Joint real-time scheduling and interference coordination for wireless factory automation

Sébastien Auroux
Paderborn University
33098 Paderborn, Germany
E-mail: sebastien.auroux@upb.de

Donald Parruca
Paderborn University
33098 Paderborn, Germany
E-mail: donald.parruca@upb.de

Holger Karl
Paderborn University
33098 Paderborn, Germany
E-mail: holger.karl@upb.de

***Abstract*—The fifth generation of wireless communication is expected to enable various new use cases that today's wireless communication systems are not able to support. One of these use cases is mission-critical machine-type communication (C-MTC) for automation processes in factory environments. The real-time requirements including firm deadlines for such communication impose very low latency and high reliability demands that can currently only be fulfilled by wired solutions. In this paper, we focus on the radio resource management aspect to enable C-MTC in the context of a factory automation scenario by proposing a joint real-time scheduling and interference coordination approach that supports low latency communication while provisioning radio resources for very high reliability.**

I. INTRODUCTION

Factory automation use-cases are characterized by high requirements on latency and reliability with application demands going down to end-to-end latencies of 1 ms and block error rates of 10^{-9} [1]. While wired solutions are able to meet these requirements, the opportunity for cost savings and mobility offered by wireless technologies brings a strong motivation to replace cabled connections with wireless communication links, but current wireless technologies provide insufficient guarantees on reliability and latency. In particular, the corresponding interference coordination mechanisms are optimized mainly for backlogged and non-real-time traffic [2], which has diametrically different requirements than the traffic type that is dominating factory automation. Additionally, the spectrum in current wireless industrial automation technologies is managed in a very conservative way by totally avoiding intra-system interference. For example, the WirelessHART standard does not allow reusing the same sub-carrier in the entire deployed network [3]. For large deployments, this approach quickly runs out of bandwidth. In special, this approach is overly conservative if several production cells are totally isolated from each other, e.g. from metallic walls, or the production cells are quite far apart from each other. In such a case, reusing frequencies would be acceptable. But up to now, such information is not exploited at all in current technologies. Hence, deployment-aware and interference-tolerant approaches are required to ensure reliable and efficient wireless communication in industrial automation systems.

To meet the low latency requirements the concept of *real-time scheduling* constitutes a promising approach. Real-time scheduling algorithms schedule data transmissions while making sure ahead of time that all traffic deadlines will be met during operation time. However, existing real-time scheduling concepts are mainly based on the assumption that a task can be assigned to any processor and preempted at any time without any constraints [4]. Unfortunately, this is not the case for wireless communications. In fact, communication over the frequency domain can be subject to deep-fades making certain sub-carriers inappropriate for reliable communication. Hence, new real-time scheduling approaches considering frequency fading channels need to be developed.

In this paper, we address both these issues. We present a two-step radio resource management approach to enable low latency wireless communication while guaranteeing high reliability. Our approach consists first of a local real-time scheduling step, which schedules communication streams exclusively on reliable communication channels while ensuring that their deadlines are met. This scheme is combined with a global interference coordination approach which uses the information obtained from the local step to realize conflict-free frequency assignments and to prevent inter-cell interference.

To present our approach, we first elaborate on related work in Section II before giving an overview of our work's background architecture in Section III. We then describe the local real-time scheduling step in Section IV and the global interference coordination step in Section V. For both, we provide reference optimization models and heuristic algorithms to solve them. Moreover, we evaluate all methods in Section VI and conclude our work in Section VII.

II. RELATED WORK

Current technologies widely used in wireless factory automation systems are ZigBee, ISA100.11a, WirelessHART and IEEE 802.11x. These technologies operate in the non-licensed ISM band, making them susceptible to inter-network interference arising from the co-existence of multiple networks in the same frequency band. The coexistence problem between WirelessHART and IEEE 802.11x networks has been addressed in [5] and the coexistence of IEEE 802.11 WLAN and IEEE 802.15 WPAN networks has been addressed in [6]. Both of these works suggest a two-stage approach. First, a long-term spectrum measurement is performed, before the transceivers are tuned to frequency channels with minimal interference level. Unfortunately, such measurements take relatively long, making the interference coordination schemes unable to react quickly to unexpected short-term interference.

