The Conditional Metric Merge Algorithm for Maximum Likelihood Multiuser-Macrodiversity Detection

Lisa Welburn*, James K. Cavers*, and Kevin W. Sowerby†
*School of Engineering Science, Simon Fraser University, Burnaby, B.C. Canada V5A 1S6
†Radio Systems Group, Department of Electrical & Electronic Engineering,
The University of Auckland, Private Bag 92019, Auckland, New Zealand.

Abstract—**The combination of macrodiversity reception with maximum likelihood (ML) multiuser detection has the capability to reduce the bit error rate (BER) for many users by several orders of magnitude compared with multiuser detectors that operate on each antenna separately [1]. In this paper, we present the Conditional Metric Merge (CMM) algorithm which reduces the computational complexity of the ML multiuser-macrodiversity detector by an enormous factor. The CMM algorithm can be viewed as a spatial variant of the Viterbi algorithm. It is a new algorithm and is the first of its kind as ML multiuser-macrodiversity detection (MUMD) is a relatively new area of research.**

I. INTRODUCTION

Traditionally, multiuser detection has been combined with antenna diversity where the antennas are co-located and have the same set of users [2]. In contrast, when multiuser detection is combined with macrodiversity, widely separated base stations or antennas share information about users. Macrodiversity is unlike microdiversity since each user's signal arrives at the various antennas with different mean square values. As a result, the antennas include different, but often overlapping, sets of users in the multiuser-macrodiversity detection (MUMD).

Brute force ML-MUMD requires an exhaustive enumeration over all users resulting in a computational complexity that increases exponentially with the number of users in the system. The spatial CMM algorithm presented in this paper minimizes the complexity while achieving the same optimality as brute force ML-MUMD. When the users are evenly distributed throughout the system, the exponent of exponential growth is scaled by a factor of about $m/L < 1$, where L is the number of macrodiversity antennas in the system and m is the average number of antenna connections per user. For example, if there are K users in the system, and each user is connected to $m = 2$ macrodiversity antennas in a cluster of $L = 7$ cells, the computational complexity is reduced from 2^K to $2^{2K/7}$.

The different and often overlapping sets of users in the ML multiuser-macrodiversity detection result in a metric that is composed of sums of terms that depend on only a few symbols at a time [1]. The resulting metric structure makes the ML multiuser-macrodiversity detector well suited for a dynamic programming approach which keeps its computational complexity at a minimum.

The Conditional Metric Merge (CMM) algorithm presented in this paper is the realization of this dynamic programming approach. It is the key to ML multiuser-macrodiversity detection as it illustrates how the ML multiuser-macrodiversity detector would be implemented in practice. Here we have assumed that the signals undergo flat fading in a synchronous DS-CDMA system to simplify the presentation of the spatial algorithm. A full treatment with delay spread in an asynchronous CDMA system can be found in [3].

The log-likelihood metric for ML-MUMD can be written as

$$\lambda_i = \sum_{l=1}^{L} \lambda_{il} \tag{1}$$

where there are a total of L macrodiversity antennas in the system. The subscript, i, is used to show that all possible data combinations are used in the decision process, and the detector selects the data which minimizes (1). The individual metrics at each antenna, λ_{il}, can be found in [1], and are structured in a similar way to the original metric for ML multiuser detection [4], [5]. Each antenna involves a unique set of users, ψ_l, in the ML-MUMD. Therefore, the metrics, λ_{il}, depend on K_{ψ_l} information symbols, where there are $K_{\psi_l} = |\psi_l|$ users involved in the ML-MUMD at antenna l, and $|\cdot|$ applied to a set denotes the length of the set.

In order to be able to utilize a dynamic programming technique, an optimization problem must have both an optimal substructure and overlapping subproblems [6]. A problem has an optimal substructure if its optimal solution is composed of optimal solutions to subproblems [6]. It can be seen that the ML multiuser-macrodiversity detector exhibits an optimal substructure since the minimization of λ_{il} is an optimal solution to a subproblem within (1). The ML multiuser-macrodiversity detector also has overlapping subproblems since the metrics, λ_{il}, include different, but often overlapping, sets of users in the MUMD. Therefore, the CMM algorithm utilizes a dynamic programming approach to reduce the number of computations required to perform the MUMD.

