JOHANNES KEPLER
UNIVERSITÄT LINZ
JKU

TNF
Technisch-Naturwissenschaftliche
Fakultät

Simulation-Based Evolutionary Dynamic and Stochastic Optimization for Smart Electric Power Systems

DISSERTATION

zur Erlangung des akademischen Grades

Doktor

im Doktoratsstudium der

Technischen Wissenschaften

Eingereicht von:
Dipl.-Ing.(FH) Stephan Hutterer

Angefertigt am:
Institut für Formale Modelle und Verifikation

Beurteilung:
FH-Prof. Priv.-Doz. Dipl.-Ing. Dr. Michael Affenzeller (Betreuung)
a.Univ.-Prof. Dipl.-Ing. Dr. Günther Blaschek

Linz, November, 2013

Acknowledgements

First and foremost I want to thank my supervisor Michael Affenzeller. He strongly supported my intended research work from the early beginnings and mainly contributed to all ideas that finally triggered the preparation of my dissertation thesis. Michael was certainly the greatest supporter of all my scientific achievements. Besides being my doctoral supervisor as well as head of the research group, I am happy to have the pleasure to call him a good friend.

Thanks to Franz Auinger, I decided to take the chance of working in the research area. After supervising my diploma thesis he motivated me to work towards a dissertation. He offered the position as a research associate to me and supported me in generating fruitful research ideas at the very beginning of my work. He is an outstanding person and still one of the most important characters and supporters of my scientific career.

Another important factor for the success of my thesis was the supervision of Günther Blaschek. Gü accepted to take the role as second advisor and supported me with extensive discussions and important suggestions when finishing the thesis. Even if starting at a very late state of my doctoral studies, he was very dedicated to discussing my research ideas and supplying me with valuable advices.

Without the support of the research group HEAL as well as the excellent software framework HeuristicLab, many parts of my research work would not have been possible. All HEAL members fundamentally contributed to my work's outcomes through many valuable discussions and intellectual support. Especially Stefan Vonolfen and Andreas Beham have always been helpful by aiding me with their extensive knowledge and experience when doing implementation work in HeuristicLab.

Finally, I want to thank my family - especially my parents. They basically encouraged me in doing technical studies which was finally the basis of all my following achievements. Additionally my two older brothers Michael and Christian need to be mentioned, they helped my a lot when proof-reading this thesis. Moreover, my girlfriend Silvia has always been a strong backing during all the years and also during the final preparation phase of this work.

Kurzfassung

Optimierung stellt einen wichtigen Bestandteil in Betrieb und Planung moderner Stromnetze dar. Zeitgleich trägt der aktuelle Wandel der Versorgungsparadigmen elektrischer Energie - hin zu so genannten Smart Grids - neue Anforderungen an Optimierungsanwendungen mit sich: Optimale Entscheidungen müssen zunehmend in dynamischen und unsicheren Systemen getroffen werden. Gleichzeitig verlangt die Steuerung einer Vielzahl verteilter Endgeräte nach Skalierbarkeit der eingesetzten Methoden.
In diesem Kontext gewinnen heuristische Optimierungsverfahren an Bedeutung. Speziell mit simulationsbasierter Optimierung können Probleme in komplexen und unsicheren Szenarien bearbeitet werden, wodurch das methodische Fundament dieser Arbeit begründet wird. Mittels Analyse aktueller und zukünftiger technologischer Herausforderungen in Stromnetzen werden sowohl statische als auch dynamische Optimierungsprobleme formuliert, behandelt und analysiert. Darauf aufbauend wird ein Ansatz der simulationsbasierten evolutionären Approximation von so genannten *policies* entwickelt, welcher sowohl generisch beschrieben, aber speziell für dynamische und stochastische Lastflussprobleme entwickelt und validiert wird. Speziell diese Verquickung von evolutionärer Simulationsoptimierung und der policybasierten dynamischen Optimierung für reellwertige Entscheidungsprobleme bietet einen neuartigen methodischen Ansatz.
Zur Bewertung entwickelter Methoden werden statische Benchmark-Szenarien optimaler Lastflussprobleme um stochastische wie auch dynamische Elemente erweitert, um realistische Testinstanzen darzustellen. Hier werden zentrale Eigenschaften der entwickelten Optimierungsmethoden erprobt, nämlich einerseits die Fähigkeit der Errechnung robuster nah-optimaler Entscheidungen in unsicheren und dynamischen Systemen, andererseits aber auch die Skalierbarkeit in Hinblick auf zukünftige Steuerung einer Vielzahl verteilter Geräte.
Der zweite experimentelle Teil spezialisiert sich auf Herausforderungen in zukünftigen Smart Grids, hier speziell auf die optimale Steuerung verteilter Lasten. Da die technische Realisierung dieses "Demand Side Management" in Zukunft vielgestaltige Ausprägungen annehmen kann, wird eine generische Problemformulierung vorgenommen. Für experimentelle Zwecke wird darauf aufbauend ein konkretes Szenario mit steuerbaren Elektrofahrzeugen definiert, welches unter Rücksichtnahme dezentraler erneuerbarer Einspeiser ein illustratives Problem für zukünftige Laststeuerung darstellt. Aufgrund der Notwendigkeit der Steuerung unter unsicheren und volatilen Bedingungen für eine hohe Anzahl an Elektrofahrzeugen wird hier einmal mehr die simulationsbasierte evolutionäre Approximation von *policies* angewandt und validiert.

Abstract

The electric power systems research society early identified the necessity of optimization both for planning and operation tasks, where formulations such as the optimal power flow (OPF) problem shape this research domain ever since. At the same time, technological changes to electric power grids challenge new methods, requiring optimization in both dynamic as well as uncertain systems. Additionally, optimization tasks for the control of numerous distributed devices fundamentally require scalability aspects.
In this context, heuristic optimization methods have evolved as being capable of managing many of those upcoming needs. Simulation optimization with metaheuristics provides a promising fundament for optimization under uncertainty, and offers the basic approach for handling manifold challenging optimization issues within this work. While various aspects of future power grid optimization tasks are being analyzed, simulation-based methods both for static but mainly for dynamic problems are developed. Further on, simulation-based evolutionary policy function approximation is being discussed for dynamic power flow control problems, which is presented both in a generic manner as well as tailored to the electric engineering domain.
In a first experimental part, the developed methods are applied both to a static probabilistic planning problem and to a dynamic OPF control task within benchmark systems. Showing that approximate optimal policy-based control yields competitive results compared to reference solutions, it additionally is able to make quick and robust control actions within dynamic and stochastic environments, being scalable to numerous devices.
The second experimental part treats optimal load control issues in the smart electric grids context. Electric vehicle (EV) charging control is defined as a generic problem for optimal load control over time. Existing works in the literature are being analyzed while major lacks can be identified that need to be tackled. Here, once more simulation-based evolutionary policy function approximation comes into play, which is shown to evolve control policies for a holistic electric vehicle charging problem that unifies all important requirements for smart EV charging. Finally, policies are evolved that enable intelligent charging decisions to multiple EVs within an experimental system. These decisions are even able to satisfy system-wide goals while being flexible to dynamic and uncertain conditions. Comparisons to deterministically optimized charging decisions in a test simulation show its validity.
Finally, a scalable technology has been presented, which enables approximate optimal control in volatile as well as uncertain systems. Representing a technology for managing dynamic stochastic optimal power flow problems makes it highly promising for future smart electric grid control.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die vorliegende Dissertation ist mit dem elektronisch übermittelten Textdokument identisch.

Contents

List of Abbreviations and Definition of Terms

This table provides various significant abbreviations and acronyms used throughout the thesis. The page of first definition or first usage is also given. Some acronyms that are used only locally in the thesis are not contained by the list.

