

**A COMPUTATIONAL PARADIGM
FOR
SPACE-TIME MULTIUSER DETECTION**

Lisa Welburn*, Jim Cavers*, Kevin Sowerby**

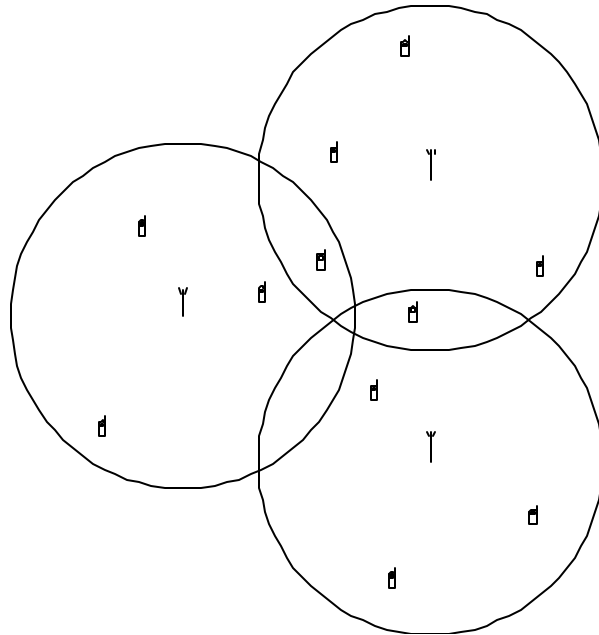
* Simon Fraser University, Canada

** The University of Auckland, New Zealand

OUTLINE:

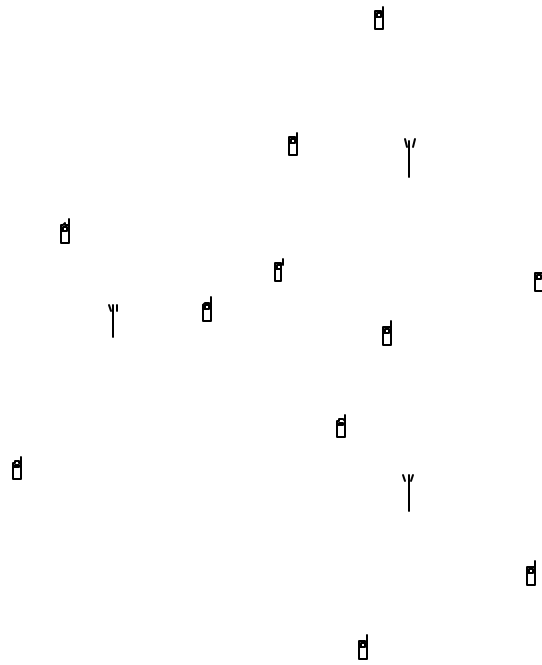
- Space-time MUD and the value of ML
- The Viterbi algorithm reinterpreted
- MUD in space – the new CMM algorithm
- MUD in space-time

1. ML-MUD in Space



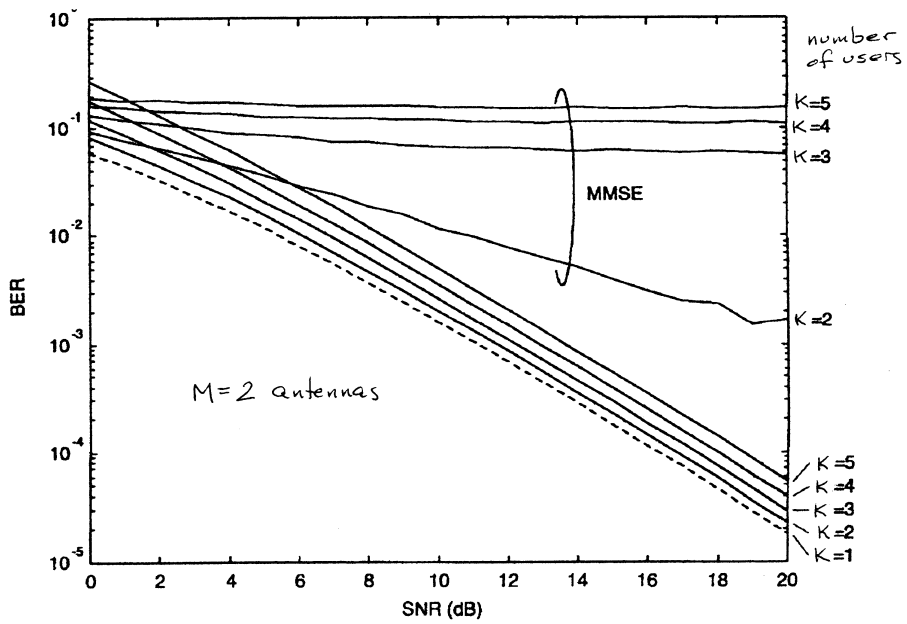
- This picture is ingrained... mobiles “belong” to a cell, that is, a base station.
- But some mobiles can be received by more than one base.
Normally, we deal with it by:
 - ⇒ ignoring it;
 - ⇒ or soft handoff;
 - ⇒ or some ad hoc form of macrodiversity.

