Proc. of 40th Allerton Conference on Communication, Control and Computing. pp. 48–57. Allerton House, Monticello, Illinois, October 2–October 4, 2001.

Unitary Constellations with Large Diversity Sum and Good Diversity Product*

Guangyue Han, Joachim Rosenthal
Department of Mathematics
University of Notre Dame
Notre Dame, IN 46556.
e-mail: Han.13@nd.edu, Rosenthal.1@nd.edu
URL: http://www.nd.edu/~eecoding/

October 2, 2002

Abstract

There exist two important design criterions for unitary space time codes. In the situation where the signal to noise ratio is large it is well known that the *diversity product* of a constellation should be as large as possible. It is less known that the diversity sum is a very important design criterion for codes working in a low SNR environment.

In this paper we introduce numerical methods which allow us to find codes with near optimal diversity sum. We also demonstrate how these numerical techniques lead to codes with excellent diversity product(sum).

1 Introduction and Model

One way to acquire high transmission rate on a wireless channel is to use multiple transmit or receive antennas. Either because of rapid changes in the channel parameters or because of limited system resources, it is reasonable to assume that both the transmitter and the receiver don't know about the channel state information.

In [7], Hochwald and Marzetta study unitary space-time modulations. Consider a wireless communication system with M transmit antennas and N receive antennas operating in a Rayleigh flat-fading channel. We assume time is discrete and at each time slot, signals are transmitted simultaneously from the M transmitter antennas. We can further assume that the wireless channel is quasi-static over a time block of length T.

Consider a constellation $\mathcal{V} := \{\Phi_1, \dots, \Phi_L\}$ consiting of L matrices having size $T \times M$ which satisfy $T \geq M$ and $\Phi_k^* \Phi_k = I_M$, the last matrix denoting the $M \times M$ identity matrix. The last equation simply says that the colums of Φ_k form a 'unitary frame', i.e. the columns have length 1 and are pairwise orthogonal. The matrices $\sqrt{T}\Phi_k$, $k = 1, 2, \cdots, L$, represent the code words used in the transmission.

Denote by $|| \ ||_F$ the Frobenius norm of a matrix. If the channel is unknown it has been shown in [7] that the maximum likelihood (ML) decoder for such a code is given by:

$$\Phi_{ML} = \arg \max_{R \in \{\Phi_1, \Phi_2, \cdots, \Phi_L\}} ||R^* \Phi_l||_F$$

Let $\delta_m(\Phi_l^* \Phi_{l'})$ be the m-th singular value of $\Phi_l^* \Phi_{l'}$ and denote by ρ the signal to noise ratio. It has also been shown in [7] that the pairwise probability of mistaking Φ_l for $\Phi_{l'}$ using Maximum Likelihood Decoding is bounded by Chernoff's bound:

*Both authors were supported in part by NSF grant DMS-00-72383. The first author was also supported by a fellowship from the Center of Applied Mathematics at the University of Notre Dame.

$$P_{\Phi_l,\Phi_{l'}} \leq \frac{1}{2}\prod_{m=1}^{M}\left[1+\frac{(\rho T/M)^2}{4(1+\rho T/M)}(1-\delta_m^2(\Phi_l^*\Phi_{l'}))\right]^{-N}. \tag{1.1}$$

In this paper we will be concerned with the numerical construction of constellations where the right hand side in (1.1), maximized over all pairs l, l' is as small as possible for fixed numbers of T, M, L. For this purpose we define a *diversity function* dependent on the constellation $\mathcal{V}$ and the SNR ρ through:

$$\mathcal{D}(\mathcal{V},\rho) := \max_{l\neq l'}\frac{1}{2}\prod_{m=1}^{M}\left[1+\frac{(\rho T/M)^2}{4(1+\rho T/M)}(1-\delta_m^2(\Phi_l^*\Phi_{l'}))\right]^{-N}. \tag{1.2}$$

In the next two sections we will study the limiting behavior of $\mathcal{D}(\mathcal{V},\rho)$ as ρ goes to zero and to infinity.