Abbreviation	**Meaning**	**Page**
kV	Kilovolts	11
OPF	Optimal Power Flow	12
EV	Electric Vehicle	13
GA	Genetic Algorithm	15
DSOPF	Dynamic Stochastic OPF	19
MDP	Markov Decision Process	23
ANN	Artificial Neural Network	24
OSGA	Offspring Selection Genetic Algorithm	27
ADP	Approximate Dynamic Programming	37
SVS	System Variable Synthesis	42
ARS	Abstract Rule Synthesis	47
GP	Genetic Programming	50
MW	Megawatts	67
ES	Evolution Strategies	61
V2G	Vehicle-to-Grid	100
SOC	State-Of-Charge	103
PV	Photovoltaic	119
MVAr	Mega Volt Ampere Reactive	126
IGA	Island Genetic Algorithm	142

Several terms that are of general importance to the thesis need to be defined for ensuring better understanding of argumentations.

Term	Definition
Static Optimization	Aims at providing one inflexible (parametric) solution to a discrete state of the system.
Dynamic Optimization	The optimization problem is solved in order to provide flexible solutions to manifold dynamic system states, rather than providing one (parametric) solution to one discrete state.
Stochastic Optimization Problem	Denotes that the optimization problem is governed by random variables, where this definition is independent from the applied optimization method. Stochastic problems may be tackled both by stochastic as well as deterministic optimization methods.
Deterministic Optimization Problem	Also addresses the type of problem rather than the solution method, i.e. the optimization problem is given through a deterministic model. Contrary to stochastic models, it yields constant outputs for same inputs.
Synthesis	The process of combining complex decision policies out of single information entities (such as system variables or abstract rules).
Optimal Power Flow	Original optimization problem formulation: aims at finding an optimal configuration of supply units in a system in order to meet economic satisfaction of power demand while incorporating power flow constraints [116]. A more general definition within this work is to control power flows in a grid such to meet some defined objective while satisfying power flow security constraints.
Dynamic Stochastic OPF	Optimal power flow in a dynamic as well as uncertain environment (see dynamic optimization).
Abstract Rule Synthesis	Combine abstract rules to a more complex decision/control- policy.
System Variable Synthesis	Combine system variables from the model to a complex decision/control- policy.

List of Symbols

For the most important (and frequently used) symbols within this thesis, a listing is provided, also giving the page of the first definition or first usage.

Symbol	Usage	Page
t	Index of time step	19
$J(t)$	System state at time t	19
u	Independent variable: control variable / decision / action	25
$f'(u)$	The apostrophe is generally used to denote a simulated (estimated) quantity with respect to the real-world's quantity	25
ϵ	Noise	25
s	Index of sample	27
S	Number of samples (drawn from the simulation)	27
x	Dependent state-variable	29
$C(...)$	Generic denotation of a cost-function	29
P_G	Generator's real-power injection	30
P_L	Real-power load	30
P_B	Real-power branch flow	30
V_G	Voltage at generator-bus	30
V_L	Voltage at load-bus	30
Q_G	Generator's reactive-power injection	30
Q_L	Reactive-power load	30
Q_C	Reactive-power injection of VAR compensator	30
T_{tap}	Transformer tap setting	30
NL	Number of load buses	30
NG	Number of generator buses	30
NB	Number of branches	30
NT	Number of transformers	30
NC	Number of VAR compensators	30

Continued on next page

Symbol	Usage	Page
a_g, b_g, c_g	Polynomial cost coefficients of generation units	30
j	Index of bus (general)	31
g	Index of generation unit	31
b	Index of branch/line	31
${P_G}^{max}$	Maximum real-power generation capacity	30
${P_G}^{min}$	Minimum real-power generation	30
${Q_G}^{max}$	Maximum reactive-power generation capacity	30
${Q_G}^{min}$	Minimum reactive-power generation	30
V_j^{max}	Maximum voltage at bus	31
V_j^{min}	Minimum voltage at bus	31
P_b^{max}	Maximum branch power flow	31
p	Generic denotation of a policy	39
$\mathbb{P}$	Space of potential policies	39
$\bar{i}$	Vector of input variables for decision making	41
$\overline{r}$	Vector of abstract rules	45
$\overline{w}$	Vector of abstract rules' weights	49
NR	Number of rules	49
nr	Index of rule	49
n	Index of controllable device	49
c	Constant	49
$E[]$	General denotation of an estimate	55
$CV_t(...)$	Constraint violation at time step t	60
$\overline{w_{CV}}$	Weights of constraints relative to objective function	60
$L_t(\overline{p_{PL}})$	System loss function, depending on placement decision	66
$\overline{p_{PL}}$	Placement decision	66
Δt	Length of time step	73
rc_x	Real-valued constant	86
x_s	Dependent variables for smart grid OPF formulation	97
u_s	Independent variables for smart grid OPF formulation	97
$P_{L,C_{NLC}}$	Reactive-power value of controllable load NLC	97
$P_{EV\,t,n}$	Charging power of EV n at time step t	103
$p_{MAX\,t}$	Global power constraint at time step t	103
cf	Discrete-valued cost function	103
$cr_{MAX\,t,n}$	Maximum charging rate of EV n at time step t	103
$E_{MIN\,n}$	Minimum charged energy of EV n	103
$p_{EV}(...)$	Policy for EV-charging decision	124

Part I

Introduction

Chapter 1

Synopsis

1.1 Motivation

Electric power grids are operated all over the world, representing complex technical systems while both appropriate planning and operation are of high economic impact. Here, the constitution of these systems most often corresponds to the traditional paradigm, that energy needs to be transported from central large-scale supply plants to numerous spatially distributed customers. While - due to the law of power balance - the load (customers' demand) and supply (power generation) need to be equal at any given point in time, in conventional operation the supply plants' generation is adapted continuously according to the non-controllable system load. This principle which characterizes power grid operation ever since is entitled by *load-dependent generation*.

Since the economic impact of decisions both in planning as well as operation of power grids is high, optimization has established as essential tool in power grid engineering. Here, fundamental formulations such as the *optimal power flow* (OPF) provide a generic framework which is applied to manifold applications in order to derive optimal planning and operation decisions in power grids. While the traditional OPF aims at optimally controlling a set of generation units in order to meet the demand at lowest financial costs, in the meanwhile it has been adapted to diverse decision problems in power grid engineering, such as different kinds of control units, objective functions, and constraints.

While the general principles of both construction and operation of electric power grids seemed to be invariant along many decades, in recent years a paradigm shift characterizes the electric power industry, introducing principles of so called *smart electric grids* as enabler for more efficient, reliable and sustainable electric power generation and distribution. This paradigm-shift not only raises diverse technical challenges, it also asks for new problem formulations as well as methods for power grid optimization. Especially when regarding optimal power flows and corresponding control tasks, central changes that come along with smart electric grids are a decentralization of the distribution grid operation on the one hand (i.e. small and distributed devices for supply, storage as well as controllable demand), and an inversion of the traditional *load-dependent generation scheme* on the other hand. While in traditional operation the controllable generation units are scheduled in order to meet uncontrollable demand, controllable load devices shall be operated for enabling efficient usage of available generation resources (i.e. uncontrollable supply from renewables) in smart power grids. From an optimization point of view, these diverse changes can be summarized in three important aspects:

- Increasing Quantity of Control Variables
  Due to the change to *generation-dependent load* as well as the implementation of numerous distributed controllable devices, the quantity of control variables for power flow actions increases significantly, affording scalable computational methods that are capable of handling high amounts of variables.

- Higher Volatility
  On the one hand, control will be performed on lower power grid levels, where dynamics of single decisions are higher than on upright levels. On the other hand, the progressive penetration of fluctuating renewable small-scale power plants further increases the volatility of power grid behavior on these lower levels, necessitating control methods that provide power flow decisions quickly.

