OPTIMAL POWER FLOW ALGORITHMS

Hans Glavitsch, Rainer Bacher

Swiss Federal Institute of Technology
CH-8092 Zürich, Switzerland

1 PROBLEM DEFINITION

1.1 Optimal power flow problem

1.1.1 The ordinary power flow

The ordinary power flow or load flow problem is stated by specifying the loads in megawatts and megavars to be supplied at certain nodes or busbars of a transmission system and by the generated powers and the voltage magnitudes at the remaining nodes of this system together with a complete topological description of the system including its impedances. The objective is to determine the complex nodal voltages from which all other quantities like line flows, currents and losses can be derived. The model of the transmission system is given in complex quantities since an alternating current system is assumed to generate and supply the powers and loads.

In mathematical terms the problem can be reduced to a set of nonlinear equations where the real and imaginary components of the nodal voltages are the variables. The number of equations equals twice the number of nodes. The nonlinearities can roughly be classified being of a quadratic nature. Gradient and relaxation techniques are the only methods for the solution of these systems.

The result of a power flow problem tells the operator or a planner of a system in which way the lines in the system are loaded, what the voltages at the various buses are, how much of the generated power is lost and where limits are exceeded.

The power flow problem is one of the basic problems in which both load powers and generator powers are given or fixed. Today, this basic problem can be efficiently handled on the computer for practically any size system.

1.1.2 The optimal power flow

For the planner and operator fixed generation corresponds to a snapshot only. Planning and operating requirements very often ask for an adjustment of the generated powers according to certain criteria. One of the obvious ones is the minimum of the generating cost. The application of such a criterion immediately assumes variable input powers and bus voltages which have to be determined in such a way that a minimum of the cost of generating these powers is achieved.

At this point it is not only the voltages at nodes where the loads are supplied but also the input powers together with the corresponding voltages at the generator nodes which have to be determined. The degree of freedom for the choice of inputs seems to be exceedingly large, but due to the presence of an objective, namely to reach the minimum of the generating cost the problem is well defined. Of course the mathematics become more demanding as compared to the original power flow problem, however, the aim still being the same, i.e. the determination of the nodal voltages in the system. They play the role of state variables from which all other quantities can be derived.

It turns out that the extended problem requires a more detailed definition and different methods of solution.

The problem can be generalized by attaching different objectives to the original power flow problem. As long as the power flow model stays the same it is considered the optimal power flow problem where the objective is a scalar function of the state variables. In essence, any optimal power flow problem can be reduced to such a form.

Now, practical requirements ask for a more realistic definition, the main addition being the statement of constraints. In the real world any variable in the system will be limited which changes the mathematical nature of the problem drastically. Whenever a variable reaches its upper or lower limit it becomes a fixed quantity and the method of solution has to recognize it as such and be sure that the fixed quantity is optimal.

Fortunately, the theory developed by Kuhn and Tucker [1] is able to provide the optimality conditions which guarantee the correctness of the result in the end. However, the optimality conditions do not offer a solution method.

Present requirements are aimed at solution methods suitable for computer implementations which are easy to handle, capable of large systems, have good convergence and are fast. Experience shows that the performance of solution methods in the power system analysis area are dependent on the nature of the system model, on the type of nonlinearities, on the type of constraints, on the number of constraints, etc.

Thus, the basic theory of optimization contribute a small part to the success of a solution method only. It is the genius of the system analyst and of the computer scientist which becomes the key factor for the success of a method.

Optimal power flow algorithms are the outcome of development work of this kind and are determinant for the performance of whole classes of programs. Hence it is worthwhile and quite rewarding to engage in the investigation of algorithms within this problem class.

Scanning through the literature [2], [3], [4], [5], [7], [9] it will be observed that there are many attempts to describe, define, formulate and solve the optimal power flow problem. However, it seems that successful solutions emerged only at the point where proven schemes of optimization such as linear and quadratic programming could be applied to this very problem [8], [10], [11]. This late development was supported by other techniques which proved useful in the area of the ordinary power flow such as the exploitation of sparsity and Newtons's method.

Thus, in the subsequent sections great emphasis will be placed on a thorough formulation of the optimal power flow problem and on techniques which lend themselves to an application of proven optimization methods.

1.2 Power flow simulation of an electrical power transmission system

This subsection discusses briefly the basics for the simulation of an electrical power transmission system on a digital computer. More information can be obtained from many textbooks which discuss the basic power flow problem in more detail.

1.2.1 Nodal current - nodal voltage relationship

The relation between the complex nodal voltages $\underline{\mathbf{V}}$ and the complex nodal currents $\underline{\mathbf{I}}$ of the transmission network, composed of the passive components, transmission lines, series elements, transformers and shunts is:

$$\underline{\mathbf{I}} = \underline{\mathbf{Y}} \cdot \underline{\mathbf{V}} \tag{1}$$

Every complex nodal current $\underline{I}_i$ can be formulated in rectangular coordinates:

$$\underline{I}_i = I_{e_i} + j \cdot I_{f_i} \quad ; \text{i} = 1...N; \; N = \text{number of electrical nodes} \tag{2}$$

For every complex nodal voltage $\underline{V}_i$, the following is valid in rectangular coordinates for the complex nodal voltage:

$$\underline{V}_i = e_i + j \cdot f_i \quad ; \text{i} = 1...N \tag{3}$$

Note that usually at one node the angle of the complex voltage is held constant. Thus the following relationship must be valid for this one node, called the slack node:

$$\frac{f_{slack}}{e_{slack}} = k_{slack} = constant \tag{4}$$

Note that very often this constant value k_{slack} is assumed to be zero, i.e. the voltage angle at this node is assumed to be zero. However, in this paper the general case of (4) is assumed to be valid.

The complex elements at row i and column j of the matrix $\underline{\mathbf{Y}}$ are as follows:

$$\underline{Y}_{ij} = g_{ij} + j \cdot b_{ij} \tag{5}$$

or in polar form

$$\underline{Y}_{ij} = y_{ij} \cdot (cos\theta_{ij} + j \cdot sin\theta_{ij}) \tag{6}$$

It follows from (1), (2) and (5)

$$I_{e_i} = \sum_{j=1}^{N} (e_j g_{ij} - f_j b_{ij}) \quad ; \text{i} = 1...N \tag{7}$$

$$I_{f_i} = \sum_{j=1}^{N} (e_j b_{ij} + f_j g_{ij}) \quad ; \text{i} = 1...N \tag{8}$$

In polar coordinates the complex voltages $\underline{V}_i$ are defined as follows:

$$\underline{V}_i = |V|_i \cdot (cos\Theta + j \cdot sin\Theta) \quad ; \text{i} = 1...N \tag{9}$$

As defined in (4), the voltage angle at the so-called slack node is kept fixed:

$$\Theta_{slack} = \arctan(k_{slack}) = constant \tag{10}$$

It should be noted that other network components like DC-transmission lines are not included in this paper. Balanced three-phase network operation is assumed.

1.2.2 Nodal power nodal voltage - nodal current relationship

In this paper in order to make certain derivations easier to understand, the following assumptions are made with respect to node numbering:

- The network has a total of N electrical nodes
- The l load PQ-nodes are numbered $1...l$
- The m generator PV-nodes are numbered $(l+1)...(l+m)$
- $l+m$=N
- The last generator node is called the slack node (i.e. the slack node number is N).

Note that the above mentioned slack node is usually treated as a normal PV-generator bus with the additional constraint of a fixed voltage angle (see (4) and (10)).

The active and reactive powers of all l PQ-load-nodes must be computed by the following relationship:

$$P_i = Real\{\underline{V}_i \cdot \underline{I}_i^*\} \;\; ; \text{i} = 1...l \tag{11}$$

$$Q_i = Imag\{\underline{V}_i \cdot \underline{I}_i^*\} \;\; ; \text{i} = 1...l \tag{12}$$

(11), (12) formulated in **rectangular coordinates**:
For all l PQ-nodes:

$$P_i = e_i I_{e_i} + f_i I_{f_i} \;\; ; \text{i} = 1...l \tag{13}$$

$$Q_i = f_i I_{e_i} - e_i I_{f_i} \;\; ; \text{i} = 1...l \tag{14}$$

For all m PV-nodes:

$$P_i = e_i I_{e_i} + f_i I_{f_i} \;\; ; \text{i} = l+1...N \tag{15}$$

$$|V|_i^2 = e_i^2 + f_i^2 \quad ; \text{i} = l+1...N \tag{16}$$

Inserting (7) and (8) into (13) and (14) yields:

$$P_i = \sum_{j=1}^{N} (e_i(e_j g_{ij} - f_j b_{ij}) + f_i(f_j g_{ij} + e_j b_{ij})) \quad ; \text{i} = 1...l \tag{17}$$

$$Q_i = \sum_{j=1}^{N} (f_i(e_j g_{ij} - f_j b_{ij}) - e_i(f_j g_{ij} + e_j b_{ij})) \quad ; \text{i} = 1...l \tag{18}$$

For the generator PV-nodes the active power P and the voltage magnitude are computed as follows:

$$P_i = \sum_{j=1}^{N} (e_i(e_j g_{ij} - f_j b_{ij}) + f_i(f_j g_{ij} + e_j b_{ij})) \quad ; \text{i} = l+1...N \tag{19}$$

$$|V|_i^2 = e_i^2 + f_i^2 \quad ; \text{i} = l+1...N \tag{20}$$

(11), (12) formulated in **polar coordinates**:
For all l PQ nodes:

$$P_i = \sum_{j=1}^{N} (V_i V_j y_{ij} cos(\Theta_i - \Theta_j - \theta_{ij})) \quad ; \text{i} = 1...l \tag{21}$$

$$Q_i = \sum_{j=1}^{N} (V_i V_j y_{ij} sin(\Theta_i - \Theta_j - \theta_{ij})) \quad ; \text{i} = 1...l \tag{22}$$

For all m PV nodes (inclusive slack node):

$$P_i = \sum_{j=1}^{N} (V_i V_j y_{ij} cos(\Theta_i - \Theta_j - \theta_{ij})) \quad ; \text{i = l+1 ... N} \tag{23}$$

$$|V|_i = V_i \quad ; \text{i = l+1 ... N} \tag{24}$$

Note that (24) is trivial and in principle not necessary. The equations of (24) are omitted in the following derivations when using the polar coordinate system.

1.2.3 Operational limits

In the real power system many of the variables used in the above equations are limited and may not be exceeded without damaging equipment or bringing the network into unstable, insecure operating states:

- Limits on active power of a (generator) PV node:

$$P_{low_i} \leq P_{PV_i} \leq P_{high_i} \tag{25}$$

- Limits on voltage of a PV or PQ node:

$$|V|_{low_i} \leq |V|_i \leq |V|_{high_i} \tag{26}$$

- Limits on tap positions of a transformer

$$t_{low_i} \leq t_i \leq t_{high_i} \tag{27}$$

- Limits on phase shift angles of a transformer

$$\theta_{low_i} \leq \theta_i \leq \theta_{high_i} \tag{28}$$

- Limits on shunt capacitances or reactances

$$s_{low_i} \leq s_i \leq s_{high_i} \tag{29}$$

- Limits on reactive power generation of a PV node

$$Q_{low_i} \leq Q_{PV_i} \leq Q_{high_i} \tag{30}$$

  In reality the reactive limits on a generator are complex and usually state dependent. (30) is a simplification of the limits, however, by adapting the actual limit values during the optimization, the real-world limits can be simulated with sufficient accuracy.

- Upper limits on active power flow in transmission lines or transformers:

$$P_{ij} \leq P_{high_{ij}} \tag{31}$$

- Upper limits on MVA flows in transmission lines or transformers

$$P_{ij}^2 + Q_{ij}^2 \leq S_{high_{ij}}^2 \tag{32}$$

- Upper limits on current magnitudes in transmission lines or transformers

$$|I|_{ij} \leq |I|_{high_{ij}} \tag{33}$$

- Limits on voltage angles between nodes:

$$\Theta_{low_{ij}} \leq \Theta_i - \Theta_j \leq \Theta_{high_{ij}} \tag{34}$$

- Limits on total flows between areas

  These inequality constraints can be formulated for MVA-, and MW-values as follows:

  - Limits on active power area flows

$$P_{low_{area_a}} \leq \sum_{\text{a to b}} P_{ab} \leq P_{high_{area_a}} \tag{35}$$

  - Limits on MVA area power flows

$$S_{low_{area_a}}^2 \leq \sum_{\text{a to b}} (P_{ab}^2 + Q_{ab}^2) \leq S_{high_{area_a}}^2 \tag{36}$$

1.2.4 Summary

It is an essential goal of the network operator to have all of above mentioned inequality constraints, representing real world operating limits, under control. The power demand which must be in balance with the generation is automatically considered in the real system. Any simulation, i.e. also the OPF, must consider this equality constraint unconditionally in order to simulate the real power system correctly.

It must be noted that not in all networks all these constraints have the same degree of importance. However, in general, and this is assumed in the formulations of this paper, all these constraints have to be satisfied. Thus, any electrical network simulation result, also the one of an OPF simulation, should observe the above operational limits in its final result.

The mathematical model must always consider the equations (1), (11) and (12), i.e. the relation between nodal voltages, currents and nodal powers must be considered correctly.

It is the goal of the OPF to simulate the state of the real power system which satisfies all of the above constraints and at the same time minimizes a given objective, e.g. network losses or generation cost.

1.3 Formulation of OPF constraints

1.3.1 Variable classification

The process of solving the (optimal) power flow problem is easier to understand if the variables are classified in several categories. They are shown in the following.

- Demand variables: They include the variables representing constant values. Demand variables are represented by the vector $\mathcal{P}$. The final simulation result must leave these variables unmodified. Typical demand variables:
  - Active power at load nodes
  - Reactive power at load nodes
  - In general all those variables which could be control variables (see below) but are not allowed to move (for operational or other reasons). Example: Voltage magnitude of a PV node where the voltage is not allowed to move
- Control variables: All real world quantities which can be modified to satisfy the load - generation balance under consideration of the operational system limits (see previous subsection). Since, especially when using the rectangular coordinate system, not all these quantities can be modelled directly, they have to be transformed into variables with purely mathematical meaning. After the computation these variables can, however, be transformed back into the real world quantities. Control variables are represented by the vector $\mathcal{U}$.

  A typical set of control variables of an OPF problem can include:
  - Rectangular Coordinates:
    * Active power of a PV node
    * Reactive power generation at a PV node (sometimes used)
    * Tap position of a transformer
    * Shunt capacitance or reactance
    * Real part of complex tap position (only if the transformer has both taps and phase shift, otherwise the tap is a real number and thus usually a control variable)
    * Imaginary part of complex tap position (see remark above) (This and the previous item are transformed back to the real

world quantities tap and phase shift of the transformer after the OPF computation)

  - Polar Coordinates
    * Active power of a PV node
    * Voltage magnitude of a PV node
    * Tap position of a transformer
    * Phase shift angle of a phase shift transformer
    * Shunt capacitance or reactance

- State variables: This set includes all the variables which can describe any unique state of the power system. State variables are represented by the vector $\mathcal{X}$.

  Examples for state variables:

  - Rectangular Coordinates:
    * Real part of complex voltage at all nodes
    * Imaginary part of complex voltage at all nodes (This and the previous item are transformed back into the real world quantities voltage magnitude and angle after the OPF computation)
  - Polar Coordinates:
    * Voltage magnitude at all nodes
    * Voltage angle at all node

- Output variables: All other variables; they must be expressed as (nonlinear) functions of the control and state variables.

  Examples:

  - Rectangular Coordinates:
    * Voltage magnitude at PQ and PV node
    * Voltage angle at PQ and PV node
    * Tap magnitude of phase shift transformer
    * Tap angle of phase shift transformer
    * Power flow (MVA, MW, MVAr, A) in the line from i to j
    * Reactive generation at PV node
  - Polar Coordinates:
    * Power flow (MVA, MW, MVAr, A) in the line from i to j

- $*$ Reactive generation at PV node

Most variables are continuous, however some, like the transformer tap or the status of shunts are discrete. In this paper all variables are assumed to be continuous. The discrete variables are assumed to be set to their nearest discrete value after the optimization has been done. This does not guarantee optimality, however, results have shown that this approach leads to practically acceptable results.

1.3.2 Equality constraints - power flow equations

As discussed in the subsection above the power flow equations have to be satisfied to achieve a valid power system simulation result. Thus, in summary, the following sets have to be satisfied unconditionally:

SET A: Nodal currents not eliminated, rectangular coordinates

- (7), (8), (13), (14), (15), (16) and (4) (i.e. $4N + 1$ equations)
- This set A includes
  - $2N$ current related variables (I_{e_i}, I_{f_i}, i = 1...N)
  - $2N$ voltage related variables (e_i, f_i, i=1...N)
  - $2l$ PQ-node power related variables (P_i, Q_i, i=1...l)
  - m PV-node active power related variables (P_i, i=$l + 1...N$)
  - m PV-node voltage magnitude related variables ($|V|_i^2$, i = $l + 1...N$)
- **For these 6N variables, 4N+1 equality constraints are given**.

