# Dynamic Resource Allocation for Energy Efficient Transmission in Digital Subscriber Lines

Nan Zhang, Zhi-Qiang Yao, *Senior Member, IEEE*, Yixian Liu, Stephen P. Boyd, *Fellow, IEEE*, and Zhi-Quan Luo, *Fellow, IEEE*

***Abstract*—Linear matrix precoding, also known as vectoring, is a well-known technique to mitigate multiuser interference in the downlink digital subscriber line (DSL) transmission. While effective in canceling interference, vectoring does incur major communication overhead and computational overhead in terms of the transmission of idle symbols and precoder-data multiplications at each data frame, resulting in significant energy consumption when the number of lines is large. To facilitate energy efficient transmission, it has been recently proposed (in the G.fast standard) that each data frame is divided into a normal operating interval (NOI) and a discontinuous operating interval (DOI). In the NOI, all lines (or users) are involved in a common vectoring group, which requires a large matrix precoder, whereas in the DOI, the lines are subdivided into multiple small nonoverlapping vectoring subgroups and are transmitted in a time division multiple access manner, requiring small matrix multiplications and, thus, improving the energy efficiency. In this paper, we consider the key dynamic resource allocation problems in downlink DSL: given the real-time demands, determine the optimal transmission scheme: The optimal NOI and DOI size in each data frame as well as the optimal grouping strategy in the DOI, and optimally adjust the transmission scheme. We formulate these optimal dynamic resource allocation problems and propose efficient real-time algorithms to solve them to global optimality. Simulation results are shown to demonstrate the efficiency and the effectiveness of the proposed algorithms.**

***Index Terms*—DSL, resource allocation, discontinuous operation, energy efficiency, dynamic programming.**

Manuscript received October 14, 2016; revised March 11, 2017 and May 9, 2017; accepted May 12, 2017. Date of publication June 2, 2017; date of current version June 21, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yue Rong. The work of Z.-Q. Yao is supported in part by NSFC under Grant 61372127 and in part by the Key Discipline of Hunan Province and Key Laboratory of Intelligent Computing & Information Processing, Xiangtan University, Ministry of Education, China. The work of Z.-Q. Luo is supported in part by NSFC under Grant 61571384 and in part by the Leading Talents of Guangdong Province program under Grant 00201510. Parts of this paper have also been presented in Globecom 2016.
*(Corresponding author: Zhi-Qiang Yao.)*

N. Zhang is with the School of Mathematical Sciences, Peking University, Beijing 100871, China (e-mail: zhangnan625@pku.edu.cn).

Z.-Q. Yao is with the College of Information Engineering, Xiangtan University, Xiangtan 411105, China (e-mail: yaozhiqiang@xtu.edu.cn).

Y. Liu is with the Huawei Technologies Company, Ltd., Shenzhen 518129, China (e-mail: liuyixian@huawei.com).

S. P. Boyd is with the Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen, Shenzhen 518172, China, and also with Stanford University, Stanford, CA 94305 USA (e-mail: boyd@stanford.edu).

Z.-Q. Luo is with the Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen, Shenzhen 518172, China, and also with the University of Minnesota, Minneapolis, MN 55455 USA (e-mail: luozq@ece.umn.edu).

Digital Object Identifier 10.1109/TSP.2017.2711513

## I. Introduction

Digital Subscriber Line (DSL) technology has been a popular way to provide wireline broadband access over twisted pair copper wires of telephone networks. Recently, a new G.fast (Fast Access to Subscriber Terminals, G.9701) recommendation [1] has emerged as a pathway to transit the copper access into the gigabit era [2]. The centerpiece of this proposed gigabit DSL technology is the Distribution Point Unit (DPU) which serves as a gateway from the end-users to the service provider's network. In practice, DPUs are usually reverse powered by the end-users' premises. As a result, energy efficiency is a very important consideration in the design of DPU [3].

As is well known, the electromagnetic interference among lines in the same cable bundle, or the so called crosstalk, is a major concern in DSL. This is especially true for the high frequency channels in G.fast where the crosstalk is very strong [4]. Linear matrix precoding, also known as vectoring, is an effective technology to eliminate crosstalk in the downlink DSL [5]–[7]. The application of vectoring technology can significantly boost the achievable data rate to over 100 Mb/s when using VDSL2 (very-high-bit-rate digital subscriber line 2) in FTTX (fiber-to-the-x) networks [7]. However, to enable vectoring, the vectored lines must operate as a group synchronously over the same operation interval with the same starting and ending time. For example, consider the case where $K$ lines are vectored as a group. Let $\mathbf{x} \in \mathbb{C}^{K\times 1}$ be the transmitted data vector, $\mathbf{P} \in \mathbb{C}^{K\times K}$ be the precoder matrix for the group, $\mathbf{H} \in \mathbb{C}^{K\times K}$ be the channel matrix and $\mathbf{n}$ be the noise vector, then the received data vector $\mathbf{y} \in \mathbb{C}^{K\times 1}$ in downlink transmission is

$$\mathbf{y} = \mathbf{HPx} + \mathbf{n}. \tag{I.1}$$

Here $\mathbf{Px}$ is exactly the signal after vector processing. To eliminate crosstalk, the precoder $\mathbf{P}$ should be chosen to make $\mathbf{HP}$ diagonal. For example $\mathbf{P}$ can be chosen as a scaled version of $\mathbf{H}^{-1}$. Notice that even if a line $j$ has no data of its own to transmit ($\mathbf{x}_j = 0$), it still must turn on its transceiver. This is because the precoder output of line $j$ ($(\mathbf{Px})_j$) containing crosstalk signals of other lines must still be transmitted, for otherwise the received data $\mathbf{y}$ would not be interference free. Therefore, by inserting "idle" symbols if necessary, all the transceivers of the lines in a vectored group must be turned on in the common operation interval even if some of the lines do not have any actual data to send.

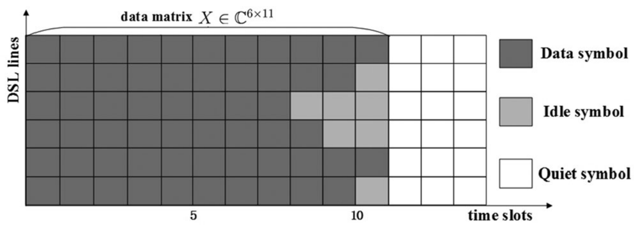



Fig. 1. A data frame with only NOI.

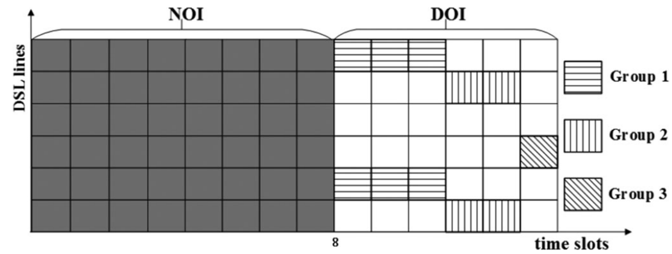



Fig. 2. A feasible DO scheme of the case in Figure 1.

Like other digital communication systems, the DSL data is transmitted in a time slotted manner. A data frame is the basic transmission unit consisting of a fixed number of symbol positions, where one symbol is one time slot. In G.fast, there are basically three kinds of symbols: data symbols during which actual data is transmitted, idle symbols during which no data is transmitted but the transceiver is still working, quiet symbols during which the transceiver is turned off. To save transmit power, the number of idle symbols should be as small as possible. An example is shown in Figure 1, where each square represents one symbol position and there are six lines and fourteen symbol positions in the data frame. Let $d_i$ be the number of transmission opportunities, then $d_1 = 10, d_2 = 11$. In this example, seven idle symbols are inserted. Besides inserted idle symbols, another source of major power consumption is the vector processing step where the data matrix $\mathbf{X}$ (see Figure 1) is multiplied by the precoder matrix $\mathbf{P}$ during the transmission of each data frame. This precoder-data matrix multiplication incurs significant energy consumption when the number of lines is large.

### A. Discontinuous Operation (DO)

To enhance the energy efficiency, G.fast has proposed a mechanism called Discontinuous Operation (DO). With the introduction of DO, each data frame is further divided into two transmission intervals: a Normal Operating Interval (NOI) and a Discontinuous Operating Interval (DOI). In the NOI, all lines are involved in a common vectoring group. In the DOI, the lines with remaining untransmitted data symbols are grouped into non-overlapping subgroups, and vector processing is performed within each subgroup so that crosstalk cancellation is maintained among the members in each subgroup. To eliminate cross-subgroup interference, the data transmission in the DOI are arranged in a TDMA (Time Division Multiple Access) manner. A feasible DO scheme of the case in Figure 1 is given in Figure 2: the first eight time slots constitute NOI during which all six lines are vectored together, the remaining six time slots constitute DOI during which the transmission of the three subgroups are scheduled in sequence. In this way, no idle symbols are needed in the transmission of user data in this data frame. Furthermore, notice that the precoder matrix for each subgroup has a smaller size than that of the large precoder used in NOI for all the lines. Thus, the total computation cost of the precoder-data matrix multiplications in NOI and all subgroups is smaller than what would be required if all lines are vectored in a single group. In this way, the energy efficiency is significantly improved. In the examples considered above (Figures 1 and 2), the effect of DO is obvious. Without DO, vectoring requires seven idle symbols and multiplying the precoder $\mathbf{P}_0 \in \mathbb{C}^{6\times 6}$ with the data matrix $\mathbf{X} \in \mathbb{C}^{6\times 11}$. In contrast, using the DO strategy described in Figure 2, no idle symbols are needed; moreover the number of multiplications is reduced: multiply $\mathbf{P}_0 \in \mathbb{C}^{6\times 6}$ with the data matrix $\mathbf{X}_0 \in \mathbb{C}^{6\times 8}$ during NOI, and multiply the precoder $\mathbf{P}_1 \in \mathbb{C}^{2\times 2}$ with the data matrix $\mathbf{X}_1 \in \mathbb{C}^{2\times 3}$ in subgroup 1 of DOI, the precoder $\mathbf{P}_2 \in \mathbb{C}^{2\times 2}$ with the data matrix $\mathbf{X}_2 \in \mathbb{C}^{2\times 2}$ in subgroup 2, the precoder $\mathbf{P}_3 \in \mathbb{C}$ with the data matrix $\mathbf{X}_3 \in \mathbb{C}$ in subgroup 3. Since multiplying $n \times n$ with $n \times m$ complex matrices requires $4n^2m$ scalar multiplications, the number of scalar multiplications is reduced from 1584 to 1236 with the introduction of DO.

### B. Design and Adjustment of DO Scheme

For frames with different demands, the optimal DO patterns (in the sense of having the minimum energy in vectoring) will be different. In this paper, DO pattern refers to the parameters that fully describe the DO scheme: the duration of NOI, the grouping strategy in DOI, and the transmission duration of each subgroup in DOI. To design DO transmission, we need to consider several issues:

1) given the number of data symbols of all lines, partition the frame as NOI and DOI;
2) assign each line with remaining untransmitted data symbols to a subgroup;
3) arrange the transmission interval for each subgroup in DOI in a TDMA manner.

Notice that once the grouping strategy changes, the precoders for the new subgroups must be computed accordingly. However, computing a new set of precoders and configuring the parameters are time consuming. Consequently, it is infeasible to adjust precoders and grouping strategy at frame rate. To reduce energy cost without changing grouping strategy frequently, we consider adjusting the durations of NOI and the subgroups in DOI in response to the traffic flow condition (or demands), while still maintaining the use of the existing DOI grouping strategy. Consider the example in Figure 3 where the number of data symbols are changed from those in Figure 2. If using the same DO pattern as Figure 2, it is infeasible to arrange the data transmission in a TDMA manner, because the sum transmission duration of the three subgroups exceeds the duration of DOI. To enable feasible transmission, either the durations of NOI and the subgroups need to be modified or the grouping strategy need

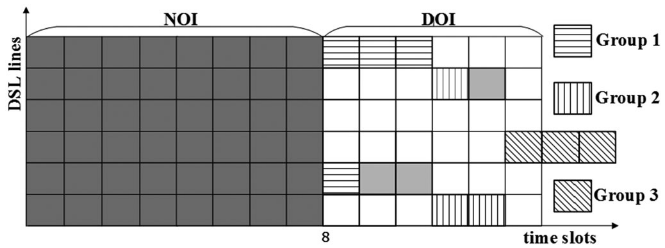



Fig. 3. The previous DO scheme becomes infeasible when demands change.

