Abstract—We investigate a relay-aided multi-cell broadcasting system using random network codes, where the focus is on devising efficient scheduling algorithms between relay and base stations. Two scheduling algorithms are proposed based on different feedback strategies; namely, a one-step scheduling algorithm with instantaneous feedback for each redundancy packet; and a multi-step scheduling algorithm with feedback only after multiple redundancy packets. For the latter case, dynamic programming is applied to determine optimal scheduling. Numerical results show that the transmission efficiency of the multi-step algorithm approaches that of the one-step algorithm, but requires significantly less feedback. They both significantly outperform corresponding ARQ and random scheduling approaches.

I. INTRODUCTION

Wide-area coverage for wireless communication is predominantly provided by cellular networks. In such networks the worst-case performance is typically experienced by users at the cell boundaries. Increasing transmitting power is a natural approach to improve performance; however, it may lead to increasing interference to other cells. Alternatively relay nodes [1], [2] may be deployed to assist users at the cell boundaries. It is a common approach that multiple cells share a relay located in the boundary area to save cost, avoid coordination among multiple relays, and reduce interference [3].

We consider a network of adjacent cells with a shared relaying node on their common cell border. With the assistance of the relay, the base stations (BSs) broadcast (down-link) information to a set of users in their respective cells. Due to practical constraints, we only consider a half-duplex relay, i.e., the relay cannot simultaneously transmit and receive information [4]. Also to avoid interference, the BS and the relay cannot transmit simultaneously to the users. Hence, in each cell it is reasonable to consider orthogonal channels for the relay-to-user link and the BS-to-user (and relay) link. A common approach is time-division, in which the relay and the BS transmit in different time slots. The relay assists the cell-edge users of all cells by directional antennas. For every receiver (user or relay), we assume perfect error detection. Thus a received packet is either error-free or removed as being in error. Consequently, the channel can be modeled as a packet-erasure channel [5].

Recently, a broadcast scheme based on random network coding [6] was proposed for a single BS station (one cell without relaying). When the field size used for the network codes is sufficiently large, a receiver is able to decode the original $N$ data packets from any set of $N$ successfully received coded packets [7], [8]. Hence, there is no need to retransmit specific lost packets for each receiver, and thus the transmission efficiency is significantly improved over traditional ARQ [9]. In our multi-cell broadcasting network, we also consider random network codes to increase efficiency. Each BS first transmits $N$ encoded packets to all users of its own cell. Then, the users in all relevant cells feed back the information about the status of the received packets. Redundancy packets are then generated based on the feedback.

A scenario with two cells and one shared relay is considered as an example for illustration purposes. In our setup, the relay is restricted to serve only one of the two cells in any particular time slot. To improve the transmission efficiency, we propose a greedy algorithm to solve the scheduling problem between the two BSs and the relay for the transmission of redundancy packets. We consider two scenarios with different feedback strategies. In the first scenario, the users in both cells provide instantaneous feedback following each redundant packet. In this scenario, we propose a greedy algorithm to solve the scheduling problem between the two BSs and the relay. In the second scenario feedback is provided only after a number of redundancy transmissions. In this case, we formulate a more involved scheduling problem which can be solved through dynamic programming (DP) [11]. The scheduling task spans multiple time slots, which is broken down into a series of simpler scheduling sub-problems over single time slots and solved recursively. Numerical results show that both algorithms gain substantially compared to current benchmarks.

The remainder of the paper is organized as follows. Section II describes our system model, and we formulate the considered scheduling problems in Section III. In Section IV, we develop algorithmic solutions to the formulated scheduling problems, while numerical results are given in Section V.

II. SYSTEM DESCRIPTION

A. System Model

As shown in Fig. 1, we consider a two-cell (cell one and cell two) system with two BSs, denoted by $BS_1$ and $BS_2$, respectively, and with $M_1$ and $M_2$ cell-edge users in cells one and two, respectively. $BS_1$ has $N_1$ information packets, denoted by $I^1_1, \ldots, I^1_{N_1}$, to be broadcasted to the $M_1$ users in cell one, and similarly, $BS_2$ has $N_2$ information packets, denoted by $I^2_1, \ldots, I^2_{N_2}$, to be broadcasted to the $M_2$ users in cell 2. Without loss of generality, we assume $N_1 = N_2 = N$ and $M_1 = M_2 = M$. Cell one and cell two share a
Fig. 1. System model of two BSs, one shared cell-edge relay and two users in each cell.

