Optimal multiple-relay selection in dual-hop amplify-and-forward cooperative networks

Xiaohua Li

Algorithms and rules for the optimal selection of multiple amplify-and-forward relays are developed in a dual-hop cooperative network. A linear fractional programming approach is first formulated. Then, a surprisingly simple water-filling-like closed-form solution is derived, which shows many interesting properties of the optimal relay selection.

Introduction: Cooperative communication has attracted much attention and has achieved rapid progress in the last few years because of its advantage of exploiting redundant nodes as relays to boost transmission performance. Nevertheless, while cooperative communication has been an active research area for years, a fundamental issue remains challenging and open, i.e. how to select relays optimally from all available redundant nodes. The relay selection problem can in general be divided into single-relay selection and multiple-relay selection problems [1]. The former has been studied extensively [2]. Unfortunately, the latter is more challenging. Some suboptimal or heuristic methods have thus been proposed, e.g. selecting relays closer to the source node, or selecting relays with SINR above a certain heuristic threshold [3]. In this Letter, we give an optimal yet simple result on multiple-relay selection.

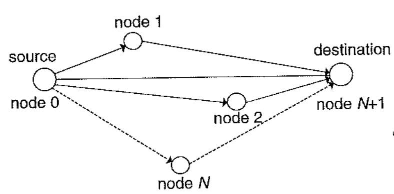

Fig. 1 *Dual-hop cooperative wireless network with N candidate relay nodes*

Dual-hop cooperative network: We consider a wireless ad hoc network with a source node (node 0), a destination node (node $N+1$) and N other nodes that can potentially work as relays, as illustrated in Fig. 1. The edge (i,j) from node i to j has complex gain g_{ij}. For any node i that is selected as the relay, we adopt the following dual-hop two-phase amplify-and-forward relaying scheme. During the first phase, the source node broadcasts the signal $s(t)$ to all the other nodes. The signal received by the node i is

$$x_i(t) = g_{0i}\sqrt{P}s(t-\tau_{0i}) + v_i(t), \quad i = 1, \cdots, N+1 \tag{1}$$

where τ_{0i} is the propagation delay, P is the maximum transmission power of any node, and $v_i(t)$ is the additive white Gaussian noise (AWGN). During the second phase, each relay i amplifies the received signal $x_i(t)$ to certain transmission power P_i and transmits signal

$$s_i(t) = \sqrt{\frac{P_i}{E[|x_i(t)|^2]}}x_i(t), 0 \le P_i \le P \tag{2}$$

The source node may transmit again the signal $s_0(t) = s(t)$ in this phase with a new transmission power P_0, where $0 \le P_0 \le P$. The destination node's received signal is

$$x_d(t) = \sum_{i=0}^{N} g_{i,N+1}s_i(t-\tau_{i,N+1}) + v_d(t) \tag{3}$$

where $\tau_{i,N+1}$ is the propagation delay and $v_d(t)$ is the AWGN.

We assume that all AWGNs are independent from each other and from $s(t)$, and have zero mean and variances $E[|v_i(t)|^2] = \sigma_i^2$. We define the nominal SINR for edge (i,j) as

$$\gamma_{ij} = \frac{P|g_{ij}|^2}{\sigma_j^2} \tag{4}$$

Linear fractional programming: From (1), the destination node has SINR $\gamma_d^{(1)} = P|g_{0,N+1}|^2/\sigma_{N+1}^2$ in the first phase. During the second phase, the destination's signal (3) can be written as

$$\begin{aligned} x_d(t) = {} & \sum_{i=1}^{N} g_{i,N+1}\frac{\sqrt{P_i}}{\sqrt{P|g_{0i}|^2+\sigma_i^2}}g_{0i}\sqrt{P}s(t-\tau_{0i}-\tau_{i,N+1}) \\ & + g_{0,N+1}\sqrt{P_0}s(t-\tau_{0,N+1}) \\ & + \sum_{i=1}^{N} g_{i,N+1}\frac{\sqrt{P_i}}{\sqrt{P|g_{0i}|^2+\sigma_i^2}}v_i(t-\tau_{i,N+1}) + v_d(t) \end{aligned} \tag{5}$$

The SINR of (5) is

$$\gamma_d^{(2)} = \frac{\sum_{i=1}^{N}|g_{i,N+1}|^2\frac{P_i}{P|g_{0i}|^2+\sigma_i^2}|g_{0i}|^2P + |g_{0,N+1}|^2P_0}{\sum_{i=1}^{N}|g_{i,N+1}|^2\frac{P_i}{P|g_{0i}|^2+\sigma_i^2}\sigma_i^2 + \sigma_d^2} \tag{6}$$