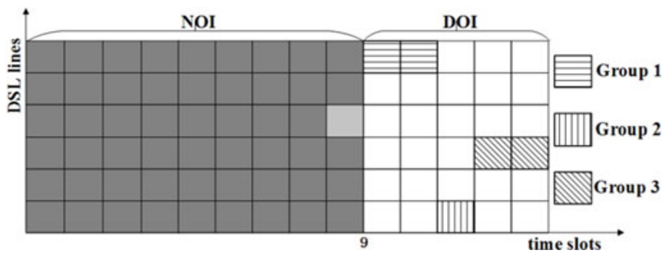



Fig. 4. A feasible DO scheme is obtained by only adjusting the NOI duration.

to be re-designed. In this case, if keeping grouping strategy unchanged while increasing the NOI duration by one symbol, a feasible DO scheme is obtained at the cost of one inserted idle symbol in NOI, as Figure 4 shows. Such adjustment requires no change in vectoring precoders. Of course, significant changes in traffic condition may require the re-design of the DO grouping strategy. In practice, it is natural to adjust the durations of NOI and the DOI subgroups while maintaining the grouping strategy in the short time scale, and to design the new grouping strategy in the long time scale.

### C. Existing Works

Many studies focused on optimizing the transmit power of DSL users [8]–[10], while few works discussed scheduling problems related to DO as G.fast standard has just been approved recently. Reference [11] considered DO with only one subgroup in DOI, disregarding the time shifting arrangement. It also ignored the effect of the group operating time on the energy consumption in a data frame by just assigning two constants to the energy consumption in the operating state and the sleep state respectively. To enhance the energy efficiency, it proposed a method of muting precoding output; however there exists residual crosstalk in that way and the benefits of vectoring are lost. Another method was proposed in [11] where a virtual precoder input is introduced. While the effectiveness of vectoring is maintained in this way, it does require recalculation which is a significant drain on energy.

### D. Summary of Contributions

In this paper, we study the real-time resource allocation problems for next generation DSL systems: (1) given the real-time demands, determine the optimal NOI duration as well as the optimal grouping strategy in DOI; (2) optimally adjust the durations of NOI and DOI subgroups while keeping the grouping strategy unchanged. We present novel formulations for these problems and propose efficient real-time algorithms that can solve the formulated integer optimization problems to global optimality. In particular, we formulate the problem of DO pattern design as a nonlinear integer program. By exploiting the problem structure, we propose an efficient dynamic programming (DP) algorithm to compute the global optimal solution of this problem. Moreover, we formulate the problem of DO pattern adjustment as an integer linear program, and derive a closed form optimal solution of this problem. In this work, we focus on network layer scheduling problems and do not consider explicitly the physical layer elements. To our knowledge, these are the first mathematical formulations and algorithms for the optimal dynamic resource allocation in downlink DSL. We report numerical results to verify the efficiency of the proposed algorithms.

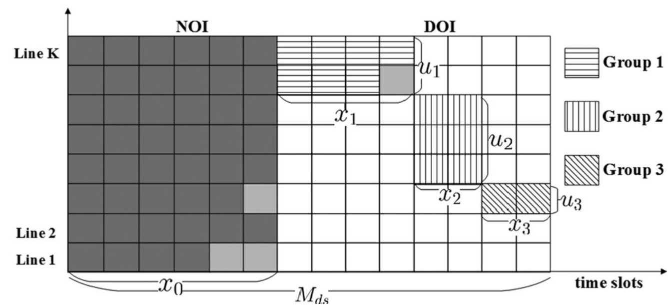



Fig. 5. An illustration of variables.

## II. Design of DO Pattern

### A. Problem Formulation

In this section, we consider designing the optimal DO pattern in a data frame. Consider a system with a total of $K$ lines. Each line $i$ has a pre-allocated demand $\hat{d}_i$, which is measured by the number of symbol positions. We assume the number of data symbols to be transmitted equals the given demand, i.e., $d_i = \hat{d}_i$. Let $M_{ds}$ denote the length (the number of symbol positions) of the data frame, $x_0$ denote the length of NOI, and $L$ denote the total number of subgroups in DOI. For subgroup $\ell$, let $u_\ell$ be the group size (the number of members), $x_\ell$ be the group length (the transmission duration of the members in the DOI). See Figure 5 for an illustration. The summary of the notations are listed in Table I.

The total operating time should not exceed the duration of the data frame due to the TDMA arrangement in DOI. This implies that the sum of group lengths should not exceed the length of DOI:

$$x_1 + \cdots + x_L \leq M_{ds} - x_0. \tag{II.1}$$

The total number of active symbols in a data frame is $Kx_0 + \sum_{\ell=1}^{L} u_\ell x_\ell$, while the total number of data symbols is $\sum_{i=1}^{K} d_i$. Their difference is the number of idle symbols. According to the definition, the energy cost of an idle symbol can be estimated by the consumed energy of a working transmitter during one idle time slot. Assume each idle symbol consumes an equal amount

TABLE I
SUMMARY OF NOTATIONS

| | |
|---|---|
| $K$ | number of lines |
| $M_{ds}$ | number of symbol positions per data frame |
| $\hat{d}_i$ | (given) pre-allocated demand of line $i$ |
| $d_i$ | number of data symbols (transmission opportunity) of line $i$ |
| $L$ | number of subgroups in DOI |
| $x_0$ | length of NOI |
| $x_\ell$ | group length of subgroup $\ell$ in DOI |
| $u_\ell$ | group size of subgroup $\ell$ in DOI |
| $a_i$ | index of the subgroup to which line $i$ is assigned |
| $p_s$ | energy consumption of an idle symbol |
| $p_c$ | energy consumption of multiplying two scalars |
| $\beta$ | constant $4p_c/p_s$ |
| $S_\ell$ | state when the first $\ell$ subgroups have been formed |
| $T(S_\ell, u)$ | achieved state moving from $S_\ell$ via a decision $u$ |
| $R(S_\ell, u)$ | stage cost of moving from $S_\ell$ via a decision $u$ |
| $D(S_\ell)$ | decision space at state $S_\ell$ |
| $f^*(S_\ell)$ | the minimum cost of moving from $S_\ell$ to a target state |

of energy $p_s$ (Watts), then the consumed energy of transmitting the idle symbols can be expressed by

$$p_s \left( Kx_0 + \sum_{\ell=1}^{L} u_\ell x_\ell - \sum_{i=1}^{K} d_i \right). \tag{II.2}$$

Another major source of energy consumption is the multiplication operation. Note that multiplying two matrices $A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{n\times m}$ requires $4n^2m$ scalar multiplication operations and $n(4n-2)m$ addition operations.[1] Since the energy cost of one addition is much cheaper than that of one multiplication, we ignore the cost of addition operations. Let $p_c$ (Watts) denote the consumed energy required to multiply two scalars. Notice that the size of the precoder matrix $P_\ell$ for subgroup $\ell$ is $u_\ell \times u_\ell$ and the size of the data matrix $X_\ell$ is $u_\ell \times x_\ell$. Then the total energy consumption of multiplying the precoder and data matrices for all groups in a data frame can be expressed by

$$p_c \left( 4K^2 x_0 + \sum_{\ell=1}^{L} 4u_\ell^2 x_\ell \right). \tag{II.3}$$

Then the total energy consumption in vector processing is the sum of (II.2) and (II.3), i.e.,

$$P = p_s \left( Kx_0 + \sum_{\ell=1}^{L} u_\ell x_\ell - \sum_{i=1}^{K} d_i \right) + 4p_c \left( K^2 x_0 + \sum_{\ell=1}^{L} u_\ell^2 x_\ell \right).$$

Let $\beta = \frac{4p_c}{p_s}$. Then without loss of generality, we can replace the above cost function by

$$P = \left( Kx_0 + \sum_{\ell=1}^{L} u_\ell x_\ell - \sum_{i=1}^{K} d_i \right) + \beta \left( K^2 x_0 + \sum_{\ell=1}^{L} u_\ell^2 x_\ell \right). \tag{II.4}$$

Suppose any line can only join at most one subgroup in DOI. We introduce an integer vector $\mathbf{a} = (a_1, \ldots, a_K)^T$ to denote the grouping strategy, where $a_i \in \{0, 1, \ldots, L\}$ is the index of the subgroup to which line $i$ is assigned. Here we set $a_i$ to 0 if line $i$ transmits data only during NOI. If $a_i \neq 0$, then $d_i > x_0$ and the total length of subgroup $a_i$ is $x_0 + x_{a_i}$, thus

$$d_i \le x_0 + x_{a_i}, \ \text{ for any } i \text{ with } d_i > x_0. \tag{II.5}$$

By the definitions of $u_\ell$, $a_i$ and $L$, we have following constraints:

$$u_\ell = \text{the number of } i\text{'s with } a_i = \ell, \quad \forall\, \ell, \tag{II.6}$$

$$L = \max_i a_i, \quad 0 \le a_i \le L, \quad \forall\, i. \tag{II.7}$$

Suppose the transmission order of subgroups in DOI makes no difference, then the DO pattern can be completely described by $(x_0, \mathbf{a}, \mathbf{x})$.

Suppose the transmission opportunity is given by the pre-allocated demand vector $\mathbf{d} = \hat{\mathbf{d}}$, we wish to minimize the energy consumption $P$ in (II.4), which includes both the communication cost (transmitting the idle symbols) and computational cost (multiplying the precoder and data matrices). The corresponding DO pattern design problem is given below.

$$\begin{aligned} &\text{minimize} \quad P \\ &\text{subject to} \quad (\text{II.1}),\ (\text{II.5}),\ (\text{II.6}),\ (\text{II.7}), \\ &\qquad\qquad x_0, L, x_\ell, u_\ell, a_i \in \mathbb{Z}_+, \quad \forall\, i,\ \ell. \end{aligned} \tag{II.9}$$

### B. A Dynamic Programming Algorithm for DO Pattern Design

Recall that the vector $\mathbf{a}$ indicates the grouping strategy. As all the variables are integers, and the constraints (II.5)–(II.7) are rather complex, directly searching for the optimal solution $(x_0^*, \mathbf{a}^*, \mathbf{x}^*, \mathbf{u}^*, L^*)$ of problem (II.8) is not easy. Our algorithm is built on the analysis of the problem structure. Specifically, the vector $\mathbf{a}$ is in fact closely related to $\mathbf{u}$ when $x_0$ is fixed, and $\mathbf{a}^*$ can be obtained by searching for $\mathbf{u}^*$. This is illustrated in Corollary 1. The following lemma presents a crucial insight into the relationship between group length vector $\mathbf{x}$ and transmission opportunity vector $\mathbf{d}$.

*Lemma 1:* There is an optimal solution of the DO pattern design problem (9) with the property that

$$d_i > d_j > x_0 \implies x_{a_i} \ge x_{a_j}. \tag{II.9}$$

In other words, in this optimal solution, a line with larger transmission opportunity is placed in a subgroup in DOI with longer group length.

*Proof:* Suppose $(x_0, \mathbf{a}, \mathbf{x}, \mathbf{u}, L)$ is an optimal solution of problem (9) such that there exist two lines $i, j$ with $d_i > d_j > x_0$ and $x_{a_i} < x_{a_j}$. We will show that exchanging the group indices of lines $i$ and $j$ will result in another optimal solution. Without loss of generality, we assume $a_i = 1, a_j = 2$. Let us exchange their group indices and define another assignment $\mathbf{a}'$ (see Figure 6):

$$a_i' = 2,\ a_j' = 1, \text{ and } a_k' = a_k,\ \forall\, k \neq i,\ j.$$

Denote the group length vector and group size vector corresponding to $\mathbf{a}'$ by $\mathbf{x}', \mathbf{u}'$. Since $d_j < d_i \le x_1 + x_0 < x_2 + x_0$,

[1] Notice that zero elements in the matrices do not contribute to computational cost. To simplify the problem and make it tractable, we do not consider this factor in the counting.

