On the applicability of an MILP solution for signal packing in CAN-FD

Marco Di Natale
Scuola Superiore S. Anna
Email: marco.dinatale@sssup.it

Celso Luiz Mendes da Silva
Federal University of Technology - Paraná
Email: celsosilva@alunos.utfpr.edu.br

Max Mauro Dias Santos
Federal University of Technology - Paraná
Email: maxsantos@utfpr.edu.br

***Abstract*—The new CAN-FD standard for automotive communication provides support for real-time predictability at an increased rate and with a larger payload than the existing CAN standard. The adoption of CAN-FD requires the migration and possibly the redesign of the existing message sets to exploit the features of the protocol, by repacking the application signals in the new larger frames. Previous works provided algorithms that consider timing constraints and try to reduce the required bandwidth. The state of the art solutions are however based on two-step heuristics. A single-step formulation, with a suitable set of (optional) bound relaxation techniques has the potential to provide better solutions (with lower bus utilization) and also be more easily adapted to the consideration of the constraints and limitations that typically apply to the signal remapping .**

I. Introduction

The Controller Area Network with Flexible Data Rate (CAN-FD) is an improvement on the CAN 2.0 protocol [1] as defined in the ISO 11898-1 standard [2]. It enables higher speeds than traditional CAN when transmitting the data part, but is designed to be backward compatible. The physical and data link layers can be shared by the message formats of both protocols in such a way that any CAN-FD node can receive messages sent from a CAN node. This property allows for a smooth transition from CAN to CAN-FD.

The CAN-FD frame allows the transmission of payloads up to 64 bytes, as opposed to the maximum 8 bytes of CAN and an increased baud rate of up to 8 Mbit/s for the bits following the arbitration stage (the message identifier). The larger payload can be exploited only by redesigning (entirely or in part) the message set. Finding a new packing of signals into CAN-FD frames is not a trivial problem, indeed, it is an instance of a variable size bin-packing.

The *Controller Area Network with Flexible Data Rate* works with two bit rates: one for the arbitration field and the other for the data field. The arbitration bit rate, also called normal bit rate, is applied until the BRS bit (included) and after the CRC delimiter bit (excluded). The data bit rate starts from the ESI field up to the CRC delimiter and allows up to 8 Mbits/s, instead of the 1 Mbit/s of the standard CAN.

The EDL (Extended Data Length) field defines the data length with the highest bit at zero if the message size is 0 to 8 bites (0000 to 1000) and using the remaining 7 combinations with the highest bit at 1 to encode the lengths 8, 12, 16, 20, 24, 32, 48, and 64 (1000 to 1111).

As in CAN, the arbitration field defines the message that wins the contention (lowest identifiers have highest priority).

Similar to CAN, in CAN-FD, a stuffing bit is added to the frame every time there is a sequence of 5 bits of the same value (bit stuffing rule). However, stuff-bits are only inserted in CAN FD from the SOF until the end of the data field, leaving the ACK and EOF fields without stuff bits. Furthermore, the CRC is stuffed with fixed-position bits [1].

II. Problem formulation

The system under design consists of a single-bus system, in which software tasks in execution on the network nodes need to exchange signals of variable size. The Network is a CAN-FD network and the set of nodes is defined as $\mathcal{N} = \{N_1, N_2 \ldots N_p\}$. The set of signals is $\mathcal{S} = \{s_i\}$, each signal defined as $s_i = \{\sigma_i, t_i, d_i, e_i\}$ where σ_i is the signal length (in bits), t_i is the signal period and d_i is the signal deadline and e_i is the sender ECU for the signal.

