POWER SYSTEM

Term Project Report

Optimal Network Reconfiguration for Improved Microgrid Resilience

SUBMITTED BY:-
Tanmay 2022EEB1219
Komal 2022EEB1182
Archit 2022EEB1158

Content

- Introduction
- Objective
- Methodology
- Pseudo Code
- Code
- Simulation Results
- Discussion
- Conclusions
- References
- Modification

Introduction

With the increasing occurrences of extreme weather events, the challenges of power outages and the need for rapid system restoration have gained significant attention. Strengthening the resilience of power delivery at the distribution level has become a key priority, and network topology reconfiguration has emerged as an effective restoration strategy.

Simulations are carried out on a modified IEEE 33-bus distribution system. The results clearly demonstrate that the strategy enhances the system's ability to maintain reliable service to critical loads, even in the event of three or four simultaneous line outages, thereby ensuring a more resilient and robust power supply.

Objective

The purpose is to develop and validate an optimal reconfiguration strategy for the modified IEEE 33-bus distribution system to enhance power distribution resilience and efficiency. By altering the system topology through strategic switch operations, the study aims to restore power supply during contingencies while maintaining a radial network structure.

Key constraint considered in the reconfiguration process include radial topology enforcement.

Methodology_

This study focuses on the optimal reconfiguration of network topology using the IEEE 33-bus distribution system as the test platform.

The proposed methodology aims to enhance system reliability, minimise total load shedding, and ensure efficient power delivery.

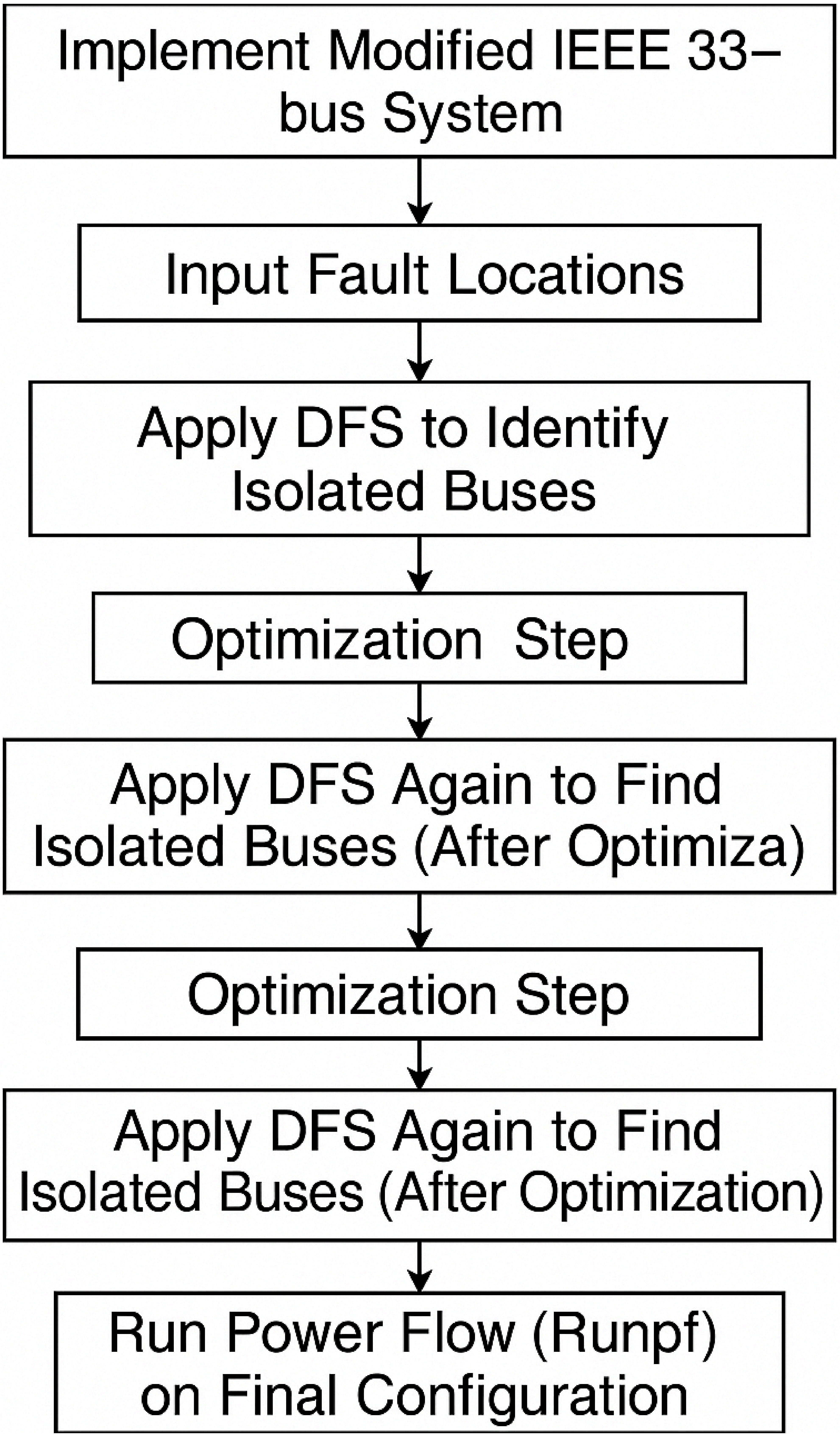
Throughout the reconfiguration process, the radial structure of the distribution network is strictly maintained to preserve operational simplicity and fault isolation capabilities. By strategically altering the network configuration, the approach seeks to reduce system load shedding and increase overall resilience against disturbances.

Used branch and bound technique for optimally switching on/off of tie switches(RCS).

Constraints:

$$\sum_{ij} z_{ij} = |N| - 1 - |A|, \ \forall ij \in L$$

Workflow

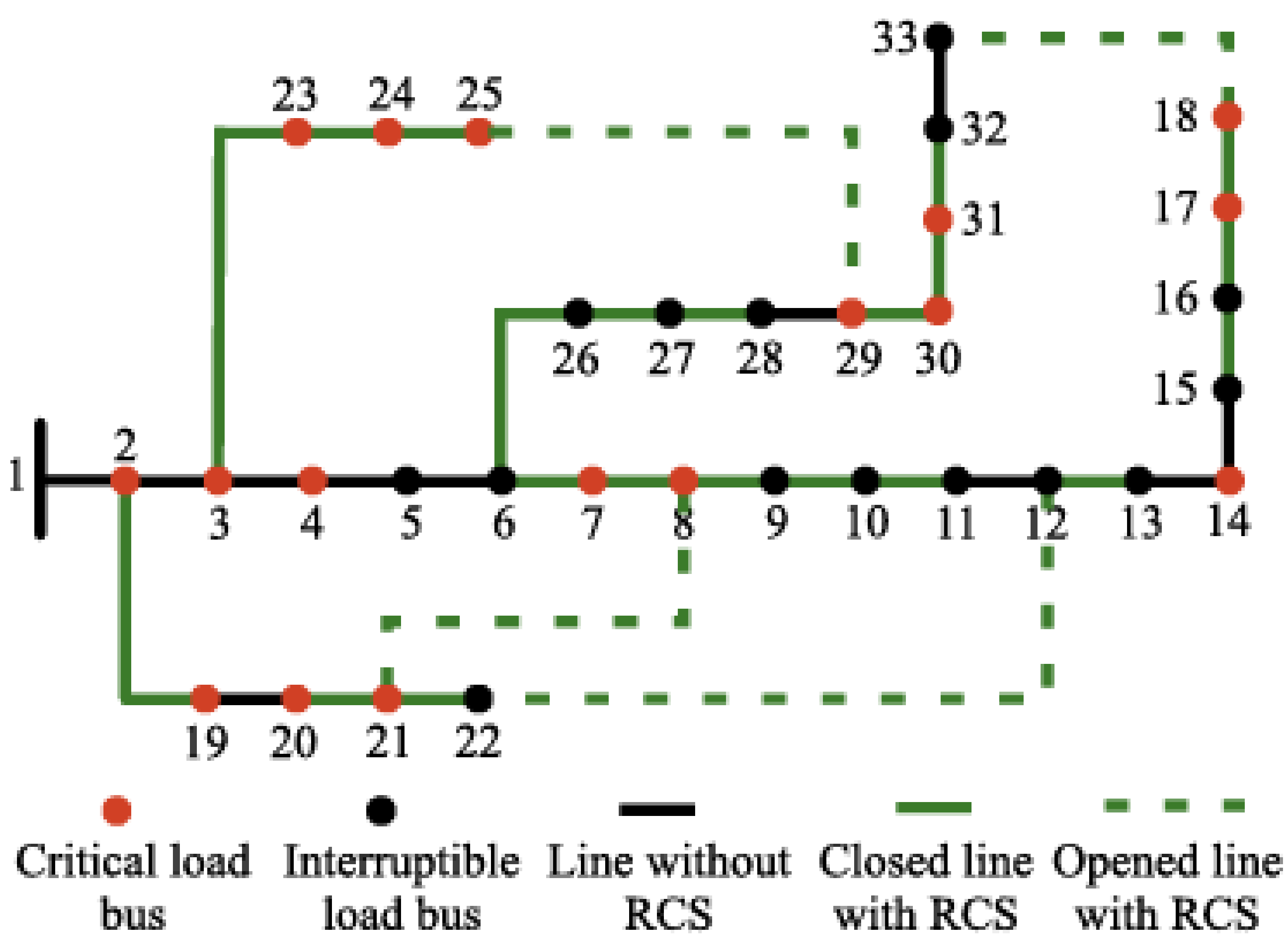

Implement Modified IEEE 33–bus System
Input Fault Locations
Apply DFS to Identify Isolated Buses
Optimization Step
Apply DFS Again to Find Isolated Buses (After Optimiza)
Optimization Step
Apply DFS Again to Find Isolated Buses (After Optimization)
Run Power Flow (Runpf) on Final Configuration