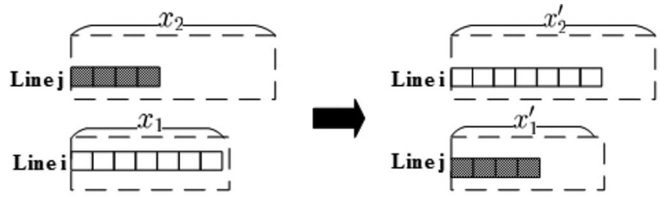



Fig. 6. Proof of Lemma 1: exchange the group indices of lines $i$ and $j$.

by the definition of $\mathbf{a}'$ we have

$$\mathbf{u}' = \mathbf{u}, \tag{II.10}$$

$$x'_1 \le x_1, \quad \text{and } x'_\ell = x_\ell, \forall\, \ell \ne 1. \tag{II.11}$$

It follows that $\mathbf{a}'$ is a feasible grouping vector.

Let us compare the values of $P$ under $\mathbf{a}$ and $\mathbf{a}'$:

$$\begin{aligned} & P(x_0, \mathbf{a}', \mathbf{x}', \mathbf{u}', L) - P(x_0, \mathbf{a}, \mathbf{x}, \mathbf{u}, L) \\ &= \left(\sum_{\ell=1}^{L} u'_\ell x'_\ell + \beta \sum_{\ell=1}^{L} u'^2_\ell x'_\ell\right) - \left(\sum_{\ell=1}^{L} u_\ell x_\ell + \beta \sum_{\ell=1}^{L} u^2_\ell x_\ell\right) \\ &= ((1+\beta u'_1)u'_1 x'_1 + (1+\beta u'_2)u'_2 x'_2) \\ &\quad - ((1+\beta u_1)u_1 x_1 + (1+\beta u_2)u_2 x_2) \\ &= (1+\beta u_1)u_1(x'_1 - x_1) \le 0. \end{aligned}$$

In this way, we obtain a feasible assignment $\mathbf{a}'$ such that the corresponding $\mathbf{x}'$ satisfies the property (II.9) and $P(x_0, \mathbf{a}', \mathbf{x}', \mathbf{u}', L) \le P(x_0, \mathbf{a}, \mathbf{x}, \mathbf{u}, L)$. Hence, $(x_0, \mathbf{a}', \mathbf{x}', \mathbf{u}', L)$ is an optimal solution of problem (II.8) satisfying the property (II.9). ■

*Corollary 1:* Given a demand vector $\mathbf{d}$ and a fixed $x_0$, there exists a way of mapping the group size vector $\mathbf{u}$ to the grouping vector $\mathbf{a}$ and the group length vector $\mathbf{x}$ such that $(\mathbf{a}, \mathbf{x}, \mathbf{u})$ is a feasible solution of problem (II.8) as long as $\mathbf{u}$ is feasible.

*Proof:* By re-indexing the lines if necessary, we assume without loss of generality that the entries of $\mathbf{d}$ are ordered:

$$d_i \ge d_j, \quad \forall\, i < j.$$

Let $\mathbf{e} = (\mathbf{d} - x_0)_{++} = (e_1, \ldots, e_{K(x_0)})^T$, where the operator $(\mathbf{z})_{++}$ retains the positive entries of $\mathbf{z}$ in the original order and $K(x_0)$ is the length of $\mathbf{e}$. Lemma 10 implies that in an optimal grouping, the lines with larger demand are assigned to subgroups with longer lengths, i.e., $d_i > d_j > x_0$ implies $x_{a_i} \ge x_{a_j}$. Thus, for any integer vector $\mathbf{u} = (u_1, \ldots, u_L)^T \in \mathbb{Z}^L_{++}$ that $\sum_{\ell=1}^{L} u_\ell = K(x_0)$, the grouping vector $\mathbf{a}$ can be uniquely determined as follows:

$$a_i = \ell, \quad \text{for } i = U_{\ell-1} + 1, \ldots, U_\ell, \tag{II.12}$$

where $U_\ell := \sum_{j=1}^{\ell} u_j$ and $U_0 := 0$. See Figure 7. The group length can also be determined from

$$x_\ell = \max_{j: a_j = \ell} e_j = e_{U_{\ell-1}+1}, \forall\, \ell. \tag{II.13}$$

As long as the constraint $x_0 + \sum_{\ell=1}^{L} x_\ell \le M_{ds}$ is satisfied, the grouping strategy given by (13) will be feasible. In this way, the group size vector $\mathbf{u}$ determines a unique grouping vector $\mathbf{a}$ and the group lengths $\mathbf{x}$. ■

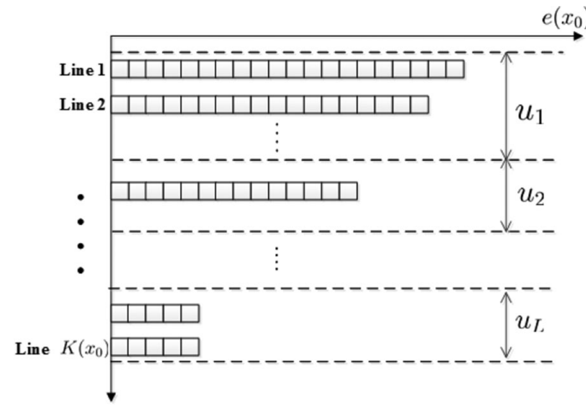



Fig. 7. Determine grouping strategy $(\mathbf{a})$ from group size $\mathbf{u}$ when $x_0$ is fixed.

In the remainder of the paper, we assume without loss of generality that $d_i \ge d_j$ for any $i < j$. For a fixed $x_0$, denote the number of remaining untransmitted data symbols as

$$\mathbf{e}(x_0) := (\mathbf{d} - x_0)_{++} \in \mathbb{Z}^{K(x_0)}_{++}, \tag{II.14}$$

where $K(x_0)$ is the number of positive entries in $\mathbf{d} - x_0$. The length of unoccupied portion in the frame is denoted by $\delta(x_0) := M_{ds} - x_0$. Since $x_0$ is fixed, all the lines with $d_i \le x_0$ (lines $K(x_0)+1$ to $K$) will only transmit data during NOI, the problem then becomes to optimally group the lines 1 to $K(x_0)$ and schedule the transmission of the remaining data symbols in a TDMA manner within DOI. For a fixed $x_0$, let us denote the optimal grouping strategy as $\mathbf{a}^*(x_0)$, the optimal group lengths as $\mathbf{x}^*(x_0)$, the optimal group sizes as $\mathbf{u}^*(x_0)$ and the minimum energy cost as $P^*(x_0)$. By Corollary 1, we only need to search for $\mathbf{u}^*(x_0)$, then $\mathbf{a}^*(x_0)$ and $\mathbf{x}^*(x_0)$ can be obtained from (13) and (14) respectively.

Notice that the objective function $P$ can be written as the sum energy consumption of the subgroups in DOI and the energy consumption in NOI:

$$P = P_0 + \sum_{\ell=1}^{L} P_\ell, \tag{II.15}$$

where $P_0$ is the energy consumed in NOI, $P_\ell$ is the energy consumed by subgroup $\ell$ in DOI. For a fixed $x_0$, the value of $P_0$ is fixed since

$$P_0 = P_0(x_0) = \sum_{i: d_i \le x_0} (x_0 - d_i) + \beta K^2 x_0. \tag{II.16}$$

The value of $P_\ell (\ell = 1, \ldots, L)$ is given by

$$P_\ell = P_\ell(x_\ell, u_\ell) = \sum_{i: a_i = \ell} (x_\ell - e_i) + \beta u^2_\ell x_\ell, \tag{II.17}$$

where $x_\ell$ is given by (II.13). Therefore, for a fixed $x_0$, problem (II.8) can be reformulated as

$$\begin{aligned} \text{minimize} \quad & g(\mathbf{u}) = P_0(x_0) + \sum_{\ell=1}^{L} P_\ell(x_\ell, u_\ell) \\ \text{subject to} \quad & x_\ell = \big(e(x_0)\big)_{U_{\ell-1}+1}, \quad \ell = 1, \ldots, L, \\ & L = \text{length}(\mathbf{u}), \\ & u_\ell \in \mathbb{Z}_{++}, \quad \ell = 1, \ldots, L, \end{aligned} \tag{II.18}$$

where length($\mathbf{u}$) represents the number of entries in vector $\mathbf{u}$. By Corollary 1, $\mathbf{u}^*(x_0)$ is an optimal solution of (II.18) and $g(\mathbf{u}^*(x_0)) = P^*(x_0)$.

Next we propose a dynamic programming (DP) algorithm [12] to solve (II.18). Dynamic programming breaks a multi-stage decision problem into simpler stages and keeps track of the evolving of states. We select the group size vector $\mathbf{u}$ to be the control variable.

*State space:* Let us first define the state space. We denote the state when the first $\ell$ subgroups have been formed with the sizes of $u_1, \ldots, u_\ell$ by

$$S_\ell = (U_\ell, X_\ell), \tag{II.19}$$

where $U_\ell := \sum_{k=1}^{\ell} u_k, X_\ell := \sum_{k=1}^{\ell} x_k$. Under state $S_\ell$, lines $1 + U_\ell$ to $K(x_0)$ are to be grouped and the occupied length of DOI is $X_\ell$. The initial state $S_0(x_0) = (0, 0)$ is fixed and the target state has the form of $S^*(x_0) = (K(x_0), X^*(x_0))$ where $X^*(x_0) \le \delta(x_0), X^*(x_0) \in \mathbb{Z}_+$. The state space can be expressed by

$$\begin{aligned} \mathcal{S}(x_0) := \Big\{ & S_\ell = (U_\ell, X_\ell) \mid U_\ell = \sum\nolimits_{k=1}^{\ell} u_k, X_\ell = \sum\nolimits_{k=1}^{\ell} x_k, \\ & 0 \le U_\ell \le K(x_0), 0 \le X_\ell \le \delta(x_0), x_k, u_k \in \mathbb{Z}_{++}, \\ & k = 1, \ldots, \ell, \ell \in \mathbb{Z}_{++} \Big\}. \end{aligned}$$

*Decision space:* For any $S_\ell = (U_\ell, X_\ell) \in \mathcal{S}(x_0)$, let us denote the decision space at state $S_\ell$ as $D(S_\ell)$. Suppose we are at state $S_\ell$, a decision of $u \in D(S_\ell)$ will transit us to a new state denoted by $T(S_\ell, u)$. Obviously, we should have $T(S_\ell, u) \in \mathcal{S}(x_0)$ for any $u \in D(S_\ell)$. In the following, we analyze the decision space $D(S_\ell)$ under three cases.

- If $U_\ell = K(x_0)$ and $X_\ell \le \delta(x_0)$, then $S_\ell$ is a target state and $D(S_\ell) = \emptyset$;
- If $U_\ell < K(x_0)$ and $X_\ell + e_{U_\ell+1} > \delta(x_0)$, then no more subgroup can be formed, i.e., $D(S_\ell) = \emptyset$; moreover, according to the property (10), the lines $U_\ell + 1$ to $K(x_0)$ should all join subgroup $\ell$;
- If $U_\ell < K(x_0)$ and $X_\ell + e_{U_\ell+1} \le \delta(x_0)$, then the size of the next subgroup $(\ell + 1)$ can be any positive integer without exceeding $K(x_0) - U_\ell$, i.e., $D(S_\ell) = \{1, \ldots, K(x_0) - U_\ell\}$.

