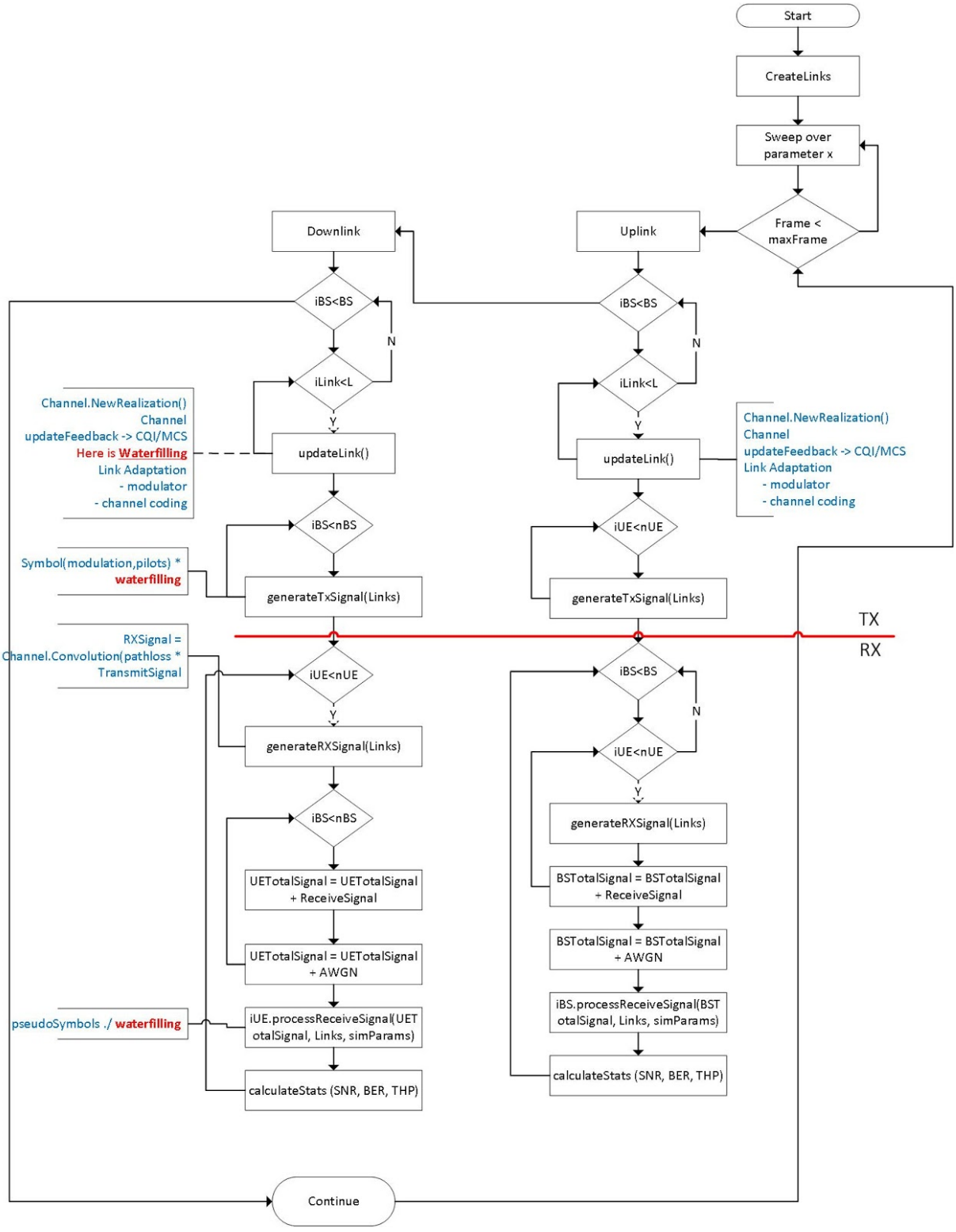
Program Współpraca Polska-RPA	**RAPORT CZĄSTKOWY z realizacji projektu w ramach programu międzynarodowego Polska-RPA**	Narodowe Centrum Badań i Rozwoju

Nr raportu	IR-RATfor5G-04			
Data aktualizacji raportu:	2022.05.17		**Wersja**	9
Numer umowy	PL-RPA2/02/RATfor5G+/2019		**Akronim**	RATfor5G+
Okres realizacji projektu	od	2019.01.01	do	2022.06.30
Tytuł projektu	Technologie dostępu radiowego dla standardu 5G i przyszłych generacji sieci bezprzewodowych			
Tytuł raportu	Modyfikacje symulatora „5G Link Level simulator”			

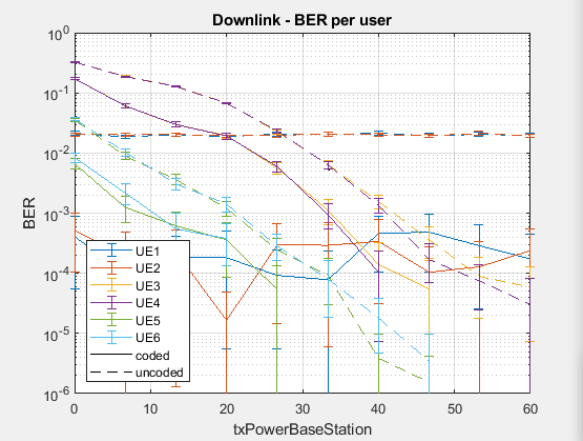
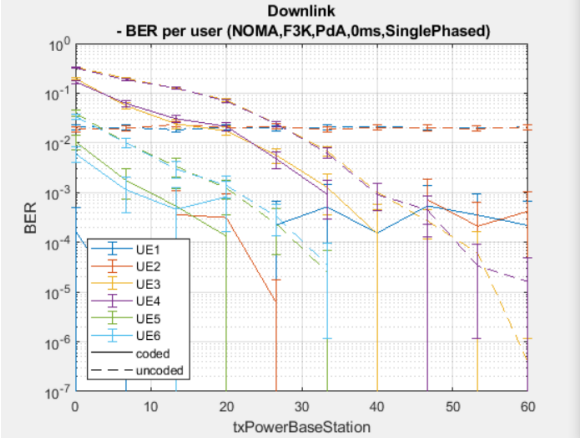
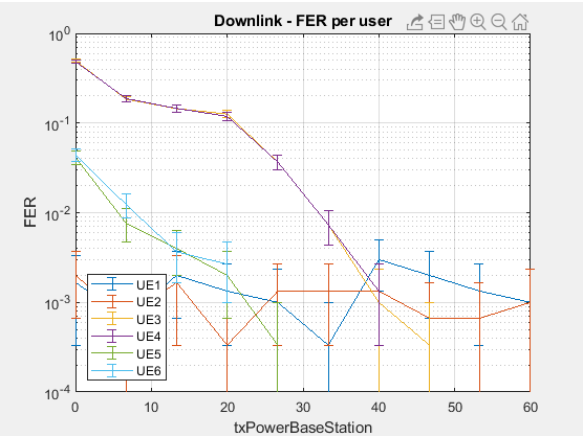