relay [3], denoted by \( R \), to assist transmissions in both cells. In contrast to multiple cell-edge relays, the shared relay can coordinate transmissions and thus alleviate the need for coordination among base stations. Furthermore, the costs of deploying multiple relays is avoided. We consider the use of directional antennas at the relay to increase coverage, and suppress interference [3], [10]. By exploiting the directionality of the antenna, the relay can transmit to the users in one cell without interfering with the transmission of the BS in another cell. Hence, \( BS_1 \) can transmit to respective cell-edge users simultaneously with the relay transmitting to cell-edge users in cell 2; and vice versa. To constrain complexity and interference, however, the relay only assists one cell in a given time slot. Thus, the concurrent transmission of the relay to two cells is not allowed. Yet clearly, the simultaneous transmissions of the relay and one BS to different cells lead to higher transmission efficiency.

The relay \( R \) is referred to as node \( 2M + 1 \) when taking part of the transmission in cell one. Likewise, the relay is referred to as node \( 2M + 2 \) when assisting the transmission in cell two. To simplify notation, node \( i \in \{1, 2, \ldots, M\} \) refers to user \( i \) in cell one, node \( i \in \{M + 1, \ldots, 2M\} \) refers to user \( i \) in cell two, while node \( i = 2M + 3 \) and node \( i = 2M + 4 \) refer to \( BS_1 \) and \( BS_2 \), respectively. Under the assumption of perfect error detection, a received packet is dropped whenever any error is detected. Hence, the links between nodes are modeled as packet-erasure channels. In cell one, the erasure probability of the channel between node \( i \) and node \( j \) is denoted by \( \mathfrak{e}_{i,j} \), with \( i \in \{2M + 1, 2M + 3\} \), \( j \in \{1, 2, \ldots, M, 2M + 1\} \), and \( i \neq j \). We assume that the \( BS_1 \)-to-relay channel and the relay-to-user channel are better than the \( BS_1 \)-to-user channel, i.e., \( \mathfrak{e}_{2M+1,2M+3} < \mathfrak{e}_{2M+1,i} \) and \( \mathfrak{e}_{2M+3,j} > \mathfrak{e}_{2M+1,i} \), \( i \in \{1, 2, \ldots, M\} \). Note that although the relay might be further away from the BS than the users are, the relay is equipped with a better antenna array than the user terminals. Similarly, we have the following assumptions for cell two: \( \mathfrak{e}_{2M+4,2M+2} < \mathfrak{e}_{2M+4,i} \) and \( \mathfrak{e}_{2M+4,i} > \mathfrak{e}_{2M+2,i} \), \( i \in \{M + 1, \ldots, 2M\} \). Moreover, we assume that all the erasure probabilities are time-invariant and known at the relay node.

To simplify analysis, the transmission suite is divided into two phases: (1) the information transmission phase, in which \( N \) packets are transmitted by each BS. (2) the redundancy transmission phase, in which redundant packets are generated and transmitted by the BSs or relay depending on the transmission schedule. A more detailed description is given later.

B. Network Coding Schemes

Since our network model is symmetric, we illustrate the coding scheme only for cell one. In our network, random linear network coding [12] is used to improve transmission. At \( BS_1 \) (node \( 2M + 3 \)), the coded packet \( \mathbf{C}_{2M+3,k} \) in time slot \( k \) is generated as \( \mathbf{C}_{2M+3,k} = \sum_{j=1}^{N} \mathbf{g}_{j,k}^{1}\mathbf{1}_{j}^{1} \), where \( \gamma_{k}^{1} = [\gamma_{k,1}^{1}, \ldots, \gamma_{k,N}^{1}] \) is the encoding vector of \( \mathbf{C}_{2M+3,k} \). The elements of \( \gamma_{k}^{1} \) are chosen independently and uniformly from the finite field \( GF(q) \). On successfully receiving packets from \( BS_1 \), the relay node does not seek to decode source information. Instead, the received packets are stored in memory for re-encoding. Suppose that \( P \) packets are received by the relay node \( 2M + 1 \) after \( r - 1 > P \) time slots, denoted by \( \mathbf{V}_{1}^{r}, \ldots, \mathbf{V}_{r}^{r} \). A redundancy packet \( \mathbf{C}_{2M+1,r} \) from the relay node \( 2M + 1 \) at time slot \( r \) is then generated as \( \mathbf{C}_{2M+1,r} = \sum_{p=1}^{P} \beta_{1,p}^{1}\mathbf{V}_{p}^{1} \), where \( \beta_{1}^{1} = [\beta_{r,1}^{1}, \ldots, \beta_{r,P}^{1}] \) is the encoding vector of \( \mathbf{C}_{2M+1,r} \). Again, all elements of \( \beta_{1}^{1} \) are chosen independently and uniformly from \( GF(q) \). Assuming that \( \mathbf{V}_{k}^{1} \) is the coded packet \( \mathbf{C}_{2M+3,s} \) from \( BS_1 \), we have