- Need for Incorporation of Uncertainty
  Similar reasons hold when considering uncertainty. Both the supply-side as well as the demand-side within the power grid cause increasing probabilistic influences, which need to be integrated appropriately for making optimal decisions in smart grids. This is an important issue in order to provide robust decisions under uncertainty that guarantee secure as well as reliable operation.

Hence, new formulations of existing optimization problems need to be conducted that are able to include all these aspects on the one hand. On the other hand, innovative optimization methods need to be investigated being capable of treating these problems for still deriving optimal decisions in operation (dynamic optimization) and planning (static optimization) of future smart electric grids.

1.2 Research Questions

Based on these new requirements to electric power grid optimization, central research questions can be formulated that are treated within this work:

- Is it possible to derive new formulations of the optimal power flow (OPF) problem that meet future requirements in order to still provide a generic framework for electric power grid optimization?
- How can increasing uncertainties be considered for deriving robust decisions in both planning as well as operational aspects?
- Which methods will be needed in order to provide robust actions for dynamic and stochastic optimal power flow control?
- Is it possible to develop such methods for complex real-world power systems with high amounts of control variables?

1.3 Overview of the Thesis

Stating the fundamentals of electric power grid operation as well as smart grid principles and challenges in Sections 2.1 and 2.2 respectively, Chapter 3 proceeds with a discussion on simulation-based optimization being a central enabler for optimization in stochastic systems. While a clear distinction between static and dynamic optimization will be defined, the latter is capable of providing optimal control abilities over time using evolutionary simulation-based approximation of flexible control policies, being developed in Chapter 4. After discussing the fundamentals of this new technology with special respect to the power grid operation domain, the reader will be supplied with experimental validations when applying it to selected OPF benchmark problems in Chapter 5. Having shown its capabilities, Chapter 6 develops essential smart grid scenarios together with challenges that come up with them, that will finally be solved using the investigated technology of evolutionary policy optimization, also validating this technique. Here, related achievements in literature will be discussed as well as the existing lacks that finally get diminished. After applying the discussed policy optimization techniques to an electric vehicle charging control scenario which comprises typical smart grid features, Chapter 7 provides concluding remarks and an outlook to potential future research issues.

1.4 Main Research Contribution

The scientific contribution when handling the defined research questions is twofold: on the one hand, new methods are introduced for enabling dynamic and stochastic power flow control, namely simulation-based evolutionary policy approximation. On the other hand, future smart grid challenges are analyzed with respect to optimization, leading to an extension of the optimal power flow problem that is valid for manifold emerging technologies. Applying the developed policy approximation technology to a generic smart grids scenario completes the research work while both validating the investigated methods as well as handling real-world research issues and applications.

1.4.1 Dynamic and Stochastic Optimal Power Flow

Providing appropriate techniques for dynamic and stochastic optimal power flow control is a fundamental challenge. Simulation optimization is demonstrated as being a promising principle that allows the integration of both very complex systems as well as their uncertainties into the optimization process. As this is an important achievement for both dynamic as well as static issues, the latter will be demonstrated for a probabilistic planning scenario.

Especially in the field of dynamic optimization, there are many open issues in the scientific literature. For the application of power flow control it is shown that a technology is necessary which allows scalability for controlling huge amounts of devices on the one hand, but which additionally is able to deliver fast and robust actions within complex and uncertain systems on the other hand. Therefore, simulation optimization gets extended in order to evolutionary approximate policy functions for dynamic control. Such policies are analytical functions that return an approximate optimal action given a state, without doing optimization at runtime. Simulation-based evolutionary policy approximation shows to be scalable while being suitable for large-scale systems (both in means of input as well as control variables). Additionally, from the point of view of policy approximation, a central requirement states the necessity of evolving policies of arbitrary mathematical structure without the need of a-priori knowledge of this structure. Being an important enabler of policy-based optimization in complex real-world systems, the policy evolution with genetic programming satisfies this requirement and is elaborated within this work. Different approaches on how to evolve policies out of given inputs get discussed, while a central achievement is the derivation of policy functions out of abstract rules rather than directly out of systems' state variables. This principle has the advantage that information from many hundreds or even more state variables within large systems can be compressed into policies on the one hand, but delivers individual actions to each of many distributed devices on the other hand. Additionally, these abstract rules provide generic information entities that do not need to be tailored to a specific problem instance (such as a specific power grid model), rather provide a generic information representation for policy evolution. Being combined with genetic programming enabled metamodel-less policy approximation, a general methodology for dynamic and stochastic optimal power flow control can be stated.

1.4.2 Smart Grid Optimization Challenges and Electric Vehicle Charging Control

Many future optimization issues in the power grid area demand multi-period considerations, where optimal decisions have to be made over time while satisfying constraints robustly under stochastic conditions.
Such multi-period considerations are often necessary for control issues. Especially in future smart electric grids, the control of huge amounts of distributed devices is a crucial task for enabling this technological change.
Within this work, it is shown that the control for various future technologies can be realized through an extension of the OPF problem. Here, an electric vehicle charging scenario is defined that generically represents future load-control problems, while being a hot-spot in actual smart grids research. Lacks of existing research works are identified, requiring a technology which is capable of integrating both the demand-side's as well as the supply side's uncertainties with high detail. The ability of deriving robust decisions with respect to actually resulting power flows in the system is an additional requirement in order to guarantee reliable power grid operation during control. A holistic optimization problem is specified that closes existing lacks in literature and is finally treated using the investigated technique of simulation-based evolutionary policy approximation, further validating this technology.

Chapter 2

Theory I: Electric Power Grids

2.1 Overview: Optimization and Control in Power Grids

This work discusses approaches for the optimization of stochastic as well as dynamic power flows, building an essential technology in future smart electric grids. In order to understand the treated issues of optimization, the reader shall be provided with fundamentals of electric power grid operation and control. Thus, a short introduction to necessary principles from electric- and energy-engineering will be discussed on a very general level before taking a more detailed view on optimization issues.

2.1.1 Electric Power Grid Operation - An Outline

The traditional operation of electric power systems - considered at an abstract level - can be simplistically entitled by the term *load-dependent generation*. In an electric grid, the demand caused by (domestic, commercial and industrial) customers effects an electric load that can typically not be influenced. Since obviously the power-balance has to be maintained within a system (i.e. supply and demand have to be equal at any point in time), the generation has to be adapted continuously according to the demand, which is performed by the (distribution/transmission) system operators. This term *load-dependent generation* is very important for later considerations when talking about so called smart grids.

Generation Unit Scheduling

The demand within a system follows daily load profiles, which vary depending on season and weekday. These profiles based on measurements and statistical investigations are sufficiently exact so that an anticipatory scheduling of generation units is enabled at a long time horizon (hours to days). Since real-time demand is influenced by individual behavior of customers and thus is partly uncertain, control-methods are applied in order to adapt the scheduled generation to the real-time demand continuously within small time units (less than seconds to some minutes). Depending on the time horizon, different control schemes are applied, classified into primary, secondary and tertiary control. The principle of such a load-dependent generation scheme is depicted in Figure 2.1, where a typical daily load profile (dotted line) is given being obtained from statistical investigations, serving as sufficient demand forecast for power grid operation. Based on such forecasts, generation units are scheduled in order to meet the demand at lowest possible costs. However, since real-time demand varies from the predicted one to a certain degree (such an uncertainty of e.g. ±4% is illustrated by the blue solid lines), generation has to be adapted online within short time ranges using the mentioned control schemes[1]. Thus, generation units try to "follow" the predicted load, which is commonly entitled as *load-dependent generation*.