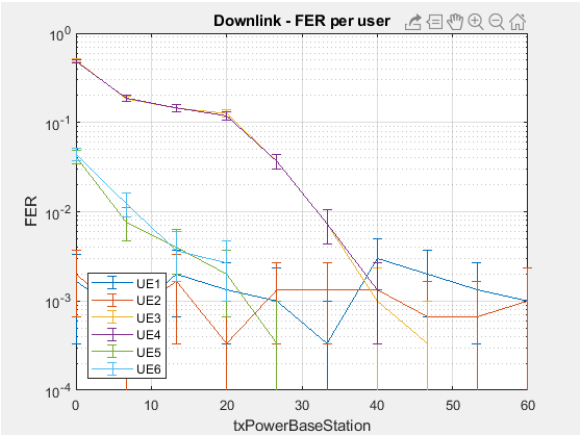
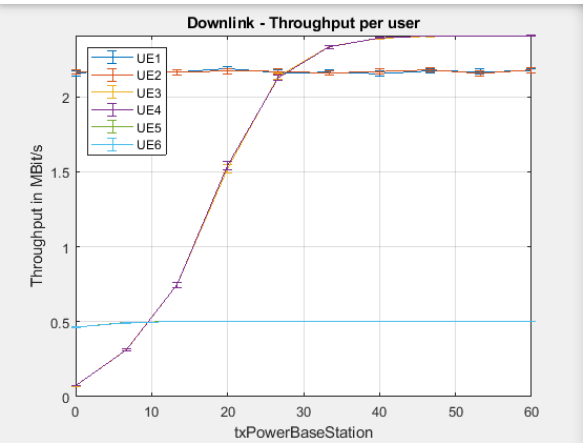
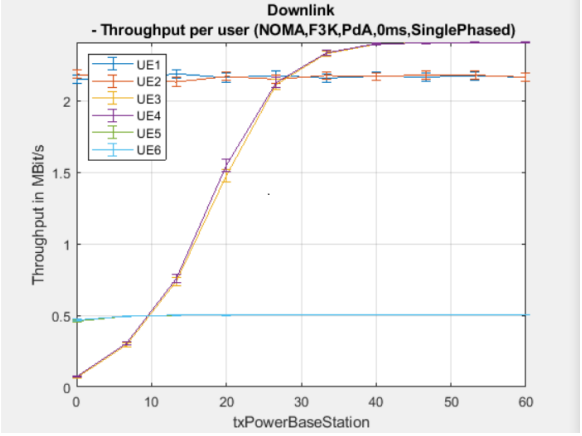
Zespół projektowy pozyskał licencję na oprogramowanie symulacyjne “Vienna 5G Link Level simulator”, oraz 5G Lena (ns3) w ramach przygotowania narzędzia symulacyjnego do wsparcia procesu projektowania metod ograniczania interferencji. Oprogramowanie “Vienna 5G LL” zostało szczegółowo przeanalizowane w zakresie istniejącej implementacji NOMA (dostępna jest tylko wersja MUST). Dokonano szerokiej gamy pomiarów dla kierunku downlink, w szczególności zaimplementowane zostały różne wersje sterowania mocą po stronie stacji bazowej (m.in. *waterfilling*). W tym celu dokonane zostały niezbędne modyfikacje oryginalnego kodu po stronie stacji bazowej 5G - podsumowanie zmian w kodzie zostało zaprezentowane na diagramie (Rys.1). Główne modyfikacje zostały zaaplikowane na poziomie tworzenia symbolu OFDM (updateLink) oraz podczas pozyskiwania sprzężenia zwrotnego z odbiornika (detekcja jakości kanału). Główna metoda realizująca sterowanie mocą zgodnie z podejściem waterfilling jest wywoływana w oparciu o parametry kanału radiowego generowanego każdorazowo przy tworzeniu nowej ramki OFDM. W załączniku Aneks1 zawarte zostały kluczowe elementy zmodyfikowanego kodu. Dodatkowe informacje o przeprowadzonych modyfikacjach dostępne są po przesłaniu maila na adres adamfli@pbs.edu.pl.