\[
\mathbf{C}_{2M+1,r} = \sum_{p=1}^{P} \beta_{1,p}^{1}\mathbf{C}_{2M+3,s} = \sum_{p=1}^{P} \beta_{1,p}^{1}\sum_{j=1}^{N} \gamma_{j,s}^{1}\mathbf{1}_{j}^{1} = \sum_{j=1}^{N} \left( \sum_{p=1}^{P} \beta_{1,p}^{1}\gamma_{j,s}^{1} \right)\mathbf{1}_{j}^{1}.
\]

Hence, the encoding vector \( \beta_{1}^{1} \) for the coded packet at a receiver (any user in cell one or the relay node \( 2M + 1 \)) may be represented as

\[
\beta_{1}^{1} = [\alpha_{u,1}, \ldots, \alpha_{u,N}],
\]

where \( \alpha_{u,j} = \gamma_{j,s}^{1} \) if the received packet is a coded packet at time slot \( k \) from \( BS_1 \), or \( \alpha_{u,j} = \beta_{1,j}^{1} \gamma_{j,s}^{1} \) if the received packet is from the relay node at time slot \( r \). Note that the packets of different cells will not be mixed in the relay \( R \), which uses different memory locations to store packets of different cells, and generates coded packet separately.

Assuming that a receiver in cell one has \( U \) successfully received packets, \( \mathbf{R}_{1}^{1}, \ldots, \mathbf{R}_{U}^{1} \), at a given time slot, then the corresponding encoding matrix is

\[
\mathbf{G}^{1} = \left( \begin{array}{cccc}
\alpha_{1,1}^{1} & \cdots & \alpha_{1,N}^{1} \\
\vdots & \ddots & \vdots \\
\alpha_{U,1}^{1} & \cdots & \alpha_{U,N}^{1}
\end{array} \right). \tag{3}
\]

The \( N \) information packets \( \mathbf{I}_{1}^{1}, \ldots, \mathbf{I}_{N}^{1} \) can be recovered if \( \text{rank}(\mathbf{G}^{1}) = N \) by e.g., Gaussian elimination [13]. The same requirements hold true for recovering source packets \( \mathbf{I}_{1}^{1}, \ldots, \mathbf{I}_{N}^{1} \) based on encoding matrix \( \mathbf{G}^{2} \) for each receiver in cell two.

Clearly, the ranks of the encoding matrices for the users and the relay are essential measures for transmission. Using the terminology from [7], a received packet is referred to as an innovative packet for the receiver if the rank of the corresponding encoding matrix increases by one. To simplify the analysis in the following sections we make two assumptions, which hold true when the field size \( q \) is sufficiently large [7] [12]. Firstly, we assume that a received coded packet from the BS is an innovative packet if the rank of the encoding matrix at the receiver is less than \( N \). Secondly, if the rank of the encoding matrix at the relay is higher than the rank of the encoding matrix at the receiver, then a received redundancy packet from the relay is an innovative packet for the receiver.
If the rank of the encoding matrix at the relay is smaller than or equal to the rank of the encoding matrix at the receiver, then the probability that a received redundancy packet from the relay is an innovative packet for the receiver is low and the transmission of the packet is invalid.