The objective of the assignment can be formulated as follows. Define a set of messages $\mathcal{M} = \{m_j\}$ with each $m_j = \{b_j, T_j, \pi_j, \epsilon_j\}$ such that b_j is the size in bytes of the message, with $b_j \in 0, 1, 2, 3, 4, 5, 6, 7, 8, 12, 16, 20, 24, 32, 48, 64$, T_j is the period of the message, π_j is the message priority, and ϵ_j is the source ECU for the message. Define also a signal-to message assignment function

$$\omega_{i,j} = \begin{cases} 1 & \text{if } s_i \text{ is transmitted by } m_j \\ 0 & \text{otherwise} \end{cases}$$

such that the following constraints are satisfied:

- Each signal is assigned to one and only one message.
- The length of each message is allowed by CAN-FD and is greater than or equal to the sum of all the signals assigned to the message. Each message has a unique π_i.
- Each message has a period that is an integer divisor of all the periods of the signals mapped onto it.
- Each message has a source ECU that is the same as that of all its signals.
- Each signal is guaranteed to complete its transmission over the network before its deadline.

For simplicity, the worst-case transmission time of signals is assumed to be the same as the one of the message that is transmitting it. The worst case transmission time of a CAN-FD message m_j is defined by the formula (from [3])

$$W_j = 32t_a + \left(28 + 5\left\lceil\frac{b_i - 16}{64}\right\rceil + 10b_j\right)t_d$$

where t_a is the bit time for the arbitration stage and t_d is the bit time for the data transmission. If the transmission rate is an integer multiple of the arbitration rate, than $t_a = K_t \cdot t_d$.

Given the possible values of b_j, the value of the ceiling term is 0 for $b_j \leq 16$ and 1 for the longer messages. Hence, for $K_d = 8$ (typical value), the worst-case transmission time is

$$W_j = \begin{cases}(284 + 10b_j)t_d & \text{if } b_j \leq 16\\ (289 + 10b_j)t_d & \text{otherwise}\end{cases} \tag{1}$$

The completion (response) time for message frames is approximated (with slight pessimism) by the formula in [4]

$$R_j = q_j + W_j \qquad q_j = B_j + \sum_{k\in hp(j)}\left\lceil\frac{q_j}{T_k}\right\rceil W_k \tag{2}$$

where k spans over all messages with priority higher than m_j, and B_j is simply the time that it takes to transmit the longest frame in the system.

$$B_j = max(W_k) \text{ with } m_k \in \mathcal{M}$$

Among all the message sets and mappings that satisfy the constraints we are interested in the one that results in the lowest bus utilization, defined as

$$U = \sum_j W_j/T_j \tag{3}$$

computed over all the messages with index j.

III. Related Work

The CAN-FD frame packing problem was approached in [3] based on the observation that the problem is an instance of the variable sized bin-packing ([5] and [6]) where messages are bins and their variable frame size is the packing size. The heuristic developed in [3] consists of two main steps. In the first step the application signals are packed into frames so that the utilization is minimized through a subset sum problem in an iterative fashion [3] that considers bandwidth waste (without considering the deadline constraints). In the second stage a heuristic iteratively assigns priorities to messages and computes their worst-case transmission times. The algorithm tries to adjust for late signals by remapping them onto different messages with higher priority.

Uru [7] proposed heuristics for CAN-FD packing using well known techniques as in [8], and [9]. His approach first tries to group signals with the same period into the same frame and then distributes signals to frames with available space while accounting for schedulability.

For regular CAN frames, Sandstrom *et al* [10] showed that CAN frame packing is a NP-hard problem (as a special case of the classical bin packing problem) and propose a polynomial time algorithm based on *classical bin packing solutions* in which signals are tentatively packed together according to their deadline. Poelzlbauer and Brenner [8] developed an heuristic to optimize frame packing as an extension of the solution in [10] for *fixed size messages*. Signals are packed using a *next fit decreasing* policy, considering their periods. In [9], Saket and Navet propose two heuristics to minimize bandwidth utilization and analyze schedulability for messages in CAN networks. The first heuristic is named *Bi-directional Frequency Fit* and groups the signals sorted by their periods, i.e., the signals that have the same period are packed together. If the set of signals has payload size greater than 8 a new frame is created to pack the the signals. This process is repeated until all signals are packed. The second heuristic works by increasing the deadlines of the new frames.

IV. Optimization using MILP

The frame packing problem for CAN-FD may be modeled as a *mixed integer linear programming* (MILP) using relaxations techniques to improve applicability to larger signal sets.

