List Detection for Multi-Access Channels

Aaron B. Reid, Alex J. Grant
Institute for Telecommunications Research
University of South Australia
Mawson Lakes SA 5095 Australia
e-mail: areid@spri.levels.unisa.edu.au

Paul D. Alexander
Southern Poro Communications
6A Nelson Street
Annandale NSW 2038 Australia

Abstract— **We propose a low complexity iterative multiuser decoder. We perform the multiuser a-posteriori probability calculation based on the marginalization of probabilities over a subset of the P highest probable sequences. Given a K user symmetric channel this list may be approximated closely with per-bit computational complexity $O(K+P+2K\log K)$. We further show that for any multiuser system possessing a polynomial complexity optimal detection algorithm it is possible to obtain the P highest probable sequences with polynomial complexity. We further show that for any multiuser channel it is possible to obtain P highly probable sequences in polynomial time and hence utilize the list detection procedure for iterative decoding.**

I. Introduction

A code-division multiple-access (CDMA) system with channel coding may be viewed as a serially-concatenated system and many iterative decoders have been proposed for these systems [1], [2], [3]. The main contribution of this paper is an algorithm that approximates *a-posteriori* probability (APP) calculation with low complexity for symmetric CDMA channels (in which all the cross-correlations are identical). This algorithm reduces the required complexity by finding a list of P sequences with high *a-posteriori* probability and marginalizing only over this list. Numerical investigations have shown that typically the size of the list required is very small compared to the total number of sequences. For the symmetric channel, we give a polynomial complexity algorithm for finding this list. We additionally show how this idea may be applied to any system for which there exists a polynomial complexity optimal detection algorithm and also the case of any multiuser channel.

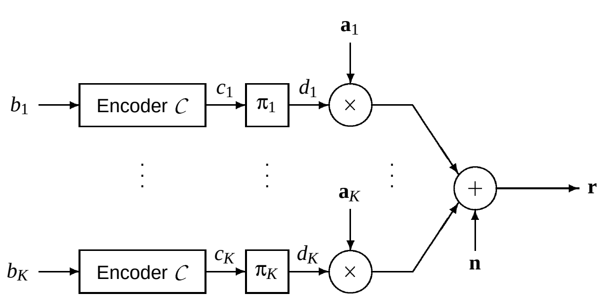

Fig. 1. Coded CDMA Channel Model

With reference to Fig. 1, each user $k=1,2,\ldots,K$ encodes their binary information sequence b_k using a rate R code $\mathcal{C}$. Each user independently permutes their encoded sequence c_k with an interleaver π_k. We denote the binary sequence output from the interleaver of user k as $d_k[i]$, where i is the symbol time. Transmission is assumed to be symbol-synchronous. The received signal in the i-th symbol duration is given by $\mathbf{r}[i]=\mathbf{A}\mathbf{d}[i]+\mathbf{n}[i]$, where $\mathbf{A}$ is an $N\times K$ real or complex matrix whose unit energy columns are the discrete signature signals $\mathbf{a}_k$ of the K users, $\mathbf{d}[i]$ is a length K column vector with elements $d_j[i]\in\{-1,+1\}$ (for binary phase-shift keying) being the transmitted binary symbol for user j, and $\mathbf{n}[i]$ is a sampled circularly symmetric complex noise vector with covariance matrix $E[\mathbf{n}\mathbf{n}^*]=\sigma^2\mathbf{I}_K$.

This work was supported by Southern-Poro Communications and the Australian Government under ARC SPIRT C00002232.

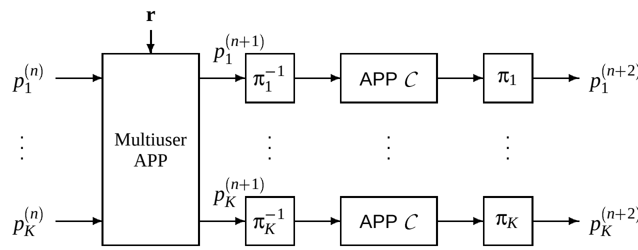

Fig. 2. Iterative Receiver Structure

Fig. 2 shows an iterative decoder for this system. The multiuser APP operates on a per-bit basis (ignoring constraints imposed by the channel code $\mathcal{C}$) and we therefore drop the bit index. Consider half-iteration n. Given the received vector $\mathbf{r}$ and prior probabilities $p_k^{(n)}(d)$ on each user's coded bits the multiuser APP produces extrinsic probabilities for each user according to