C. Transmission Schemes

To avoid inter-cell interference, $BS_1$ and $BS_2$ cannot transmit simultaneously. Further, $BS_1$ (node $2M + 3$) and the relay node $2M + 1$ cannot transmit simultaneously to avoid interference to the users of cell one. For the same reason, $BS_2$ (node $2M + 4$) and the relay node $2M + 2$ cannot transmit simultaneously. Moreover, the relay node operates in half-duplex mode and can only assist one cell in one time slot. Due to these constraints, only the following six actions are allowed in one time slot:

1) $a = 1$: $BS_1$ (node $2M + 3$) transmits and all the receivers (nodes $i = 1, ..., M, 2M + 1$) in cell one receive;
2) $a = 2$: $BS_2$ (node $2M + 4$) transmits and all the receivers (nodes $i = M + 1, ..., 2M, 2M + 2$) in cell two receive;
3) $a = 3$: $BS_1$ (node $2M + 3$) transmits. At the same time, the relay node $2M + 2$ transmits to the cell-edge users in cell two. All the users in cell one (nodes $i = 1, ..., M$) receive the packet from $BS_1$ (node $2M + 3$). All the users in cell two (nodes $i = M + 1, ..., 2M$) receive the packet from the relay node $2M + 2$.
4) $a = 4$: $BS_2$ (node $2M + 4$) transmits. At the same time, the relay node $2M + 1$ transmits to the cell-edge users in cell one. All the users in cell two (nodes $i = M + 1, ..., 2M$) receive the packet from $BS_2$ (node $2M + 4$). All the users in cell one (nodes $i = 1, ..., M$) receive the packet from the relay node $2M + 1$.
5) $a = 5$: the relay node $2M + 1$ transmits and all the users in cell one receive (nodes $i = 1, ..., M$).
6) $a = 6$: the relay node $2M + 2$ transmits and all the users in cell two receive (nodes $i = M + 1, ..., 2M$).

Clearly, to recover the source information packets at the user side, at least $N$ packets are needed. Thus, $BS_1$ and $BS_2$ broadcast $N$ coded packets to their respective users and the relay in the initial transmission phase. Some packets are lost due to erasures in the BS-to-user channels and the BS-to-relay channel. At the end of the phase, each user feeds back the rank of the respective encoding matrix to the relay and BSs. For simplicity we assume orthogonal and error-free feedback channels. In the redundancy phase, redundant packets are transmitted by the BSs or relay according to a scheduling algorithm as detailed in the following.

To quantify the efficiency of the scheduling algorithms, we define the overhead by,

$$\eta = \frac{X}{2N},$$  \hspace{1cm} (4)

where the factor of 2 in the denominator accounts for the number of participating cells and $X$ is the total number of time slots required for all users to have successfully received their respective $N$ packets. Our objective is to minimize the overhead $\eta$, which is equivalent to maximizing the average information received by all users (rank increase) in one time slot.

III. Problem Formulation

In the redundancy phase, the rank of the encoding matrix of user $i$ in cell one increases by one if one packet from $BS_1$ (node $2M + 3$) is successfully received by the user, unless the encoding matrix is already full-rank. Yet this may not hold true for a packet received successfully from the relay node. Since random packet-erasures occur on the $BS_1$-to-relay channel, it is possible that the rank of the encoding matrix for broadcasting information in cell one at the relay node $2M + 1$ is lower than the rank of the encoding matrix at user $i$. Therefore, user $i$ cannot receive an innovative packet from the relay node $2M + 1$ [7]. However, the $BS_1$-to-user channels have higher packet-erasure probabilities than the corresponding relay-to-user channels. Thus, it follows that there is an inherent scheduling problem of whether $BS_1$ or the relay node $2M + 1$ should transmit in cell one. The same scheduling problem between $BS_2$ and the relay node $2M + 2$ exists in cell two. This is clearly an intra-cell scheduling problem.

Moreover, there are inter-cell scheduling problems between these two cells. Firstly, the two BSs cannot transmit at the same time, which leads to a BS selection problem. Also, when anyone of the BSs transmits, the relay can assist the transmission in the other cell. However, when the half-duplex relay transmits, it cannot concurrently receive innovative packet from any BS. Yet the relay needs packet from the BSs to be able to assist users. Likewise, when the relay receives packets from a BS, it cannot transmit to the other cell. Hence, the shared relay scheduling is another important issue.

We investigate the scheduling problem of both intra- and inter-cell scheduling. Based on different requirements on the feedback, two scheduling problems will be studied. First, instantaneous feedback is available from all users in both cells for each redundancy packet. Then, the system can schedule the relay and BSs transmissions for the following time slot. We denote this scenario as one-step scheduling since we schedule redundancy packets for only one time slot ahead.