- What if there were no cells, or irregular, overlapping cells?
Examples: indoor systems, campus networks. The picture:



- View antennas as sensors in a common field. Each antenna picks up a subset of the users, and the subsets overlap.
- Interference mitigation is a central issue. It is most effective if bases pool information over a backbone wireline network and perform multiuser detection.

- Linear or nonlinear MUD?
- Contrast a single base station's use of MMSE estimation and ML estimation of transmitted data from mobiles:



⇒ With MMSE estimation, error rate degrades rapidly with increasing user numbers.

⇒ ML detection retains diversity order and good error rate, almost irrespective of number of users – but its computational load is very large.

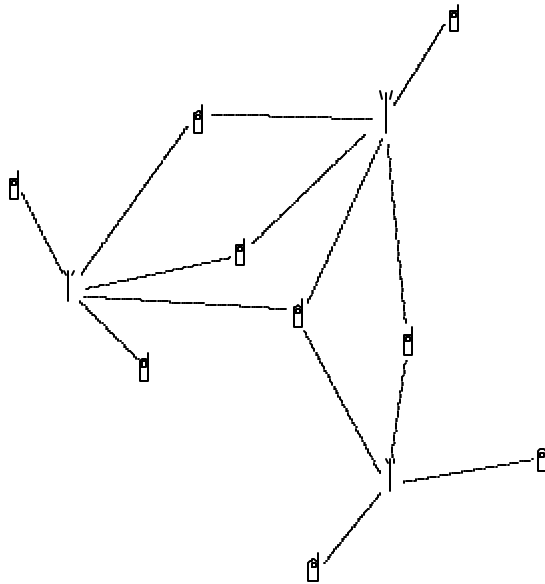
- There is strong motivation find methods that
 - ⇒ offer near-ML performance with greatly reduced computational load, or
 - ⇒ do ML in some way that keeps the exponent of growth to a minimum.

We explore the second tack today. Our goal:
joint ML detection of every bit sent by every user.

- The spatial dimension – overlapping subsets of received users – is critical. Our solution:
 - ⇒ a new algorithm, based on dynamic programming, that keeps the exponent of growth to a minimum in various time-space configurations;
 - ⇒ “conditional metric merge” (CMM);
 - ⇒ the algorithm serves as a computational model on which suboptimal, lower computation, algorithms can be based.

2. Signals and Interference

- Many mobiles, many antennas...

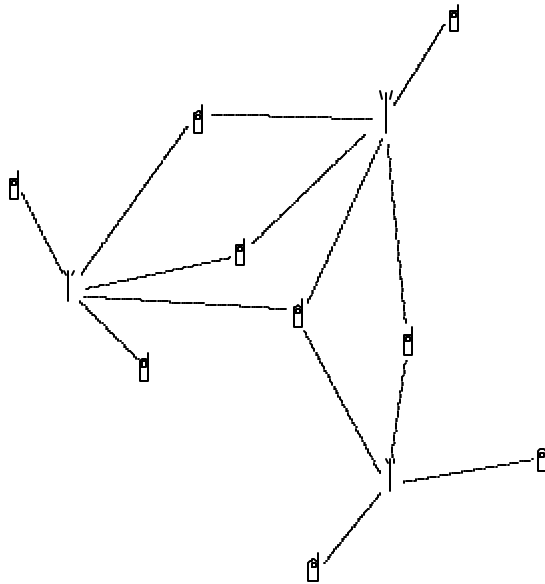


- Mobiles send sequence of pulses weighted by discrete data values, like $b_k(n) = \pm 1$ for mobile $k = 1 \dots K$, time $n = 1 \dots N$. Aggregate symbols to $\mathbf{b}(n)$.
- Measurement at antennas at each symbol time: $y_m(n)$ for antenna $m = 1 \dots M$. Aggregate to $\mathbf{y}(n)$.
- Stack all measurements and data for $n = 1 \dots N$, to get

$$\mathbf{y} = \mathbf{C}\mathbf{b} + \mathbf{n}$$

Discrete \mathbf{b} affects noisy measurements linearly.