Pseudo Code

Algorithm 1 Reconfiguration of Faults in IEEE-33 Modified Bus System

```
Define IEEE 33-bus system with bus, generator, and branch data
Convert bus loads from kW/kVAR to MW/MVAR
Convert branch impedances from Ohm to per-unit (p.u.)
Plot initial system layout with:
    Active branches (black lines)
    Critical buses (red markers)
    Non-critical buses (green markers)
Ask user for number of faults to simulate, n
Plot during fault layout
Using DFS Algorithm calculate isolated buses
Using isolated buses calculate load_shed
for i = 1 to n do
    Prompt user to enter from_bus and to_bus
    Find corresponding branch between from_bus and to_bus
    Set branch status to 0 (disconnect branch)
end for
Replot system layout with updated branch statuses
Recalculate isolated buses and Update minimum load_shed
Running PowerFlow (runpf)
```

The modified IEEE 33-bus distribution system



Code

```
function mpc = case33bw
%CASE33BW  Power flow data for 33 bus distribution system from Baran & Wu
%   Please see CASEFORMAT for details on the case file format.
%
%   Data from ...
%       M. E. Baran and F. F. Wu, "Network reconfiguration in distribution
%       systems for loss reduction and load balancing," in IEEE Transactions
%       on Power Delivery, vol. 4, no. 2, pp. 1401-1407, Apr 1989.
%       doi: 10.1109/61.25627
%       URL: https://doi.org/10.1109/61.25627

%% MATPOWER Case Format : Version 2
mpc.version = '2';

%%-----  Power Flow Data  -----%%
%% system MVA base
mpc.baseMVA = 10;

%% bus data
%   bus_i   type    Pd  Qd  Gs  Bs  area    Vm  Va  baseKV  zone    Vmax    Vmin
mpc.bus = [ %% (Pd and Qd are specified in kW & kW here, converted to MW & MW below)
    1   3   0       0   0   0   1   1   0   12.66   1   1   1;
    2   1   490     0   0   0   1   1   0   12.66   1   1.1 0.9;
    3   1   495     0   0   0   1   1   0   12.66   1   1.1 0.9;
    4   1   585     0   0   0   1   1   0   12.66   1   1.1 0.9;
    5   1   440     0   0   0   1   1   0   12.66   1   1.1 0.9;
    6   1   460     0   0   0   1   1   0   12.66   1   1.1 0.9;
    7   1   395     0   0   0   1   1   0   12.66   1   1.1 0.9;
    8   1   395     0   0   0   1   1   0   12.66   1   1.1 0.9;
    9   1   380     0   0   0   1   1   0   12.66   1   1.1 0.9;
    10  1   380     0   0   0   1   1   0   12.66   1   1.1 0.9;
    11  1   380     0   0   0   1   1   0   12.66   1   1.1 0.9;
    12  1   380     0   0   0   1   1   0   12.66   1   1.1 0.9;
    13  1   380     0   0   0   1   1   0   12.66   1   1.1 0.9;
    14  1   395     0   0   0   1   1   0   12.66   1   1.1 0.9;
    15  1   480     0   0   0   1   1   0   12.66   1   1.1 0.9;
    16  1   480     0   0   0   1   1   0   12.66   1   1.1 0.9;
    17  1   495     0   0   0   1   1   0   12.66   1   1.1 0.9;
    18  1   495     0   0   0   1   1   0   12.66   1   1.1 0.9;
    19  1   495     0   0   0   1   1   0   12.66   1   1.1 0.9;
    20  1   495     0   0   0   1   1   0   12.66   1   1.1 0.9;
    21  1   442.5   0   0   0   1   1   0   12.66   1   1.1 0.9;
    22  1   560     0   0   0   1   1   0   12.66   1   1.1 0.9;
    23  1   490     0   0   0   1   1   0   12.66   1   1.1 0.9;
    24  1   885     0   0   0   1   1   0   12.66   1   1.1 0.9;
    25  1   742.5   0   0   0   1   1   0   12.66   1   1.1 0.9;
    26  1   1760    0   0   0   1   1   0   12.66   1   1.1 0.9;
    27  1   420     0   0   0   1   1   0   12.66   1   1.1 0.9;
    28  1   460     0   0   0   1   1   0   12.66   1   1.1 0.9;
    29  1   442.5   0   0   0   1   1   0   12.66   1   1.1 0.9;
    30  1   585     0   0   0   1   1   0   12.66   1   1.1 0.9;
    31  1   595     0   0   0   1   1   0   12.66   1   1.1 0.9;
    32  1   580     0   0   0   1   1   0   12.66   1   1.1 0.9;
    33  1   590     0   0   0   1   1   0   12.66   1   1.1 0.9;
];
```

Generator Data

```
%% generator data
%   bus Pg  Qg  Qmax    Qmin    Vg  mBase   status  Pmax    Pmin    Pc1 Pc2 Qc1min  Qc1max  Qc
mpc.gen = [
    1   0   0   10  -10 1   100 1   10  0   0   0   0   0   0   0   0   0   0   0   0;
    2   0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    3   0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    4   0   0   10  -10 1   100 1   0.5 0   0   0   0   0   0   0   0   0   0   0   0;
    5   0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    6   0   0   10  -10 1   100 1   0.3 0   0   0   0   0   0   0   0   0   0   0   0;
    7   0   0   10  -10 1   100 1   0.3 0   0   0   0   0   0   0   0   0   0   0   0;
    8   0   0   10  -10 1   100 1   0.2 0   0   0   0   0   0   0   0   0   0   0   0;
    9   0   0   10  -10 1   100 1   0.2 0   0   0   0   0   0   0   0   0   0   0   0;
    10  0   0   10  -10 1   100 1   0.2 0   0   0   0   0   0   0   0   0   0   0   0;
    11  0   0   10  -10 1   100 1   0.2 0   0   0   0   0   0   0   0   0   0   0   0;
    12  0   0   10  -10 1   100 1   0.3 0   0   0   0   0   0   0   0   0   0   0   0;
    13  0   0   10  -10 1   100 1   0.3 0   0   0   0   0   0   0   0   0   0   0   0;
    14  0   0   10  -10 1   100 1   0.3 0   0   0   0   0   0   0   0   0   0   0   0;
    15  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    16  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    17  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    18  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    19  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    20  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    21  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    22  0   0   10  -10 1   100 1   0.5 0   0   0   0   0   0   0   0   0   0   0   0;
    23  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    24  0   0   10  -10 1   100 1   0.7 0   0   0   0   0   0   0   0   0   0   0   0;
    25  0   0   10  -10 1   100 1   0.7 0   0   0   0   0   0   0   0   0   0   0   0;
    26  0   0   10  -10 1   100 1   1.7 0   0   0   0   0   0   0   0   0   0   0   0;
    27  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    28  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    29  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    30  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    31  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    32  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    33  0   0   10  -10 1   100 1   0.4 0   0   0   0   0   0   0   0   0   0   0   0;
    ];
```

Branch Data

```
%% branch data
%   fbus    tbus    r   x   b   rateA   rateB   rateC   ratio   angle   status  angmin  angmax
mpc.branch = [  %% (r and x specified in ohms here, converted to p.u. below)
    1   2   0.0922  0.0470  0   0   0   0   0   0   1   -360    360;
    2   3   0.4930  0.2511  0   0   0   0   0   0   1   -360    360;
    3   4   0.3660  0.1864  0   0   0   0   0   0   1   -360    360;
    4   5   0.3811  0.1941  0   0   0   0   0   0   1   -360    360;
    5   6   0.8190  0.7070  0   0   0   0   0   0   1   -360    360;
    6   7   0.1872  0.6188  0   0   0   0   0   0   1   -360    360;
    7   8   0.7114  0.2351  0   0   0   0   0   0   1   -360    360;
    8   9   1.0300  0.7400  0   0   0   0   0   0   1   -360    360;
    9   10  1.0440  0.7400  0   0   0   0   0   0   1   -360    360;
    10  11  0.1966  0.0650  0   0   0   0   0   0   1   -360    360;
    11  12  0.3744  0.1238  0   0   0   0   0   0   1   -360    360;
    12  13  1.4680  1.1550  0   0   0   0   0   0   1   -360    360;
    13  14  0.5416  0.7129  0   0   0   0   0   0   1   -360    360;
    14  15  0.5910  0.5260  0   0   0   0   0   0   1   -360    360;
    15  16  0.7463  0.5450  0   0   0   0   0   0   1   -360    360;
    16  17  1.2890  1.7210  0   0   0   0   0   0   1   -360    360;
    17  18  0.7320  0.5740  0   0   0   0   0   0   1   -360    360;
    2   19  0.1640  0.1565  0   0   0   0   0   0   1   -360    360;
    19  20  1.5042  1.3554  0   0   0   0   0   0   1   -360    360;
    20  21  0.4095  0.4784  0   0   0   0   0   0   1   -360    360;
    21  22  0.7089  0.9373  0   0   0   0   0   0   1   -360    360;
    3   23  0.4512  0.3083  0   0   0   0   0   0   1   -360    360;
    23  24  0.8980  0.7091  0   0   0   0   0   0   1   -360    360;
    24  25  0.8960  0.7011  0   0   0   0   0   0   1   -360    360;
    6   26  0.2030  0.1034  0   0   0   0   0   0   1   -360    360;
    26  27  0.2842  0.1447  0   0   0   0   0   0   1   -360    360;
    27  28  1.0590  0.9337  0   0   0   0   0   0   1   -360    360;
    28  29  0.8042  0.7006  0   0   0   0   0   0   1   -360    360;
    29  30  0.5075  0.2585  0   0   0   0   0   0   1   -360    360;
    30  31  0.9744  0.9630  0   0   0   0   0   0   1   -360    360;
    31  32  0.3105  0.3619  0   0   0   0   0   0   1   -360    360;
    32  33  0.3410  0.5302  0   0   0   0   0   0   1   -360    360;
    21  8   2.0000  2.0000  0   0   0   0   0   0   0   -360    360;
    12  22  2.0000  2.0000  0   0   0   0   0   0   0   -360    360;
    18  33  0.5000  0.5000  0   0   0   0   0   0   0   -360    360;
    25  29  0.5000  0.5000  0   0   0   0   0   0   0   -360    360;
];
```

```
has_critical = [2 3 4 7 8 14 17 18 19 20 21 23 24 25 29 30 31] ;
has_noncritical = [1 5 6 9 10 11 12 13 15 16 22 26 27 28 32 33] ;
```