Some of the drawbacks of this strategy are that feedback requires resources and leads to delay. Hence, one-step scheduling may not be suitable, especially for large-scale networks. When the number of users in the system is large, the resources and resulting latency required for feedback may be prohibitive. Therefore in the second scenario we consider multiple-step scheduling, as an alternative requiring less feedback. The feedback is transmitted after a block of redundancy packets, and thus, redundant packets are scheduled for multiple packets. In this case, dynamic programming (DP) is a natural approach for optimizing scheduling.

A. States

We define the state of the system as a vector of length $2M + 2$, where the element $\ell_i, i \in \{1, ..., 2M + 2\}$ is the rank of the encoding matrix of node $i$. Clearly, each element of a rank vector belongs to the set $\{0, 1, 2, ..., N\}$. Thus, there are in total $(N + 1)^{(2M+2)}$ states, where state $j \in J = \{1, 2, ..., (N + 1)^{(2M+2)}\}$ of the system is expressed as,

$$S^j = [s_1^j, s_2^j, ..., s_{2M+2}^j].$$  \hspace{1cm} (5)
B. Actions

As defined in Section II-C, the actions are \( a \in A = \{1, 2, ..., 6\} \) for the six-action case. For the four-action case in Section V, the actions are \( a \in A = \{1, 2, 5, 6\} \).

C. Transition Probability Function

The state transition probability from state \( S^i \) to state \( S^j \) caused by action \( a \) is denoted by \( p_{i,j,a} = \Pr[S^j|S^i, a] \), which can be easily computed based on the corresponding channel erasure probabilities. For each valid action, we can construct a corresponding probability transition matrix \( \mathbf{P}_a \) as,

\[
\mathbf{P}_a = \begin{pmatrix}
p_{1,1,a} & \cdots & p_{1,(N+1)2M+2,a}
\vdots & \ddots & \vdots
p_{(N+1)2M+2,1,a} & \cdots & p_{(N+1)2M+2,(N+1)2M+2,a}
\end{pmatrix}.
\]

The size of \( \mathbf{P}_a \) is \((N+1)2M+2 \times (N+1)2M+2\), where \( p_{i,j,a} \) is the element in row \( i \), column \( j \).

D. Benefit Function

We define the benefit of the transition from state \( S^i \) to state \( S^j \) as the summation of the corresponding rank increases of the encoding matrices (innovative packets) of all users. Formally we have the benefit \( B_{i,j} \) as

\[
B_{i,j} = \sum_{k=1}^{2M} (s^i_k - s^j_k).
\]  

(6)

IV. SCHEDULING ALGORITHMS

In this section we describe in more detail the one-step and the multi-step scheduling algorithms.

A. One-Step Scheduling Algorithm

As discussed above, when all users provide feedback for each redundancy packet, we can use a one-step scheduling algorithm. Given the state of the system, a greedy algorithm can be used to maximize the expected rank increase of the encoding matrices of all the users for the following time slot. The one-step algorithm is detailed as follows. If we assume the current state of the system is \( S^b \), where \( b \in J = \{1, ..., (N+1)(2M+2)\} \), then the action of the system, \( A^* \), is defined as

\[
A^* = \arg \max_{a \in A} E[B_{b,j}|a] = \arg \max_{a \in A} \sum_{j \in J} p_{b,j,a} B_{b,j},
\]

where \( E[B_{b,j}|a] \) is the expected value of \( B_{b,j} \) for a given action \( a \). As the algorithm is provided with feedback information following each redundancy transmission, then the one-step overhead performance gives a lower bound on the overhead for the multi-step algorithm.

B. Multiple-Step Scheduling Algorithm

To reduce the requirements for feedback information we now consider to schedule multiple redundancies transmission. Assume that the system is in state \( S^b \), where \( b \in J = \{1, 2, ..., (N+1)(2M+2)\} \), at a certain time of a redundancy transmission. Without loss of generality, let \( t = 0 \) denote the corresponding time slot. To obtain full-rank encoding matrices for all the users, at least \( \chi = \max_{k \in \{1, 2, ..., 2M\}} (N-s^b_k) \) redundancy packets are required. Thus, at least \( \chi \) time slots are required for redundancy transmission. Based on this observation, our task is to schedule the redundancy transmission for the following \( \chi \) time slots without any further feedback. To find the optimal scheduling by brute-force is prohibitive due to the size of the state-space. Hence, we propose to use the DP algorithm to efficiently solve this problem [11]. The DP algorithm breaks down the joint problem over \( \chi \) time slots into \( \chi \) simpler sub-problems, one for each time slot, which can be solved in a recursive manner.