- Details: gain matrix \mathbf{C} is composed of $M \times K$ blocks; system structure determines block structure.



⇒ If echoes are limited, then $\mathbf{y}(n)$ depends on mobile bits $\mathbf{b}(n)$ at a few consecutive times; *e.g.*, $n, n-1$ if a short impulse response, giving \mathbf{C} a block diagonal and block subdiagonal only. Viterbi algorithm.

⇒ Not all mobiles appear significantly at all antennas. The blocks will contain zeros, and rows will contain different user subsets. Spatial CMM algorithm.

- ML detection:

⇒ With uncorrelated noise components, the log-likelihood of some candidate data vector \mathbf{b} (all bits from all users) is proportional to

$$\|\mathbf{y} - \mathbf{C}\mathbf{b}\|^2$$

⇒ Just run through all 2^{NK} candidate \mathbf{b} values, and choose the one with the smallest metric (!)

- But all is not lost – the metric is a sum of contributions from each measurement (in time and space)

$$\|\mathbf{y} - \mathbf{C}\mathbf{b}\|^2 = \sum_n \sum_m \left| y_m(n) - \sum_k C_{nM+m, nK+k} b_k(n) \right|^2$$

The gain matrix \mathbf{C} has a sparse structure, so each measurement depends on a relatively small number of bits.

- Dynamic programming can be applied usefully to a problem if:

1. The optimal solution consists of optimal solutions to a set of subproblems, each of which consists of optimal solutions to a set of subsubproblems... i.e., the optimal solution has a recursive structure.

In our case, the lowest level subproblem is optimisation at each measurement $y_m(n)$, although recursion is based on larger aggregates of measurements.

and

2. The subproblems overlap, so that subproblems share subsubproblems.

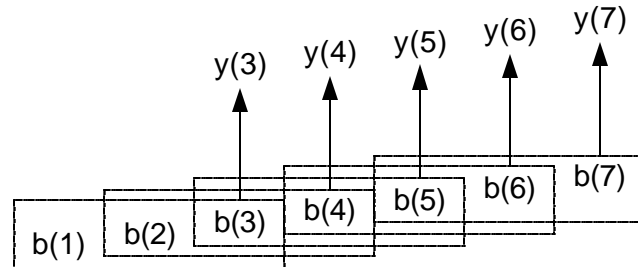
In our case, the metrics associated with the measurements include different, but possibly overlapping, sets of users.

- Unlike pure recursion, which may solve shared subsubproblems repeatedly, DP solves them once and saves the result for combinations at the next higher level.

3. Viterbi Reinterpreted

- The most famous application of dynamic programming is the Viterbi Algorithm, a ubiquitous component of communication systems.
- We'll review what is now the conventional description, then reinterpret it as a stepping stone to the new spatial extension.

- Consider a single source and a single receiver, binary (± 1) transmission, some echoes, so that the measurement at any symbol time is affected by (say) 3 successive bits.



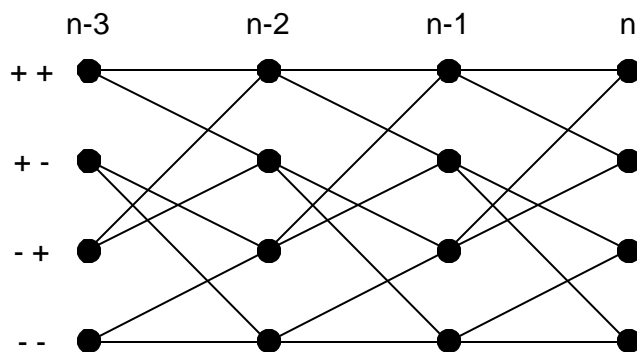
- The contribution to the metric from measurement n is

$$\mathbf{m} = |y(n) - c_0 b(n) - c_1 b(n-1) - c_2 b(n-2)|^2$$

and the gains matrix looks like

$$\mathbf{C} = \begin{pmatrix} c_2 & c_1 & c_0 & 0 & 0 & 0 & \blacksquare & 0 & 0 \\ 0 & c_2 & c_1 & c_0 & 0 & 0 & \blacksquare & 0 & 0 \\ 0 & 0 & c_2 & c_1 & c_0 & 0 & \blacksquare & 0 & 0 \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ 0 & 0 & 0 & 0 & 0 & 0 & \blacksquare & c_0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \blacksquare & c_1 & c_0 \end{pmatrix}$$

- Signals are usually represented by a trellis of data possibilities, where a state is 2 successive bits, so that a trellis transition (a branch) specifies the required 3 successive bits.