Code for bus plot

```
%% plot for 33 bus system(before fault)
% Coordinates to match visual layout (can be adjusted as needed)
    % x and y should be the same length as number of buses (33)
    % These are illustrative - you can refine based on your image
    xy = [...
        0 0;     % Bus 1
        1 0;     % Bus 2
        2 0;     % Bus 3
        3 0;     % Bus 4
        4 0;     % Bus 5
        5 0;     % Bus 6
        6 0;     % Bus 7
        7 0;     % Bus 8
        8 0;     % Bus 9
        9 0;     % Bus 10
        10 0;    % Bus 11
        11 0;    % Bus 12
        12 0;    % Bus 13
        13 0;    % Bus 14
        14 0;    % Bus 15
        15 0;    % Bus 16
        16 0;    % Bus 17
        17 0;    % Bus 18
        1 1;     % Bus 19
        2 1;     % Bus 20
        3 1;     % Bus 21
        4 1;     % Bus 22
        2 -1;    % Bus 23
        3 -1;    % Bus 24
        4 -1;    % Bus 25
        5 -1;    % Bus 26
        6 -1;    % Bus 27
        7 -1;    % Bus 28
        8 -1;    % Bus 29
        9 -1;    % Bus 30
        10 -1;   % Bus 31
        11 -1;   % Bus 32
        12 -1;   % Bus 33
    ];

    % Get branch data
    branch = mpc.branch;

    figure; hold on;
    axis equal;
    title('IEEE 33-Bus System Layout (Only Active Branches)');
    xlabel('X');
    ylabel('Y');

    % Plot all active branches
    for i = 1:size(branch, 1)
        if branch(i, 11) == 1  % Only plot if status == 1
            from = branch(i, 1);
            to = branch(i, 2);
            x = [xy(from, 1), xy(to, 1)];
            y = [xy(from, 2), xy(to, 2)];
            plot(x, y, 'b-', 'LineWidth', 2);
        end
    end

    % Plot buses
    scatter(xy(has_critical, 1), xy(has_critical, 2), 100, 'r', 'filled');      % Cri
    scatter(xy(has_noncritical, 1), xy(has_noncritical, 2), 100, 'k', 'filled'); % N
    text(xy(:, 1)+0.1, xy(:, 2)+0.1, string(1:size(xy, 1)));

    grid on;
```

Giving Fault

```
    %% giving fault to bus system
% Identify the line to disconnect (e.g., between buses 8 and 9)
i= input('enter the no. of faults: ');
fault = zeros(i,2);

while i>0
target_fbus = input('enter the from bus: ');
target_tbus = input('enter the to bus: ');
fault(i,1) = target_fbus;
fault(i,2) = target_tbus;
% Find the branch index
branch_index = find(mpc.branch(:, 1) == target_fbus & mpc.branch(:, 2) == target_tbus);

% Disconnect the line by setting status to 0
mpc.branch(branch_index, 11) = 0;  % Column 11 is BR_STATUS
i=i-1;
end
```

Load shedding calculation

```
%% calculating load shedding

    % Number of buses
nb = size(mpc.bus, 1);

% Initialize PV generation vector (MW) per bus
PV_generation = zeros(nb, 1);

% Loop through generators to accumulate generation at each bus
for k = 1:size(mpc.gen, 1)
    bus_idx = mpc.gen(k, 1);      % bus number where generator is connected
    PG = mpc.gen(k, 9);           % active power generation (MW)
    PV_generation(bus_idx) = PV_generation(bus_idx) + PG;
end

% Compute net load after shedding PV generation (but do not modify mpc.bus)
load_shed = mpc.bus(:, 3) - PV_generation;
display(load_shed)
```

Code for finding island buses and total load shed

```
%% applying DFS for finding island buses
visited = zeros(1, 33);
 visited(1) = 1; % Start DFS from bus 1
 adj = cell(1, 33);
    for i = 1:size(mpc.branch, 1)
        if mpc.branch(i, 11) == 1
            from = mpc.branch(i, 1);
            to = mpc.branch(i, 2);
            adj{from} = [adj{from}, to];
            adj{to} = [adj{to}, from]; % Assuming undirected connectivity
        end
    end
stack = [1];
    while ~isempty(stack)
        current = stack(end);
        stack(end) = [];

        for neighbor = adj{current}
            if ~visited(neighbor)
                visited(neighbor) = 1;
                stack(end + 1) = neighbor; % push to stack
            end
        end
    end
    total_loadshed = 0;

    % Check unvisited buses (2 to 33)
    for bus = 2:33
        if visited(bus) == 0
            fprintf('Bus %d is isolated.\n', bus);
            total_loadshed = total_loadshed + load_shed(bus);
        end
    end
    display(total_loadshed);
    total_loadshedmin= total_loadshed;
     branch_c= mpc.branch;
```

Code for reconfiguration

```
%% Reconfiguration part

for q10=1:2
     if isempty(find(fault(:,1) == 2 & fault(:,2) == 19)) == 0
            if(q10 == 1)
                continue;
            end
        else
    if (q10 == 1)
        mpc.branch(18,11)=1;
    elseif (q10 ==2)
        mpc.branch(18,11)=0;
    end
     end
for q11=1:2
     if isempty(find(fault(:,1) == 20 & fault(:,2) == 21)) == 0
            if(q11 == 1)
                continue;
            end
        else
    if (q11 == 1)
        mpc.branch(20,11)=1;
    elseif (q11 ==2)
        mpc.branch(20,11)=0;
    end
     end
for q12=1:2
     if isempty(find(fault(:,1) == 21 & fault(:,2) == 22)) == 0
            if(q12 == 1)
                continue;
            end
        else
    if (q12 == 1)
        mpc.branch(21,11)=1;
    elseif (q12 ==2)
        mpc.branch(21,11)=0;
    end
     end
for q13=1:2
    if isempty(find(fault(:,1) == 3 & fault(:,2) == 23)) == 0
            if(q13 == 1)
                continue;
            end
        else
    if (q13 == 1)
        mpc.branch(22,11)=1;
    elseif (q13 ==2)
        mpc.branch(22,11)=0;
    end
    end
```

```
for q14=1:2
     if isempty(find(fault(:,1) == 23 & fault(:,2) == 24)) == 0
            if(q14 == 1)
                continue;
            end
        else
    if (q14 == 1)
        mpc.branch(23,11)=1;
    elseif (q14 ==2)
        mpc.branch(23,11)=0;
    end
     end
for q15=1:2
     if isempty(find(fault(:,1) == 24 & fault(:,2) == 25)) == 0
            if(q15 == 1)
                continue;
            end
        else
    if (q15 == 1)
        mpc.branch(24,11)=1;
    elseif (q15 ==2)
        mpc.branch(24,11)=0;
    end
     end
for q16=1:2
     if isempty(find(fault(:,1) == 6 & fault(:,2) == 26)) == 0
            if(q16 == 1)
                continue;
            end
        else
    if (q16 == 1)
        mpc.branch(25,11)=1;
    elseif (q16 ==2)
        mpc.branch(25,11)=0;
    end
     end
for q17=1:2
    if isempty(find(fault(:,1) == 26 & fault(:,2) == 27)) == 0
            if(q17 == 1)
                continue;
            end
        else
    if (q17 == 1)
        mpc.branch(26,11)=1;
    elseif (q17 ==2)
        mpc.branch(26,11)=0;
    end
    end
```

```
for q18=1:2
     if isempty(find(fault(:,1) == 27 & fault(:,2) == 28)) == 0
            if(q18 == 1)
                continue;
            end
        else
    if (q18 == 1)
        mpc.branch(27,11)=1;
    elseif (q18 ==2)
        mpc.branch(27,11)=0;
    end
     end
for q19=1:2
     if isempty(find(fault(:,1) == 29 & fault(:,2) == 30)) == 0
            if(q19 == 1)
                continue;
            end
        else
    if (q19 == 1)
        mpc.branch(29,11)=1;
    elseif (q19 ==2)
        mpc.branch(29,11)=0;
    end
     end
for q20=1:2
     if isempty(find(fault(:,1) == 30 & fault(:,2) == 31)) == 0
            if(q20 == 1)
                continue;
            end
        else
    if (q20 == 1)
        mpc.branch(30,11)=1;
    elseif (q20 ==2)
        mpc.branch(30,11)=0;
    end
     end
for q21=1:2
    if isempty(find(fault(:,1) == 31 & fault(:,2) == 32)) == 0
            if(q21 == 1)
                continue;
            end
        else
    if (q21 == 1)
        mpc.branch(31,11)=1;
    elseif (q21 ==2)
        mpc.branch(31,11)=0;
    end
    end
```

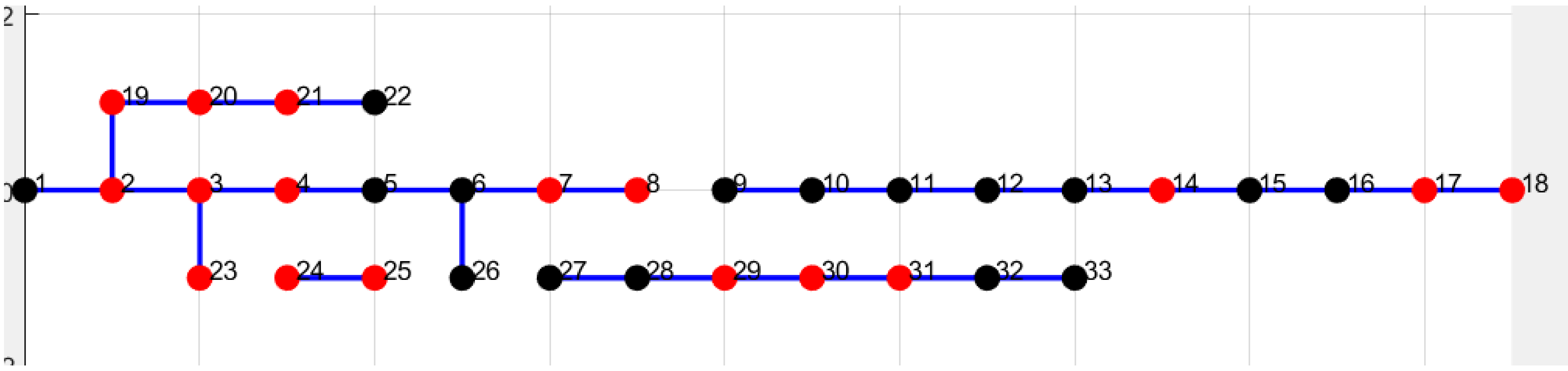
```
for q22=1:2
     if isempty(find(fault(:,1) == 21 & fault(:,2) == 8)) == 0
            if(q22 == 1)
                continue;
            end
        else
    if (q22 == 1)
        mpc.branch(33,11)=1;
    elseif (q22 ==2)
        mpc.branch(33,11)=0;
    end
     end
for q23=1:2
     if isempty(find(fault(:,1) == 12 & fault(:,2) == 22)) == 0
            if(q23 == 1)
                continue;
            end
        else
    if (q23 == 1)
        mpc.branch(34,11)=1;
    elseif (q23 ==2)
        mpc.branch(34,11)=0;
    end
     end
for q24=1:2
    if isempty(find(fault(:,1) == 18 & fault(:,2) == 33)) == 0
            if(q24 == 1)
                continue;
            end
        else
    if (q24 == 1)
        mpc.branch(35,11)=1;
    elseif (q24 ==2)
        mpc.branch(35,11)=0;
    end
    end
for q25=1:2
     if isempty(find(fault(:,1) == 25 & fault(:,2) == 29)) == 0
            if(q25 == 1)
                continue;
            end
        else
    if (q25 == 1)
        mpc.branch(36,11)=1;
    elseif (q25 ==2)
        mpc.branch(36,11)=0;
    end
     end
```

