Location Dependent Optimal Relay Selection in Mobile Cooperative Environment

Esam A. Obiedat*, Chirag Warty†, Lei Cao‡
* CommScope Inc., obiedat@gmail.com
† University of Illinois Chicago,cwarty2@uic.edu
‡ University of Mississippi, lcao@olemiss.edu

***Abstract*—This paper proposes an optimal relay selection criteria based on the location of the relay ($0 \leq \lambda \leq 1$) relative to source and destination in the cooperative coded system. The proposed optimization algorithm employs distributed turbo product coding technique with hard and soft decoding. It is shown that the link quality depends on the location of the relay which in turn affects overall system Bit Error Rate (BER) performance. The simulation model creates several scenarios for location of intermediate relays when the inter-user channel is experiencing distortion in presence of different Signal to Noise Ratio (SNR)s. It is observed that the link performance degrades as the relay proximity changes with respect to the source and the destination. The relay selection optimization algorithm provides the participating nodes necessary information to select neighboring nodes depending on link quality, thus lowering the BER and increasing the overall network capacity.**

***Index Terms*—Power Allocation, Distributed Coding, Turbo Decoding, Optimal Relay Selection, Location Optimization, Soft Decode And Forward, Cooperative Communication**

I. Introduction

To mitigate the fading and multipath propagation effects, time, frequency and spatial adversity techniques or a hybrid of them can be utilized. Among which the spatial diversity has been studied extensively in context of point to multipoint communication by using intermediate relays, increasing the system throughput and reliability [1]. In a Cooperative strategy when a node has information to transmit it cooperates with other nodes in the vicinity to transmit its information to the destination thus forming a virtual antenna array [1].

In the coding technique employed here the relay forwards an incremental redundancy for the recovered message. Turbo product codes have shown higher decoding performance with a very low decoding Complexity in high code rates, making them very favorable for wireless networks [2]. In the decode and forward scheme the relay decodes and re-encodes the received message before forwarding it to the destination. The two parts of the DTPC matrix are received over two or more different channels and they experience different SNRs. The second part of the code is received over lower SNR (the direct channel). However, if the relay makes decoding errors, then the part of the Turbo Product Codes (TPC) matrix received over the relay channel will have more effect on the direction of the decoding because Soft Input Soft Output (SISO) decoders grant the bits with higher SNR more confidence. The question that arises in dense cooperative network with distributed coding is the choice of neighboring node to be used as the relay in order to reduce propagation errors and to maximize link quality on source to relay and relay to destination channels.

Most of the existing reported distributed coding schemes are constructed based on fixed power allocations. However as shown in [2], [3], the power allocated for the transmission is proportional to the location of the relay relative to source and destination. The available optimization algorithms for cooperative network focus on power allocation. None of the existing algorithms has studied the location dependency for distributed coding systems. The location of the relay in the distributed coding system can be simply adjusted to yield level of performance required at the destination. In this paper, we investigate the optimum location that a node can occupy to obtain the required BER. Rather than assigning equal power to the source and the relay, as done in [4], [5]. We use the locations of the relay with respect to the source and destination to find the optimum distance of the relay that might result in the desired performance and link quality requirement at the destination. In this paper the concept of distributed encoding is applied for source transmitted messages over multiple relay nodes and use a modified iterative Turbo product decoding at the destination to decode the received distributed TPC over multiple channels.

The remainder of the paper is organized as follows: cooperative network model is presented in section II. The location optimization algorithm for the DTPC system is demonstrated in section III. The simulation results are presented in section IV and in section V the optimum results are discussed and conclusions are presented.

