different timing properties of these communication protocols, the frame packing problem (with or without offset assignment) is different from that for CAN and CAN-FD. Hence, these approaches are not directly applicable to CAN-FD frame packing.

In this work, we for the first time study the signal offset assignment problem in frame packing, where suitable offsets may be assigned to signals in a frame. This problem shares several characteristics with the makespan minimization problem (MMP) (also known as the *load balancing problem*), which arises in parallel computing. However, the presence of signals with different periods adds to the complexity and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2019.2907921, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

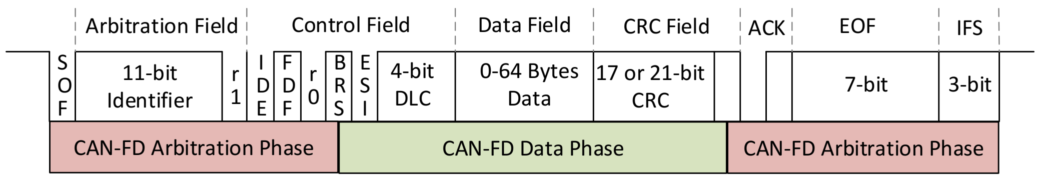


Fig. 2. CAN-FD Frame Format (from [4]).

hence we cannot directly use any of the existing heuristics for MMP to solve SOAP. In general, the goal of MMP is to distribute a set of independent jobs with known execution times on multiple processors so that the makespan (i.e., the maximum completion time on any processor) is minimized. Graham [22] shows that a simple greedy algorithm for MMP provides a solution within twice the optimal value. Further, in [23], he shows that when the jobs are sorted in decreasing order of execution times, the greedy approach gives a solution within a factor of $4/3$ of the optimal value. Hochbaum and Shmoys [12] present a polynomial time approximation scheme for MMP which, for any $\epsilon > 0$, provides a solution within the factor $(1 + \epsilon)$ of the optimal value.

In addition to MMP, we also propose a framework for SOAP that uses an approach based on 2-dimensional strip packing. References [24], [25] provide comprehensive surveys on solutions for 2-D strip packing. In this work we use a *Bottom Left Fill* algorithm [26], [27] for solving the strip packing problem. In [16] the authors optimize the FlexRay static segment using a 2-dimensional bin packing approach. However, in their problem they consider fixed width and height and the objective is to minimize the number of utilized bins; in contrast, the goal of strip packing is to pack all the signals in one bin so that the height is minimized. Hence the approach in [16] is not applicable to our setting.

It should be noted that our proposed approach of assigning offset assignments to **signals** is different from assigning offsets to **frames** considered in [28]. In the latter, the offset assignment to frames is similar to task offset assignment that is proposed in real-time scheduling theory [29] to improve system schedulability. Unlike our work, in [29], [28] offsets are assigned to software tasks and frames, i.e., the scheduling entity. The main benefit of signal offset assignment is improved bus utilization, although we also observe some improvement in the schedulability of frames as a by-product of the main goal. In the frame packing context, we ensure the real-time schedulability of the packed frames using the analysis given by Davis et al. in [2], a correction to the original analysis [1].

III. CAN-FD Overview

In this section, we briefly describe the main features of CAN-FD frame format shown in Figure 2. For additional details, we refer the reader to [3]. A **dominant** bit is a logical 0 and a **recessive** bit is a logical 1. A CAN-FD frame is divided into two phases: arbitration phase and data phase.

Arbitration Phase. The arbitration phase in the CAN-FD frame consists of the following fields: SOF (Start Of Frame), arbitration field, part of the control field, ACK (Acknowledgment), EOF (End OF Frame), and IFS (Inter-Frame Space). The 11-bit (or 29-bit in case of extended format) identifier denotes the frame priority: the lower the value of the identifier, the higher the priority. The arbitration for transmission happens as follows. During the idle state of the bus, all the nodes with some ready frames send the 11-bit identifier after the SOF bit. During the transmission of the identifier bits, if a node transmits a recessive bit but finds a dominant bit on the bus, it stops transmission due to the presence of a higher priority frame contesting for transmission. In the end, the node with the highest priority frame wins the arbitration and continues the transmission.

The transmission of bits in the arbitration phase occurs at the arbitration bit-rate, and the duration of transmission for each bit is denoted as t_a. For example, if the arbitration rate is chosen as 500 Kbps, then $t_a = 2\mu s$.

Data Phase. The BRS (Bit-Rate Switch) bit used to decide whether the bit-rate in the data phase is the same as that of the arbitration phase (BRS = 0) or it switches to the increased bit rate (BRS = 1). It is an addition to the CAN-FD frame format. Since our focus is on CAN-FD, we consider the BRS bit in the frames to be recessive (i.e., BRS = 1). At the increased rate of data transmission, each bit transmission occurs with a duration denoted by t_d. For example, if the data rate is chosen as 2 Mbps, $t_d = 0.5\mu s$. The 4-bit DLC (data-length code) field specifies the payload size (in bytes) of the data field. CAN-FD offers 16 distinct frame payload sizes: 0 through 8, 12, 16, 20, 24, 32, 48 and 64 bytes.

The data field is followed by the Cyclic Redundancy Check (CRC) field, which has 17 bits for payloads up to 16 bytes, and 21 bits otherwise. The CRC delimiter bit (recessive) is transmitted next. After this, the bit rate is changed back to that of the arbitration phase.

Transmission Time. The worst-case transmission time (WCTT) of a CAN-FD frame is a function of its payload size (i.e., the size of the data field) and the data rates. As presented in [3], if p is the payload size (in bytes) of a CAN-FD frame, its WCTT is given by:

$$\text{WCTT}(p) \; = \; 32\, t_a + \Big(28 + 5 \left\lceil \frac{p - 16}{64} \right\rceil + 10p\Big) t_d \qquad (1)$$

IV. Offset Assignment Problem for Frame Packing

In this section, we formulate the problem of *offset assignment* for signals in a frame. As input, we are given a set of ECUs connected via a CAN-FD bus and a set of *periodic* signals from each ECU. The goal is to pack the given signals into frames and assign suitable offsets to signals such that the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2019.2907921, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

bus utilization is minimized. Thus, we have a main problem of packing signals into frames (*frame packing*) such that the bus utilization is minimized and a sub-problem of *offset assignment* to signals within a frame to augment the main goal.

Frame Packing in CAN-FD: Let $\Psi = \{\psi_i : i = 1, 2, \ldots, |\Psi|\}$ denote the set of ECUs. The set of signals from an ECU ψ_i is given by $\mathbf{S}(\psi_i) = \{\sigma_j^i : j = 1, 2, \ldots, |\mathbf{S}(\psi_i)|\}$. Each signal $\sigma \in \mathbf{S}(\psi_i)$ is specified using a triplet of parameters $\langle t(\sigma), d(\sigma), p(\sigma)\rangle$ which denote the period (in ms), deadline (in ms) and payload size (in bytes) of the signal respectively.

The desired output is a set of frames where each frame consists of a subset of signals from one ECU. The output set of frames is denoted as $\Gamma = \{\gamma_1, \gamma_2, \ldots, \}$, and each frame γ is characterized by a tuple $\langle\, \mathbf{S}(\gamma), T(\gamma), D(\gamma), P(\gamma), C(\gamma), \pi(\gamma)\,\rangle$, where $\mathbf{S}(\gamma)$ is the set of signals packed into γ. The quantities $T(\gamma)$, $D(\gamma)$, $P(\gamma)$, $C(\gamma)$ and $\pi(\gamma)$ are respectively the period, deadline, payload size, WCTT, and priority of γ. (Additional information about these parameters appears later in this section.) All the frames must satisfy the following properties: a) Each signal σ is placed in exactly one frame γ, and b) For each frame γ, all the signals in γ are from the *same* ECU.

Signal Offset Assignment: A frame may consist of a group of signals with different periods. For every frame having signals with different periods, each signal $\sigma \in \mathbf{S}(\gamma)$ is assigned an offset from its set of *permissible offsets*. Figure 1(b) shows an example of a frame with signals having two different periods: 10 and 20 ms; for the rest of this section we will use it as a running example for offset assignment. As mentioned earlier, the goal of offset assignment is to minimize the bandwidth utilization. To specify the conditions for a valid offset assignment, we first introduce a few definitions.

- The *base period* t_b for a set $\mathbf{S}(\gamma)$ of signals with two or more distinct periods is defined as the greatest common divisor (gcd) of the signal periods. That is, $t_b = \mathrm{gcd}\{t(\sigma) : \sigma \in \mathbf{S}(\gamma)\}$. For the example in Figure 1(b), the base period is the gcd of 10 and 20; thus $t_b = 10$.
- The *hyperperiod* t_h for a set $\mathbf{S}(\gamma)$ of signals is defined as the least common multiple (lcm) of all the periods; that is, $t_h = \mathrm{lcm}\{t(\sigma) : \sigma \in \mathbf{S}(\gamma)\}$. Note that when the signal periods are harmonic, t_b is the smallest and t_h is the largest period in a signal set. In Figure 1(b), the hyperperiod is the lcm of 10 and 20; thus $t_h = 20$.
- Each frame γ has a tuple $F(\gamma) = \langle F_0, F_1, ..., F_{N(\gamma)-1}\rangle$ of *frame instances*, where $N(\gamma) = t_h/t_b$ denotes the total number of instances of γ in the hyperperiod t_h. The *activation time* $A(F_n)$ of frame instance $F_n \in F(\gamma)$, where $n \in \{0, 1, .., N(\gamma) - 1\}$, is given by $A(F_n) = n \times t_b$. The total number of instances in Figure 1(b) in a single hyperperiod are $N = 2$ and they are at $A(F_0) = 0$ and $A(F_1) = 10$ ms.