Checking radial topology and load shed in the iteration

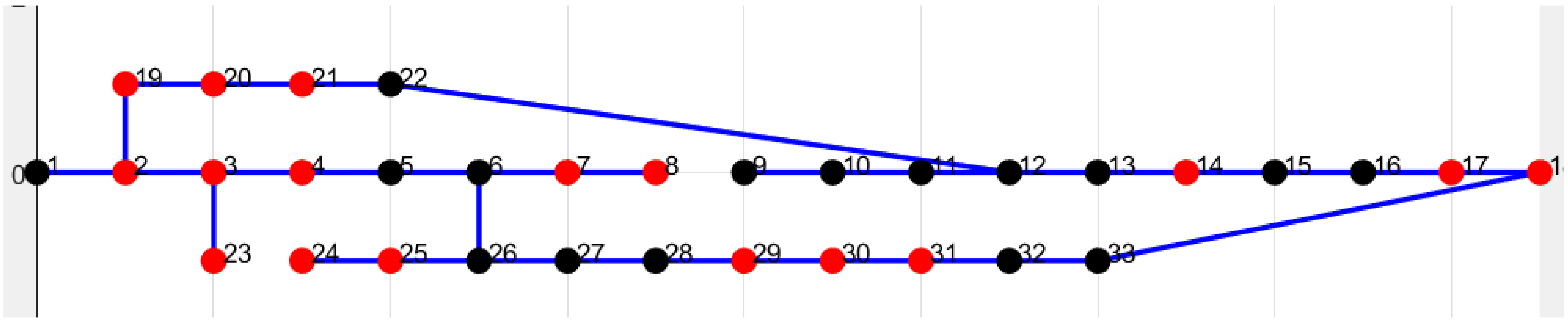
```
num_buses = size(mpc.bus, 1);
    island_buses= length(isolated);
    count = 0;
    for k=1:size(mpc.branch,1)
       if isempty(find(isolated == mpc.branch(k,1)))
           if isempty(find(isolated == mpc.branch(k,2)))
           count = count + mpc.branch(k,11);
           end
        end
    end

    % If all buses are visited and there is no loop, it's radial
    if (num_buses - island_buses - 1 == count)
    total_loadshed = 0;
if isempty(isolated) == 0
    for y= 1: length(isolated)
        total_loadshed = total_loadshed + load_shed(isolated(y));
    end
end
if (total_loadshed < total_loadshedmin)
    total_loadshedmin = total_loadshed;
    branch_c= mpc.branch;
end
    else
        continue;
    end
```

Simulation Results:

3 line outages in line 8-9, line 23-24, line 26-27



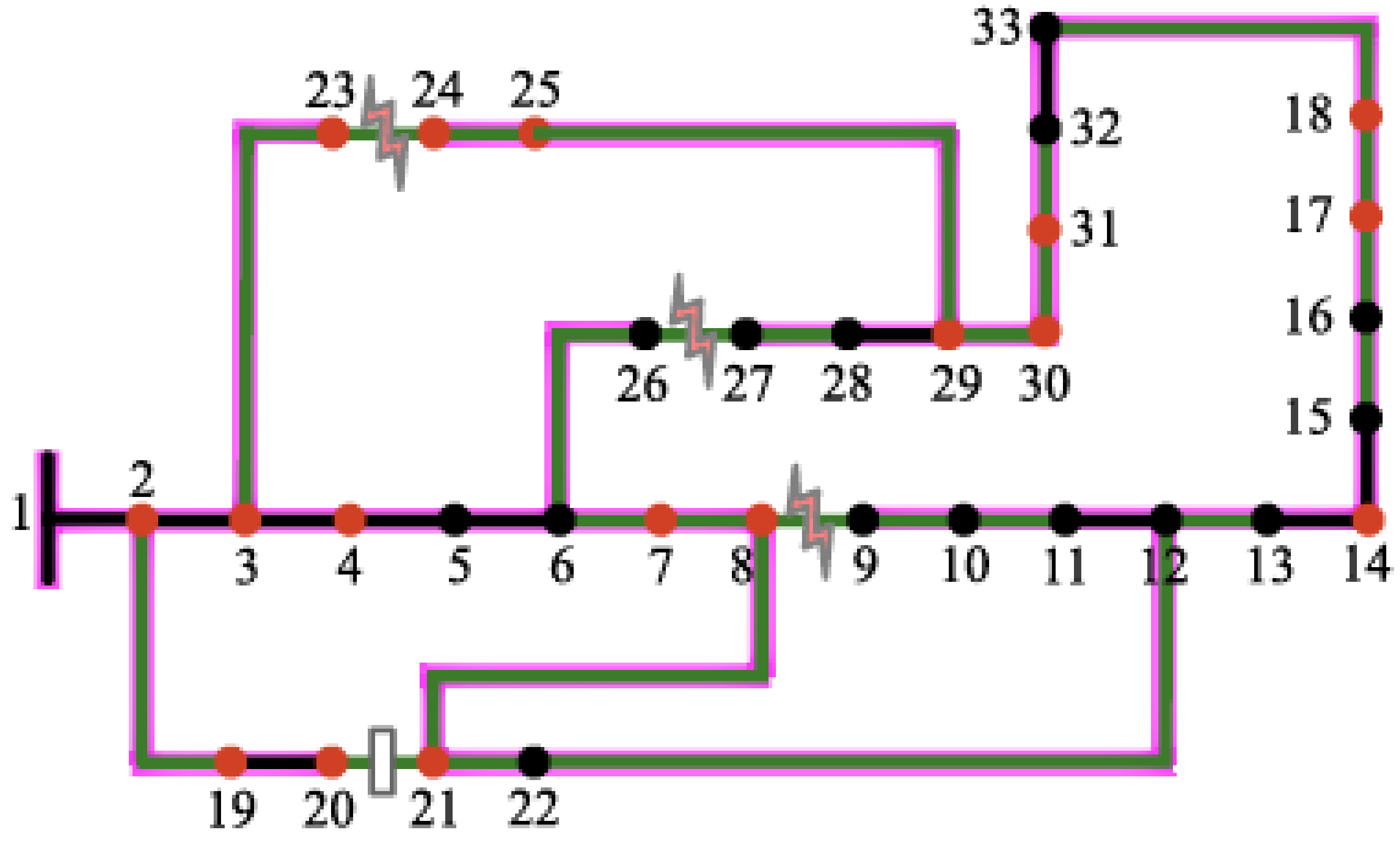
Optimally reconfigured network topology



Load shedding before and after optimization

```
total_loadshed =

    2.2450


total_loadshedmin =

     0
```

Results from paper



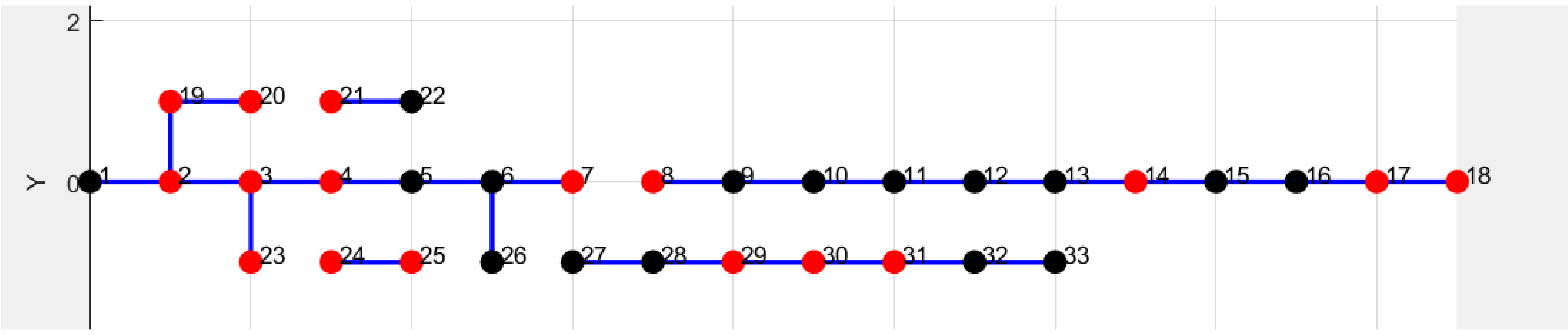
(a) Optimized topology with 3 line outages.