SET B: Nodal currents eliminated, rectangular coordinates

- (17), (18), (19), (20) and (4) (i.e. $2N + 1$ equations)
- This set B includes
  - $2N$ voltage related variables (e_i, f_i, i = 1...N)
  - $2l$ PQ-node power related variables (P_i, Q_i, i=1...l)
  - m PV-node power related variables (P_i, i = $l + 1...N$)
  - m voltage magnitude related variables ($|V|_i^2$, i=$l + 1...N$).

- **For these 4N variables, 2N+1 equality constraints are given**.

SET C: Polar coordinates

- (21), (22), (23) and (10) (i.e. $2N-m+1$ equations). This set C includes
  - $2N$ voltage related variables (V_i, Θ_i, i = 1...N)
  - $2l$ PQ-node power related variables (P_i, Q_i, i = 1...l)
  - m PV-node power related variables (P_i, i = $l+1...N$).
- **For these** $2N-m$ **variables,** $2N-m+1$ **equality constraints are given**.

Note, that in the actual implementation, only one of these sets A, B or C will actually be chosen. If one is satisfied, the other two are also satisfied. Also note that set C has fewer variables and equations than sets A and B. However, this does not mean that set C and as a consequence the polar coordinate system should always be preferred for power system modelling.

The complex tap of a transformer is also a variable which should be included in the above sets A, B or C. However, since they do not change the principles of the following derivations and also for space reasons, they are omitted in the subsequent sections.

1.3.3 Equality constraints - demand variables

For every demand variable an additional equality constraint has to be formulated. The loads in a power system are usually assumed to have a constant active part P and a constant reactive part Q. These two values usually cannot be changed by the operator (not taking into consideration load management) and must not be modified by the normal OPF computation. Thus for every load node where the load cannot be controlled, the two following equality constraints must be valid:

$$P_{scheduled_{PQ_i}} - P_i = 0 \tag{37}$$

$$Q_{scheduled_{PQ_i}} - Q_i = 0 \tag{38}$$

An additional demand variable is the voltage magnitude of a generator PV node where the voltage is not allowed to move. This is represented in the following simple equation with **polar** coordinates:

$$V_{scheduled_{PV_i}} - V_i = 0 \tag{39}$$

In **rectangular** coordinates this is:

$$V^2_{scheduled_{PV_i}} - e_i^2 - f_i^2 = 0 \tag{40}$$

For other demand variables (and fixed control variables) similar equality constraints can be formulated.

1.3.4 Summary - equality constraints

The equations for those equality constraints which have to be satisfied unconditionally can be summarized in general form as follows:

$$\mathbf{g}(\mathcal{X},\mathcal{U},\mathcal{P}) = \mathbf{0} \tag{41}$$

In (41), $\mathbf{g}(\mathcal{X},\mathcal{U},\mathcal{P})$ represents either the equality constraints of sets A, B or C and also those for all demand variables. The variables of the vectors $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{P}$ are either all rectangular coordinates or all polar coordinates.

1.3.5 Inequality constraints

As shown in a previous subsection, many operational values must be limited in the real power system. These limits must be modelled correctly in the OPF simulation in order to have valid simulation results. Mathematically they are formulated as inequality constraints.

The inequality constraints (25) ... (36) can be used in the OPF formulation directly only if they represent bounds on OPF control or state variables or functions of OPF control or state variables. E.g. (31) where the active flow between nodes i and j is limited, cannot be taken directly in the OPF formulation since the variable P_{ij} is an output variable and must be expressed as a function of the control and state variables.

The active and reactive flows P_{ij} and Q_{ij} are computed with the state and control variables in rectangular coordinates as follows:

$$P_{ij} = (e_i f_j - e_j f_i)B_{ij} + (e_i^2 + f_i^2 - e_i e_j - f_i f_j)G_{ij} \tag{42}$$

$$Q_{ij} = (-e_i^2 - f_i^2 + e_i e_j + f_i f_j)B_{ij} + (e_i f_j - e_j f_i)G_{ij} - (e_i^2 + f_i^2)B_{i_o} \tag{43}$$

In polar coordinates this is:

$$P_{ij} = V_i^2 y_{ij} cos\theta_{ij} - V_i V_j y_{ij} cos(\Theta_i - \Theta_j - \theta_{ij}) \tag{44}$$

$$Q_{ij} = V_i^2(-B_{i_o} - y_{ij} sin\theta_{ij}) - V_i V_j y_{ij} sin(\Theta_i - \Theta_j - \theta_{ij}) \tag{45}$$

(42) and (44) will result in the OPF inequality constraints for pure active (MW) -flow limits:

$$P_{ij} \leq P_{high_{ij}} \tag{46}$$

For MVA-flow limits the following inequality constraints are valid:

$$P_{ij}^2 + Q_{ij}^2 \leq S_{high_{ij}}^2 \tag{47}$$

Depending on the choice of the coordinate system either (42) and (43) or (44) and (45) have to be substituted into (47).

The rule that all inequality constraints are either written in polar or all in rectangular coordinates is also valid here.

All inequality constraints must be expressed as functions of the vectors $\mathcal{U}$ and $\mathcal{X}$ which contain all the control and state variables. The general formulation for all these inequality constraints is as follows:

$$\mathbf{h}(\mathcal{X},\mathcal{U}) \leq \mathbf{0} \tag{48}$$

In (48) every function $h_i(\mathcal{X},\mathcal{U})$ represents one of the above inequality constraints. The actual limit values are put to the left hand side of the equation in order to have a vector $\mathbf{0}$ at the right hand side of (48).

1.3.6 Summary - OPF constraints

The constraints of the OPF problem can be split into two parts: The equality constraints, representing the power flow equations and the demand variables and the inequality constraint set, representing all the operational constraints. The following is the general mathematical expression for these two sets:

$$\begin{aligned} &\mathbf{g}(\mathcal{X},\mathcal{U},\mathcal{P}) = \mathbf{0} \\ &\mathbf{h}(\mathcal{X},\mathcal{U}) \leq \mathbf{0} \end{aligned} \tag{49}$$

Every OPF algorithm must try to satisfy (49). Only then will the result simulate the real power system correctly and show a practically useful result.

In the subsequent mathematical treatment of the OPF, it is usually not important to make a distinction between the various types of variables. Thus (49) can be formulated with general OPF variables $\mathbf{x}$:

$$
\boxed{\begin{aligned} \mathbf{g}(\mathbf{x}) &= \mathbf{0} \\ \mathbf{h}(\mathbf{x}) &\leq \mathbf{0} \end{aligned} \qquad (50)}
$$

1.4 Objective functions

1.4.1 Introduction

The formulation of equality and inequality constraints to model the power system and its operational constraints correctly has been discussed in the preceding subsections. These mathematical constraints, however, do not specify one unique network state. An enormous number of power system states can be computed when taking these constraints into account only. Thus the choice of an objective to simulate special, maybe extreme or optimal power system states follows naturally.

There are mainly two objectives which present-day electric utilities try to achieve beside the consideration of the operational constraints:

- Reduction of the total cost of the generated power: Although the switching in and out of generating units (with consideration of operational constraints like minimum down time, etc.) should also be considered this is usually not part of the OPF computation and handled outside by special unit commitment algorithms. Unit commitment algorithms consider the network only as a set of point sources and loads with predicted changes over time and do not take into account constraints like maximum branch flows and voltage limits. Thus today the scope of the OPF is limited to short term (i.e. approx. 15min. - 1h) network optimization with a given and fixed set of on-line generating units. This is also assumed in this paper.

- Reduction of active transmission losses in the whole or parts of the network: This is a common goal of utilities since the reduction of active power losses saves both generating cost (economic reasons) and creates at the same time higher generating reserves (security reasons).

The operator at a utility has to decide which goals are most important. Often the type of utility and its network, generation and load characteristics (e.g. predominant hydro power against predominant thermal power, a network with many long lines with few meshes against a highly meshed network, etc.) determines the main goals of a utility.

1.4.2 Objective function A: minimization of total generating cost

Usually generator cost curves, i.e. the relationship between generated power and the cost for this generated power is given in piecewise linear **incremental** cost. This has an origin in the simplification of piecewise concave cost curves with the valve-points as cost curve breakpoints. Since concave objective functions are very hard to optimize they were made piecewise quadratic which again corresponds to piecewise linear incremental cost curves. This type of objective function could be used in the simple so-called Lambda-Dispatch (Economic Dispatch, ED) where the set of optimal unit base can be determined easily by graphic methods with the consideration of generating unit upper and lower active power limits only.

Piecewise linear incremental cost curves (incremental cost usually monotonically increasing with increasing power) correspond to piecewise quadratic cost curves by doing an integration of the incremental cost curves. This type of cost curve with smooth transition in the cost curve breakpoints (i.e. same first derivative of cost curve segment at left and right hand side of the cost curve break points) can be approximated with very high accuracy by one convex non-linear function.

Although specialized algorithms can use the fact that the cost curves are piecewise quadratic it is assumed in this paper that the cost curves are of general nature with the only condition of being convex and monotonically increasing.

Generation cost curve objective functions are usually functions of their own generated power and not the power of another generating unit j.

Thus for the following derivations the total cost C of all generated powers to be optimized can be written as follows in function of the generated powers:

$$\text{Minimize } \mathcal{F}_{cost} = \sum_{i=l+1}^{N} \mathcal{F}_{cost_i}(P_i) \tag{51}$$

$m = N - l =$ number of generating units to be optimized

$l =$ number of fixed load PQ-nodes

Note that the power generated at the slack node N has also a cost function. This must be considered in the cost objective function of (51).

Also note that in many algorithms the cost curves $\mathcal{F}_{cost_i}$ are assumed to be quadratic or piecewise quadratic.

1.4.3 Objective function B: minimization of active transmission losses

The active transmission losses can be expressed in different ways: a) By a summation of the branch losses of all branches to be considered or b) by a summation of the active nodal powers over all nodes of the network.

a) Losses: computed over branches The total losses are the sum of the losses of all branches and transformers in the area of the network (or the whole network) where the losses are to be minimized:

$$\mathcal{F}_{Loss} = \sum_{i=1}^{NB} \mathcal{F}_{Loss_i} \tag{52}$$

$NB =$ Number of branches of optimized area

where

$$\mathcal{F}_{Loss_i} = P_{km} + P_{mk} \;\; ; \text{branch } i \text{ lies between nodes } k \text{ and } m \tag{53}$$

In (53) the flows between nodes k and m can be replaced by the equations (42) and (43) for rectangular coordinates respectively (44) and (45) for polar coordinates:

In rectangular coordinates the following results:

$$\mathcal{F}_{Loss_i} = G_{mk}((e_m - e_k)^2 + (f_m - f_k)^2) \tag{54}$$

In polar coordinates the following results:

$$\begin{aligned} \mathcal{F}_{Loss_i} &= (V_k^2 - V_m^2) y_{mk} cos\theta_{mk} \\ &+ V_m V_k y_{km} \left(cos(\Theta_m - \Theta_k - \theta_{km}) - cos(\Theta_k - \Theta_m - \theta_{km})\right) \end{aligned} \tag{55}$$

b) Losses: computed over nodes In this case only the losses of the whole network can be computed and not those of a subnetwork. The computation of the total losses is very similar to the computation of the total cost: The total network losses are given when all active nodal powers are added.

The total active losses are computed as follows:

$$\mathcal{F}_{Loss} = \sum_{i=1}^{N} P_i \;\; ; N = \text{Number of network nodes} \tag{56}$$

The slack node is always included in the total loss objective function.

1.4.4 Discussion

As has been shown in the preceding two subsections the losses can be formulated in two different ways, one going over branches the other over the nodes. Method a (branches) is more flexible since it allows to formulate the losses for only parts of a network. This corresponds often to a practical case where each utility models its own network and also those of neighbouring utilities (for reasons of the accuracy of the result) but it can optimize and control its own area only.

Method b on the other side has certain advantages since it allows a rather simple formulation for the total network losses which again allows the use of specialized algorithms for their solution as will be shown in the next section.

For the following derivations both objective functions are assumed to be of general nature and can be formulated as follows.

$$\text{Minimize } \mathcal{F}(\mathcal{X},\mathcal{U}) = \sum_{i\in EL} \mathcal{F}_i(\mathcal{X}_i,\mathcal{U}_i,\mathcal{X}_j,\mathcal{U}_j ...) = \sum_{i\in EL} \mathcal{F}_i(\mathcal{X},\mathcal{U}) \tag{57}$$

where EL = set containing either

a) m generator nodes (cost optimization) or

b) N network nodes (total network loss minimization) or

c) NB area branches (partial network area loss minimization).

Since the OPF does not need a distinction between control ($\mathcal{X}$) and state variables ($\mathcal{U}$) the general objective function formulation in OPF variables is as follows.

$$\text{Minimize } \mathcal{F}(\mathbf{x}) = \sum_{i\in EL} \mathcal{F}_i(\mathbf{x}) \tag{58}$$

This general formulation covers both the losses and also the cost objective functions.

1.5 Optimality conditions

In this subsection the conditions which have to be satisfied in the optimal solution are discussed. The way how to reach the solution where these optimality conditions must be satisfied is not discussed here. The subsequent sections discuss how to reach the optimum.

The general OPF problem formulation is summarized as follows:

$$
\boxed{\begin{array}{lll} \text{Minimize} & \mathcal{F}(\mathbf{x}) & \\ \text{subject to} & \mathbf{g}(\mathbf{x}) & = \mathbf{0} \\ \text{and} & \mathbf{h}(\mathbf{x}) & \leq \mathbf{0} \end{array} \qquad (59)}
$$

The optimality conditions for (59) can be derived by formulating the Lagrange function $\mathcal{L}$:

$$
\mathcal{L} = \mathcal{F}(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{h}(\mathbf{x}) \tag{60}
$$

The Kuhn-Tucker theorem [1] says that if $\widehat{\mathbf{x}}$ is the relative extremum of $\mathcal{F}(\mathbf{x})$ which satisfies at the same time all constraints of (59), vectors $\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\mu}}$ must exist which satisfy the following equation system:

$$
\begin{array}{rcrcl}
\dfrac{\partial \mathcal{L}}{\partial \mathbf{x}} & = & \dfrac{\partial}{\partial \mathbf{x}} \Big(\mathcal{F}(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{h}(\mathbf{x}) \Big) \,|_{\widehat{\mathbf{x}}, \widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\mu}}} & = & \mathbf{0} \\
\dfrac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} & = & \mathbf{g}(\mathbf{x}) \,|_{\widehat{\mathbf{x}}} & = & \mathbf{0} \\
\mathbf{diag}\{\boldsymbol{\mu}\} \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} & = & \mathbf{diag}\{\boldsymbol{\mu}\} \; \mathbf{h}(\mathbf{x}) \,|_{\widehat{\mathbf{x}}, \widehat{\boldsymbol{\mu}}} & = & \mathbf{0} \\
 & & \widehat{\boldsymbol{\mu}} & \geq & \mathbf{0}
\end{array} \tag{61}
$$

The third constraint set together with the last set means that an inequality constraint is only active when $\mu_i > 0$.

It is the goal of the OPF algorithms to find a solution point $\widehat{\mathbf{x}}$ and corresponding vector $\widehat{\boldsymbol{\lambda}}, \widehat{\boldsymbol{\mu}}$ which satisfy the above conditions.

If this solution is found there is no guarantee that the global optimum is found. The Kuhn-Tucker conditions guarantee a local or relative optimum only. However, although no formal proof is possible, usually only one optimum (i.e. the global optimum) exists for practical OPF problem formulations. æ

2 HISTORICAL REVIEW OF OPF DEVELOPMENT

2.1 The early period up to 1979

The development of an optimal solution to network problems was initiated by the desire to find the minimum of the operating cost for the supply of electric power to a given load [2], [3]. The problem evolved as the socalled dispatch problem. The principle of equal incremental cost to be achieved for each of the control variables or controllers has already been realized in the pre-computer era when slide rules and the like were applied.

A major step in encompassing not only the cost characteristics but also the influence of the network, in particular the losses was the formation of an approximate quadratic function of the network losses expressed by the active injections [2]. Its core was the B-matrix which was derived from a load flow and was easily combined with the principle of equal incremental cost thus modifying the dispatched powers by loss factors. The method has lent itself to analog computer solutions in the online operation of systems. At this point, however, no constraints could be considered.

In the following period the development has mainly emphasized the formulation of a more complete optimal power flow towards the inclusion of the entire AC network [4], [5], [7], [9], [10]. The necessity to consider independent and dependent variables has led to a considerable increase of the system of equations which where nonlinear and thus difficult to handle. The formulation of the problem must be considered as a remarkable improvement as shown by Squires, Carpentier, however, still there was no effective algorithm available. At that time the ordinary load flow made considerable progress [6], [12] and the capabilities of computers showed promising aspects. Hence, the analysts were intrigued by the possibilities in the area of the load flow and tried to incorporate this success in the area of the optimal power flow.