We now state the conditions to be satisfied by a valid offset assignment. Let t_1, t_2, $\ldots$, t_K denote the periods of signals in $\mathbf{S}(\gamma)$, where K is the number of distinct signal periods.

- For a signal σ with period $t_i = t(\sigma)$, the set of *permissible offsets* is given by $\Phi(t(\sigma)) = \{j \times t_b : j = 0, 1, .., N_i - 1\}$ where $N_i = t(\sigma)/t_b$. Thus, signal σ must be assigned an offset $\phi(\sigma) \in \Phi(t(\sigma))$. For values of $j > (N_i - 1)$, the offsets are repeated; hence, we do not consider them. Note that for a signal σ with period equal to the base period (i.e., $t(\sigma) = t_b$), the set of permissible offsets is $\Phi(t_b) = \{0\}$. For example, the sets of permissible offsets in Figure 1(b) are $\Phi(10) = \{0\}$ and $\Phi(20) = \{0, 10\}$.
- Each signal σ is transmitted in $t_h/t(\sigma)$ frame instances for any valid offset assignment: σ is assigned to all frame instances F_n such that $A(F_n) = \phi(\sigma) + (q-1) \times t(\sigma)$ for $q \in \{1, ..., \frac{t_h}{t(\sigma)}\}$. We use the notation $\sigma \in F_n$ to indicate that signal σ is assigned to frame instance F_n. Thus, the frame instances act like *bins* for signals, and each offset assignment represents an assignment of signals to a different set of bins.

From the above formulation of offset assignment, we can observe that assigning non-zero offsets to signals allows us to carefully balance the loads of frame instances. Given $\mathbf{S}(\gamma)$ and the offset of each signal in $\mathbf{S}(\gamma)$, the other parameters of the frame γ are determined as follows.

- The period $T(\gamma)$ of frame γ is equal to the base period of the signals in γ.
- $D(\gamma) = \min\{d(\sigma) : \sigma \in \mathbf{S}(\gamma)\}$; i.e., the deadline of γ is the smallest deadline among the signals in γ. All instances of a frame have the same deadline.
- The *occupancy* of a frame instance F_n is defined as the sum of its constituent signal payloads, i.e., $\sum_{\sigma\in F_n} p(\sigma)$. The payload $P(F_n)$ of the frame instance F_n is the smallest CAN-FD payload size that is no smaller than the occupancy of F_n; thus, $P(F_n) \geq \sum_{\sigma\in F_n} p(\sigma)$. CAN-FD standard [4] restricts $P(\gamma)$ to be one of the following values: 0 through 8, 12, 16, 20, 24, 32, 48 and 64 bytes. The payload of a frame is taken as the maximum payload of all its frame instances, $P(\gamma) = \max\{P(F_n) : n = 0, 1, \ldots, N(\gamma)-1\}$. For example, the occupancy of each frame instance in Figure 1(b) is 24; thus, the frame payload is $P(\gamma) = 24$.
- The WCTT $C(\gamma)$ of a frame γ is determined by Equation (1), with the variable p being replaced by $P(\gamma)$. The value of WCTT in Figure 1(b), computed using $P(\gamma) = 24$, is 200.5μs.
- $\pi(\gamma)$ represents the unique priority assigned to frame γ. (Priority assignment is discussed later in this paper.)

The bandwidth utilization $U(\gamma)$ of frame γ is defined as

$$U(\gamma) = \frac{C(\gamma)}{T(\gamma)} \tag{2}$$

In Figure 1(b), the frame bandwidth utilization is given by $U(\gamma) = 2\%$.

SOAP and its Complexity: Given a set $\mathbf{S}(\gamma)$ of signals in frame γ which satisfy the conditions mentioned earlier, the signal offset assignment problem (SOAP) is to ensure: (i) the number of frame instances per hyperperiod is $N(\gamma) = t_h/t_b$; and (ii) the maximum occupancy of all frame instances is minimized. Here we use the *occupancy* of a frame as a proxy for its bandwidth utilization, for two reasons. First, the bandwidth of a frame γ is a monotonic non-decreasing function of the occupancy of γ. Hence, minimizing the occupancy also reduces the bandwidth. Second, in CAN-FD systems, unlike the occupancy (which is the sum of the signal payloads), the frame utilization is a discontinuous and nonlinear function of the signal payloads. In general, it is difficult to handle such functions in complexity proofs and approximation analysis.

In [10] we show that SOAP is strongly **NP**-complete even if the number of distinct signal periods is 2. Thus, there is no pseudo-polynomial time algorithm (i.e., an algorithm whose time complexity is a polynomial function of the size of the input and the maximum integer value in the input) for SOAP unless **P** = **NP** [9]. Hence, we propose two approximation frameworks: 1) a Makespan Minimization Problem (MMP) based Generalized Approximation Framework (GAF) and 2) a 2-Dimensional Strip Packing based Framework (2DSPF).

V. GAF FOR SOAP

In this section, we present a generalized approximation framework (GAF) for SOAP. Before describing this framework, we recall a necessary definition. An approximation algorithm $\mathcal{B}$ for a minimization problem provides a (worst-case) **performance guarantee** of $\rho \geq 1$ if for every instance of the problem, the solution value produced by $\mathcal{B}$ is at most $\rho\times$ OPT, where OPT is the optimal solution value. The smaller the value of ρ, the better the quality of approximation. For many optimization problems, approximation algorithms with good performance guarantees are known (e.g., [30]).

GAF relies on approximation algorithms for the well known **makespan minimization problem** (MMP) defined below. Given an approximation algorithm with a performance guarantee of ρ for MMP, we prove that our framework provides a performance guarantee of at most ρK, where K is the number of different signal periods. After proving this result (Theorem 1), we point out how one can derive several approximation algorithms for SOAP using known approximation algorithms for MMP. As mentioned earlier, the goal of SOAP is to minimize the maximum occupancy of the frame. Our performance guarantee results are with respect to this objective.

We begin by defining the **Makespan Minimization Problem**[1] (MMP) for scheduling jobs on multiprocessor systems. In MMP, it is assumed that jobs are independent (i.e., there are no precedence constraints among the jobs) and non-preemptive. Given an assignment of jobs to processors, the **completion time** for a processor is the sum of the execution times of the jobs assigned to that processor. The **makespan** of the schedule is the *maximum* completion time over all the processors. A formal statement of the MMP problem is as follows.

Definition 1. (MAKESPAN MINIMIZATION PROBLEM) (MMP) *An instance of MMP consists of a set* $\mathbf{T} = \{T_1, T_2, \ldots\}$ *of jobs where the processing time of job* T_i *is* β_i *(a positive integer). The goal is to find an assignment of each job in* $\mathbf{T}$ *to one of* m *(identical) processors so that the makespan is minimized.*

[1]This problem is also known as the **Load Balancing Problem** in the literature (e.g., [11]).

The decision version of MMP is known to be strongly **NP**-complete [9]. However, it has many approximation algorithms with good performance guarantees (e.g., [11], [12]).

To see the usefulness of MMP in obtaining an approximation algorithm for SOAP, consider a set of signals $\mathbf{S}_i$ such that all the signals in $\mathbf{S}_i$ have period t_i. With a slight abuse of notation, let $\Phi(t_i)$ denote the set of permissible offsets for signals with period t_i. We can think of each signal $\sigma \in \mathbf{S}_i$ as a job whose execution time is equal to $p(\sigma)$ (i.e., the payload size of σ). Further, each offset value in $\Phi(t_i)$ can be thought of as a processor. With this correspondence, it can be seen that assigning offsets to signals in $\mathbf{S}_i$ to minimize the maximum occupancy at any offset value corresponds to the MMP problem. While this observation is useful when all the signals have the same period, any heuristic for SOAP must consider signals with different periods. We now explain how our approximation framework GAF works.

Idea behind GAF for SOAP: Given a frame γ with different period signals (an instance of SOAP) and an approximation algorithm $\mathcal{A}$ with a performance guarantee of ρ for MMP, GAF provides a performance guarantee of at most ρK, where K is the number of distinct periods of signals in the SOAP instance. Since we consider SOAP for a particular frame, for simplicity we omit γ from the notation in the remainder of this section. Our framework GAF uses Algorithm $\mathcal{A}$ for MMP as a blackbox. Let $t_1 < t_2 < \ldots < t_K$ denote the K distinct periods of the input signals in increasing order. Let $\mathbf{S}_i$ denote the set of signals with period t_i in γ, $1 \leq i \leq K$. The number of frame instances to be used is $N = t_h/t_b$ where $t_h = \text{lcm}\{t_i, i = 1, ..., K\}$ and $t_b = \text{gcd}\{t_i, i = 1, ..., K\}$. As mentioned earlier, let $\Phi(t_i)$ denote the set of *permissible offsets* for signals with period t_i. The set of all possible offset values is given by $\Phi = \{j \times t_b \ : \ 0 \leq j \leq N-1\}$.