2 Design criterion for high SNR

When the SNR ρ is very large then $\mathcal{D}(\mathcal{V},\rho)$ can be aprroximated via:

$$\mathcal{D}(\mathcal{V},\rho) \simeq \max_{l\neq l'}\frac{1}{2}\left(\frac{(\rho T/M)^2}{4(1+\rho T/M)}\right)^{-NM}\prod_{m=1}^{M}\frac{1}{(1-\delta_m^2(\Phi_l^*\Phi_{l'}))^N}. \tag{2.3}$$

It is the design objective to construct a constellation $\Phi_1, \Phi_2, \cdots, \Phi_n$ such that

$$\min_{l\neq l'}\prod_{m=1}^{M}(1-\delta_m^2(\Phi_l^*\Phi_{l'}))$$

is as large as possible. This last expression defines in essence the diversity product. In order to compare different dimensional constellations it is customary to use the definition:

Definition 2.1. (See [6]) The *diversity product* of a unitary constellation $\mathcal{V}$ is defined as

$$\prod\mathcal{V} = \min_{l\neq l'}\left(\prod_{m=1}^{M}(1-\delta_m(\Phi_l^*\Phi_{l'})^2)\right)^{\frac{1}{2M}}.$$

An important special case occurs when $T = 2M$. In this situation it is customary to reprsent all unitary matrices Φ_k in the form:

$$\Phi_k = \frac{\sqrt{2}}{2}\begin{pmatrix} I \\ \Psi_k \end{pmatrix}. \tag{2.4}$$

Note that by definition of Φ_k the matrix Ψ_k is a $M\times M$ unitary matrix. The diversity product as defined in Definition 2.1 has then a nice form in terms of the unitary matrices. For this let λ_m be the mth eigenvalue of a matrix. Then we have:

$$1-\delta_m^2(\Phi_{l'}^*\Phi_l) = \frac{1}{4}\lambda_m(2I_M-\Phi_l^*\Phi_{l'}-\Phi_{l'}^*\Phi_l) = \frac{1}{4}\delta_m^2(I_M-\Psi_{l'}^*\Psi_l) = \frac{1}{4}\delta_m^2(\Psi_{l'}-\Psi_l).$$

So we have

$$\prod_{m=1}^{M}(1-\delta_m^2(\Phi_{l'}^*\Phi_l))^{\frac{1}{2M}} = \frac{1}{2}\prod_{m=1}^{M}\delta_m(\Phi_{l'}-\Phi_l)^{\frac{1}{M}} = \frac{1}{2}|\det(\Psi_{l'}-\Psi_l)|^{\frac{1}{M}}.$$

When $T = 2M$ and the constellation $\mathcal{V}$ is defined as above, then the formula of the diversity product assumes the simple form:

$$\prod \mathcal{V} = \frac{1}{2} \min_{0 \leq l < l' \leq L} |\det(\Psi_l - \Psi_{l'})|^{\frac{1}{M}} \tag{2.5}$$

We call a constellation $\mathcal{V}$ a fully diverse constellation if $\prod \mathcal{V} > 0$.

A lot of efforts have been taken to construct constellations with large diversity product. (See e.g. [6, 8, 4, 14, 13]).

3 Design criterion for low SNR channel

When ρ is very small, the $\mathcal{D}(\mathcal{V}, \rho)$ can be approximated via:

$$\mathcal{D}(\mathcal{V}, \rho) \simeq \max_{l \neq l'} \frac{1}{2} \prod_{m=1}^{M} \frac{1}{(1 + (M - \sum_{m=1}^{M} \delta_m^2(\Phi_l^* \Phi_{l'})) \frac{(\rho T/M)^2}{4(1+\rho T/M)})^N}. \tag{3.6}$$

It is design objective to construct a constellation $\Phi_1, \Phi_2, \cdots, \Phi_L$ such that

$$\min_{l \neq l'} (M - \sum_{m=1}^{M} \sigma_m^2(\Phi_l^* \Phi_{l'}))$$

is as large as possible. In order to compare the constellation of different dimension, we have the definition of diversity sum.