A remarkable conceptual progress was made by Dommel, Tinney [7] when they formulated the exact optimality conditions for an AC based OPF which allowed the use of the solution of an ordinary load flow. By eliminating the dependent variable with the help of a solved load flow iteration a gradient method was designed which led to a true optimal solution of a dispatch problem including the detailed effects of the AC network. This step marks an important step in the development of the OPF since there was an algorithm which had several ramifications (reduced gradients, etc.) and it considered already constraints of variables. The technique employed was based on penalty functions which could easily be attached to the Lagrangian function of the basic method. The gradient or reduced gradient included derivatives of the quadratic penalty functions also which by their character had quite different magnitudes as compared to the gradients of the objective functions. As a consequence the parameter which determined the step length in the direction of the gradient was not able to confine the solution sufficiently close. The result was that the convergence of the whole approach was quite poor. In particular, maintaining constraints by taking in and releasing constrained variables was not satisfactory. Programming packages were developed but required detailed tuning and turned out not to be applicable to general problems. A quite complete overview of these developments is given in [17].

2.2 Recent developments since 1979

Since the gradient concept did not turn out to be successful, also from the point of view of treating constraints several other concepts were pursued. One line was the application of linear programming which offered a clear approach to handling constraints [15], [16]. Another direction was the use of quadratic programming whereby standard quadratic routines were used [14], [20]. A different approach led to exploiting the optimality conditions in the form of Newton's method.

The first two methods are characterized by the use of a solved load flow which yields a feasible starting point. Newton's method led to iterative solution steps which approach the optimal result in a global way [19].

Each of these approaches showed considerable progress over gradient methods both as far as convergence is concerned and with regard to treating constraints.

Linear programming methods showed a first success in the area of dispatching generator outputs whereby cost curves have been represented by linear segments and the load flow was incorporated in a linearized fashion (Stott, [15]). This line has been further refined recently such that an AC model of the network could be treated as well and a reactive dispatch for the purposes of minimizing losses was made possible.

Quadratic programming followed [18] more closely the facts of the system model which shows piecewise quadratic cost curves, a quadratic behavior of losses and of powers in general, e.g. the slack power. Since the quadratic behavior is sufficiently accurate for small deviations only the quadratic approach is also iterative whereby standard quadratic programming routines were applied. The general observation was that convergence of these methods was extremely good, however, the formation of quadratic forms, of loss formulae and other conversions require a considerable effort which turned out to be a drawback as far as the overall performance was concerned.

For both linear and quadratic methods the load flow solution has to be converted to a compact form or the socalled incremental power flow which can be extended to a quadratic form. It was instrumental for the application of these methods and still is for the most recent forms of the OPF.

The development of the Newton approach for the purposes of the OPF is a consequence of the success of the techniques derived for the ordinary power flow [19]. Sparsity techniques, ordering, decoupling methods, etc. have suggested to maintain and keep the original optimality conditions derived from the Lagrangian and to treat the large system of equations as if it would be a power flow problem which nowadays can be solved for thousands of nodes. The formulation and the solution of the problem is easy for the unconstrained case. Constraints had to be treated by penalty functions, however, no straightforward routine could be devised which leads to active constraints. The method remains with heuristic steps which take in and release constraints which requires updating steps of the factorized system matrix. Although Newton's method was considered as the only approach to treat the loss minimization problem effectively some time ago this image is fading somewhat and is giving way to methods which incorporate linear programming routines for reasons of performance, uniqueness of approach and use of proven routines.

In a broader perspective the optimal power flow is becoming the main tool for the assessment and enhancement of the security of the system [22], [23]. The objective function may have a direct relation to security, e.g. in the case where the deviation from a desirable voltage profile is to be minimized. Otherwise it is the tool to achieve a well defined solution, with an economic benefit, as given by minimum losses.

Security, however, is a problem where constraints are to be maintained or where excess variables are corrected. A modern OPF lends itself to the treatment of these requirements and the recent efforts in improving the methods, in particular, as far as constraints are concerned, prove the great interest in this aspect of the OPF.

3 CLASSIFICATION OF ALGORITHMS TO ACHIEVE OPF OPTIMALITY CONDITIONS

3.1 Practical constraints and desirable features of the algorithms

It has been shown in the preceding sections that the OPF problems can be defined in different ways. The determination of an optimal, steady state network operation is the general goal. Utilities are interested in achieving this goal for both network planning studies and also in real-time operation.

In **planning studies** the utility wants to know how to expand or change its network in order to achieve e.g. minimum losses under a variety of load scenarios. Another problem is the minimization of cost of future planned generation. The OPF is used to propose to the utility where to put what generator capacity in the present or future network to achieve minimum cost operation. It is obvious that statistical values for load changes or approximations for the expected cost of new generators will have to be considered and thus make the result of the OPF subject to many assumptions, predictions and uncertainties. The OPF algorithm used for planning studies should be able to handle this data which is usually based on statistics.

Another important area where the OPF is and will be applied, is the **real-time OPF**, i.e. the use of the OPF result for the actual network operation. The goal is here to take the OPF result and try to realize the computed values in the actual, real-time network. This real-time network optimization is usually done under operator control, i.e. the computed optimal values are read by the operator who changes the actual controls to achieve the same network state as obtained in the OPF simulation. A closed loop OPF, i.e. the automatic realization of the optimal computed solution in the real network, is - at least within the near future - not realistic, but may be approached by a close interaction between the operator and the simulated OPF result, maybe with expert system guidance. The practical aspects of the OPF implementation are key to the real-time use of the OPF. In this application of the OPF the algorithms are useless if their output does not conform to practical aspects. Under the assumption that the operator tries to achieve the optimal solution some practical constraint considerations are critical to the application of the OPF:

- **Computational speed**: The OPF result must be obtained within a reasonable timeframe, starting at the time when the real-time data is obtained from the network. Since state estimation algorithms usually take the raw data before being used by the OPF another time delay exists. Both state estimation and succeeding OPF computations must be fast enough to be practically applicable. The realization of speed is a combination of fast hardware and fast algorithms.

  - The **hardware** must be fast, but must be in the right price range and computer class used in the energy management systems at utilities. A practical solution to this constraint is today, with the systems offered by the energy management system vendors, often quite difficult to achieve. New technologies, fully applied to the energy management systems, should help to solve this problem in the near future.
  - The **software** must be such that it can compute OPF problems with network sizes of thousands of electrical nodes within a reasonable (wall-clock) time. Speed can mainly be achieved by translating the physically given special characteristics of the electrical network in special OPF algorithms. An example is given by the loosely connected network topology which is translated into a special sparsity storage scheme in the computer which again makes fast iterations possible (only non-zero value arithmetic operations). Another typical electrical behavior is the locality of network state changes, e.g. the effect of changing the voltage at a generator node remains in the local vicinity of the changed generator and does not spread over the whole network. This is translated into algorithms which use the localized behavior of the network and speed up computation by not having to compute all network variables but only the local ones. Also the fact that not many branch limits will be active at the optimal solution can be used by the OPF algorithm and computational speed will be improved by doing so. Consideration of data uncertainties can be used to speed up the algorithmic solution: E.g. if the accuracy of a large generator output power measurement is about five MW, making a computation with an accuracy of one MW is useless and consumes unnecessary computing time.

- **Robustness**: The OPF may not, under any circumstances, diverge or even crash. Fast and straight-forward convergence is important to acceptance and real-time application of the OPF result. Even in cases where there is no optimal solution with consideration of all constraints the OPF must tell the user that there is no solution and output a near-optimal solution which satisfies most of the constraints. Operator or expert system involvement in these difficult to solve cases is desirable to achieve a practically useful OPF solution.

- **Controller movements**: The OPF assigns an optimal value to each possible control variable. Assuming that there is a large number of possible control variables the OPF algorithm would move most of them from the actual state to the optimal state. However, a practical real-time realization of this optimal state is not possible since the operator cannot have e.g. hundred generator voltages be moved to different settings within a reasonable time. Only the most effective subset should be moved, which means that within the OPF the algorithmic problem of moving the minimum number of controllers with maximum effect has to be solved. Another problem with the movement of controllers is the distance it has to move from the actual to the desired, OPF computed optimal value. Time constraints like maximum controller movement per minute must be considered to achieve practical OPF use. This again leads to another critical OPF point: When talking about time aspects of movement the load changes within pre-determined time frames should also be considered. As an example, when the load changes very rapidly within the next fifteen minutes the generation should be optimized with consideration of the actual and the expected load in fifteen minutes. The OPF can result in different optimum solution points depending on the constraints considered in the optimum.

- **Local controls**: Tap changers are usually used to regulate voltages locally to scheduled values. These scheduled voltage values can be more desirable than any optimal voltages computed by the OPF. A localized, not optimal control might practically be preferred to the solution for this control obtained by the OPF. If this is the case the OPF algorithm has to handle this situation.

Some of these practical constraints can be incorporated into the classical OPF formulation shown in preceding sections. Where possible this is done in the inequality constraint set. However, some practical constraints like the local control discussed above is usually taken out of the optimization algorithm. These constraints are taken into account separately as part of an overall OPF solution, where one part is the optimization algorithm and the other is an algorithm based usually on heuristics and algorithmic application of special characteristics of the electrical network. This separation will be discussed in the next sections.

The solution of the classical OPF problem formulation (see section 1), the practical aspects discussed above and the mathematically known algorithms lead to OPF classifications which are discussed in the next subsection.

3.2 Classification of OPF algorithms

3.2.1 Distinction of two classes

The separation of OPF algorithms into classes is mainly governed by the fact that very powerful methods exist for the ordinary load flow which provide an easy access to intermediate solutions in the course of an iterative process. Further, it can be observed that the optimum solution is usually near an existing load flow solution and hence sensitivity relations lead the way to the optimum. Hence, one class exists which relies on a solved load flow and on tools provided by the load flow.

The second class originates from a rigorous formulation of the OPF problem, employs the exact optimality conditions and uses techniques to fulfill the latter. In this case a solved load flow is not a prerequiste. The preferred method for reaching the optimality conditions is Newton's method.

There are advantages and disadvantages in both methods which have a certain bearing depending on the objective, the size of the problem and the envisaged application.

Hence, optimal power flow algorithms will be discussed in two classes:

- Class A: Methods whereby the optimization starts from a solved load flow. The Jacobian and other sensitivity relations are used in the optimizing process. The process as a whole is iterative. After each iteration the load flow is solved anew.
- Class B: Methods relying on the exact optimality conditions whereby the load flow relations are attached as equality constraints. There is no prior knowledge of a load flow solution. The process is iterative and each intermediate solution approaches the load flow solution.

3.2.2 Discussion of class A algorithms

When the load flow is solved in the known way the following information is available or can be extracted.

- the set of nodal voltages (complex or amplitude/angle)
- the Jacobian matrix either original or in factorized form

- the incremental power flow either in linearized form or with a quadratic extension

The dependent and independent variables fulfill the load flow equations and are consistent. The variables are within limits or not too far off. Hence the Jacobian and any derived functions may be used as sensitivity relations.

The actual optimizing process is separate whereby sensitivity relations of the load flow are incorporated. Constraints are introduced at this stage. In some cases dependent variables are eliminated before the actual solution process in order to arrive at smaller size matrices, tableaus, etc.

An examples for class A methods is given by Dommel [7].

The choice of class A methods can be appreciated when performance aspects and certain limitations are considered.

One outstanding advantage is the clear and systematic treatment of constraints when linear and quadratic programming methods are employed in the optimization part. The load flow supplies sensitivity relations which are quite often extractable in a reduced form, e.g. linear incremental power flow which is a scalar relation. Constraints are formulated in terms of the set of remaining variables (when a subset of variables has been eliminated). The active power dispatch is an excellent example of a class A method. The Hessian matrix derived from the quadratic cost functions is diagonal and the incremental power flow is a scalar.

In Stott [15] cost curves are approximated by straight lines. Hence the optimization is done on the basis of linear programming.

Class A methods have been applied to loss minimization but in this case the quadratic form has to be derived from the load flow (extended incremental power flow). The computational effort in forming the quadratic form and its treatment within the quadratic programming routine limits the application of class A methods for loss minimization. The observation is that systems above 300 nodes require comparatively large computing times.

There is however one aspect of class A methods, namely the use of approximations in the formation of the Hessian or the use of linear approximations. It turns out that it is the linear relations of the load flow (incremental power flow) which determine the exact optimum. Quadratic relations and their approximations determine the speed of convergence, they limit step length etc. If suitable approximations to the Hessian can be found, quadratic and linear methods within class A can be quite powerful.

3.2.3 Discussion of class B algorithms

Class B algorithms start from the optimality conditions evolving from a Lagrangian function. The optimality conditions comprise derivatives of the objective functions and equality constraints. It is to be remembered that they are conditions and give little indications as to their fulfillment. Class B methods aim at the satisfaction of the optimality conditions in a direct way whereby inequality constraints usually are treated in a special form.

There are two approaches which fall into this category. It is Newton's method which allows to meet the optimality conditions as long as they are differentiable. A second method is available if the Lagrangian is quadratic which results in linear optimality conditions. Constraints can be treated by linear programming as will be shown later. As a matter of fact Newton's method and this quadratic approach merge into one single method when the Lagrangian is quadratic or when the first derivatives of the Lagrangian are kept constant (quasi- Newton).

The advantages of class B methods lie in the fact that the Hessian is very sparse or remains constant or can be inserted in approximate terms. It is a non-compact method which does not result in a progressive increase in computation time for the formation of the Hessian or for the solution of the optimization part. The overall system of equations can be very large in dimension but it is very sparse. Large numbers of nodes can be handled. In case of Newton's methods the coefficients of the matrices need not be precise since the accuracy of the solution is guaranteed by the mismatches (right hand sides), e.g. decoupled loadflow methods can be employed.

As it stands now class B methods have difficulties in handling constraints. The standard approach at the moment seems to be to treat constraints by penalty terms whereby active constraints are determined by heuristic methods. The consequence is that the system of equations needs updating and refactorization which in the end deteriorates the performance.

The quadratic method mentioned above avoids this problem and is able to treat constraints in a systematic fashion.

The recent development has favored class B methods for large systems, in particular when losses are to be minimized.

4 OPF CLASS A: POWER FLOW SOLVED SEPARATELY FROM OPTIMIZATION ALGORITHM

4.1 Introduction

In this section the OPF formulation is solved by a class of algorithms where the power flow is used in the conventional way to solve the power flow problem for a given set of control and demand variables with fixed values. This solution is then taken to be the starting point for an optimization. The optimization is thus separated from the conventional power flow solution algorithm. Since as will be shown in the next subsection the optimization represents only an approximation to the original OPF problem, its solution may not be the final one and so the optimized OPF variables are transferred back to the power flow problem which is solved again. The result of the optimization is thus taken as the input for the power flow which is solved, this result is again taken as input for the optimization problem, etc. All OPF Class A algorithms have this procedure in common.

The power flow is not discussed in this paper and is assumed to be known. Extensive literature can be found in papers and student text books. However, the optimization part where several algorithms can be used is discussed in the following subsections.

Thus the various OPF class A algorithms show differences mainly in the optimization part. One of two algorithms is usually used for the optimization part: Either a linear programming (LP) or a quadratic programming (QP) based algorithm. Both algorithms can solve their respective optimization problem with straightforward procedures and no heuristics are needed. The main difference between both optimization problem definitions can be found in the objective function formulation: The LP can handle only linear objective functions,

$$\text{LP: Minimize } \mathcal{F}(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \tag{62}$$

and the QP handles quadratic objective functions:

$$\text{QP: Minimize } \mathcal{F}(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} \tag{63}$$

Both optimizations are restricted to consider linear equality and inequality constraints:

$$\mathbf{Jx} = \mathbf{b_1} \tag{64}$$

and

$$\mathbf{Ax} \leq \mathbf{b_2} \tag{65}$$

The LP objective function can be seen as a simplification of the QP objective function by neglecting the quadratic objective function terms as represented in the matrix $\mathbf{Q}$. From this point of view any QP formulation can easily be transformed into an LP formulation.

Note, however, that the actual solution processes for both LP and QP are distinctly different.

Both LP and QP solution algorithms are described in textbooks and mathematical details of how to get the iterative optimal LP or QP solution are briefly discussed in the appendix section A.1 (LP) and A.2 (QP) of this paper. However, in section 4.5 of this paper, an engineered LP version is mentioned which goes beyond the conventional LP linear objective: This LP-based algorithm is tuned to the typical OPF problem objective functions and can solve general separable, convex objective functions. In addition, in the appendix A.2 a QP-algorithm is described which works with well known LP tools. It is important to note, that independent of the engineered modifications to the original LP or QP algorithms, the basic principles of the chosen LP or QP optimization remain always valid.

In the OPF class A approaches the general OPF problem formulation is approximated around an operating point vector x^k. The index ^{k} means that this operating point will vary during the OPF class A solution process where k is incremented by 1 from one iteration to the next. The OPF problem is formulated in a quadratic approximation around this operating point x^k for the objective function $\mathcal{F}$, however in linearized form for the equality and inequality constraints. The linearization of the constraints is justified by the fact that both LP and QP algorithms can handle linearized constraints only. Thus the problem formulation is adapted to the mathematical problem formulation, which then leads to a straightforward optimization solution.