As stated previously, the metric, λ_{il}, depends on K_{ψ_l} information symbols; therefore, the number of probabilities calculated per stage of the algorithm is $2^{K_{\psi_l}}$. However, since each user may be connected to a different number of antennas, the complexity per binary decision is variable. For example, if user x is connected to the antennas in $\mathcal{A}_x$, the number of probabilities calculated per binary decision

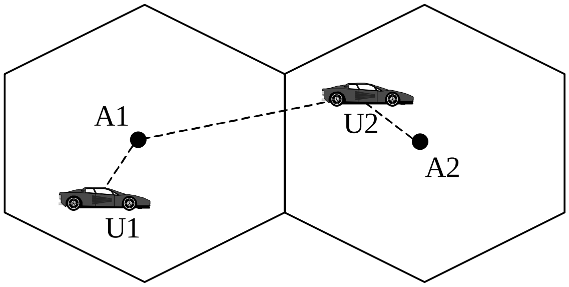


Fig. 1. Typical connections for 2 users and 2 antennas.

for user x is $\sum_{l\in\mathcal{A}_x} 2^{K_{\psi_l}}$.

In order to describe the CMM algorithm, an index variable, h, is used to count through the antennas, where $h = l$, and $1 \leq l \leq L$. The data symbols that are included in the metric, λ_{il}, are stored in the set

$$\Psi\left(h\right) = \left\{\, b_x\left(n\right) \middle|\, x \in \psi_l \right\} \tag{2}$$

where $\left|\Psi\left(h\right)\right| = K_{\psi_l}$. For example, the data symbol sets, $\Psi\left(h\right)$, can be constructed using the connections depicted in Fig. 1. As can be seen from the figure, user 1, U1, is communicating with antenna 1, A1, only, while user 2, U2, is communicating with both antennas 1 and 2, A1 and A2. By using Fig. 1 and (2), it can be observed that the data set for A1 is $\Psi\left(1\right) = \left\{b_1\left(n\right), b_2\left(n\right)\right\}$, and the data set for A2 is $\Psi\left(2\right) = \left\{b_2\left(n\right)\right\}$.

The metric, λ_{il}, is calculated for all values of the data symbols in $\Psi\left(h\right)$. If we define $\alpha = \{0,1\}$ as the alphabet for a BPSK signal, there are $\left|\alpha\right|^{\left|\Psi(h)\right|} = 2^{K_{\psi_l}}$ possible combinations of the data symbol values in $\Psi\left(h\right)$ and thus $2^{K_{\psi_l}}$ metric calculations.

The metrics that are calculated for all of the various data symbol values in $\Psi\left(h\right)$ are stored in the set

$$\Lambda\left(h\right) = \left\{ \lambda_{il} \middle|\, i = 1, ..., \left|\alpha\right|^{\left|\Psi(h)\right|} \right\} \tag{3}$$

The metrics at each stage of the algorithm can be merged by using the Combine routine. If we define $\mathbf{h} = [1, ..., h]$, the combined metrics may be expressed as

$$\Lambda\left(\mathbf{h}\right) = \Lambda\left(\mathbf{f}\right) \oplus \Lambda\left(h\right) \tag{4}$$

where $\mathbf{f} = [1, ..., h-1]$ and $\oplus$ represents the invoked Combine routine. The Combine routine is analogous to metric extension in the Viterbi algorithm where the computed metrics form continuations on the surviving sequences from the previous stage [7]. The following section describes how the Combine routine merges the two different sets of metrics, $\Lambda\left(\mathbf{f}\right)$ and $\Lambda\left(h\right)$.

II. The Combine Routine

This section provides a description of the Combine routine in II-A, and it also provides a practical example of this routine in II-B.