GAF (detailed in Figure 3) considers each set $\mathbf{S}_i$ of signals with period t_i $(1 \leq i \leq K)$ *separately* and uses $\mathcal{A}$ to assign offsets to those signals as follows. Let $N_i = t_i/t_b$. As explained above, we treat each signal σ in $\mathbf{S}_i$ as a job with processing time $p(\sigma)$ (the payload size of σ) and the set $\Phi(t_i) = \{j \times t_b \ : \ 0 \leq j \leq N_i - 1\}$ of possible offset values for the signals in $\mathbf{S}_i$ as the set of processors to which the jobs must be assigned. Any solution to the resulting MMP represents an offset assignment to the signals in $\mathbf{S}_i$. After $\mathcal{A}$ returns a solution, for any signal $\sigma \in \mathbf{S}_i$ which is assigned offset $\phi(\sigma) \in \Phi(t_i)$, we place σ in the frame instances F_n such that $A(F_n) = \phi(\sigma) + (q-1) \times t_i$, where $q = 1, 2, \ldots, t_h/t_i$ and $0 \leq n \leq N-1$. When this process (Step 2.c) in Figure 3) is repeated for all the K periods, we get K separate offset assignments. In the last step, we merge all these assignments by considering each offset value $j \times t_b$ $(0 \leq j \leq N-1)$ separately and combining the signals from various periods which are assigned that offset value. Thus, in the final solution, for $0 \leq j \leq N-1$, the occupancy of the frame instance with activation time $j \times t_b$ is the sum of the occupancies of that offset value over all the periods.

To establish the performance guarantee for GAF, we need to introduce some notation and prove a lemma. For the subset $\mathbf{S}_i$ of signals (having period t_i), let OPT_i denote the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2019.2907921, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

Steps of GAF:

1) Let $\mathbf{S}_i$ be the set of signals with period t_i $(1 \leq i \leq K)$.
2) **for** $i = 1$ **to** K **do**
   a) Let $N_i = t_i/t_b$.
   b) Use the approximation algorithm $\mathcal{A}$ for the signal set $\mathbf{S}_i$ with $\Phi(t_i) = \{j \times t_b : 0 \leq j \leq N_i - 1\}$ as the permissible offsets.
   c) For each signal σ which is assigned offset $\phi(\sigma) \in \Phi(t_i)$ place σ in the frame instances F_n such that $A(F_n) = \phi(\sigma) + (q-1) \times t_i$, where $q = 1, 2, \ldots, t_h/t_i$ and $0 \leq n \leq N-1$.
   d) Let P_i denote the resulting assignment for $\mathbf{S}_i$.
3) Merge the K assignments P_1, P_2 ..., P_K into a single offset assignment P by combining all the signals with the same offset in different assignments into the same frame instance. Output the assignment P.

Fig. 3. Proposed GAF for SOAP

optimal solution value (i.e., the largest occupancy of any frame instance in any optimal solution) for SOAP restricted to subset $\mathbf{S}_i$, $1 \leq i \leq K$. Let OPT denote the optimal solution value for SOAP for the set $\mathbf{S}$. The following lemma shows a relationship between OPT and OPT_i for $1 \leq i \leq K$.

Lemma 1. *For* $1 \leq i \leq K$, $\text{OPT}_i \leq \text{OPT}$.

Proof. Please refer to Appendix A. □

We now establish the performance guarantee from GAF.

Theorem 1. *Let* $\mathbf{S}$ *denote the given set of signals and let* K *denote the number of distinct periods of the signals in* $\mathbf{S}$. *Let* ρ *denote the performance guarantee provided by* $\mathcal{A}$ *for MMP.* GAF *provides a performance guarantee of* $1 + \rho(K-1)$ *if there are signals with base period; otherwise,* GAF *provides a performance guarantee of* ρK.

Proof. Please refer to Appendix B. □

In the automotive domain, the number K of distinct periods of the signals is typically a constant which does not depend on the number of signals. For example, as indicated in [31], $K = 9$ in the real-world automotive benchmark. Moreover, the number of distinct periods of signals actually packed in a frame is even much smaller (due to the packing algorithm which minimizes bandwidth utilization). Hence, in practice, one would expect the approximation algorithm to perform better than what its worst-case performance guarantee would indicate.

A. Deriving approximation algorithms from the framework

Several approximation algorithms with proven performance guarantees are known for MMP. We now point out how GAF enables us to get different approximation algorithms for SOAP using known approximations for MMP.

(i) The simple greedy algorithm $\mathcal{A}_1$, which considers signals in an arbitrary order and assigns each signal an offset which corresponds to a frame instance that currently has the least load, provides a performance guarantee of 2 for MMP [11], [23]. Using this algorithm in Step 2.b) of Figure 3, our framework leads to an approximation algorithm for SOAP with a performance guarantee of $2K-1$ when there are signals with base period and $2K$ otherwise.

(ii) The variant of the greedy algorithm (denoted by $\mathcal{A}_2$) which considers signals in *decreasing* order of sizes provides a performance guarantee of $4/3$ for MMP [23]. Using this algorithm, our framework leads to an approximation algorithm for SOAP with a performance guarantee of $(4K-1)/3$ when there are signals with base period and $4K/3$ otherwise.

(iii) Hochbaum and Shmoys [12] present an approximation scheme (denoted by $\mathcal{A}_\epsilon$), which for any given $\epsilon > 0$ provides a performance guarantee of $(1+\epsilon)$ for MMP. Using $\mathcal{A}_\epsilon$, our framework provides a performance guarantee of $[1 + (K-1)(1+\epsilon)]$ when there are signals of base period and $(1+\epsilon)K$ otherwise. By choosing ϵ appropriately, this performance guarantee can be made arbitrarily close to K. Theoretically, this algorithm runs in polynomial time. However, as observed in [12], it is impractical for small values of ϵ since its time complexity is $O((n/\epsilon)^{1/\epsilon^2})$ (For example, for $\epsilon = 0.1$, the exponent of n in the running time is 100). Hence in our experiments we do not use $\mathcal{A}_\epsilon$.

B. An ILP Formulation for the MMP

One can also formulate MMP as an integer linear program (ILP) as we show in this section. The resulting ILP can be solved using any ILP solver to get an optimal solution to MMP. Let us denote this algorithm by $\mathcal{A}_3$. Using $\mathcal{A}_3$ (which provides a performance guarantee of 1 for MMP), our framework provides a performance guarantee of K for SOAP. While the methods (i), (ii) and (iii) discussed in Section V-A run in polynomial time, the worst-case running time of $\mathcal{A}_3$ is not polynomial. However, there are many ILP solvers (such as [32]) that work very well in practice. So, this is indeed a viable approach. We present the ILP formulation for a group of signals having the same period. Suppose we are given a set $\mathbf{S}$ of n signals denoted by σ_1, σ_2, ..., σ_n, all of which have the same period. Signal σ_i has a size of p_i (a positive real number), $1 \leq i \leq n$. We are also given a set of m frame instances denoted by F_1, F_2, ..., F_m. The ILP formulation uses binary variables x_{ij}, $1 \leq i \leq n$ and $1 \leq j \leq m$. The interpretation is that $x_{ij} = 1$ if signal σ_i is assigned to frame instance F_j; otherwise, $x_{ij} = 0$. A real variable Δ denotes the value of makespan (which corresponds to the maximum occupancy of a frame). Now, the ILP formulation for MMP is:

$$\text{Minimize} \quad \Delta$$

$$\text{such that} \sum_{j=1}^{m} x_{ij} = 1, \quad \text{for all } i, \quad 1 \leq i \leq n \tag{3}$$

$$\sum_{i=1}^{n} x_{ij} \cdot p_i \leq \Delta, \quad \text{for all } j, \quad 1 \leq j \leq m \tag{4}$$

Here, the set of constraints in Equation (3) ensures that each signal is assigned to exactly one frame instance. The second set of constraints (Equation (4)) ensures that the occupancy of each frame instance is at most Δ.

VI. Improving Offset Assignment by Using 2D Strip Packing Approach

With the focus on improving our approach for assigning offsets to signals in order to further reduce the bandwidth utilization due to frame packing in CAN-FD, we propose and develop a new framework that utilizes a **2-dimensional strip packing** approach for offset assignment. In the new framework we first propose a transformation function to convert the offset assignment (OA) instance to a 2-dimensional (2-D) strip packing (SP) instance. Next, we use an algorithm to solve the SP problem and then transform the SP solution back to obtain the OA solution. We show that the new framework provides a possibly better worst-case approximation ratio as compared to GAF (Section V). We show the improvement in bandwidth utilization through experiments, where we observe an improvement in average bandwidth utilization of up to 2.8% as compared to GAF. We also apply the new framework to an automotive case study and obtain a modest improvement of 0.1% in bandwidth utilization.