$$p_k^{(n+1)}(d)=\frac{1}{p(\mathbf{r})p_k^{(n)}(d)}\sum_{\mathbf{d}:d_k=d}p(\mathbf{r}\mid\mathbf{d})\prod_{j=1}^{K}p_j^{(n)}(d_j) \qquad (1)$$

where the product term is the contribution of the a-priori information received from the previous half-iteration (at the commencement of the first iteration the priors are uniformly initialized) and $p(\mathbf{r}\mid\mathbf{d})$ is the probability of receiving $\mathbf{r}$ given $\mathbf{d}$ (specified by the channel model). For the second half of the iteration, user k performs single-user symbol-wise APP decoding of $\mathcal{C}$ using $p_k^{(n+1)}$ as priors, and outputs extrinsic probabilities $p_k^{(n+2)}(d_k)$ which serve as code-symbol priors for the next half-iteration.

The number of vectors $\mathbf{d}$ with $d_i=d$ is 2^{K-1} and hence the marginalization (1) is, in general, intractable for large K. We will approximate (1) using the key empirical observation that the vast majority of the $\mathbf{d}$ sequences contribute a negligible amount to the total probability when performing the marginalization of each coded bit.

II. Approximate Joint APP

We propose to decrease the complexity of the calculation of (1) by summing over a high-probability subset of the sequences $\mathbf{d}$. Suppose we can find a *prior list* of the P sequences $\mathbf{d}$ with the largest prior probabilities, $\prod_{k=1}^{K} p_k(d_k)$ (dropping the iteration index) and another *channel list* of the P sequences $\mathbf{d}$ with the highest channel probabilities, $p(\mathbf{r}|\mathbf{d})$. Once both lists have been found, we may then merge them into a single list of size P by taking the P highest probable sequences from either list. Note that this procedure does not guarantee that we obtain even the most probable sequence according to $p(\mathbf{r},\mathbf{d})$. The reasoning for the merging of lists is to obtain a more reliable list after each iteration for the marginalization. The actual merging process is not unique - we could have easily made the overall list to be the union of all sequences in both the *channel list* and *prior list* but through simulations we have found that there is little to no gain in more elaborate merging methods.

Clearly, the *channel list* needs to be calculated only once. The *prior list* must be updated every iteration. For the first iteration we calculate the overall list based purely on the *channel list*, since the priors give no additional information at this stage.

The *prior list* may be calculated very easily. Create a graph with $(K+1)$ nodes and place parallel directed edges from the first node to the second and so forth to the node $(K+1)$. Label the edges from node k to node $k+1$ with $-\log p_k(+1)$ and $-\log p_k(-1), k=1,\ldots,K$. With this representation we may find the P most probable sequences using the P Shortest Paths Algorithm [4] which has complexity $O(K+P+(K+1)\log K)$.

Finding the *channel list* is in general NP-hard [5] (note it induces the optimal detection problem). We now describe a special case namely the K-symmetric channel for which we give a polynomially complex algorithm and a general procedure which gives the P highest probability sequences when optimum detection can be performed in polynomial time. We also give a procedure for obtaining P highly probable sequences in polynomial time for any multiuser channel.

III. P Most Probable Sequences

Consider the cross-correlation matrix $\mathbf{R}=\mathbf{A}^t\mathbf{A}$ such that

$$R_{ij} = \begin{cases} 1 & i=j, \\ \rho & i\neq j, \end{cases} \tag{2}$$

for $1\leq i,j\leq K$. This is what we mean by the symmetric channel. Such $\mathbf{R}$ occurs with use of different shifts of m-sequences by the users. It also occurs in certain multi-beam narrowband satellite models [6].