A. The Combine Routine's Description

In this routine, we combine the metrics, $\Lambda\left(\mathbf{f}\right)$ and $\Lambda\left(h\right)$. The data symbols that correspond with $\Lambda\left(\mathbf{f}\right)$ and $\Lambda\left(h\right)$ are given by $\Psi\left(\mathbf{f}\right)$ and $\Psi\left(h\right)$ respectively. In the atypical situation where $\Psi\left(\mathbf{f}\right) = \Psi\left(h\right)$, $\Lambda\left(\mathbf{h}\right) = \Lambda\left(\mathbf{f}\right) \oplus \Lambda\left(h\right)$ could be performed by simply adding the respective metrics within each set. However, normally $\Psi\left(\mathbf{f}\right) \neq \Psi\left(h\right)$. Therefore, we create $\Psi\left(\mathbf{h}\right) = \Psi\left(\mathbf{f}\right) \cup \Psi\left(h\right)$, and we repeat the metrics in $\Lambda\left(\mathbf{f}\right)$ and $\Lambda\left(h\right)$ in a manner so that they correspond with $\Psi\left(\mathbf{h}\right)$. In order to repeat the metrics, we use the function, $I_w = repindex\left(\Psi\left(w\right), \Psi\left(\mathbf{h}\right)\right)$ for both $w = \mathbf{f}$ and $w = h$. The returned set, I_w, contains the relevant indices in $\Psi\left(\mathbf{h}\right)$ of the elements in $\Psi\left(\mathbf{h}\right) - \Psi\left(w\right)$. Once we have I_w for both $w = \mathbf{f}$ and $w = h$, we can use this information to repeat the metrics in $\Lambda\left(\mathbf{f}\right)$ and $\Lambda\left(h\right)$ so that $\left|\Lambda\left(\mathbf{f}\right)\right| = \left|\Lambda\left(h\right)\right| = \left|\alpha\right|^{\left|\Psi(\mathbf{h})\right|}$, and the sets can be merged by simply adding the respective metrics. This type of repetition can be performed since the data symbols in $\Psi\left(\mathbf{h}\right) - \Psi\left(\mathbf{f}\right)$ have no effect on the metrics in $\Lambda\left(\mathbf{f}\right)$, and the data symbols in $\Psi\left(\mathbf{h}\right) - \Psi\left(h\right)$ have no effect on the metrics in $\Lambda\left(h\right)$.

In order to repeat the metrics in $\Lambda\left(w\right)$, we use the elements in I_w which are given by $I_{w,j}$ where $1 \leq j \leq \left|I_w\right|$. By starting with $j = 1$ and working toward $j = \left|I_w\right|$, we repeat the metrics in $\Lambda\left(w\right)$ as follows. The metrics are sequentially divided into $\left|\alpha\right|^{\left|\Psi(w)\right|+j-I_{w,j}}$ groups, and each group of $\left|\alpha\right|^{I_{w,j}-1}$ metrics is repeated so that it appears twice in succession. Therefore, within the Combine routine, the metrics, $\Lambda\left(w\right)$, that initially correspond with $\Psi\left(w\right)$ are repeated so that they correspond with $\Psi\left(\mathbf{h}\right)$. Once this repetition is performed for both $w = \mathbf{f}$ and $w = h$, $\Lambda\left(\mathbf{h}\right) = \Lambda\left(\mathbf{f}\right) \oplus \Lambda\left(h\right)$ is easily calculated, and the Combine routine returns both $\Lambda\left(\mathbf{h}\right)$ and $\Psi\left(\mathbf{h}\right)$.

B. Example of the Combine Routine

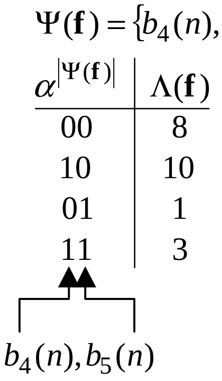
The inputs to the Combine routine are the data symbol sets, $\Psi\left(\mathbf{f}\right)$ and $\Psi\left(h\right)$, and their corresponding metrics, $\Lambda\left(\mathbf{f}\right)$ and $\Lambda\left(h\right)$ respectively. Suppose the Combine routine is initialized with the following data symbol sets

$$\Psi\left(\mathbf{f}\right) = \left\{b_4\left(n\right), b_5\left(n\right)\right\} \tag{5}$$

$$\Psi\left(h\right) = \left\{b_5\left(n\right), b_6\left(n\right)\right\} \tag{6}$$

and the metrics in Figures 2 and 3, where $\alpha^{\left|\Psi(w)\right|}$ denotes $\left|\Psi\left(w\right)\right|$ cross products on the set α [8]. In Figure 2, $\alpha^{\left|\Psi(\mathbf{f})\right|}$ lists all possible combinations of the data symbol values in $\Psi\left(\mathbf{f}\right)$, and the metrics in $\Lambda\left(\mathbf{f}\right)$ are calculated for each corresponding entry of $\alpha^{\left|\Psi(\mathbf{f})\right|}$. Normally the metrics are calculated using the ML multiuser-macrodiversity detector. However, in this example, the actual metrics are replaced with integers for simplicity. In a similar manner to that described above, $\alpha^{\left|\Psi(h)\right|}$ in Figure 3 lists all possible combinations of the data symbol values in $\Psi\left(h\right)$, and each entry of $\alpha^{\left|\Psi(h)\right|}$ is used to calculate the metrics in $\Lambda\left(h\right)$.