A. 2-Dimensional Strip Packing Problem

In a 2-D SP problem we are given a set of items I with $\kappa = |\mathbf{I}|$. Each item $\tau_i \in \mathbf{I}$ is a rectangle with width w_i and height g_i. We have a strip of width W, i.e., the x-coordinate is in the range $[0, W)$. The objective of 2-dimensional SP is to pack all the items $\{\tau_1, \tau_2, \ldots, \tau_\kappa\}$ in the strip such that the height used is minimized. Note that the items cannot be rotated and no overlap of items is permitted. Also, all numbers in our 2-D SP problem are non-negative integers.

In the 2-D SP problem, *packing* an item is defined as follows: *packing* τ_i at (x_i, y_i) means that the lower left corner of the rectangle τ_i is placed at (x_i, y_i). Such a rectangle covers all the points inside and on the boundary of τ_i. In other words, τ_i placed at (x_i, y_i) covers the set of points given by

$$\mathbf{C}(x_i, y_i) = \{(x, y) \mid x_i \leq x < x_i + w_i \text{ and } y_i \leq y < y_i + g_i\}. \tag{5}$$

The goal is to pack all items in a rectangular region (i.e., a strip) with width $W \geq \max_{1 \leq i \leq \kappa} w_i$ and minimum height. More specifically, an optimal solution to the 2-D SP problem should satisfy:

1) The items are packed within the strip, i.e., each item τ_i is placed at (x_i, y_i) with $0 \leq x_i < W - w_i$.
2) No two items overlap; that is, any point (x, y) with $0 \leq x < W$, must satisfy *exactly one* of the following conditions: (i) it is not covered by any item, (ii) it is on the boundary of at least one item, (iii) it is inside exactly one item.
3) The maximum height is minimized; that is, our objective is to minimize $\max_{1 \leq i \leq \kappa}(y_i + g_i)$.

As mentioned earlier, in the 2-D SP problem, items cannot be rotated; that is, for any τ_i, g_i and w_i cannot be interchanged.

Two additional terms are used throughout this section: 1) **fully empty space** and 2) **fully occupied space**. We say that a 2-D space is "fully occupied" if every point *inside* the space is covered by a packed item. Similarly, we say that a 2-D space is "fully empty" when none of the points *inside* the space is covered by a packed item. Note that these concepts are defined on the interior points of the space; they don't concern its boundaries.

B. Problem Transformations

We convert the OA problem into the 2-D SP problem and vice versa using two transformation functions. The transformation from OA to SP ($OA \rightarrow SP$) converts each signal into a 2-D rectangle. For the SP to OA transformation ($SP \rightarrow OA$), it is valid when (a) the signals have periods that form a geometric series with the common ratio being a positive integer $\eta \geq 2$; (b) in the solution to the 2-D SP problem, the x-coordinate of the lower left corner (x_i, y_i) of τ_i must be $x_i = m_i \cdot w_i$, where m_i is some positive integer and w_i is the width of τ_i. With Theorem 2 in Section VI-C, we show that given condition (a), the solution returned by Bottom Left Fill [26] satisfies condition (b). Hence, condition (a) is a constraint on applying the 2-D SP approach to solve SOAP. In practice, this constraint may not hold, and in Section VI-D we discuss how we handle such a situation.

We now define the transformations, under the assumption that the periods of the signals are in a geometric series and their common ratio is a positive integer $\eta \geq 2$. Consider the set of signals $\mathbf{S}(\gamma)$ packed to the same frame γ. Let $t_1 < t_2 < \ldots < t_K$ denote the periods of signals in $\mathbf{S}(\gamma)$, where K is the number of distinct signal periods. Hence, $t_b = t_1$, $t_h = t_K$. Since these periods form a geometric series with a common factor η, we have $\forall 1 \leq k \leq K$, $t_k = t_1 \times \eta^{q_k}$ where q_k is a non-negative integer. For the **forward transformation**, $T_{OA \rightarrow SP}$, we convert each signal in the OA to an item in SP as follows:

1) Each signal σ_i corresponds to an item (rectangle) τ_i. Since $\mathbf{S}(\gamma)$ is the set of signals in frame γ, let $\mathbf{I}(\gamma)$ denote the set of items in frame γ. Thus, $|\mathbf{S}(\gamma)| = |\mathbf{I}(\gamma)|$.
2) The width w_i of item $\tau_i \in \mathbf{I}(\gamma)$ is given by the total number of instances of the corresponding signal $\sigma_i \in \mathbf{S}(\gamma)$ in the hyperperiod. Thus
$$w_i = t_h / t(\sigma_i) = \eta^{q_K - q_i}.$$
3) The height g_i of item $\tau_i \in \mathbf{I}(\gamma)$ is the payload $p(\sigma_i)$ of signal $\sigma_i \in \mathbf{S}(\gamma)$.
4) The maximum width W of the SP instance is
$$W = t_h / t_b = \eta^{q_K}.$$

Note that W is also the number of frame instances in γ.

The **backward transformation** $T_{SP \rightarrow OA}$ is defined as follows. In order to transform an SP instance to an OA instance, we map each integer $x \in \{0, 1, ..., W-1\}$ in the SP instance to a unique slot in the OA instance which corresponds to the activation time of the frame instance in OA. Specifically, we write x as a base-η number

$$x = \sum_{j=0}^{q_K - 1} b_j \times \eta^j, \text{ where } b_j \in \{0, 1, ..., \eta - 1\} \tag{6}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2019.2907921, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

We then reverse the digits in x to get a new base-η number x'

$$x' = \sum_{j=0}^{q_K-1} b_{q_K-1-j} \times \eta^j \tag{7}$$

Finally, the activation time of the frame instance that corresponds to x is

$$t_b \times x' \tag{8}$$

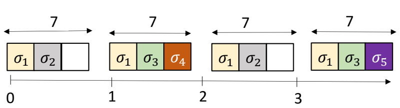
Hence, for an item τ_i that is placed at (x_i, y_i) in the solution to the 2-D SP problem, the OA instance transmits the corresponding signal σ_i in all the frame instances F_n with activation times as

$$A(F_n) \in \{t_b \times x' \mid x_i \leq x < x_i + w_i\} \tag{9}$$

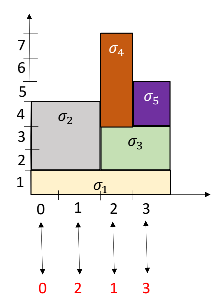
The frame payload is chosen as the maximum payload among all frame instances (as described in Section IV).

To illustrate the transformation of OA to SP and vice versa, consider Figure 4. The list of signals and their parameters is given in the table on the top left corner in Figure 4. For this example the value of $\eta = 2$ and $W = 4$. Figure 4(a) presents an offset assignment for the given signals. Signal σ_1 has a period of 1ms and it is assigned an offset of 0ms. Signals σ_2 and σ_3 have a period of 2ms and they are assigned offsets 0ms and 1ms respectively. Signals σ_4 and σ_5 have period of 4ms and they are assigned offsets of 1ms and 3ms respectively. The maximum size of all the frame instances is 7.

Signals	Size	Period
σ_1	1	1
σ_2	3	2
σ_3	2	2
σ_4	4	4
σ_5	2	4



(a) Offset Assignment Solution



(b) Strip Packing Solution

Fig. 4. Example to illustrate transformation function

The forward transformation ($T_{OA \to SP}$) is straightforward as each signal is converted to a 2-D rectangular item, as shown in Figure 4(b). For example, the item corresponding to σ_1 has a width of 4 and height of 1. The colors used for the signals and their corresponding items in Figure 4 are the same.

In this work we use an existing approximation algorithm called Bottom Left Fill (BLF) [26] for 2-D strip packing. The BLF algorithm first sorts the items in decreasing order of width. The algorithm iterates over the items in sorted order, and places each item at the bottom-most and left-most position available. Once an item is placed, its position is not changed during any subsequent iteration.

The 2-D strip packing solution produced by BLF is presented in Figure 4(b). The maximum height is 7 in this solution as well. For the backward transformation, $T_{SP \to OA}$, we use the function defined in Equations (6)–(9). The transformed instances corresponding to each $x_k, k \in \{0, 1, 2, 3\}$ are shown in Figure 4(b) in red numbers (which denote the activation times of the instances in OA) below the 2-D strip packing solution. We can observe that both Figure 4(a) and 4(b) give the same offset assignment solution. Thus this example illustrates how it is possible to use the 2-D strip packing approach for solving the offset assignment problem.

C. Correctness and Performance Bound of BLF

We now prove the correctness of the BLF algorithm in that combined with the transformation defined in Section VI-B, it generates a valid solution to OA. We also study its performance guarantee. The BLF algorithm has been shown to have an approximation ratio of 3 for a generic 2-D strip packing problem [26]. However, for the signal offset assignment problem, we show that by using BLF we obtain a performance bound of 2 when the signals have periods that form a geometric series. In order to prove the bound of 2, we show that BLF obtains a packing solution with no holes, by placing each item at the current minimum height (in the 2-D instance).