To avoid this problem, licensed spectrum technologies, where the network operator has exclusive control of the intra-network interference, are a candidate for reliable communication. But current licensed wireless networks such as LTE and LTE-A provide too loose guarantees on reliability and latency for factory automation requirements. Although high reliability is a challenging goal to achieve [7], evaluations [8], [9] show that if intra-system interference can be kept under control, a considerable amount of bandwidth can be saved. The framework introduced in [10] proposes a power-control and rate-adaption algorithm for C-MTC control applications that minimizes the time required for concurrent transmissions. However, this work is limited to single-carrier systems and does not consider fading in the frequency domain. Another framework [11] performs a timing analysis for hard real-time systems employing retransmissions due to an ARQ mechanism. Although ARQ schemes help to improve reliability, it is to be expected that for low-latency applications there is not enough time for acknowledgements [8]. Additionally, no interference coordination aspects have been considered in this work.

To the best of our knowledge, no other work has been presented so far performing joint real-time scheduling and interference coordination in highly reliable and low-latency systems deployed in frequency-fading environments.

III. BACKGROUND ARCHITECTURE

This work has been conducted in the context of the KoI project [12]. In this section, we provide a short overview of its industrial background scenario and its two-layer coordination hierarchy, which is adopted by our approach.

As part of a wireless communication concept for industrial environments, the project proposes a two-layer hierarchy for radio resource coordination:

- several *Local Radio Coordinators (LRC)*, have a limited scope but take fast, short-time decisions (down to below 1 ms) based on fine-grained local information and,
- a single *Global Radio Coordinator (GRC)*, operates on a broader scope (e.g. the complete factory hall) and makes longer-time decisions based on global, but coarse-grained information provided by the LRCs.

The operation area of an LRC is called a *local cell*, containing several *devices*, e.g. sensors or actuators, which are coordinated by the LRC. While the GRC and the LRCs communicate via conventional LTE, each LRC and all devices within one local cell communicate over a special, low-latency air interface developed within KoI. This air interface employs Orthogonal frequency-division multiplexing (OFDM) and the LTE frame structure, scaled by a certain factor as suggested for low-latency communication in 5G [1]. This kind of scaling allows to reduce the Transmission Time Interval (TTI) at the cost of a larger bandwidth consumption (also see Section VI-A). Figure 1 provides an exemplary illustration of the KoI coordination architecture.

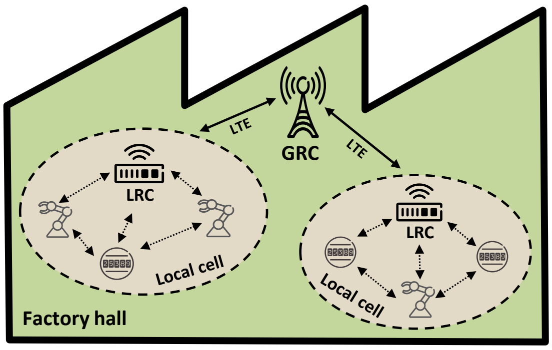


Figure 1. Illustration of the background architecture

While we are assuming the use of the KoI architecture in this paper, our approach is also applicable without any adaptation to any other kind of system design providing a two-tier coordination architecture and a slotted multi-carrier air interface, e.g. FDMA or OFDMA systems.

IV. LOCAL REAL-TIME SCHEDULING

Local real-time scheduling is regularly performed by each LRC within its local cell in the order of tens to hundreds of milliseconds, for which we assume the channel states to be consistent. Relying on a background architecture as in Section III, we assume each LRC to be aware of all traffic requirement and recent channel state information (CSI) of all links within its cell. Based on this, the LRC determines sets of *potential channels*, so that each set allows for real-time scheduling of the traffic inside its cell.

A. Problem statement

For a given local cell, we consider a set C of *wireless channels* (or *resource blocks* in LTE terminology) and a set P of periodic data streams whose source and destination are either the LRC or a device from the local cell. Each data stream $p \in P$ generates a packet of $\mathrm{ps}(p)$ bits every $\mathrm{per}(p)$ seconds with a relative deadline of $\mathrm{D}(p) \leq \mathrm{per}(p)$ seconds. In addition, we consider a connection matrix $K \in \{0,1\}^{|P|\times|C|}$, obtained by assessing the available CSI with regard to each data stream's reliability requirements, with $K_{p,c} = 1$ if and only if $c \in C$ allows a *reliable communication* for $p \in P$.