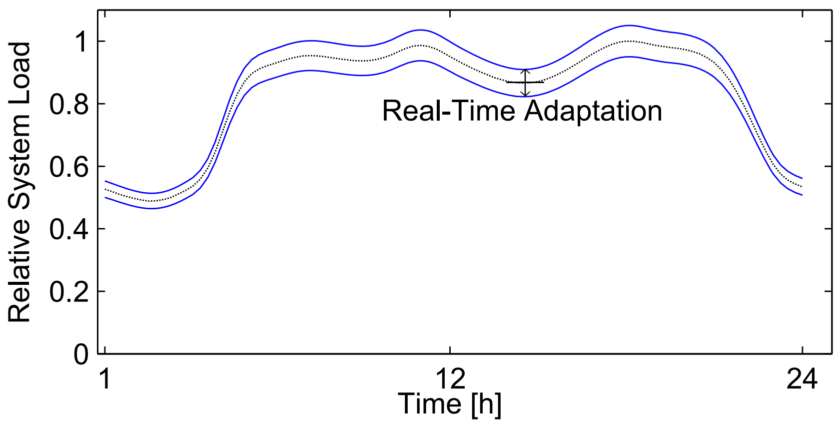

Figure 2.1: Principle of Load Dependent Generation

The scheduling of generation units is not only necessary to satisfy the electric demand, but to guarantee this satisfaction at lowest possible financial

[1]At this point, the differentiation between active- and reactive-power control is being neglected for simplicity reasons. In fact, this differentiation is indeed fundamental to power grid operation.

costs. Note that the economic impact of such scheduling is high, considering that 21% [25] of the total energy consumption in the European Union is caused by electric energy. To give an estimate on this scheduling issue: in central europe the relative difference between the lowest demand in summer and the highest demand in winter is 100%. Furthermore, considering the electricity spot market in Europe[2], the price of electric energy injected to power grids may vary by 300 - 700% during a day. Thus, accurate power grid control is not only of technical, but also of economic interest.

Modeling Electric Power Grids

Power grids are operated at different (voltage-) levels, ranging from low voltage grids for supplying small customers over distribution- to transmission grids which finally perform the bulk power transmission throughout regions of different scale (ranging from regional to international connections), where the voltage level increases in the same order from 230/400 Volts (V) to many 10 or even many 100 Kilovolts (kV). In traditional power grid operation, mainly the upper levels - where most generation units are connected to - are of interest for scheduling issues. While the smart grid developments introduce distributed controllable small-scale devices at lower levels, these will become of higher interest within this work as well.
The mathematical models as well as computational methods for power grid simulation are similar along the different levels, while a strong differentiation has to be mentioned concerning the considered time domain: namely between steady-state and transient models.

Steady-State vs. Transient Representations

For simplicity reasons, at this point we consider generation units to be the only controllable devices in power grids, where continuous control of voltage magnitude, real power injection and reactive power production takes place (sure, especially in actual technologies manifold controllable devices are available, but these three quantities are considered to remain the same). By measuring voltage as well as frequency[3] values of the power grid, error signals are derived (for turbine exciter as well as governor systems in the case of generation units) as input for respective control mechanisms. However, response times of such control schemes are quite low ($0.01 - 0.1s$ for

[2] European Power Exchange, http://www.epexspot.com
[3] While the frequency measurement is related to the real-power balance of the system, the voltage-value provides a measurement on the reactive-power balance.

exciters, $0.1 - 1s$ for governors). Since load variations are slow compared to these response times, it is valid to assume that perfect (voltage and frequency) control is always present as long as the power grid is within secure operation state. Hence, usual power grid operation is to a high degree sinusoidal steady-state operation. Only during system faults or immediately after switching operations, transient computations are needed [73]. Summing up, steady-state computation is sufficient for considering power flow decisions respectively dispatching actions that happen at a time scale of minutes or even longer (which is an important point for later discussions), no matter if these decisions address generation units, controllable loads, voltage regulation devices or even other equipment.

The complete theory on power grid operation and control would be too extensive to be discussed at this point, filling hundreds of practical as well as scientific books and other literature since many decades. The provided information is given at an abstract level, supplying the reader with sufficient knowledge for understanding the herein considerations. For further reading, appropriate literature is recommended [73, 96, 116]. More detailed discussions on specific aspects will be provided when needed herein.

2.1.2 Optimization - A Central Tool for Power Grid Operators

Survey on Power Grid Optimization

Optimization plays an essential role in power grid engineering, offering a wide range of different practical applications in for example power flow studies, maintenance scheduling or infrastructure planning. Here, numerous findings in systems engineering and analysis as well as actual achievements in computer sciences enabled an established usage of optimization in todays operation centers.
To give some brief overview on established optimization issues: Back in the early 1960s, the economic dispatch has been stated as the basic optimization problem in power system engineering, which tries to find an optimal configuration of supply units in a system in order to meet economic satisfaction of power demand [116]. Extending this formulation with additional constraints from a physical transmission/distribution grid point of view, the optimal power flow problem (OPF) aspires to meet optimality while incorporating power flow restrictions [73, 116]. Various OPF-formulations exist for realizing different aspects of power grid security at the constraints side, while numerous investigations have been performed using varying objective func-

tions like the minimization of fuel costs, active power losses or environmental impacts [2, 73]. Including additional aspects from generation plant operation, unit commitment can be stated as third basic problem. It incorporates OPF for finding an optimal commitment of available units, considering dynamic characteristics like up- and down ramps of possible generation capacity. Based on these essential formulations, further optimization applications have been identified within the last decades like infrastructure planning issues or computation of optimal maintenance schedules of supply, transmission or distribution equipment [73].
When talking about actual developments within this research field, mainly two areas need to be mentioned: namely the deployment of novel optimization applications on the one hand, which actually gets pushed by the smart grid vision. On the other hand, the evolution of recent solution methods such as metaheuristic algorithms, which is mainly supported by the steady growth of available computational resources as well as parallelization abilities, needs to be considered at this point.

Recent Practical Developments

Regarding the progress to novel applications, various trends can be identified: Venayagamoorthy [106] and Werbos [114] provide a sophisticated overview of computational intelligence issues to modern power grids respectively smart grids, which are majorly dominated by optimization aspects. Here, new decentral infrastructure abilities offer a highly challenging research ground like distributed small-scale storages, decentral and partly non-dispatchable generation, integration of plug-in electric vehicles (EV), distributed switchable appliances, as well as combinations of them, which offer innovative control abilities. While probabilistic characteristics arise within these new fields, especially the optimal control of electric vehicle charging strategies provide numerous optimization topics. While the minimization of additional electric load for peak-shaving aspects as well as system losses can be stated here as central claims [23, 24, 100], numerous objective functions have been investigated in recent works [22], like financial considerations [94] or optimal incorporation of probabilistic renewables like wind power plants [109].

While all these novel control issues address variants of decision problems, these can be practically implemented due to the emerging trend of automated control in distribution systems. Here, actual ICT capabilities in modern power grids provide deep-going automation possibilities in order to realize intelligence in smart grids. This trend is clearly substantiated taking a look at actual standards like the IEC 61850 [26] for substation automation, which is used as well as extended in numerous works. [42] shows this trend when harmonizing this standard with the IEC 61499 in order to enable distributed automation in electric power grids, using well-established software architectures from state-of-the-art industrial automation systems. When talking about decision problems for optimal integration of electric vehicles as discussed above, essential standards have already been established like the IEC 61851 [27] for PWM[4]-based charging current control for conductive charging. The probably most important development in recent times from an ICT- point of view is the higher sophisticated standard IEC 15118 for vehicle to grid communication.

In the end it can clearly be stated, that optimization already plays an important role in power grid operation and control, whose significance tends to increase steadily. Additionally, the control of distributed (small-scale) devices in even lower-level networks is no more a theoretic concept, but yields real-world implementations all over the world and represents a key-technology within the deployment of smart electric grids.