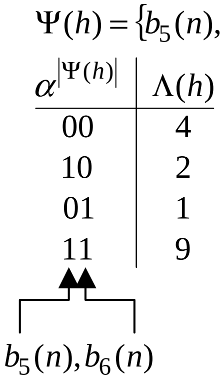
Now that the data symbol sets and their corresponding metrics have been defined, the $repindex$ function is used



$\Psi(\mathbf{f}) = \{b_4(n), b_5(n)\}$

$\alpha^{\lvert\Psi(\mathbf{f})\rvert}$	$\Lambda(\mathbf{f})$
00	8
10	10
01	1
11	3

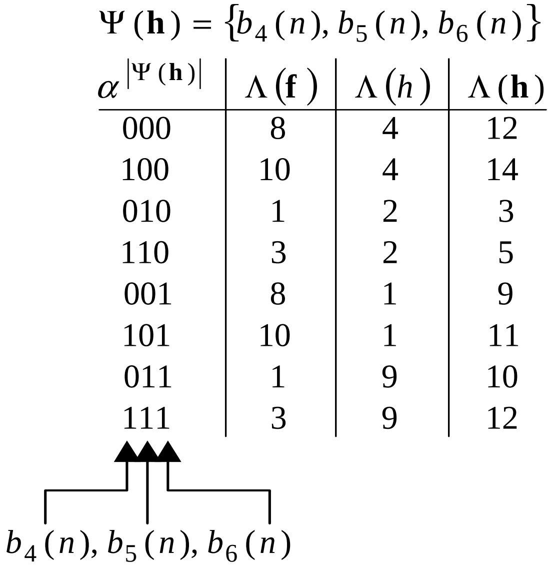
Fig. 2. Sample metrics for $\Lambda(\mathbf{f})$



$\Psi(h) = \{b_5(n), b_6(n)\}$

$\alpha^{\lvert\Psi(h)\rvert}$	$\Lambda(h)$
00	4
10	2
01	1
11	9

Fig. 3. Sample metrics for $\Lambda(h)$



$\Psi(\mathbf{h}) = \{b_4(n), b_5(n), b_6(n)\}$

$\alpha^{\lvert\Psi(\mathbf{h})\rvert}$	$\Lambda(\mathbf{f})$	$\Lambda(h)$	$\Lambda(\mathbf{h})$
000	8	4	12
100	10	4	14
010	1	2	3
110	3	2	5
001	8	1	9
101	10	1	11
011	1	9	10
111	3	9	12

Fig. 4. The merging of sample metrics, $\Lambda(\mathbf{f})$ and $\Lambda(h)$

to provide the proper repetition for both $\Lambda(\mathbf{f})$ and $\Lambda(h)$. The function, $I_w = repindex(\Psi(w), \Psi(\mathbf{h}))$ returns the relevant indices in $\Psi(\mathbf{h})$ of the elements in $\Psi(\mathbf{h}) - \Psi(w)$ for both $w = \mathbf{f}$ and $w = h$. $\Psi(\mathbf{h})$ is simply the union of the sets, $\Psi(\mathbf{f})$ and $\Psi(h)$. Therefore,

$$\Psi(\mathbf{h}) = \Psi(\mathbf{f}) \cup \Psi(h) = \{b_4(n), b_5(n), b_6(n)\} \quad (7)$$

From (5) and (6), we calculate

$$\Psi(\mathbf{h}) - \Psi(\mathbf{f}) = \{b_6(n)\} \quad (8)$$

$$\Psi(\mathbf{h}) - \Psi(h) = \{b_4(n)\} \quad (9)$$

Therefore,

$$I_{\mathbf{f}} = repindex(\Psi(\mathbf{f}), \Psi(\mathbf{h})) = \{3\} \quad (10)$$

and

$$I_h = repindex(\Psi(h), \Psi(\mathbf{h})) = \{1\} \quad (11)$$

In other words, counting from left to right, the symbol $b_6(n)$ in $\Psi(\mathbf{h}) - \Psi(\mathbf{f})$ appears in position $I_{\mathbf{f}} = \{3\}$ of $\Psi(\mathbf{h})$, and the symbol $b_4(n)$ in $\Psi(\mathbf{h}) - \Psi(h)$ appears in position $I_h = \{1\}$ of $\Psi(\mathbf{h})$. Now that $I_{\mathbf{f}}$ and I_h have been calculated, they will be used to repeat the metrics in $\Lambda(\mathbf{f})$ and $\Lambda(h)$ so that $|\Lambda(\mathbf{f})| = |\Lambda(h)| = |\alpha|^{|\Psi(\mathbf{h})|}$.