Based on this, we aim to generate a transmission schedule for a time frame of length T seconds, where T is the least common multiple of $\{\mathrm{per}(p), p \in P\}$, using a minimum amount of channels while not missing any deadline of any packet from any stream. As a multiprocessor scheduling problem with one resource, the problem is NP-hard [13].

B. Optimization model

Our local real-time scheduling optimization model is an Integer Linear Program (ILP). It takes the parameters listed in Table I as inputs (corresponding to Section IV-A) and the variables listed in Table II to store the scheduling assignments.

The parameters $\mathrm{dur}(p)$, $p \in P$ are obtained via preprocessing by taking into account the packet sizes $\mathrm{ps}(p)$ and other factors (see Section VI-A). Similarly, the parameters $\mathrm{n}(p)$, $p \in P$ are determined as $\mathrm{n}(p) = \frac{T}{\mathrm{per}(p)}$.

Table I
LOCAL REAL-TIME SCHEDULING: ILP INPUT PARAMETERS

C	set of available wireless channels
P	set of data streams (within the local cell)
K	matrix with $K_{p,c} = 1$ iff $p \in P$ can be scheduled on $c \in C$
$\mathrm{N_s}$	amount of time slots in time frame T
$\mathrm{T_s}$	size of a time slot
$\mathrm{dur}(p)$	amount of time slots needed for each packet of stream $p \in P$
$\mathrm{per}(p)$	arrival period of stream $p \in P$
$\mathrm{D}(p)$	deadline for each packet of stream $p \in P$
$\mathrm{n}(p)$	amount of packets of $p \in P$ in time frame T

Table II
LOCAL REAL-TIME SCHEDULING: ILP VARIABLES

$x_{p,i,c,k} \in \{0, 1\}$	determines whether the i-th packet of $p \in P$ is scheduled on $c \in C$ in time slot k
$z_c \in \{0, 1\}$	determines whether channel $c \in C$ is used at all

The objective of this optimization problem to find the minimum amount of wireless channels that allows to schedule the traffic of all communication pairs within the local cell (1).

$$\min \sum_{c \in C} z_c \tag{1}$$

The following constraints have to be ensured. Some constraints use a big-M constant denoted as $\mathcal{M}$. At first, we need to ensure the validity of the z variables (2).

$$\sum_{p \in P} \sum_{i=1}^{\mathrm{n}(p)} \sum_{k=1}^{\mathrm{N_s}} x_{p,i,c,k} \leq \mathcal{M} \cdot z_c, \quad \forall c \in C \tag{2}$$

All packets need to be scheduled for exactly their amount of required time slots (3) and only on channels where they can be scheduled (4).

$$\sum_{c \in C} \sum_{k=1}^{\mathrm{N_s}} x_{p,i,c,k} = \mathrm{dur}(p), \quad \forall p \in P, i \in \{1, \ldots, \mathrm{n}(p)\} \tag{3}$$

$$x_{p,i,c,k} \leq K_{p,c}, \quad \begin{matrix} \forall p \in P, i \in \{1, \ldots, \mathrm{n}(p)\}, \\ c \in C, k \in \{1, \ldots, \mathrm{N_s}\} \end{matrix} \tag{4}$$

Of course, each time slot of each channel can only be used at most once (5).

$$\sum_{p \in P} \sum_{i=1}^{\mathrm{n}(p)} x_{p,i,c,k} \leq 1, \quad \forall c \in C, k \in \{1, \ldots, \mathrm{N_s}\} \tag{5}$$

At last, each packet must be scheduled between its arrival time (6) and all of its scheduled time slots must end before its deadline is reached (7).

$$\forall p \in P, i \in \{1, \ldots, \mathrm{n}(p)\}, c \in C, k \in \{1, \ldots, \mathrm{N_s}\} :$$

$$(1 - x_{p,i,c,k}) \cdot \mathcal{M} + x_{p,i,c,k} \cdot (k-1) \cdot \mathrm{T_s} \geq (i-1) \cdot \mathrm{per}(p), \tag{6}$$

$$x_{p,i,c,k} \cdot k \cdot \mathrm{T_s} \leq (i-1) \cdot \mathrm{per}(p) + \mathrm{D}(p). \tag{7}$$

C. Heuristic EDF scheduling algorithms

To perform local real-time scheduling within the envisioned order of tens of milliseconds, we propose two heuristic algorithms. Both are based on Earliest Deadline First (EDF) scheduling and use the input parameters from Table I.