Definition 3.1. The *diversity sum* of a unitary constellation $\mathcal{V}$ is defined as

$$\sum \mathcal{V} = \min_{l \neq l'} \sqrt{1 - \frac{\|\Phi_l^* \Phi_{l'}\|_F^2}{M}}.$$

Again one has the important special case where $T = 2M$ and the matrices Φ_k take the special form (2.4). In this case one verifies that

$$\begin{aligned}
\|\Phi_{l'} \Phi_l\|_F^2 = \frac{1}{4}\|I + \Psi_{l'} \Psi_l\|_F^2 = \frac{1}{4}\mathrm{tr}\,((I + \Psi_{l'}^* \Psi_l)(I + \Psi_l^* \Psi_{l'}) \\
= \frac{1}{4}\mathrm{tr}\,(2I + \Psi_{l'}^* \Psi_l + \Psi_l^* \Psi_{l'}) = \frac{1}{4}(4M - (2M - \mathrm{tr}\,(\Psi_{l'}^* \Psi_l + \Psi_l^* \Psi_{l'}))) \\
= \frac{1}{4}(4M - \mathrm{tr}\,((\Psi_l - \Psi_{l'})^* (\Psi_l - \Psi_{l'}))) = \frac{1}{4}(4M - \|\Psi_{l'} - \Psi_l\|_F^2)
\end{aligned}$$

For the form (2.4) the diversity sum assumes the following simple form:

$$\sum \mathcal{V} = \min_{l,l'} \frac{1}{2\sqrt{M}} \|\Psi_l^* - \Psi_{l'}\|_F.$$

4 Low SNR Criterion versus High SNR Criterion

So far, we have already had two criterions for designing good performing codes. In fact, the strategy to have low error rate is to lower the upper bound of it in the case of SNR going to ∞ or 0. So that doesn't necessarily mean that the resulted codes would perform well even if they have a very high diversity product or diversity sum.

In the following we are comparing three constellations derived by different methods:

Orthogonal Design: This constellation has been considered by several authors [2, 14]. For our purpose we simply define this code through:

$$\left\{ \frac{\sqrt{2}}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ e^{\frac{2m\pi i}{11}} & e^{\frac{2n\pi i}{11}} \\ -e^{-\frac{2n\pi i}{11}} & e^{-\frac{2m\pi i}{11}} \end{pmatrix} | m, n = 0, 1, \cdots, 10 \right\}.$$

Numerically Derived Constellation: Using simulated annealing we found after short computation a constellation with near perfect diversity sum:

$$\{A^k B^l | A = \begin{pmatrix} -0.9049 + 0.3265i & 0.1635 + 0.2188i \\ 0.0364 + 0.2707i & -0.8748 + 0.4002i \end{pmatrix},$$

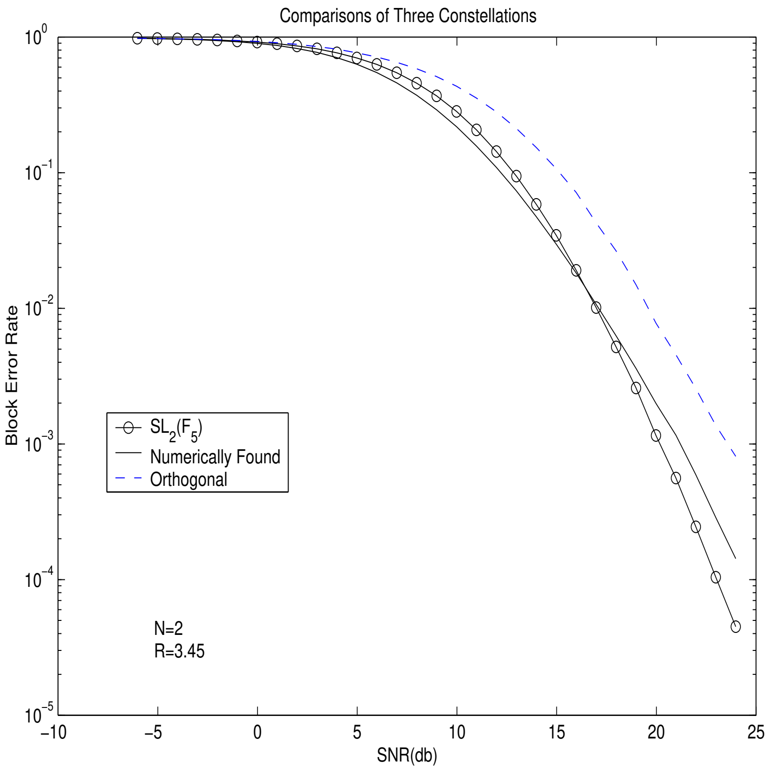
$$B = \begin{pmatrix} -0.1596 + 0.9767i & -0.1038 + 0.0994i \\ 0.0833 - 0.1171i & -0.9432 + 0.2995i \end{pmatrix}, k, l = 0, 1, \cdots, 10\}.$$