II. Cooperative Network Model

The cooperative technique can improve the overall system capacity by adjusting relay positions in the network compared to original non-cooperative system. The cooperative scheme is used in which the relay forwards an incremental redundancy to the destination about the recovered message from the transmission source. The destination uses the two parts of the code received via the direct path and the relay channel to conduct message decoding. This paper considers a single relay model, consisting of source 's', relay 'r' and destination 'd' as depicted in the Fig. 1. All three terminals are operating in a half duplex mode and any transmissions from source to destination requires tools time slots. The relay decodes

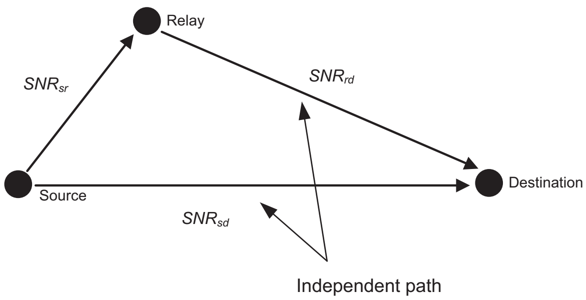

Fig. 1. Single Relay Model

the received message and encodes it before sending it to the destination.

This model is more practical in real systems, when the relay is usually located between the two terminals and the separation distances are relatively large. A model which returns the simulation problem from three-dimensional to two-dimensional problem has been used in many other works, e.g. [6]–[8]. The received signals at the relay and the destination during the two time slots for the line model can be generally expressed as follows:

$$\begin{aligned} y_d[2k-1] &= \sqrt{E_s}\alpha_{sd}[2k-1]x_s[2k-1] \\ &\quad +n_{sd}[2k-1] \qquad (1) \\ y_r[2k-1] &= \frac{\sqrt{E_s}}{(1-\lambda)^2}\alpha_{sr}[2k-1]x_s[2k-1] \\ &\quad +n_{sr}[2k-1] \qquad (2) \\ y_d[2k] &= \frac{\sqrt{E_r}}{\lambda^2}\alpha_{rd}[2k]x_r[2k]+n_{rd}[2k] \qquad (3) \end{aligned}$$

where y_j denotes the received signal at node j while x_i is the transmitted signal from node i and k is the time slot. The channel between the two nodes i and j has AWGN noise n_{ij}, and channel attenuation α_{ij}. E_s and E_r are the transmit energy/bit for the source and relay, respectively.

Using free space propagation on the line model and assuming *fixed transmission energy per bit*, the SNR values for the three channels, i.e. γ_{sd}, γ_{sr} and γ_{rd} for the direct, inter-user and relay channel respectively, are related by the following expressions [9]:

$$\gamma_{sr} = \frac{\gamma_{sd}}{(1-\lambda)^2} \qquad (4)$$

$$\gamma_{rd} = \frac{\gamma_{sd}}{\lambda^2} \qquad (5)$$

where $0 \leq \lambda \leq 1$ indicates the position of the relay with respect to the destination when the distance between the source and the destination is normalized to 1, with $\lambda = 0$ when the relay is located at the destination. Fig. 2 displays how the values of SNR at the destination and the relay change when the relay is moved across the source-destination line for a fixed γ_{sd}.

A. Turbo Product Codes

The source employs a simple component code for the input data using (n, k, δ) Extended Bose-Chaudhuri-Hochquenghem (EBCH) to encode the message signal. The EBCH encoder adds the overall parity check to the conventional Bose-Chaudhuri-Hochquenghem (BCH) codeword to expand the minimum hamming distance from δ to $\delta+1$ [3]. In the first time slot, source broadcasts the message containing a block of EBCH encoded codewords to relay and destination. The relay decodes the received message and generates a vertical parity. The vertical parity is produced by arranging the decoded codewords in rows and encoding them vertically with EBCH codes. Consider two EBCH systematic linear block codes (n_1, k_1, δ_1), and (n_2, k_2, δ_2), where n_i is code word length, k_i is input information block length and δ_i is the minimum hamming distance. A serial concatenation of the two linear block codes by transposing the encoder intermediate matrix a complete product code is generated, Fig. 3. Rows are encoded by $\mathcal{C}_1$ to produce horizontal parity (P_h) and columns of the matrix including the columns of P_h are then encoded by $\mathcal{C}_2$ to produce vertical parity (P_v). The primary advantage of BCH codes is the ease with which they can be decoded by applying syndrome decoding algebraic method. The product code matrix is produced by encoding k_2 rows by code $\mathcal{C}_1$ and n_2 columns by code $\mathcal{C}_2$ of the $k_2 \times k_1$ information matrix. The resultant product code matrix assumes the characteristics of $N = n_1 \times n_2$, $K = k_1 \times k_2$ and $\Delta = \delta_1 \times \delta_2$. Here we assume the two component codes $\mathcal{C}_1$ and $\mathcal{C}_2$ to be identical such that $n_1 = n_2 = n$, $k_1 = k_2 = k$ and $\delta_1 = \delta_2 = \delta$.