As stated previously, the metrics in $\Lambda(w)$ for both $w = \mathbf{f}$ and $w = h$ are sequentially divided into $|\alpha|^{|\Psi(w)|+j-I_{w,j}}$ groups, and each group of $|\alpha|^{I_{w,j}-1}$ metrics is repeated so that it appears twice in succession. Since $|I_{\mathbf{f}}| = |I_h| = 1$ in this example, only one repetition is needed for both $\Lambda(\mathbf{f})$ and $\Lambda(h)$, and j attains a maximum value of 1 for both $w = \mathbf{f}$ and $w = h$.

The metrics in $\Lambda(\mathbf{f})$ are divided into $|\alpha|^{|\Psi(\mathbf{f})|+j-I_{\mathbf{f},j}} = 2^{2+1-3} = 1$ group of $|\alpha|^{I_{\mathbf{f},j}-1} = 2^{3-1} = 4$ metrics, and this group is repeated so that the metrics appear as shown in $\Lambda(\mathbf{f})$ of Figure 4. Similarly, the metrics in $\Lambda(h)$ are divided into $|\alpha|^{|\Psi(h)|+j-I_{h,j}} = 2^{2+1-1} = 4$ groups of $|\alpha|^{I_{h,j}-1} = 2^{1-1} = 1$ metric, and these groups are each repeated so that they appear twice in succession as shown in $\Lambda(h)$ of Figure 4. As can be seen from Figure 4, $|\Lambda(\mathbf{f})| = |\Lambda(h)| = |\alpha|^{|\Psi(\mathbf{h})|}$ as expected, and $\Lambda(\mathbf{h})$ can now be calculated by simply adding the respective metrics within $\Lambda(\mathbf{f})$ and $\Lambda(h)$.

It is observed that $\Psi(\mathbf{h})$ may contain data symbols that do not appear in any of the subsequent data symbol sets, $\Psi(k)$ for $k > h$. If this is the case, we can discard the least-probable metrics with respect to these data symbols in the Select routine. The Select routine is analogous to the selection process in the Viterbi algorithm where the sequences are subdivided into groups that differ only in the symbol which is not involved in any subsequently received signals. From each group, the surviving sequence is the one having the largest probability, and the rest are discarded [7].

III. The Select Routine

This section describes the Select routine in III-A, and a practical example is provided in III-B which further portrays the functionality of this routine.

A. The Select Routine's Description

We can make tentative decisions on the data symbols in the set

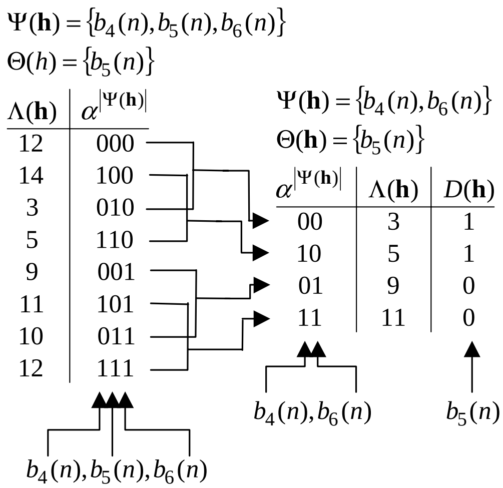
$$\Theta(h) = \Psi(\mathbf{h}) - \bigcup_{k>h} \Psi(k) \tag{12}$$