Approximations to both the objective functions and to the constraints lead to inaccuracies which must be corrected by some means. In OPF class A algorithms this is done by solving an exact AC power flow once an optimized solution (which is optimal only with respect to the approximated problem formulation) has been obtained. The repetitive execution of power flow and LP, respective QP optimization must lead to better, more accurate approximations, as more power flow-LP or QP optimizations are executed. The solution to the problem of getting this iterative process to converge is critical. Note, since the power flow has no degree of freedom and thus no ability to influence the overall convergence process, the iterative LP or QP optimization steps alone are responsible for obtaining convergence. In order to clarify this point, an example is given: In order to justify the approximations it might be necessary to restrict the movement of certain variables x from the starting point x^k to its optimum x^k_{opt}. No straighforward mathematical algorithm exists which tells, how far the variables are allowed to move within the optimization algorithm. Thus, since approximations are valid only for small deviations from an operating point, the definition of what small means can be critical to the overall convergence.

In the following subsection a derivation is given of how to get an LP or QP problem formulation, starting from the general OPF problem formulation.

4.2 OPF class A optimization problem formulation

The original OPF problem formulation as given in (59) is taken as starting point for an approximated optimization problem. In the following, a special formulation with an approximation of the quadratic objective function with second and first order approximated equality constraints and linearized inequality constraints is derived. This formulation is needed to derive a QP formulation which can be solved by the algorithms given from the mathematicians. The LP formulation can easily be derived from the QP by neglecting the quadratic terms of the objective function. Note that an LP can always be derived from a QP. However, it is not evident that the LP algorithms for the LP problem formulation (even if derived from the original QP problem) converge in a comparable way to QP algorithms for the QP problem formulation.

The following general derivations are made such that in a later subsection the different LP and QP optimization problem formulations for the cost and the loss optimization are easy to understand.

In the following formulas the OPF variable vector $\mathbf{x}$ is split into several subvectors:

$$\mathbf{x}^T = (\mathbf{x_1^T x_2^T x_3^T x_4^T}) \tag{66}$$

where

- $\mathbf{x_1}$: All active power variables P_i at generator PV nodes (dimension: m)
- $\mathbf{x_2}$: All active power variables P_i at load PQ nodes (dimension: l)
- $\mathbf{x_3}$: Vector containing the subvectors $\mathbf{x_{31}}$ and $\mathbf{x_{32}}$:
  - $\mathbf{x_{31}}$: All reactive power variables Q_i of all PQ-load nodes (dimension: l)

– $\mathbf{x_{32}}$: All voltage magnitude variables $|V|_i^2$ of all generator PV nodes (dimension: m) (only when taking rectangular coordinates; when using polar coordinates, the vector $\mathbf{x}_{32}$ does not exist)

- $\mathbf{x_4}$: Either all real and imaginary parts of voltage variables e_i, f_i (dimension: 2N) (when taking rectangular coordinates) or all voltage magnitudes and all voltage angle variables V_i, θ_i (dimension: 2N) (when taking polar coordinates)

The equality constraint set is also split into several subsets. Note that the subset B, as explained in subsection 1.3.2 of this paper, is taken in the following derivations. For the other sets, similar derivations can be made.

$$\mathbf{g}^T = (\mathbf{g_1^T} \mathbf{g_2^T} \mathbf{g_3^T} \mathbf{g_4^T}) \tag{67}$$

where

- $\mathbf{g}_1$: Load flow equations representing the active powers at all PV nodes (number of equality constraints of type $\mathbf{g_1}$: m).

- $\mathbf{g}_2$: Load flow equations representing the active powers at all PQ nodes (number of equality constraints of type $\mathbf{g_2}$: l).

- $\mathbf{g}_3$: Load flow equations representing the other non-active-power variables like voltage magnitude at PV nodes, reactive power at PQ nodes and the equality constraint for the fixed slack-node angle (number of equality constraints of type $\mathbf{g_3}$: $N + 1$).

- $\mathbf{g}_4$: Demand variable related equality constraints: Fixed active and reactive loads at some PQ nodes, fixed voltage at some PV nodes, etc. (number of equality constraints of type $\mathbf{g_4}$: d; note that the number cannot be given in function of network nodes or other typical network parameters; the actual number, assumed to be d, depends on the available choice of demand variables of the network).

The approximated optimization problem is now as follows: Minimize either the total generation cost

$$\mathcal{F}_{\mathbf{cost}}(\mathbf{x_1}) = \mathcal{F}_{\mathbf{cost}}(\mathbf{x_1^k}) + \mathbf{c}^{T^k} \Delta\mathbf{x_1} + \frac{1}{2} \Delta\mathbf{x_1}^T \mathbf{Q}^k \Delta\mathbf{x_1} \tag{68}$$

or minimize the total network losses:

$$\mathcal{F}_{\mathbf{loss}}(\mathbf{x_1}, \mathbf{x_2}) = \mathcal{F}_{\mathbf{loss}}(\mathbf{x_1^k}, \mathbf{x_2^k}) + \mathbf{1}^T \Delta\mathbf{x_1} + \mathbf{1}^T \Delta\mathbf{x_2} \tag{69}$$

(In this paper only the loss objective function of (56) is used for further derivations. Similar derivations are possible for the other loss objective function (52).)

subject to the equality constraints (quadratic approximation for all equality constraints $\mathbf{g_1}$, $\mathbf{g_2}$ and $\mathbf{g_3}$):

$$\mathbf{g_1}(\mathbf{x_1^k}, \mathbf{x_4^k}) + \Delta\mathbf{x_1} + \mathbf{J_{14}}^k \Delta\mathbf{x_4} + \frac{1}{2}\Delta\mathbf{x_4}^T \mathbf{M_{14}}^k \Delta\mathbf{x_4} = \mathbf{0} \tag{70}$$

$$\mathbf{g_2}(\mathbf{x_2^k}, \mathbf{x_4^k}) + \Delta\mathbf{x_2} + \mathbf{J_{24}}^k \Delta\mathbf{x_4} + \frac{1}{2}\Delta\mathbf{x_4}^T \mathbf{M_{24}}^k \Delta\mathbf{x_4} = \mathbf{0} \tag{71}$$

$$\mathbf{g_3}(\mathbf{x_3^k}, \mathbf{x_4^k}) + \Delta\mathbf{x_3} + \mathbf{J_{34}}^k \Delta\mathbf{x_4} + \frac{1}{2}\Delta\mathbf{x_4}^T \mathbf{M_{34}}^k \Delta\mathbf{x_4} = \mathbf{0} \tag{72}$$

For the equality constraint set $\mathbf{g_4}$ only a linearized approximation is used:

$$\mathbf{g_4}(\mathbf{x_1^k}, \mathbf{x_2^k}, \mathbf{x_3^k}, \mathbf{x_4^k}) + \sum_{i=1}^{4} \mathbf{J_{4i}}^k \Delta\mathbf{x_i} = \mathbf{0} \tag{73}$$

The same holds for the inequality constraint set $\mathbf{h}$:

$$\mathbf{h}(\mathbf{x_1^k}, \mathbf{x_2^k}, \mathbf{x_3^k}, \mathbf{x_4^k}) + \sum_{i=1}^{4} \mathbf{A_i}^k \Delta\mathbf{x_i} \leq \mathbf{0} \tag{74}$$

In (68) ... (74) some abbreviations have been used:

$$\mathbf{c}^k = \left.\frac{\partial \mathcal{F}_{\mathbf{cost}}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x^k}} \quad ; \mathbf{Q}^k = \left.\frac{\partial^2 \mathcal{F}_{\mathbf{cost}}}{\partial \mathbf{x^2}}\right|_{\mathbf{x}=\mathbf{x^k}}$$

$$\mathbf{J}_{ij}^k = \left.\frac{\partial \mathbf{g}_i}{\partial \mathbf{x}_j}\right|_{\mathbf{x}=\mathbf{x^k}} \quad ; \mathbf{M}_{ij}^k = \left.\frac{\partial^2 \mathbf{g_i}}{\partial \mathbf{x_j^2}}\right|_{\mathbf{x}=\mathbf{x^k}}$$

$$\mathbf{A}_i^k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_i}\right|_{\mathbf{x}=\mathbf{x^k}}$$

Note that index ^{k} means that these variables, vectors and matrices are state dependent and can vary from one state to the other (or from iteration to iteration).

Assume that a power flow has been solved for this operating point, thus the equality constraints $\mathbf{g}(\mathbf{x^k}) = \mathbf{0}$ are satisfied:

$$\mathbf{g}(\mathbf{x}^k) = \mathbf{0} \tag{75}$$

The optimization problem defined with (68) ... (74) is not a classic QP formulation because quadratic equality constraints exist. Now, different steps can be undertaken for cost and loss optimization in order to derive QP or LP formulations.

Because of their different nature, different assumptions can be made when setting up the above optimization problem for the cost and the loss minimization OPF problem. Both derivations are given in the following two subsections.

4.3 Total generation cost as objective function in OPF class A formulations

4.3.1 Sparse, non-compact QP cost optimization problem

After the general derivation of the previous subsection the total generation cost as OPF objective function is discussed in this subsection.

Since the cost of each generator active power is not dependent on the cost of another generator the second derivatives of the cost function with respect to the active power variables of all generators ($\mathbf{x}_1$) lead to a diagonal matrix:

$$\mathbf{Q}^k = \mathbf{diag}(q_i^k) \tag{76}$$

with

$$q_i^k = \left.\frac{\partial^2 \mathcal{F}_i}{\partial x_{1i}^2}\right|_{x_i = x_i^k} \tag{77}$$

and x_{1i}: active power of the generator i; $\mathcal{F}_i$: cost of generator i in function of its active power.

Note that when assuming quadratic cost curves these factors q_i^k are constant, i.e. not state dependent.

When optimizing cost, all quadratic terms of the optimization problem exclusive the one of the objective function are usually neglected. This is possible because the cost curves are already of a (near) quadratic nature and turn out to be dominant. Thus the cost optimization problem is as follows:

$$\text{Minimize } \mathcal{F}_{cost} = \mathcal{F}_{cost}(\mathbf{x_1}^k) + \mathbf{c}^{Tk}\Delta\mathbf{x_1} + \frac{1}{2}\Delta\mathbf{x_1}^T \mathbf{diag}(q_i^k)\Delta\mathbf{x_1} \tag{78}$$

subject to

$$\mathbf{g}(\mathbf{x^k}) + \mathbf{J}^k \Delta\mathbf{x} = \mathbf{0} \tag{79}$$

and

$$\mathbf{h}(\mathbf{x^k}) + \mathbf{A}^k \Delta\mathbf{x} \leq \mathbf{0} \tag{80}$$

with

$$\mathbf{J}^k = \begin{bmatrix} \mathbf{U} & \mathbf{0} & \mathbf{0} & \mathbf{J_{14}}^k \\ \mathbf{0} & \mathbf{U} & \mathbf{0} & \mathbf{J_{24}}^k \\ \mathbf{0} & \mathbf{0} & \mathbf{U} & \mathbf{J_{34}}^k \\ \mathbf{J_{41}}^k & \mathbf{J_{42}}^k & \mathbf{J_{43}}^k & \mathbf{J_{44}}^k \end{bmatrix} ; \mathbf{g}(\mathbf{x^k}) = \begin{bmatrix} \mathbf{g_1}(\mathbf{x^k}) \\ \mathbf{g_2}(\mathbf{x^k}) \\ \mathbf{g_3}(\mathbf{x^k}) \\ \mathbf{g_4}(\mathbf{x^k}) \end{bmatrix} ; \tag{81}$$

$$\mathbf{A}^k = \begin{bmatrix} \mathbf{A_1^k A_2^k A_3^k A_4^k} \end{bmatrix} ; \mathbf{\Delta x}^T = \begin{bmatrix} \mathbf{\Delta x_1^T \Delta x_2^T \Delta x_3^T \Delta x_4^T} \end{bmatrix} ;$$

The resulting problem is now a classic QP problem. Note that in this formulation the problem is very sparse. This sparsity must be considered when applying the QP algorithms to this problem. In [21] sparsity techniques are discussed in detail.

4.3.2 Compact, non-sparse QP cost optimization problem (Linear incremental power flow)

The cost optimization problem has been formulated as a QP with sparse matrices. However, the number of variables is very high and thus many variable related operations will result. In the following a derivation of the cost optimization problem is given where on one side the number of variables is reduced to a much smaller set, however, on the other side the sparsity of the matrices gets lost.

In order to achieve this compact QP formulation variables have to be eliminated from the equality constraint set $\mathbf{g}(\mathbf{x^k}) + \mathbf{J}^k \Delta \mathbf{x} = \mathbf{0}$ (79).

Note that this equality constraint set contains $2N + 1 + d$ equality constraints. The variable vector $\boldsymbol{\Delta x}$ contains $4N$ variables.

This set can be reduced to one equation with $4N - (2N + 1 + d) + 1 = 2N - d$ variables. This means that from the total of $4N$ variables, $2N + d$ variables must be eliminated. Note that $d \leq 2N$.

In order to achieve a compact formulation, the variables of the vector $\Delta \mathbf{x_4}$ (without the real and imaginary slack node voltage variable) (i.e. $2N - 2$ variables) are eliminated. From the vectors $\Delta \mathbf{x_2}$ and $\Delta \mathbf{x_3}$, $d + 2$ variables have to be eliminated: The rule is to eliminate first the variables of $\Delta \mathbf{x_2}$ for which demand variable constraints exist (formulated in the equality constraint set $\mathbf{g_4}$). Doing this will eliminate all active reactive power variables of non-manageable load PQ nodes. The remaining variables to be eliminated are taken from the vector $\Delta \mathbf{x_3}$.

Eliminating the variables accordingly in the inequality constraint set $\mathbf{h}(\mathbf{x^k})+\mathbf{A}^k\Delta\mathbf{x} \leq \mathbf{0}$ reduces the optimization problem to $2N-d$ non-eliminated variables.

Two voltage variables at the slack node are not eliminated. This comes from the fact that there is a chance of having singularity or linearly dependent equality constraints among the equality constraints of the set $\mathbf{g}$. Linear dependence can lead to zero pivots during factorization. A division by a zero pivot can usually be avoided if the real and imaginary part of the slack node voltage are not eliminated.

Since the variable set $\Delta\mathbf{x_1}$ is not eliminated the objective function is unchanged.

The optimization problem is now as follows:

$$\text{Minimize } \mathcal{F}_{cost} = \mathcal{F}_{cost}(\mathbf{x_1}^k) + \mathbf{c}^{T^k}\Delta\mathbf{x_1} + \frac{1}{2}\Delta\mathbf{x_1}^T\mathbf{diag}(q_i^k)\Delta\mathbf{x_1} \tag{82}$$

subject to

$$\boldsymbol{\alpha_1}^{T^k}\Delta\mathbf{x_1} + \boldsymbol{\alpha_2}^{T^k}\Delta\mathbf{x_2'} = \alpha_0 \tag{83}$$

((83) is called the **linear incremental power flow** equation.)
and

$$\mathbf{h}(\mathbf{x_1^k}, \mathbf{x_2^k}) + \mathbf{A_1'}^k\Delta\mathbf{x_1} + \mathbf{A_2'}^k\Delta\mathbf{x_2'} \leq \mathbf{0} \tag{84}$$

with

- $\Delta\mathbf{x_2'}$ including all non-eliminated variables excluding $\Delta\mathbf{x_1}$ (see paragraph above for what variable types are included).
- $\mathbf{h}(\mathbf{x_1^k}, \mathbf{x_2^k})$ representing the inequality constraint set values at the operating point $\mathbf{x}^k$
- $\mathbf{A_1'}^k$ representing the sensitivities of the inequality constraints with respect to $\Delta\mathbf{x_1}$ at the operating point $\mathbf{x}^k$.
- $\mathbf{A_2'}^k$ representing the sensitivities of the inequality constraints with respect to $\Delta\mathbf{x_2'}$ at the operating point $\mathbf{x}^k$.

Note that all these matrices can be derived by simple variable elimination in all equality (79) and inequality constraints(80).

4.4 Total network losses as objective function in OPF class A formulations

The formulation of the loss QP optimization problem must be derived differently than the cost QP optimization problem. The main reason comes from the fact that the loss objective function as shown in (69) is linear when using the active powers of all nodes as a subset of the OPF variables.

Several QP derivations are possible. Two of them are shown in the following two subsections.