The first heuristic (Algorithm 1) schedules complete data streams on certain channels, so that a valid scheduling to individual time slots is guaranteed by a deterministic EDF post-processing step for each channel. First, EDFSTREAMSCHED sorts both P and C so that data streams with only few available channels are scheduled first and so that channels that are available to many data streams are considered first. Next, it attempts to schedule each data stream on the most utilized wireless channels it can be scheduled on. Both the non-preemptive EDF schedulability test TESTEDFSCHEDULABILITY and the EDF utilization function UTILEDF are taken from Liu [4].

Algorithm 1 EDFSTREAMSCHED(C)

```
sort $P$ by $\sum_{c \in C} K_{p,c}$ ascending
sort $C$ by $\sum_{p \in P} K_{p,c}$ descending
for $c \in C$ do
    util($c$) = 0
for $p \in P$ do
    $C_p = \{c \in C,\ K_{p,c} = 1\}$
    sort $C_p$ by util($c$) descending
    for $c \in C_p$ do
        if TESTEDFSCHEDULABILITY($c, p$) = True then
            schedule $p$ on $c$
            util($c$) = UTILEDF($c$)
            break
return $\{c \in C,\ \mathrm{util}(c) > 0\}$
```

In contrast to Algorithm 1, our second EDF-based heuristic (Algorithm 2) schedules packets to individual time slots, similarly to the optimization model from Section IV-B. EDFPACKETWISESCHED first itemizes all packets of all data streams and sorts them according to their absolute deadline. Next, all channels are sorted as in Algorithm 1. Then, EDFPACKETWISESCHED schedules each packet (p, i) by determining the suitable time slots, starting from the packet's generation time to its deadline, and all suitable channels with available slots in this time frame. At last, dur(p) time slots are reserved for each packet.

Algorithm 2 EDFPACKETWISESCHED(C)

```
$P^* = \{(p, i)$ for $p \in P,\ i = 1, \ldots, \mathrm{n}(p)\}$
sort $P^*$ by $(i-1) \cdot \mathrm{per}(p) + \mathrm{D}(p)$ // absolute deadline
sort $C$ by $\sum_{p \in P} K_{p,c}$ descending
for $c \in C, t \in \{1, ..., \mathrm{N_s}\}$ do
    util($c, t$) = 0
for $(p, i) \in P^*$ do
    $T_{\mathrm{suit}} = \{(i-1) \cdot \mathrm{per}(p), ..., (i-1) \cdot \mathrm{per}(p) + \mathrm{D}(p) - 1\}$
    $C_{(p,i)} = \{c \in C,\ K_{p,c} = 1,\ \sum_{t \in T_{\mathrm{suit}}} \mathrm{util}(c, t) < |T_{\mathrm{suit}}|\}$
    sort $C_{(p,i)}$ by $\sum_{t \in T_{\mathrm{suit}}} \mathrm{util}(c, t)$ descencing
    for $k \in \{1, ..., \mathrm{dur}(p)\}$ do
        for $c \in C_{(p,i)}, t \in T_{\mathrm{suit}}$ do
            if util($c, t$) = 0 then
                schedule $k$-th part of $(p, i)$ for slot $t$ of $c$
                util($c, t$) = 1
                break
return $\{c \in C,\ \sum_{t \in T} \mathrm{util}(c, t) > 0\}$
```

D. Potential channel set generation

In this section, we describe how we use the aforementioned approaches to generate various potential channel sets as an input for the global interference coordination step (Section V).

The procedure generating a certain amount of potential channel sets is described in Algorithm 3 with the general idea to exclude certain channels for each run. For the first potential channel set, we exclude no channel. Then, to generate further sets, we exclude a random amount of at most $\frac{|C|}{2}$ random channels per run. To increase the solution diversity, we have further implemented the *dualCalc* procedure, which tries to double the amount of potential channel sets by attempting a scheduling for each so far found schedule, excluding exactly the channels used in it. At last, the generic procedure SCHEDULE is a placeholder for any of the three aforementioned approaches.