The problem is expressed by aset of integer, binary and continuous variables, and design parameters; linear equalities and inequalities represent the constraints on the solution and the linear objective (performance or cost) function.

For each ECU e_i we should define a set of virtual messages $m_{i,j}$ that contains at least one message for each possible period of the signals originating from e_i, and each possible message size. This results in a very large number of possible messages that are likely unnecessary and would increase the complexity of the formulation. This is the reason for the definition of a reduced (but most likely close to optimal or optimal) set. If the set of all the signals with the same period originating from the same ECU has size larger than 64 then the set is further extended with one additional subset of messages for each period exceeding the size limit of 64 and sizes 12, 16, 20, 24, 32, 48, 64. If the sum of all the signal data with the same period is larger than 80, the set for that period is further extended with a subset with sizes 32, 48, 64. From that point on, the set is extended with an additional number n_{add} of messages with size 64 until the required number of bits from all the signals with the same period is lower than $80 + 64 * n_{add}$. The rationale is that with a larger size of signals, the opportunities for packing increase and the need for small size messages decrease (at most one for the remainders).

The messages from all ECUs are defined as a single set $m_j = \{b_j, T_j, \epsilon_j\}$.

In addition to the binary mapping variable $\omega_{i,j}$ for each pair signal, message; we define for each pair of messages a binary variable

$$\pi_{i,j} = \begin{cases}1 & \text{if } m_i \text{ has priority higher than } m_j\\ 0 & \text{otherwise}\end{cases}$$

Also, for each message m_j, we define a mapping size variable

$$\mu_j = \sum_i \omega_{i,j}\sigma_i \tag{4}$$

and a binary variable z_j defining if the message m_j is empty, the following must be true for all signal indexes i.

$$z_j \geq \omega_{i,j} \text{ for all } i \qquad z_j \leq \sum_i \omega_{i,j} \tag{5}$$

Each constraint can be expressed as follows.

Each signal is assigned to one and only one CAN-FD message.

$$\sum_j \omega_{i,j} = 1 \text{ for all } i \tag{6}$$

The message lenght is greater than or equal to the sum of the bits of all the signals assigned to the message

$$\mu_j \leq b_j \tag{7}$$

Each message has a unique priority level. The partial order of priority $\pi_{i,j}$ must define a global order. That is, the anti-symmetric and transitivity properties must hold.

$$\pi_{i,j} = 1 - \pi_{j,i} \qquad \pi_{i,k} \geq \pi_{i,j} + \pi_{j,k} - 1 \tag{8}$$

Each message has a period that is an integer divisor of all the periods of the signals mapped onto it. By setting $\omega_{i,j} = 0$ if the period of the signal s_i is not an integer multiple of the period of m_j.

Each message has a source ECU that is the same as that of all its signals. Enforced by constraining $\omega_{i,j} = 0$ if the signal s_i and the message m_j have a different source ECU.

Each signal is guaranteed to be transmitted before its deadline. We add a response time variable R_j for each message m_j, and a response time variable r_i for each signal s_i. The deadline requirements for all signals are simply encoded as

$$r_i \leq d_i \tag{9}$$

and the correspondence between signal response times and message response times as

$$r_i \geq R_j \text{ if } \omega_{i,j} = 1 \tag{10}$$

This conditional constraint is encoded with a "big M" formulation as follows.

$$r_i \geq R_j - M * (1 - \omega_{i,j}) \tag{11}$$

Finally, the formulation for the worst-case transmission time of message m_i is

$$W_i = \begin{cases} (284 + 10b_i)t_d & \text{if } b_i \leq 16 \\ (289 + 10b_i)t_d & \text{otherwise} \end{cases} \tag{12}$$

and the worst case response time is upper bound by the formula

$$r_i = W_i + \sum_{k \in hp(i)} \left\lceil \frac{r_i}{T_k} \right\rceil W_k \tag{13}$$

A set of integer variables $y_{i,k}$ define the possible number of interferences of message k on message i (the term $\left\lceil \frac{r_i}{T_k} \right\rceil$.)