Branch data

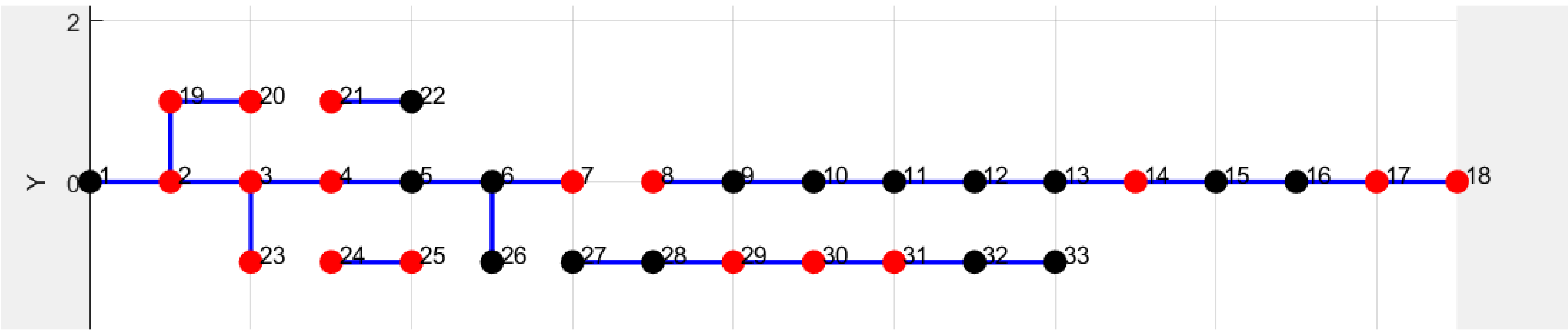
branch_c =

1.0000	2.0000	0.0058	0.0029	0	0	0	0	0	0	1.0000	-360.0000	360.0000
2.0000	3.0000	0.0308	0.0157	0	0	0	0	0	0	1.0000	-360.0000	360.0000
3.0000	4.0000	0.0228	0.0116	0	0	0	0	0	0	1.0000	-360.0000	360.0000
4.0000	5.0000	0.0238	0.0121	0	0	0	0	0	0	1.0000	-360.0000	360.0000
5.0000	6.0000	0.0511	0.0441	0	0	0	0	0	0	1.0000	-360.0000	360.0000
6.0000	7.0000	0.0117	0.0386	0	0	0	0	0	0	1.0000	-360.0000	360.0000
7.0000	8.0000	0.0444	0.0147	0	0	0	0	0	0	1.0000	-360.0000	360.0000
8.0000	9.0000	0.0643	0.0462	0	0	0	0	0	0	0	-360.0000	360.0000
9.0000	10.0000	0.0651	0.0462	0	0	0	0	0	0	1.0000	-360.0000	360.0000
10.0000	11.0000	0.0123	0.0041	0	0	0	0	0	0	1.0000	-360.0000	360.0000
11.0000	12.0000	0.0234	0.0077	0	0	0	0	0	0	1.0000	-360.0000	360.0000
12.0000	13.0000	0.0916	0.0721	0	0	0	0	0	0	1.0000	-360.0000	360.0000
13.0000	14.0000	0.0338	0.0445	0	0	0	0	0	0	1.0000	-360.0000	360.0000
14.0000	15.0000	0.0369	0.0328	0	0	0	0	0	0	1.0000	-360.0000	360.0000
15.0000	16.0000	0.0466	0.0340	0	0	0	0	0	0	1.0000	-360.0000	360.0000
16.0000	17.0000	0.0804	0.1074	0	0	0	0	0	0	1.0000	-360.0000	360.0000
17.0000	18.0000	0.0457	0.0358	0	0	0	0	0	0	1.0000	-360.0000	360.0000
2.0000	19.0000	0.0102	0.0098	0	0	0	0	0	0	1.0000	-360.0000	360.0000
19.0000	20.0000	0.0939	0.0846	0	0	0	0	0	0	1.0000	-360.0000	360.0000
20.0000	21.0000	0.0255	0.0298	0	0	0	0	0	0	1.0000	-360.0000	360.0000
21.0000	22.0000	0.0442	0.0585	0	0	0	0	0	0	1.0000	-360.0000	360.0000
3.0000	23.0000	0.0282	0.0192	0	0	0	0	0	0	1.0000	-360.0000	360.0000
23.0000	24.0000	0.0560	0.0442	0	0	0	0	0	0	0	-360.0000	360.0000
24.0000	25.0000	0.0559	0.0437	0	0	0	0	0	0	1.0000	-360.0000	360.0000
6.0000	26.0000	0.0127	0.0065	0	0	0	0	0	0	1.0000	-360.0000	360.0000
26.0000	27.0000	0.0177	0.0090	0	0	0	0	0	0	0	-360.0000	360.0000
27.0000	28.0000	0.0661	0.0583	0	0	0	0	0	0	1.0000	-360.0000	360.0000
28.0000	29.0000	0.0502	0.0437	0	0	0	0	0	0	1.0000	-360.0000	360.0000
29.0000	30.0000	0.0317	0.0161	0	0	0	0	0	0	1.0000	-360.0000	360.0000
30.0000	31.0000	0.0608	0.0601	0	0	0	0	0	0	1.0000	-360.0000	360.0000
31.0000	32.0000	0.0194	0.0226	0	0	0	0	0	0	1.0000	-360.0000	360.0000
32.0000	33.0000	0.0213	0.0331	0	0	0	0	0	0	1.0000	-360.0000	360.0000
21.0000	8.0000	0.1248	0.1248	0	0	0	0	0	0	0	-360.0000	360.0000
12.0000	22.0000	0.1248	0.1248	0	0	0	0	0	0	1.0000	-360.0000	360.0000
18.0000	33.0000	0.0312	0.0312	0	0	0	0	0	0	1.0000	-360.0000	360.0000
25.0000	29.0000	0.0312	0.0312	0	0	0	0	0	0	1.0000	-360.0000	360.0000

4 line outages in line 7-8, line 20-21, line 23-24, line26-27



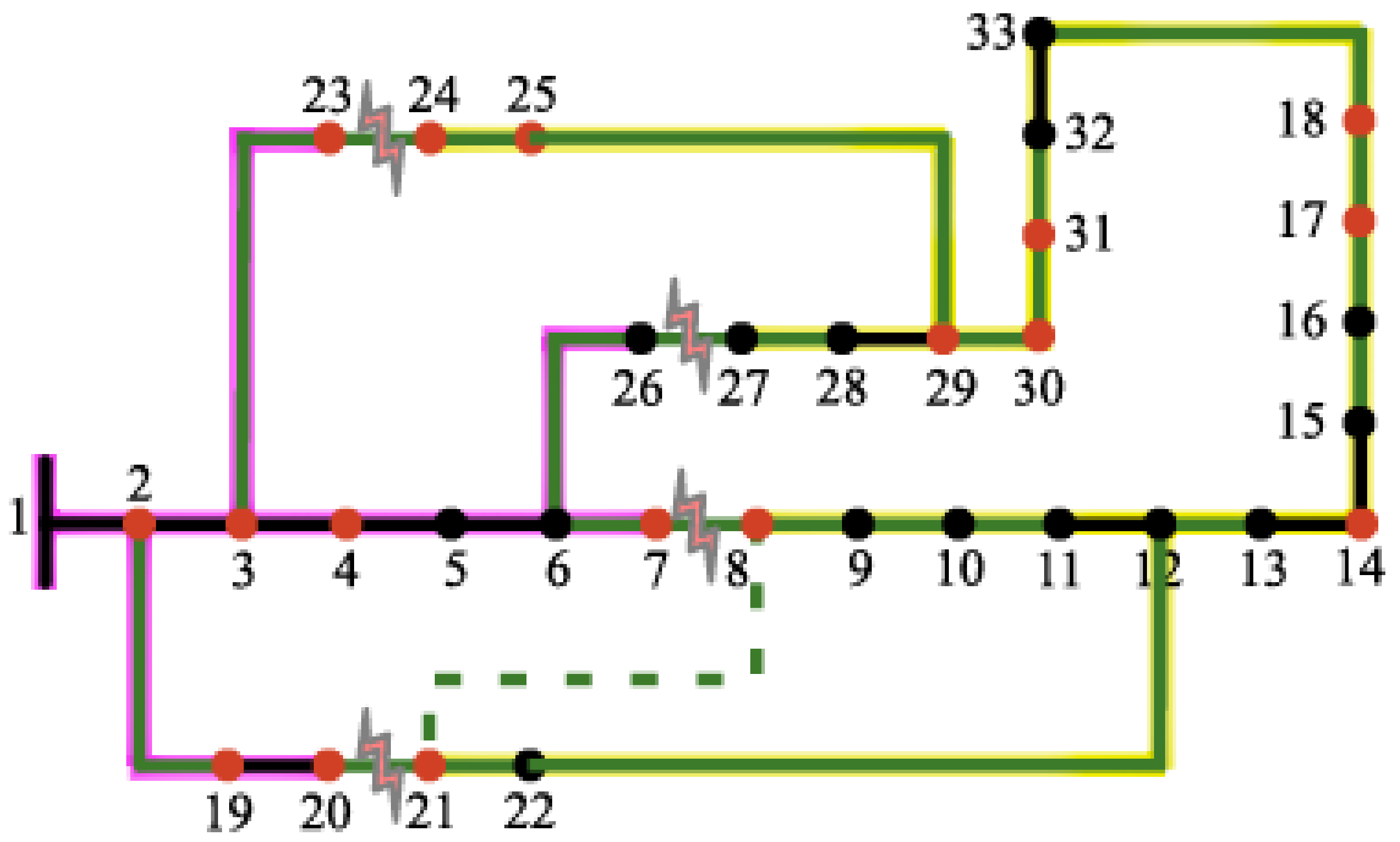
Optimally reconfigured network topology



Load shedding before and after optimization

```
total_loadshed =

    2.5425


total_loadshedmin =

    2.5425
```

Results from paper



(b) Optimized topology with 4 line outages.