Algorithm 3 GENPOTSETS(amount, dualCalc)

```
if dualCalc = True then
    $n = 2 \cdot$ amount
else
    $n$ = amount
$C_{\text{pot}} = \{\}$
for $k \in \{1, ..., n\}$ do
    if $|C_{\text{pot}}| = 0$ then
        $C_{\text{used}} = C$
    else if $|C_{\text{pot}}| <$ amount then
        $b =$ RANDINT$(1, \frac{|C|}{2})$
        $C_{\text{ban}}$ = RANDSAMPLE(C,$b$)
        $C_{\text{used}} =$ C $\setminus C_{\text{ban}}$
    else if $|C_{\text{pot}}| \geq$ amount then
        $C_{\text{used}} =$ C $\setminus C_{\text{pot}}[k$ - amount$]$
    $C_{\text{pot}}$.add(SCHEDULE($C_{\text{used}}$))
return $C_{\text{pot}}$
```

It is important to note, that each scheduling run with any approach always either results in a *valid schedule*, i.e. a schedule for all data streams where all deadline are met, or in a failed attempt that adds no new potential channel set.

V. GLOBAL INTERFERENCE COORDINATION

In the context of our architectural background, the GRC performs global interference coordination based on the potential channel sets determined by the LRCs (see Section IV-D). Drawing the link to Section IV, the assigned channels are then used by the LRCs to perform real-time scheduling in their local cells without suffering from inter-cell interference.

A. Problem statement

We consider a factory hall as an undirected graph $G = (V, E)$ with nodes V, representing the local cells, and links E, indicating resource conflicts, e.g. because of potential inter-cell interference. If $(v, w) \in E$, any channel may only be used in either v or w but not in both. Further, each node $v \in V$ provides a set $C_{\text{pot}}(v)$ of potential channel sets.

The goal of this step is to assign to each local cell one of its channel sets so that all nodes v, w with $(v, w) \in E$ do not use any common wireless channel. The objective function, which simultaneously serves as a performance metric, aims to minimize the sum of the lengths of all selected channel sets. As the NP-complete graph coloring problem can obviously be reduced to it, this step is NP-hard [14].

B. Optimization model

The global interference coordination optimization model is an Integer Linear Program (ILP). It takes the parameters listed in Table III as inputs, which correspond to the problem definition in Section V, and uses the variables listed in Table IV to store the frequency assignments.

Table III
GLOBAL INTERFERENCE COORDINATION: ILP INPUT PARAMETERS

V	set of nodes (equaling local cells)
E	set of conflict links
C	set of available wireless channels
$C_{\text{pot}}(v) \subseteq 2^C$	potential channel sets for $v \in V$

Table IV
GLOBAL INTERFERENCE COORDINATION: ILP VARIABLES

$x_{v,c} \in \{0, 1\}$	determines if $v \in V$ uses $c \in C$
$z_{v,S} \in \{0, 1\}$	determines if $v \in V$ uses all $c \in S$, $S \in C_{\text{pot}}(v)$

The objective of the optimization problem (8) corresponds to the description in Section V-A.

$$\min \quad \sum_{v \in V} \sum_{c \in C} x_{v,c} \tag{8}$$

The following constraints have to be ensured. As in Section IV-B, some constraints use a big-M constant denoted as $\mathcal{M}$. At first, all conflict links need to be considered (9).

$$x_{v,c} + x_{w,c} \leq 1, \quad \forall (v, w) \in E, c \in C \tag{9}$$

Also, the z variables need to be set (10) to determine if a cell uses a potential channel set.

$$z_{v,S} \leq x_{v,c}, \quad \forall v \in V, S \in C_{\text{pot}}(v), c \in S \tag{10}$$

At last, each local cell must be assigned with at least one complete potential channel set (11).

$$\sum_{S \in C_{\text{pot}}(v)} z_{v,S} \geq 1, \quad \forall v \in V \tag{11}$$

C. Heuristic greedy algorithm

To provide fast frequency allocations, we propose a heuristic algorithm based on a greedy approach (Algorithm 4).