With the nominal edge SINR (4), we can readily rewrite the overall destination SINR as

$$\gamma = \gamma_d^{(2)} + \gamma_d^{(1)} = \frac{a_{N+1}z_0 + \sum_{i=1}^{N}a_ib_iz_i}{1+\sum_{i=1}^{N}b_iz_i} + a_{N+1} \tag{7}$$

where $a_i = \gamma_{0i}$, $b_i = \frac{\gamma_{i,N+1}}{\gamma_{0i}+1}$, $z_i = \frac{P_i}{P}$. Then, the relay selection problem becomes the optimisation

$$\max_{\{z_i|0\le z_i\le 1,\ i=0,\cdots,N\}} \gamma \tag{8}$$

It is easy to see that $\gamma_d^{(1)} = a_{N+1}$ is a constant, and $\gamma_d^{(2)}$ is in a linear fractional programming (LFP) form. Therefore, the optimisation (8) can be solved by standard linear programming algorithms [4].

Optimal relay selection rule: It is more desirable to look for a closed-form solution to (8). Surprisingly, there is such a simple yet optimal closed-form solution, as shown below:

Proposition: Optimal multiple-relay selection (8) leads to $P_0 = P$ and for all other nodes $i = 1,\cdots,N$

$$P_i = \begin{cases} P, & \text{if } \sum_{j=1}^{N}\left(\frac{1}{\gamma_{0i}}-\frac{1}{\gamma_{0j}}\right)^+\frac{\gamma_{j,N+1}}{1+\frac{1}{\gamma_{0j}}} < 1-\frac{\gamma_{0,N+1}}{\gamma_{0i}} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $(y)^+ = \begin{cases} y, & \text{if } y>0 \\ 0, & \text{if } y\le 0 \end{cases}$. The optimal SINR γ can then be calculated from (7) by letting $z_i = 1$ or $z_i = 0$ according to (9).

Proof: Consider (7). Obviously, for maximum γ, we should have $z_0 = 1$ since $a_{N+1} \ge 0$. Then we just need to maximise $f(\{z_i\}) = \frac{a_1b_1z_1+\cdots+a_Nb_Nz_N+a_{N+1}}{b_1z_1+\cdots+b_Nz_N+1}$. Consider any two nodes i,j with $a_i \le a_j$. We can rewrite $f(\{z_i\})$ as $f(\{z_i\}) = (a_ib_iz_i + a_jb_jz_j + \alpha)/(b_iz_i + b_jz_j + \beta)$ where α, β denote all the remaining items. Then we change to $f(\{z_i\}) = a_i + h_i/(b_iz_i + b_jz_j + \beta)$ where $h_i = \alpha - a_i\beta + (a_j - a_i)b_jz_j$. Because all the variables a_i, b_i, z_i are non-negative, to maximise $f(\{z_i\})$ we should have: if $h_i \ge 0$ then $z_i = 0$; and if $h_i < 0$ then $z_i = 1$. Next, we change $f(\{z_i\})$ similarly to $f(\{z_i\}) = a_j + h_j/(b_iz_i + b_jz_j + \beta)$ where $h_j = \alpha - a_j\beta + (a_i - a_j)b_iz_i$. It is easy to verify that $h_i - h_j \ge 0$. Then for all the following three cases, $0 \ge h_i \ge h_j$, $h_i \ge 0 \ge h_j$, and $h_i \ge h_j \ge 0$, we must have, respectively, $\{z_i = 1, z_j = 1\}$, $\{z_i = 0, z_j = 1\}$, and $\{z_i = 0, z_j = 0\}$. This result means that for any two nodes with $a_i \le a_j$, if $z_j = 0$ then $z_i = 0$; and if $z_i = 1$ then $z_j = 1$. In other words, there exists a node i^* such that for all the nodes $j = 1, \cdots, N$ we must have

$$z_j = \begin{cases} 0, & \text{if } a_j < a_{i^*} \\ 1, & \text{if } a_j \ge a_{i^*} \end{cases} \tag{10}$$

Therefore, to determine whether a node i can work as a relay, we let $z_j = 0$ for all $a_j < a_i$ and $z_j = 1$ for all $a_j \ge a_i$. Then we have