The approach proceeds as follows. Let \( a_t \), where \( t = 1, 2, ..., \chi \), denote the optimal action for step \( t \). Then, the optimal actions for the future \( \chi \) steps can be denoted by \( A^\chi = [a_1, ..., a_\chi] \). The algorithm proceeds backwards in time, starting from \( t = \chi \) to \( t = 0 \). We denote the state of the system before step \( t \) by \( \beta_{t-1} \), where \( t = 1, 2, ..., \chi \). We initialize the optimal action \( a_1 \) for state \( \beta_{\chi-1} = S^1 \), where \( i \in J \), to maximize the possible gain; namely,

\[
A(i, \chi) = \arg \max_{a \in A} \{ \max_{j \in J} p_{i,j,a} B_{i,j} \}.
\]  

(7)

The corresponding maximum possible gain is defined by

\[
\overline{B}(i, \chi) = \max_{j \in J} p_{i,j,A(i,\chi)} B_{i,j}.
\]  

(8)

We define a sequence of actions as \( A_s(i, t) \), where \( i \in J \) and \( t = 1, 2, ..., \chi \), to record the optimal sequence of actions from \( t \) to \( \chi \), given the state at \( t-1 \), \( \beta_{t-1} = S^t \), where \( i \in J \). The length of each sequence \( A_s(i, t) \) is \( \chi-t+1 \), and the sequence at \( t = \chi \) is initialized as,

\[
A_s(i, \chi) = A(i, \chi).
\]  

(9)

Based on (7), (8) and (9), the optimal action at step \( t \), with \( \beta_{t-1} = S^t \), where \( t = 1, 2, ..., \chi \) and \( i \in J \), is obtained by a DP recursion as

\[
A(i, t) = \arg \max_{a \in A} \max_{j \in J} \{ p_{i,j,a} (B_{i,j} + \overline{B}(j, t+1)) \}.
\]

The possible gain for the corresponding optimal action given \( \beta_{t-1} = S^t \) is,

\[
\overline{B}(i, t) = \max_{j \in J} \{ p_{i,j,A(i,t)} (B_{i,j} + \overline{B}(j, t+1)) \}.
\]

We assume the maximal gain \( \overline{B}(i, t) \) is obtained based on \( \beta_t = S^t(i, t) \), and thus,

\[
j^*(i, t) = \arg \max_{j \in J} \{ p_{i,j,A(i,t)} (B_{i,j} + \overline{B}(j, t+1)) \}.
\]

Then, \( A_s(i,t) \) is given by,

\[
A_s(i,t) = [A(i,t), A_s(j^*(i,t), t+1)].
\]  

(10)

At the end of the recursion, we obtain \( A_s(i,1) \) for \( i \in J = \{1, 2, ..., (N+1)(2M+2)\} \). Since the starting state is \( S^b \), then \( \beta_0 = S^b \), and thus, the optimal redundancy transmission schedule \( A^* \) for the next \( \chi \) time slots is determined by

\[
A^* = A_s(b, 1),
\]  

(11)

where \( A_s(b, 1) \) is defined as the optimal sequence of actions from \( t = 1 \) to \( t = \chi \) based on \( \beta_0 = S^b \).
V. NUMERICAL RESULTS

We consider $M = 2$ for illustration. The notation for each node in the system is provided in Table I. We also make the assumptions that the two cells are symmetric and the BS-to-user and relay-to-user links are symmetric across the users, i.e., the erasure probabilities of the links satisfy $\varepsilon_{7,1} = \varepsilon_{7,2} = \varepsilon_{8,3} = \varepsilon_{8,4}, \varepsilon_{7,5} = \varepsilon_{8,6}$, and $\varepsilon_{5,1} = \varepsilon_{5,2} = \varepsilon_{6,3} = \varepsilon_{6,4}$. We show the performance in terms of transmission overhead for the one-step scheduling and the DP-based multiple-step scheduling algorithms. To evaluate the gain from simultaneous transmissions ($a = 3, 4$), we provide results for a four-action case, where we only have actions $a = 1, 2, 5, 6$. Moreover, the overhead performance applying random scheduling or ARQ are given as well for comparison.