$Sl_2(\mathbb{F}_5)$: Shokrollahi e. a. [14] derived a constellation using the theory of fixed point free representations whose diversity product is near optimal. See above reference for detail.

The parameters of the three constellations are summarized by:

	Orthogonal design	Numerically derived	$Sl_2(\mathbb{F}_5)$
Number of elements	121	121	120
diversity sum	0.1992	0.3886	0.309
diversity product	0.1992	0.0278	0.309

We simulated all three codes and the following gives the result:



Note that the numerically designed code who has a very bad diversity product is performing very well nevertheless due to the exceptional diversity sum.

To design codes which have both high diversity product and large diversity sum, one way is to construct it is algebraically. A good candidate is a subset of $SU(2)$ since for such sets the diversity product and the diversity sum are the same. For this let A, C be two elements in $SU(2)$:

$$A = \begin{pmatrix} a & b \\ -\bar{b} & \bar{a} \end{pmatrix}, \quad C = \begin{pmatrix} c & d \\ -\bar{d} & \bar{c} \end{pmatrix}$$

One verifies that: $\det(A - C) = |a - c|^2 + |b - d|^2$ and that $\|A - C\|_F^2 = 2(|a - c|^2 + |b - d|^2)$.

By definition of the diversity sum/product is follows that $\prod \mathcal{V} = \sum \mathcal{V}$ for constellations inside $SU(2)$. Of course this is an exceptional situation.

In general we do not know of a good algebraic way which optimizes simultaneously the diversity product and the diversity sum.

One way to achieve this goal is via numerics and we devote the rest of the paper to explain this.

5 Non-Group Constellation with Algebraic Structure

$U(n)$ is a n^2 dimensional Lie group, so when we consider a constellation as a subset of $U(n)$ with m elements, we are dealing with a mn^2 dimensional searching problem. When m or n goes a little bigger, Brute Force searching algorithm is already not reasonable even for the most powerful computer.

Consider a general non-group constellation:

$$\mathcal{V} = \{\Psi_1, \Psi_2, \cdots, \Psi_n\},$$

in order to calculate the diversity product(sum), one needs to do $\frac{n(n-1)}{2}$ calculations: $|\det(v_i - v_j)|$, for every pair of different i, j.

In case of a group constellation, to calculate the diversity product(sum), one only need to calculate the "distance" between every non-identity element and identity. The reason can be shown by the following:

Pick two elements v_i, v_j in $\mathcal{V}$, we have

$$|\det(v_i - v_j)| = |\det(v_i)\det(I - v_i^* v_j)| = |\det(I - v_i^* v_j)|,$$

and apparently, $v_i^* v_j$ is still in the group. That really means we only have to maximize n "target". That could be the reason some group constellation works extremely well. However in most cases, group constellations are not good performing because very strict algebraic constraints are required in the group constellation.

We are going to present some constellations in between general non-group constellations and group constellations. We do have some algebraic structure in the constellations which will avoid "dimension explosion disaster" for computation, also we try to use more loose structure which also reduce the number of target.