Algorithm 4 COLORINGGREEDY()

```
$Scheduled = \{\}$
for $v \in V$ do
    $C_{\text{ass}}(v) = \{\}$
while $|Scheduled| < |V|$ do
    $candidates = V \setminus Scheduled$
    sort $candidates$ by $|C_{\text{pot}}(v)|$ ascending
    sort $candidates$ by amount of non-scheduled neighbors descending
    $v = candidates[0]$
    $C_{\text{good}} = \{S \in C_{\text{pot}}(v), S \cap C_{\text{ass}}(w) = \{\} \, \forall \, w \in N(v)\}$
    sort $C_{\text{good}}$ descending by 1. max-min amount, 2. total amount of disjoint potential channel sets among all non-scheduled neighbors
    $C_{\text{ass}}(v) = C_{\text{good}}[0]$
```

COLORINGGREEDY iteratively assigns potential channel sets to each $v \in V$. The next node is always the one with fewest potential channel sets among the nodes with the most non-scheduled neighbors. After selecting a node v, the algorithm determines all $S \in C_{\mathrm{pot}}(v)$ that are disjoint with the assigned sets of already scheduled neighbors. For each of those, ColoringGreedy then determines the number of disjoint potential channel sets for all non-scheduled neighbors of v. It then assigns that $S \in C_{\mathrm{good}}(v)$ to v that leaves the maximal minimum amount, in case of a tie the highest total amount, of potential channel sets to each non-scheduled neighbor. By this, we aim to avoid the choice of a potential channel that leaves non-scheduled neighbors in an infeasible state.

VI. EVALUATION

In this section, we present evaluation results for our local real-time scheduling and our global interference coordination approaches. Our evaluation is twofold: we first evaluate the local step approaches in Section VI-A and then analyse the performance of the global step approaches in Section VI-B.

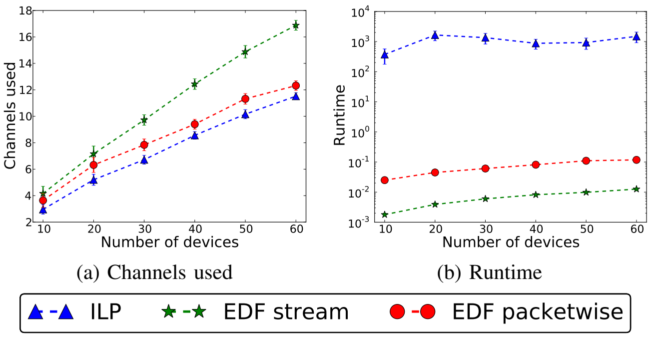
All evaluations are executed on Intel Xeon X5650 CPUs running at 2.67 GHz. The optimization models have been implemented using the Pyomo package for optimization modeling in Python [15] and solved with Gurobi 6.0.4 [16] running in single-threaded mode and all heuristic algorithms were implemented in Python. All plots contain confidence intervals at a 95% level unless they are too small and covered by the plot markers.

A. *Local real-time scheduling results*

To compare the local optimization model and the two local heuristics, we generated 150 local cells, 25 each with 10, 20, ..., 60 devices. Each cell size equals the number of devices in metres and the LRC is placed in its center, while the devices are positioned uniformly at random in the cell. Corresponding to the architecture of Section III, we generate two data streams for each device, one from the LRC to the device and one from the device to the LRC. Each device is capable of device-to-device (D2D) communication with probability 0.5 and we generate D2D-streams between two D2D-capable devices if their distance is equal to or less than 5 metres.

Each data stream p is generated with a period $\mathrm{per}(p)$ uniformly at random chosen from $\{1, 2, 3, 4, 6, 8\}$ ms, resulting in $T \leq 24$ ms, and a deadline $\mathrm{D}(p)$ uniformly at random chosen from $\{1, ..., \mathrm{per}(p)\}$ ms. Further, we set $\mathrm{ps}(p) = 100$ bits for all streams. Next, we assume a scaling factor, as described in Section III, of 5, which results in a channel bandwidth of 900 kHz and a time slot size (TTI) of $\mathrm{T_s} = 0.2$ ms [1]. As modulation and coding scheme we assume QPSK with a coding rate of $\frac{1}{2}$, adding up to a bitrate of $2 \cdot \frac{1}{2} \cdot 12 \cdot 14 = 168$ bits per time slot, so that $\mathrm{dur}(p) = 1$ for all data streams.

The total available bandwidth for this evaluation part has been set to 50 Mhz, giving $|C| = \lfloor \frac{50\text{Mhz}}{900\text{kHz}} \rfloor = 55$ wireless channels. Regarding the generation of the K matrix, we assume that all communication links provide a satisfying quality under a simple path loss model but are vulnerable to frequency fading. For each data stream p, we generate frequency fading values independently for all channels $c \in C$ using a simulated Rayleigh fading channel in Matlab [17] and set $K_{p,c} = 0$ if the fading value is below $0.5 \approx -3$ dB, $K_{p,c} = 1$ otherwise.



