

```

1 function [ SumCapacity, SelectedReceiveAntenna, SelectedUser, DataStreams ]
SuboptimalAlgorithm1Final( NumOfTransmitAntennas, NumOfReceiveAntennasPerUser, VarianceSq,
NumOfUsers, SNRindB)
2
3 % Delcaration
4
5 Nt = NumOfTransmitAntennas;
6 Nr = NumOfReceiveAntennasPerUser;
7 v = VarianceSq;
8 k = NumOfUsers;
9 SNR = power(10, SNRindB/10);
10 Ebs = SNR * v;
11 rx = zeros(1, k*Nr);
12 user = zeros(1, k*Nr);
13 for i = 1:(k*Nr)
14     rx(i) = i;
15     user(i) = floor((i-1)/Nr) + 1;
16 end
17
18 % Mapping of user and receive antennas
19
20 UserId = containers.Map(rx, user);
21
22 % generating full channel matrix
23
24 Hf = sqrt(1/2)*randn(Nr, Nt, k) + sqrt(1/2)*randn(Nr, Nt, k)*1i;
25
26 % Initialization
27
28 S=[];
29 U=[];
30 H_tilda=zeros(Nr, Nt);
31 Cmax=0;
32 L=zeros(1, k);
33 ltmp = zeros(1, k);
34 flag=1;
35 phase=1;
36
37 % Algorithm Starts
38
39 while flag == 1
40     if sum(L) < Nt
41         Cr = zeros(1, Nr*k);
42         H = zeros(Nr, Nt, k);
43         for r = rx
44             ltmp = L;
45             Stmp = union(S, r);
46             u = UserId(r);
47             r_id = r - ((u-1)*Nr);
48             Utmp = union(U, u);
49             H(r_id, :, u) = Hf(r_id, :, u);

```

```

50
51     % Increasing the Data Steams for each user u for r
52
53     if phase == 1
54         ltmp(u) = ltmp(u) + 1;
55     end
56
57     % Total Data streams
58
59     Ltmp = sum(ltmp);
60     C = zeros(1,k);
61     Hz = zeros(Nr,Nt);
62     for j = Utmp
63         Hz = H(:, :, j);
64         Hz( ~any(Hz,2), : ) = [];
65         H1 = Hz' * Hz ;
66         H2 = H_tilda' * H_tilda;
67         Mj = size(Hz,1);
68         Ej = (Ebs * ltmp(j)) / Ltmp ;
69
70         % Precoding matrix calculation for user j
71
72         Wj = eig(H1 , (Mj*v/Ej)*eye(size(H1,1)) + H2);
73         Wj_tilda = zeros(Nt,ltmp(j));
74         Hz1 = zeros(Nr,Nt);
75         for user1 = Utmp
76
77             % Calculating the tilda of Precoding Matrix for user j
78
79             if user1 ~= j
80                 Hz1 = H(:, :, user1);
81                 Hz1( ~any(Hz1,2), : ) = [];
82                 H11 = Hz1' * Hz1 ;
83                 H22 = H_tilda' * H_tilda;
84                 Wj_t = eig(H11 , (Mj*v/Ej)*eye(size(H11,2)) + H22);
85                 Wj_tilda( :, ~any(Wj_tilda,1) ) = [];
86                 Wj_tilda = [Wj_tilda Wj_t];
87             end
88         end
89         Hzj = zeros(Nr,Nt);
90
91         % Calculating SINR value for each Lj
92
93         for l = 1:ltmp(j)
94             Hzj = H(:, :, j);
95             Hzj( ~any(Hzj,2), : ) = [];
96             Hj = Hzj;
97             Dj = Wj' * Hj' * Hj * Wj ;
98             Num = Dj * Dj' ;
99             Qj = Wj' * Hj' * Hj ;
100            Wj_tilda(:, ~any(Wj_tilda,1) ) = [];

```

```

101         Dnum = ((Ltmp * v / Ebs) * Dj + ( Qj * Wj_tilda * Wj_tilda' \
Qj') );
102         Numerator = Num;
103         Denominator = Dnum;
104         SINR_j_1 = Numerator / Denominator ;
105
106         % Calculating capacity for user j
107
108         C(j) = C(j) + log2( 1 + SINR_j_1 ) ;
109     end
110 end
111
112     % Calculating Sum Capacity for r receive antenna
113
114     Cr(r) = sum(C);
115 end
116
117     % finding the receive antenna which provides maximum sum capacity
118
119     [r_max,r_bar] = max(Cr);
120     if Cr(r_bar) > Cmax
121         Cmax = Cr(r_bar);
122         S = union(S,r_bar);
123         u_bar = UserId(r_bar);
124         rx = setdiff(rx,r_bar);
125         U = union(U,u_bar);
126         r_bar_ID = r_bar - ((u_bar-1)*Nr);
127         H_tilda( ~any(H_tilda,2), : ) = [];
128
129         % updating H tilda matrix
130
131         H_tilda = [H_tilda; H(:, :,u_bar) ];
132
133         % updating the Data stream matrix of user u_bar which is
134         % selected
135
136         if phase == 1
137             L(u_bar) = L(u_bar) + 1 ;
138         end
139     elseif phase == 1
140         rs = [];
141         for x = rx
142             if ismember(UserId(x),U) == 1
143                 rs = union(rs,x);
144             end
145         end
146         rx = rs;
147         phase = 2;
148     else
149         flag = 0;
150     end

```

```
151     else
152         flag=0;
153     end
154 end
155
156 % generaing the required output
157
158 SumCapacity = Cmax;
159 SelectedReceiveAntenna = S;
160 SelectedUser = U;
161 DataStreams = sum(L);
162 end
163 % -----END OF PROGRAM-----
```