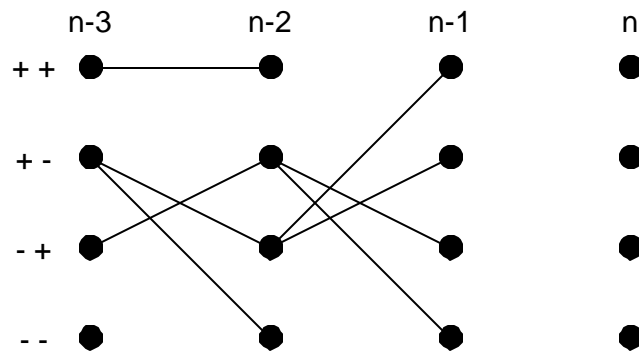
⇒ Problem: Many paths through the trellis terminate at a particular state at the final time N . Find the one with minimum sum of branch metrics. Then select the state with the correspondingly minimised metric and release its minimising path.

- The Viterbi recursion:

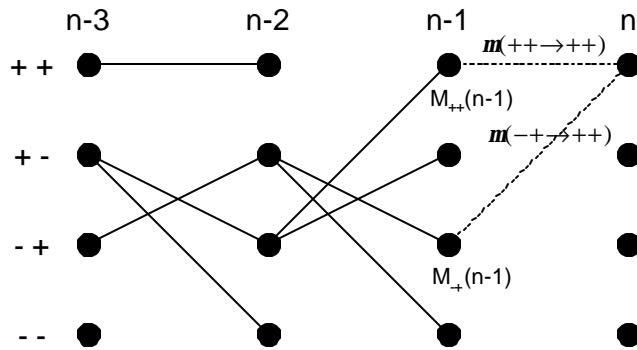
⇒ Assume that the problem has been solved up to time

$n-1$; that is, for each state at $n-1$ (say, +-), we have

- the accumulated metric $M_{+-}(n-1)$
- the “survivor path” $\mathbf{P}_{+-}(n-1)$ of bits prior to those in the state



⇒ The update: the best path terminating in state (say) ++ at time n must go through one of ++ or -+ at time $n-1$.



- Add the metrics:

$$M_{++}(n) = \begin{cases} M_{++}(n-1) + \mathbf{m}(y(n), ++ \rightarrow ++) \\ \text{or} \\ M_{-+}(n-1) + \mathbf{m}(y(n), -+ \rightarrow ++) \end{cases}$$

Select the smaller. Suppose it is -+...

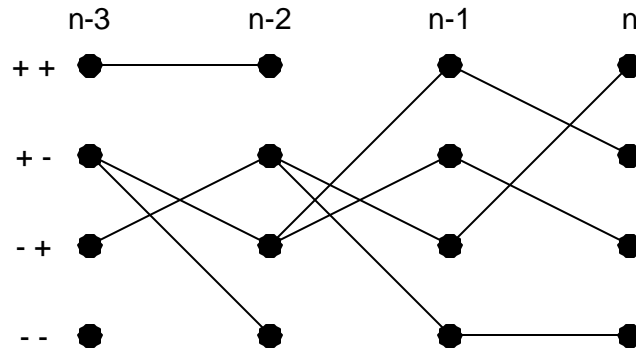
- ... the metric is then

$$M_{++}(n) = M_{-+}(n-1) + \mathbf{m}(y(n), -+ \rightarrow ++)$$

and the survivor path is

$$\mathbf{P}_{++}(n) = (\mathbf{P}_{-+}(n-1), -)$$

⇒ Repeat for remaining states at time n :



then move on to $n+1, \dots, N$ and it's done.

- The VA exploits the sparsity and locality of \mathbf{C} entries to keep the computation proportional to the number of branches. So it's $2^3 N$ – linear in N , exponential only in the number of bits that affect any measurement.