4.4.1 Sparse, non-compact QP loss optimization problem

The basic idea of this optimization problem formulation is the elimination of the variables of the vectors $\boldsymbol{\Delta x_1}$, $\boldsymbol{\Delta x_2}$, $\boldsymbol{\Delta x_3}$ from the optimization problem as formulated with (69) ... (74). Thus the goal is to formulate the optimization problem only in variables of the vector $\boldsymbol{\Delta x_4}$ ($\boldsymbol{\Delta x_4}$ represents the complex nodal voltages). The loss optimization problem is now as follows:

$$\text{Minimize } \mathcal{F}_{loss} = \mathcal{F}_{loss}^{\prime k} + \mathbf{c}^{\prime T^k} \Delta\mathbf{x_4} + \frac{1}{2} \Delta\mathbf{x_4}^T \mathbf{M_4}^{\prime k} \Delta\mathbf{x_4} \tag{85}$$

subject to the equality constraints $\mathbf{g_4}$ (Note that a quadratic approximation is used for the variables $\boldsymbol{\Delta x_1}$, $\boldsymbol{\Delta x_2}$, $\boldsymbol{\Delta x_3}$ for the substitution in the loss-objective function, however, a linearized approximation is used for the variables $\boldsymbol{\Delta x_1}$, $\boldsymbol{\Delta \mathrm{x}_2}$, $\boldsymbol{\Delta x_3}$) in the constraint sets:

$$\mathbf{g}_4^{\prime k} + \mathbf{J_4}^{\prime k} \Delta\mathbf{x_4} = \mathbf{0} \tag{86}$$

The same holds for the inequality constraint set $\mathbf{h}$:

$$\mathbf{h}(\mathbf{x^k}) + \mathbf{A_4}^{\prime k} \Delta\mathbf{x} \leq \mathbf{0} \tag{87}$$

with

$$\mathcal{F}_{loss}^{\prime k} = \mathcal{F}_{loss}(\mathbf{x}_1^k, \mathbf{x}_2^k) - \mathbf{1}^T \mathbf{g_1}(\mathbf{x_1^k}, \mathbf{x_4^k}) - \mathbf{1}^T \mathbf{g_2}(\mathbf{x_2^k}, \mathbf{x_4^k})$$

$$\mathbf{c}^{\prime T^k} = -\mathbf{1}^T \mathbf{J_{14}}^k - \mathbf{1}^T \mathbf{J_{24}}^k \;\; ; \mathbf{M_4}^{\prime k} = -\sum_{i=1}^{m} \mathbf{M_{14}}_i^k - \sum_{i=1}^{l} \mathbf{M_{24}}_i^k$$

$$\mathbf{J_4}^{\prime k} = \mathbf{J}_{44}^k - \sum_{i=1}^{3} \left(\mathbf{J}_{4i}^k \mathbf{J}_{i4}^k \right) \;\; ; \mathbf{g}_4^{\prime k} = \mathbf{g}_4(\mathbf{x}^k) - \sum_{i=1}^{3} \mathbf{J}_{4i}^k \mathbf{g}_i(\mathbf{x}^k)$$

$$\mathbf{A_4}^{\prime k} = \mathbf{A}_4^k - \sum_{i=1}^{3} \left(\mathbf{J}_{i4}^k \mathbf{A}_i^k \right)$$

Assuming that a power flow has been calculated with high accuracy for the solution point $\mathbf{x^k}$, the following is valid: $\mathbf{g}(\mathbf{x}^k) = \mathbf{0}$. This leads to some simplifications in the above formulas:

$\mathcal{F}_{loss}^{\prime k} = \mathcal{F}_{loss}(\mathbf{x}_1^k, \mathbf{x}_2^k)$; $\mathbf{g}_4^{\prime k} = \mathbf{0}$

The optimization problem formulated with (85) ... (87) is a QP formulation. Note that the matrices are still sparse. The optimization problem is now stated with the variables of the vector $\mathbf{\Delta x_4}$, i.e. the nodal voltage related variables.

Solving this problem with a standard QP program is possible, however, due to the large dimension of the problem ($2N$ variables), the number of non-zero matrix and vector elements gets very large, as long as no sparsity techniques are applied during the QP solution process.

In the following a derivation is given where the number of OPF variables is again reduced to a much smaller set. It must be noted, however, that sparsity is lost by doing the following steps.

4.4.2 Compact, non-sparse QP loss optimization problem (Quadratic incremental power flow)

The goal of this OPF loss formulation is to reduce the variable set to the same set as used for the compact QP cost optimization formulation as shown in the previous subsection 4.3.2. There are several ways to derive compact, non-sparse loss QP-optimization problem formulations. All these derivations have in common that at some point linearizations have to be applied to the original quadratic approximations of the equality constraints.

Without showing the derivations the compact loss optimization problem formulation is as follows:

$$\text{Minimize } \mathcal{F}_{loss} = \mathcal{F}_{loss}(\mathbf{x_1^k}, \mathbf{x_2^k}) + \Delta x_{1_N} + \mathbf{1}^T \Delta \mathbf{x}_1' + \begin{bmatrix} \mathbf{1}^T & \mathbf{0}^T \end{bmatrix} \Delta \mathbf{x}_2' \quad (88)$$

subject to

$$\begin{aligned} \alpha_{1_N} \Delta x_{1_N} \quad & + \boldsymbol{\alpha_1}'^{T^k} \Delta \mathbf{x}_1' + \boldsymbol{\alpha_2}^{T^k} \Delta \mathbf{x}_2' \\ & + \frac{1}{2} \left(\begin{bmatrix} \Delta \mathbf{x}_1'^T \Delta \mathbf{x}_2'^T \end{bmatrix} \mathbf{Q_{Loss}}^k \begin{bmatrix} \mathbf{\Delta x}_1' \\ \mathbf{\Delta x}_2' \end{bmatrix} \right) = \alpha_0 \end{aligned} \quad (89)$$

((89) is an extension to (83) and is called the **quadratic incremental power flow** equation.)
and

$$\mathbf{h}'(\mathbf{x_1^k}, \mathbf{x_2'^k}) + \mathbf{A}_1'^k \Delta \mathbf{x}_1' + \mathbf{A}_2'^k \Delta \mathbf{x}_2' \leq \mathbf{0} \quad (90)$$

Note that

- $\Delta \mathbf{x_1}^T = \left[\Delta \mathbf{x}_1'^T \Delta x_{1_N} \right]$. The separation of this vector into two parts is only needed for conceptual reasons.

- $\left[\mathbf{1}^T \;\; \mathbf{0}^T \right]$: This has to be represented in such a way, since the losses are, in the reduced variable set form, a linear function of the active power variables of PQ load nodes with manageable active load which represent only a subset of the vector $\Delta \mathbf{x_2}'$.

- $\Delta \mathbf{x_2}'$: This is the same vector of non-eliminated variables as in the compact cost optimization problem.

- in (89) the same variables appear as in the compact cost optimization problem (82) ... (84).

- one variable (Δx_{1_N}) does not show a quadratic extension in the equality constraint formulation of (89).

- the inequality constraints formulation of (90) is identical to the one of (84). However, it is assumed that no limits will be active for functions of the variable Δx_{1_N}. This can be justified by using an active power of a generator as this variable which is far away from its limits and/or which is not sensitive to optimum solution movements for different OPF problem conditions. This is important because this variable will be eliminated, as discussed below and it should not create any quadratic terms in the (linear) QP inequality constraint set. This assumption can be justified since usually no functions of this variable (it is an active generation variable) are used for inequality or equality constraints formulations. Only the variable itself (i.e. the corresponding active generation) can in principle be limited. In the actual OPF implementation care has to be taken that this variable should not be limited at the OPF optimum.

The OPF problem of (88) ... (90) can be transformed into a classical QP formulation by eliminating the variable Δx_{1_N}, i.e. replacing it in the objective function by the other non-eliminated variables of (89):

$$
\begin{aligned}
\text{Minimize } \mathcal{F}_{loss} \;\; &= \mathcal{F}_{loss}(\mathbf{x_1^k}, \mathbf{x_2^k}) + \left(\mathbf{1} - \frac{\boldsymbol{\alpha_1}'}{\alpha_{1_N}} \right)^{T^k} \Delta \mathbf{x}_1' \\
&+ \left(\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} - \frac{\boldsymbol{\alpha_2}'}{\alpha_{1_N}} \right)^{T^k} \Delta \mathbf{x}_2' \\
&- \frac{1}{2\alpha_{1_N}} \left(\left[\Delta \mathbf{x}_1'^T \Delta \mathbf{x}_2'^T \right] \mathbf{Q_{Loss}}^k \begin{bmatrix} \boldsymbol{\Delta} \mathbf{x}_1' \\ \boldsymbol{\Delta} \mathbf{x}_2' \end{bmatrix} \right) + \frac{\alpha_0}{\alpha_{1_N}}
\end{aligned} \tag{91}
$$

subject to

$$\mathbf{h}'(\mathbf{x_1^k}, \mathbf{x_2'^k}) + \mathbf{A_1'}^k \Delta\mathbf{x_1'} + \mathbf{A_2'}^k \Delta\mathbf{x_2'} \leq \mathbf{0} \tag{92}$$

The exact derivation of the matrix $\mathbf{Q_{Loss}}^k$ cannot be given in this paper due to space limitations. Note however, that several derivations are possible. The problem to be solved is always to find the point at which during the derivations the quadratic terms are to be neglected or replaced by a linear approximation.

Note that an exact computation of this matrix $\mathbf{Q_{Loss}}^k$ can be very CPU time consuming and is usually not worth the effort [18]. The key in this OPF method is the right approximation of the quadratic terms by the right variable. It has been shown with prototypes that even a diagonal matrix approximation for the matrix $\mathbf{Q_{Loss}}^k$ can lead to good and fast convergence. However in any case, care has to be exercised by these approximations: They are the driving values for the optimization, i.e. they determine how fast the variables move towards the optimum, how much they move during the intermediate QP steps. Research is still going on in this area of OPF problem formulation and solutions look quite promising.

4.5 Class A algorithms: Linear programming (LP)

4.5.1 LP formulation

In the following formulations will be given which lead to practical applications of linear programming and finally to efficient programs.

According to class A a basic requirement is the derivation of linearized relations for the load flow. This can be either in the form of the Jacobian

$$\mathbf{J}\Delta\mathbf{x} = \mathbf{0} \tag{93}$$

or in the form of the incremental power flow

$$\boldsymbol{\alpha}_1^T \Delta\mathbf{x}' + \Delta x_{1N} = 0 \tag{94}$$

Note that in (94), as compared to (83), the equality constraint has been normalized in such a way that the factor associated with the variable x_{1_N} is 1. For both (93) and (94) it is assumed that a power flow has been solved with high accuracy around the operating point, leading to a right hand side value of $\mathbf{0}$.

Both forms (93), (94) can be readily incorporated in an LP-tableau.

Since these forms are equality constraints a part of the variables may be eliminated according to the requirements of the LP-algorithm.

A delicate problem is the formation of a linearized objective function. Thereby it is to be observed that the LP-algorithm requires a separable objective function

$$\text{Minimize} \quad \mathcal{F} = \mathbf{c}^T \mathbf{x} \tag{95}$$

Cost curves are a good example of separable objective functions. Quadratic cost curves for each generator are assumed to be the true cost curves for the following derivations. Note that general smooth, convex cost curves could also be taken and similar derivations could be made.

With quadratic cost curves the optimization problem is as follows:

$$\text{Minimize} \quad \mathcal{F}_{cost} = \sum C_i \tag{96}$$

$$\text{where} \quad C_i = \frac{1}{2} q_i P_i^2 + c_i P_i + \mathcal{C}_{i_o}$$

(separable quadratic cost functions)

In order to use an LP algorithm for the solution of this optimization problem a further approximate step must be considered, namely the conversion of real cost curves to piece-wise linear curves which can be done to any desired accuracy, see schematic sketch in Figure 1.

Abbildung 1: Cost curves (piece-wise linear)

An analytic expression for the approximation for the generating cost of one generating unit is

$$\begin{aligned} C_i &\geq d_{o_{1_i}} + d_{1_i} P_i \\ C_i &\geq d_{o_{2_i}} + d_{2_i} P_i \\ C_i &\geq d_{o_{3_i}} + d_{3_i} P_i \end{aligned} \tag{97}$$

Thereby the expressions $d_{o_{j_i}} + d_{j_i} P_i$ represent the straight lines which form the approximation to the quadratic cost curve.

For the purposes of the class A algorithm this model has to be converted to an incremental form whereby both costs and generating powers appear as variables.

$$C_{i_o} + \Delta C_i \geq d_{o_{j_i}} + d_{j_i}(P_{i_o} + \Delta P_i) \tag{98}$$

The vectors $\mathbf{\Delta C}$ and $\mathbf{\Delta P}$ may be replaced by general x_i- variables:

$$\begin{aligned} &\mathbf{\Delta P} \quad ...\Delta \mathbf{x}_1 \\ &\mathbf{\Delta C} \quad ...\Delta \mathbf{x}_5 \end{aligned} \tag{99}$$

Then

$$\mathcal{F}(\mathbf{x}) = C_o + [1, 1, 1, ...1]\, \Delta \mathbf{x}_5 \tag{100}$$

subject to

$$\mathbf{diag}\,(\mathbf{d}_j)_i\, \Delta \mathbf{x}_1 - \Delta \mathbf{x}_{5_i} \leq \mathbf{C}_{o_i} - \mathbf{d}_{o_{j_i}} \mathbf{P}_{i_o} \tag{101}$$

(i = 1, 2, ... m (m: Number of generators to be optimized); j = 1, 2, ... S (S: number of straight line sections per generator))

Here it becomes obvious that the formulation of the cost function leads to numerous entries in the LP-tableau. At this point a relatively small number of straight line sections for generators is considered only so as to limit the size of the LP-tableau.

There are further relations in the form of inequality constraints to be considered for the tableau, namely limits on the control variables and functional constraints.

Again, the reasons of keeping the tableau small, generating powers P_i are considered as control variables only.

Hence, limits and functional constraints are given by

$$\begin{aligned} &+-^{**}\, \Delta \mathbf{x}_1^{'} &&\leq \mathbf{b}_v &&\text{(variable limits)} \\ &\mathbf{A}^{'} \Delta \mathbf{x}_1^{'} &&\leq \mathbf{b}_{fc}^{'} &&\text{(functional constraints)} \end{aligned} \tag{102}$$

(**: meaning that both the upper and lower limits of the variables must be considered)
where $\mathbf{A}^{'}$ can be full.

Beyond that there is the incremental power flow which is taken as the scalar equality constraint. It must be incorporated in the tableau. This is done by eliminating one of the control variables.

Thus the LP problem is given by

$$\mathrm{M}inimize \;\; \mathcal{F}_{cost} = [1, 1, 1, ...]\, \Delta \mathbf{x}_5 \tag{103}$$

subject to

$$\begin{array}{llll} \mathbf{diag}\,(\mathbf{d}_j)_i\,\Delta\mathbf{x}_1^{'} & -\Delta\mathbf{x}_{5_i} & \leq \mathbf{C}_{i_o} - \mathbf{d}_{o_{j_i}}\mathbf{P}_{i_o} & \text{(i = 1, 2, ... m-1)} \\ +-\Delta\mathbf{x}_1^{'} & & \leq \mathbf{b}_v & \\ \mathbf{A}^{'}\Delta\mathbf{x}_1^{'} & & \leq \mathbf{b}_{fc}^{'} & \\ \mathbf{D}\Delta\mathbf{x}_1^{'} & -\Delta\mathbf{x}_{5_m} & \leq \mathbf{C}_{m_o} - \mathbf{d}_{o_{jm}} & \text{(only } m^{th} \text{ variable)} \end{array} \tag{104}$$

The last entry is due to the elimination of the equality constraint. Hence $\Delta\mathbf{x}_1^{'}$ comprises $m-1$ variables only (m = number of generating powers to be optimized).

It must also been observed that $\mathbf{x}_5$ is not constrained.

As the set of relations above stands it is quite sparse which may be an advantage depending on the method of solution to be chosen.

If a small number of variables is desired the variables of the vector $\mathbf{x}_5$ can be eliminated and expressed by components of $\mathbf{x}_1$ which leads to a tableau whose variables are control variables only (generating powers).

This general approach to the use of LP within class A algorithm may be extended to other OPF-problems as long as a separable cost function can be formulated.

A most recent application of this kind is loss minimization (Stott, [26]) whereby losses are approximated by linearized relations in terms of active and reactive injections. A basic requirement in this approach is an exact representation of the linear incremental power flow. The segments to the left and the right of the operating point need not be accurate.

The problem of choosing the right approximation is pronounced in the case of loss minimization by reactive injections only. As long as there is no technical constraint on reactive injections the straight line subsections are the only means for the limitation of the variables. The subsections must be made artificially smaller in the course of the iterations (e.g. dichotomy).

4.5.2 LP-solution

For purposes of illustration this particular method of solution is dispensed in more detail thereby referring to the standard LP method in appendix A.1.

The starting point is an operating point of the power system given by a load flow solution. This solution is designated by the vector of P_{o_i}'s around which an improved solution is sought. According to the linearized model the individual P_{o_i}'s are located at the breakpoints of the straight line sections (besides one variable). The situation for one generator is depicted by the sketch in Fig. 2.

Abbildung 2: Change of segment in piece-wise linear cost curves

Since an incremental model is used it is to be observed that the increments must be feasible

$$\Delta P_i \geq 0 \ \text{ or } \ \Delta x_1^{'} \geq 0 \tag{105}$$

For this purpose each generator power variation is to be modeled by two LP-variables as indicated in Fig. 2.

At this point it is assumed that the vector $\mathbf{\Delta x}_5$ is eliminated and substituted by $\mathbf{\Delta x}_1^{'}$. As a consequence the cost function is modified and will consist of $\mathbf{c}^T \mathbf{\Delta x}_1^{'}$ whereby the c's are the result of a transformation.

$$\text{Minimize } \ \mathcal{F}_{cost} = \mathbf{c}^T \mathbf{\Delta x}_1^{'} \tag{106}$$

Since the starting point was a solution to the load flow and, of course, to a previous LP step the vector $\mathbf{\Delta x}_1^{'}$ is zero and can be considered the non basic vector of the LP tableau (see appendix A.1). Thereby it is taken for granted that at this point no control variables are exceeded. Functional constraints are not considered at the moment.