Novel Methodical Trend: Application of Heuristic Search Strategies

From a methodical point of view, especially heuristic algorithms have been proven to be capable of solving upcoming optimization problems in electric power grid engineering in recent years. While basic OPF problems have been tackled with extended aspects like multiobjectivity [3], actual smart grid issues have been handled as well with various heuristic methods, ranging from simulated annealing to population-based strategies such as particle swarm optimization [1, 94], but being mainly dominated by variants of evolutionary algorithms like used by [4, 109].
General advantages of using metaheuristics have already been identified in the power engineering society [73], like capabilities of multiobjectivity and parallelization, performance on objective functions that are hard to optimize (discontinuities, non-linearity), or the ability of escaping local optima in multimodal problems, substantiating their increasing application as stated above. Especially when trying to compute optimal charging decisions for a huge number of EVs, these advantages come into play. Numerous researchers already identified the application of metaheuristics for this class of problems, using algorithms ranging from particle swarm optimization (PSO) [94, 106]

[4]Pulse Width Modulation

to various evolutionary algorithms like genetic algorithms (GA) , estimation of distribution algorithms (EDA) [103] or evolution strategies (ES) [52]. A more detailed overview will be provided in Chapter 6, when discussing EV charging control in the context of smart electric grids.
Not only to the electric vehicular problem domain, but also to manifold other fields heuristic (especially metaheuristic) methods play an essential role, like infrastructure planning issues [7, 59] or different kinds of OPF problems [2, 3, 73], to name just a few.

Thus, heuristic optimization founds a fruitful ground in electric power systems research, offering new abilities in various fields in power grid operation and planning. However, with the actual technological revolution to intelligent power grids, so called *smart grids*, operational aspects of power grids are faced with a significant change, that raises new challenges to respective optimization issues.

2.2 Smart Electric Power Grids

2.2.1 General Motivation of Smart Grids

In coming decades, the electric power grids are faced with decentralization, liberalization of the energy market, and an increasing demand for high-quality and reliable electricity [9]. The change to smart electric grids is seen to be a promising approach to match these upcoming needs.
Concerning the term *smart electric grid*, multiple definitions exist in the literature all partially varying in the features that the power grids should implement. Among them, the standard is the usage of sensors and communications technologies, enabling more efficient use of energy, improved reliability, and additionally enabling consumer access to a wider range of services [83]. Therefore, a core feature will be the integrated intelligence, which allows more efficient energy generation and distribution through better information. The term "efficient" thereby addresses on the one hand a decrease in overall energy consumption, on the other hand an increase of the reliability of electrical power supply while, at the same time, improving environmental friendliness. This smart electrical grid requires new technologies for power system operation, where especially optimization issues get more complex, being faced with completely new challenges.
The reasons for this increasing complexity are obvious: the progressive decentralization of comparatively smaller generation units and the hype of environmentally friendly zero-emission generators like photovoltaic plants or

wind turbines that cause stochastic delivery of electric power, complicate the supply-side situation in the power grid drastically. At the other side of the grid, the consumer side, the possibility of demand side management (load control) as well as the utilization of small-scale distributed storages leads to additional control parameters for optimization problems.

2.2.2 Changes Related to Operation and Optimization

In order to provide the reader with the necessary understanding on smart grid functionalities, those features shall be deepened that have main impact to the optimization area. As the optimal power flow (OPF) is a main tool for power grid operation and planning, the focus is laid on core aspects of smart electric grids that influence power flow optimization.

Reliability

As proposed before and building a central aim of [9], a major requirement to future power grids will be increased reliability of power supply due to the fact that so many critical infrastructures like communications and finance depend on reliable electrical power supply. Just this reliability is getting more and more stressed because of decentralization, stochastic behavior of zero-emission plants and complex interdependencies in the grid.

So, a central feature of smart electric grids should be the provisioning of autonomous and fast power flow control actions, preserving security and reliability considerations in power grid operation. Additionally, increasing stochastic influences (like from renewable generation units) have to be considered both in control as well as planning tasks for keeping power supply reliable[5]*.*

[5] When discussing reliability in power grids, the appropriate usage of such a well defined term makes it necessary to specify that this is a challenging topic in power systems research. Numerous complex partial problem classes such as security, stability, static and dynamic analysis as well as other subtopics belong to it. In this context, reliability is generally understood as the ability to supply appropriate electric service over a period of time [63], which yields - considering optimization applications - the guaranteed satisfaction of security constraints. In this work, this definition is assumed to be sufficient.

Availability and Usage of Distributed Electrical Storage

The application of energy storage in the electric grid has been a long-time issue of research and development, not only investigated in terms of the implementation of smart electric grids. For example, large scale storage units like pumped hydroelectric power plants or the technology of compressed air energy storage are already established for maintaining control reserves and enabling peak-shaving functionalities. Concerning reliability and its aspects like voltage stability, the application of smaller electrochemical storages, for instance lead-acid batteries, for critical infrastructures like large server-centers, is an important practice in modern power grids.
Complementary to the mentioned large-scale storage devices, the progressive development of small-scale storages like lithium-ion or sodium-sulphur batteries establish new possibilities for the modern grid in regard to cost aspects, energy- and power-densities or duration of load cycles of storage devices. Especially these small-scale devices substantiate the way to the implementation of distributed energy storages, being a major characteristic of smart grids.

There exists a large spectrum of possible applications. For example, diurnal peak shaving of energy storage devices would actually reduce transmission and distribution losses. Another aspect especially of distributed storage is the ability of storing energy in multiple locations closer to the end-use consumer, which would lead to increased reliability and lower system losses. Particularly considering renewable sources, the ability of time-shift generated energy from the moment of generation to a later point when it is needed or when transmission/distribution capacity is available is an important benefit. However, such small-scale storages would need to be implemented and controlled at lower power grid levels and in a highly distributed manner. Their operation will need to consider complex interdependencies between stochastic supply from renewables and volatile demand situations.

Hence, in the area of electrical storages for power grid operation, control methods need to be developed for handling high quantities of distributed devices (i.e. high amounts of control variables) in dynamic power flow environments.

Load Control

At the customer-side in the smart electric grid, another important characteristic is the usage of automated meter reading for generation and load control. Digital meters provide real-time energy consumption data to the utility, realized through interconnected devices installed at each end-user, the so called smart meters. Additional to the simple reading of actual power consumption, advanced metering systems should enable the possibility to control load demand. Therefore, the customer accepts the risk of delay or refusal of energy-use by the utility in order to receive some discount, which enables the so called load control. This advanced metering technology not only allows end-users to actively influence their energy consumption, it is important for utilities and distribution companies to control the demand side effectively during times of emergency or peak-demand and thus gaining control energy for reliability services.

From a computational point of view, both features concerning distributed electrical storages as well as load control functionalities have a tremendous influence on optimization problems since they increase the amount of control variables for providing accurate control decisions. For example, taking the general OPF formulation, not only power supply decisions from central generation units need to be made, but numerous control variables are related to all the distributed appliances in a system that need to be controlled. Additionally, reliability concerns challenge optimization methods to provide robust solutions under increasingly stochastic influences, both for fast control actions as well as reliable planning decisions.

For concluding all these issues, methods need to be developed that enable (approximate) optimal control and planning of volatile, noisy as well as nonlinear distributed power systems. Hence, new techniques for power flow optimization strongly need to deal with stochasticity, dynamics as well as scalability to numerous devices.