$$0 \leq y_{i,k} - r_i/T_k < 1 \tag{14}$$

the number of interferences is only of interest if the priority of message k is higher than the priority of i, another set of variables $w_{i,k}$ addresses this condition

$$y_{i,k} - (1 - \pi_{k,i})M \leq w_{i,k} \leq y_{i,k} \tag{15}$$

$$0 \leq w_{i,k} \leq \pi_{k,i} M \tag{16}$$

Hence, the constraint is linearized as

$$r_i = W_i + \sum_k w_{i,k} W_k \tag{17}$$

The previous formula suffices with one provision. We need to make sure that all the virtual messages that have no content (no signal mapped onto them) are ignored. This is obtained by assigning them the lowest priority levels. For all the pair of message indexes j,k it must be

$$z_j - z_k M \leq \pi_{k,j} \tag{18}$$

A. Dealing with legacy and application-specific constraints

An MILP formulation can be typically easily extended to deal additional constraints related to application needs or the need to incorporate in the problem solution legacy messages.

For example, if the application designer requires that two signals s_i, s_k are mapped onto the same message it suffices to enforce.

$$\forall j \, \omega_{i,j} = \omega_{k,j}$$

B. Complexity and Relaxations

The complexity of the problem formulation is dominated by the integer and binary variables expressing the priority relation, the signal to message mapping and the number of interferences. The number of these variables increases with the square of the message set size or the product of the number of signals by the number of messages.

There are several techniques and relaxations that could be employed to speed up the solver for larger signal sets, at the price of overconstraining the problem and giving up on the guarantee of optimality.

The relaxations that have been tried are the following:

- Even if the rate monotonic priority order is not optimal for message scheduling, it is extremely unlikely that any optimal assignment is too far from it [11]. Hence, we tried adding a constraint that forces message m_j to have a priority higher than message m_k if $T_j \leq T_k \cdot p$, with p constant (variable from 10 down to 1, with 1 forcing the assignment to be Rate Monotonic.
- The mapping of signals to messages may be restricted to only those messages that have a period that is an integer divisor and also no more than k times lower, that is $\omega_{i,j} = 0$ if $T_j < T_i/k$

V. Experimental Results

We could not directly compare with the results in [3] for several reasons. First, the signal set proposed in [3] is simply not suited to a mapping problem. Since one of the features of a good packing algorithm is to pack together signals with periods that are integer multiples, the signal periods cannot be randomly selected. Otherwise, the probability of having two signals with harmonic periods is extremely low.

Similarly, the utilization values shown in [3] clearly do not correspond to the typical values for a CAN-FD network. A simple computation shows that at 8Mb/s a set of 200 signals with size randomly selected between 1 and 14 bytes and period randomly selected between 10 and 5000 ms will never achieve utilizations close to 80% as shown in the paper.

Our signal set is constructed similar to the one in [3] by selecting sets of signals of variable size (from 20 and increasing by 5), with each signal having a period randomly selected in the set $\{5, 10, 20, 50, 100, 200, 500, 1000, 2000, 4000, 5000\}$ (in ms) and a size randomly selected between 1 and 12 bytes.

In our experiment we simulate signals coming all from the same ECU (a local packing problem) and 50 random configurations are generated for each signal set size and each relaxation option.

The runtime of the solver for the different relaxation options is shown in Figure 1. The graph shows the average runtimes (in seconds) for different signals set sizes. The Y-axis is in logarithmic scale, showing with good approximation an exponential growth of the runtime with respect to the problem size. The original formulation, without relaxations, can only handle a problem size of up to 30 signals with a runtime of approximately one hour for each set. If the problem is relaxed, by allowing the mapping of signals to messages only if the period of the signal is no more than 4 times the message period (parameter k) and an inversion of the rate monotonic rule only if the periods are no more than a factor of 5 apart (parameter p), then the solver can handle sets up until size 40. If the rule is further relaxed, with k=2 and p=2, then the maximum size becomes 65. A signal set of size 100 can be handled by enforcing a rate monotonic priority order and allowing mapping signals only to messages with the same period or a period that is 2 times theirs. Further, the CPLEX solver was allowed to stop when arriving within a worst case of 1% of the true optimum (MILP solvers can estimate this distance thanks to the duality theorem).

