NOVA

Power / Energy Track

A review of selected optimal power flow literature to 1993.
I. Nonlinear and Quadratic Programming Approaches

JA Momoh, R Adapa, ME EI-Hawary

IEEE Transactions on Power Systems (2002)

숭실대학교 전기공학부 학술 소모임 NOVA
발표자 : 유나현

Contents

01 논문의 핵심 목적

본 논문은 OPF(Optimal Power Flow) 문제를 어떻게 풀어왔는지 역사적으로 정리하고 방법들을 비교한다.

02 OPF

OPF = 전력 시스템을 가장 효율적으로 운용하는 문제, 제약조건 내에서 최적의 값을 찾는 것

02 OPF

OPF = 전력 시스템을 가장 효율적으로 운용하는 문제, 제약조건 내에서 최적의 값을 찾는 것

OPF 구조

=

목적함수

- 발전 비용 최소화
- 전력 손실 최소화
- 전압 안정성 유지

+

제약조건

- 전력 수지 (P,Q balance)
- 전압 제한
- 송전선 용량 제한
- 발전기 출력 한계

03 OPF의 해결방법 분류 - Nonlinear Programming (NLP)

Nonlinear Programming (NLP)

- 가장 현실을 고려한 방법
- 목적함수와 제약조건이 비선형
- 정확도 높음, 계산 어려움(수렴의 문제)

Nonlinear Programming (NLP)를 풀기위한 최적화 기법 예시

Lagrange Multiplier (라그랑주 승수법) : 제약조건을 목적함수에 합치는 방법

SUMT (Sequential Unconstrained Minimization Technique)

Quasi-Newton Method

03 OPF의 해결방법 분류 - Nonlinear Programming (NLP)

Nonlinear Programming (NLP)

- 가장 현실을 고려한 방법
- 목적함수와 제약조건이 비선형
- 정확도 높음, 계산 어려움(수렴의 문제)

Nonlinear Programming (NLP)를 풀기위한 최적화 기법 예시

Lagrange Multiplier (라그랑주 승수법) : 제약조건을 목적함수에 합치는 방법

목적함수: f(x)

제약조건: g(x) = 0

$$\mathcal{L}(x, \lambda) = f(x) + \lambda g(x)$$

*라그랑주 승수 : 제약조건이 목적함수에 얼마나 영향을 주는지를 나타낸다.

SUMT (Sequential Unconstrained Minimization Technique)

Quasi-Newton Method

03 OPF의 해결방법 분류 - Nonlinear Programming (NLP)

Nonlinear Programming (NLP)

- 가장 현실을 고려한 방법
- 목적함수와 제약조건이 비선형
- 정확도 높음, 계산 어려움(수렴의 문제)

Nonlinear Programming (NLP)를 풀기위한 최적화 기법 예시

Lagrange Multiplier (라그랑주 승수법) : 제약조건을 목적함수에 합치는 방법

SUMT (Sequential Unconstrained Minimization Technique)

: 제약조건조건 문제를 제약이 없는 문제로 바꾸어 반복적으로 푸는 방법

$$F(x) = f(x) + r \cdot \text{Penalty}(g(x))$$

*penalty : 제약을 위반하면 큰 값을 준다.

*r : penalty 계수로, 점점 크게 증가한다.

Quasi-Newton Method

03 OPF의 해결방법 분류 - Nonlinear Programming (NLP)

Nonlinear Programming (NLP)

- 가장 현실을 고려한 방법
- 목적함수와 제약조건이 비선형
- 정확도 높음, 계산 어려움(수렴의 문제)

Nonlinear Programming (NLP)를 풀기위한 최적화 기법 예시

Lagrange Multiplier (라그랑주 승수법) : 제약조건을 목적함수에 합치는 방법

+

SUMT (Sequential Unconstrained Minimization Technique)

+

Quasi-Newton Method : Newton-Raphson Method 를 간단하게 만든 방법

Newton-Raphson Method은 정확하지만 Hessian 계산이 매우 비쌈.