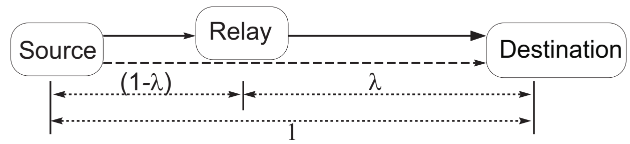

Fig. 2. The simplified three terminals line network topology

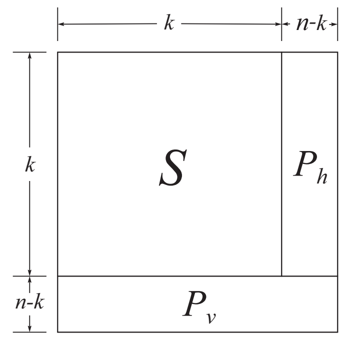

Fig. 3. The Structure of TPC matrix. S is systematic information, P_h and P_v are the horizontal and vertical parities.

B. Cooperative DTPC

To establish the distributed encoding for the TPC, the source broadcasts the $k_2 \times n_1$ matrix resulted from the first encoding stage by the $\mathcal{C}_1$ code to the destination and the neighboring relay nodes. One pre-selected relay corrects the received message and uses the second code $\mathcal{C}_2$ to encode the columns of the decoded bits to obtain the $n_2 \times n_1$ matrix of the complete TPC shown in Fig. 3. The relay then transmits the generated parity bits only ($(n_2 - k_2) \times n_1$) from the last encoding process to the distention. Depending on the relaying protocol used, the transmitted bits from the relay can be in two forms: hard bits with Decode and Forward (DF) or soft bits with Soft Decode and Forward (SDF) if the relay employs Soft Input Soft Output (SISO) decoding and encoding or not [2], [3]. After receiving the two parts of the transmitted data, the destination constructs a complete TPC by joining the two received parts. The soft decode and forward process is carried out in three steps: First, the relay soft-decodes the received sequences and generates the Log-Likelihood Ratio (LLR) output for the decision bits. Then the LLR values are used to infer the LLR values of the vertical parity bits. Finally, the soft output for the parity bits is obtained and forwarded to the destination.

Upon the receipt of sources transmission at the relay, Chase II decoding algorithm is used to decode the transpose of the received matrix ($\mathbf{Y}^T$) to get the Maximum-Likelihood (ML) decision $\mathbf{D}$ (matrix of dimension $n \times k$). The received matrix can be written in the form:

$$\mathbf{Y} = [\bar{\mathbf{y}}_1, \bar{\mathbf{y}}_2, \bar{\mathbf{y}}_3, \cdots, \bar{\mathbf{y}}_k]^T,$$

where $\bar{\mathbf{y}}_i = [y_i^1, y_i^2, y_i^3, \cdots, y_i^n]^T$,$y_i^j = x_i^j + z_i^j$, $1 \le i \le k$, $1 \le j \le n$, x_i^j is the transmitted Binary Phase Shift Keying (BPSK) symbol, $z_i^j \sim N(0, \sigma^2)$, σ^2 is the variance of the Additive White Gaussian Noise (AWGN) channel. The decoder's decision is:

$$\mathbf{D} = [\bar{\mathbf{d}}_1, \bar{\mathbf{d}}_2, \bar{\mathbf{d}}_3, \cdots, \bar{\mathbf{d}}_k],$$