- Now to go over the problem again with the tabular orientation typical of much dynamic programming.
- Two subproblems at time n :
 - \Rightarrow best path $b[1:n-1]$ and metric $M(n-1)$ based on $y[1:n-1]$;
 - \Rightarrow best selection of bits $b[n-2:n]$ based on $y[n]$.
 - \Rightarrow They overlap.
- Distinguish between “live bits” – ones that can affect future measurements – and “dead bits” – ones that form the survivor path in the trellis description.
- We index the solutions to the first subproblem by its live bits:

$b(n-2)$	$b(n-1)$	$\mathbf{P}(n-1)$	$M(n-1)$
+1	+1	$\mathbf{P}_{++}(n-1)$	$M_{++}(n-1)$
+1	-1	$\mathbf{P}_{+-}(n-1)$	$M_{+-}(n-1)$
-1	+1	$\mathbf{P}_{-+}(n-1)$	$M_{-+}(n-1)$
-1	-1	$\mathbf{P}_{--}(n-1)$	$M_{--}(n-1)$

- Index solutions to the second subproblem by the bits that affect it

$b(n-2)$	$b(n-1)$	$b(n)$	μ
+1	+1	+1	$\mu(+++)$
+1	+1	-1	$\mu(++-)$
+1	-1	+1	$\mu(+--)$
+1	-1	-1	$\mu(+--)$
-1	+1	+1	$\mu(-++)$
-1	+1	-1	$\mu(-+-)$
-1	-1	+1	$\mu(--+)$
-1	-1	-1	$\mu(---)$

- We can add metrics of the two subproblems directly only if they have the same live bits.

- Unfortunately, the first subproblem has only $b(n-2)$, $b(n-1)$ – a table of four rows.

- Bookkeeping solution: replicate those table entries for alternative values of $b(n)$, since they don't depend on that bit. The result:

$b(n-2)$	$b(n-1)$	$b(n)$	$M(n-1)$	μ	$M(n)$
+1	+1	+1	$M_{++}(n-1)$	$\mu(+++)$	M_{+++}
+1	+1	-1	$M_{++}(n-1)$	$\mu(++-)$	M_{++-}
+1	-1	+1	$M_{+-}(n-1)$	$\mu(+--)$	M_{+--}
+1	-1	-1	$M_{+-}(n-1)$	$\mu(+--)$	M_{+--}
-1	+1	+1	$M_{-+}(n-1)$	$\mu(-++)$	M_{-++}
-1	+1	-1	$M_{-+}(n-1)$	$\mu(-+-)$	M_{-+-}
-1	-1	+1	$M_{--}(n-1)$	$\mu(--+)$	M_{--+}
-1	-1	-1	$M_{--}(n-1)$	$\mu(---)$	M_{---}

That was the merge step – combining the subproblems.

- Table size – a measure of complexity – is exponential in the number of bits that affect the measurement.

- Not all bits remain live after processing the measurement. Bit $b(n-2)$ has no effect on future measurements, so we can select the better of those metrics which differ only in that bit; *e.g.*, M_{+++} and M_{-++} . This gives a tentative decision; add it to the survivor path of the live bits.

This reduces it to a subproblem-1 type of table

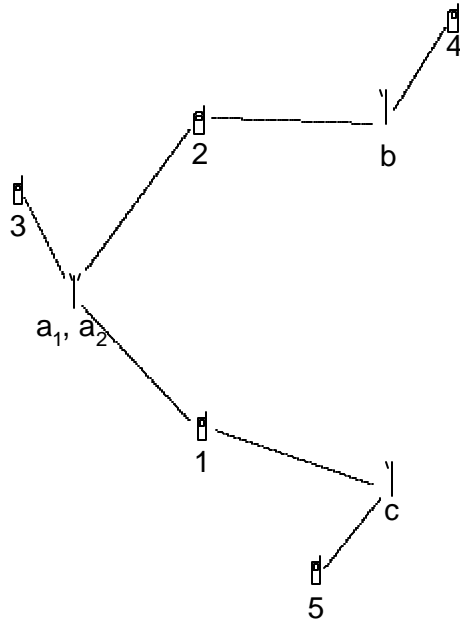
$b(n-1)$	$b(n)$	$\mathbf{P}(n)$	$M(n-)$
+1	+1	$\mathbf{P}_{++}(n)$	$M_{++}(n)$
+1	-1	$\mathbf{P}_{+-}(n)$	$M_{+-}(n)$
-1	+1	$\mathbf{P}_{-+}(n)$	$M_{-+}(n)$
-1	-1	$\mathbf{P}_{--}(n)$	$M_{--}(n)$

and the recursion step is complete.