In the following we first introduce a definition of the **profile** of a 2-D rectangle packing. Throughout this section, the reader should bear in mind that the widths of the rectangles to be packed form a geometric series with a ratio $\eta \geq 2$ and that BLF considers the rectangles in non-increasing order of width. Thus, the width of the first rectangle packed is the maximum width W of the packing. Recall that when BLF **places** a rectangle τ_i at (x_i, y_i), the lower left corner of τ_i is assigned the coordinates (x_i, y_i).

Definition 2 (Profile)**.** *The **profile** of a packing of 2-D rectangles is defined as the function $f(x)$ over the interval $0 \leq x < W$. For any $x \in [0, W)$, the value of $f(x)$ is defined as the largest y such that there is an item (rectangle) whose horizontal edge contains the coordinate (x, y).*

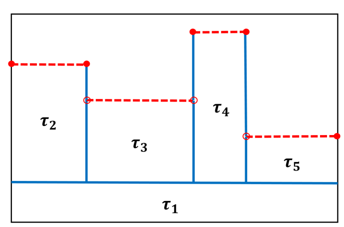


Fig. 5. Example to illustrate the definition of profile

We illustrate the definition of profile using the packing example introduced by Fig. 5, where dotted red lines show the profile of the current packing solution.

It is clear that the profile function before placing any item is given by,

$$f(x) \ = \ 0 \quad \text{for all } x, \quad 0 \leq x < W.$$

After placing the first item, the profile is given by,

$$f(x) \ = \ g_1 \quad \text{for all } x, \quad 0 \leq x < W,$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2019.2907921, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

where g_1 is the height of the first item.

As BLF packs the items one by one, the profile gets modified. However, we can prove that BLF always obtains a packing solution where the space below the profile is fully occupied while the space above the profile is fully empty, implying that no hole exists in the packing produced by BLF.

For some $i \geq 2$, suppose BLF places item τ_i at (x_i, y_i) after placing items τ_1, ..., τ_{i-1}. Specifically, suppose (x', y') is a feasible point for placing τ_i; that is, the space $\{(x, y) : x' < x < x' + w_i \leq W - 1, y' < y < y' + g_i\}$ is fully empty, where w_i and g_i are respectively the width and height of τ_i. Further, suppose $\mathcal{F}_i$ is the set containing all feasible points for placing τ_i. Then (x_i, y_i) obtained by BLF to pack τ_i must satisfy:

1) $y_i \leq y$, for all $(x, y) \in \mathcal{F}_i$ and
2) $x_i \leq x$, for all $(x, y_i) \in \mathcal{F}_i$.

In the following lemma, we prove that BLF always places an item at the current minimum height, and no hole is created in the packing. As mentioned earlier, we assume the widths of items are in non-increasing geometric order; that is,

1) $w_i = w_j \cdot n_{ij}, \quad n_{ij} \in \mathbb{Z}^+, \ \forall\, i, j, 1 \leq i < j \leq \kappa$, and
2) $W = w_1$.

where $\mathbb{Z}^+$ is the set of positive integers.

Lemma 2. *Let $L = \langle \tau_1, \tau_2, \ldots, \tau_\kappa \rangle$ be a list of items such that the widths are in non-increasing order and form a geometric series. Let W denote the width of item τ_1, that is also the maximum width of the packing. Suppose BLF packs the items one at a time in the order specified by L, and suppose (x_i, y_i) is the packing coordinate of the item τ_i. For $1 \leq i \leq \kappa$, let $L_i = \langle \tau_1, \ldots, \tau_i \rangle$ and let f_i be the profile after packing all the items in L_i. Then, for each i, $1 \leq i \leq \kappa$, the following conditions hold.*

1) *The region on or below the profile is fully occupied by the items in L_i; i.e., each point in the region Γ_i defined by*
$$\Gamma_i = \{(x, y) : 0 \leq x < W \text{ and } 0 \leq y \leq f_i(x)\}.$$
is in the interior or boundary of an item in L_i.
2) *The region above the profile is fully empty; i.e., no point in the region Γ'_i defined by*
$$\Gamma'_i = \{(x, y) : 0 \leq x < W \text{ and } y > f_i(x)\}.$$
is in the interior or boundary of an item in L_i.
3) *The y-coordinate of the point (x_i, y_i) for packing τ_i is on the minimum height with respect to the packed items in L_{i-1}; that is,*
$$y_i = \min_{0 \leq x \leq W} f_{i-1}(x).$$
4) *The x-coordinate of the point (x_i, y_i) for packing τ_i is a multiple of the width w_i; that is,*
$$x_i = m_i \cdot w_i \text{ where } m_i \in \mathbb{Z}^+ \cup \{0\}.$$

Proof. Please refer to Appendix C. □

The following theorem formally states the correctness of BLF, in that it generates a solution to 2-D SP problem that can be transformed to a valid signal offset assignment.

Theorem 2. *Combined with the transformation $T_{SP \to OA}$ defined in Equations (6)–(9), BLF correctly produces a valid solution for SOAP.*

Proof. Please refer to Appendix D. □

We now study the performance guarantee of BLF. A direct consequence of Lemma 2 is that BLF algorithm places each item at the minimum height of the 2-D strip at that instant and that the packing does not produce any holes. (Conditions (1) and (2) of the lemma imply that no holes are created below the profile while condition (3) shows that the placement is at the minimum height). We first present two lemmas which are straightforward and are used later to prove the approximation ratio of BLF for the packing that arises in the context of SOAP.

Lemma 3. *Suppose BLF packs the κ items given by the list of $L = \langle \tau_1, \tau_2, \ldots, \tau_\kappa \rangle$. Let $f_\kappa(x)$ denote the profile after packing L_κ. Then, the height H^* of the optimal solution to problem SP is at least the minimum height of L_κ; that is,*
$$H^* \geq \min_{0 \leq x < W} f_\kappa(x).$$

Proof. This is a simple consequence of Lemma 2 since the space below the minimum height is fully occupied by packed items with no holes. □

The next lemma points out the simple fact that the optimal height of the packing must be at least as large as the height of each item to be packed.

Lemma 4. *The height H^* of the optimal solution to problem SP is no smaller than the height of any item, namely*
$$H^* \geq g_i, \text{ for all } i : 1 \leq i \leq \kappa.$$

Now, using the above lemmas, we can prove the approximation ratio of BLF.

Theorem 3. *BLF provides a performance bound of 2 for SOAP.*

Proof. Please refer to Appendix E. □

D. SOAP Using 2D Strip Packing Framework (2DSPF)

As discussed in Section VI-B, in order to apply SP for offset assignment, the periods of the signals must form a geometric series with the common ratio being an integer ≥ 2. However, in practice, this may not always be the case. Therefore, in our proposed framework for OAP using BLF (Figure 6), we first partition the signal periods in the frame such that each group in the partition contains elements that form a geometric series. We call such a partition a *valid partition* for SP.

Example of a *valid* partition: Given a set of signals with periods $\{1, 2, 5, 10\}$, some *valid* partitions are: (i) $\{1, 2\}, \{5, 10\}$ and (ii)$\{1, 5\}, \{2, 10\}$. Note that partitions such as $\{1, 10\}, \{2, 5\}$ or $\{1, 2, 5\}, \{10\}$ are *not valid* with respect to SP.

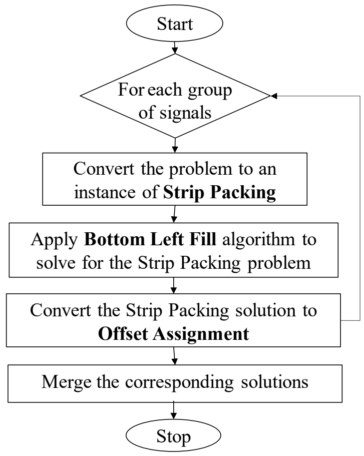


Fig. 6. Framework for Offset Assignment using 2D Strip Packing

Since there can be several ways of partitioning the signal periods, we consider all possible partitions of the given periods in a frame. Note that enumeration of all possible partitions is not trivial. In this work, we have used a straightforward iterative approach to enumerate all possible partitions of signal periods, thus, our approach is an exponential one and it would not scale for large number of periods. We discard the ones that are not *valid*. For each group in a (valid) partition we first transform the OA instance to an SP instance. Next we use the BLF algorithm for SP and apply the backward transformation to obtain a solution to the OA instance. For each partition we merge the solutions corresponding to all the groups (of that partition). Finally we find solutions of all partitions and choose the minimum solution (corresponding to the frame payload size) from all the partitions. If there are G groups, then by using the 2DSPF we can get a performance bound of $2G$.

This framework is then used in the greedy frame packing algorithm proposed in our prior work [13]. In the decision making step, when the assignment of a signal to an existing frame or a new frame is made, we apply the offset assignment scheme using an SP approach.