-> Hessian approximation을 이용해서 최적점을 찾아간다.

*Hessinn : 최적화 함수에서 함수의 곡률을 나타내는 2차 미분 행렬 -> 엄청난 계산량

* Hessian approximation : Hessian 대신 근사 행렬을 만들어 곡률을 추정하는 방법

03 OPF의 해결방법 분류 -Quadratic Programming (QP)

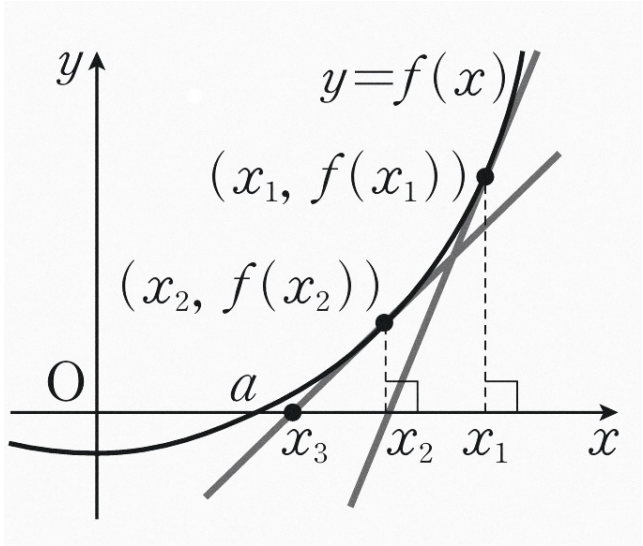
Quadratic Programming (QP)

- NPL의 한 종류
- 목적함수 : 2차 함수, 제약조건 : 선형
- 계산이 상대적으로 쉬움

03 OPF의 해결방법 분류 - Newton

Newton

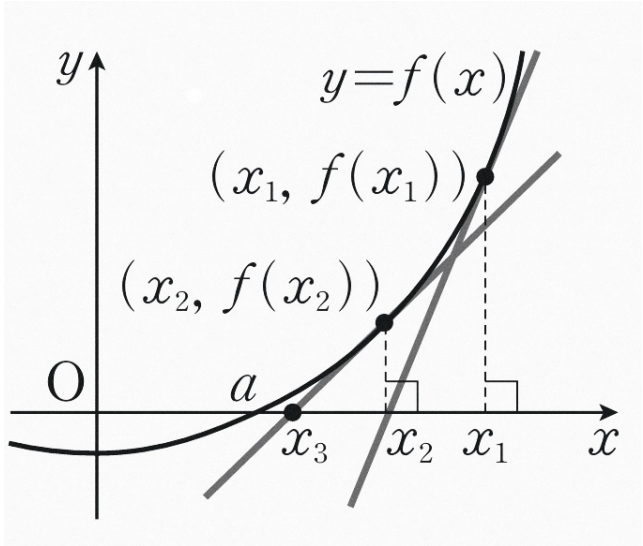
- 최적 조건 (kkt 조건) 을 직접 풂
- 매우 빠른 수렴
- 초기값 민감
- 계산복잡



03 OPF의 해결방법 분류 - Newton

Newton

- 최적 조건 (kkt 조건) 을 직접 풂
- 매우 빠른 수렴
- 초기값 민감
- 계산복잡



KKT 조건 : 제약조건이 있는 최적화 문제에서 최적해가 되기 위한 수학적 조건

ex) 발전기 최대 출력이 A이다

- A > 출력 : 제약은 영향이 없음
- A< 출력 : 그 제약은 영향 없음

03 OPF의 해결방법 분류 - Linear Programming (LP)

Linear Programming (LP)

- 문제를 선형으로 근사해서 풂
- 계산 빠름
- 정확도 떨어짐

OPF는 전압과 위상각이 곱과 삼각함수 형태로 얽혀있는 비선형 형태이다.

$$P_i = \sum_{j=1}^{n} V_i V_j (G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij})$$

$$Q_i = \sum_{j=1}^{n} V_i V_j (G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij})$$

→ 변수 (V , θ)의 곱의 형태이다.