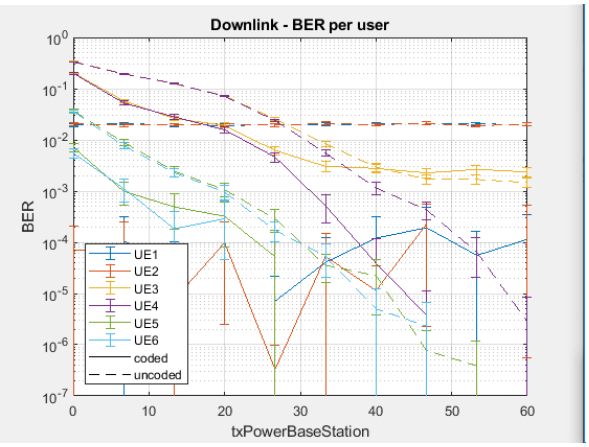
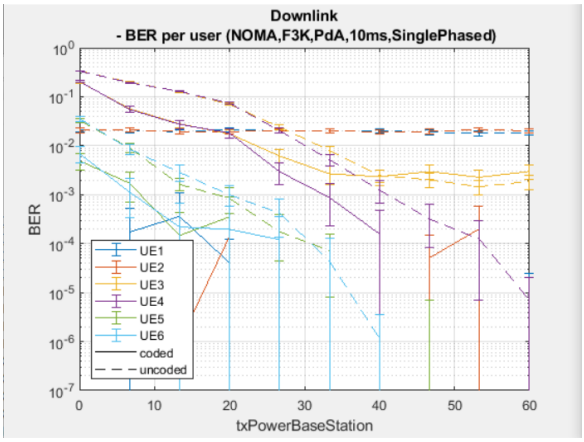
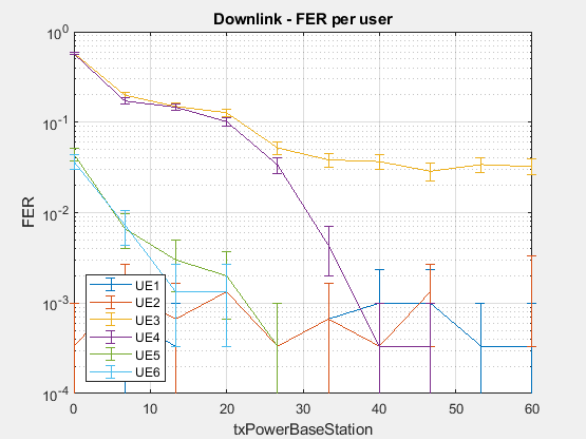
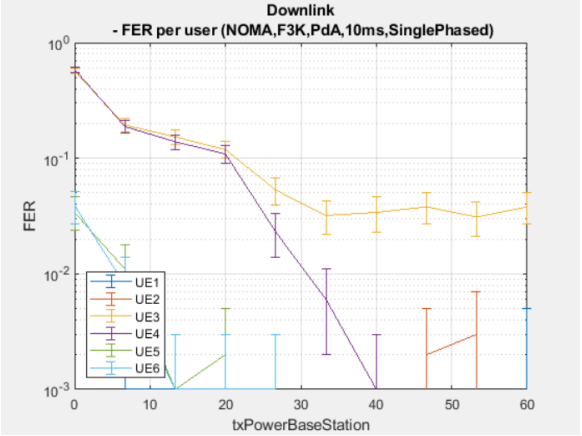
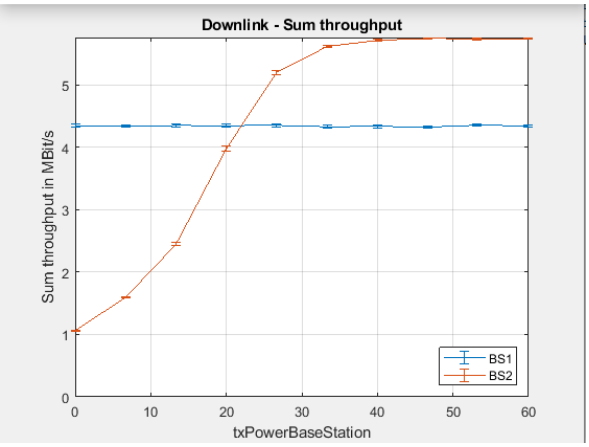
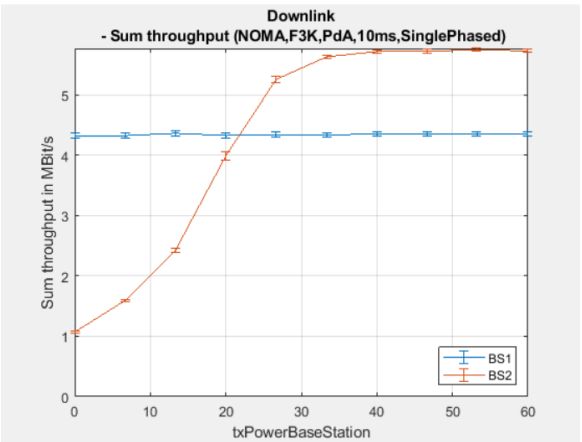


Rys.1 Schemat blokowy implementacji algorytmu waterfilling

Aby zapewnić brak negatywnego wpływu zastosowanych zmian w kodzie na oryginalny kod, zespół wykonał niezbędne testy integracyjne i kalibracyjne. W wyniku testów potwierdzono poprawne i nie zaburzone działanie zmodyfikowanego kodu. Kontrola mocy na wyjściu w nadajniku pokazała, że zarówno przed jak i po załączeniu algorytmu waterfilling poziom mocy całkowitej pozostaje niezmieniony, co potwierdza poprawność generacji sygnału. Ponadto zespół projektowy wszedł w bieżący kontakt z naukowcami z uniwersytetu wiedeńskiego TU Vienna (autorami kodu symulatora), celem bezpośredniej weryfikacji i walidacji przeprowadzanych zmian w kodzie. W ramach tego kontaktu potwierdzona została poprawność zrealizowanych modyfikacji. Poniższe tabele pokazują dla przypadku Pedestrian-A, oraz przy prędkości węzła 0m/s - że wykresy uzyskane za pomocą oryginalnego kodu