$+1) \triangleq \frac{a_i b_i z_i + \tilde{\alpha}}{b_i z_i + \tilde{\beta}}$, which can be changed to $f(z_i) = a_i + (\tilde{\alpha} - a_i\tilde{\beta})/(b_i z_i + \tilde{\beta})$. If $\tilde{\alpha} - a_i\tilde{\beta} < 0$ we must have $z_i = 1$, which is the same as (9). Otherwise, if $z_i = 0$ instead, then according to (10) we have $a_{i*} > a_i$. Considering $f(z_{i*}) = a_{i*} + (\alpha^* - a_{i*}\beta^*)/(b_{i*}z_{i*} + \beta^*)$, it is easy to verify that $\alpha^* - a_{i*}\beta^* \leq \tilde{\alpha} - a_i\tilde{\beta} < 0$. This means $z_{i*} = 0$, a contradiction to (10). This concludes the proof.

Some interesting properties can be observed from this proposition. First, relays either use full transmission power or stop transmission. Secondly, the higher the nominal SINR, the higher the priority of using this edge (and thus this relay). Thirdly, since $f(x) = \sum_{j=1}^{N} \left(\frac{1}{x} - \frac{1}{\gamma_{0j}}\right)^+ \frac{\gamma_{j,N+1}}{1 + 1/\gamma_{0j}} + \frac{\gamma_{0,N+1}}{x} - 1$ is monotone non-increasing, we can find an x^* such that $f(x^*) = 0$. Then nodes (edges) with nominal SINR $\gamma_{0i} > x^*$ will be selected as relays. The procedure is somewhat similar to the classical water-filling principle.

Simulation results: We simulated a random wireless ad hoc network where the nodes' positions were randomly generated within a square of 1000×1000 m. Nominal edge SINRs were calculated as $\gamma_{ij} = 10^8 d_{ij}^{-2.6}$, where d_{ij} is the propagation distance. Source/destination nodes were fixed with distance $d_{0,N+1} = 1000$ m. We simulated the LFP algorithm ('LFP'), our new rule (9), and two heuristic methods: one uses a single relay closest to the source ('use single relay'), and the other uses all the N relay nodes ('use all relays'). Baseline source–destination transmission ('S-D direct') SINR was compared as well. Simulation results in Fig. 2 indicate that our optimal relay selection rule (9) gave the same results as LFP, and thus is optimal. They both achieved the optimal destination SINR.

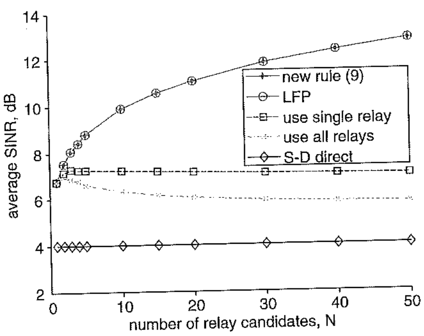

Fig. 2 *Average signal to interference and noise ratio (SINR) against number of relay candidates*

Conclusion: For a dual-hop amplify-and-forward cooperative network, we give a closed-form solution to the problem of multiple-relay selection for maximum destination SINR. It shows many interesting properties of the optimal relay selection.

20 January 2012
doi: 10.1049/el.2012.0194
One or more of the Figures in this Letter are available in colour online.
Xiaohua Li (*Department of Electrical and Computer Engineering, State University of New York at Binghamton, Binghamton, NY 13902, USA*)

E-mail: xli@binghamton.edu

References

1 Li, Y., Wang, P., Niyato, D., and Zhuang, W.: 'A dynamic relay selection scheme for mobile users in wireless relay networks'. IEEE INFOCOM'2011, Shanghai, China, April 2011
2 Bletsas, A., Shin, H., and Win, M.Z.: 'Outage optimality of opportunistic amplify-and-forward relaying', *IEEE Commun. Lett.*, 2007, **11**, (3), pp. 261–263
3 Jing, Y., and Jafarkhani, H.: 'Single and multiple relay selection schemes and their achievable diversity orders', *IEEE Trans. Wirel. Commun.*, 2009, **8**, (3), pp. 1414–1423
4 Boyd, S.P., and Vandenberghe, L.: 'Convex optimization' (Cambridge University Press, 2004)

K. Niu and K. Chen

A successive cancellation stack (SCS) decoding algorithm is proposed to improve the performance of polar codes. Unlike the conventional successive cancellation decoder which determines the bits successively with a local optimal strategy, the SCS algorithm stores a number of candidate partial paths in an ordered stack and tries to find the global optimal estimation by searching along the best path in the stack. Simulation results in the binary-input additive white Gaussian noise channel show that the SCS algorithm has the same performance as the successive cancellation list (SCL) algorithm and can approach that of the maximum likelihood algorithm. Moreover, the time complexity of the SCS decoder is much lower than that of the SCL and can be very close to that of the SC in the high SNR regime.