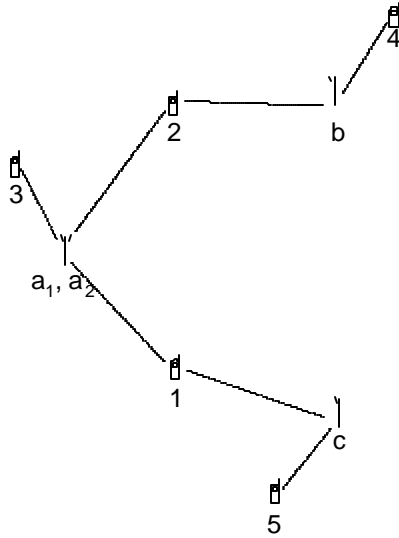
- We have now looked at the temporal Viterbi Algorithm in two closely related ways, and we are ready for a step into space.

4. MUD in Space – the CMM Algorithm

- Our original target was ML detection of users scattered across many cells, like this simplified macrodiversity/microdiversity example.



- For simplicity, assume synchronous transmissions, no echoes, so $\mathbf{y} = \mathbf{C}\mathbf{b} + \mathbf{n}$ is only M antennas x K users – a purely spatial problem. We would still rather not solve for \mathbf{b} by enumeration over its 2^K possibilities.



- Again, we can obtain the overall metric by processing measurements in sequence

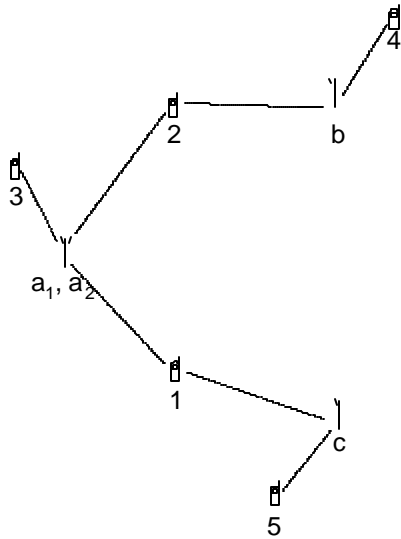
$$\|\mathbf{y} - \mathbf{Cb}\|^2 = \sum_m |y_m - \mathbf{c}_m \mathbf{b}|^2$$

but the gains matrix is not well structured for direct application of the Viterbi algorithm:

$$\mathbf{C} = \begin{pmatrix} \mathbf{X} & \mathbf{X} & \mathbf{X} & 0 & 0 \\ \mathbf{X} & \mathbf{X} & \mathbf{X} & 0 & 0 \\ 0 & \mathbf{X} & 0 & \mathbf{X} & 0 \\ \mathbf{X} & 0 & 0 & 0 & \mathbf{X} \end{pmatrix}$$

- It's time for the *conditional metric merge (CMM) algorithm*.

- Dynamic programming view:
 - ⇒ The problem of calculating the accumulated metrics $M(m)$ up to antenna m is a combination of two subproblems: accumulated metrics $M(m-1)$ up to antenna $m-1$ and the metrics $\mu(m)$ at antenna m .
 - ⇒ Continue to antenna M and it is done.
- An example will show how it works.



- Start with the measurement at antenna a_1 .

b_1	b_2	b_3	$M(a_1)$
+	+	+	$M_{+++}(a_1)$
+	+	-	$M_{++-}(a_1)$
+	-	+	$M_{+-+}(a_1)$
+	-	-	$M_{+--}(a_1)$
-	+	+	$M_{-++}(a_1)$
-	+	-	$M_{-+-}(a_1)$
-	-	+	$M_{-+-}(a_1)$
-	-	-	$M_{---}(a_1)$

All bits remain live, since they all affect the measurement at antenna a_2 , at least.

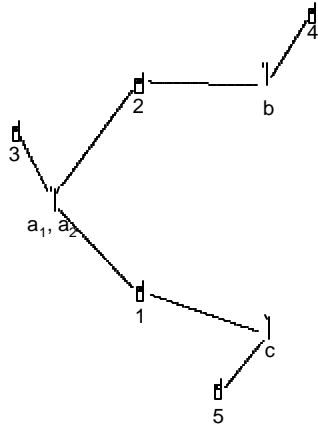
- The measurement at antenna a_2 involves exactly the same bits (microdiversity)