In Fig. 2, the overhead performance is shown as a function of the number of information packets $N$ to be broadcasted in each cell. From the figure, we observe that the performance of all but one of the algorithms improves with $N$. The exceptions are the ARQ schemes. Our proposed scheduling algorithms perform better than the random scheduling algorithms and the ARQ schemes. For both four-action and six-action (all actions are used) cases, the performance loss of the DP-based algorithm compared to the corresponding one-step algorithm is small. Moreover, the six-action algorithms perform better than the four-action algorithms.

In Fig. 3, the overhead performance is shown as a function of the erasure probability of the BS-to-relay link ($\varepsilon_{7,5}$). Comparing the DP-based multi-step algorithm with the corresponding one-step algorithm, there is only a minor loss of performance. Yet, the DP-based algorithms need significantly less feedback. As shown in the figure, with the increase of $\varepsilon_{7,5}$, the performance gaps between the six-action algorithms and the four-action algorithms become small. When $\varepsilon_{7,5} = 0$, the relay node always transmits to one of the two cells in the redundancy phase. The gains of the six-action algorithms compared to the corresponding four-action algorithms come from the assistance to the BS in the other cell (the additional two actions). However, when $\varepsilon_{7,5} = 0.5$, which equals the erasure probability of the BS-to-user link, almost all the packets at the user side are obtained from the transmission of the BSs. When the actions of simultaneous transmission in two cells are chosen, it is with high probability that the packets from the relay are non-innovative for the users. This explains why the performance of the six-action and four-action algorithms are comparable at the $\varepsilon_{7,5} = 0.5$ point.

Fig. 4 shows the overhead performance as a function of the erasure probability of the BS-to-user link ($\varepsilon_{7,1}$). As shown in the figure, the six-action algorithms are always better than the four-action algorithms. When $\varepsilon_{7,1} = 0.2$, the six-action algorithms have similar performance as the four-action algorithms. This is based on the same reason as the $\varepsilon_{7,5} = 0.5$ point in Fig. 3. With the increase of $\varepsilon_{7,1}$, the performance gap increases. However, the performance gaps decrease with the further increase of $\varepsilon_{7,1}$. When $\varepsilon_{7,1}$ is sufficiently large, most of the received packets at the user side are transmitted by the relay. When simultaneous transmission actions are allowed, the packets transmitted by the BSs are hardly received successfully. Thus, the performance gaps between the six-action algorithms and the four-action algorithms decreases.

VI. CONCLUSIONS

We study efficient relay-aided multi-cell broadcasting with random network coding. Scheduling algorithms are proposed...
to maximize the transmission efficiency. We use a system with two cells and one shared relay for illustration. The scheduler chooses one of six pre-defined actions for a redundancy packet transmission. Based on instantaneous feedback information after each time slot, we propose a greedy algorithm to determine the schedule for the next time slot. Furthermore, we propose a DP-based scheduling algorithm, which requires feedback only after multiple redundancy packets. The results show that significant performance gains are achieved by our schemes, compared to the random scheduling approaches and ARQ schemes. Moreover, we also provide four-action DP algorithms to show the performance gain the six-action DP schemes, compared to the random scheduling approaches and ARQ schemes. The performance gap between the greedy algorithm and the corresponding DP-based algorithm is small.

APPENDIX: STATE TRANSFER PROBABILITIES

Since the calculations in both cells are symmetric, we illustrate only for $a = 1, 3, 6$. We set $K_1 = \{1, ..., M, 2M+1\}$ and $K_2 = \{M+1, ..., 2M, 2M+2\}$. Moreover, we define a function $I(A, B)$ to indicate whether $A$ is larger than $B$ or not, which is,

$$I(A, B) = \begin{cases} 1 & \text{if } A < B, \\ 0 & \text{otherwise}. \end{cases}$$

(12)