2.2.3 Central Challenge: Dynamic and Stochastic Optimal Power Flow

Intelligence Through Optimization

"How can we develop the algorithms needed in order to better approximate optimal control over time of extremely large, noisy, nonlinear systems?"
P. J. Werbos [112]

Since the early years when researchers started to investigate what they called "artificial intelligence", optimization represents a fundamental building block, since nearly every "intelligent" decision making process can be described by formulating an optimization problem. Essential statements have been published decades ago, when Simon [77] and Raiffa [88] showed that "intelligent" problem-solving or goal-seeking behavior in any case can be understood as an application of optimization over time. However, this insight shapes research in computational intelligence since that time. Considering power grids, OPF provides a framework of integrating manifold decision issues across a complex power system, where stochasticity as well as dynamics have to be taken into account. Researchers already identified this aim, stating the solution of *dynamic stochastic optimal power flows* (DSOPF) as core-issue for enabling intelligent control of future power grids [69, 72, 114].

Optimization over Time / Dynamic Optimization in Power Grid Engineering

Many works in smart grid research are being performed to improve the OPF by adding the "S" (stochastic) rather than the "D" (dynamic). Sure, these investigations are important for increasing the understanding of stochastic issues, but at the same time do not enable the essential ability of making predictive, foresightful control, integrating future states as well as their impact into present decisions.

Taking a look at power grid optimization, tasks that necessitate fast and robust dynamic control are obvious. Taking exemplarily the general OPF as mentioned before, the aim is to find the optimal configuration of all controllable units for satisfying a given load situation, using steady-state representation of the power grid. Thus, the solution of this problem addresses exactly one stationary state $J(t)$, disregarding possible states in the near future or eventual uncertain conditions in the system. Considering the system one time step later ($J(t+1)$) due to changing conditions of weather, customer-behavior or any other influence, the power flow in the system would change, hence, requiring a new solution to the optimal power flow problem further necessitated by the non-linear behavior of an electric power distribution system. Such a new computation would require a robust and fast-converging solution method, that guarantees quick support with a new optimal solution, independent of system complexity and starting point, which cannot be guaranteed by traditional steady-state OPF methods[6] [111]. This concern is further complicated by the steady increase of the number of control variables in smart grid applications, as stated above. Thus, electric power systems fundamentally represent applications that require dynamic optimization techniques, respectively methods that enable optimization over time.

[6]Especially when considering large-scale systems and mixed-integer formulations with different kinds of controllable appliances.

Principally, anticipatory optimal solutions can be computed in a deterministic way for future time steps $t + K$, assuming that the near future can be predicted sufficiently. In such a case, K solutions can be obtained beforehand, considering variations of the system in the near-term interval, but expecting them to be deterministic and ignoring their potential stochastic nature. Such approaches entitled by the term *deterministic lookahead optimization* [86] are intuitive but have the disadvantage of being unable to react to volatile situations.
However, in a dynamic and volatile system such as an electric power grid, it is more appropriate to make decisions as they come up, and therefore react to new situations very quickly without computing a completely new solution at each time step t. This can be seen as the general aim of so called optimization over time respectively dynamic (control-) optimization. In order to meet future smart grid requirements, a scalable and generic approach is needed, that enables such dynamic optimization in volatile as well as stochastic environments.
Especially in the field of dynamic optimization within stochastic systems, principles related to so called *policy function approximation* [86] have evolved in recent years. Policy function approximation assumes that K arbitrary optimal solutions in the future can be approximated by an analytic function (i.e. policy) that returns a (near-) optimal action given a specific state at runtime. Here, the optimization problem aims at finding this function, while avoiding the need of doing (re-)optimization at runtime. Such a technology will have to be developed in the field of optimal power flow control in order to meet future operational issues in smart electric grids.

Having discussed the principles of the problem domain as well as the challenges that evolve herein concerning optimization, one central aim of this work is the development appropriate methods in order to overcome these challenges. Therefore, optimization under uncertainty plays an important role where the next chapter aims at building the methodical fundament for later optimization methods development.

Chapter 3

Theory II: Simulation Optimization and Dynamic Optimization Under Uncertainty

3.1 General Introduction: Optimization in Stochastic Systems

Whenever optimizing a system, an appropriate representation of this system is needed - such representations are commonly called "models". Generally, an operations researcher seeks finding a model in a form that is manageable from an optimization point of view, such as a linear model, which allows the application of deterministic, robust and exact optimization methods. In the more general case, as long as it is possible to derive a closed-form algebraic model, well-established analytic optimization methods exist in a high variety, that can be applied to find solutions deterministically and in a robust manner. To name just a few, most popular methods are simplex method, originally developed by Dantzig [30, 31] and multiply extended like by Nelder and Mead [76], or gradient descent-based methods [38]. A manifold of diverse other methods exist as well, being too extensive to be discussed at this point.

However, for many real-world applications stochastic systems have to be considered, that are governed by the uncertain nature of (random) variables. For such systems, it is very hard or even not possible to derive a closed-form algebraic model, but computational methods can be applied to obtain the system's estimated response to a given situation with respect to specified control variables by drawing samples of the random variables from their distributions. As during optimization one aims at minimizing or maximizing this response, it is called objective function. Thus, for stochastic optimization problems one has to deal with optimization methods that perform without knowing an analytical expression of this objective function, while - in the general case - two approaches can be distinguished: one general approach is to approximate the stochastic system by a deterministic one (in order to apply deterministic optimization methods to it), the other is to work with optimization methods that only rely on function evaluations for given points in the solution space - so called (heuristic) search based optimization methods. The latter will be the core technology within this work, since they allow the integration of a system's stochastic behavior into the optimization process without the need for approximations.

At this point it needs to be defined that the term *stochastic problem* is distinct from *stochastic optimization*. While the latter addresses the stochastic optimization procedure (such as metaheuristic search methods), the first term considers the stochasticity of the problem. Within this work "stochastic optimization problem" is defined to denote that the treated problem itself is stochastic, where this definition is independent from the concretely applied optimization method.

3.2 Review on Existing Approaches

Several general approaches exist for optimizing stochastic systems, an overview shall be given in order to position this work into the contextual literature.

State Space Discretization: Scenario Trees and Markov Decision Processes

The application of scenario trees is a suitable approach for handling stochastic optimization problems, where a stochastic system is reduced to a set of discrete system states (i.e. scenarios), which are later used for optimization. A scenario tree therefore is a discrete approximation of the evolution of the system's uncertainties over time, considering a root-state at time t and all its possible successor states at time $t+1$ ... $t+K$, where K is the considered time horizon. The scenario tree consists of a finite set of decision making points that represent system states obtained by sampling the probabilistic variables. Since each state is connected to its predecessor respectively successors via transition probabilities, this approach as exemplarily shown in Figure

3.1 is similar to modeling a system as a Markov Decision Process (MDP). Here, the evolution of the system beginning from a defined root state is depicted, where transition probabilities are indicated with P [1]. Performing optimization on such a model usually means finding a solution with the highest expected objective function value over all paths in the network (respectively graph) of the considered states, taking into account the states' probabilistic interdependencies. [81] used this approach for computing control variables for a wind-integrated power system based on particle swarm optimization. As similar approach, [38] extensively describes the optimization of stochastic systems using Markov Decision Processes in operations research generally (and will come into play once more later), while [60] describes a smart grids related application when scheduling decentral load devices under uncertain price conditions.