b_1	b_2	b_3	$\mathbf{m}(a_2)$
+	+	+	$\mathbf{m}_{+++}(a_2)$
+	+	-	$\mathbf{m}_{+-}(a_2)$
+	-	+	$\mathbf{m}_{-+}(a_2)$
+	-	-	$\mathbf{m}_{--}(a_2)$
-	+	+	$\mathbf{m}_{++}(a_2)$
-	+	-	$\mathbf{m}_{+-}(a_2)$
-	-	+	$\mathbf{m}_{-+}(a_2)$
-	-	-	$\mathbf{m}_{--}(a_2)$

so they can be added directly to accumulate:

b_1	b_2	b_3	$M(a_2)$
+	+	+	$M_{+++}(a_2)$
+	+	-	$M_{+-}(a_2)$
+	-	+	$M_{-+}(a_2)$
+	-	-	$M_{--}(a_2)$
-	+	+	$M_{++}(a_2)$
-	+	-	$M_{+-}(a_2)$
-	-	+	$M_{-+}(a_2)$
-	-	-	$M_{--}(a_2)$

Bit 3 is no longer live, so select better metric from pairs that differ in b_3 ; selected b_3 value is a survivor of that pair.



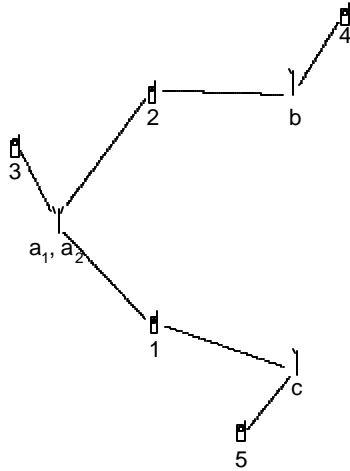
- The legacy from antenna a_2 is then

b_1	b_2	$M(a_2)$
+	+	$M_{++}(a_2)$
+	-	$M_{+-}(a_2)$
-	+	$M_{-+}(a_2)$
-	-	$M_{--}(a_2)$

- But the antenna b measurement depends on a different set of bits, so it can't be added directly.

b_2	b_4	$\mathbf{m}(b)$
+	+	$\mathbf{m}_{++}(b)$
+	-	$\mathbf{m}_{+-}(b)$
-	+	$\mathbf{m}_{-+}(b)$
-	-	$\mathbf{m}_{--}(b)$

- Replicate *both* tables: legacy metrics on b_4 , new metrics on b_1 , since they are each independent of these respective bits.

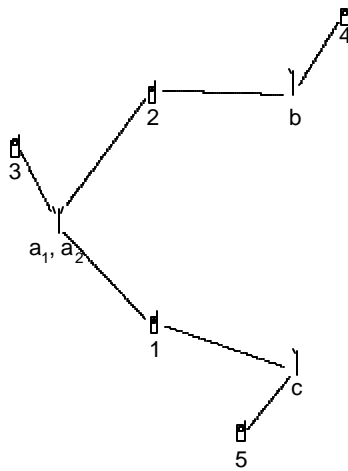


- The result of replication is

b_1	b_2	b_4	$M(a_2)$	$\mathbf{m}(b)$	$M(b)$
+	+	+	$M_{++}(a_2)$	$\mathbf{m}_+(b)$	$M_{+++}(b)$
+	+	-	$M_{++}(a_2)$	$\mathbf{m}_-(b)$	$M_{++}(b)$
+	-	+	$M_{+-}(a_2)$	$\mathbf{m}_+(b)$	$M_{+-}(b)$
+	-	-	$M_{+-}(a_2)$	$\mathbf{m}_-(b)$	$M_{+-}(b)$
-	+	+	$M_{-+}(a_2)$	$\mathbf{m}_+(b)$	$M_{-++}(b)$
-	+	-	$M_{-+}(a_2)$	$\mathbf{m}_-(b)$	$M_{-+}(b)$
-	-	+	$M_{--}(a_2)$	$\mathbf{m}_+(b)$	$M_{--}(b)$
-	-	-	$M_{--}(a_2)$	$\mathbf{m}_-(b)$	$M_{---}(b)$

Table size exponential in number of live bits.

- Only b_1 remains live. For each value of b_1 , choose the best metric from the four b_2, b_3 combinations, and the corresponding b_2, b_3 become survivors of that b_1 .