The complete expression of $D(S_\ell)$ is shown in the following equation.

$$D(S_\ell) = \begin{cases} \emptyset, & \text{if } U_\ell = K(x_0) \text{ or } X_\ell + e_{U_\ell+1} > \delta(x_0), \\ \{1, \ldots, K(x_0) - U_\ell\}, & \text{otherwise.} \end{cases} \tag{II.20}$$

Correspondingly, the obtained state after moving from $S_\ell$ via a decision $u \in D(S_\ell)$, i.e., $S_{\ell+1} = T(S_\ell, u) = (U_{\ell+1}, X_{\ell+1})$, can be expressed by the following two equations.

$$U_{\ell+1} = U_{\ell+1}(S_\ell, u) = \begin{cases} K(x_0), & \text{if } D(S_\ell) = \emptyset, \\ U_\ell + u, & \text{otherwise,} \end{cases} \tag{II.21}$$

$$X_{\ell+1} = X_{\ell+1}(S_\ell) = \begin{cases} X_\ell, & \text{if } D(S_\ell) = \emptyset, \\ X_\ell + e_{(U_\ell+1)}, & \text{otherwise.} \end{cases} \tag{II.22}$$

*Stage cost:* Let $R(S_\ell, u)$ be the stage cost, which refers to the cost of moving from $S_\ell$ to $T(S_\ell, u)$ via the decision $u \in D(S_\ell)$. If $D(S_\ell) = \emptyset$ and $S_\ell$ is not a target state, the remaining unassigned lines will be grouped into subgroup $\ell$, and $R(S_\ell, u)$ is exactly the energy cost contributed by the new members in subgroup $\ell$. In particular, $u_\ell$ will be updated by $u'_\ell = u_\ell + (K(x_0) - U_\ell)$ and

$$\begin{aligned} R(S_\ell, u) = R(S_\ell) &= P_\ell(x_\ell, u'_\ell) - P_\ell(x_\ell, u_\ell) \\ &= \sum_{j=U_\ell+1}^{K(x_0)} (x_\ell - e_j) + \beta(u'^2_\ell - u^2_\ell) x_\ell. \end{aligned} \tag{II.23}$$

Otherwise, if $D(S_\ell) \ne \emptyset$, the stage cost equals the energy consumption of the newly formed subgroup with the size of $u$, which is given by $P_{\ell+1}$, i.e.,

$$R(S_\ell, u) = P_{\ell+1}(x_{\ell+1}, u) = \sum_{j=1+U_\ell}^{U_{\ell+1}} (x_{\ell+1} - e_j) + \beta u^2 x_{\ell+1}.$$

To summarize, the expression of $R(S_\ell, u)$ is

$$R(S_\ell, u) = \begin{cases} 0, & \text{if } U_\ell = K(x_0), \\ \sum_{j=1+U_\ell}^{U_{\ell+1}} (x_{\ell+1} - e_j) + \beta u^2 x_{\ell+1}, & \text{if } D(S_\ell) \ne \emptyset, \\ \sum_{j=U_\ell+1}^{K(x_0)} (x_\ell - e_j) + \beta(u'^2_\ell - u^2_\ell) x_\ell, & \text{otherwise.} \end{cases} \tag{II.24}$$

With the above discussions, problem (II.18) can be presented as a multi-stage decision problem: consider a sequence of decisions $u_1, \ldots, u_\ell, \ldots, u_L$ which generates a sequence of states in $\mathcal{S}(x_0)$:

$$S_1 = T(S_0(x_0), u_1), \ldots, S_{\ell+1} = T(S_\ell, u_{\ell+1}), \ldots, S_L = T(S_{L-1}, u_L) = S^*(x_0),$$

with the cost of $P(\mathbf{u}) = \sum_{\ell=1}^{L} R(S_{\ell-1}, u_\ell)$, find the optimal decision sequence such that the cost is minimized. To solve the problem by dynamic programming, we need to establish the Bellman equation which breaks the problem recursively into simple subproblems.

*Bellman equation:* Let $f^*(S_\ell)$ be the minimum cost of moving from $S_\ell \in \mathcal{S}(x_0)$ to $S^*(x_0)$, which is exactly the energy

TABLE II
THE ALGORITHM TO COMPUTE THE RECURSIVE FUNCTION $f^*(S_\ell)$

| | |
|---|---|
| **Step 1**. | If $U_\ell = K(x_0)$,<br>output $f^*(S_\ell) = 0$;<br>Else<br>compute the decision space $D(S_\ell)$ by (II.20), go to step 2; |
| **Step 2**. | If $D(S_\ell) = \emptyset$,<br>compute $R(S_\ell)$ according to (II.23);<br>output $f^*(S_\ell) = R(S_\ell)$;<br>Else<br>go to step 3; |
| **Step 3**. | $x_{\ell+1} = e_{U_{\ell+1}}$;<br>For each $i \in D(S_\ell)$<br>$u = i$;<br>compute $R(S_\ell, u)$, $S_{\ell+1} = T(S_\ell, u)$ and $f^*(S_{\ell+1})$;<br>$v_i = f^*(S_{\ell+1}) + R(S_\ell, u)$;<br>End<br>output $f^*(S_\ell) = \min v$. |

TABLE III
THE DP ALGORITHM FOR OPTIMAL DO PATTERN DESIGN

| **Input**: $\mathbf{d}\ (= \hat{\mathbf{d}})$ |
|---|
| **For** each feasible $x_0$<br>compute $\mathbf{e}(x_0)$ according to (II.14);<br>$x_1 = e_1$;<br>**For** $u_1 \in D(S_0(x_0))$<br>compute $R(S_0(x_0), u_1)$ according to (II.24);<br>compute $S_1 = T(S_0(x_0), u_1)$ and $f^*(S_1)$;<br>$f^*(u_1 \mid S_0(x_0)) = f^*(S_1) + R(S_0(x_0), u_1)$;<br>**End**<br>$P^*(x_0) = \min f^*(u_1 \mid S_0(x_0))$;<br>**End**<br>$P^* = \min P^*(x_0)$;<br>$x_0^* = \arg\min_{x_0} P^*(x_0)$. |

consumption associated with the optimal grouping of the lines $U_\ell + 1$ to $K(x_0)$ and scheduling their transmissions in the remaining $\delta(x_0) - X_\ell$ symbol positions. Let $f^*(u \mid S_\ell)$ be the minimum cost of moving from $S_\ell \in \mathcal{S}(x_0)$ to $S^*(x_0)$ with the first decision being $u$. Then by the definitions of $f^*(S_\ell)$ and $f^*(u \mid S_\ell)$, the following equation holds:

$$f^*(S_\ell) = \min_{u \in D(S_\ell)} f^*(u \mid S_\ell), \quad \forall\, \ell. \tag{II.25}$$

Notice that to achieve $f^*(u \mid S_\ell)$, the transition from $S_\ell$ to $S^*(x_0)$ is divided into two steps: first transit $S_\ell$ to $T(S_\ell, u)$ with the cost of $R(S_\ell, u)$, second transit $T(S_\ell, u)$ to $S^*(x_0)$ optimally with the cost of $f^*\big(T(S_\ell, u)\big)$. The corresponding mathematical relationship is given by

$$f^*(u \mid S_\ell) = R(S_\ell, u) + f^*\big(T(S_\ell, u)\big). \tag{II.26}$$

Combining (26) and (27) we can obtain the Bellman equation:

$$f^*(S_\ell) = \min_{u \in D(S_\ell)} \{R(S_\ell, u) + f^*\big(T(S_\ell, u)\big)\}, \quad \forall\, \ell, \tag{II.27}$$

which provides the recursive relationship with respect to $f^*(S_\ell)$.

As described above, the new state $T(S_\ell, u)$ is given by (II.21) and (II.22), the stage cost function $R(S_\ell, u)$ is given by (II.24), and the Bellman equation is defined in (II.27). By the Bellman equation, the computation of $f^*(S_\ell)$ can be implemented in a recursive manner, which is shown in Table II. Recall that

$$f^*(S_0(x_0)) = g(\mathbf{u}^*(x_0)) = P^*(x_0), \tag{II.28}$$

therefore $P^*(x_0)$ can be obtained from $f^*(S_0(x_0))$.

Finally, to solve the DO pattern design problem (II.8), we only need to execute the DP algorithm for each feasible value of $x_0$. The framework of our algorithm to solve II.8) is: for each feasible value of $x_0$, obtain the optimal grouping strategy by the DP algorithm, then obtain $P^*$ from $\min_{x_0} P^*(x_0)$. See the complete algorithm for DO pattern design in Table III. The global optimality is guaranteed since for each subproblem with a fixed value of $x_0$, the obtained solution $\mathbf{a}^*(x_0)$ is optimal, see Proposition 1.

*Proposition 1:* The DP algorithm in Table III gives an optimal solution of problem (II.8).

*Proof:* Since the optimal value of (II.8) $P^*$ satisfies:

$$P^* = \min_{\mathbf{a},\mathbf{x},\mathbf{u},x_0,L} P = \min_{x_0} \min_{\mathbf{a},\mathbf{x},\mathbf{u},L} P = \min_{x_0} P^*(x_0),$$

and $P^*(x_0)$ can be obtained by the DP algorithm in Table II, then the output of Table III gives the optimal value $P^*$.

*Complexity of the DP Algorithm:* By the definition of the state space $\mathcal{S}(x_0)$, there are at most $K(x_0)\delta(x_0)$ states per stage. The number of legitimate transitions from the initial state to a target state is at most $K(x_0)$, thus the number of states that the DP algorithm searches is at most $K(x_0)\delta(x_0) \times K(x_0)$, which can be upper bounded by $K^2 M_{ds}$. By the expressions of decision space in (21) and stage cost in (25), the size of $D(S_\ell)$ is at most $K(x_0)$ and the computation of $R(S_\ell, u)$ takes $O(K(x_0))$ operations. Then the total computation of the stage costs has a complexity of $O(K) \times O(K) \times O(K^2 M_{ds}) = O(K^4 M_{ds})$. According to the Bellman equation (28), the computation of $f^*(S_\ell)$ takes $O(K)$ operations per state once the $R(S_\ell, u)$ are computed. Therefore, the DP algorithm to compute $P^*(x_0)$ has the polynomial time complexity of $O(K^4 M_{ds}) + O(K^2 M_{ds}) \times O(K) = O(K^4 M_{ds})$. Since $1 \le x_0 \le M_{ds}$, searching over $x_0$ gives another factor of $M_{ds}$, therefore the complete algorithm for DO pattern design has the polynomial time complexity of $O(K^4 M_{ds}^2)$.

*Extension:* Actually, if it is allowed to modify the given demands in a certain range, we can jointly reshape the pre-allocated demands and design the optimal DO pattern at the same time. In particular, the formulation (II.8) and the DP algorithm assume $\mathbf{d} = \hat{\mathbf{d}}$ is fixed. For joint DO pattern design and demand adjustment, we can additionally let $\mathbf{d}$ be an optimization variable which varies around the given demand vector $\hat{\mathbf{d}}$. This joint design can further reduce energy consumption. See the details in Appendix B.

## III. ADJUSTMENT OF DO PATTERN

Implementing a new DO pattern is not cheap since a new set of precoders needs to be generated and configured, which is time consuming. In practice, it is necessary to fix the grouping strategy and adjust only the durations of the subgroups in DOI as well as the duration of NOI, while the goal is still the minimization of energy cost. The grouping strategy (precoders) can be kept unchanged unless a sufficient time has lapsed and the traffic condition has changed significantly to warrant a new grouping strategy.

In this section, we consider optimally adjusting the durations of NOI and the subgroups in DOI as well as the given demands while keeping the members of each subgroup in DOI unchanged. We formulate the problem as an integer linear program and provide a closed form optimal solution for this problem.

### A. Problem Formulation

For an existing grouping strategy, we denote $L$ as the number of subgroups in DOI, $G_\ell$ as the index set of the members in subgroup $\ell$:

$$G_\ell = \{i \mid a_i = \ell\}, \quad \ell = 1, \ldots, L,$$

and $G_0$ as the index set of the remaining lines that only transmit during the NOI:

$$G_0 = \{i \mid a_i = 0\} = \{1, \ldots, K\} \setminus \cup_{\ell=1}^{L} G_\ell.$$

Different from the DO pattern design problem where we take the number of data symbols to be transmitted as the given pre-allocated demands (i.e., $\mathbf{d} = \hat{\mathbf{d}}$), we will set $\mathbf{d}$ as a variable and optimize it in a range around $\hat{\mathbf{d}}$ to further reduce energy consumption. The reasons are two folds. First, the instantaneously pre-allocated demands $\hat{\mathbf{d}}$ may not be very accurate (they are estimated from traffic prediction, channel condition, and buffer status information, etc.). Second, the use of buffer makes it possible to transmit slightly more/fewer data than the estimated demands, which in turn can lead to additional energy saving when the demands in each group are made more uniform. This motivates the following constraint:

$$\max\{d_{\min}, \alpha_1 \hat{d}_i\} \le d_i \le \min\{\alpha_2 \hat{d}_i, M_{ds}\}, \forall\, i, \qquad \text{(III.1)}$$

where $d_{\min} \in \mathbb{Z}_+, \alpha_1, \alpha_2$ are predefined constants and $0 < \alpha_1 \le 1 \le \alpha_2$. Here the "min" and "max" operators ensure each $d_i$ satisfies $d_{\min} \le d_i \le M_{ds}$ (e.g., $d_{\min} = 1$). A special case when $\alpha_1 = \alpha_2 = 1$ will require the allocated transmission opportunities exactly follow the given demands. In practice, the values of $\alpha_1$ and $\alpha_2$ should be properly to ensure buffer stability. Letting $\alpha_1 < 1, \alpha_2 > 1$ in (III.1) allows the demands of different lines to be jointly adjusted, and can result in additional energy saving (see Section IV).