Thus, a classical LP tableau can be established whereby the vector $\mathbf{\Delta x}_1^{'}$ corresponds to the non-basic solution $\mathbf{x}_D$ of the tableau. The slack variables (Luenberger [8]) are the basic variables.

The relative cost vector will indicate which variable will have to become a basic variable.

The LP-tableau is exactly the one in Luenberger [8].

A change of base may be caused by one of the following items:

- due to the change of α's for the new load flow solution a cost coefficient has changed sign
- the straight line approximation to the quadratic cost curve of a generator has been changed.

These items are assumed to be of such a nature that a cost coefficient has changed its sign.

Beyond that there are indications that constraints and limits have been exceeded. These may be due to

- a change in the straight line approximations of the cost curves
- functional constraints which have not been considered so far

- consequences of an updated load flow solution, e.g. the m^{th} control variable not explicit in the tableau has exceeded its limits

These constraint violations require another type of change of base as explained in appendix A.1.2.

The necessary changeover to a feasible solution may be performed step-by-step, i.e. constraint by constraint in order to keep the tableau small.

The computational effort in using the linear programming method depends on

- the number of update operations for the incremental power flow
- the number of update operations for the inequalities

Updating on the right hand side is not very demanding. Updating the coefficient of the tableau results in a complete recalculation of the partially inverted tableau. In the iterative process updating is necessary whenever a new load flow solution becomes available.

It is obvious that the overall effort depends on the dimension of the tableau which can be kept to a minimum if the cost curves are modelled by small number of segments (straight line approximations). However, in order to achieve the required accuracy the lengths of the segments have to be reduced as the number of iterations increases. This process is called segment refinement.

The idea of segment refinement is to keep the number of segments in the tableau fixed and to reduce the lengths of the segments.

One possible procedure is the following: The tableau always comprises a fixed number of segments which cannot be less than two, if limits (artificial or real) are applied on the outside of the segments or four, if the limits are located at a distance of the operating region.

Whenever an optimal solution for a given segmentation is found the lengths of the segments are reduced thereby changing the coefficients of the rows in the tableau corresponding to the representation of the cost curves. If at this point the solution turns out to be infeasible a change of base has to be performed as outlined in the appendix A.1. (problem a).

From here on the refinement process can be continued or a new load flow solution can be asked. The decision will depend on the segment size, the relative improvement of the objective function and the mismatches at the iteration where the optimization is performed.

The overall effort depends on the dimension of the tableau which can be kept to a minimum if the cost curves are modelled by a small number of segments only, namely in an adaptive way in the vicinity of the solution point (segment refinement). However, adapting the segments also requires updating of the tableau.

Finally, the various steps in the course of one iteration will be as follows:

- 1. solve an ordinary load flow
- 2. extract Jacobian or incremental power flow
- 3. create or update segments of cost function, form functional constraints
- 4. generate LP-tableau
- 5. solve LP
- 6. check: size of segments; active limits; size of corrections resulting from LP
- 7. if corrections, steps etc. small enough stop, otherwise go to 1.

The effectiveness of linear programming in class A methods will depend on the programming skill, in particular in handling the tableau, base change operations, updating and segment refinement.

4.6 Class A algorithms: Quadratic programming (QP)

4.6.1 QP formulation

As under 4.5.1 a basic requirement is the derivation of linearized relations for the load flow. Again this can be done by taking the Jacobian (93) or by working out the incremental power flow (94). Either form will be needed in the formulation of the Lagrangian which plays a central role in QP.

The objective function can either be quadratic (cost) or linear (losses) as given by the relations (68) and (69).

The quadratic function describing operating cost consists of a quadratic form having a diagonal matrix only (separable functions) as given by (78).

$$\text{Minimize } \mathcal{F}_{cost} = \mathcal{F}_{cost}(\mathbf{x_1}^k) + \mathbf{c}^{Tk}\Delta\mathbf{x_1} + \frac{1}{2}\Delta\mathbf{x_1}^T\mathbf{diag}(q_i^k)\Delta\mathbf{x_1} \qquad (107)$$

In order to convert the loss minimization problem to a quadratic one the incremental power flow is extended as explained in chapter 4.4. Thereby a number of variables is eliminated and the incremental power flow is incorporated in the objective function yielding the relation (91). This can be done if the slack power can be expressed by other non-eliminated variables, i.e. active power, voltage magnitude or reactive injections variables.

The problem is thus brought into a form where a quadratic objective function is left without the need to consider an equality constraint any further. This can also be understood by the fact that the n^{th} reactive injection need not be considered since there is no cost attached to it.

In the cost minimization problem (MW dispatch) the equality constraint cannot be eliminated because all control variables have a quadratic or in general convex, non-linear cost function.

Thus, the general QP-problem is formulated as follows

$$\text{Minimize } \mathcal{F} = \mathcal{F}^k + \mathbf{c}^T \Delta\mathbf{x} + \frac{1}{2} \Delta\mathbf{x}^T \mathbf{Q} \Delta\mathbf{x} \tag{108}$$

subject to

$$\mathbf{g}(\mathbf{x}^k) + \mathbf{J}\Delta\mathbf{x} = \mathbf{0} \tag{109}$$

As outlined above the equality constraint disappears when a compact loss minimization problem with a reduced variable set is considered.

Beyond that variable and functional constraints have to be attached which in general will be given by

$$\mathbf{h}(\mathbf{x}^k) + \mathbf{A}\Delta\mathbf{x} \leq \mathbf{0} \tag{110}$$

Here $\Delta\mathbf{x}$ is understood as the deviation of the control variable from its operating point as determined by the power flow.

At this point the Lagrangian in terms of the deviations can be formulated as

$$\begin{aligned} \mathcal{L} &= \mathbf{c}^T \Delta\mathbf{x} + \tfrac{1}{2} \Delta\mathbf{x}^T \mathbf{Q} \Delta\mathbf{x} \\ &+ \lambda^T (\mathbf{g}(\mathbf{x}^k) + \mathbf{J}\Delta\mathbf{x}) + \mu^T (\mathbf{h}(\mathbf{x}^k) + \mathbf{A}\Delta\mathbf{x}) \Rightarrow min. \end{aligned} \tag{111}$$

Since the Lagrangian in this form is quadratic one of the QP- algorithms may be applied for the solution of the QP-problem.

4.6.2 QP solution

The Lagrangian above or its components are suitable for a direct application of a QP-algorithm.

One example is the use of the Beale algorithm which is successful for networks up to about 250 - 300 nodes and to 50 - 80 control variables.

For larger networks other methods have to be used.

For the dispatch problem (MW-Dispatch), i.e. cost minimization the method outlined under A.2 is quite suitable. The important feature of the dispatch problem is the fact that $\mathbf{Q}$ is a diagonal matrix and the equality constraint is a scalar only.

The system to be treated for the unconstrained solution is extremely sparse as shown below

$$\mathbf{M}\mathbf{u_o} = \begin{bmatrix} -\mathbf{c} \\ \mathbf{b}_1 \end{bmatrix} \tag{112}$$

Due to the sparsity of the matrix M the formation of

$$-\begin{bmatrix}\mathbf{A} & \mathbf{0}\end{bmatrix}\mathbf{M}^{-1}\begin{bmatrix} \mathbf{A}^T \\ \mathbf{0} \end{bmatrix} \tag{113}$$

will benefit considerably from various sparsity techniques.

As explained under A.2 the further steps are LP-like and in the end the final solution is obtained by superposition.

$$\mathbf{u}_c = \mathbf{u}_o + \Delta\mathbf{u} \tag{114}$$

Working with this method will show that it is advisable to add constraints step by step, in particular functional constraints in order to maintain a small tableau.

The interesting feature of the lastmentioned algorithm is that it is fully based on linear methods. In a first step the unconstrained problem is linear. The superimposed corrections are determined by linear programming methods. The linear methods are fully effective if the sparsity of the system can be exploited.

In summary, the various steps in the course of one iteration will be as follows:

- 1. solve an ordinary load flow
- 2. extract Jacobian or incremental power flow
- (2.a. extract extended incremental power flow for loss minimization)
- 3. setup sparse system which determines the unconstrained solution
- 4. generate LP-tableau
- 5. solve LP
- 6. determine superimposed solution and update
- 7. if corrections, steps, etc. small enough stop, otherwise go to 1.

4.7 Summary

In summary the class A OPF algorithms are based on the iterative and separate use of the power flow to solve for a given operating point and a LP or QP for the optimization problem around the power flow solution.

The power flow part of these class A OPF algorithms is the conventional power flow as known from student text books. All special features like PV-PQ node type switching, local tap control can be handled by the power flow.

The classical LP and QP algorithms as described in mathematical text books are often quite slow for the solution of the OPF optimization problem. In the appendix some points are discussed about efficient handling of the LP and QP algorithms considering the special features of the OPF.

In principle the only necessary link between the power flow part and the optimization part is the transfer of the operating point x^k, representing the OPF variables: The power flow solution is transferred to the OPF to be used as the solution around which the approximations are made. Then the LP or QP algorithm is solved. The optimal solution (note: optimality is valid only with respect to the approximations around the previous power flow solution) is transferred back to the power flow and represents another power flow input data set. The power flow corrects the approximations made in the preceding LP or QP optimization. Thus the power flow adapts the nodal voltages and the slack power such that the mismatches are below predefined, small tolerance values. By executing this procedure several times the power solution point tends to go toward the optimum, i.e. the result of the very last LP or QP solution should be identical (within a certain tolerance) to the preceding power flow solution. At this point the optimal solution is reached.

5 OPF CLASS B. POWER FLOW INTEGRATED IN OPTIMIZATION ALGORITHM

5.1 Introduction

In this section the OPF formulation is solved by an integrated method as compared to the OPF formulation of the Class A where the power flow is separated from the optimization part.

First the easiest case is discussed: The solution of the OPF problem with a given set of equality constraints only. Although this certainly does not satisfy the real-world constraints (which would include inequality constraints), it is discussed here in order to show the principles of the Newton-Raphson based approach which are also used in the following sections. There the more realistic OPF problem is solved with consideration of both equality and inequality constraints.

The objective function will usually be formulated as a general function $\mathcal{F}(\mathbf{x})$, however, where the OPF algorithm results in special cases for either cost or loss objective functions special discussion is given.

The same holds for the inequality constraints $\mathbf{h}(\mathbf{x})$: When any special derivation results this is discussed.

5.2 Solution of OPF with equality constraints only

The problem is as follows:

$$\begin{aligned} &\text{Minimize} \quad \mathcal{F}(\mathbf{x}) \\ &\text{subject to} \;\; \mathbf{g}(\mathbf{x}) = \mathbf{0} \end{aligned} \tag{115}$$

The solution is based on the Lagrange formulation (the index $_{eq}$ refers to the **equality** constrained OPF problem):

$$\mathcal{L}_{eq} = \mathcal{F}(\mathbf{x}) + \lambda^T \mathbf{g}(\mathbf{x}) \tag{116}$$

The optimality conditions for (116) are:

$$\begin{aligned} \frac{\partial \mathcal{L}_{eq}}{\partial \mathbf{x}} &= \frac{\partial}{\partial \mathbf{x}} \left(\mathcal{F}(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x}) \right) \Big|_{\mathbf{x}=\widehat{\mathbf{x}}, \lambda=\widehat{\lambda}} &= \mathbf{0} \\ \frac{\partial \mathcal{L}_{eq}}{\partial \boldsymbol{\lambda}} &= \mathbf{g}(\mathbf{x}) \Big|_{\mathbf{x}=\widehat{\mathbf{x}}, \lambda=\widehat{\lambda}} &= \mathbf{0} \end{aligned} \tag{117}$$

In (117) the following substitutions can be made; $\mathbf{J}$ is the Jacobian matrix:

$$\mathbf{J} = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} \tag{118}$$

Thus the following system has to be solved to achieve these optimality conditions:

$$\begin{aligned} \tfrac{\partial \mathcal{F}(\mathbf{x})}{\partial \mathbf{x}} + \mathbf{J}^T \boldsymbol{\lambda} &= \mathbf{0} \\ \mathbf{g}(\mathbf{x}) &= \mathbf{0} \end{aligned} \tag{119}$$

(119) can be summarized as one non-linear system:

$$\mathbf{W}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0} \tag{120}$$

This non-linear system must be solved by any efficient method. General mathematical methods for solving non-linear systems can be used. However, the solution based on the Newton approach is most often employed.

5.2.1 Newton based solution

(119) or (120) can be solved by the iterative Newton-Raphson approach which leads to the following linear system for the solution of (120) (the index ^{k} refers to the value of the associated variable at iteration k):

$$\mathbf{W}(\mathbf{x^k}, \lambda^{\mathbf{k}}) + \left.\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{x}}\right|_{\mathbf{x}=\boldsymbol{x}^k, \boldsymbol{\lambda}=\boldsymbol{\lambda}^k} \Delta \boldsymbol{x}^{\boldsymbol{k}} + \left.\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{\lambda}}\right|_{\mathbf{x}=\boldsymbol{x}^k, \boldsymbol{\lambda}=\boldsymbol{\lambda}^k} \Delta \boldsymbol{\lambda}^{\boldsymbol{k}} = \mathbf{0} \tag{121}$$

Now, the linear system which must be solved iteratively, takes the form:

$$\begin{bmatrix} \mathbf{H} & \mathbf{J}^T \\ \mathbf{J} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta}\mathbf{x^k} \\ \boldsymbol{\Delta\lambda}^{\mathbf{k}} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^k \\ \mathbf{g}^k \end{bmatrix} \tag{122}$$

with

$$\mathbf{H} = \mathbf{H_{eq}} = \frac{\partial \mathcal{F}^2(\boldsymbol{x})}{\partial \boldsymbol{x}^2} + \mathrm{diag}(\boldsymbol{\lambda}) \frac{\partial \boldsymbol{g}^2(\boldsymbol{x})}{\partial \boldsymbol{x}^2} \tag{123}$$

and

$$\begin{bmatrix} \mathbf{r}^k \\ \mathbf{g}^k \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{eq^k} \\ \mathbf{g}_{eq^k} \end{bmatrix} = \begin{bmatrix} -\left(\tfrac{\partial \mathcal{F}(\mathbf{x})}{\partial \mathbf{x}} + \mathbf{J^T}\lambda\right)\Big|_{\mathbf{x^k}, \lambda^{\mathbf{k}}} \\ -\mathbf{g}(\mathbf{x^k}) \end{bmatrix} \tag{124}$$

(122) is solved iteratively, i.e. the values for $\mathbf{x}$ and λ from the previous iteration are inserted into $\mathbf{H}$ and $\mathbf{J}$ and the right hand side of (122). Then (122) is solved for $\Delta\mathbf{x}$ and $\Delta\lambda$ which again are used to update the values for $\mathbf{x}$ and λ as follows:

$$\mathbf{x^{k+1}} = \mathbf{x^k} + \Delta\mathbf{x^k} \tag{125}$$

$$\lambda^{\mathbf{k+1}} = \lambda^{\mathbf{k}} + \Delta\lambda^{\mathbf{k}}$$

Doing this for some iterations will usually result in a convergent solution. This solution is the optimum for the OPF equality constrained problem as given in (115), i.e. the resulting values for $\mathbf{x}$ and λ are the values where the objective function $\mathcal{F}(\mathbf{x})$ is minimal and where all equality constraints $\mathrm{g}(\mathbf{x})$ are satisfied.

(122) is a linear system which, in principle, can be solved by any linear equation solving algorithm. Note, however, that the matrices can be very sparse and thus specialized sparsity algorithms must be applied to solve the system efficiently [21].

Decoupling principles as used in the decoupled power flow could be used if polar coordinates are chosen. However, experience has shown that for the OPF decoupling can have drawbacks when looking at overall robustness. However, in general, most algorithms which have been developed for power flows, can be applied to the equality constrained OPF problem with little modifications.

The conclusion from this subsection is, that whenever the equality constraint for an OPF problem is given the solution is not more difficult than the solution of an ordinary power flow problem. The problem, however, are the inequality constraints. If one would know beforehand which inequality constraints will be active, i.e. limited in the OPF optimum, one could include these constraints as equality constraints from the beginning of the optimization and solve with the procedure discussed above.

The active set of inequality constraints, however, is not known in advance and thus special algorithms have to be found to determine whether to make an inequality constraint active or not. This is discussed in the next subsection.

5.3 OPF solution with consideration of inequality constraints

5.3.1 Introduction

The Kuhn-Tucker conditions (see (61)) determine if at any solution point a relative optimum has been found, i.e. for all inequality constraints which has been included in the active constraint set, the Lagrange multiplier μ must be positive in order to justify the inclusion of the corresponding inequality constraint in the active set. This active set includes all inequality constraints being binding at their respective limits. In the OPF class B, discussed in this section, two approaches are used to solve the inequality constraints problem: The handling of inequality constraints by penalty techniques, mainly used for variable related limits and the explicit modelling of functional inequality constraints as functional equality constraints, once they become active at their limits. Note that active functional constraints can also be modelled by the penalty approach.