Fig. 3 shows the cumulative distribution of the probabilities of all sequences for the cases of $K=10, \rho=0.6, E_s/N_o=1$dB, 6dB and $K=30, \rho=0.5, E_s/N_o=1$dB, 6dB, where E_s/N_o is the ratio of the energy per coded symbol to noise spectral density. We see that the number of sequences which contribute to the overall total probability is only a small fraction of the total number of

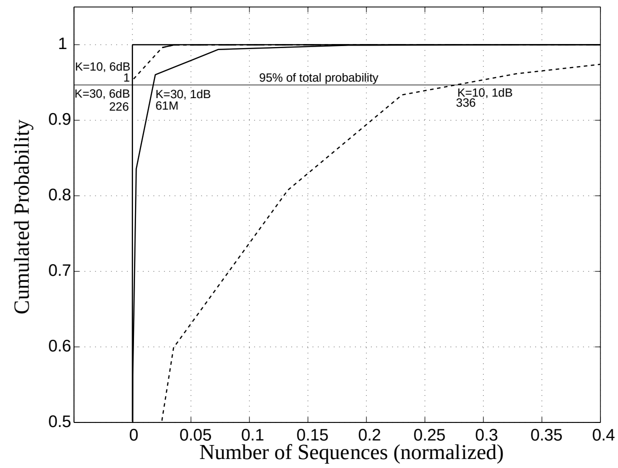

Fig. 3. Probability distribution

sequences (the number of sequence probabilities which cumulatively sum to 95% are shown). We also note that as the amount of interference (eg. noise or multiaccess interference, ρ) in the system is increased the number of sequences to obtain a high probability set increases proportionally.

A polynomial complexity optimal detection algorithm was presented in [7], [8] for $\mathbf{R}$ of the form (2). The optimal detection problem for this channel is

$$\hat{\mathbf{d}} = \arg \min_{\mathbf{d}\in\{-1,+1\}^K} \rho \sum_{i=1}^{K}\sum_{j=1}^{K} d_i d_j - 2\sum_{i=1}^{K} d_i y_i$$

where $\mathbf{y}=\mathbf{A}^t\mathbf{r}$ is the matched filter output. The polynomial complexity algorithm relies upon the fact that the first term on the RHS depends only upon the number of negative elements in $\mathbf{d}$, denoted $n(\mathbf{d})$.

Part of the development in [7], [8] shows that the most probable $\mathbf{d}$ conditioned upon $n(\mathbf{d})=0,1,\ldots,K$ can be found easily (optimal detection simply involves picking the best of these). For fixed $n(\mathbf{d})$ the most probable $\mathbf{d}$ is found as follows. Let $\mathbf{y}_\pi$ be $\mathbf{y}$ ordered such that the elements are non-decreasing. Let $\mathbf{d}_\pi$ be $\mathbf{d}$ in the same order. The optimum $\mathbf{d}_\pi$ is obtained by making the first $n(\mathbf{d})$ elements in $\mathbf{d}_\pi$ negative.

The key to finding the *channel list* of size P is that beginning with the optimal $\mathbf{d}_\pi$ with $n=n(\mathbf{d})$ negatives, we may systematically obtain in order of decreasing probability, every other $\mathbf{d}_\pi$ with $n(\mathbf{d})$ by a series of swaps between the negative and positive elements in $\mathbf{d}_\pi$. This is Algorithm 1 in the Appendix. We can show that this process of swapping has complexity $O(n(K-n)+P\log P)$. This proof is based on the fact that for each new sequence found, there are at most two candidate sequences created, and sequences are selected only by choosing among these candidates.

Given that we can determine in order the sequences of highest probability with fixed $n(\mathbf{d})$, we now have the problem of determining the overall highest probability $\mathbf{d}_\pi$. This problem is easily solved by growing $K+1$ subtrees from a root node corresponding to $n(\mathbf{d})=0,1,\ldots,K$. Each subtree is grown according to Algorithm 1 (with $n(\mathbf{d})$-specific variables local to each sub-tree), with the modification that we only extend the best node from the entire tree at each stage.

A general procedure is given in [9] which shows that if the number of computations to find an optimal solution to an integer optimization programming problem with n binary variables is $O(c(n))$, then the amount of computation required to obtain the P best solutions is at most $O(Pnc(n))$. As an example, if we have a multiuser channel of the type in [10], [11] where all off-diagonal elements in the cross-correlation matrix $\mathbf{R}$ are negative, the optimum multiuser detection may be performed in $O(K^3)$. Hence, we may form the *channel list* by finding the P most probable sequences according to the $p(\mathbf{r}|\mathbf{d})$ in $O(P\dot{K}^4)$ (a method for finding the P best sequences for this type of cross-correlation matrix is outlined in [12] which can be easily adapted to the multiple-access system by use of [10], [11]). We may then utilize the principle of merging the *prior list* and the *channel list* into one overall list to perform the marginalization (1). In general, we can use the procedure in [9] to obtain a method to find the P most likely sequences in polynomial time for any system in which polynomial complexity optimal detection is possible. We note that to the best of our knowledge, Algorithm 1 is new and is not based on the procedure of [9].