Finally, a fully relaxed formulation (signals only mapped to same period messages and up to 3% error) shows feasibility up until size 140. Considering that the set of signals with true real-time constraints is typically only a subset of all signals, the method appears applicable to the design of CAN-FD message sets with respect to its runtime.

With respect to the quality of the results, the set of experiments that relaxed the priority order definition and the signal to message mapping (but not the quality of the solution, labeled as relaxed3 in Figure 1) was computed to be in the average *within 0.3% in terms of relative error with respect to the true optimum utilization* (computed by the exact formulation), with only 34 runs out of 150 showing a solution different from the optimum.

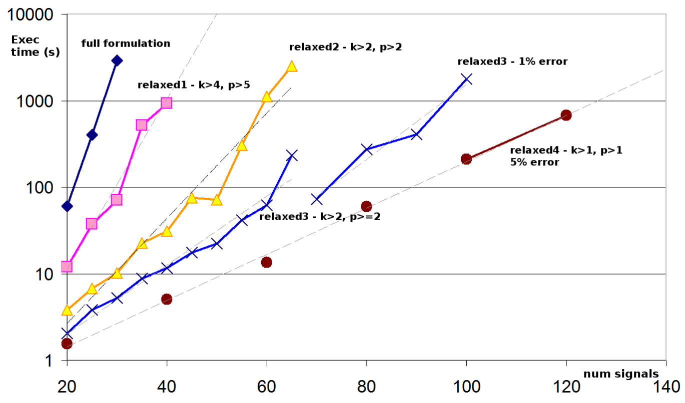

Figure 1. Runtime of the algorithm with respect to the number of signals and the relaxation options

The final set, with full relaxation (relaxed4 in Figure 1) was in the average within 3.5% of the solutions found by the relaxed formulation denominated relaxed3. Considering the 3% error tolerance that is allowed to CPLEX to speed up the computation, it appears that the impact of the relaxations on the priority and the signal to message mapping (for large sets) have very little effect on the quality of the result.

VI. Conclusions

We presented an MILP formulation for the signal to frame packing problem in CAN-FD. The MILP formulation allows to target at once the timing constraints and the optimization goals, without a two step heuristics. Results show applicability to medium sized problems.

References

[1] Vector, "Introduction to CAN." [Online]. Available: https://elearning.vector.com

[2] F. Data-rate, "CAN with Flexible Data-Rate," vol. 0, 2012.

[3] U. D. Bordoloi and S. Samii, "The Frame Packing Problem for CAN-FD," *2014 IEEE Real-Time Systems Symposium*, no. Section III, pp. 284–289, 2014.

[4] 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.

[5] M. Haouari and M. Serairi, "Computers & Operations Research Heuristics for the variable sized bin-packing problem," vol. 36, 2009.

[6] J. Kang and S. Park, "Algorithms for the variable sized bin packing problem," *European Journal of Operational Research*, vol. 147, no. 2, pp. 365–372, 2003.

[7] G. Uru, "A Frame Packing Method To Iomprove The Schedulability On CAN And CAN-FD," Master's thesis, 2015.

[8] F. Poelzlbauer and E. Brenner, "Optimized Frame Packing for Embedded Systems," *IEEE Embedded Systems Letters*, p. 3, 2012.

[9] R. Saket and N. Navet, "Frame packing algorithms for automotive applications," *Journal of Embedded Computing*, vol. 2, no. 1, pp. 93–102, 2006. [Online]. Available: http://iospress.metapress.com/index/b1vk8xbkhan96e41.pdf

[10] K. Sandstrom, "Frame packing in real-time communication," *Real-Time Computing ...*, pp. 399–403, 2000. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=896418

[11] M. Di Natale and A. Meschi, "Scheduling messages with earliest deadline techniques," *Real-Time Systems*, 2001.