oraz kodu zmodyfikowanego ale z wyłączonym waterfillingiem są praktycznie tożsame. Oznacza to że sposób dodania kodu RATfro5G+ nie pogarsza parametrów oryginalnego symulatora. Dopiero załączenie optymalizacji “waterfilling” pozwala zauważyć zmiany jakościowe.

Kod oryginalny	Modyfikacja RATfor5G+ (bez waterfilling)
NOMA, 1000 ramek OFDM, PdA, 0ms	
	
	
	

Kod oryginalny	Modyfikacja RATfor5G+ (bez waterfilling)
NOMA, 1000 ramek OFDM, PdA, 10ms	
	
	
	

Wyniki uzyskane przy wykorzystaniu zaimplementowanego algorytmu waterfilling zostały przedstawione w raporcie z wynikami pomiarów.

Aneks 1 Wybrane modyfikacje symulatora 5G Vienna LL

Funkcja getSChedule() – w której wywoływana jest funkcja ofdmwaterfilling(…)

```
function [f d] = getSchedule(sweep, n, obj, channelsOfUsers)
    %% n - frame number
    %% obj - simParams
    %% channelsOfUsers - keeps the realizations of UE channels (Links)
multiplied by conjugate complex value, in order to be prepared for calling
ofdmwaterfilling


    i=1; % divisor, after how many i-Frames we want to keep udpating the
schedule

    frameAF = mod (n,i); % we select every 10 frame as out "epoch" for
picking another SNR value

    % the idea here is to enter into "waterfilling" like allocation every
    % i-frames in order to align the number of steps from "random walk" and
    % the number of Frames passed...