and $\bar{\mathbf{d}}_i = [d_i^1, d_i^2, d_i^3, \cdots, d_i^n]^T$, with $d_i^j \in \{-1, +1\}$. Chase II algorithm searches for the decision codeword $\bar{\mathbf{d}}_i$ with the minimum Euclidean distance from the received vector $\bar{\mathbf{y}}_i$. After finding $\mathbf{D}$, the LLR of each element d_i^j is calculated using the Distance Based Decoding (DBD) algorithm [10]:

$$L(d_i^j) = d_i^j \ln\left(\frac{\phi_i + \exp(2y_i^j d_i^j / \sigma^2)}{1 - \phi_i}\right). \tag{6}$$

where ϕ_i is the confidence value which is defined as the probability that the decoder makes a correct decision given the received sequence $\bar{\mathbf{y}}_i$. Using the result found in [11] for the LLR of a parity bit for two statically independent random bits u_1 and u_2 given by:

$$\begin{aligned} L(u_1 \oplus u_2) &= \log \frac{1 + e^{L(u_1)} e^{L(u_2)}}{e^{L(u_1)} + e^{L(u_2)}} \\ &\approx \operatorname{sign}(L(u_1) \cdot L(u_2)) \cdot \min(|L(u_1)|, |L(u_2)|), \end{aligned} \tag{7}$$

The LLR for a parity bit e_i^j, $k+1 \le i \le n-1, 1 \le j \le n$ resulted from block encoding the matrix $\mathbf{D}$ can be generalized to [9]:

$$\begin{aligned} L(e_i^j) &= L\left(\bar{\mathbf{p}}_i^T \bar{\mathbf{d}}^j\right) \\ &= L\left(\sum_{l \in \mathcal{X}_i}^{\oplus} d_l^j\right), \mathcal{X}_i = \{l | p_i^l = 1\} \\ &\approx \operatorname{sign}\left(\prod_{l \in \mathcal{X}_i} L(d_l^j)\right) \cdot \min_{l \in \mathcal{X}_i} \left|L(d_l^j)\right|, \end{aligned} \tag{8}$$

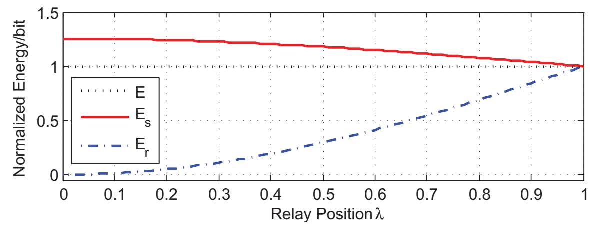

Fig. 4. Normalized energy/bit for the source and the relay using the proposed power allocation

where $\bar{\mathbf{d}}^j$ is the jth row in $\mathbf{D}$, $\mathcal{X}_i$ refers to the set of indices in which the vector $\bar{\mathbf{p}}_i$ has 1's. The block encoded matrix $\mathbf{E}$ is defined here as:

$$\begin{aligned} \mathbf{E} &= [\mathbf{D}\mathbf{I}_k | \mathbf{D}\mathbf{P} | \ \bar{\mathbf{e}}_n] \\ &= [\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2, \bar{\mathbf{e}}_3, \cdots, \bar{\mathbf{e}}_k, \bar{\mathbf{e}}_{k+1}, \cdots, \bar{\mathbf{e}}_{n-1}, \bar{\mathbf{e}}_n]. \end{aligned}$$

where $\mathbf{P} = [\bar{\mathbf{p}}_{k+1}, \bar{\mathbf{p}}_{k+2}, \cdots, \bar{\mathbf{p}}_{n-1}]$ is the parity generator matrix, $\mathbf{I}_k$ is the identity matrix and $\bar{\mathbf{e}}_\mathbf{n}$ contains the overall parity bits. In the final step, the soft information for all parity bits e_i^j, $k+1 \le i \le n-1, 1 \le j \le n$, is calculated as $\frac{2}{\sigma^2} L(e_i^j)$ and forwarded to the destination.