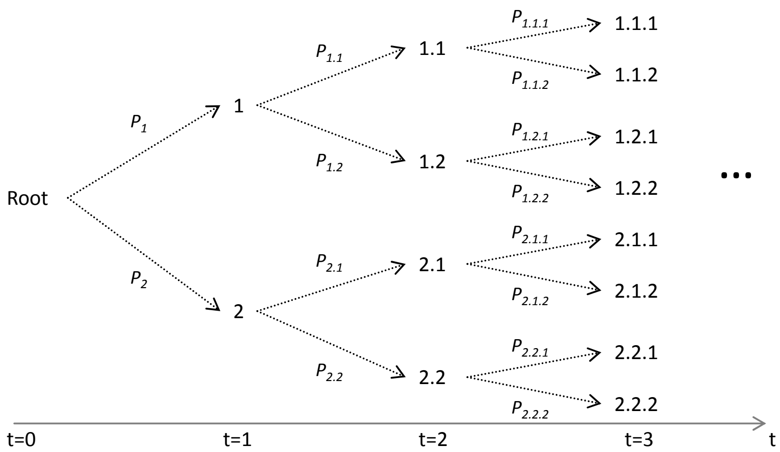

Figure 3.1: Principle of State Space Discretization

Surrogate Modeling

Another approach of approximating a stochastic optimization problem by a deterministic one is based on creating metamodels. Different principles exist like kriging (coming from geostatistics), response surface modeling, surrogate modeling or simply metamodeling, that all address mainly the same issue. The general aim is to draw samples from the stochastic model and

[1]The variable P is only used within this explanation for denoting a probability. Throughout the rest of the thesis, P provides a variable for real-power values (as in the list of symbols).

approximate them using an exact (partly closed-form) representation that is suitable for subsequent optimization. Here, local approaches model neighboring points by comparably simple models that estimate local gradients used for gradient-based optimization methods, building trust-region methods [33, 110]. Those local estimates are based on interpolation, linear, piecewise-linear, quadratic or Gaussian models [110] or nearest neighbor regression [38]. Global methods address the aim of finding an overall model based on sampled points from greater regions in the state space, where more powerful modeling methods are applied like artificial neural network (ANN) regression [38].
Surrogate modeling is considered as alternative but less popular approach for handling stochastic problems and will not be treated further within this work. More detailed information can be obtained from the referenced literature.

Heuristic Optimization Under Uncertainty

Most established linear as well as nonlinear programming methods require a closed-form model for optimization. Many of those exact/analytic methods base on the computation of a system's derivatives.
In numerous real-world applications, a closed-form mathematical representation of the treated system cannot be obtained. Reasons for this are mostly too high system complexity, discontinuous variables or probabilistic influences. Therefore, operations researchers tend to apply simplifications using Markov Decision Chains, queuing theory or even simulation for describing the system. Here, a high interest in optimization methods evolved over the last decades, that do not require a closed-form. So called heuristic optimization methods have been developed, that do not need derivatives, but only rely on the value of the objective function at any given point. Contrary to analytic/exact methods, they have the advantage of performing without a closed form model. Additionally, since they deploy an informed search procedure, they avoid the need of enumeratively evaluating the complete solution space. Furthermore, they do not require specific solution representations (such as vectors of real-valued numbers). With heuristic search methods, a solution candidate can take any arbitrary representation such as a graph-structure, a set of rules, a tree, or even any other. Thus, they provide a fruitful ground for optimizing complex decisions in large-scale stochastic real world systems, where the closed form is mostly unobservable, but the objective function value can be evaluated at any point, for example with simulation. This ability is essential when treating stochastic systems.

Solution Evaluation Within a Stochastic System

At this point, the issue of solution evaluation in stochastic systems needs to be considered.
For a deterministic system (in case of minimization), if solutions u_1 and u_2 both satisfy all defined constraints, then u_1 is of higher quality than u_2 if and only if $f(u_1) < f(u_2)$ (with $f(u)$ being the objective function). In stochastic systems, this relation is much more difficult.
A system is defined to be stochastic if its state variables depend on random influences, which causes a special challenge for optimization. When performing heuristic optimization as stated above within a stochastic system, the issue arises that the objective function evaluation at a given point in solution space is not deterministic, but has to be approximated by its estimate. This issue will be discussed more in detail later. However, since most stochastic and complex systems cannot be described in closed form, it is common to model them using simulation.

3.3 Simulation-Based Optimization

In simulation optimization (synonym: simulation-based optimization), the aim is to unify simulation with (search-based) optimization methods. Here, the objective function value of a given solution is evaluated by simulation, where - especially for stochastic systems - this evaluation only gives an estimation of the objective function value rather than the true value. This issue can be considered as stated in Equation 3.1, where $f'(u)$ gives the estimate of the true objective function value $f(u)$ that is biased by some noise ϵ. u represents the solution, whereas its concrete outlook will be discussed later.

$$f'(u) = f(u) + \epsilon \tag{3.1}$$

Many investigations in literature on simulation optimization are about minimizing the difference between $f'(u)$ and $f(u)$ for objective function evaluation, thus, maximizing the estimate's quality. While estimating the objective function value is of great importance, the interested reader shall be referred to related literature. A complete overview can be obtained from works by Branke [18], Rinott [90] or Stagge [102]. However, since the core of this work will be laid on developing control optimization methods in stochastic, dynamic, large scale "intelligent power grids", an appropriate estimation method for $f'(u)$ is considered to be given and will not be discussed in more detail.

3.3.1 Metaheuristic Simulation Optimization

For simulation optimization, metaheuristic algorithms are already proven to be suitable optimization algorithms [38, 54], since they principally offer generic problem-independent functionality, even if their specific configuration has to be tuned according to the concretely applied problem instance. The operations research society already identified the capabilities of simulation optimization methods when treating probabilistic (and partly dynamic) systems, profound reviews can be found in survey papers such as [11, 19, 36, 37, 67, 104]. Successful applications have been performed in popular areas like logistics planning, production scheduling or even business process optimization. However, the simulation optimization principles seem to be suitable to the herein defined issues, hence, shall now be considered for dynamic and stochastic optimization tasks in power grid engineering.

As discussed before, the main idea of simulation optimization is to evaluate the fitness of a solution candidate through simulation, thus the only needed information of a candidate (i.e. point in solution space) is its objective function value $f(u)$ (respectively its estimate $f'(u)$). Generating candidates according to a specific strategy, the optimization algorithm further searches for optima in a given state space that is defined by simulation. This procedure is shown in Figure 3.2. Since evaluation within the stochastic system representation is uncertain, for a given solution candidate, varying objective function values result from simulation. In order to overcome this variance, each solution candidate has to be simulated multiple times in order to get an appropriate estimate $f'(u)$ of the solution's performance. This is mostly referred to the term "sampling".

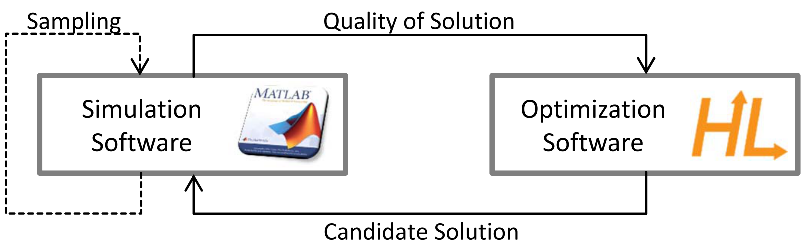

Figure 3.2: Principle of Simulation Optimization

Computational Costs

Using metaheuristic simulation optimization is a big issue of computational time, since in most applications many 1000 points in solution space have to be evaluated. Additionally, when stochastic simulation is applied, computational costs of evaluating a solution candidate grow additionally because of the need for multiple sampling. Many researchers already investigated this issue [18, 90, 102], where the central aim is to find a tradeoff between computational costs of solution evaluation and the quality of the estimate $f'(u)$. Since the accuracy of this estimate grows with $1/\sqrt{s}$, with s being the number of samples, a suitable choice for s is crucial. While a high value increases the estimate's accuracy, it leads to waste of computational effort for worse solution candidates. At the other hand, a too low sampling rate would lead to misjudgment of a solution's quality, which may decrease the efficiency of a metaheuristic search process. A thorough overview on this issue can be obtained from the referenced literature, while the author of this work proposed a new sampling scheme with special consideration of offspring selection genetic algorithms (OSGA) [44]. An appropriate sampling scheme is assumed to be given within this work and will be specified when experimental investigations are conducted.