To minimize the energy cost $P$ in (5), we consider the following optimization problem:

$$\begin{aligned}
&\text{minimize} \quad P\\
&\text{subject to} \quad \text{(II.1), (III.1)},\\
&\qquad d_i \le x_0, \forall\, i \in G_0,\\
&\qquad d_i \le x_0 + x_\ell, \forall\, i \in G_\ell, \ell = 1, \ldots, L,\\
&\qquad d_i, x_\ell, x_0 \in \mathbb{Z}_+, \forall\, i, \ell = 1, \ldots, L. \qquad \text{(III.2)}
\end{aligned}$$

Notice that problem (III.2) is the same as DO pattern design problem (II.8) except that:

i) the grouping strategy is fixed and given by $\{G_\ell\}_{\ell=1}^{L}$ rather than the optimization variable $\mathbf{a}$;

ii) the constraints describing the relation of the number of data symbols and the group length are formulated as

$$\begin{cases} d_i \le x_0, \forall\, i \in G_0,\\ d_i \le x_0 + x_\ell, \forall\, i \in G_\ell,\ \ell = 1, \ldots, L. \end{cases}$$

iii) a box constraint on $\mathbf{d}$ is added:

$$\max\{d_{\min}, \alpha_1 \hat{d}_i\} \le d_i \le \min\{\alpha_2 \hat{d}_i, M_{ds}\}, i = 1 \ldots, K.$$

This constraint is used to reshape given demands in order to further reduce energy cost while staying close to the pre-allocated ones. If $\alpha_1 = \alpha_2 = 1$, we have $d_i = \hat{d}_i$ for all $i$. In this case, it is relatively simple to obtain an optimal solution of problem (III.2). Since the grouping strategy and demands are given, we only need to compute the minimal group length $x_\ell$ needed for each subgroup $\ell$ when $x_0$ is fixed. However, energy cost can not be further reduced with the fixed demands in this case.

### B. The Algorithm for DO Pattern Adjustment

The problem (III.2) is an integer linear program with respect to $\mathbf{d}, \mathbf{x}$ and $x_0$. Generally speaking, integer linear programming problems are NP-complete [13], thus difficult to solve them to global optimality. Surprisingly, we derive a closed form optimal solution of the integer linear program (III.2). We further extend the closed form solution to the case where the objective function $P$ is in a more general form.

To solve (III.2), a key observation is that when $\mathbf{x}, x_0$ are fixed, the optimal value of $d_i$ can be uniquely determined by:

$$d_i^* = \min\{x_0, \lfloor \alpha_2 \hat{d}_i \rfloor\},\ i \in G_0, \qquad \text{(III.3)}$$

$$d_i^* = \min\{x_\ell + x_0, \lfloor \alpha_2 \hat{d}_i \rfloor\},\ i \in G_\ell,\ \ell = 1, \ldots, L. \qquad \text{(III.4)}$$

The above two equations imply that the optimal solution $(\mathbf{d}^*, \mathbf{x}^*, x_0^*)$ can be derived from $\mathbf{x}^*$, $x_0^*$. Due to a hidden monotonicity in the problem, the optimal value of $x_0$ is actually the minimum feasible value of $x_0$ (see Proposition 2). To calculate the minimum feasible solution of $x_0$, we first define the notations:

$$A_\ell = \max\{d_{\min}, \max_{j \in G_\ell} \lceil \alpha_1 \hat{d}_j \rceil\}, \ell = 0, 1, \ldots, L,$$

then obtain tighter bounds for $\mathbf{x}, x_0$ from the constraints in problem (III.2):

$$\begin{aligned} &x_0 \ge d_i \ge \max\{d_{\min}, \lceil \alpha_1 \hat{d}_i \rceil\}, \forall\, i \in G_0,\\ &\qquad \Longrightarrow x_0 \ge A_0, \end{aligned} \qquad \text{(III.5)}$$

$$\begin{aligned} &x_\ell + x_0 \ge d_i \ge \max\{d_{\min}, \lceil \alpha_1 \hat{d}_i \rceil\}, \forall\, i \in G_\ell,\\ &\qquad \Longrightarrow x_\ell \ge \max\{A_\ell - x_0, 0\}, \forall\, \ell = 1, \ldots, L. \end{aligned} \qquad \text{(III.6)}$$

Notice that another constraint relating to $\mathbf{x}, x_0$ is the time limitation constraint: $x_0 + \sum_\ell x_\ell \le M_{ds}$. Substituting $x_\ell$ in this constraint by its lower bound given in (35), we have

$$x_0 + \sum_{\ell=1}^{L} \max\{A_\ell - x_0, 0\} \le M_{ds}. \qquad \text{(III.7)}$$

TABLE IV
A CLOSED FORM SOLUTION OF DO PATTERN ADJUSTMENT

| **Input**: $\hat{\mathbf{d}}, \alpha_1, \alpha_2, \{G_\ell\}$ |
|---|
| 0. $A_\ell = \max\{d_{\min}, \max_{j \in G_\ell} \lceil \alpha_1 \hat{d}_j \rceil\}, \ell = 0, \ldots, L$; |
| 1. $x_0 = A_0$; |
| 2. while $\phi(x_0) > M_{ds}$ |
| $x_0 = x_0 + 1$; |
| 3. $x_\ell = \max\{0, A_\ell - x_0\}, \ell = 1, \ldots, L$; |
| 4. $d_i = \min\{x_0, \lfloor \alpha_2 \hat{d}_i \rfloor\}$ for $i \in G_0$; |
| $d_i = \min\{x_\ell + x_0, \lfloor \alpha_2 \hat{d}_i \rfloor\}$ for $i \in G_\ell$, $\ell = 1, \ldots, L$. |

Let $\phi(x) := x + \sum_{\ell=1}^{L} \max\{A_\ell - x, 0\}$, then the minimum feasible value of $x_0$ just equals $\bar{x}_0 := \min\{x \mid \phi(x) \leq M_{ds}, x \geq A_0, x \in \mathbb{Z}\}$ (see the proof of Proposition 2).

The algorithm for solving problem (III.2) is given in Table IV. It is obvious that the algorithm is easily implemented. Moreover, notice that $\phi$ is a convex function, we can apply bisection method to find $\bar{x}_0$ to further reduce the computational complexity when $M_{ds}$ is large. For space reasons, we omit the details in the paper.

For a fixed value of $x_0$, let $P^*(x_0)$ denote the optimal value of (III.2). Proposition 2 establishes the optimality of our algorithm.

*Proposition 2:* $P^*(x_0)$ is nondecreasing with $x_0$. The algorithm in Table IV gives an optimal solution of problem (III.2).

Since $P^*(x_0)$ is nondecreasing with $x_0$, the minimum feasible value of $x_0$ is actually the optimal $x_0^*$. For a different objective function, Proposition 2 still holds as long as the nondecreasing property of $P^*(x_0)$ is maintained. We extend our conclusion in Proposition 3. The objective function in problem (III.2) is actually a special case of $P$ in (III.8) where the constants $\alpha_0, \alpha_\ell, \beta_\ell$ are determined by $K, \beta, u_\ell$. The proof of Propositions 2 and 3 are relegated to the Appendix A.

*Proposition 3:* If the objective function in problem (31) has the form of

$$P = C \left[ \sum_{i \in G_0} (x_0 - \alpha_0 d_i) + \sum_{\ell=1}^{L} \beta_\ell \sum_{i \in G_\ell} (x_0 + x_\ell - \alpha_\ell d_i) \right], \tag{III.8}$$

where $C$, $\alpha_0$, $\alpha_\ell$, $\beta_\ell$ $(\ell = 1, \ldots, L)$ are positive constants and $\alpha_0 \leq 1$, $\alpha_\ell \leq 1$ for all $\ell$. Then the algorithm in Table IV still gives an optimal solution of problem (31).

*Complexity of the Algorithm:* As Table IV shows, the algorithm for DO pattern adjustment is in closed form. The step 2 checks at most $M_{ds}$ values of $x_0$. The other steps in the algorithm just directly compute the value of the optimal solution. Thus the algorithm in Table IV has a linear complexity of $O(M_{ds})$. If replacing step 2 by the bisection algorithm, then the complexity can be reduced to $O(\log_2(M_{ds}))$.

## IV. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed DRA algorithms. We will first show the result of optimal DO pattern design, then simulate the schedule of data transmission in a relatively long period of time. To simulate the pre-allocated demands for each frame, we generate the incoming traffic rates by the FARIMA (fractional autoregressive integrated moving average) model and then generate $\hat{d}_i$ by the traffic rate and buffer state via a scheme provided by Huawei. The FARIMA process is able to account for long-range dependence which is an important property of internet traffic [14].

### A. Simulation of DO Pattern Design

We will compare our algorithm with three simple algorithms: (1) transmission only in NOI, which implies all the lines are grouped in a common group; (2) transmission in NOI and DOI with only one subgroup; (3) divide the lines into equal-size subgroups, fix the grouping strategy and schedule the transmission by solving problem (31) with $\alpha_1 = \alpha_2 = 1$. For simplicity, we call algorithm (1) as "No-DO" algorithm, (2) as "1-DOgroup" algorithm and (3) as "Equal-size-group" algorithm. The No-DO algorithm is very easy to implement. To find the optimal solution of the 1-DOgroup algorithm, we will also first calculate the objective value $\bar{P}(x_0)$ for each feasible value of $x_0$, which is easy since all the lines with untransmitted data symbols will join the single DOI subgroup after the duration of NOI $(x_0)$ is determined; then we obtained the solution of 1-DOgroup algorithm from $\min_{x_0} \bar{P}(x_0)$ by enumeration.

We consider the DO pattern design for $K = 16$ lines in 100 data frames, where $M_{ds} = 32$ and the duration of each frame is set to 1 millisecond. For Equal-size-group algorithm, we divide the lines into 4 groups, each consisting of 4 lines with the closest mean demands. Figure 8 shows the statistic of energy saving when $\beta = 0.001$, $0.01$. The energy saving is defined as $\frac{\bar{P} - P^*}{\bar{P}}$, where $P^*$ is the global optimal value given by the DP algorithm and $\bar{P}$ is objective value returned by the other three algorithms. From the figure we can see that compared with the three simple algorithms, the energy cost returned by the proposed DP algorithm is much smaller. Specifically, compared with No-DO algorithm, the mean energy saving reaches 70% when $\beta = 0.001$ and 62% when $\beta = 0.01$. Among the three algorithms, 1-DOgroup performs the best, which to some extent shows the advantage of DO.

Further, we perform DO pattern design for 500,000 sequential data frames with different values of $K$ (number of users) and $\beta$. Table V shows the mean energy saving (ES), maximum/minimum ES per frame, the standard deviation of ES of the solutions obtained by DP algorithm and 1-DOgroup algorithm. We can see from the table that mean energy saving decreases with $\beta$ when $K$ is fixed. The reason is that the ES in terms of the inserted idle symbols is usually larger than that in terms of the precoder-data multiplications.

Table VI shows the mean data symbol percentage (DSp) per frame of the solutions obtained by the four algorithms. Here DSp refers to the ratio of the number of data symbols and the total number of transmitted symbols, i.e., $\frac{\sum_{i=1}^{K} d_i}{K x_0 + \sum_{\ell=1}^{L} u_\ell x_\ell}$. From Table VI we can see that the mean DSp of the DP algorithm is over 90% in all cases. These results demonstrate the efficiency of the DP algorithm. In the next subsection, we simulate the DRA algorithms in a more practical scenario.