C. Power Allocation

In this paper we use the power allocation method between the source and the destination from our previous work [12]. It is found that the error propagation is caused by the fact that the two parts of the code have different SNRs. Therefore, by simply assigning the power such that the two received parts of the code at the destination have equal SNRs, all the bits of the code will be received with equal SNR and therefore will be processed by the decoder with equal trustiness. Therefore, the power allocation method that we use in this paper is based on the condition that the received signal to noise ratio for all the parts of the code are equal, i.e. in terms of the SNR values:

$$\gamma_{sd} = \gamma_{rd}. \tag{9}$$

Using the free space propagation model, and assuming that the relay and the destination are separated by a fraction λ of the source-destination distance, and using (9), the energy per bit at the source and the relay can be derived from E which is the energy/bit for the non-cooperative case as [12]:

$$E_s = \frac{n^2}{kn + \lambda^2(n-k)n} E \tag{10}$$

$$E_r = \frac{n^2}{\frac{kn}{\lambda^2} + (n-k)n} E \tag{11}$$

Fig. 4 shows how the source and relay energy/bit levels changes as the relay-destination separation approaches the distance between the source and the destination.

In the case of line model, the relations in (4) and (5) become:

$$\gamma_{sr} = \frac{\gamma_{sd}}{(1-\lambda)^2} \tag{12}$$

$$\gamma_{rd} = \gamma_{sd} \tag{13}$$

However, all the SNR values in the two previous equations now depends on the position of the relay λ, even for γ_{sd}, unlike the fixed power allocation.

III. Location Optimization

A. Optimization Algorithm Requirements

In this section we apply optimization rules on the power optimized Distributed Turbo Product Code (DTPC) cooperative system using the DF and SDF relaying protocols presented in [2] to search for the maximum attainable performance using the line experimental setup. Our main target in this paper is to find the optimal relay location which minimizes the final BER at the destination node. The performance of the cooperative systems, can be improved by relaying with optimal power allocations as found on [12] for the DTPC system. Therefore, we assume that a maximal overall transmit power from the source and the relay is fixed and is equal to the same power required to transmit the complete TPC codeword from the source to the destination in the non-cooperative scenario. Then, the overall total transmitting power is to be optimally shared between the source and the relay, so that the power is efficiently utilized to gain the maximum performance possible for the DTPC. Basically, we optimize the power allocation by minimizing bit-error probability at the destination. Optimal power allocation will not only give better performance but also saves energy for the relay node which in many situations will be battery operated which makes the power a scarce resource, not like the source or the destination, a typical example wireless sensor networks.

Since the main target for power optimization is to reduce the probability of error at the destination, the target function for our optimization problem is therefore the BER after the decoder. However, there is no exact expression available to model the probability of error after the decoder, but one way to characterize this unknown function is by the empirical function given by:

$$BER \quad = \quad f(\alpha, P) \tag{14}$$

where the parameter α is used to represent the location of the relay that we want to optimize, P is the total transmit power. This relation is monotonically decreasing function with respect to the power P, so to find the optimal location where the relay will help maximizing the BER performance we have to find the values of α that results in the minimum BER. Thus, the optimization problem is reduced to a one dimensional problem with only one variable parameter α.

B. Optimization Algorithm

Simulation errors could lead the optimization algorithm to a wrong solution. This limitation is solved using the sliding ball principle on a slope as in Fig. 5. If a ball is dropped from any peak of the slope it will slid and will exceed the lowest point on the curve and then traverses more distance upward beyond the solution until it stops and reverses its direction of movement. This continues until the ball reaches steady state at the lowest point on the slope. The numbers on the balls in Fig. 5 indicates the positions of the ball when it reverses the sliding direction, where the number 1 indicates the starting point. Note that if a sliding ball runs over a small bulge, it will pass this bulge and will continue sliding until it reaches an uphill. The required optimization algorithm should be designed to minimize the run time and the complexity of the algorithm. We set the optimization algorithm to work on bit error rate level close to 10^{-3} bit errors/frame to have accurate results with lower number of repetitions (the number of transmitted and received frames for a single SNR and α pair).