Branch data

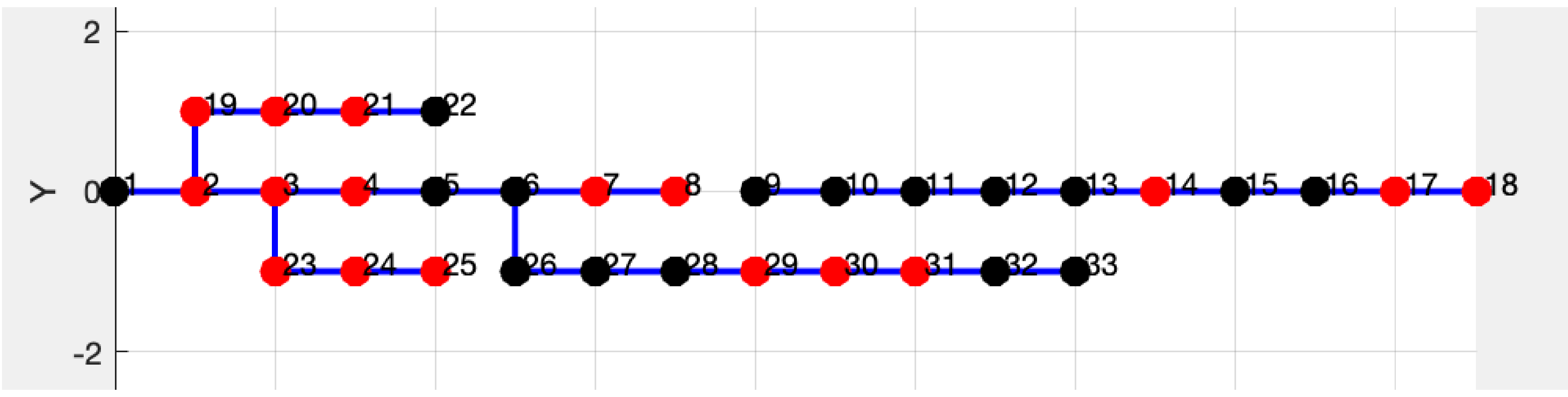
```
branch_c =
```

1.0000	2.0000	0.0058	0.0029	0	0	0	0	0	0	1.0000	-360.0000	360.0000
2.0000	3.0000	0.0308	0.0157	0	0	0	0	0	0	1.0000	-360.0000	360.0000
3.0000	4.0000	0.0228	0.0116	0	0	0	0	0	0	1.0000	-360.0000	360.0000
4.0000	5.0000	0.0238	0.0121	0	0	0	0	0	0	1.0000	-360.0000	360.0000
5.0000	6.0000	0.0511	0.0441	0	0	0	0	0	0	1.0000	-360.0000	360.0000
6.0000	7.0000	0.0117	0.0386	0	0	0	0	0	0	1.0000	-360.0000	360.0000
7.0000	8.0000	0.0444	0.0147	0	0	0	0	0	0	0	-360.0000	360.0000
8.0000	9.0000	0.0643	0.0462	0	0	0	0	0	0	1.0000	-360.0000	360.0000
9.0000	10.0000	0.0651	0.0462	0	0	0	0	0	0	1.0000	-360.0000	360.0000
10.0000	11.0000	0.0123	0.0041	0	0	0	0	0	0	1.0000	-360.0000	360.0000
11.0000	12.0000	0.0234	0.0077	0	0	0	0	0	0	1.0000	-360.0000	360.0000
12.0000	13.0000	0.0916	0.0721	0	0	0	0	0	0	1.0000	-360.0000	360.0000
13.0000	14.0000	0.0338	0.0445	0	0	0	0	0	0	1.0000	-360.0000	360.0000
14.0000	15.0000	0.0369	0.0328	0	0	0	0	0	0	1.0000	-360.0000	360.0000
15.0000	16.0000	0.0466	0.0340	0	0	0	0	0	0	1.0000	-360.0000	360.0000
16.0000	17.0000	0.0804	0.1074	0	0	0	0	0	0	1.0000	-360.0000	360.0000
17.0000	18.0000	0.0457	0.0358	0	0	0	0	0	0	1.0000	-360.0000	360.0000
2.0000	19.0000	0.0102	0.0098	0	0	0	0	0	0	1.0000	-360.0000	360.0000
19.0000	20.0000	0.0939	0.0846	0	0	0	0	0	0	1.0000	-360.0000	360.0000
20.0000	21.0000	0.0255	0.0298	0	0	0	0	0	0	0	-360.0000	360.0000
21.0000	22.0000	0.0442	0.0585	0	0	0	0	0	0	1.0000	-360.0000	360.0000
3.0000	23.0000	0.0282	0.0192	0	0	0	0	0	0	1.0000	-360.0000	360.0000
23.0000	24.0000	0.0560	0.0442	0	0	0	0	0	0	0	-360.0000	360.0000
24.0000	25.0000	0.0559	0.0437	0	0	0	0	0	0	1.0000	-360.0000	360.0000
6.0000	26.0000	0.0127	0.0065	0	0	0	0	0	0	1.0000	-360.0000	360.0000
26.0000	27.0000	0.0177	0.0090	0	0	0	0	0	0	0	-360.0000	360.0000
27.0000	28.0000	0.0661	0.0583	0	0	0	0	0	0	1.0000	-360.0000	360.0000
28.0000	29.0000	0.0502	0.0437	0	0	0	0	0	0	1.0000	-360.0000	360.0000
29.0000	30.0000	0.0317	0.0161	0	0	0	0	0	0	1.0000	-360.0000	360.0000
30.0000	31.0000	0.0608	0.0601	0	0	0	0	0	0	1.0000	-360.0000	360.0000
31.0000	32.0000	0.0194	0.0226	0	0	0	0	0	0	1.0000	-360.0000	360.0000
32.0000	33.0000	0.0213	0.0331	0	0	0	0	0	0	1.0000	-360.0000	360.0000
21.0000	8.0000	0.1248	0.1248	0	0	0	0	0	0	0	-360.0000	360.0000
12.0000	22.0000	0.1248	0.1248	0	0	0	0	0	0	0	-360.0000	360.0000
18.0000	33.0000	0.0312	0.0312	0	0	0	0	0	0	0	-360.0000	360.0000
25.0000	29.0000	0.0312	0.0312	0	0	0	0	0	0	0	-360.0000	360.0000

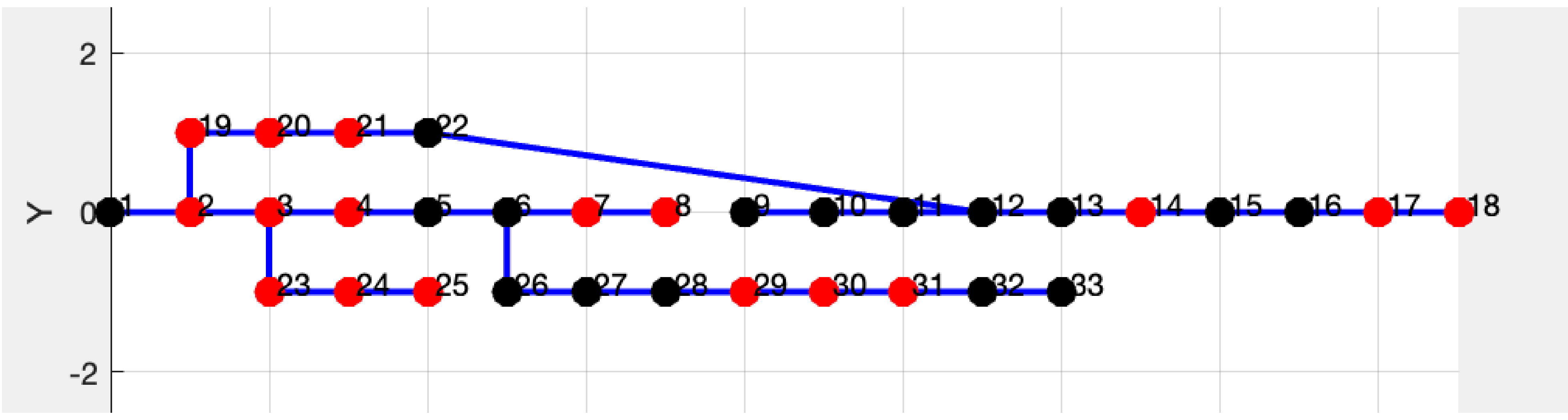
Single Fault

line: 8-9

During Fault



After Fault



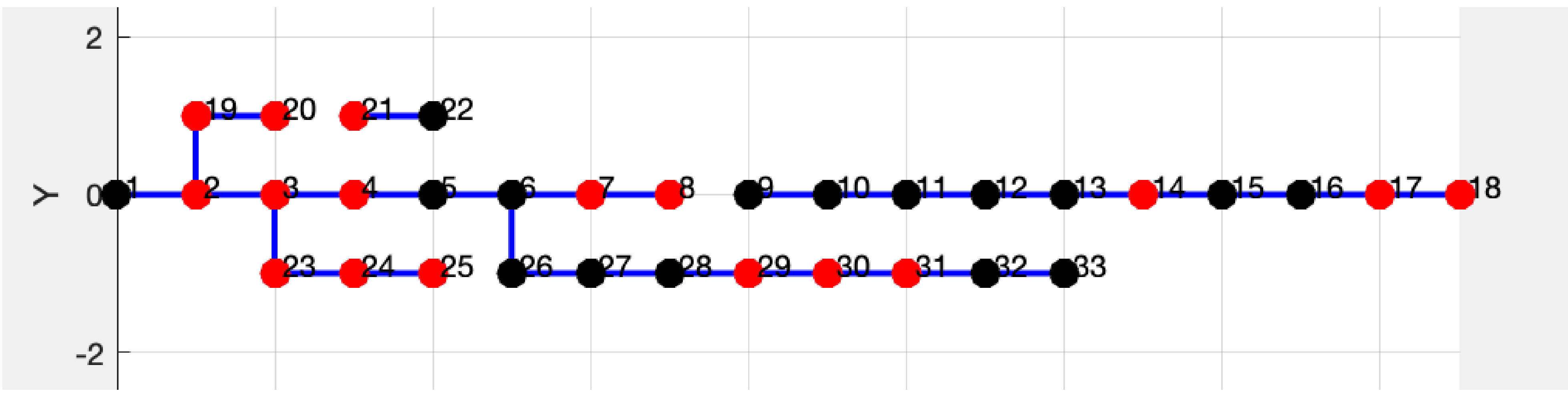

```
total_loadshed =

    0.6750


total_loadshedmin =

     0
```

```
MATPOWER Version 8.0, 17-May-2024
Power Flow -- AC-polar-power formulation

Newton's method converged in 3 iterations.
PF successful

Converged in 0.26 seconds
================================================================================
|     System Summary                                                           |
================================================================================

How many?                How much?              P (MW)            Q (MVAr)
---------------------    -------------------  -------------  -----------------
Buses             33     Total Gen Capacity      10.0          -10.0 to 10.0
Generators         1     On-line Capacity        10.0          -10.0 to 10.0
Committed Gens     1     Generation (actual)      3.9                2.4
Loads             32     Load                     3.7                2.3
  Fixed           32       Fixed                  3.7                2.3
  Dispatchable     0       Dispatchable          -0.0 of -0.0       -0.0
Shunts             0     Shunt (inj)             -0.0                0.0
Branches          37     Losses (I^2 * Z)         0.15               0.11
Transformers       0     Branch Charging (inj)     -                 0.0
Inter-ties         0     Total Inter-tie Flow     0.0                0.0
Areas              1
```