IV. P Highly Probable Sequences

In the absence of a polynomial complexity optimum detection algorithm we may still obtain a *channel list* of P highly probable sequences in polynomial time. This method does not however guarantee that even the most probable sequence is found with respect to $p(\mathbf{r} \mid \mathbf{d})$.

The metric to be used for calculating the probability of a received sequence over the channel is based on a noise whitened version of the matched filter output [13]. Let $\mathbf{W}$ be an upper triangular matrix obtained by the cholesky decomposition such that $\mathbf{R} = \mathbf{W}^t\mathbf{W}$, we may write the detection problem as,

$$\begin{aligned}\hat{\mathbf{d}} &= \arg \min_{\mathbf{d}\in\{-1,+1\}^K} \|\mathbf{r} - \mathbf{A}\mathbf{d}\|_2^2 \\ &= \arg \min_{\mathbf{d}\in\{-1,+1\}^K} \|\mathbf{W}(\xi - \mathbf{d})\|_2^2 + c \\ &= \arg \min_{\mathbf{d}\in\{-1,+1\}^K} \sum_{i=1}^{K} \Bigg(W_{ii}(\xi_i - d_i) \\ &\qquad + \sum_{j=i+1}^{K} W_{ij}(\xi_j - d_j) \Bigg)^2 + c. \qquad (3)\end{aligned}$$

where $\xi = \mathbf{R}^+\mathbf{y}$, $(\cdot)^+$ denotes the pseudo-inverse operation, and $c = \mathbf{r}^t\mathbf{r} - \mathbf{y}^t\mathbf{R}^+\mathbf{y}$ is a constant. $\mathbf{W}^{+t}$ is a noise whitening operation performed on the matched filter output and hence the whitened observation is now defined as $\mathbf{W}^{+t}\mathbf{y}$ and the received point is now given by $\mathbf{W}^{+t}\mathbf{R}\mathbf{d}$. A method for performing the cholesky decomposition for a singular positive definite matrix is outlined in [14].

From (3) we may write the partial metric accounting for users i to K in the recursive form,

$$m_i = m_{i+1} + \left(W_{ii}(\xi_i - d_i) + \sum_{j=i+1}^{K} W_{ij}(\xi_j - d_j) \right)^2 \qquad (4)$$

where $m_{K+1} = c$, and we note that m_1 is the total metric accounting for all users.

In the form (4) the ith partial metric relies only on bits d_k where $k = i, i+1, \ldots, K$. We may now implement the m-algorithm [15] to obtain P highly probable sequences which will form the *channel list*. The performance of the m-algorithm relies on the number of paths it retains at each stage of the corresponding tree structure. For our simulations we have limited the maximum number of paths retained at each level in the tree to be P, which gives a complexity of $O(PK)$. The m-algorithm operates on a tree in reverse user order. A root node is created with metric $m_{K+1} = c$ and is grown with the possible transmitted bits for user K and a partial metric is calculated by (4). We continue to grow the tree from each of the P leaves with each user in the order $K-1, K-2, \ldots, 1$. At each extension only the P leaves possessing the P least partial metrics are retained (the rest are discarded).

V. Performance Results

We now present simulated performance results for the proposed systems. The codes used were the maximal free distance rate 1/2, 4 state convolutional codes. We use information sequences of length 100, resulting in an interleaver size of 200.

Fig. 4 shows the performance of $K = 10$ users across the K-symmetric channel using Algorithm 1 with $\rho = 0.6$. We observe that after 4 iterations single user performance is almost achieved for all users using the proposed receiver with a list size of only $P = 80$ (compared to the full marginalization, which would require $P = 1024$).

Fig. 5 shows the performance of the receiver versus the list size P for $\rho = 0.5$ and $\rho = 0.6$ at $E_b/N_0 = 5dB$ ($\mathbf{R}$ of the form (2)). We see that the performance of the full-complexity ($P = 1024$) system is equal to the performance when $P = 80$ for $\rho = 0.5$ and $P = 90$ for $\rho = 0.6$.