We list some constellations with "good" algebraic structures as follows:

1.$\mathcal{V} = \{A^k B^l | A, B \in U(n), k = 0, \cdots, p, l = 0, \cdots, q\}$

2.$\mathcal{V} = \{A^k B^l C^m | A, B, C \in U(n), k = 0, \cdots, p, l = 0, \cdots, q, m = 0, \cdots, r\}$

3. $I, A, AB, ABA, ABAB, ABABA, \cdots$

4. $I, A, AB, ABC, ABCA, ABCAB, ABCABC, ABCABCA, \cdots$

5. $I, AB, A^2B^2, A^3B^3, \cdots$

6. $I, ABC, A^2B^2C^2, A^3B^3C^3, \cdots$

Apparently, the above similar ideas can be applied to constellations with m generators.And inspired by this, we can consider the constellations to be the product of the above constellations. For instance,

7. Let $S_1 = \{I, C, C^2, C^3, \cdots\}$ and $S_2 = \{I, A, AB, ABA, \cdots\}$, we consider the constellation to be

$$S = S_1 \times S_2 = \{s_1 s_2 | s_1 \in S_1, s_2 \in S_2\}$$

8.Let $S_1 = \{I, A, AB, ABA, \cdots\}$ and $S_2 = \{I, C, CD, DCD, CDCD, \cdots\}$, we consider the constellation to be

$$S = S_1 \times S_2 = \{s_1 s_2 | s_1 \in S_1, s_2 \in S_2\}$$

One can verify that in the low SNR scenario, with the above constellation for the low SNR design criterion, we still have the same number of targets. In fact, it can be shown it is even suitable if Chernoff bound(related to SNR) is chosen to be design criterion.

6 2 Dimensional Constellation Design

We are going to talk about 2 dimensional constellation design separately because unlike higher dimensional ones, we have very explicit representation of $U(2)$:

$$U(2) = \left\{ \begin{pmatrix} a & be^{i\theta} \\ -\bar{b} & \bar{a}e^{i\theta} \end{pmatrix} ||a|^2 + |b|^2 = 1, \qquad 0 \leq \theta \leq 2\pi \right\}$$

From above, one can see $U(2)$ is 4 dimensional, so consider the constellation with small number of generators, it is possible to use Brute Force method to calculate the almost optimal generator. Of course, in this process, more finer grid and more computing power will make better result.

Still we have other methods. The common technique used in optimization with no constraints is iteratively finding better points. Such methods includes Steepest Descent Method, Newton Method and Conjugate Gradient Method [10]. But these methods only work for one target function and they assume that certain differential conditions. Because we are usually dealing with a large number of target non-differential functions, we find Simulated Annealing method and Genetic Algorithm [12] is more suitable for this kind of job.

Comparisons of different methods

	size	diversity product	structure
Simulated Annealing	125	0.2127	$A^k B^l C^m$
Simulated Annealing	120	0.2202	$A^k B^l$
Simulated Annealing	121	0.2417	$A^k B^l$
Brute Force	120	0.1914	$A^k B^l$
Genetic Algorithm	120	0.2377	N/A

SA with different structure(computation less than 3 minutes)

structure/size	36	49	64	256	400	900	10000
$A^k B^l$	0.3860	0.3781	0.2742	0.1025	0.0866	0.0834	0.0158
AB	0.3205	0.2659	0.2450	0.1030	0.0800	0.0579	0.0122
$A^k B^k$	0.3769	0.3502	0.3090	0.1651	0.1342	0.0820	0.0187

structure/size	27	64	216	343	512	729	9261
$A^k B^l C^m$	0.3418	0.2616	0.1833	0.1401	0.0632	0.1012	0.0031
ABC	0.3299	0.1832	0.1033	0.0725	0.0555	0.0430	N/A
$A^k B^k C^k$	0.4122	0.2512	0.0583	0.0206	0.0087	0.0039	N/A

The above tables show some of the numerical results in 2 dimension. One thing is needed to justify that some of the codes here are not the best code, some of them are even far from the best codes. For instance, $SL_2(\mathbb{F}_5)$ group code with 120 elements in [14], it has diversity product 0.309 and is much better than the results we are showing here. That doesn't mean SA simply doesn't work for searching diversity product. When it comes to higher dimension, it seems to be too complicated to construct it in algebraic way, numerical methods begin to show its power.