### B. Simulation of the Algorithms for DRA

We consider the data transmission in 100 sequential superframes where each superframe consists of 8 frames. We assume that DO pattern design can only be implemented in the

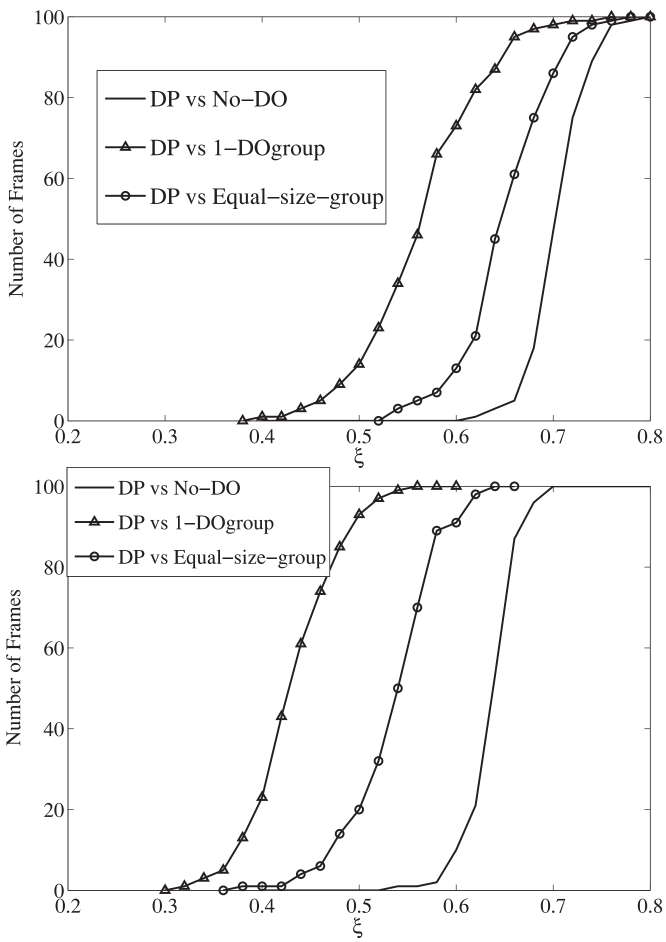



Fig. 8. The number of frames where the energy saving is less than or equal to $\xi \in [0.2, 0.8]$. Top: $\beta = 0.001$; Bottom: $\beta = 0.01$.

TABLE V
STATISTICS OF ENERGY SAVING: DP VS 1-DOGROUP

| $(K,\ \beta)$ | Mean_ES | Max_ES | Min_ES | SD_ES |
|---|---|---|---|---|
| (16, 0.001) | 52.32% | 73.78% | 37.06% | 0.0604 |
| (16, 0.01) | 41.03% | 51.92% | 28.50% | 0.0451 |
| (40, 0.001) | 45.83% | 53.76% | 37.20% | 0.0370 |
| (40, 0.01) | 30.07% | 38.30% | 24.58% | 0.0275 |
| (100, 0.001) | 34.46% | 44.15% | 29.17% | 0.0184 |
| (100, 0.01) | 23.29% | 30.59% | 16.84% | 0.0137 |

TABLE VI
STATISTICS OF MEAN DATA SYMBOL PERCENTAGE IN 500,000 FRAMES

| $(K,\ \beta)$ | DP | No-DO | 1-DOgroup | Equal-size-group |
|---|---|---|---|---|
| (16, 0.001) | 93.43% | 61.22% | 79.84% | 71.56% |
| (16, 0.01) | 92.20% | 60.57% | 77.61% | 70.63% |
| (40, 0.001) | 92.81% | 61.24% | 79.18% | 71.76% |
| (40, 0.01) | 92.31% | 60.25% | 78.46% | 69.20% |

superframe scale and the frames in a common superframe have the same set of precoders. A new grouping strategy will be designed in the first frame of the next superframe when the data symbol percentage (DSp) is below a threshold $\theta$. We set $K = 16, M_{ds} = 32, \beta = 0.001$, and the threshold $\theta = 90\%$.

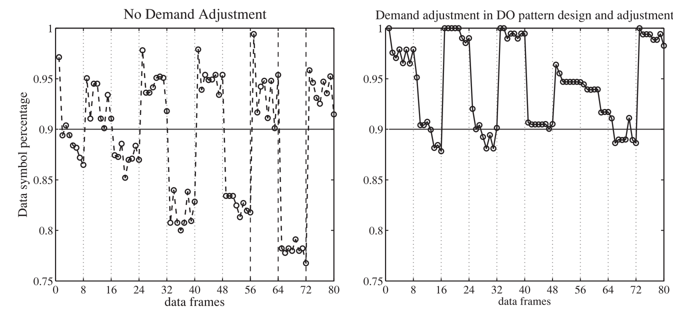



Fig. 9. $[\beta = 0.001]$ Data symbol percentage in the first 10 superframes. Left: No demand adjustment is involved; Right: Demand adjustment is involved in both DO pattern design and adjustment with $\alpha_1 = 0.9, \alpha_2 = 1.3$.

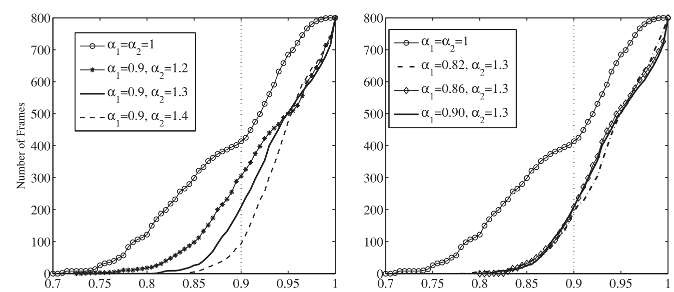



Fig. 10. $[\beta = 0.001]$ The distribution of data symbol percentage in 100 superframes with different choices of $\alpha_1, \alpha_2$.

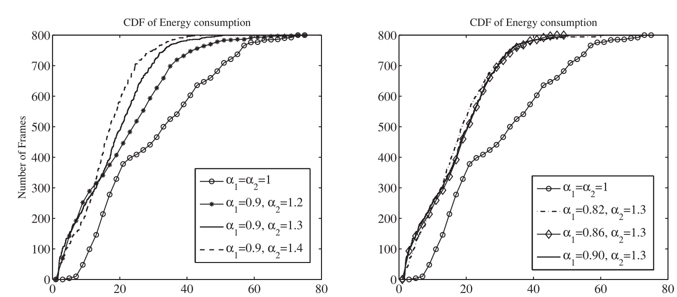



Fig. 11. $[\beta = 0.001]$ The distribution of energy consumption in 100 superframes with different choices of $\alpha_1, \alpha_2$.

Notice that problem (II.8) is a special case of problem (B.2) where there is no demand adjustment, i.e., $\alpha_1 = \alpha_2 = 1$. We will compare the obtained solutions with and without demand adjustment involved in both DO pattern design and adjustment. Figure 9 shows the DSp as it varies with time in the first 10 superframes. We can see that DSp increases due to the new grouping strategy, like the 4th superframe in the left figure. By comparing the two figures we can see that if we allow demand adjustment ($\alpha_1 < 1, \alpha_2 > 1$) instead of strictly following the pre-allocated demands ($\alpha_1 = \alpha_2 = 1$), fewer idle symbols are inserted and DSp is improved. Figure 10 and Figure 11 show the cumulative distribution of DSp and energy consumption in the 100 superframes respectively. We can see that DSp increases and energy consumption decreases after demand adjustment is involved. This is because that the objective function value of DO pattern design/adjustment problem can be further reduced when

TABLE VII
STATISTICS OF DATA SYMBOL PERCENTAGE IN 50,000 SUPERFRAMES

| $(K,\ \beta)$ | Mean_DSp | Max_DSp | Min_DSp | SD_DSp |
|---|---|---|---|---|
| (16, 0.001) | 92.42% | 100% | 79.32% | 0.0439 |
| (16, 0.01) | 90.85% | 97.19% | 80.41% | 0.0381 |
| (40, 0.001) | 92.05% | 99.06% | 81.38% | 0.0304 |
| (40, 0.01) | 90.68% | 97.31% | 82.42% | 0.0262 |

the corresponding feasible region is enlarged by involving constraint (30). Moreover, it is interesting to find that DSp increases and energy consumption decreases with $\alpha_2$, meanwhile they are insensitive to the choice of $\alpha_1$. These results demonstrate the effectiveness of our proposed DRA algorithms and the benefits of involving demand adjustment.

We further perform the DRA algorithms (with demand adjustment only in DO pattern adjustment) in 50,000 superframes with different values of $K$ and $\beta$. Table VII shows that the mean DSp is over 90% in all cases, which implies little energy is wasted on idle symbols.

These simulations are implemented both in MATLAB (R2012a) and Microsoft Visual C++ (6.0) with 1.80 GHz CPUs. To run the DRA algorithms for a total of $K = 16$ lines in $T = 100$ superframes, it uses about 2 seconds with MATLAB and 2 milliseconds with C.

## V. CONCLUSION

To reduce the energy consumption in vector processing for DSL ($P$ in (II.4)), we consider in this paper two key DRA problems: (1) the global optimal grouping, and (2) the dynamic adjustment of the durations of NOI and subgroups in DOI, and propose efficient real-time algorithms to solve them. Surprisingly, our work shows that the global optimal grouping strategy can be obtained in polynomial time using dynamic programming, and that the optimal adjustment of the DO pattern can be found in closed form. The numerical results have demonstrated the effectiveness of our algorithms. In future, it will be interesting to see if the techniques and formulations developed in this paper can be applied to other communication systems.

## APPENDIX A
## PROOF OF PROPOSITIONS 2 AND 3

We first rewrite the objective function $P$.

*Case I:* If $K > \sum_\ell u_\ell$, i.e., $G_0 \neq \emptyset$, then

$$\begin{aligned} P &= \left(Kx_0 + \sum_{\ell=1}^{L} u_\ell x_\ell - \sum_{i=1}^{K} d_i\right) + \beta\left(K^2 x_0 + \sum_{\ell=1}^{L} u_\ell^2 x_\ell\right) \\ &= \sigma \sum_{i\in G_0}\left(x_0 - \frac{1}{\sigma} d_i\right) \\ &\quad + \sum_{\ell=1}^{L}(1+\beta u_\ell)\sum_{i\in G_\ell}\left(x_0 + x_\ell - \frac{1}{1+\beta u_\ell} d_i\right) \\ &= \frac{1}{\alpha_0}\sum_{i\in G_0}(x_0 - \alpha_0 d_i) + \sum_{\ell=1}^{L}\frac{1}{\alpha_\ell}\sum_{i\in G_\ell}(x_0 + x_\ell - \alpha_\ell d_i), \end{aligned}$$

where $\beta = \frac{4p_c}{p_s}$, $\sigma = 1 + \beta\frac{K^2 - \sum_{\ell=1}^{L} u_\ell^2}{K - \sum_{\ell=1}^{L} u_\ell}$, $\alpha_0 = \sigma^{-1}$, $\alpha_\ell = (1 + \beta u_\ell)^{-1}$, for all $\ell = 1, \ldots, L$.

*Case II:* Otherwise, i.e., $G_0 = \emptyset$, then the objective function can be rewritten as

$$P = \beta\left(K^2 - \sum_\ell u_\ell^2\right)x_0 + \sum_{\ell=1}^{L}\frac{1}{\alpha_\ell}\sum_{i\in G_\ell}(x_0 + x_\ell - \alpha_\ell d_i).$$

Denote the solution obtained from the proposed algorithm in Table IV as $(\bar{\mathbf{d}}, \bar{\mathbf{x}}, \bar{x}_0)$, where $\bar{\mathbf{x}} = (\bar{x}_1, \ldots, \bar{x}_L)^T$, $\bar{\mathbf{d}} = (\bar{d}_1, \ldots, \bar{d}_K)^T$, $\bar{P}$ as the corresponding objective value. We will prove $\bar{P} = P^*$, thus $(\bar{\mathbf{d}}, \bar{\mathbf{x}}, \bar{x}_0)$ is an optimal solution of (III.2). In the following, we will consider Case I and introduce two lemmas. For Case II, Lemma 3 and Lemma 4 also hold and the proof is very similar.

*Lemma 2:* $\bar{x}_0$ is the minimum feasible value of $x_0$, i.e., if $x_0 = y_0$ is a feasible value for (III.2), then $y_0 \geq \bar{x}_0$.