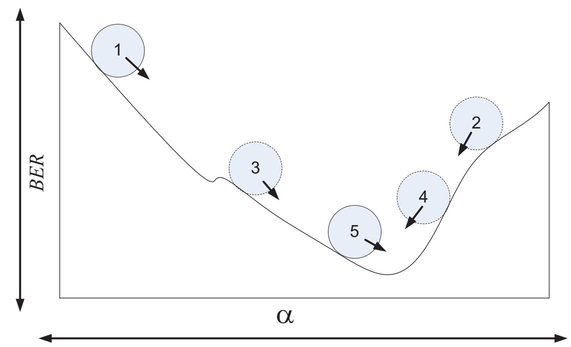

Fig. 5. The principle of sliding ball used in designing the optimization algorithm

The algorithm calculates the step size based on the length search segment and the number of steps. For each step, the algorithm compares the current bit error rate of the decoding result with the previous step. If the BER at current step is smaller than it at the previous step, the algorithm continues to the next simulation step. If the current decoding error rate is larger than rate at the last step, then the algorithm compares the previous step result with the result two steps back: if BER one step back is also larger than the BER at the two steps back, then it sets new boundaries (search segment) and step size. Otherwise, if the result one step back is smaller than the result two steps back, then it continues to the next simulation step.

The optimization algorithm continues on steps until it reaches two consecutive points on the upward direction of the curve (i.e. last two results of BER are larger then previous step) or until it reaches the boundary of the curve segment. In the two cases, a new search segment is determined from the length covered by the two steps before the current step. The step size is calculated from the length of the search segment and the required number of steps. The optimization algorithm used to find the value of α for each signal to noise ratio value γ_{sd} across the line model is shown in algorithm 1.

We used the accuracy threshold Th as stopping criterion to determine when the algorithm has approached to the solution with a predetermined accuracy level. The number of steps $NSteps$ is the number of of sections that the search segment is divided to. As noted from the algorithm, the step size reduces every time when a new search segment is found. The new search segment is determined to be the the last two sections coming before the current segment at which the condition to find new search segment is satisfied.

Input: DTPC simulator with inputs α, and SNR and output BER
for $SNR : 0$ **to** 2 **do**
  Set search segment boundaries: α_{end} and α_{start},
  Set accuracy threshold Th,
  Set number of steps to $NSteps$,
  Calculate step size $Step = (\alpha_{start} - \alpha_{end})/NSteps$,
  while $\alpha_{start} - \alpha_{end} > Th$ **do**
    Set $\alpha = \alpha_{start}$;
    for $j = 1$ **to** $NSteps$ **do**
      repeat
        Simulation inputs: α and SNR;
      until *Maximum Number of Frames*;
      The output is $BER[j]$;
      if $j > 2$ **then**
        if $BER[j] > BER[j-1] > BER[j-2]$ **then**
          /* Going Uphill */
          Set $\alpha_{start} = \alpha + Step$,
          Set $\alpha_{end} = \alpha + 3 * Step$,
          Set $Step = (\alpha_{start} - \alpha_{end})/NSteps$;
          Break;
        end
      end
      if $j < NSteps$ **then**
        /* Going Downhill */
        Set $\alpha = \alpha - Step$
      end
      if $j = NSteps$ **then**
        /* Reached the end */
        Set $\alpha_{start} = \alpha$,
        Set $\alpha_{end} = \alpha + 2 * Step$,
        Set $Step = (\alpha_{start} - \alpha_{end})/NSteps$;
      end
    end
  end
end
Output: Location α where minimum BER is obtained

Algorithm 1: Location optimization algorithm

IV. Performance Simulations

All the EBCH encoded n-bits codewords from the source and the relay are BPSK modulated and sent to the destination. All the three channels are considered to be orthogonal and have AWGN and the transmitted signals are considered to decay according to free space propagation model, where the path loss exponent is 2. The two component codes used in the DTPC simulations have the same parameters, where $n = 64$, $k = 51$ and $\delta = 6$. The TPC decoding at the destination is based on the DBD SISO decoder [10], where channel statistics are assumed to be available for the decoding process.