The penalty based approach leads to an extension of the equality constrained OPF problem as discussed in the previous subsection, i.e. the possible inequality constraints are handled in a quadratic form as extensions to the original objective function. By using small or large weights (penalties) for these additional quadratic objective functions terms, the equality constrained OPF problem is forced to a solution which is optimal with respect to the equality constraint set, but in addition to that, considers the inequality constraints. Those with a large weighting factor, will have the effect of being binding, i.e. limited, those with small weighting factors will be free, i.e., these inequality constraints will not be binding at their limits in the OPF optimum. In summary, this penalty technique based approach can be seen as an equality constrained OPF problem with an artificially extended objective function.

This approach has one problem: When should an inequality constraint be held at its limit and when should it be freed.

It must be noted that there are no penalty based approaches known today, for solving the Kuhn-Tucker conditions with straightforward solution processes. Today, in order to improve speed, convergence and robustness, trial passes, heuristics or other similar measures are used in this approach. The use of very fast sparsity routines for updating factorized matrices, to add or remove rows and columns is usually the selling point for the penalty based methods for the OPF problems. Without them this approach would not make much sense, since very quickly they would become slow and the use of some heuristics or trial iterations for the determination of the correct constraint set could not be justified any more.

5.3.2 Penalty term approaches for handling inequality constraints

When using the penalty term approach two main categories of inequality constraints can be distinguished.

- Limits on OPF variables

- Limits on output variables, i.e. non-linear or linear functions of OPF variables

The distinction is done because these two types can be handled with different efficiency in the penalty term based OPF algorithms. Among the various constraints of these categories most can be treated in the same way in the algorithms. However, there are distinct differences between the implementations of these types.

In the following subsections the penalty term approaches for the two inequality constraint types are discussed.

Limits on OPF variables The general idea of the penalty term techniques is to add an additional quadratic function for every inequality constraint to the original objective function. By using large weights for these quadratic functions, the optimization algorithm is forced to move constraint values, which are thus made artificially expensive, to desired limit values. The effect of this penalty term technique corresponds to including the violated constraint into the active set.

The function added to the original objective function looks as follows with limited OPF variables x_i:

$$\mathcal{L} = \mathcal{L}_{eq} + \sum \left(\frac{1}{2} \mathcal{W}_i (x_i - x_{i_{Lim}})^2 \right) \tag{126}$$

In (126), the Lagrangian $\mathcal{L}_{eq}$ corresponds to the Lagrangian as given in (116) of the equality constrained OPF problem. The $\sum$ goes over all control variables x which could become limited at the OPF optimum.

The Lagrange optimality conditions are derived in exactly the same way as in (119). The main difference lies in the derivatives of $\mathcal{L}$ with respect to the variables $\mathbf{x}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial \mathcal{L}_{eq}}{\partial \mathbf{x}} + \mathrm{diag}\left(\mathcal{W}\right)\left(\mathbf{x} - \mathbf{x}_{Lim}\right) \tag{127}$$

Making now the same derivation as for the equality constrained OPF problem, i.e. solve the optimality conditions by an iterative Newton solution, the matrices $\mathbf{H}$ and the right hand side of the equation (122) must be adapted:

$$\mathbf{H} = \mathbf{H}_{eq} + \mathrm{diag}(\mathcal{W}) \tag{128}$$

$$\mathbf{r}^k = \mathbf{r}^k_{eq} - \mathrm{diag}(\mathcal{W})(\mathbf{x^k} - \mathbf{x}_{Lim}) \tag{129}$$

Adding the terms for a possibly binding OPF variable i to the original objective function with a large value for $\mathcal{W}_i$ will force the variable x_i within ϵ to its limit value x_{iLim}. The rule is that for larger $\mathcal{W}_i$ smaller ϵ values will result. Note that by adding this term to the objective function, i.e. also to the Lagrangian, the optimality conditions and also the subsequent Newton-based solution matrices are changed. This is shown in the above equations (126) ... (129). In (122) only diagonal terms and the right hand side are changed (see (128)) with this type of constraint which means that a fast factor update technique can be used to update the factorized matrix. A large value $\mathcal{W}_i$ is used to enforce the constraint, a small value $\mathcal{W}_i$ is used to relax the constraint. The sparsity schemes, i.e. the fill-in patterns are not affected whether this constraint is activated or not during the iterations.

Other techniques can be used to speed up this process: Assuming that a variable x_i violates its limit by $+\Delta x_i$ in the present iteration the limit value $x_{i_{Lim}}$ can be shifted by $-\Delta x_i$ so that in the next iteration the variable x_i will be forced near its real limits. Doing this iteratively has the advantage that only the right hand side of the iterative solution process has to be changed and not the matrix factors. However, the speed gain could be offset by less accuracy in the limit enforcement.

The question when to enforce a limit is usually quite simple, i.e. whenever it violates its limit. However, the problem when to relax a variable during the solution process, i.e. when to use small $\mathcal{W}_i$ values, is not as clear. The use of quadratic penalty terms in second order methods, however, tells, if an enforced, highly penalized variable is truly binding or not: If the variable is on the violated side by a value ϵ it can be assumed that the variable is actually binding. If this is not the case, the variable should be freed, i.e. the weight variable must be reduced to a small, non-penalizing value.

Another method is the usage of soft constraints, i.e. the enforcing of an inequality constraint i with a value for $\mathcal{W}_i$ being finite and much less than the maximum value needed for complete inequality constraint enforcement. By doing this an intermediate solution can be obtained which can show which of the variables tend to go their respective limits and which ones not.

It is obvious that the chance of finding the active inequality constraints immediately is quite low. Thus trial iterations can be employed to find a better set of binding inequality constraints. This is usually done by holding the matrices involved constant, i.e. no refactorization in done. Only the set of possibly binding constraints is changed from trial iteration to the next. Note, that for this reason, trial iterations can be much faster than the normal Newton-based iterations.

Limits on output variables Output variables are represented by functions of OPF variables. Branch flow or voltage magnitude (only when using rectangular coordinates) constraints are typical examples for this constraint type. Two different ways to implement them are possible. One method is to use the same technique as for state variables, i.e. the addition of quadratic penalty terms for each potentially binding output variable. In the other method those inequality constraints which have been determined by some heuristic method to become active are explicitely added as equality constraints, i.e. they are treated in exactly the same manner as equality constraints.

The treatment of equality constraints has been discussed in the previous subsection. Note, however, that adding or removing equality constraints must be done with consideration of sparsity techniques in order to maintain overall speed. Further a Lagrangian multiplier has to be used whose sign indicates if the constraint should be active or not. This method of handling inequality constraints is not discussed further in this text.

When adding a functional inequality constraint $h_i(x)$ in penalty form, the general form for the Lagrangian function looks as follows :

$$\mathcal{L} = \mathcal{L}_{eq} + \sum_{i=1}^{H} \frac{1}{2} \mathcal{W}_i \left(h_i(\mathbf{x}) - h_{i_{Lim}} \right)^2 \tag{130}$$

(H=number of output variable constraints)

The optimality conditions (first order derivations) and the necessary matrices and right hand sides for the Newton based solution process (second order derivations) are not given here for space reasons. Their derivations, however, are straightforward.

The constraints would be enforced by either changing the weighting factor $\mathcal{W}_i$ or by moving the limits in order to enforce or relax the inequality constraint i. This penalty approach for output variables is, mathematically seen, possible, however, new terms will be created in the optimality condition matrices and its subsequent Newton-based solution process which will need sophisticated matrix-factor updating algorithms in order to maintain a fast solution process. However, the usual output variable inequality constraints do not destroy the general sparse structure of the Newton based OPF solution process and in principle do allow sparsity storage and matrix factor techniques.

The rules to enforce and to relax a variable by changing the weight h_i are in analogy to the procedure for handling limits on OPF variables by penalty techniques. Thus trial iterations, soft limit enforcement and other heuristic techniques can be applied.

However, note, when using penalty techniques, no systematic algorithm exists to determine which inequality constraints should be relaxed and which should be enforced at any stage during the Newton solution process.

Thus, convergence problems are quite common when the network is not tuned to this penalty based approach. Tuned penalty based algorithms for OPF problems can converge well and fast, however, one tuning set might only be valid for a small load variation and must be adapted to other load conditions.

5.4 Summary

The OPF class B algorithms solve iteratively for the Kuhn-Tucker conditions without explicitely using a conventional power flow. Thus in this class B of OPF algorithms all active constraints, i.e. all power flow equality constraints and all binding inequality constraints, the objective function reduction and the OPF variable movements are handled simultaneously. The OPF class B can be compared with the conventional power flow solved with the Newton-Raphson method. The main problem of the OPF class B algorithms lies in the handling of inequality constraints, i.e. the determination of the set of binding inequality constraints. This is done with heuristic methods which include mainly trial iterations and soft limit enforcement.

6 FINAL EVALUATION OF THE METHODS

As with the ordinary power flow OPF methods are judged by their performance with respect to speed, versatility and robustness. At this point in time, however, there is no single OPF method which meets all requirements satisfactorily.

Class A and class B methods have their relative merits and perform well for one or the other particular application. In any one problem, however, a method could show poor performance.

LP methods in class A have the advantage of treating constraints in a systematic and efficient way. However, cost minimization and loss minimization, although being treated by this approach are not equally efficient. Constraints can be treated well in both cases whereas the exact extremum of the objective function can be reached in case of cost minimization only. The loss minimum is approximated.

When applying QP methods in class A both abovementioned problems can be handled accurately. Cost minimization is at least as efficient as with LP. Loss minimization is hampered by the cumbersome quadratic form specifying the objective function and its treatment by the QP algorithm. The experience is, however, that a few iterations are needed only.

Class A methods are also attractive because the starting point is a solved load flow which in most cases represents a feasible solution for the optimization problem. Quite often the iterative solutions in the beginning need not be very accurate. So the total number of load flow iterations is not considerably larger than for an ordinary load flow, e.g. twice as high.

Class B methods are attractive at a first glance . They solve the problem, i.e. they meet the optimality conditions in a global way. Convergence in the Newton approach is very good. However, when considering the way in which constraints have to be handled its attractiveness is moderated. Heuristics and tuning are needed which is somewhat compensated by the advantage that sparsity techniques can be employed, refactorization of the Hessian is avoided and well-known techniques of the ordinary load flow are applicable.

At the moment it seems that class A methods are taking the lead and this will be even more so when LP- and QP-methods are being further improved.

A APPENDIX

A.1 Linear programming (LP) algorithms

A.1.1 The basic linear programming method (Simplex)

In the following a series of LP-methods and -algorithms is presented which follows closely Luenberger [8]. The nomenclature and definitions are taken from there.

The standard linear programming problem is defined as

$$\mathcal{F} = \mathbf{c}^T\mathbf{x} \quad \Rightarrow \quad min \tag{131}$$

subject to:

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned} \tag{132}$$

where

- $\mathbf{x}$ is the vector of unknowns ($\mathbf{x}$ comprises both original and LP-slack variables), dim $\mathbf{x}$ = n
- $\mathbf{c}$ is the vector of cost coefficients
- $\mathbf{A}$ is an m x n matrix
- $\mathbf{b}$ is the vector specifying the constraints, dim $\mathbf{b}$ = m

By partitioning the matrix $\mathbf{A}$ into $\mathbf{B}$ (m x m) and $\mathbf{D}$ (m x n-m), the vector $\mathbf{x}$ into $\mathbf{x_B}$ and $\mathbf{x_D}$ the problem is formulated as

$$\mathcal{F} = \mathbf{c_B}^T\mathbf{x_B} + \mathbf{c_D}^T\mathbf{x_D} \quad \Rightarrow \quad min \tag{133}$$

subject to

$$\begin{aligned} \mathbf{B} \quad \mathbf{x_B} + \mathbf{D} \quad \mathbf{x_D} &= \mathbf{b} \\ \mathbf{x_B} \qquad\qquad &\geq \mathbf{0} \\ \mathbf{x_D} &\geq \mathbf{0} \end{aligned} \tag{134}$$

where

- $\mathbf{B}$ is the basis,

- $\mathbf{x_B}$ is the basic solution and
- $\mathbf{x_D}$ is the non-basic solution.

Since it is known that the optimum solution will be found at one of the feasible basic solutions, the latter are checked only.

At the start it is assumed that a feasible basic solution is available, i.e. $\mathbf{x_B} \geq \mathbf{0}$, $\mathbf{x_D} \geq \mathbf{0}$. Methods will be shown later which allow to find a feasible solution if such one is not given. Then

$$\mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b} \tag{135}$$

or

$$\mathbf{x_B} = \mathbf{B}^{-1}\mathbf{x_B} - \mathbf{B}^{-1}\mathbf{D}\mathbf{x_D} \tag{136}$$

The cost function z is given by

$$\begin{aligned} z &= \mathbf{c_B}^T(\mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{D}\mathbf{x_D}) + \mathbf{c_D}^T\mathbf{x_D} = \\ &= \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{b} + (\mathbf{c_D}^T - \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{D})\mathbf{x_D} \end{aligned} \tag{137}$$

The last term is called the relative cost vector consisting of relative cost coefficients

$$\mathbf{r}^T = \mathbf{c_D}^T - \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{D} \tag{138}$$

These relations are put in a frame which is called the tableau

$$\mathcal{T} = \begin{bmatrix} \mathbf{U} & \mathbf{B}^{-1}\mathbf{D} & \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{0} & \mathbf{c_B}^T - \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{D} & -\mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{b} \end{bmatrix} \tag{139}$$

whereby the left side matrix $\begin{bmatrix} \mathbf{U} \\ \mathbf{0} \end{bmatrix}$ is superfluous and need not be stored or manipulated ($\mathbf{U}$ is a unity matrix).

The tableau contains the following important information.

- $-\mathbf{c_B}^T\mathbf{b}^{-1}$ is the negative value of the cost function of the current base
- $\mathbf{B}^{-1}\mathbf{b}$ is the base vector (current)
- $\mathbf{c_B}^T - \mathbf{c_D}^T\mathbf{B}^{-1}\mathbf{D}$ is a row vector whose elements indicate by their sign if the cost function can be further decreased

A negative sign of an element of the relative cost vector says that a further decrease of the objective function is possible. The change of the corresponding non-basic variable in $\mathbf{x_D}$ against a basic variable in $\mathbf{x_B}$ will yield this decrease. The basic variable is located by checking the ratios y_{i_o}/y_{ij} and taking the smallest positive values (y_{i_o} = value of $\mathbf{x_B}$ in row i, y_{ij} = coefficient in column j which has the negative cost coefficient).

The base change is executed by manipulating all elements of $\mathcal{T}$. The minimum of the objective function is found when all cost coefficients are positive (=optimum feasible basic solution).

A.1.2 Changeover from a non-feasible to a feasible solution

Problem statement In LP- and QP-problems there are situations or starting solutions which are not feasible, i.e. $\mathbf{x_B} < \mathbf{0}$. This means that the base point is outside the feasible region.

If a feasible region exists a feasible basic solution can be reached by one or several base change operations. The operations will depend on the specific problem. In the OPF- algorithms two kinds of problems are encountered, namely

- Problem a.: A constraint is added to the tableau which generates a negative slack variable when the current basis solution is inserted
- Problem b.: The basic solution is not feasible right from the beginning

Problem a. is faced in LP-based OPF methods, e.g. after completing a load flow or after segment refinement. The cost coefficients may be the same or may have changed also. A change of base is necessary. The question is how to perform the base change operation.

Problem b. is found in the QP-method which treats constraints by LP-steps, see appendix A.2. In this particular case base change operations are confined to the row with the negative base value and the column where i=j (diagonal).

Solution of problem a. For the explanation of the algorithm the LP-tableau is extended the following way:

$$\mathcal{T} = \begin{bmatrix} \mathbf{U} & \mathbf{0} & \mathbf{B}^{-1}\mathbf{D} & \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{d}^T & 1 & \mathbf{0}^T & b_A \\ \mathbf{0} & \mathbf{0} & \mathbf{c_B}^T - \mathbf{c_B}^T\mathbf{B}^{-1}\mathbf{D} & -\mathbf{c_B}\mathbf{B}^{-1}\mathbf{b} \end{bmatrix} \tag{140}$$

where $\mathbf{d}^T\mathbf{x_B} \geq b_A$ is the violated constraint. ($\mathbf{d}$ is a row vector, b_A is a scalar).

In a first step the elements of $\mathbf{d}^T$ are eliminated by adding rows appropriately scaled to the last row such that the elements of the row disappear (**LU** factorization). The result is a standard tableau with the only difference that the values of the last element of the base vector will be negative $y_{i_o} < 0$.

It is now obvious that the last basic variable has to leave the base and the non-basic variable showing the smallest positive value of y_{i_o}/y_{ij} has to enter the base. After the change of base the basic solution is feasible but not necessarily optimum. However, the subsequent base change operation is standard.

Solution of problem b. In this problem the tableau contains $\mathbf{B^{-1}D}$ and $\mathbf{B^{-1}b}$ only. There is no relative cost vector nor is there a cost function, see appendix A 2.