*Proof:* Since $x_0 = y_0$ is feasible for (III.2), there exists $\mathbf{y} = (y_1, \ldots, y_L)^T$ which is a feasible value of $\mathbf{x}$, thus

$$y_0 + \sum_{\ell=1}^{L} y_\ell \leq M_{ds}, \tag{A.1}$$

$$\text{(III.6)} \implies y_\ell \geq \max\{0, A_\ell - y_0\}, \ell = 1, \ldots, L. \tag{A.2}$$

Combining (A.1) and (A.2) yields

$$y_0 + \sum_{\ell=1}^{L}\max\{0, A_\ell - y_0\} \leq M_{ds}. \tag{A.3}$$

The step 2 of the algorithm indicates that $\bar{x}_0$ is the minimum value of $x_0$ satisfying (A.3). Therefore, we have $y_0 \geq \bar{x}_0$. ■

For any fixed $y \geq y_0$ and a constant $\alpha \in (0, 1]$, let $f^*(y \mid \{I_i\}, y_0, \alpha)$ denote the optimal value of (P2) with input parameters $\{I_i\}, y_0$ and $\alpha$.

$$\begin{aligned} \min_{\mathbf{d}} \quad & \sum_{i=1}^{m}(y - \alpha d_i) \\ \text{s.t.} \quad & y \geq d_i,\ i = 1, \ldots, m, \\ & d_i \in I_i = [a_i, b_i],\ i = 1, \ldots, m. \end{aligned} \tag{P2}$$

For a fixed $y_0$ and a constant $\alpha \in (0, 1]$, let $f^*(\{I_i\}, y_0, \alpha)$ denote the optimal value of (P3) with input parameters $\{I_i\}, y_0$ and $\alpha$.

$$\begin{aligned} \min_{\mathbf{d}, y} \quad & \sum_{i=1}^{m}(y - \alpha d_i) \\ \text{s.t.} \quad & y \geq y_0, \\ & y \geq d_i,\ i = 1, \ldots, m, \\ & d_i \in I_i = [a_i, b_i],\ i = 1, \ldots, m. \end{aligned} \tag{P3}$$

*Lemma 3:* The function $f^*(y \mid \{I_i\}, y_0, \alpha)$ is nondecreasing with $y$, and the optimal solution of (P2) is: $d_i^*(y) = \min\{y, b_i\}$, $i = 1, \ldots, m$. An optimal solution of (P3) is: $\bar{y} := \max\{y_0, \max_i a_i\}$, $\bar{d}_i := \min\{\bar{y}, b_i\}$, $i = 1, \ldots, m$. Moreover, if it requires $d_i, y \in \mathbb{Z}$ in (P3), then $(\bar{y}, \bar{\mathbf{d}})$

is still an optimal solution of (P3) after modifying the input parameters by $y_0 = \lceil y_0 \rceil, a_i = \lceil a_i \rceil, b_i = \lfloor b_i \rfloor$, $i = 1, \ldots, m$.

*Proof:* The proof consists of three steps.

**(i)** Notice that (P2) is equivalent to

$$\max\ d_i \quad \text{s.t.} \quad y \geq d_i,\ d_i \in I_i = [a_i, b_i]$$

for $i = 1, \ldots, m$. Thus $d_i^* = \min\{y, b_i\}$.

Suppose (P2) has nonempty feasible sets with two inputs $y = y_1, y_2$ and $y_2 \geq y_1$. Then

$$y_2 - \alpha \min\{y_2, b_i\} \geq y_1 - \alpha \min\{y_1, b_i\}, \forall\, i, \tag{A.6}$$

which implies that

$$f^*(y_2 \mid \{I_i\}, y_0, \alpha) \geq f^*(y_1 \mid \{I_i\}, y_0, \alpha). \tag{A.7}$$

This proves that $f^*(y \mid \{I_i\}, y_0, \alpha)$ is nondecreasing with $y$.

**(ii)** For (P3), if $y$ is feasible, then

$$y \geq d_i \geq a_i, \forall\, i \quad \Longrightarrow \quad y \geq \max_i a_i,$$

thus $y \geq \max\{y_0, \max_i a_i\} = \bar{y}$. Thus $\bar{y}$ is the minimum feasible value of $y$.

Since $f^*(\{I_i\}, y_0, \alpha) = \min_{y \geq \bar{y}} f^*(y \mid \{I_i\}, y_0, \alpha)$ and from **(i)** we know that $f^*(y \mid \{I_i\}, y_0, \alpha)$ is nondecreasing with $y$, so $f^*(\{I_i\}, y_0, \alpha) = f^*(\bar{y} \mid \{I_i\}, y_0, \alpha)$ .

**(iii)** Since $d_i, y \in \mathbb{Z}$, the feasible set of (P3) is unchanged after modifying the input parameters by taking $y_0 = \lceil y_0 \rceil, a_i = \lceil a_i \rceil, b_i = \lfloor b_i \rfloor$. Similar to the analysis in **(i)-(ii)**, the conclusions still hold. ∎

*Proof of Proposition 2:* For any $i \in \{1, \ldots, K\}$, let us define an interval

$$I_i = \big[\max\{d_{\min}, \lceil \alpha_1 \hat{d}_i \rceil\}, \min\{\lfloor \alpha_2 \hat{d}_i \rfloor, M_{ds}\}\big].$$

Let $P^*(y_0)$ denote the optimal value of (31) when $x_0 = y_0$, $X_0$ denote the feasible region of $x_0$. Obviously,

$$P^* = \min_{y_0 \,\in X_0} P^*(y_0). \tag{A.8}$$

Since $(\bar{\mathbf{d}}, \bar{\mathbf{x}}, \bar{x}_0)$ is obtained from the algorithm in Table IV, it is easy to verify by Lemma 3 that

$$\sum_{i \,\in G_0} (\bar{x}_0 - \alpha_0 \bar{d}_i) = f^*(\{I_i\}_{i \in G_0}, \bar{x}_0, \alpha_0), \tag{A.9}$$

$$\sum_{i \,\in G_\ell} (\bar{x}_0 + \bar{x}_\ell - \alpha_\ell \bar{d}_i) = f^*(\{I_i\}_{i \in G_\ell}, \bar{x}_0, \alpha_\ell), \, \forall\, \ell. \tag{A.10}$$

For any $y_0 \in X_0$, it can be deduced from Lemma 2 that $y_0 \geq \bar{x}_0$, and by Lemma 3 we have

$$f^*(y_0 \mid \{I_i\}_{i \in G_0}, \bar{x}_0, \alpha_0) \geq f^*(\{I_i\}_{i \in G_0}, \bar{x}_0, \alpha_0), \tag{A.11}$$

$$f^*(y_0 \mid \{I_i\}_{i \in G_\ell}, \bar{x}_0, \alpha_\ell) \geq f^*(\{I_i\}_{i \in G_\ell}, \bar{x}_0, \alpha_\ell), \, \forall\, \ell. \tag{A.12}$$

By the definitions of $f^*(y_0 \mid \{I_i\}_{i \in G_\ell}, \bar{x}_0, \alpha_\ell)$ and $P^*(y_0)$,

$$\begin{aligned} P^*(y_0) &\geq \frac{1}{\alpha_0} f^*(y_0 \mid \{I_i\}_{i \in G_0}, \bar{x}_0, \alpha_0) \\ &\quad + \sum_{\ell=1}^{L} \frac{1}{\alpha_\ell} f^*(y_0 \mid \{I_i\}_{i \in G_\ell}, \bar{x}_0, \alpha_\ell) \\ &\geq \frac{1}{\alpha_0} f^*(\{I_i\}_{i \,\in G_0}, \bar{x}_0, \alpha_0) + \sum_{\ell=1}^{L} \frac{1}{\alpha_\ell} f^*(\{I_i\}_{i \in G_\ell}, \bar{x}_0, \alpha_\ell) \\ &= \frac{1}{\alpha_0} \sum_{i \in G_0} (\bar{x}_0 - \alpha_0 \bar{d}_i) + \sum_{\ell=1}^{L} \frac{1}{\alpha_\ell} \sum_{i \in G_\ell} (\bar{x}_0 + \bar{x}_\ell - \alpha_\ell \bar{d}_i) \\ &= \bar{P}. \end{aligned} \tag{A.13}$$

The first inequality holds since when $x_0 = y_0$ the feasible set of (III.2) is contained in that of (P2) which corresponds to $f^*(y_0 \mid \{I_i\}_{i \in G_\ell}, \bar{x}_0, \alpha_\ell)$. Since (A.13) holds for any $y_0 \in X_0$, combining with (45) we can obtain $P^* \geq \bar{P}$. Moreover, $P^* \leq \bar{P}$. Therefore $\bar{P} = P^*$, and $(\bar{\mathbf{d}}, \bar{\mathbf{x}}, \bar{x}_0)$ is an optimal solution of (III.2). ∎

*Proof of Proposition 3:* Similar to the proof of Proposition 2, we can prove that when $\alpha_\ell \in (0, 1]$, for all $\ell = 0, \ldots, L$, $P^*(x_0)$ is nondecreasing with respect to $x_0$ if $P$ is in the form (III.8). We omit the details here for the space reason.

## Appendix B
## Joint DO Pattern Design and Adjustment of Pre-allocated Demands

To further reduce the energy consumption of a new DO pattern, we consider the option of jointly designing DO pattern and adjusting the given demands in a certain range. Instead of letting $\mathbf{d} = \hat{\mathbf{d}}$, we include the constraint (III.1):

$$\max\{d_{\min}, \alpha_1 \hat{d}_i\} \leq d_i \leq \min\{\alpha_2 \hat{d}_i, M_{ds}\}, \forall\, i, \tag{B.1}$$

where $d_{min} \in \mathbb{Z}, \alpha_1, \alpha_2$ are predefined positive constants. Then the resulting DRA problem jointly optimizes $\mathbf{d}$ and the DO pattern, as shown in (B.2) below. Here the objective function $P$ is defined in (II.4).

$$\begin{aligned} &\text{minimize} \quad P \\ &\text{subject to} \quad (\text{II}.1), (\text{II}.5), (\text{II}.6), (\text{II}.7), (\text{B}.1), \\ &\qquad\qquad\quad x_0, x_\ell, u_\ell, L, a_i, d_i \in \mathbb{Z}_+, \forall\, i,\, \ell. \end{aligned} \tag{B.2}$$

We claim that the DP algorithm still works after a few modifications. We first present the variant of Lemma 10 in Lemma 4, which provides an insight into the relation between the pre-allocated demands $\hat{\mathbf{d}}$ and the group length vector $\mathbf{x}$.

*Lemma 4:* There exists an optimal solution of (52) with the property that for any $i, j \in Q := \{i \mid x_0 < \lceil \alpha_1 \hat{d}_i \rceil\}$,

$$\hat{d}_i > \hat{d}_j \implies x_{a_i} \geq x_{a_j}. \tag{B.3}$$

In other words, in this optimal solution, a line with larger expected demand is placed in a subgroup with longer length.

*Proof:* Suppose $(\mathbf{d}, x_0, \mathbf{a}, \mathbf{x}, \mathbf{u}, L)$ is an optimal solution of (52) such that there exist two lines $i, j \in Q$ with $\hat{d}_i > \hat{d}_j$ and $x_{a_i} < x_{a_j}$.

Due to constraint (51) and the optimality of the solution, it is easy to verify that: for any $i$, if $\lfloor \alpha_2 \hat{d}_i \rfloor \le x_0$, then the optimal value of $d_i$ is set to $d_i = \lfloor \alpha_2 \hat{d}_i \rfloor$; if $\max\{d_{\min}, \lceil \alpha_1 \hat{d}_i \rceil\} \le x_0 < \lfloor \alpha_2 \hat{d}_i \rfloor$, then the optimal value of $d_i$ is given by $d_i = x_0$. Thus, the set of lines that transmit in DOI is $Q = \{i \mid x_0 < \lceil \alpha_1 \hat{d}_i \rceil\}$. We will show that another optimal solution satisfying (B.3) can be obtained by modifying $\mathbf{d}$ and $\mathbf{a}$.

- If $d_i > d_j$, then let $\mathbf{d}' = \mathbf{d}$, same to the proof of Lemma 10 we can show that there exists $\mathbf{a}'$ such that (B.3) holds;
- If $d_i = d_j$, we define a new assignment $\mathbf{a}'$ by exchanging the group indices of lines $i$ and $j$, i.e., $a'_i = a_j, a'_j = a_i$, and let $\mathbf{d}' = \mathbf{d}$ and $\mathbf{x}' = \mathbf{x}$, then the energy cost is unchanged and $x'_{a'_i} \ge x'_{a'_j}$ is satisfied.
- If $d_i < d_j$, we define $\mathbf{d}', \mathbf{a}'$ such that $d'_i = d_j, d'_j = d_i, d'_k = d_k, k \ne i, j$ (such definition is reasonable since $\hat{d}_i > \hat{d}_j$), and $a'_i = a_j, a'_j = a_i, a'_k = a_k, k \ne i, j$, then the energy cost is unchanged and $x'_{a'_i} \ge x'_{a'_j}$ is satisfied.