In Fig. 6, the optimization simulation for a DF cooperative coding is run for each step of the SNR of the source to destination. After about 30 iterations, the optimal location for the relay which results in the highest BER performance is recorded when the step size drops below 1×10^{-5}. For the DF cooperative scheme it is found that the optimal location for the relay for the highest performance is near 0.45 of the distance between the source and the destination. When the SNR for the source to destination signal is lower than 0.4dB, the optimal performance is found to be when the relay is closer to the source between 0.35 and 0.45 of the normalized source-destination distance. As the source-destination SNR get higher beyond 1.4dB the optimal location for the highest performance tend to gradually be closer to the destination from 0.45 to 0.55 of the normalized distance.

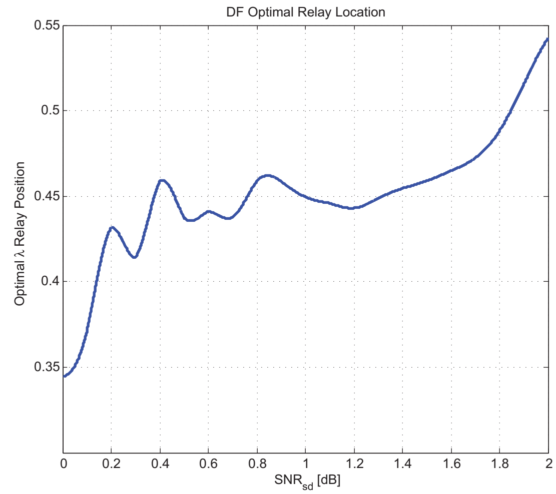

Fig. 6. Decode and forward optimal relay location

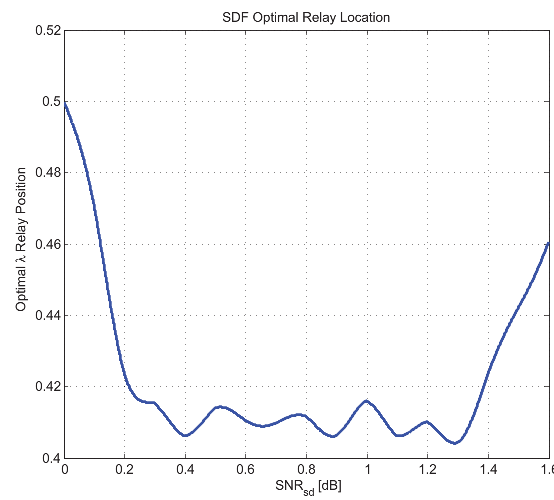

Fig. 7. Optimal relay location selection using Soft DF processing scheme

The Fig. 7 shows the results for optimizing the location of the relay on the line between the source and the destination in the cooperative coding with SDF scenario implemented at the relay. The SNR for the message transmitted directly from the source to the destination is changed each time and for each step the optimal relay location is recorded after running the optimization method aforementioned that will result on the lowest BER. Each time the optimization is run, it will stop when the step size becomes less than 1×10^{-5}. The results

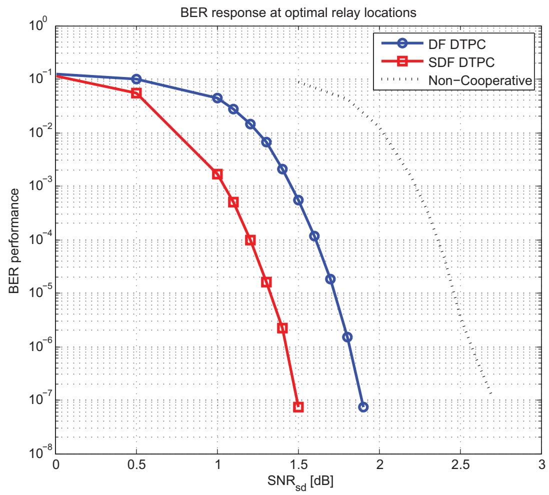

Fig. 8. BER performance of the DTPC system with DF and SDF relaying

for the SDF case show that the optimal location for a wide range of SNR_{sd} between 0.2 and 1.3dB is about 0.41 of the source-destination distance, i.e. closer to the source. In Fig. 7 when the same SNR drops below 0.2dB or becomes higher than 1.3dB the relay prefers a location closer to the midway between the source and destination to contribute the highest BER performance possible.