The objective of the base change operation is to achieve a feasible basic solution subject to the condition that the operation is pivoted around the diagonal of $\mathbf{B^{-1}D}$. This is a condition of the QP-algorithm.

The algorithm starts with one or more elements of $\mathbf{B^{-1}b}$ being negative. The pivot element is the diagonal element of this particular row. Hence the base change operation is straight forward. If there are further negative elements in the base the process is continued.

The process stops when all elements of the base vector are positive. There is just one solution to the problem (for a convex QP-problem).

A.2 Quadratic Programming

The classic objective function of a QP problem is as follows:

$$\mathcal{F} = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \quad \Rightarrow \quad min \tag{141}$$

subject to linearized equality and inequality constraints:

$$\begin{aligned} \mathbf{J}\mathbf{x} - \mathbf{b}_1 \quad &= \mathbf{0} \\ \mathbf{A}\mathbf{x} - \mathbf{b}_2 \quad &\leq \mathbf{0} \end{aligned} \tag{142}$$

The matrices $\mathbf{Q}, \mathbf{J}, \mathbf{A}$ are of general nature. Depending on the OPF QP-variable choice they can be either sparse, constant or also non-sparse.

In the following the QP will be transformed into an unconstrained QP optimization problem whose solution is trivial. In order to achieve the QP solution

with consideration of the inequality constraints a superposition is applied. The resulting optimization problem is a Linear Programming based optimization problem ([24], [25]). This derivation is briefly shown in the following.
The Lagrange function with consideration of equality constraints only and the corresponding optimality conditions are as follows:

$$\mathcal{L} = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T(\mathbf{J}\mathbf{x} - \mathbf{b}_1) \tag{143}$$

$$\begin{bmatrix} \mathbf{Q} & \mathbf{J}^T \\ \mathbf{J} & \mathbf{0} \end{bmatrix} \mathbf{u}_0 = \mathbf{M}\mathbf{u}_0 = \begin{bmatrix} -\mathbf{c} \\ \mathbf{b}_1 \end{bmatrix} \tag{144}$$

with

$$\mathbf{u}_0 = \begin{bmatrix} \mathbf{x}_0 \\ \boldsymbol{\lambda}_0 \end{bmatrix} \tag{145}$$

The Lagrangian for the problem with inequality constraints and its optimality solutions is as follows:

$$\mathcal{L} = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T(\mathbf{J}\mathbf{x} - \mathbf{b}_1) + \boldsymbol{\mu}^T(\mathbf{A}\mathbf{x} - \mathbf{b}_2) \tag{146}$$

$$\begin{aligned} \mathbf{M}\mathbf{u}_c + \begin{bmatrix} \mathbf{A}^T \\ 0 \end{bmatrix} \boldsymbol{\mu}_c &= \begin{bmatrix} -\mathbf{b}_0 \\ \mathbf{b}_1 \end{bmatrix} \\ \mathbf{A}\mathbf{x}_c &\leq \mathbf{b}_2 \\ \boldsymbol{\mu}_c &\geq \mathbf{0} \end{aligned} \tag{147}$$

The solution of this inequality constrained problem is now split into the equality constraint solution and a superposition:

$$\mathbf{u}_c = \mathbf{u}_0 + \Delta\mathbf{u} \tag{148}$$

It follows for the optimality conditions for the inequality constrainted OPF problem:

$$\begin{aligned} \mathbf{M}\Delta\mathbf{u} + \begin{bmatrix} \mathbf{A}^T \\ 0 \end{bmatrix} \boldsymbol{\mu}_c &= \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \\ \mathbf{A}(\mathbf{x}_0 + \Delta\mathbf{x}) &\leq \mathbf{b}_2 \end{aligned} \tag{149}$$

Since the vector $\mathbf{x}$ is a subvector of the vector $\mathbf{u}$ the inequality constraints can be rewritten. If substituting also the change of the variables $\mathbf{\Delta u}$ the following inequality constraint set results:

$$-\left[\mathbf{A},\mathbf{0}\right]\mathbf{M}^{-1}\begin{bmatrix}\mathbf{A}^T\\ 0\end{bmatrix}\boldsymbol{\mu}_c \leq \mathbf{b}_2 - \mathbf{A}\mathbf{x}_0 \tag{150}$$

This inequality constraint system corresponds conceptually to the following problem:

$$\mathbf{T}\boldsymbol{\mu}_c \leq \mathbf{b} \qquad \boldsymbol{\mu}_c \geq \mathbf{0}\ ,\ \ \mathbf{b} \geq \mathbf{0} \tag{151}$$

The problem is to find a vector $\boldsymbol{\mu}_c$ which satisfies the above inequality constraints. Conventional LP techniques can be applied to do this.

After having found the feasible point for the above inequality constraint problem the other (eliminated) variables can be found be replacing the values for $\boldsymbol{\mu}_c$ into the relevant equations:

$$\Delta\mathbf{u} = -\mathbf{M}^{-1}\begin{bmatrix}\mathbf{A}^T\\ 0\end{bmatrix}\boldsymbol{\mu}_c \tag{152}$$

Of course the inversion of the matrix $\mathbf{M}$ is not actually done in a computer implementation. A forward and backward solution is executed with the factors of the matrix $\mathbf{M}$.

As derived above the solution must be found for the following inequality constrained system:

$$\mathbf{T}\boldsymbol{\mu}_c \leq \mathbf{b} \tag{153}$$

This is in principle a classical LP problem. Several solution methods can be found in literature. In this appendix one possible solution is briefly discussed.

A vector of slack variables $\mathbf{x_B}$ is introduced. They can be seen as a set of base variables. $\mathbf{U}$ is a unity matrix.

$$\mathbf{T}\boldsymbol{\mu}_c + \mathbf{U}\,\mathbf{x_B} = \mathbf{b} \tag{154}$$

The base variables of non-satisfied inequality constraints are negative. In the optimum all variables of the LP problem must be positive. By choosing a negative pivot in the row a negative base variable it can be made positive. The principle is to make base changes such that all base variables are finally

positive. If all base variables are positive a feasible solution for the inequality is found.
In this special case of inequality consideration a special choice for the pivot is necessary: If an inequality constraint i becomes active, i.e. binding at its limit, the associated base variable $x_{B_i} = 0$ becomes zero. At the same time the associated variable $\mu_i \neq 0$, i.e. each equality constraint or binding inequality constraint must have an associated Lagrange multiplier with a value $\neq 0$. This means that for every set of associated variables (x_{B_i}, μ_i) one and only one of them must be exactly zero. This means that in the LP tableau of the inequality constraints the pivot for base changes can only be a diagonal element.

Without giving a proof in this paper, it can be shown that the solution for the problem, if it exists, is unique.

It can be also be shown that the actual implementation of this LP-optimization can be done with clever and fast updating techniques when the size of the inequality constraint set changes. However, due to space reasons this is not done in this paper.

A.3 Symbols

The following notations are used througout this text:

- Symbols representing complex variables are underlined.
- Matrices are shown in capital boldface letters.
- Vectors are shown in small boldface letters.

A.3.1 Symbols used in the power flow

The following symbols are used in the conventional Power Flow equations.
j: complex multiplier (for imaginary part of complex variable)
*: conjugate complex operator
^{k}: associated variable or expression is state (or iteration) dependent
$_{opt}$: associated variable is optimum variable
$Real$: Real part of following complex expression
$Imag$: Imaginary part of following complex expression
^{T}: Transposed - operator
$_{low}$: low limit of a variable
$_{high}$: upper (high) limit of a variable
$_{scheduled}$: related to variable with scheduled, predetermined value
Δ: change operator for variables, matrices, vectors

∂: derivative operator
N: total number of electrical nodes
m: total number of generator PV nodes
l: total number of load PQ nodes
EL: number of elements in loss objective summation function
$_{slack}$: slack node index
k_{slack}: constant slack node voltage ratio
P_i: active power at node i
Q_i: reactive power at node i
$P_{scheduled_{PQ}{}_i}$: scheduled active power at PQ node i
$Q_{scheduled_{PQ}{}_i}$: scheduled reactive power at PQ node i
$\underline{\mathbf{V}}$: vector of complex voltages
$\underline{V}_i$: complex voltage at node i
V_i: voltage magnitude at node i
e_i: real part of $\underline{V}_i$
f_i: imaginary part of $\underline{V}_i$
Θ_i: voltage angle at bus i : $arctan\frac{f_i}{e_i}$
e_{slack}: real part of $\underline{V}_i$, i: slack node
f_{slack}: imaginary part of $\underline{V}_i$, i: slack node
$V_{scheduled_{PV}{}_i}$: scheduled voltage magnitude at PV node i
$\underline{\mathbf{I}}$: vector of complex currents
$\underline{I}_i$: complex current at node i
I_{e_i}: real part of $\underline{I}_i$
I_{f_i}: imaginary part of $\underline{I}_i$
P_{ij}: active power flow in the branch from node i to node j
Q_{ij}: reactive power flow in the branch from node i to node j
$P_{high_{ij}}$: upper MW flow limit in the branch from node i to node j
$S_{high_{ij}}$: upper MVA flow limit in the branch from node i to node j
Q_{ij}: reactive power flow in the branch from node i to node j
$\underline{\mathbf{Y}}$: complex nodal admittance matrix
$\underline{Y}_{ij}$: complex element of $\underline{\mathbf{Y}}$-matrix at row i and column j
y_{ij}: absolute value of $\underline{Y}_{ij}$
g_{ij}: real part of $\underline{Y}_{ij}$
b_{ij}: imaginary part of $\underline{Y}_{ij}$
G_{ij}: real part of admittance of a π - element between nodes i and j
B_{ij}: imaginary part of admittance of a π - element between nodes i and j
θ_{ij}: angle of admittance $g_{ij} + jb_{ij}$: $arctan\frac{b_{ij}}{g_{ij}}$
B_{i_o}: charging/2 (purely capacitive) of line from i to j measured at node i
t_{ij}: tap of transformer between nodes i and j

A.3.2 Symbols used in optimal power flow optimization algorithm

The following symbols are used only in connection with the OPF.

^{k}: index referring to state and iteration dependent matrices, vectors
diag: representing a diagonal matrix
U: identity (unity) matrix
$\mathcal{X}$: vector of control variables
$\mathcal{U}$: vector of state variables
$\mathcal{P}$: vector of demand variables
x_i: OPF variable i
x: vector of OPF variables
$\mathbf{x_i}$: subset i (i = 1 ... 4) of vector **x**
$\mathbf{x_{3j}}$: subset j (j = 1 or 2) of vector $\mathbf{x_3}$
$\mathcal{F}$: objective function
$\mathcal{F}_{cost}$: total cost objective function
$\mathcal{F}_{cost_i}$: cost function of generator i
$\mathcal{F}_{loss}$: total loss objective function
$\mathcal{F}_{loss_i}$: losses related to branch i
g: set of OPF equality constraints
$\mathbf{g_i}$: subset i (i = 1 ... 4) of OPF equality constraints **g**
h: set of inequality constraints
λ_i: Lagrange function multiplier for equality constraint i
μ_i: Lagrange function multiplier for inequality constraint i
$\boldsymbol{\lambda}$: vector of all λ_i
$\boldsymbol{\mu}$: vector of all μ_i
$\mathcal{L}$: Lagrange function
H: Hessian matrix
Q: quadratic cost coefficient matrix of quadratic objective function
$\mathbf{Q_{Loss}}$: quadratic loss coefficient matrix of quadratic loss objective function
q_i: quadratic cost coefficient of variable active generator power i
c: vector of linear cost coefficients of objective function
A: sensitivity matrix for inequality constraints in linearized form
$\mathbf{A_i}$: submatrix i (i = 1 ... 4) of **A**
M: matrix representing second derivatives of the power flow equations
$\mathbf{M}_{ij}$: submatrix of **M**
$\mathbf{A_i}$: submatrix i (i = 1 ... 4) of **A**
$\mathbf{b_1}$: right hand side values of linearized equality constraints
$\mathbf{b_2}$: right hand side limit values of linearized inequality constraints
J: Jacobian matrix (first derivatives of power flow equations)

$\mathbf{J}_{ij}$: submatrix of $\mathbf{J}$
$\boldsymbol{\alpha}$: vector of linear incremental power flow equality constraint
$\boldsymbol{\alpha}_i$: subvector i (i = 1 or 2) of $\boldsymbol{\alpha}$
$\mathbf{W}$: non-linear system representing optimality conditions (OPF class B)
$\mathbf{H}$: matrix representing second derivatives of the Lagrangian
$_{eq}$: index associated with a variable of the equality constrained OPF æ

Literatur

[1] H.W.Kuhn, A.W.Tucker; *Non-linear programming, Proc. 2nd Berkeley Symposium on Mathematics, Statistics and Probability*, University of California Press, Berkeley, California (1951)

[2] L.K.Kirchmayer; *Economic Operation of Power Systems*, Wiley, New York (1958)

[3] L.K.Kirchmayer; *Economic Control of Interconnected Systems*; Wiley, New York (1959)

[4] R.B.Squires; *Economic Dispatch of Generation Directly from Power System Voltages and Admittances*, AIEE Trans. PAS Vol. 52. Part III (1961),pp. 1235-1244

[5] J.Carpentier; *Application of Newton's Method to Load Flow Computations*, Proc. PSCC 1, London (1963)

[6] W.F. Tinney, J.W. Walker; *Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization* Proceedings of the IEEE, Vol. 55, No. 11, Nov. 1967

[7] H.W. Dommel, W.F. Tinney; *Optimal Power Flow Solutions;* IEEE Transactions on Power Apparatus and Systems, Vol. PAS-87, pp. 1866-1867, Oct. 1968

[8] D.G. Luenberger; *Introduction to Linear and Nonlinear Programming;* Addison-Wesley Publishing Company, Reading, Massachusetts, 1973

[9] J.Carpentier; *Differential Injection Method, a General Method for Secure and Optimal Load Flows*, Proc. PICA (1973), pp. 255-262

[10] H.H.Happ; *Optimal Power Dispatch*, IEEE Trans. PAS, Vol. 93. (1974), pp. 820-830

[11] O.Alsac, B.Stott; *Optimal Power Flow with Steady-State Security*, IEEE Trans. PAS, Vol. 93. (1974), pp. 745-751

[12] B. Stott, O. Alsac; *Fast Decoupled Load Flow;* IEEE Transactions on Power Apparatus and Systems, Vol. PAS 93, pp. 859-869, May/June 1974

[13] J. Carpentier; *System Security in the Differential Injection Method for Optimal Power Flows*, Proc. PSCC 5, (1975)

[14] H.H.Happ; *Optimal Power Dispatch. A Comprehensive Survey*, IEEE Trans. PAS, Vol. 96. (1977), pp. 841-853

[15] B.Stott, E.Hobson; *Power System Security Control Using Linear Programming*, Part I+II, IEEE Trans. PAS, Vol. 97. (1978), No. 5

[16] B. Stott, J.L. Marinho, O.Alsac; *Review of Linear Programming Applied to Power System Rescheduling;* IEEE PICA Conf. Proc., pp. 142-154, Cleveland, OH, May 1979

[17] J. Carpentier; *Optimal Power Flows;* Electrical Power & Energy Systems, Butterworths; Vol 1 No.1, April 1979

[18] M. Spoerry, H. Glavitsch; *Quadratic Loss Formula for Reactive Dispatch;* IEEE PICA Proceedings, 17-20 May 1983 Houston USA

[19] D.I. Sun, B. Ashley, B. Brewer, A. Hughes, W.F. Tinney; *Optimal Power Flow by Newton Method;* IEEE Transactions on Power Apparatus and Systems, Vol. PAS 103, No. 10, pp. 2864-2880, Oct. 1984

[20] R.C. Burchett, H.H. Happ, D.R. Vierath; *Quadratically Convergent Optimal Power Flow;* IEEE Trans. PAS-103, No.11, 1984, pp. 3267-3275

[21] W.F. Tinney, V.Brandwajn, S.M. Chan; *Sparse Vector Methods* IEEE Transactions on Power Apparatus and Systems, Vol. PAS-104, No.2 pp. 295-301, February, 1985

[22] B. Stott, O. Alsac, A. Monticelli; *Security and Optimization;* Proceedings of the IEEE, Vol. 75, No. 12, Dec. 1987

[23] F.F. Wu; *Real-time Network Security Monitoring, Assessment and Optimization;* Electrical Power & Energy Systems, Butterworths; Vol 10 No.2, April 1988

[24] W. Hollenstein, H. Glavitsch; *Constraints in Quadratic Programming Treated by Switching Concepts;* Proceedings of the 10th Power Systems Computation Conference PSCC, Graz, Austria, August 19-24, 1990

[25] W. Hollenstein, H. Glavitsch; *Linear Programming as a Tool for Treating Constraints in a Newton OPF;* Proceedings of the 10th Power Systems Computation Conference PSCC, Graz, Austria, August 19-24, 1990

[26] O. Alsac, J. Bright, M. Prais, B. Stott; *Further Developments in LP-Based Optimal Power Flow;* IEEE Transactions on Power Systems, Aug. 1990

Inhaltsverzeichnis