We first compute $p_{i,j,1}$. The rank of the encoding matrix of receiver $k \in K_1$ cannot increase by more than one. The rank of node $k \in K_2$ keeps the same. Moreover, the rank of a node in the system cannot decrease. Then, $p_{i,j,1} = 0$ if $\exists s_k' = s_k$, $k \in K_1$; or $s_k' - s_k > 1, k \in K_1$; or $s_k' \neq s_k$, $k \in K_2$. Otherwise, we have $p_{i,j,1} = \prod_{k \in K_1} e_{2M+3,k,0} \prod_{k \in K_2} (1 - e_{2M+3,k})$, where $e_{2M+3,k,0} = I(s_{k}', N)[1 - (s_{k}' - s_k)]$ and $e_{2M+3,k,1} = I(s_{k}', N)(s_{k}' - s_k)$. If the rank of receiver $k$ increases by one, the coded packet is successfully received and the receiver did not have a full-rank matrix at the previous state $S'$: $e_{2M+3,k,1} = 1$ and $e_{2M+3,k,0} = 0$. If the rank of receiver $k$ does not change, and it was not a full-rank matrix at $S'$, then $e_{2M+3,k,1} = 0$ and $e_{2M+3,k,0} = 1$. If receiver $k$ has full-rank, it will not contribute to $p_{i,j,1}$, namely, $e_{2M+3,k,1} = 0$ and $e_{2M+3,k,0} = 0$.

We next determine $p_{i,j,6}$. When this action is chosen, none of the nodes $k \in K_1$ can get an innovative packet. If the rank of the encoding matrix of node $k \in K_2 \backslash \{2M+2\}$ is larger than the encoding matrix at the relay node $2M+2$, the rank of the node cannot increase. Thus, $p_{i,j,6} = 0$ if $\exists s_k' - s_k > 1, k \in K_2 \backslash \{2M+2\}$, or $s_k' < s_k$, $k \in K_2 \backslash \{2M+2\}$, or $s_k' - s_k = 1$ and $s_{2M+2} \leq s_k', k \in K_2 \backslash \{2M+2\}$, or $s_{2M+2} \neq s_k', k \in K_1$. The third condition means that if the rank of the relay is smaller than the rank of node $k$, then the rank of user $k$ cannot increase. The fourth condition means that the rank of the relay cannot change since it cannot receive while it is transmitting. If none of the five conditions occur, we have $p_{i,j,6} = \prod_{k \in K_2 \backslash \{2M+2\}} (1 - e_{2M+2,k})$, where $e_{2M+2,k,0} = I(s_k', s_{2M+2})[1 - (s_k' - s_k)]$ and $e_{2M+2,k,1} = I(s_k', s_{2M+2}')(s_k' - s_k')$.

We now determine $p_{i,j,3}$. Since node $2M+2$ transmits, the ranks of both nodes $2M+1$ and $2M+2$ cannot change. The ranks of all the nodes cannot decrease or increase by more than one. Thus, $p_{i,j,3} = 0$ if $\exists s_k' - s_k < 1, k \in K_1 \backslash K_2$ or $\exists s_k' < s_k, k \in K_1 \cup K_2$ or $s_{2M+2} \neq s_k', k \in K_2 \backslash \{2M+2\}$ or $s_k' \neq s_k$, $k \in K_1 \cup K_2$. Based on the similar reasons as previous calculations, we have $p_{i,j,3} = p_{i,j,6}$, where $p_{c1} = \prod_{k \in K_1 \backslash \{2M+1\}} e_{2M+3,k,0} (1 - e_{2M+2,k}) e_{2M+3,k,1}$ and $p_{c2} = \prod_{k \in K_1 \backslash \{2M+2\}} e_{2M+2,k,0} (1 - e_{2M+2,k}) e_{2M+2,k,1}$, in which, $e_{2M+3,k,0} = I(s_k', N)(1 - (s_k' - s_k))$, $e_{2M+3,k,1} = I(s_k', N)(s_k' - s_k)$, $e_{2M+2,k,0} = I(s_k', s_{2M+2})[1 - (s_k' - s_k)]$ and $e_{2M+2,k,1} = I(s_k', s_{2M+2}')(s_k' - s_k')$.

ACKNOWLEDGMENT

The work has been supported in parts by the Swedish VINNOVA under the “Joint Sweden-China Strategic Cooperation Program”, the ARC Grant DP0986089, the Swedish Research Council under VR grant 621-2008-4249 and 621-2009-4666, and the European Research Council under the Seventh Framework Programme ERC Grant 228044.

REFERENCES