VII. Application of SOAP to Frame Packing

In order to show the efficacy of our proposed framework for SOAP, we apply it to a frame packing algorithm from the literature. We choose the greedy algorithm discussed in [13], since the offset assignment step can be directly plugged into the bandwidth computation step (see Figure 7). In this algorithm, the frame payload is not fixed and it is computed on the basis of the frame packing. Using our offset assignment scheme, we try to minimize the frame payload by assigning suitable offsets to signals.

Specifically, as observed in [13], the frame packing part of the algorithm gives the best performance in terms of bandwidth utilization when signals are sorted in increasing order of signal period; hence, we use the same ordering in our implementation. However, we modify the algorithm to embed our offset assignment algorithm. In Figure 7 the frame packing approach (in [13]) is given on the left and our proposed SOAP frameworks are shown as a black box on the right. In each

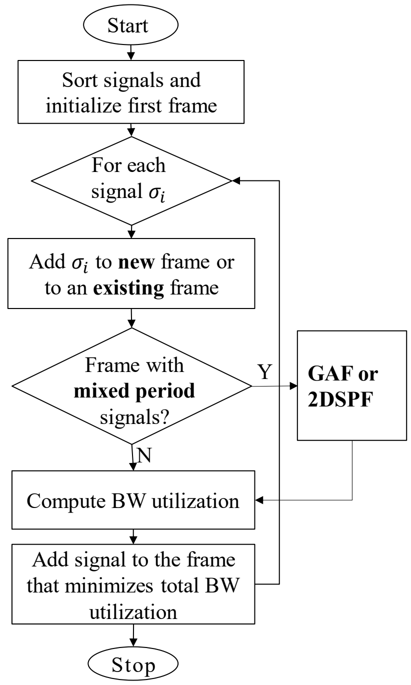


Fig. 7. Flowchart describing the application of SOAP to a frame packing approach in CAN-FD

iteration of frame packing, the algorithm tries to add a signal σ to an existing frame or create a new frame (containing only σ). The bandwidth of each of these alternatives is evaluated after assigning the signal offsets (by calling the SOAP framework, GAF or 2DSPF). If all the signals in a frame have same period, the bandwidth computation is straightforward since the offset for all the signals is 0. If the signals in a frame have two or more periods, we apply our proposed SOAP approaches to improve the bandwidth utilization.

After obtaining a frame packing solution using offset assignment, we analyze the schedulability of the generated frames over the CAN-FD system. The schedulability analysis of CAN-FD system used by us follows that of CAN [2], and it is described in our prior work [10].

VIII. Experimental Results

In this section, we describe the experimental results obtained by applying our proposed frameworks for SOAP: 1) GAF (with algorithms $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$ for MMP as detailed in Section V) and 2) 2DSPF with BLF. We use synthetic systems as well as an industrial case study for the evaluation.

The synthetic systems are generated according to the guidelines for real-world automotive benchmarks [31], with minor modifications. Specifically, we redistribute the share of signals with size larger than 64 bytes to the bin "33-64 bytes". (In this work, we only consider signals with size up to 64 bytes as the CAN-FD frame payload size is limited to 64 bytes.) Further, we also redistribute the share of signals sent by engine control tasks (triggered by rotational events) to those with periods between 1 and 20 ms (as we do not consider the signals with angle-synchronous periods). The distribution of signal periods

Period (ms)	**Share**	**Size** (Bytes)	**Share**
1	4%	1	35%
2	3%	2	49%
5	3%	4	13%
10	31%	5-8	0.8%
20	31%	9-16	1.3%
50	3%	17-32	0.5%
100	20%	33-64	0.4%
200	1%		
1000	4%		

TABLE I
SIGNAL PARAMETERS AND THEIR DISTRIBUTION

and payload sizes is given in Table I. In all our systems, the deadline of any signal is equal to its period. We generate 1000 systems for each experiment and average the performance parameters over all the systems. All systems contain a single CAN-FD bus that connects 3 ECUs, and the number of signals are as follows: 50, 80, 100, 120, 150 and 200. The arbitration and data bit rates are set to 500 kbit/sec and 2 Mbit/sec, respectively. We use CPLEX as the ILP solver [32].

Since there is no prior work on offset assignment for CAN-FD frame packing, we compare our proposed frameworks (the 2DSPF framework with BLF, the GAF framework with $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$ respectively) against a baseline frame packing algorithm in [13] which does not perform offset assignment.

A. Comparison on Bus Utilization

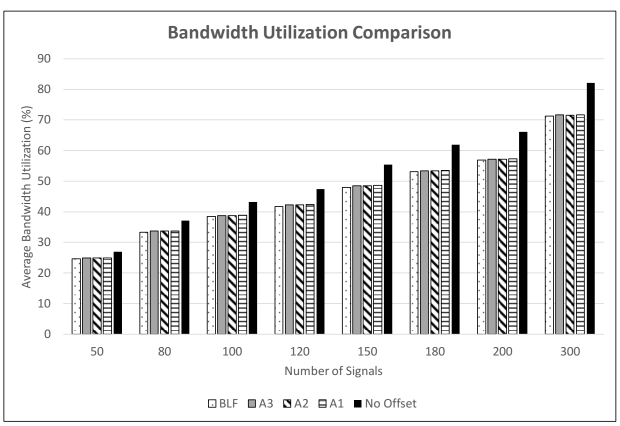


Fig. 8. Average bandwidth utilization comparison using offset assignment versus no offset assignment.

Figure 8 presents the comparison on bandwidth utilization. As in the figure, offset assignment (BLF, $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$) provides a steadily increasing improvement in bus utilization. In the case of 300 signals the improvement is about 10.83% over the No Offset case. We also observe that the bandwidth utilization observed with the offset assignment approaches (in $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$) is almost similar. This shows that in terms of bandwidth utilization, the performance of the GAF framework with polynomial time heuristics $\mathcal{A}_1$ and $\mathcal{A}_2$ is nearly the same as with $\mathcal{A}_3$ (ILP). By using BLF we further improve the offset assignment and obtain a reduction in average bandwidth utilization of upto 0.55% (in the case of 150 signals) as compared to GAF.

B. Comparison on System Schedulability

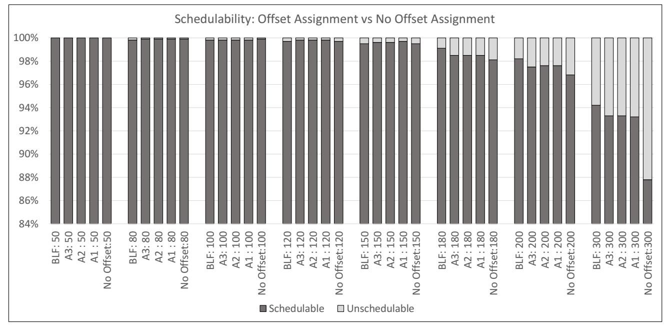


Fig. 9. Comparison on system schedulability.

In addition to bandwidth utilization, we also analyze the schedulability of the packed CAN-FD frames as described in Section VII. In Figure 9 we plot the percentage of schedulable and unchedulable systems for each signal size using BLF, $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$, and compare them against the baseline packing (i.e., no offset assignment). We observe that for small numbers of signals, the performance of the offset assignment schemes with respect to schedulability is similar to the one without offset assignment. However, for larger numbers of signals, the GAF ($\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$) scheme is able to increase the number of schedulable systems by 5.4%. The 2DSPF theme further increases the number of schedulable systems (as compared to GAF) by about 1.1%. This shows that as a side effect, the decrease in frame payload size also reduces the response time of the frame and thus improves schedulability.

C. An Automotive Case Study

We also applied the aforementioned frame packing algorithms on a real automotive case study, containing 3429 signals over a CAN-FD bus shared by 17 ECUs. It involves systems which are broadly in the domains of chassis electronics, body control, adaptive cruise control, and active safety.

Figure 10 shows the bandwidth utilization for this case study with and without offset assignment. As in the figure, GAF reduces the bandwidth utilization from 29% to about 26.7% and 2DSPF reduces it further to 26.6%. There are several reasons why we see only a modest improvement in bandwidth utilization in this case study. First, the CAN bus is under-utilized, since the simple frame packing scheme (without offset assignment) itself provides a low bandwidth utilization of about 29%. Even so, our offset assignment further reduces the bandwidth utilization by about 2.4%, which is a relative improvement of about 8% considering the original low bandwidth utilization of 29%. In the automotive domain this could enable a potential extension of the architecture's lifetime by several product cycles. Second, the signal set in the case study has a large share (about 69%) with the same period (1000 ms). Therefore, the application of SOAP is restricted in this case. Finally, about 60% of the signals are of size 2 bits or less. As a result, the reduction in the occupancy of a frame resulting from offset assignment is rather small, and due to the discontinuity in CAN-FD frame sizes, a small change in occupancy has very little effect on the size of a frame.