Fig. 6 shows results for $K = 30$ and $\rho = 0.5$ ($\mathbf{R}$ of the form (2)). There we see that if we take only $P = 3000$ out of the possible 2^{30} sequences, we obtain performance within 0.5 dB from single user performance at $E_b/N_0 = 6$ dB after 4 iter-

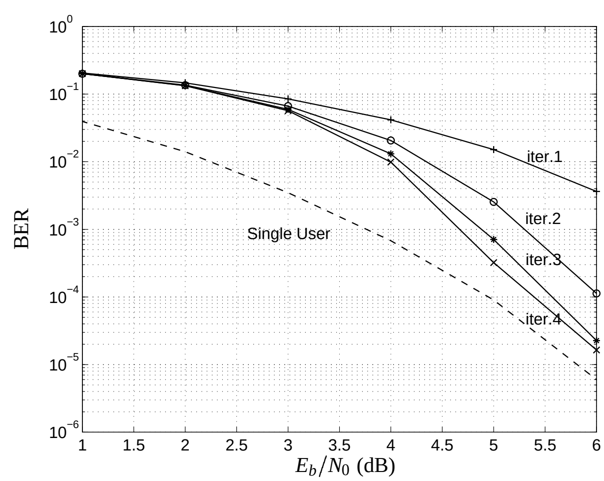

Fig. 4. K-symm. ch. (using Alg. 1) $K = 10, \rho = 0.6, P = 80$

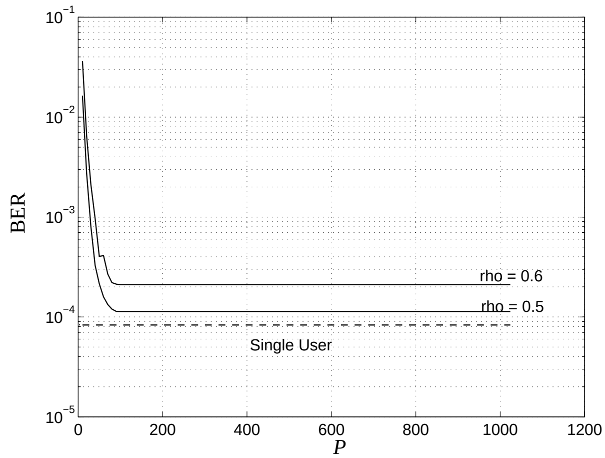

Fig. 5. K-symm. ch. (using Alg. 1), $K = 10, E_b/N_0 = 5$ dB

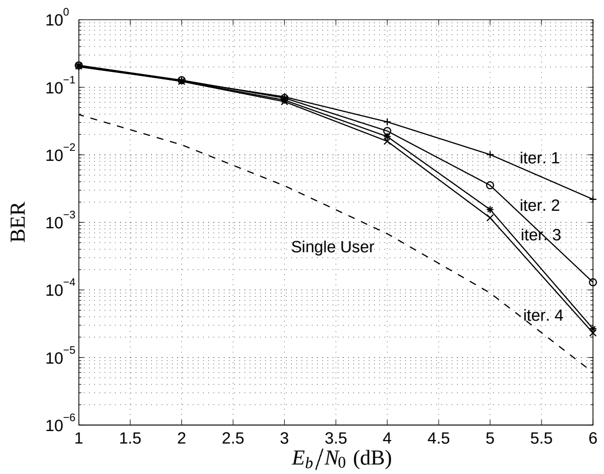

Fig. 6. K-symm. ch. (using Alg. 1), $K = 30, \rho = 0.5, P = 3000$

ations. This (and other similar numerical results that we have obtained) indicates that the required P does not need to increase exponentially with K in order to obtain near single-user performance. If we decrease the value of ρ (i.e decrease the amount of multiuser interference), the required list size P also decreases dramatically and single user performance is achieved at lower values of E_b/N_0.

Fig. 7 shows results where the $N \times K$ spreading matrix $\mathbf{A}$ is randomly generated at every time interval. We form the *channel list* by taking P highly probable sequences by use of the m-algorithm as described in Section IV. The number of users in the simulation of Fig. 7 is $K = 30$, spreading factor $N = 30$ and list size $P = 1500$. Despite the fact that the *channel list* is not as accurate in this case compared to the case of the K-symmetric channel, we can reduce the overall list size P significantly while observing that the system approaches single user performance at a lower value of E_b/N_0. This is due to the reduction in average MAI.