Introduction: Polar codes (PCs) have been proved to achieve the capacity of symmetric binary input discrete memoryless channels (B-DMCs) under a successive cancellation (SC) decoding [1]. However the finite-length performance under SC is not satisfactory. The belief propagation (BP) decoding algorithm can improve the performance [2, 3], whereas the optimal scheduling of messages in BP is hard to know. In [4], a linear programming (LP) decoder is introduced; unfortunately, it cannot work on channels other than the binary erasure channel (BEC). Moreover, there is a gap between the BP/LP algorithm and the maximum likelihood (ML) algorithm. The successive-cancellation list (SCL) decoder [5] can approach the performance of the ML decoder for a moderate list size L. Nevertheless, the time complexity of the SCL decoder is fixed and slightly high. Inspired by the stack decoding of the convolutional code [6], we propose an alternative decoding strategy called the successive cancellation stack (SCS) algorithm. Compared with the SCL decoder, the SCS algorithm can achieve the same performance and has lower time complexity.

Polar coding: In this Letter, we apply the same notation defined in [1]. We assume communication over a symmetric B-DMC $W : \{0, 1\} \to \mathbb{R}$. Let $N = 2^n$ denote the code length. After channel combining, we get a vector channel $W_N(y_1^N|u_1^N) = \prod_{i=1}^{N} W(y_i|x_i)$, where $x_1^N = u_1^N G_N$, G_N is the generator matrix defined in [1], $u_1^N, x_1^N \in \{0, 1\}^N$ are the source and code block, respectively, $y_1^N \in \mathbb{R}^N$ is the received signal and $W(y_i|x_i)$ is the channel transition probability of W. By applying channel splitting, the transition probability of the polarised subchannel $W_N^{(i)}$ can be defined as

$$W_N^{(i)}(y_1^N, u_1^{i-1}|u_i) = \frac{1}{2^{N-1}} \sum_{u_{i+1}^N} W_N(y_1^N|u_1^N) \qquad (1)$$

where u_1^{i-1}, u_{i+1}^N are the subvectors of u_1^N. Moreover, the transition probabilities in (1) can be easily calculated by a recursive structure [1]. After using Bhattacharyya parameters to calculate the reliability, the information bits $u_{\mathcal{A}}$ are assigned to the more reliable subchannels, $\mathcal{A} \subseteq \{1, 2, \ldots, N\}$. Frozen bits $u_{\mathcal{A}^c}$ are transmitted through the others, where $\mathcal{A}^c$ is the complement set of $\mathcal{A}$. Without loss of generality, the frozen bits can be set to zeros when the channels are symmetric.

Decoding: In [1], the successive cancellation (SC) algorithm is used to decode polar codes with low complexity $O(N \log N)$. Let the estimation vector of transmitted bits be $\hat{u}_1^N = (\hat{u}_1, \hat{u}_2, \cdots, \hat{u}_N)$. If bit i is frozen, then $\hat{u}_i = 0$. Otherwise, the decoding rule is as follows:

$$\hat{u}_i = \begin{cases} 0, & W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|u_i = 0) \geq W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|u_i = 1) \\ 1, & \text{otherwise} \end{cases} \qquad (2)$$

where $i \in \mathcal{A}$. A code tree of polar codes can be constructed by this successive bit determination procedure. A decoding path $d_1^i = (d_1, d_2, \ldots, d_i)$, $i \in \{1, 2, \ldots, N\}$ is composed of i branches in the tree. Along the path, the branch in the level-i denotes the corresponding bit taking the value of d_i. SC decoding can be seen as a greedy search algorithm in the code tree and in each level only the one of two branches with larger probability is selected for further processing. Once a decision error occurs, there is no chance to correct it in the later procedure.

Theoretically, the performance of ML decoding can be obtained by traversing all the N-length paths in the code tree. But the time complexity of this brute-force search is exponential. As a width-first search algorithm, SCL doubles the number of decision paths and selects the