→ 삼각함수가 포함된 형태이다.

03 OPF의 해결방법 분류 -Mixed Integer Programming (MIP)

Mixed Integer Programming (MIP)

- 정수 변수가 포함된 함수로 바꾸어 푼다.
- 계산이 매우 어려움

실제 전역시스템에는 연속값으로 표현할 수 없는 장치가 존재한다.

Ex) 스위치 상태 (ON = 1 , OFF = 0)

발전기 상태 (ON = 1 , OFF = 0)

변압기 탭 위치 (Tap = {0.9 , 1.0 , 1.1}

04 OPF의 시간 흐름에 따른 연구 발전

1960 ~ 1970 년대

- OPF 개념 등장
- NLP 방식을 도입
- 작은 시스템만 적용 가능

→ “이론 정립 단계”

1970~1980년대

- Newton, Quasi-Newton 등장
- 계산 속도 개선 시도
- IEEE 30-bus 시스템 사용

→ “계산 방법 발전”

1980년대

-1000 bus 이상 대규모 시스템

- 분해법 (decomposition)
- P-Q 분리 방법 등장

→ “실제 시스템 적용 시도”

1990년대 직전

QP, NLP 혼합

실시간 OPF 등장

계산 효율성 개선

→ “실무 적용 단계”

05 현재 OPF 방법의 단점

NLP

➡ 큰 시스템에서 느리다.

Quasi - Newton

➡ 과거에는 실제 연구에 사용되었지만,
큰 시스템에서 비효율을 보인다.