    if(~frameAF) % it is the i-th frame we pick the PATHLOSS value from the
"random walk of user"

        %user[pathloss, K] = getChannel(user, step);

        % obj.schedule.fixedScheduleDL{2} =
['UE3:24,UE4:48,UE5:UE3,UE6:UE4']; % ta zmienna jest tablicą przechowującą
jednego

        %path_loss(n); % here we are using the path_loss n-th component

        %obj.channel.PHY.K = getNewValueOfK(); % here we pick random value
of K based on some distribution
                                            % CAUTION: at the moment K is
                                            % updated for ALL users the same

        % fprintf("Path loss (n=%2.2f) = %2.2f", n, path_loss(n)); % this
matrix has already been popoulated offline


        % Things below are COPIED as "defaults" from "ofdmwaterfilling" to
        % run the ofdmwaterfilling

        useDefaults = 0;

        if (useDefaults)
```

```
            nSubChannel             = 36;
            totalPower              = -10;

            bandwidth               = 1e6;            %  1 MHz
            noiseDensity            = 1e-11;          % -80 dBm
        else


        % MAPOWANIE parametrów pod waterfilling
        % nSubChannel: means how many subchannels in general are available
        nSubChannel           = 36; %obj.modulation.numberOfSubcarriers(2) /
obj.modulation.nSubcarriersPerSubband;
        %nSubChannel               = length(obj.topology.nLinks); %
RATfor5G+ number of links = number of active users

        % totalPower: in case of NOMA LL Simulator we have the
"txPowerBaseStation" parameter
        totalPower            = obj.simulation.txPowerBaseStation(2); % dBm
= 46

        % bandiwth: total BW of a channel in Mhz. E.g. below we have 72 *
15 Khz = 1080000Hz
        bandwidth             = obj.modulation.numberOfSubcarriers(2) *
obj.modulation.subcarrierSpacing(2);

        % noiseDensity
        noiseDensity          = 1e-11;          % -80 dBm
        %noiseDensity             = obj.phy.noisePower;

        % DOMYSLNE WARTOŚCI parametrów ze scenariusza "DL_NOMA.m"
        % scStr.modulation.nSubcarriersPerSubband     = [12];
% number of subcarriers per subband
        % scStr.modulation.numerOfSubcarriers         = [72];
% per BS; number of used subcarriers
        % scStr.modulation.subcarrierSpacing          = [15e3];
% per BS; per base station in Hz
        % scStr.modulation.nSymbolsTotal              = [15];
%
        end

        A = find(~cellfun(@isempty,channelsOfUsers')); % RATfor5G+: indexes
of the users which we want to allocate power to

        %% 28.07 %%%%%%%%%%%%%for i = 1:length(A)

        % channelStateInformation: here we need to COMBINE (A) the slow-
scale Pathloss cahnges for user mobility with (B) fast changes of K
parameter for the fast-fading
        % CAUTION: pathloss is generated based on RandomWalk in DL_NOMA, K
is picked randomly from getNewValueOfK()
        % channelStateInformation   = "pathloss + K" per user (SNR)
```

```
        % channelStateInformation =
random('rayleigh',1/0.6552,1,nSubChannel);
            if(~useDefaults)
                %channelStateInformation = channelsOfUsers{1,A(i)}(:,1); %
channel of user "i"
                                                                        %
channelsOfUsers{1,5}(:,1);
                channelStateInformationTOTAL =
vertcat(channelsOfUsers{1,A(1)}(:,1),channelsOfUsers{1,A(2)}(:,1));
            else
                channelStateInformation =
random('rayleigh',1/0.6552,1,nSubChannel);
                %this section allows comparing the ofdmwaterfilling() with
                %default values
            end



            stringBEFORE = obj.schedule.fixedScheduleDL{2};
            scheduleMATRIX = strToMatrix(stringBEFORE); % zamiana stringa
zawierającego "schedule" na macierz
                                                        % chodzi o to żeby było
                                                        % łatwiej procesowac
                                                        % wszystkie schedule

            %% ofdmwaterfilling - tutaj dzieje się główna magia dotycząca
            % aktualizacji schedule dla OFDMA-users

            %[shanonCapacity powerAllocated] =
ofdmwaterfilling(nSubChannel,totalPower,channelStateInformation,bandwidth,n
oiseDensity);
            [shanonCapacity powerAllocated] = ofdmwaterfilling(sweep, n,
nSubChannel,totalPower,(channelStateInformationTOTAL/(mean(abs(channelState
InformationTOTAL)))),bandwidth,noiseDensity);

            channelsOfUsers{2,A(1)} = powerAllocated(:,1:36); % we save the
optimal value of power per user
            channelsOfUsers{2,A(2)} = powerAllocated(:,37:72);

            stringAFTER = matrixToStr(scheduleMATRIX);  % Jako że w
waterfilingu już zrobiliśmy nowy schedule teraz wynik aktualizacji
odwaracamy tj. macierz->string

            %% UWAGA!!!