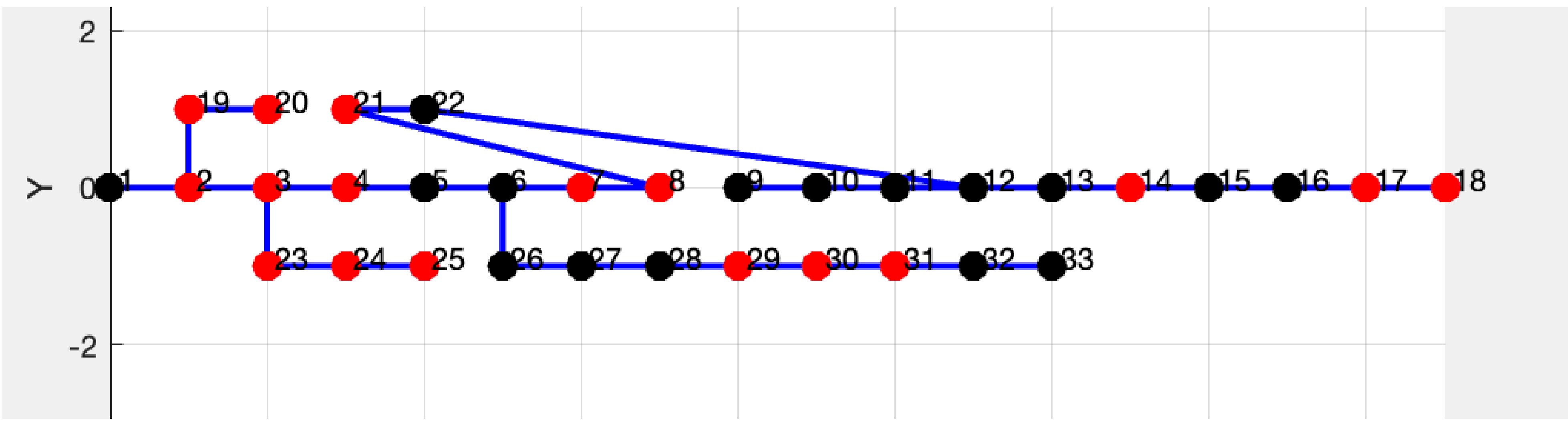
Double Fault

line : 8-9 , 20-21

During Fault



After Fault

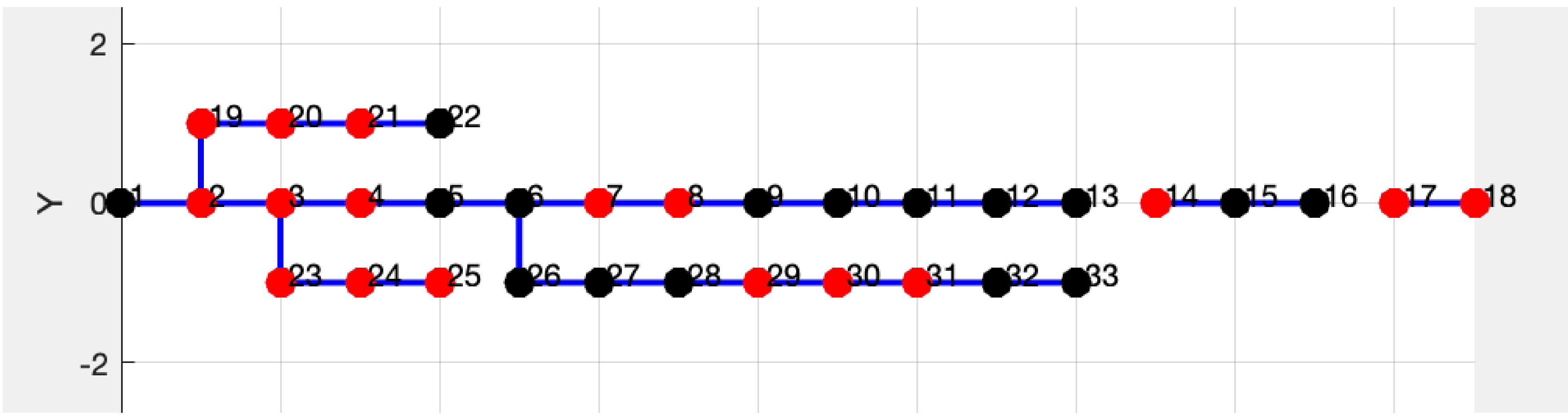



```
MATPOWER Version 8.0, 17-May-2024
Power Flow -- AC-polar-power formulation

Newton's method converged in 4 iterations.
PF successful

Converged in 0.33 seconds
================================================================================
|     System Summary                                                           |
================================================================================

How many?                How much?              P (MW)            Q (MVAr)
---------------------    -------------------  -------------  -----------------
Buses              33    Total Gen Capacity      10.0         -10.0 to 10.0
Generators          1    On-line Capacity        10.0         -10.0 to 10.0
Committed Gens      1    Generation (actual)      4.0               2.5
Loads              32    Load                     3.7               2.3
  Fixed            32      Fixed                  3.7               2.3
  Dispatchable      0      Dispatchable        -0.0 of -0.0        -0.0
Shunts              0    Shunt (inj)             -0.0               0.0
Branches           37    Losses (I^2 * Z)         0.24              0.17
Transformers        0    Branch Charging (inj)     -                0.0
Inter-ties          0    Total Inter-tie Flow     0.0               0.0
Areas               1
```

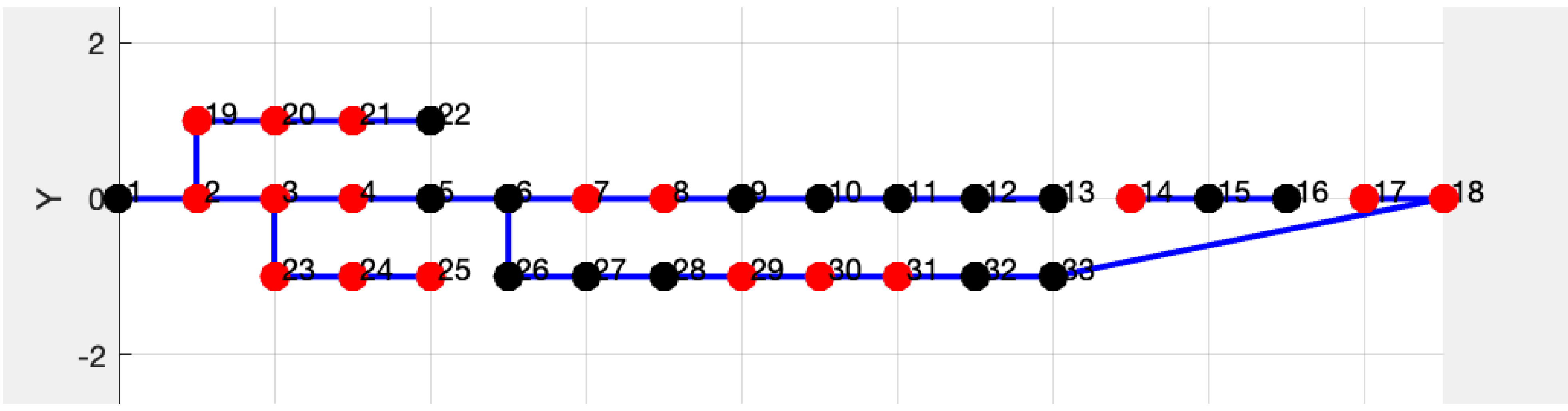
```
total_loadshed =

   0.8550

total_loadshedmin =

    0
```

line : 13-14 , 16-17

During Fault



After Fault



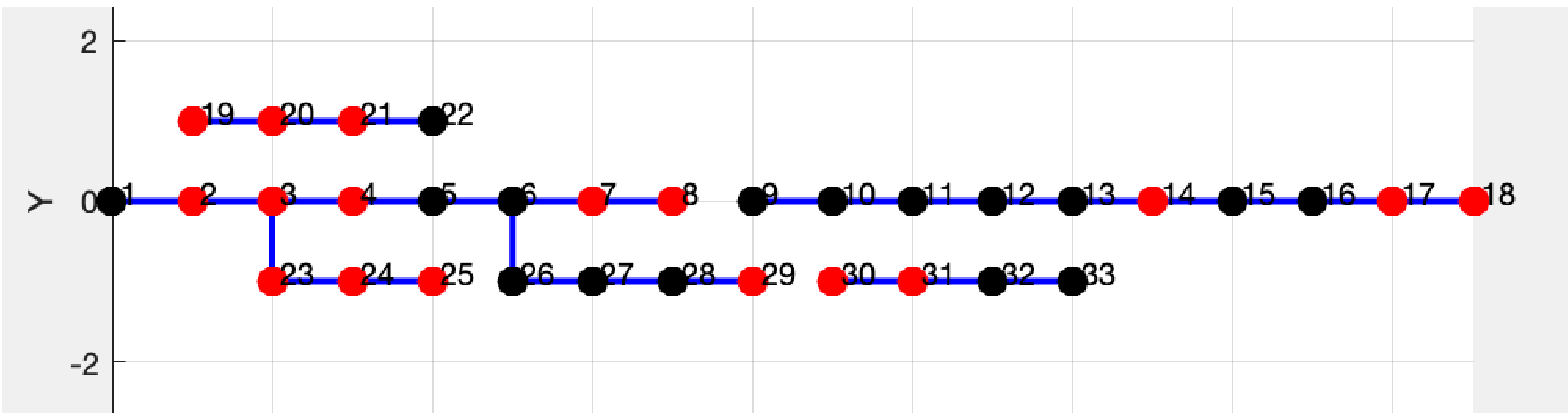

```
Newton's method did not converge in 10 iterations.
PF failed

>>>>>  Did NOT converge (0.36 seconds)  <<<<<
```

```
total_loadshed =

    0.3900


total_loadshedmin =

    0.2400
```