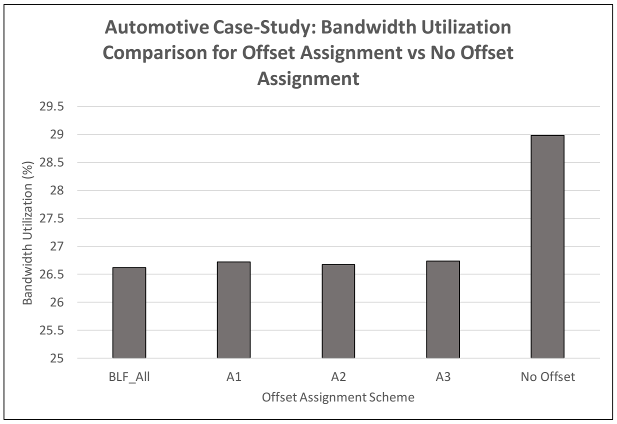


Fig. 10. Bandwidth utilization comparison for frame packing for the real automotive case study with and without offset assignment.

Generation Scheme	ECUs	Improvement in BU
Automotive Benchmark [31]	3	0.54%
Automotive Benchmark [31]	5	0.77%
Automotive Benchmark [31]	10	1.66%
Random Systems [33]	10	2.83%
Random Systems (Periods in GP)	10	0.35%

TABLE II
BANDWIDTH UTILIZATION IMPROVEMENT OF BLF OVER GAF

D. Comparison on Bus Utilization: Different Random Systems

In order to test the performance (in terms of bandwidth utilization) of the proposed framework 2DSPF, we perform experiments with different systems as summarized in Table II. We generate synthetic systems using three different rules, and the number of signals range over the set $\{50, 80, 100, 120, 150, 180, 200, 300\}$. The first three entries in Table II are results of experiments performed using synthetic systems generated using the automotive benchmark rules in [31]. For each synthetic system, we considered 3, 5 and 10 ECUs per system. We compare the average bandwidth utilization for all the systems corresponding to the BLF and GAF with $\mathcal{A}_3$. We observe about 0.54%, 0.77% and 1.66% bandwidth utilization improvement from BLF in the three experiments.

For the next experiment (Random Systems), we generate systems in a different way compared to previous experiments. We follow the generation scheme used in [33]. In these systems the periods of the signals are generated by the product of one to three factors, each randomly drawn from three harmonic sets (2, 4), (6, 12), (5,10). The number of signals range from 50 to 300 and the number of ECUs per system is 10. We observe an improvement of about 2.83% in bandwidth utilization by using BLF as compared to GAF. We also conduct an experiment on a set of systems that follow [33] but signal periods are in a geometric series. We observe an improvement of about 0.35% in bandwidth utilization by using BLF as compared to $\mathcal{A}_3$.

IX. CONCLUSION AND FUTURE WORK

The problem of optimizing bandwidth utilization in CAN and CAN-FD is important for better extensibility and cost reduction. The wastage of bandwidth in frame-packing for CAN-FD occurs mainly from the variable periods and sizes of signals. In this work, we have approached the frame packing problem in CAN-FD systems in a novel way by systematically assigning offsets to signals in a frame and thus distributing the load over the frame instances more evenly. We have shown that this strategy is effective in reducing the bandwidth utilization of the CAN-FD system and improving the schedulability. We presented experimental results using synthetic systems and an automotive case-study.

As our future work, we plan to improve our approximation bound for GAF. We also intend to investigate the applicability of our offset assignment framework to other frame-packing algorithms in the literature (such as [3]). In this paper, our focus was on SOAP for a single domain. In the future, we plan to extend this approach to multi-domain systems.

ACKNOWLEDGMENTS

This work was supported in part by NSF CSR Grant 1812963, NSF DIBBS Grant ACI-1443054, NSF BIG DATA Grant IIS-1633028 and NSF EAGER Grant CMMI-1745207.

APPENDIX

A. Proof of Lemma 1

For any $i \geq 1$, we prove by contradiction that $\mathrm{OPT}_i \leq \mathrm{OPT}$. Suppose for some i, $\mathrm{OPT}_i > \mathrm{OPT}$. Consider any optimal solution for set $\mathbf{S}$ and remove from that solution all the signals except those from $\mathbf{S}_i$. The result is a solution for $\mathbf{S}_i$. Note that the removal procedure does not increase the occupancy of any frame. Therefore, we have a solution for $\mathbf{S}_i$ where the maximum occupancy is *less than* OPT_i. This contradicts the assumption that OPT_i is an optimal solution value for $\mathbf{S}_i$, and the lemma follows. □

B. Proof of Theorem 1

For the set $\mathbf{S}$, let OPT and APPROX denote the maximum occupancy of frames in an optimal solution and that produced by GAF respectively. Similarly, for set $\mathbf{S}_i$, let APPROX_i denote the maximum occupancy of a frame produced by GAF, $1 \leq i \leq K$. We divide our analysis into two cases: (1) when signals with base period are present and (2) when there are no signals with base period.

Case 1: Signals with base period are present. In this case, $t_1 = t_b$ and for the signal set $\mathbf{S}_1$ (consisting of all signals with period t_1), the offset assignment produced by an optimal solution and that by GAF are the same: each signal in $\mathbf{S}_1$ appears at all the N possible offset values. Therefore,

$$\mathrm{APPROX}_1 \;=\; \mathrm{OPT}_1 \;\leq\; \mathrm{OPT},$$

where the last step is a consequence of Lemma 1. For signal sets $\mathbf{S}_2$ through $\mathbf{S}_K$, Step 2.b) of GAF (Figure 3) uses Algorithm $\mathcal{A}$ which provides a performance guarantee of ρ. Using this fact and Lemma 1, we have for $2 \leq i \leq K$,

$$\mathrm{APPROX}_i \;\leq\; \rho \times \mathrm{OPT}_i \;\leq\; \rho \times \mathrm{OPT}.$$

Adding the inequalities for APPROX_i, $1 \le i \le K$, we get

$$\sum_{i=1}^{K} \text{APPROX}_i \;\le\; \text{OPT} \times [1 + \rho(K-1)] \,.$$

Because of the merging procedure used in the algorithm (Step 3 of Figure 3), we have

$$\text{APPROX} \;\le\; \sum_{i=1}^{K} \text{APPROX}_i \;\le\; \text{OPT} \times [1 + \rho(K-1)] \,.$$

Case 2: Signals with base period are absent. When there are no signals with period equal to the base period (i.e., $t_1 \neq t_b$), then the offset assignment for the set of signals $\mathbf{S}_1$ may not be optimal; however, the maximum occupancy is still within the factor ρ of the optimal value. Hence using the same argument as in Case 1, we have for $1 \le i \le K$,

$$\text{APPROX}_i \;\le\; \rho \times \text{OPT}_i \;\le\; \rho \times \text{OPT}.$$

Adding the inequalities for APPROX_i, $1 \le i \le K$, we get

$$\text{APPROX} \;\le\; \sum_{i=1}^{K} \text{APPROX}_i \;\le\; \text{OPT} \times \rho K.$$

This completes the proof of Theorem 1. □

C. Proof of Lemma 2

We prove by induction.

Base Case. Consider the base case of $i = 1$. It is straightforward to see that $(x_1, y_1) = (0, 0)$, and

$$f_1(x) \;=\; g_1, \quad \forall x : 0 \le x < W$$

meets all conditions.

Inductive Step. We assume that Conditions (1) through (4) are always satisfied after packing any of the previous items $\tau_1, ..., \tau_i$, and prove that all conditions hold after BLF packs the item τ_{i+1}.

In order to pack τ_{i+1}, we first find the **left-most** coordinate that is on the **lowest** edge of the profile f_i (if there are multiple of such edges, pick the left-most one). The coordinates $(\bar{x}, \bar{y})$ of such a point are

$$\bar{y} = \min_{0 \le x < W} f_i(x), \quad \bar{x} = \min\{x \mid f_i(x) = \bar{y}\} \tag{10}$$

In the profile f_i, the length of the (horizontal) edge starting from $(\bar{x}, \bar{y})$ is denoted as d, which must satisfy

$$d = \max\{d \mid \forall x' : \bar{x} \le x' < \bar{x} + d, \;\; f_i(x') = \bar{y}\} \tag{11}$$

We first prove that if $d \ge w_{i+1}$, all four conditions are satisfied after BLF places τ_{i+1} at the point $(\bar{x}, \bar{y})$. Subsequently, we will prove that $d \ge w_{i+1}$. Hence, $(\bar{x}, \bar{y})$ will be the point (x_{i+1}, y_{i+1}) at which BLF places τ_{i+1}.

Now assume that $d \ge w_{i+1}$. Using the assumption that all conditions hold after packing all items in L_i, it is easy to verify that Conditions (1) and (2) are satisfied after BLF places τ_{i+1} at $(\bar{x}, \bar{y})$ when $d \ge w_{i+1}$. Also, because of Equality (10), Condition (3) holds when $d \ge w_{i+1}$.