This Fig. 8 shows the optimal BER performance for both SDF and DF cooperative relaying schemes. Here BER performance is plotted when the relay is located at the optimal relay position. The location where the optimal performance is obtained from the optimization simulation discussed above for both SDF and DF methods and then the corresponding highest BER performance resulted at the optimal location is recorded and then plotted against the signal to noise ratio of the source to destination link. The result depicted because is the highest possible BER performance since each point is the result of an optimization process that looks for the lowest BER and it is done based on the power optimization discussed in previous works [12].

V. Conclusions

In this paper we propose an optimal relay selection optimization algorithm depending on the location of the relay. It can be observed from the simulation results that to have a superior link performance the selected relays should be located at $(0.35 \leq \lambda < 0.55)$. The paper points out the possibility for a relay to relocate and position itself to improve the link quality. After comparing the results obtained by applying the algorithm to general case large gains in link performance (BER) can be seen. This, however, may change with the variation in coding technique and forwarding scheme applied the relay. . The results obtained from applying the proposed location optimization method show large gains in BER performance and showed effectiveness in allocating power between the transmitting nodes. Relay selection optimization for a distributed, decentralized network is critical. In applications such as routing in military ad hoc mobile networks, optimal relay selection can be vital element in providing the desired level of quality of service.

References

[1] A. Sadek, W. Su, and K. Liu, "Multinode Cooperative Communication In Wireless Networks," *IEEE transactions on signal processing*.

[2] E. Obiedat, W. Xiang, J. Leis, and Cao L., "Soft Incremental Redundancy for Distributed Turbo Product Codes," in *7th Annual IEEE Consumer Communications and Networking Conference*. IEEE, 2010.

[3] E. Obiedat, G. Chen, and L. Cao, "Distributed Turbo Product Codes Over Multiple Relays," in *7th Annual IEEE Consumer Communications and Networking Conference*. IEEE, 2010.

[4] B. Zhao and M. C. Valenti, "Distributed turbo coded diversity for relay channel," *Electronics Letters*, vol. 39, no. 10, pp. 786–787, 2003.

[5] C.T.K. Ng and A.J. Goldsmith, "Capacity and power allocation for transmitter and receiver cooperation in fading channels," in *IEEE International Conference on Communications, 2006. ICC '06*, June 2006, vol. 8, pp. 3741–3746.

[6] L. Hanzo, T.H. Liew, and B.L. Yeap, *Turbo coding, turbo equalisation and space-time coding for transmission over fading channels*, Wiley-IEEE Press, 2002.

[7] A. Chakrabarti, A. Baynast, A. Sabharwal, and Behnaam Aazhang, "Low density parity check codes for the relay channel," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 2, pp. 280–291, 2007.

[8] Zhixin Liu, V. Stankovic, and Zixiang Xiong, "Wyner-Ziv coding for the half-duplex relay channel," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, 2005, vol. 5, pp. v/1113–v/1116 Vol. 5.

[9] E. Obiedat and L. Cao, "Soft Information Relaying for Distributed Turbo Product Codes (SIR-DTPC)," *IEEE Signal Processing Letters*, vol. 17, pp. 363–366, 2010.

[10] N. Le, A.R. Soleymani, and Y.R. Shayan, "Distance-based-decoding of block turbo codes," *Communications Letters, IEEE*, vol. 9, no. 11, pp. 1006–1008, Nov. 2005.

[11] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *Information Theory, IEEE Transactions on*, vol. 42, no. 2, pp. 429–445, Mar 1996.

[12] E.A. Obiedat and Lei Cao, "Power allocation for distributed turbo product codes (dtpc)," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, dec. 2010, pp. 1 –6.