Comparison of different methods

	size	diversity sum	structure
Simulated Annealing	121	0.3919	A^kB^l
Simulated Annealing	120	0.3696	A^kB^l
Simulated Annealing	121	0.3886	A^kB^l
Brute Force	120	0.3673	A^kB^l
Genetic Algorithm	120	0.3867	N/A

SA with different structure(computation less than 3 minutes)

structure/size	36	49	64	256	400	900	10000
A^kB^l	0.5113	0.4733	0.4474	0.2875	0.2504	0.1848	0.0785
AB	0.5530	0.4240	0.3821	0.1994	0.1629	0.1064	0.0310
A^kB^k	0.5466	0.5121	0.4735	0.3088	0.2637	0.2047	0.0869

structure/size	27	64	216	343	512	729	9261
$A^kB^lC^m$	0.5400	0.4210	0.2992	0.2663	0.2099	0.2060	0.0772
ABC	0.5382	0.4497	0.2614	0.2065	0.1695	0.1447	0.0398
$A^kB^kC^k$	0.5630	0.4271	0.2864	0.2198	0.1969	0.1423	N/A

Numerical method is extremely good for designing constellation with large diversity sum, most of the results are the best or very close to the best codes. One thing is very interesting is the $SL_2(\mathbb{F}_5)$ group code also has diversity sum 0.309, we believe that would be the reason its performance is so good. Again, as for higher dimension and large cardinality, it is even more powerful.

In fact, we did most of our experiments in 2 dimensional case. For diversity product, SA and GA works very well and certainly one can find the same good or much better constellation with the same number of elements. While for diversity sum, it seems like the numerical method is exactly the right method for this aim. Most of the cases, the program produce the best codes ever known. Even the cardinality of the constellation comes to be very huge, like 10000, SA with algebraic structure works even very well, sometimes the program just produce the best codes in several minutes.

While for the Genetic Algorithm, when the cardinality of the constellation is relatively small $(3-100)$, it works extremely well. One can check the output of the GA within 3 minutes is almost the very optimal theoretical result when cardinality is within 10. This makes us believe the higher dimension search result is also very good although we don't really know the upper bound. Reader can see some numerical tables in the following discussion.

7 Constellation Design for Any Dimension

If we use the above constellation with nice algebraic structure, it is possible to use Brute Force algorithm to find some good constellation. What about higher dimensional constellations? Can we represent a higher dimensional matrix in a good way?

There are several ways to represent a unitary matrix. But most of them makes the problem very complicated although they do work. While Cayley transformation is one effective way to represent unitary matrices (or more precisely, a subset of unitary matrices). For more details about Cayley transformation, we refer to [5].

It is very important to have Cayley transformation in numerically designing the constellation because it makes the local topology of $U(n)$ clear. One can see that most optimization method require us to consider the neighborhood of one elements in $U(n)$.

7.1 Simulated Annealing Algorithm

Simulated Annealing is a method which mimics the process of melted metal getting cooled off. In the annealing process of the melted metal, first the metal is heated to melt, then the temperature is getting down gradually. The metal will get to a minimized energy state if the temperature is lowering slow enough. For more details about this algorithm, we refer to [1, 15, 11].

In fact, we would rather call it a general method instead of a concrete algorithm. Generally speaking, for a given optimization problem, we always take a initial solution in some certain way. Then consider the "neighborhood" of this solution, we will accept the solution in the "neighborhood" according to some predefined criterion which might include a probability threshold.

As we discuss in the 2 dimensional case, SA is also a method which will find "better" point in a iterative way. And with Cayley transformation which is a good representation of higher dimensional unitary matrix, one can still apply this method to higher dimensional constellation design.