Now consider Condition (4), which is trivially true when $\bar{x} = 0$. Otherwise, we know there exists an item $\tau_r : r \le i$ which has been placed at $(x_r, y_r) = (\bar{x} - w_r, y_r)$. Therefore, it holds that

$$\bar{x} = x_r + w_r \overset{(a)}{=} (m_r + 1) \cdot w_r \overset{(b)}{=} m_{i+1} \cdot w_{i+1} \tag{12}$$

where $m_{i+1} \in \mathbb{Z}^+ \cup \{0\}$. Here Equality (a) comes from the induction hypothesis and Equality (b) holds due to the non-increasing geometric order of widths of items. Hence, Condition (4) is always satisfied.

Now we prove that $d \ge w_{i+1}$ is true. We consider two cases. In the case where $\bar{x} + d = W$, since both $\bar{x}$ and W are different integer multiples of w_{i+1}, $d = W - \bar{x} > 0$ is also an integer multiple of w_{i+1}. Hence, d must be no smaller than w_{i+1}.

When $\bar{x} + d < W$, there must exist an item τ_t, $t \le i$, which has been placed at $(\bar{x} + d, y_t)$. Therefore, it holds that

$$\bar{x} + d = m_t \cdot w_t = m \cdot w_{i+1}, \quad m \in \mathbb{Z}^+ \cup \{0\}. \tag{13}$$

Comparing (12) with (13), it is clear that $m > m_{i+1}$ since $\bar{x} + d > \bar{x}$. Therefore, we have

$$d \;=\; (m - m_{i+1}) \cdot w_{i+1} \;\overset{(c)}{\ge}\; w_{i+1},$$

where the inequality (c) comes from that $m > m_{i+1}$ and both m and m_{i+1} are integers.

Thus, all the four conditions hold after BLF packs τ_{i+1} at $(x_{i+1}, y_{i+1}) = (\bar{x}, \bar{y})$, i.e., after BLF packs all the items in L_{i+1}. This completes the proof of Lemma 2. □

D. Proof of Theorem 2

Consider any item τ_i packed at (x_i, y_i). By Condition (3) in Lemma 2, $x_i = m_i \cdot w_i$ where m_i is a non-negative integer and w_i is the width of τ_i. For any x that is covered by τ_i, i.e., $x_i \le x \le x_i + w_i - 1$, by Equation (6), its most significant q_i digits remain the same as those of x_i where $w_i = \eta^{q_K - q_i}$. By Equation (7), their reversed numbers x' and x'_i must have the same least significant q_i digits, or equivalently

$$x' \equiv x'_i \mod \eta^{q_i} \;\Rightarrow\; t_b \times x' \equiv t_b \times x'_i \mod t_i.$$

where $t_i = t_b \cdot \eta^{q_i}$.

This effectively means that the activation time of the frame instance corresponding to x must be equivalent to that of the one corresponding to x_i (modulo t_i). Given that we have in total $w_i = t_h / t_i$ such instances, they must form the set of all frame instances in a hyperperiod (i.e., of length t_h) with the same offset $(t_b \times x'_i) \mod t_i$. Hence, it is a valid offset assignment to the corresponding signal σ_i. □

E. Proof of Theorem 3

Let $H \triangleq \max_{1 \le x \le W} f_\kappa(x)$ denote the height of the solution for the list L_κ given by BLF, and suppose the height H is achieved by packing the item τ_r at (x_r, y_r); that is,

$$f_\kappa(x) \;=\; H \;= y_r + g_r, \quad \forall x : x_r < x < x_r + w_r.$$

According to the condition (3) of Lemma 2, because BLF is always able to pack τ_r on the minimum height, we have

$$H - g_r \;=\; \min_{0 \le x < W} f_{r-1}(x) \;\le\; \min_{0 \le x < W} f_\kappa(x). \tag{14}$$

From Lemma 3 and Lemma 4, we get

$$H = (H - g_r) + g_r \leq 2 \times H^* \tag{15}$$

which means that the performance bound of BLF is at most 2. □

REFERENCES

[1] K. Tindell, H. Hansson, and A. Wellings, "Analysing real-time communications: controller area network (CAN)," in *IEEE Real-Time Systems Symposium*, 1994.

[2] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller area network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, no. 3, pp. 239–272, 2007.

[3] U. Bordoloi and S. Samii, "The frame packing problem for CAN-FD," in *IEEE Real-Time Systems Symposium*, December 2014.

[4] F. Hartwich, "CAN with flexible data-rate," in *13th International CAN Conference*, 2012.

[5] M. Di Natale, C. Silva, and M.Santos, "On the applicability of an MILP solution for signal packing in CAN-FD," in *IEEE International Conference on Industrial Informatics*, July 2016.

[6] G. Urul, "A frame packing method to improve the schedulability of CAN and CAN-FD," Ph.D. dissertation, Middle East Technical Univ., Turkey, 2015.

[7] K. Sandstrom, C. Norstom, and M. Ahlmark, "Frame packing in real-time communication," in *International Conference on Real-Time Computing Systems and Applications*, 2000.

[8] *DBC Communication Database for CAN*, Vector, [Online] https://vector.com/vi_candb_en.html.

[9] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., 1979.

[10] P. Joshi, S. Ravi, S. Samii, U. D. Bordoloi, S. Shukla, and H. Zeng, "Offset assignment to signals for improving frame packing in CAN-FD," in *IEEE Real-Time Systems Symposium*, 2017, pp. 167–177.

[11] J. Kleinberg and E. Tardos, *Algorithm Design*. Boston, MA: Pearson Education Inc., 2006.

[12] D. S. Hochbaum and D. B. Shmoys, "Using dual approximation algorithms for scheduling problems: Theoretical and practical results," *J. ACM*, vol. 34, no. 1, pp. 144–162, Jan. 1987.

[13] P. Joshi, H. Zeng, U. Bordoloi, S. Samii, S. S. Ravi, and S. Shukla, "The multi-domain frame packing problem for CAN-FD," in *Euromicro Conference on Real-time Systems*, June 2017.

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

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

[16] M. Lukasiewycz, M. Glaß, J. Teich, and P. Milbredt, "FlexRay schedule optimization of the static segment," in *IEEE/ACM international conference on Hardware/software codesign and system synthesis*, 2009.

[17] H. Zeng, M. Di Natale, A. Ghosal, and A. Sangiovanni-Vincentelli, "Schedule optimization of time-triggered systems communicating over the FlexRay static segment," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 1, pp. 1–17, 2011.

[18] B. Tanasa, U. D. Bordoloi, P. Eles, and Z. Peng, "Reliability-aware frame packing for the static segment of FlexRay," in *ACM international conference on Embedded software*, 2011.

[19] M. Kang, K. Park, and M.-K. Jeong, "Frame packing for minimizing the bandwidth consumption of the FlexRay static segment," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4001–4008, 2013.

[20] A. Darbandi, S. Kwon, and M. K. Kim, "Scheduling of time triggered messages in static segment of FlexRay," *International Journal of Software Engineering and its Applications*, vol. 8, no. 6, pp. 195–208, 2014.

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

[22] R. L. Graham, "Bounds for certain multiprocessing anomalies," *Bell Labs Technical Journal*, vol. 45, no. 9, pp. 1563–1581, 1966.

[23] ——, "Bounds on multiprocessing timing anomalies," *SIAM Journal on Applied Mathematics*, vol. 17, no. 2, pp. 416–429, 1969.

[24] A. Lodi, S. Martello, and M. Monaci, "Two-dimensional packing problems: A survey," *European Journal of Operational Research*, vol. 141, no. 2, pp. 241–252, 2002.

[25] J. F. Oliveira, A. Neuenfeldt Júnior, E. Silva, and M. A. Carravilla, "A survey on heuristics for the two-dimensional rectangular strip packing problem," *Pesquisa Operacional*, vol. 36, no. 2, pp. 197–226, 2016.

[26] B. S. Baker, E. G. Coffman, Jr, and R. L. Rivest, "Orthogonal packings in two dimensions," *SIAM Journal on computing*, vol. 9, no. 4, pp. 846–855, 1980.

[27] B. Chazelle, "The bottomn-left bin-packing heuristic: An efficient implementation," *IEEE Transactions on Computers*, no. 8, pp. 697–707, 1983.

[28] M. Grenier, L. Havet, and N. Navet, "Pushing the limits of CAN-scheduling frames with offsets provides a major performance boost," in *European Congress on Embedded Real Time Software*, 2008.

[29] J. Palencia and M. Harbour, "Schedulability analysis for tasks with static and dynamic offsets," in *IEEE Real-Time Systems Symposium*, 1998.

[30] V. V. Vazirani, *Approximation Algorithms*. New York, NY: Springer, 2001.

[31] S. Kramer, D. Ziegenbein, and A. Hamann, "Real world automotive benchmark for free," in *Workshop on Analysis Tools and Methodologies for Embedded and Real-Time Systems*, 2015.

[32] *CPLEX Optimization Studio*, IBM, [Online] http://www-01. ibm. com/software/integration/optimization/cplex-optimization-studio.

[33] H. Zeng and M. Di Natale, "An efficient formulation of the real-time feasibility region for design optimization," *IEEE Transactions on Computers*, vol. 62, no. 4, pp. 644–661, 2013.