SUMT

➡ Penalty 값이 커지면 문제가 발생

“ 기존 방법에는 한계가 있고, 더 좋은 방법이 필요하다. " ➡ 논문 PART 2

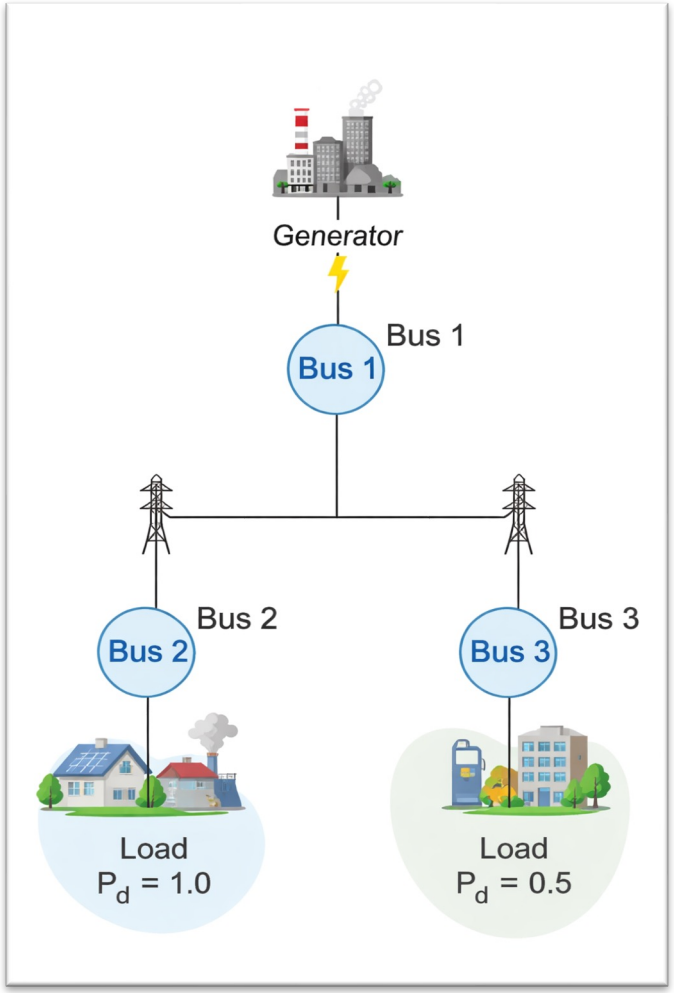
06 향후 연구 방향

“ 3개의 노드로 구성된 전력망에서 1개의 발전기가 2개의 부하를 공급하면서 전력흐름을 만족하고 발전비용이 최소가 되도록 하는 시스템을 MATLAB 으로 구현해보도록 한다. “

```
clear; clc;
%% ------------------------------
% 시스템 데이터 (3-bus)
% ------------------------------
% Ybus (간단한 예시)
Y = [10-30i  -5+15i  -5+15i;
     -5+15i  10-30i  -5+15i;
     -5+15i  -5+15i  10-30i];
G = real(Y);
B = imag(Y);
nbus = 3;
%% ------------------------------
% 초기값
% ------------------------------
V = ones(nbus,1);           % 전압 크기
theta = zeros(nbus,1);      % 위상각
Pg = 1.5;   % 발전기 출력 초기값
Pd = [0; 1; 0.5];  % 부하
%% 비용 함수 계수 (aP^2 + bP + c)
a = 0.1; b = 1; c = 0;
%% 반복 설정
max_iter = 20;
tol = 1e-6;
%% ------------------------------
% Newton-Raphson OPF
% ------------------------------
for iter = 1:max_iter

    % ----------------------
    % 전력 계산 (P mismatch)
    % ----------------------
    P = zeros(nbus,1);

    for i = 1:nbus
        for j = 1:nbus
            P(i) = P(i) + V(i)*V(j)*...
                (G(i,j)*cos(theta(i)-theta(j)) + ...
                 B(i,j)*sin(theta(i)-theta(j)));
        end
    end

    % 발전 - 부하
    P_spec = [Pg; 0; 0] - Pd;

    dP = P_spec - P;
```

```
    % slack bus 제거 (bus 1)
    mismatch = dP(2:end);

    % ----------------------
    % Jacobian 계산
    % ----------------------
    J = zeros(nbus-1);

    for i = 2:nbus
        for j = 2:nbus

            if i == j
                for k = 1:nbus
                    J(i-1,j-1) = J(i-1,j-1) + ...
                        V(i)*V(k)*(-G(i,k)*sin(theta(i)-theta(k)) + ...
                                    B(i,k)*cos(theta(i)-theta(k)));
                end
            else
                J(i-1,j-1) = V(i)*V(j)*...
                    (G(i,j)*sin(theta(i)-theta(j)) - ...
                     B(i,j)*cos(theta(i)-theta(j)));
            end

        end
    end

    % ----------------------
    % Newton 업데이트
    % ----------------------
    dtheta = J \ mismatch;
    theta(2:end) = theta(2:end) + dtheta;

    % ----------------------
    % 발전량 업데이트 (간단 gradient)
    % ----------------------
    dCost_dPg = 2*a*Pg + b;
    Pg = Pg - 0.01 * dCost_dPg;

    % ----------------------
    % 수렴 체크
    % ----------------------
    if norm(mismatch) < tol
        disp(['Converged in ', num2str(iter), ' iterations']);
        break;
    end
end
```

```
end
%% ------------------------------
% 결과 출력
% ------------------------------
disp('--- 결과 ---');
disp(['Pg = ', num2str(Pg)]);
disp('Voltage Magnitude:');
disp(V);
disp('Voltage Angle:');
disp(theta);
cost = a*Pg^2 + b*Pg + c;
disp(['Total Cost = ', num2str(cost)]);
```



NOVA

Power / Energy Track

Thank you

숭실대학교 전기공학부 학술 소모임 NOVA
발표자 : 유나현