            % w biezacej sytuacji po przejsciu przez matrixToStr() zostają
usunięte NOMA UE!!!!
            obj.schedule.fixedScheduleDL{2} = stringAFTER{1};
        %%
```

```
        %%%% 28.07 %%%%%%%%%%%%%%%%%%%%%%%% end


    % Zamieniłem tą zmienną na tablicę przechowującą osobno każdy UE,
        % dzięki czemu można odnieść się teraz bezpośrednio do każdego
        % elementu tej tablicy według indexu.

        changedFixedSchedule = split(obj.schedule.fixedScheduleDL{2},",");
% stworzona zmienna, która dzieli nam całego stringa w miejscu "," na

% pojedyncze stringi przechowywane w tablicy, więc można się też odnieść

% bezpośrednio do każdego według indexu.

    else
        %obj.schedule.fixedScheduleDL{2} =
['UE3:24,UE4:12,UE7:24,UE8:12,UE5:UE3,UE6:UE4,UE9:UE7,UE10:UE8'];

    end

% see "simulation parameters" line 395
%                             if ~isempty(obj.schedule.fixedScheduleDL{iBS})
%                             tempStr =
strsplit(obj.schedule.fixedScheduleDL{iBS},',');
%                             nSubcarriersTemp = 0;
%                             for iNode = 1:length(tempStr)
%                                 tmpNodeStr = strsplit(tempStr{iNode},':');
%                                 if length(tmpNodeStr) == 2
%                                     if strcmp(tmpNodeStr{1}(1:2), 'UE')
%                                         if ~strcmp(tmpNodeStr{2}(1:2),
'UE')
%                                             if str2double(tmpNodeStr{2}) >
0
%                                                 if
mod(str2double(tmpNodeStr{2}),12)>0 && strcmp(obj.simulation.pilotPattern,
'LTE Downlink')
%                                                     error('When the LTE
Downlink pilot pattern is selected, the number of scheduled subcarriers
must be a mutiple of 12!');
%


     f = obj.schedule.fixedScheduleDL{2};
     d = channelsOfUsers;
end
```

Funkcja ofdmwaterfilling()

```
function [shanonCapacity powerAllocated] = ofdmwaterfilling(sweep, frame,
nSubChannel,totalPower_dBm,channelStateInformation,bandwidth,noiseDensity);

standalone = 0;% if set to "1", the function is executed locally, not
called from getSchedule()

%
%========================================================================
% by:
%   Hamid Ramezani
%   my email : hamid.ramezani@gamil.com
%
%   UPDATE by: A.Flizikowski, T. Wysocki (Prof), T. Marciniak (PhD),
%   University of Technology and Life Sciences

% Date:
%   20 OCT 2006 , version 1
%   15 JUN 2020 , version 2 (RATfor5Gplus)
%
%
% to maximize the capacity of a frequency selective channel, the
% waterfilling algorithm is used. OFDM modulation devides the total
% bandwidth into N subchannels. Large number of subchannels cause each
% subchannel experience a flat fading channel if the proper cyclic prefix
% is added to the end of each OFDM symbol. The waterfilling algorithm
% assigns more power to subchannels which experience good condition and may
% assign no power to bad conditioned subchannels (subchannels withdeep
fading).