(a) Channels used

(b) Runtime

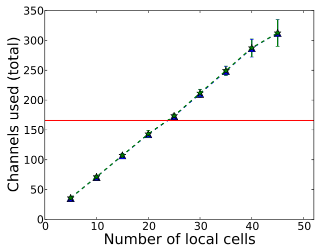
Figure 2. Evaluation: Local optimization model vs. heuristic approaches

The results of this evaluation are depicted in Figure 2, averaged for each number of devices. While all three solution methods manage to find valid schedules for all generated local cells, Figure 2a shows that the gap between the EDF stream heuristic and the optimization model's optimal solution increases with the number of devices per cell. Meanwhile, the EDF packetwise heuristic manages to stay close to the optimal solution. In turn, Figure 2b illustrates that the EDF stream heuristic runs about one order of magnitude faster than the EDF packetwise heuristic, which then again needs about 4 orders of magnitude less runtime as the optimization model. Hence, both heuristics fit to our envisioned execute cycles of tens to hundreds of milliseconds.

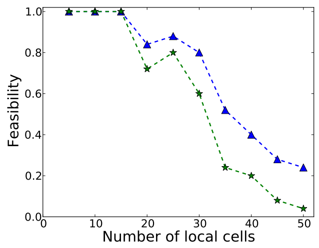
B. *Global interference coordination results*

To compare the results of the global optimization model and the greedy heuristic, we generated 500 factory hall scenarios. 250 scenarios each are based on the EDF stream heuristic and on the EDF packetwise heuristic, with 5, 10, ..., 50 local cells, 25 for each number. All local cells are generated as described in Section VI-A with a number of devices uniformly and randomly chosen from $\{10, ..., 30\}$ and a random position, but without overlap, in a factory hall of 1000 m × 1000 m. Conflict links between two local cells have been generated if and only if the distance between the cell borders is less than or equal to 200 m. For this evaluation part, we assume a total bandwidth of 150 Mhz, resulting in $|C| = \lfloor \frac{150\text{Mhz}}{900\text{kHz}} \rfloor = 166$ wireless channels. As for the potential channel sets, we have generated up to 30 sets per local cell using GenPotSets with amount $= 15$ and *dualCalc* $=$ True (see Section IV-D).

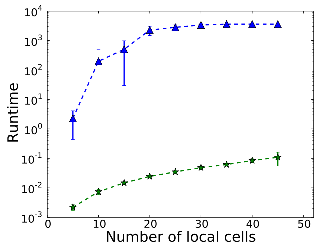
Figure 3 illustrates the results of this evaluation part with the results based on the EDF stream heurstic and the EDF packetwise heuristic being on the left and right side respectively. First, Figures 3a and 3b depict the total amount of channels used in the factory halls. The first thing to note is that the greedy heuristic achieves very good results, consistently being extremely close to the optimization model's optimal solution. Both plots also include a horizontal line indicating $|C| = 166$, forming an upper feasibility bound for alternate approaches that deny any kind of frequency reuse, such as WirelessHART. To understand this, one needs to recall from Section V-A that the global step's objective function accounts for the use of each wireless channel *multiple times*, if it is used in multiple local cells. Any approach that does not allow frequency reuse however, would hence not be able to pass this line. At last, one can observe that the instances using the EDF stream heuristic require more resources, since the potential channel sets are expected to be larger (Figure 3a vs. 3b, cf. Figure 2a).



(a) Objective value (stream) (b) Objective value (packetwise)



(c) Feasibility (stream) (d) Feasibility (packetwise)



(e) Runtime (stream) (f) Runtime (packetwise)

Figure 3. Evaluation: Global optimization model vs. greedy heuristic

Next, Figures 3c and 3d show the feasibility of the generated instances. Given that the optimization model solves optimally, it can be guaranteed that the instances it does not manage to solve are indeed infeasible. The most important fact to notice by comparing both plots is the effect of the local step's solution quality on the feasibility on the global level. While only 40% of the instances with 40 local cells based on EDF stream scheduling are solvable, the same number for EDF packetwise scheduling is above 90%. It can also be seen that the greedy heuristic is not able to solve all feasible instances, especially for many local cells, but the results are still satisfactory.