Newton's Methods requires that we should choose a good initial guess, otherwise it may not even converges to the real solution. The algorithm we are using also depends on the initial guess. In 2 dimensional constellation search, one can use the Brute Force with coarser grid to find a good guess, then use SA to iteratively find better point. But when it comes to higher dimension, it is not plausible to use the this anymore with limited computing power, one way to find a good initial guess is to use the existing group constellation. In the sequel, we use the group constellation $G_{21,4}$ in [14]:

$$\mathcal{V}_1 = \{A^k B^l | A = \begin{pmatrix} \eta & 0 & 0 \\ 0 & \eta^4 & 0 \\ 0 & 0 & \eta^{16} \end{pmatrix}, B = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \eta^7 & 0 & 0 \end{pmatrix}, k = 0, 1, \cdots, 20, l = 0, 1, 2\}$$

One can verify that

$$\prod \mathcal{V}_1 = 0.3851$$

It seems like the $G_{21,4}$ is already very good constellation, our algorithm only improves a little, while one can check most of the case, the algorithm will improve much compared to the original group constellation.

$$\mathcal{V}_2 = \{A^k B^l | k = 0, 1, \cdots, 20, l = 0, 1, 2\},$$

where

$$A = \begin{pmatrix} 0.9415 + 0.3155i & 0.0573 - 0.0222i & 0.0496 + 0.0882i \\ 0.0160 - 0.0555i & 0.4005 + 0.9136i & 0.0326 - 0.0212i \\ 0.0579 + 0.0855i & -0.0312 - 0.0099i & 0.1384 - 0.9844i \end{pmatrix},$$

$$B = \begin{pmatrix} 0.0175 + 0.0095i & 0.9997 + 0.0111i & 0.0079 + 0.0042i \\ 0.0086 + 0.0100i & -0.0082 + 0.0040i & 0.9999 + 0.0036i \\ -0.4836 + 0.8750i & 0.0004 - 0.0198i & -0.0045 - 0.0126i \end{pmatrix}.$$

One verifies that

$$\prod \mathcal{V}_2 = 0.3874.$$

We have the following tables to show the results of some experiments. The experiments are based on the computation on a Intel Pentium 800MHz PC and no computation lasts longer than 3 minutes.

Results on Diversity Product(SA)

number of elements	dim=2	dim=3	dim=4	dim=5
4	0.7071	0.7657	0.7388	0.6768
9	0.5701	0.5754	0.4774	0.4259
16	0.4018	0.4574	0.4651	0.3877
25	0.3443	0.3834	0.3809	0.3467
36	0.2865	0.3450	0.3501	0.3760

Results on Diversity Sum(SA)

number of elements	dim=2	dim=3	dim=4	dim=5
4	0.8147	0.8160	0.7861	0.7377
9	0.6956	0.6861	0.6539	0.6389
16	0.5908	0.6459	0.6288	0.5916
25	0.5618	0.6268	0.6190	0.5795
36	0.5286	0.6054	0.6148	0.5853

From above table, SA works fine when the cardinality of the constellation is small. When it comes to large number of constellation, SA works even better. We try SA on 10000 elements constellation using $A^k B^l$ structure, we can get diversity sum 0.1000, while for construction in [4] , we can only get diversity sum 0.0654 with 8433 elements or diversity sum 0.0604 with 10770 elements. Also SA's results on higher dimension showing the pretty much the same size with 2 dimension which makes us believe it will also work in higher dimension, although we don't have many comparisons available.

7.2 Genetic Algorithm

Genetic Algorithm is a optimization algorithm emulating the species evolutionary model by J.H. Holland. GA doesn't assume much specific information about the given problem, first a group of candidates are selected in some certain way and they are encoded using binary coding in most cases, then consider the offsprings of the candidates(the mating processing is also defined with respect to specific problem), also there might be new random candidates adding in. The whole idea is "better survive", which really means some candidates with higher "survive index" keep staying in the group and other "worse" ones will get discarded. The algorithm will stop if some threshold is reached after certain number of iterations. One can further consult [3, 9].

In our implementation of Genetic Algorithm, we use the simplest one. First for a fixed number of constellation searching, we use some predefined initial points or just simply generate some random points, then we add more points randomly to the old constellation, then use the "survive index" to determine which ones are "better", then on some probability, we determine to keep the "better" ones and discard the "worse" ones.

We have the following tables to show the results of some experiments. The experiments are based on the computation on a Intel Pentium 800MHz PC and no computation lasts longer than 3 minutes.