In this way, we can obtain a feasible $\mathbf{a}'$ and $\mathbf{d}'$ such that the conclusion holds. ■

Lemma 4 implies that for a fixed $x_0$, we can still solve (B.2) by applying the DP algorithm, i.e., searching for the optimal decision $\mathbf{u}^*(x_0)$. Since the Bellman equation is the same as (II.27), the DP algorithm for (B.2) is similar to the algorithm in Table II, except for the steps such as those involving the computation of $\mathbf{d}$. Moreover, notice that $\mathbf{e}(x_0)$ is not a constant vector, the computation of the decision space $D(S_\ell)$ as well as the stage cost $R(S_\ell, u)$ are also different.

Without loss of generality, we assume that the entries of $\hat{\mathbf{d}}$ are ordered: $\hat{d}_i \ge \hat{d}_j, \forall\, i < j$. Define $K(x_0)$ as the number of elements in $Q(x_0) = \{i \mid x_0 < \lceil \alpha_1 \hat{d}_i \rceil\}$. Then lines 1 to $K(x_0)$ are to be grouped into small subgroups. The following procedure gives a simple description of the computation of $D(S_\ell)$, $R(S_\ell, u)$ and $\mathbf{d}$.

- If $U_\ell = K(x_0)$ and $X_\ell \le \delta(x_0)$, then $S_\ell$ is a target state, $D(S_\ell) = \emptyset, R(S_\ell) = 0$;
- If $U_\ell < K(x_0)$ and $X_\ell + \lceil \alpha_1 \hat{d}_{U_\ell+1} \rceil > M_{ds}$, then no more subgroup can be formed, $D(S_\ell) = \emptyset$ and the lines $U_\ell + 1$ to $K(x_0)$ should join subgroup $\ell$. To reduce the energy cost contributed by lines $U_\ell + 1$ to $K(x_0)$, the numbers of data symbols of these lines should be as large as possible, thus

$$d_j = \min\{x_0 + x_\ell, \lfloor \alpha_2 \hat{d}_j \rfloor\}, \quad j = U_\ell + 1, \ldots, K(x_0).$$

The stage cost of transiting from $S_\ell = (U_\ell, X_\ell)$ to $S^*(x_0) = (K(x_0), X_\ell)$ is

$$\begin{aligned} R(S_\ell) =&\; P_\ell(u'_\ell, x_\ell) - P_\ell(u_\ell, x_\ell) \\ =&\; \sum_{j=U_\ell+1}^{K(x_0)} (x_0 + x_\ell - d_j) + \beta x_\ell (u'^2_\ell - u^2_\ell), \end{aligned}$$

where $u'_\ell = u_\ell + K(x_0) - U_\ell$.

- If $U_\ell < K(x_0)$ and $X_\ell + \lceil \alpha_1 \hat{d}_{U_\ell+1} \rceil \le M_{ds}$, then $D(S_\ell) = \{1, \ldots, K(x_0) - U_\ell\}$. To make the stage cost as small as possible, the demand of line $U_\ell + 1$ should be as small as possible. Then the length of the new subgroup is $x_{\ell+1} = \lceil \alpha_1 \hat{d}_{U_\ell+1} \rceil - x_0$. Under a decision $u \in D(S_\ell)$, $U_{\ell+1} = U_\ell + u$, and the demands of the members in subgroup $\ell + 1$ are

$$\begin{aligned} d_{U_\ell+1} =&\; \lceil \alpha_1 \hat{d}_{U_\ell+1} \rceil, \\ d_j =&\; \min\{x_0 + x_{\ell+1}, \lfloor \alpha_2 \hat{d}_j \rfloor\}, j = U_\ell + 2, \ldots, U_{\ell+1}. \end{aligned}$$

The related stage cost is given by the energy cost of subgroup $\ell + 1$:

$$\begin{aligned} R(S_\ell, u) =&\; P_{\ell+1}(u, x_{\ell+1}) \\ =&\; \sum_{j=U_\ell+1}^{U_{\ell+1}} (x_0 + x_{\ell+1} - d_j) + \beta u^2 x_{\ell+1}. \end{aligned}$$

By changing in Table II the computation of decision space $D(S_\ell)$, stage cost $R(S_\ell, u)$ and demands $\mathbf{d}$ listed above, we obtain a DP algorithm to solve (B.2) for a fixed $x_0$. Same as before, this modified DP algorithm generates a global optimal solution of (B.2) by first computing the values of $P^*(x_0)$ for each feasible value of $x_0$ and then picking the minimum.

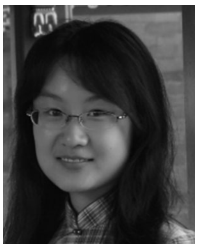
## ACKNOWLEDGMENT

This work is conducted at the Shenzhen Research Institute of Big Data, Shenzhen, China.

## REFERENCES

[1] ITU, *Fast Access to Subscriber Terminals (FAST)—Physical Layer Specification*, ITU-T Recommendation G.9701, Dec. 2014.
[2] V. Oksman *et al.*, “The ITU-T’s new G.fast standard brings DSL into the gigabit era,” *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 118–126, Mar. 2016.
[3] M. Timmers, K. Hooghe, M. Guenach, C. Stony, and J. Maes, “System design of reverse-powered G.fast,” in *Proc. IEEE Int. Conf. Commun.*, Jun. 2012, pp. 6869–6873.
[4] M. Timmers, M. Guenach, C. Nuzman, and J. Maes, “G.fast: Evolving the copper access network,” *IEEE Commun. Mag.*, vol. 51, no. 8, pp. 74–79, Aug. 2013.
[5] G. Ginis and J. M. Cioffi, “A multi-user precoding scheme achieving crosstalk cancellation with application to DSL systems,” in *Proc. IEEE Conf. Rec. 34th Asilomar Conf. Signals, Syst. Comput.*, Oct. 2000, vol. 2, pp. 1627–1631.
[6] G. Ginis and J. M. Cioffi, “Vectored transmission for digital subscriber line systems,” *IEEE J. Sel. Areas Commun.*, vol. 20, no. 5, pp. 1085–1104, Jun. 2002.
[7] V. Oksman *et al.*, “The ITU-T’s new G.vector standard proliferates 100 Mb/s DSL,” *IEEE Commun. Mag.*, vol. 48, no. 10, pp. 140–148, Oct. 2010.
[8] R. Cendrillon, W. Yu, M. Moonen, J. Verlinden, and T. Bostoen, “Optimal multiuser spectrum balancing for digital subscriber lines,” *IEEE Trans. Commun.*, vol. 54, no. 5, pp. 922–933, May 2006.
[9] P. Tsiaflakis, M. Diehl, and M. Moonen, “Distributed spectrum management algorithms for multiuser DSL networks,” *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4825–4843, Oct. 2008.
[10] M. Guenach, C. Nuzman, P. Tsiaflakis, and J. Maes, “Power optimization in vectored and non-vectored G.fast transmission,” in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 2229–2233.
[11] J. Maes and C. Nuzman, “Energy efficient discontinuous operation in vectored G.fast,” in *Proc. IEEE Int. Conf. Commun.*, Jun. 2014, pp. 3854–3858.
[12] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. I. Bermont, MA, USA: Athena Scientific, 1995.

[13] A. Schrijver, *Theory of Linear and Integer Programming*. Hoboken, NJ, USA: Wiley, 1998.
[14] C. W. J. Granger and R. Joyeux, “An introduction to long-memory time series models and fractional differencing,” *J. Time Ser. Anal.*, vol. 1, pp. 15–29, 1980.

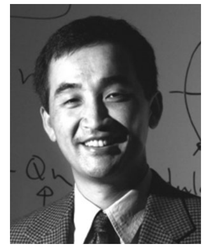


**Nan Zhang** received the B.Sc. degree in mathematics, in 2013, from Peking University, Beijing, China, where she is currently working toward the Ph.D. degree in the Department of Scientific and Engineering Computing, School of Mathematical Sciences. She was a visiting student at the Chinese University of Hong Kong, Shenzhen, and supervised by Prof. Zhi-Quan Luo. Her research interests include optimization, signal processing, and wireless communications.

**Zhi-Qiang Yao** (SM’16) received the B.Sc. degree from College of Information Engineering, XiangTan University, Xiangtan, China, in 1997 and the M.S. and Ph.D. degrees from the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China, in 2004 and 2010, respectively. He was a Postdoctoral Fellow at the Chinese University of Hong Kong, Shenzhen, China, in Prof. Zhi-Quan Luo’s research group. Since 2010, he has been with XiangTan University, where he is currently a Professor and the Vice-Dean in the College of Information Engineering. His research interests include signal processing, communication, localization, and optimization. He is Senior Member of the Chinese Institute of Electronics.

**Yixian Liu** received the Ph.D. degree in communication and information system from the South China University of Technology, Guangzhou, China, in 2011. Since then, he was affiliated with Huawei Technologies Company, Ltd., where he held a position of Senior Research Engineer with responsibility for the design of physical layer solutions for the DSL systems. His current research interests include service experience, communication theory, signal processing, big data, and their applications in network design.

**Stephen P. Boyd** (F’99) received an A.B. degree in mathematics *(summa cum laude)* from Harvard University, Cambridge, MA, USA, in 1980, and the Ph.D. degree in EECS from University of California Berkeley, Berkeley, CA, USA, in 1985. He holds honorary doctorates from Royal Institute of Technology (KTH), Stockholm, Sweden, and Catholic University of Louvain (UCL), Louvain-la-Neuve, Belgium. He is currently the Samsung Professor of Engineering, in the Information Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA, USA. He holds courtesy appointments in the Departments of Computer Science and Management Science and Engineering, and is a member of the Institute for Computational Mathematics and Engineering. His current research interests include convex programming applications in control, machine learning, signal processing, finance, and circuit design. He is the author of many papers and several books, including *Convex Optimization* (Cambridge University Press, 2004), coauthored with Lieven Vandenberghe, and *Linear Matrix Inequalities in Systems and Control* (SIAM, 1994) coauthored with El Ghaoui, Feron, and Balakrishnan. His group has created many open source software packages, including the widely used packages for convex optimization CVX, CVXPY, and Convex.jl. He is a Fellow of SIAM and INFORMS, and a member of the National Academy of Engineering.



**Zhi-Quan Luo** (F’07) received the B.Sc. degree in applied mathematics from Peking University, Beijing, China, in 1984, and the Ph.D. degree in operations research from Massachusetts Institute of Technology, Cambridge, MA, USA, in 1989.

From 1989 to 2003, he was on the faculty in the Department of Electrical and Computer Engineering, McMaster University, Canada, where he eventually served as the Department Head and was awarded a Canada Research Chair (Tier I) in Information Processing. From 2003 to 2014, he was a Full Professor in the Department of Electrical and Computer Engineering, University of Minnesota and held an endowed ADC Chair in digital technology. He is currently the Vice-President (Academic) of the Chinese University of Hong Kong, Shenzhen, China, and concurrently the Director of Shenzhen Research Institute of Big Data, Shenzhen, China. His research mainly addresses mathematical issues in information sciences, with particular focus on the design, analysis, and applications of optimization algorithms.

Prof. Luo received the 2010 Farkas Prize from the INFORMS Optimization Society for outstanding contributions to the field of optimization; three Best Paper Awards from the IEEE Signal Processing Society in 2004, 2009, and 2011, respectively; and the 2011 Best Paper Award from the EURASIP. He is a Fellow of the Society for Industrial and Applied Mathematics, and a Fellow of the Royal Society of Canada, the highest honor a Canadian Scholar can achieve in the Arts, Humanities, and Sciences. In 2016, he was elected to the Leading Talent Program of Guangdong Province. He consults regularly with industry on topics related to signal processing and digital communication. He was the semiplenary speaker for the International Symposium on Mathematical Programming in 2003 and IEEE CDC in 2011, the Distinguished Lecturer for the IEEE Sensor Array and Multichannel Signal Processing Workshop in 2006, the plenary speaker for the IEEE Signal Processing Advance for Wireless Communications Workshop in 2013, and IEEE Signal Processing Theory and Methods Workshop in 2014. He was the Chair of the IEEE Signal Processing Society Technical Committee on Signal Processing for Communications. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSINGfrom 2012 to 2014, and was the Associate Editor of many internationally recognized journals, including *Mathematics of Operations Research*, *Management Science*, *Mathematical Programming*.