VI. Discussion and Conclusion

We have proposed a low complexity multiuser iterative decoder for systems where it is possible to perform polynomial

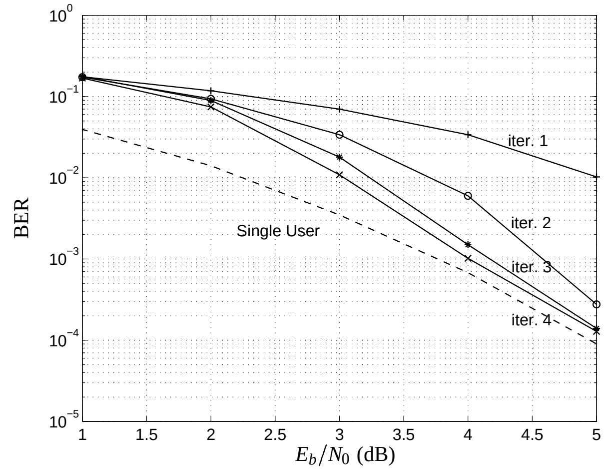

Fig. 7. Random channel (using m-alg), $K = 30, N = 30, P = 1500$

complexity optimum multiuser detection on the channel code and also when it is not possible. When we have a system with the channel described by (2), it was shown that the marginalization of probabilities in (1) may be approximated closely with per-bit computational complexity $O(K + P + 2K \log K)$. An integral part of the good performance of the system is due to Algorithm 1 which gives the P highest probable sequences when we have a symmteric channel as in (2).

Also proposed is a low complexity multiuser iterative decoder for the case of any multiuser channel (i.e. $\mathbf{R}$ is any symmetric cross-correlation matrix). In this case (1) may be approximated closely with per-bit computational complexity $O(K + (K+1)(P + \log K))$.

VII. Acknowledgment

Thanks to C. Schlegel who contributed to early discussion of this work.

VIII. Appendix

For convenience, the algorithm shall operate using the metric

$$M(\mathbf{d}) = \mathbf{r}^*\mathbf{r} + \mathbf{d}^t\mathbf{R}\mathbf{d} - 2\mathbf{d}^t\mathbf{y} \tag{5}$$

rather than probabilities (which may be easily calculated based on the metrics later). Let $\mathbf{d}_{\text{opt}}(n)$ be the best sequence with $n = n(\mathbf{d})$ negatives (easily found using the algorithm in [7], [8]). Calculate the cost (change to (5)) of exchanging the signs of each pair of non-equal bits d_i, d_j in $\mathbf{d}_{\text{opt}}(n)$. Note that because we retain $n(\mathbf{d})$ and we have channel symmetry the cost is simply $m_{ij} = 4d_{\pi(i)}y_{\pi(i)} + 4d_{\pi(j)}y_{\pi(j)}$ where $1 \leq i \leq n$ and $n+1 \leq j \leq K$. Consecutively number these exchanges (swaps) in non-decreasing order to get an array $\Delta = [\delta(1), \delta(2), \ldots, \delta(n(K-n))]$, where the i-th best swap $\delta(i)$ has members $\delta.m$ (cost of the swap) and $\delta.\mathsf{swapbits}$, a pair of numbers identifying which bits were swapped.

The algorithm will operate on a tree. Each node n of the tree (in addition to the child-parent relationships) has the following members, $n.\mathbf{d}_\pi$, the permuted $\mathbf{d}$ sequence; $n.\mathsf{metric}$ according to (5); $n.\mathsf{swapbits}$, a list of bits which have been swapped from $\mathbf{d}_{\text{opt}}(n)$ to obtain $n.\mathbf{d}_\pi$; $n.\mathsf{largestswap}$, where $\delta(n.\mathsf{largestswap})$ is the most costly swap which has been performed to reach n; and n.state $\in \{$ACTIVE, POTENTIAL$\}$.

REFERENCES

[1] P. D. Alexander, A. J. Grant, and M. C. Reed, "Iterative detection in Code-Division Multiple-Access with error control coding," *European Trans. Telecomm.*, vol. 9, no. 5, pp. 419–425, Sep./Oct. 1998.