since the symbols in $\Theta(h)$ do not appear in any of the subsequent data symbol sets, $\Psi(k)$ for $k > h$. The elements in $\Theta(h)$ are given by $\Theta_j(h)$ where $1 \leq j \leq |\Theta(h)|$. Assuming that we start with $j = 1$ and work toward $j = |\Theta(h)|$, we can make a tentative decision on $\Theta_j(h)$ by dividing the $|\alpha|^{|\Psi(\mathbf{h})|-j+1}$ metrics into $|\alpha|^{|\Psi(\mathbf{h})|-j}$ groups with each group containing $|\alpha|$ metrics that differ in the symbol $\Theta_j(h)$. From each group of the $|\alpha|$ metrics, we select the minimum and discard the remaining $|\alpha| - 1$ metrics. In a similar way to the Viterbi algorithm, the decision for $\Theta_j(h)$ is final if all of the groups make the same decision. Otherwise the decision for $\Theta_j(h)$ remains tentative and is deferred to a later stage. Once tentative decisions have been made on all of the symbols in $\Theta(h)$, the number of metrics in $\Lambda(\mathbf{h})$ has been reduced by a factor of $|\alpha|^{|\Theta(h)|}$, and $\Psi(\mathbf{h})$ is replaced by $\Psi(\mathbf{h}) - \Theta(h)$. In total, we have made tentative decisions on the data symbols in $\Theta(\mathbf{h}) = \bigcup_{1\leq j\leq h} \Theta(j)$, and these tentative decisions are stored in $D(\mathbf{h})$, where each entry of $D(\mathbf{h})$ contains $|\Theta(\mathbf{h})|$ tentative decisions, and the entire set has $|\alpha|^{|\Psi(\mathbf{h})|}$ entries.

B. Example of the Select Routine

The outputs of the Combine routine are used as inputs for the Select routine. From the example in Section II-B, the data set, $\Psi(\mathbf{h})$ and corresponding metric set, $\Lambda(\mathbf{h})$ are shown in Figure 4. As stated previously, tentative decisions can be made on the data symbols in the set, $\Theta(h) = \Psi(\mathbf{h}) - \bigcup_{k>h} \Psi(k)$. For example, if the data symbol, $b_5(n)$, does not appear in any of the future data symbol sets, i.e. $b_5(n) \notin \bigcup_{k>h} \Psi(k)$, then $b_5(n) \in \Theta(h)$. Further, if $b_5(n)$ is the only symbol in $\Psi(\mathbf{h})$ that does not appear in future data symbol sets, then $\Theta(h) = \{b_5(n)\}$. The elements in $\Theta(h)$ are given by $\Theta_j(h)$ where $1 \leq j \leq |\Theta(h)|$, and since $|\Theta(h)| = 1$ in this example, j attains a maximum value of 1. Therefore, a tentative decision is currently made for $b_5(n)$ only. In order to make a tentative decision for $b_5(n)$, the $|\alpha|^{|\Psi(\mathbf{h})|-j+1} = 2^{3-1+1} = 8$ metrics are divided into $|\alpha|^{|\Psi(\mathbf{h})|-j} = 2^{3-1} = 4$ groups with each group containing $|\alpha| = 2$ metrics that differ in the symbol $\Theta_j(h) = b_5(n)$.

In Figure 4, $\alpha^{|\Psi(\mathbf{h})|}$ lists all possible combinations of the data symbol values in $\Psi(\mathbf{h})$, and $\Lambda(\mathbf{h})$ lists the corresponding metrics. Fig. 5 repeats this information, but initially reverses the order of $\alpha^{|\Psi(\mathbf{h})|}$ and $\Lambda(\mathbf{h})$ for purposes of description. As can be seen from Fig. 5, the arrows show how the 8 metrics are divided into 4 groups, and each group differs only in the symbol, $b_5(n)$. Within each group, a decision is formed by selecting the minimum metric and discarding the larger of the 2. It can be seen from Fig. 5 that the number of metrics has been reduced by a factor of $|\alpha|^{|\Theta(h)|} = 2^1 = 2$, and $\Psi(\mathbf{h})$ is replaced by $\Psi(\mathbf{h}) - \Theta(h)$. Thus, $\Psi(\mathbf{h})$ becomes $\{b_4(n), b_6(n)\}$, and the tentative decisions for $b_5(n)$ are stored in $D(\mathbf{h})$.