%
%   for further information see:
%     J. A. C. Bingham, "Multicarrier Modulation for Data Transmission:
%     an Idea whose Time Has Come" IEEE Communications Magazine, vol. 28,
%     no. 5, pp. 5-14, May 1990
%
% ========================================================================
%                       Parameter definition
% ========================================================================
%
% nSubChannel               : Number of subchannels (4,16,32,...,2^N)
% totalPower                : Total available power for each OFDM symbol
%                             (p watt)
% channelStateInformation   : Channel state information 1_by_nSubChannel
%                             matrix "random('rayleigh',1,1,nSubChannel)"
% bandwidth                 : Total available bandwidth (in Hz)
% noiseDensity              : one sided noise spectral dencity (watt/Hz)
% ========================================================================
%                                Example
% ========================================================================
if(~standalone) % if the function is called from outside then use external
values
    totalPower = 10^((totalPower_dBm-30)/10); % convert to Watts as Vienna
LL uses dBm's in the DL_NOMA.m
```

```
else
    nSubChannel             = 36; %10 bylo default
    totalPower              = 1e-4;           % in Watts, equals -10 dBm
    %totalPower              = 1;             %  30 dBm
    channelStateInformation = random('rayleigh',1/0.6552,1,nSubChannel); %
CAUTION: here we need to modify to be able to change it into "pathloss + K"
    bandwidth               = 1e6;            %  1 MHz
    noiseDensity            = 1e-11;          % -80 dBm
end

% [Capacity PowerAllocated] = ofdmwaterfilling(...
%    nSubChannel,totalPower,channelStateInformation,bandwidth,noiseDensity)
%< Parameter Computation >




    subchannelNoise     = (noiseDensity*bandwidth)/(1*nSubChannel); %%
UWAGA; pamietać żeby "2" dostosować do liczby userów!!!

    if(standalone)
        carrierToNoiseRatio =
(abs(channelStateInformation)).^2/subchannelNoise; % C/N = 10 log10(Pc/Pn)
in dB
    else

        carrierToNoiseRatio =
(abs(channelStateInformation)).^2/subchannelNoise; % C/N = 10 log10(Pc/Pn)
in dB
    end

    revSNR = 1./carrierToNoiseRatio; % 1/SNR for each Subchannel
    suma = sum(revSNR);              % 1/SNR averaged across whole BW

    %initPowerAllo       = (0.001 + suma)/nSubChannel - revSNR;
    initPowerAllo       = (0.001 + suma)/nSubChannel - revSNR;
    initPower = initPowerAllo;

    TESTinitPowerSUM = sum(initPowerAllo);

    %testowanie zmiennych do tego miejsca

%ileRazyNegativePower = 0;
found = 0;

    while(length( find(initPowerAllo < 0 )) > 0 )
        negIndex       = find(initPowerAllo <= 0);
        posIndex       = find(initPowerAllo >  0);
        nSubchannelRem = length(posIndex);
        initPowerAllo(negIndex) = 0;
        CnrRem         = carrierToNoiseRatio(posIndex);
```

```
        powerAlloTemp  = (totalPower + sum(1./CnrRem))/nSubchannelRem -
1./CnrRem;
        initPowerAllo(posIndex) = powerAlloTemp;


        if(found==0)  %% sprawdzamy ile było sytuacji zerowych mocy

            found = 1;
%            if(verboseOutput)
                       display(sprintf(newline));
                       fprintf('Subchannels with NEG power! (Sweep=%i,
Frame=%i)',sweep, frame);
                       %display(sprintf(newline));
%            end
        end

    end

    TESTpowerAllocatedSUM = sum(initPowerAllo);

% < Output Computation >
% amount of power allocated to each subchannel
% IT has been scaled for the needs of RATfor5Gplus
    powerAllocated = initPowerAllo'/ mean(initPowerAllo);

% total capacity of a channel based on shanon theory
    shanonCapacity = bandwidth/nSubChannel * sum(log2(1 +
initPowerAllo.*carrierToNoiseRatio));


if(standalone)
f1 = figure(1);
    clf;
    set(f1,'Color',[1 1 1]);
    bar( ( initPowerAllo + 1./carrierToNoiseRatio),1,'r')  % initPowerAllo
     hold on;
    bar(1./carrierToNoiseRatio,1);
    xlabel('subchannel indices');
    title('Water filling algorithm')
    legend('amount of power allocated to each subchannel',...
           'Noise to Carrier Ratio')

end
```