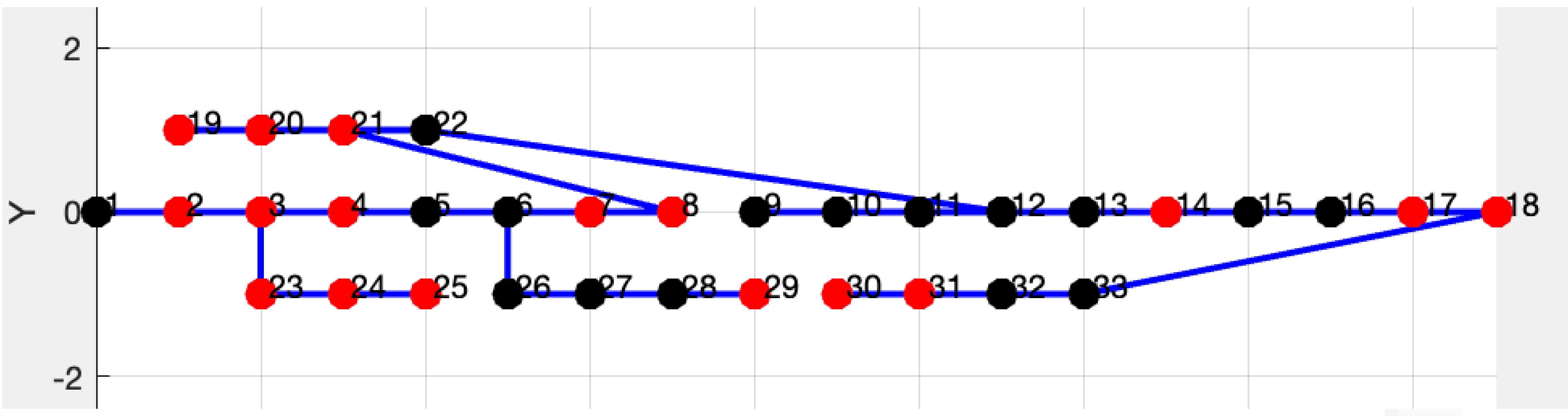
Triple Fault

line : 2-19, 8-9, 29-30

During Fault



After Fault



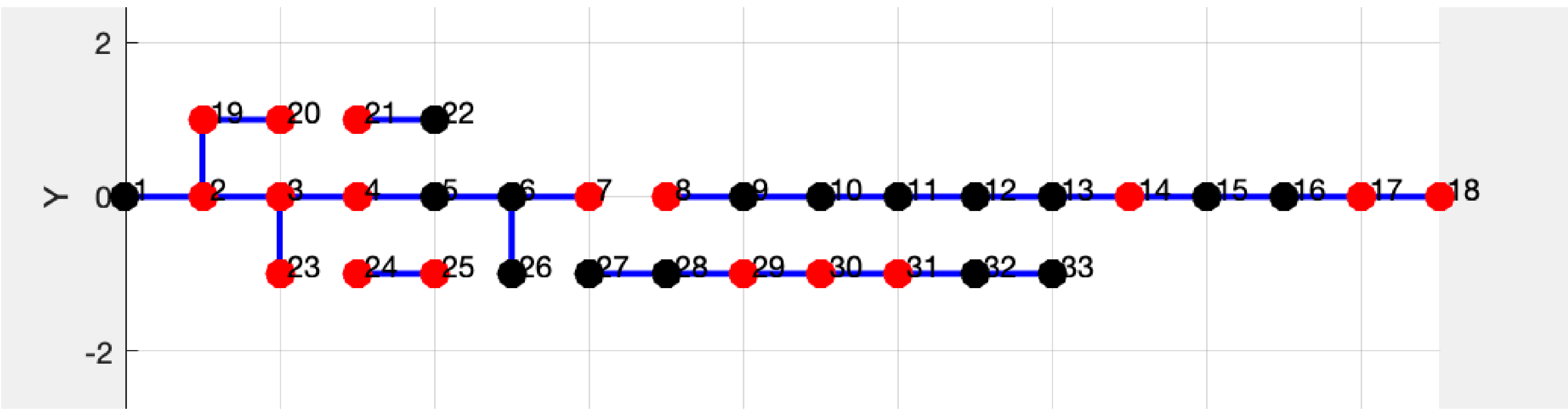

```
total_loadshed =

   1.6550

total_loadshedmin =

   0
```

```
MATPOWER Version 8.0, 17-May-2024
Power Flow -- AC-polar-power formulation

Newton's method converged in 5 iterations.
PF successful

Converged in 0.30 seconds
================================================================================
|     System Summary                                                           |
================================================================================

How many?                How much?              P (MW)            Q (MVAr)
---------------------    -------------------  -------------  -----------------
Buses             33     Total Gen Capacity     10.0          -10.0 to 10.0
Generators         1     On-line Capacity       10.0          -10.0 to 10.0
Committed Gens     1     Generation (actual)     4.3               2.8
Loads             32     Load                    3.7               2.3
  Fixed           32       Fixed                 3.7               2.3
  Dispatchable     0       Dispatchable         -0.0 of -0.0      -0.0
Shunts             0     Shunt (inj)            -0.0               0.0
Branches          37     Losses (I^2 * Z)        0.62              0.54
Transformers       0     Branch Charging (inj)     -               0.0
Inter-ties         0     Total Inter-tie Flow    0.0               0.0
Areas              1
```

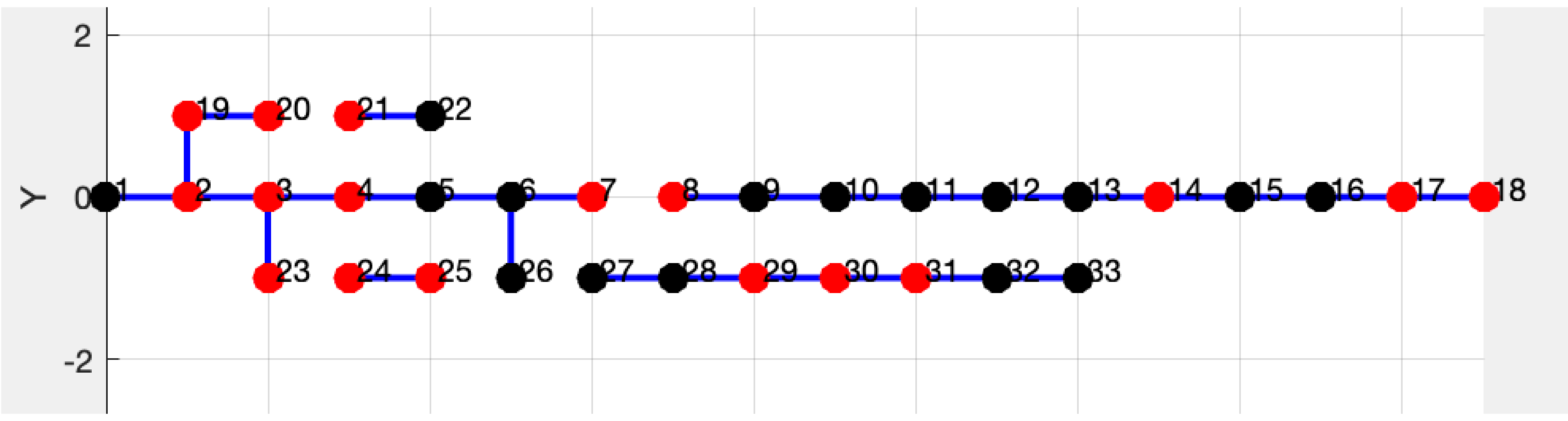
Four line Fault

line : 7-8, 20-21, 23-24, 26-27

During Fault



After Fault



```
Newton's method did not converge in 10 iterations.
PF failed

>>>>>  Did NOT converge (0.74 seconds)  <<<<<
```

```
total_loadshed =

    2.7550


total_loadshedmin =

    2.7550
```

Discussion

- In the case of the modified IEEE 33-bus system with three line outages, the network was successfully reconfigured to minimize load shedding to zero while maintaining a radial topology.

- In the case of four line outages, as observed from the referenced research paper, the system cannot be fully reconfigured because it would form two separate subsystems. Therefore, the load shedding remains the same even after attempted reconfiguration.

- During testing, it was observed that the modified IEEE 33-bus system showed errors while running the power flow analysis (Runpf) in MATLAB.

- This is likely due to insufficient or missing data in the research paper about the modified system, which forced us to make certain assumptions during its creation. These assumptions could have caused errors during the power flow analysis.

- To verify the correctness of our reconfiguration strategy, we tested it on the original IEEE 33-bus system, where we were able to successfully perform reconfiguration and power flow analysis without any errors.

Conclusion

We have successfully completed the reconfiguration of the modified IEEE 33-bus system. This reconfiguration can be used to minimize load shedding as much as possible for any number of faults. We have used DFS (Depth-First Search) to identify isolated buses in the system and developed our own reconfiguration algorithm in this project. Additionally, we have minimized the time complexity, allowing the algorithm to produce results within seconds. Through this project, we had the opportunity to apply our theoretical knowledge to solve practical problems.

Modification

The referenced paper primarily addressed the reconfiguration of the modified IEEE 33-bus system but did not incorporate any optimization techniques. To address this gap, we developed our own optimization approach for network reconfiguration using a brute-force method, based on the analysis of tie switches. This constituted the first major improvement over the original study. Furthermore, in their initial modified IEEE 33-bus system, the power flow analysis was not functioning as described. Consequently, we applied our optimization method to the standard IEEE 33-bus system, performed the reconfiguration accordingly, and successfully obtained the desired results.

References

[1] Y. Chen et al., "Optimal Network Reconfiguration and Scheduling With Hardware-in-the-Loop Validation for Improved Microgrid Resilience," in IEEE Access, vol. 13, pp. 8042-8059, 2025, doi: 10.1109/ACCESS.2025.3527329.