$\Psi(\mathbf{h}) = \{b_4(n), b_5(n), b_6(n)\}$
$\Theta(h) = \{b_5(n)\}$
$\Psi(\mathbf{h}) = \{b_4(n), b_6(n)\}$
$\Theta(\mathbf{h}) = \{b_5(n)\}$
$b_4(n), b_5(n), b_6(n)$
$b_4(n), b_6(n)$
$b_5(n)$

$\Lambda(\mathbf{h})$	$\alpha^{\lvert\Psi(\mathbf{h})\rvert}$
12	000
14	100
3	010
5	110
9	001
11	101
10	011
12	111

$\alpha^{\lvert\Psi(\mathbf{h})\rvert}$	$\Lambda(\mathbf{h})$	$D(\mathbf{h})$
00	3	1
10	5	1
01	9	0
11	11	0

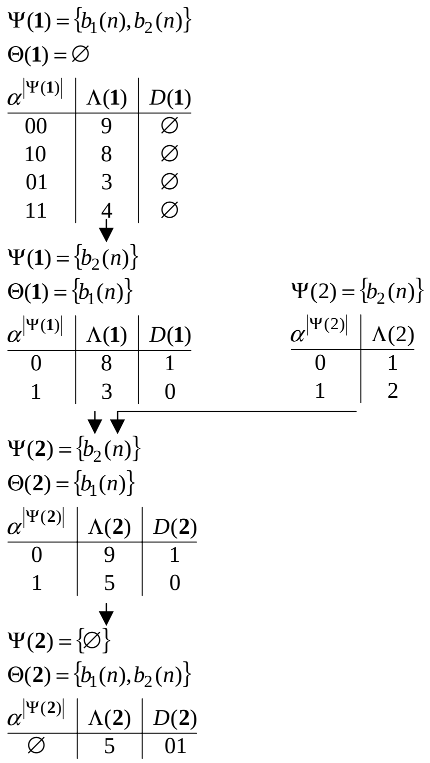
Fig. 5. The Select routine applied to sample metrics, $\Lambda(\mathbf{h})$

The Combine and Select routines form the major components of the CMM algorithm. Now that these routines have been described in detail, the following section will provide an additional example which further illustrates the CMM algorithm's operation.

IV. A Typical System Using the CMM Algorithm

The major components of the CMM algorithm have now been addressed in terms of the Combine and Select routines. This example is used to demonstrate how these components fit together to form the overall algorithm. It is based on the connections for the $K = 2$ users and $L = 2$ antennas depicted in Fig. 1. Neither the Combine routine or the Select routine will be described in detail as examples of both of these were given in the Sections II-B and III-B respectively. Therefore, in Fig. 6, double arrows are used to represent the Combine routine, while single arrows represent the Select routine.

The relevant metrics, $\Lambda(h)$, are normally calculated using the ML multiuser-macrodiversity detector. However, for this example the actual metrics are replaced with integers for simplicity. As can be seen in Fig. 6, the data set for A1 is $\Psi(1) = \{b_1(n), b_2(n)\}$, and the data set for A2 is $\Psi(2) = \{b_2(n)\}$. The CMM algorithm in Fig. 6



$\Psi(\mathbf{1}) = \{b_1(n), b_2(n)\}$

$\Theta(\mathbf{1}) = \varnothing$

$\alpha^{\lvert\Psi(\mathbf{1})\rvert}$	$\Lambda(\mathbf{1})$	$D(\mathbf{1})$
00	9	∅
10	8	∅
01	3	∅
11	4	∅

$\Psi(\mathbf{1}) = \{b_2(n)\}$

$\Theta(\mathbf{1}) = \{b_1(n)\}$

$\alpha^{\lvert\Psi(\mathbf{1})\rvert}$	$\Lambda(\mathbf{1})$	$D(\mathbf{1})$
0	8	1
1	3	0

$\Psi(2) = \{b_2(n)\}$

$\alpha^{\lvert\Psi(2)\rvert}$	$\Lambda(2)$
0	1
1	2

$\Psi(\mathbf{2}) = \{b_2(n)\}$

$\Theta(\mathbf{2}) = \{b_1(n)\}$

$\alpha^{\lvert\Psi(\mathbf{2})\rvert}$	$\Lambda(\mathbf{2})$	$D(\mathbf{2})$
0	9	1
1	5	0

$\Psi(\mathbf{2}) = \{\varnothing\}$

$\Theta(\mathbf{2}) = \{b_1(n), b_2(n)\}$

$\alpha^{\lvert\Psi(\mathbf{2})\rvert}$	$\Lambda(\mathbf{2})$	$D(\mathbf{2})$
∅	5	01