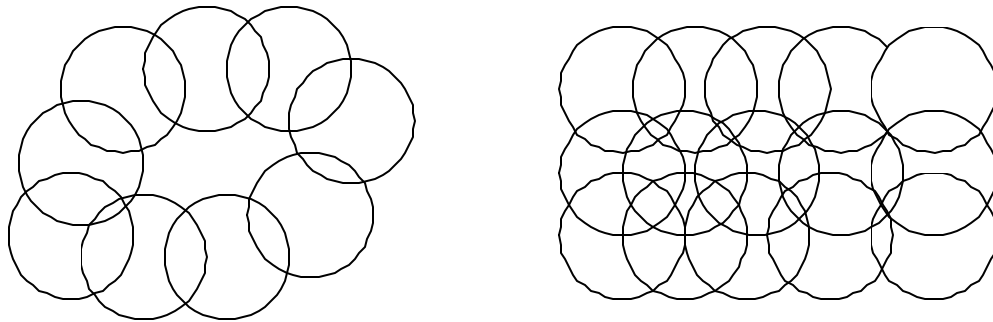


- The legacy from antenna b is just

b_1	$M(b)$
+	$M_+(b)$
-	$M_-(b)$

- Processing of the antenna c measurement is now clear:
 - \Rightarrow The metric $\mathbf{m}(c)$ depends only on b_1 and b_5 .
 - \Rightarrow To combine, replicate the legacy on b_5 , then add the legacy and new metric for each of the four combinations.
 - \Rightarrow No more measurements, so neither bit remains live. Select the b_1, b_5 combination with smallest metric and release those bits, along with the corresponding survivors from b_2, b_3, b_4 .

- What do we have, in this new Conditional Metric Merge (CMM) algorithm?
 - ⇒ Like the Viterbi algorithm, it exploits sparsity and locality in the gains matrix \mathbf{C} .
 - ⇒ It handles problems with less regular structure than does the VA.
 - ⇒ Like the VA, it is subject to exponential growth in the number of interfering bits at any measurement, but it keeps the exponent of that growth to a minimum.
 - ⇒ The actual load depends on the spatial arrangement and overlap of cells –an area of continuing investigation.



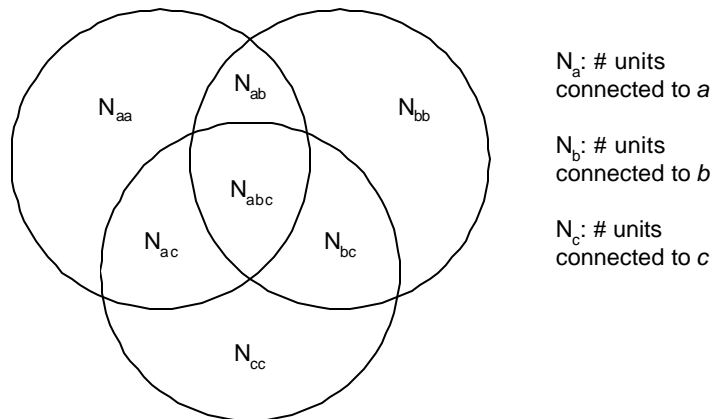
5. MUD in Space and Time

- Most (but not all) systems feature echoes and lack of synchronism among users, as well as spatial interference.
- However, each measurement $y_m(n)$ is affected by a small number of bits in temporal and spatial proximity. The CMM algorithm is a least-computation method of obtaining the optimum (ML) solution.
- CMM reduces to Viterbi for purely temporal problems.
- The nightmare: having to carry around many live bits from time $n-1$ as we traverse the cells at time n .
Possible solution: process measurements in other than strict time:antenna ordering – *i.e.*, reorder rows and columns of \mathbf{C} to move live bits into survivors as quickly as possible. A continuing investigation.

5. Summary

- The exceptional performance of ML motivated development of an algorithm to reduce the computational load while producing a true ML solution.
- Conditional Metric Merge (CMM) uses dynamic programming principles to handle decision-making from loosely structured space-time measurements.
- Further reduction in load by appropriate sequencing of measurements is an area of further investigation.

Evolution of computation:



- Antenna a

New metrics: $\exp_2(N_a)$

Live entries after selection: $\exp_2(N_{ab} + N_{abc} + N_{ac})$

- Antenna b

New metrics: $\exp_2(N_b)$

Table entries after combine: $\exp_2(N_b + N_{ac})$

Live entries after selection: $\exp_2(N_{ac} + N_{abc} + N_{bc})$

- Antenna c

New metrics: $\exp_2(N_c)$

Table entries after combine: $\exp_2(N_c)$

Live entries after selection: 0