[2] M. C. Reed, C. B. Schlegel, P. D. Alexander, and J. A. Asenstorfer, ," *IEEE Trans. Inform. Theory*, vol. 46, no. 12, pp. 1693–1699, Dec. 1998.

[3] H. El Gamal and E. Geraniotis, "Iterative multiuser detection for coded CDMA signals in AWGN and fading channels," *IEEE Sel. Areas in Comm.*, vol. 18, no. 1, pp. 30–41, Jan. 2000.

[4] D. Eppstein, "Finding the k shortest paths," Tech. Rep., Dep. of Info. and Comp. Sci., University of California, 1994.

[5] S. Verdú, *Multiuser Detection*, Cambridge University Press, 1998.

[6] M. Moher, "Multiuser decoding for multibeam systems," *IEEE Trans. Veh. Technol.*, vol. 49, no. 4, pp. 1226–1234, July 2000.

[7] C. Schlegel and A. J. Grant, "Polynomial complexity optimal detection of certain multiple access systems," *IEEE Trans. Inform. Theory*, vol. 46, no. 6, pp. 2246 –2248, Sep. 2000.

[8] A. B. Reid, A. J. Grant, and P. D. Alexander, "Direct proof of a polynomial complexity optimum detection algorithm," *IEE Electron. Lett.*, vol. 37, no. 19, pp. 1203–1205, Sep. 2001.

[9] E. L. Lawler, "A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem," *Management Science*, vol. 18, no. 7, pp. 401–405, Mar. 1972.

[10] S. Ulukus and R. Yates, "Optimum multiuser detection is tractable for synchronous CDMA systems using m-sequences," *IEEE Comm. Lett.*, vol. 2, no. 4, pp. 89–91, Apr. 1998.

[11] C. Sankaran and A. Ephremides, "Solving a class of optimum multiuser detection problems with polynomial complexity," *IEEE Trans. Inform. Theory*, vol. 44, no. 5, pp. 1958–1961, Sep. 1998.

[12] H. Hamacher, "An $O(K \cdot n^4)$ algorithm for finding the K best cuts in a network," *Operations Research Letters*, vol. 1, no. 5, pp. 186–189, Nov. 1982.

[13] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1639–1642, Jul. 1999.

[14] M. O. Damen, K. Abed-Meraim, and J. C. Belfiore, "A generalized sphere decoder for asymmetrical space-time communication architecture," *IEEE Electron. Lett.*, vol. 36, no. 2, pp. 166, Jan. 2000.

[15] C. Schlegel, *Trellis Coding*, IEEE Press, 1997.

Algorithm 1. P best sequences for given $n = n(\mathbf{d})$

```
Create node n with members n.d_π ← d_opt(n), n.metric← M(d_opt(n)), n.state
←POTENTIAL, n.swapbits← ∅, n.largestswap← 0.
sequences_found ← 0
POTENTIAL_LIST ← n
PATH_LIST ← ∅
while sequences_found ≠num_sequences_desired do
   // Choose Best Potential
   if POTENTIAL_LIST = ∅ then
      All paths found for n(d) negatives. Terminate Algorithm.
   end if
   p ← node p′ ∈POTENTIAL_LIST with smallest p′.metric.
   Append p to PATH_LIST
   Remove p from POTENTIAL_LIST
   p.state←ACTIVE
   sequences_found ← sequences_found +1
   // Extend Active Nodes
   for all n ∈ {p, p.parent} such that n does not have a POTENTIAL child do
      s ← n.largestswap +1
      FIND_s:
      if there is such an s then
         n.largestswap ← s
         if δ(s).swapbits ∩ n.swapbits = ∅ then
            d ← n.d_π with the bits δ(s).swapbits swapped
            if there is no node m ∈POTENTIAL_LIST with m.d_π = d then
               Create child node n′ of n with members
               n′.metric ← n.metric + δ(s).metric
               n′.d_π ← d
               n′.swapbits ← n.swapbits ∪s.swapbits
               n′.largestswap ← s
               n′.state←POTENTIAL
               Append n′ to POTENTIAL_LIST
            else
               Increment s
               goto FIND_s:
            end if
         else
            Increment s
            goto FIND_s:
         end if
      end if
   end for
end while
```