Fig. 6. CMM example for the connections shown in Fig. 1

is initialized by setting $\Psi(\mathbf{1}) = \Psi(1)$ and $\Lambda(\mathbf{1}) = \Lambda(1)$. Since no tentative decisions have been formed at this point, $\Theta(\mathbf{1}) = \varnothing$, and each entry of $D(\mathbf{1})$ is set to $\varnothing$. It is immediately observed that the symbol, $b_1(n)$ does not appear in the subsequent data symbol set, $\Psi(2)$. Therefore, a tentative decision can be made on $b_1(n)$ without impacting the final result. As expected, the number of metrics in $\Lambda(\mathbf{1})$ has been reduced by a factor of 2. Once the appropriate tentative decisions have been made, the Combine routine is used to execute $\Lambda(\mathbf{2}) = \Lambda(\mathbf{1}) \oplus \Lambda(2)$. Once $\Lambda(\mathbf{2})$ has been calculated, no further metrics need to be combined to this result. Therefore, final decisions can now be made for $b_1(n)$ and $b_2(n)$ by calling the Select routine one last time. These final decisions are found in $D(\mathbf{2})$ at the bottom of Fig. 6.

V. Practical Considerations of the CMM Algorithm

In each stage of the CMM algorithm, $2^{K_{\psi_l}}$ probabilities are computed. Therefore, in total $\sum_{l=1}^{L} 2^{K_{\psi_l}}$ probabilities are computed to detect the symbols for each of the K different users at any time n.

So far in the CMM algorithm's description, $\Lambda(\mathbf{h})$ has been calculated in a consecutive fashion, starting with $h = 1$ and ending with $h = L$. However, in order to speed up the process, the $\Lambda(h)$'s could be calculated simultaneously since data is received at a number of different antennas. In a similar way to traditional ML multiuser detection, a processor at each antenna could calculate its metrics, λ_{il}. The unique aspects of ML-MUMD enter through the use of the Combine and Select routines in the CMM algorithm. In fact, the Combine routine could be invoked many times in parallel, but the Select routine should not be invoked until the interim metrics have been added to the current total at the system controller.

VI. Conclusions

This paper has described the CMM algorithm for ML multiuser-macrodiversity detection in general terms, and it has also provided specific examples which illustrate the algorithm's operation. The complexity of the ML multiuser-macrodiversity detector grows exponentially with the number of users since it is based on the detector developed in [4]. To mitigate this growth, the new CMM algorithm is designed to keep the exponent of exponential growth for the ML multiuser-macrodiversity detector to a minimum while producing the same data decisions as the brute force ML multiuser-macrodiversity detector. The CMM algorithm presented in this paper can be viewed as a spatial variant of the Viterbi algorithm. It can be applied to a synchronous DS-CDMA system using ML-MUMD when the signals undergo flat fading. A later publication will show how the CMM algorithm can be extended to operate in both space and time to encompass asynchronous DS-CDMA systems in which the users experience frequency selective fading.

Acknowledgment

The authors gratefully acknowledge TELUS Mobility for funding this work.

References

[1] L. Welburn, J. K. Cavers, and K. W. Sowerby, "Multiuser-macrodiversity detection in Rayleigh fading channels," *Proc. IEEE Veh. Tech. Conf.*, Boston, MA, September 2000.

[2] Z. Zvonar, "Combined multiuser detection and diversity reception for wireless CDMA systems," *IEEE Trans. on Vehicular Technology*, vol. 45, no. 1, pp. 205-211, Feb. 1996.

[3] L. Welburn, "Macrodiversity and multiuser detection: high performance tools in a DS-CDMA system," Ph.D. Thesis, Simon Fraser University, May 2001.

[4] S. Verdu, "Minimum probability of error for asynchronous Gaussian multiple-access channels," *IEEE Trans. on Information Theory*, vol. IT-32, no. 1, pp. 85-96, Jan. 1986.

[5] Z. Zvonar and D. Brady, "Multiuser detection in single-path fading channels," *IEEE Trans. on Communications*, vol. 42, no. 2/3/4, pp. 1729-1739, Feb./Mar./April 1994.

[6] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. New York: McGraw-Hill, 1990.

[7] J. G. Proakis, *Digital Communications*. Second Edition, New York: McGraw-Hill Book Company, 1989.

[8] R. P. Grimaldi, *Discrete and Combinatorial Mathematics, An Applied Introduction* 3rd Edition. Reading, Massachusetts, Addison-Wesley Publishing Company, 1994.