At last, the runtimes depicted in Figures 3e and 3f do not seem to depend much on the used local scheduling algorithm and show that the greedy heuristic beats the optimization model by several orders of magnitude. It also has to be noted that the global optimization model's runtime had been limited to one hour per run, which has almost consistently been reached for instances with many local cells. Again, we find that our proposed heuristic algorithm perfectly fits to our envisioned execute cycles, thus confirming the feasibility of our approach.

VII. CONCLUSION AND FUTURE WORK

We presented an approach jointly performing local real-time scheduling and global interference coordination to enable low-latency wireless communication with high reliability as demanded in use cases such as factory automation. In contrast to existing approaches, our work takes into account both frequency fading on a local scope and deployment awareness on a global scope. As a result, we obtain reliable real-time scheduling assignments and increase the feasibility of scenarios with bandwidth scarcity. To enable the use of our approach in practice, we proposed efficient and fast heuristic algorithms with near-optimal results.

As for future work, we will further improve our methods and consider approaches such as power management to extend our model and improve its solution quality.

ACKNOWLEDGMENT

This work has been partially funded by the German Federal Ministry of Education and Research (BMBF) within the project KoI (grant identifier 16KIS0198).

REFERENCES

[1] S. Ashraf, F. Lindqvist, R. Baldemair, and B. Lindoff, "Control Channel Design Trade-Offs for Ultra-Reliable and Low-Latency Communication System," in *2015 IEEE Globecom Workshops*, 2015, pp. 1–6.

[2] D. Parruca, M. Grysla, H. Zhou, F. Naghibi, M. Petrova, P. Mähönen, and J. Gross, "On semi-static interference coordination under proportional fair scheduling in LTE systems," in *Proceedings of the 2013 19th European Wireless Conference (EW)*, 2013, pp. 1–8.

[3] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Real-time scheduling for WirelessHART networks," in *IEEE 31st Real-Time Systems Symposium (RTSS)*. IEEE, 2010, pp. 150–159.

[4] J. Liu, *Real-time systems*. Prentice Hall, 2000.

[5] S. Petersen and S. Carlsen, "Performance evaluation of WirelessHART for factory automation," in *IEEE Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE, 2009, pp. 1–9.

[6] L. Voskov and S. Tsyganov, "The interference problem for wireless networks in automation systems of industrial enterprises: The statement and a solution method," *Automation and Remote Control*, vol. 74, no. 12, pp. 2194–2201, 2013.

[7] P. Popovski, "Ultra-reliable communication in 5G wireless systems," in *IEEE 1st International Conference on 5G for Ubiquitous Connectivity (5GU)*, 2014, pp. 146–151.

[8] N. Johansson, Y. Wang, E. Eriksson, and M. Hessler, "Radio access for ultra-reliable and low-latency 5G communications," in *IEEE International Conference on Communication Workshop (ICCW)*, 2015, pp. 1184–1189.

[9] N. Brahmi, O. Yilmaz, K. Helmersson, S. Ashraf, and J. Torsner, "Deployment Strategies for Ultra-Reliable and Low-Latency Communication in Factory Automation," in *IEEE Globecom Workshops*, 2015, pp. 1–6.

[10] B. Farayev, Y. Sadi, and S. C. Ergen, "Optimal Power Control and Rate Adaptation for Ultra-Reliable M2M Control Applications," in *IEEE Globecom Workshops*, 2015, pp. 1–6.

[11] M. Jonsson and K. Kunert, "Towards Reliable Wireless Industrial Communication With Real-Time Guarantees," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 4, pp. 429–442, 2009.

[12] "Koordinierte Industriekommunikation (KoI)," http://www.koi-projekt.de/, German Federal Ministry of Education and Research (BMBF).

[13] M. R. Garey and D. S. Johnson, "Complexity results for multiprocessor scheduling under resource constraints," *SIAM Journal on Computing*, vol. 4, no. 4, pp. 397–411, 1975.

[14] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations*. Springer, 1972, pp. 85–103.

[15] W. E. Hart, "Python optimization modeling objects (pyomo)," in *Operations Research and Cyber-Infrastructure*. Springer, 2009, pp. 3–19.

[16] "Gurobi Optimization," http://www.gurobi.com/.

[17] MATLAB Communications System Toolbox $^{\text{TM}}$, "User's Guide (R2015b)," 2015.