Results on Diversity Product(GA)

number of elements	dim=2	optimal DP	dim=3	dim=4	dim=5
3	0.8644	$\frac{\sqrt{3}}{2}$=0.8860	0.8264	0.7305	0.6737
4	0.8051	$\sqrt{\frac{2}{3}}$=0.8165	0.7343	0.6521	0.6305
6	0.6924	unknown	0.6632	0.6154	0.5721
10	0.5768	unknown	0.5497	0.5742	0.4942

Results on Diversity Sum(GA)

number of elements	dim=2	optimal DS	dim=3	dim=4	dim=5
3	0.8601	$\frac{\sqrt{3}}{2}$=0.8860	0.8331	0.8118	0.7798
4	0.8029	$\sqrt{\frac{2}{3}}$=0.8165	0.7802	0.7757	0.7492
6	0.7443	$\sqrt{\frac{3}{5}}$=0.7746	0.7502	0.7293	0.7176
10	0.6826	$\frac{\sqrt{2}}{2}$=(0.7071)	0.6981	0.6920	0.6817

References

[1] E. H. L. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines.* Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Ltd., Chichester, 1989. A stochastic approach to combinatorial optimization and neural computing.

[2] S. M. Alamouti. A simple transmitter diversity scheme for wireless communications. *IEEE J. Selected Areas of Commun.*, pages 1451–1458, October 1998.

[3] D. A. Coley. *An introduction to genetic algorithms for scientists and engineers.* Singapore; River Edge, NJ: World Scientific, 1999.

[4] G. Han and J. Rosenthal. Unitary constellation design and its application to space-time coding. In D. Gilliam and J. Rosenthal, editors, *Proceedings of the 15-th International Symposium on the Mathematical Theory of Networks and Systems*, University of Notre Dame, August 2002.

[5] B. Hassibi and B. M. Hochwald. Cayley differential unitary space-time codes. *IEEE Trans. Inform. Theory*, 48(6):1485–1503, 2002. Special issue on Shannon theory: perspective, trends, and applications.

[6] B. Hochwald and W. Sweldens. Differential unitary space-time modulation. *IEEE Trans. Comm.*, pages 2041–2052, December 2000.

[7] B. M. Hochwald and T. L. Marzetta. Unitary space-time modulation for multiple-antenna communications in Rayleigh flat fading. *IEEE Trans. Inform. Theory*, 46(2):543–564, 2000.

[8] X.-B. Liang and X.-G. Xia. Unitary signal constellations for differential space-time modulation with two transmit antennas: Parametric codes, optimal designs and bounds. *IEEE Trans. Inform. Theory*, 48(8):2291–2322, August 2002.

[9] K. F. Man, K. S. Tang, and S. Kwong. *Genetic algorithms.* Advanced Textbooks in Control and Signal Processing. Springer-Verlag London Ltd., London, 1999. Concepts and designs, With 1 IBM-PC floppy disk (3.5 inch; HD).

[10] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer Series in Operations Research. Springer-Verlag, New York, 1999.

[11] R. H. J. M. Otten and L. P. P. P. van Ginneken. *The Annealing Algorithm.* The Kluwer International Series in Engineering and Computer Science. VLSI, Computer Architecture and Digital Signal Processing. Kluwer Academic Publishers, Boston, MA, 1989.

[12] D. T. Pham and D. Karaboga. *Intelligent Optimisation Techniques.* Springer-Verlag London Ltd., London, 2000. Genetic algorithms, tabu search, simulated annealing and neural networks.

[13] A. Shokrollahi. Computing the performance of unitary space-time group codes from their character table. *IEEE Trans. Inform. Theory*, 48(6):1355–1371, 2002. Special issue on Shannon theory: perspective, trends, and applications.

[14] A. Shokrollahi, B. Hassibi, B. M. Hochwald, and W. Sweldens. Representation theory for high-rate multiple-antenna code design. *IEEE Trans. Inform. Theory*, 47(6):2335–2367, 2001.

[15] P. J. M. van Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications*, volume 37 of *Mathematics and its Applications.* D. Reidel Publishing Co., Dordrecht, 1987.