3.3.2 Unifying Simulation with HeuristicLab

Simulation tools already exist both in high quantity and with manifold functionality. Generally, too many tools exist in the field of power grid engineering to be mentioned here, a detailed overview can be found in [12]. At this point, Matlab has to be discussed in more detail, since it is probably the most popular framework when talking about simulation. Matlab provides an extensive method base containing general simulation solvers as well as analysis functions. Even if it is a quite general framework applied in different technical and scientific fields, Matlab-based simulation tools for power grid engineering have come up in recent years, applied all over the world. To name just a few, SimPowerSystem, Power System Analysis Toolbox (PSAT) as well as MATPOWER provide all standard-functionalities used in power system analysis. Especially the last two mentioned tools are open source and therefore highly attractive to not only practitioners, but researchers as well. Within this work, finally MATPOWER proved to be the best choice, both because of its extensive power system analysis functions as well as its usability and extension capabilities.

At the same time, optimization frameworks have evolved in the metaheuristics community. This is primarily due to the No-Free-Lunch theorem [115], which expresses that no single algorithm can beat any other on all possible applications. Hence, especially when applying metaheuristic optimization to practical problems, appropriate algorithm selection and tuning is necessary in order to obtain good solutions. Therefore, experimentation abilities need to be offered by such frameworks, that mostly provide an extensive workbench consisting of different algorithms. A thorough overview and comparison of frameworks is provided in [82], while in this work HeuristicLab is applied.
In the increasing field of simulation optimization various complete tools evolved too in recent years, where [36] and [37] give an overview on existing solutions. However, all these solutions are restricted to a specific spectrum of optimization functionalities tailored to the concrete application domain, curtailing the full power of metaheuristic algorithms. A richer approach is to provide an optimization environment that is independent from the simulation software, and which provides additional optimization specific features. It is a powerful approach to realize simulation-based optimization by connecting arbitrary simulation tools to a sophisticated framework for metaheuristic optimization, making full use of both frameworks' abilities. This can be realized using HeuristicLab, which is an open source framework whose architecture is designed especially for plug-in based extension and therefore offers a suitable solution for generic simulation-optimization interoperability.

Interprocess Communication

For performing simulation optimization according to Figure 3.2, interprocess communication is needed for enabling the interplay of simulation and optimization software. All experiments of this work are performed by unifying HeuristicLab with Matlab-MATPOWER for simulation optimization. The interface is realized using Matlab automation server, which uses component object model (COM) protocol. Therefore, a HeuristicLab plug-in controls COM objects exported by Matlab. Within these objects, simulation is performed and the fitness of the solution candidate is assigned. This architecture is equal for all following experiments within this work, thus forms the basic framework for experimentation. Further information in simulation optimization with HeuristicLab can be obtained from [54] as well as the HeuristicLab documentation[2].

[2] http://dev.heuristiclab.com

3.4 Optimization Paradigms: Static and Dynamic Optimization

At this point, the specific structure of a solution candidate has not been specified, but depends on the optimization problem. Here, the choice whether a static or dynamic problem is at hand yields essential consequences on the needed optimization method respectively the solution representation.

3.4.1 General Description: Static vs. Dynamic Problems

Essentially, two different problem classes can be distinguished which characterize the time domain of the optimization, namely static and dynamic optimization problems. Static optimization mainly aims at finding a set of parameters to a system in steady-state condition (often called parametric optimization). For dynamic optimization, the principal task has to be handled that the treated system changes continuously, requiring flexible solutions or reoptimization of them. Most often dynamic optimization problems are related to control optimization. Since within this work approaches for dynamic stochastic optimization problems are developed for the power grid engineering problem area, the following considerations regarding the time domain of optimization shall be conducted with respect to the optimal power flow problem as reference application.

3.4.2 Optimal Power Flow Problem Statement

The standard optimal power flow problem aims at minimizing the fuel cost per hour of a power system in steady state conditions, while fulfilling all equality and inequality constraints. The OPF problem can be mathematically formulated as follows [73, 116]:

$$Minimize : C(\overline{x}, \overline{u}) \tag{3.2}$$

subject to:

$$g(\overline{x}, \overline{u}) = 0, \tag{3.3}$$
$$h(\overline{x}, \overline{u}) \leq 0, \tag{3.4}$$

where $C(\overline{x},\overline{u})$ gives the objective function, $g(\overline{x},\overline{u})$ is the set of equality constraints and represents typical load flow equations, $h(\overline{x},\overline{u})$ are the (inequality) system operation constraints. The inequality constraints $h(\overline{x},\overline{u})$ reflect the limits on physical devices in the power system as well as the operational limits created to ensure system security.
$\overline{x}$ and $\overline{u}$ are the vectors of dependent and independent variables. The vector $\overline{x}$ consists of (dependent) state variables: slack bus[3] real power output P_{G_1}, load bus voltages V_L, load bus power demands P_L and Q_L, generator reactive power outputs Q_G and branch power flows P_B.
The vector $\overline{u}$ comprises control variables: generator voltages V_G, generator real power outputs P_G, transformer tap settings T_{tap} and the output of shunt VAR compensators Q_C. Hence, both vectors are expressed as:

$$x^T = [P_{G_1}, V_{L_1}...V_{L_{NL}}, P_{L_1}...P_{L_{NL}}, Q_{L_1}...Q_{L_{NL}}, Q_{G_1}...Q_{G_{NG}}, P_{B_1}...P_{B_{NB}}] \tag{3.5}$$

$$u^T = [V_{G_1}...V_{G_{NG}}, P_{G_2}...P_{G_{NG}}, T_{tap_1}...T_{tap_{NT}}, Q_{C_1}...Q_{C_{NC}}], \tag{3.6}$$

with NL, NG, NB, NT and NC being the number of load buses, generator buses, branches, transformers and shunt VAR compensators respectively.

The objective function is to minimize the total fuel costs per hour over all generators. The individual costs of each unit are expressed typically by a polynomial:

$$C(P_{G_g}) = a_g + b_g * P_{G_g} + c_g * P_{G_g}^2, \tag{3.7}$$

with a_g, b_g and c_g representing each generation unit's cost coefficients.

While minimizing the objective function, the following constraints have to be satisfied, considering real- and reactive-power generation capacities

$$P_{G_g}^{min} \leq P_{G_g} \leq P_{G_g}^{max}, \tag{3.8}$$

and

$$Q_{G_g}^{min} \leq Q_{G_g} \leq Q_{G_g}^{max}, \tag{3.9}$$

[3]The slack bus method is used in power flow computation, where the injection (generation) value of one defined bus in the system is not fixed beforehand, but gets adapted at the end of the computation in order to meet the system losses and thus satisfy power balance in the system. This bus is entitled as slack bus.

as well as the voltage deviation being restricted to

$$V_j^{min} \leq V_j \leq V_j^{max} \tag{3.10}$$

over all buses $j = 1, ..., J$ and all generators $g = 1, ..., NG$. Branch power flows shall be constrained to

$$P_b \leq P_b^{max} \tag{3.11}$$

for all branches $b = 1, ..., NB$ for enabling secure distribution grid operation.

Equations 3.8 - 3.11 define standard load flow constraints in power grids an will be necessary too in most experiments later.
Additionally, the control variables for transformer tap settings and VAR compensators need to be restricted to the ranges:

$$T_{tap_{nt}}^{min} \leq T_{tap_{nt}} \leq T_{tap_{nt}}^{max}, \tag{3.12}$$

and

$$Q_{C_{nc}}^{min} \leq Q_{C_{nc}} \leq Q_{C_{nc}}^{max}, \tag{3.13}$$

for all transformers $nt = 1...NT$ and VAR compensators $nc = 1...NC$. The variable $Q_{C_{nc}}$ represents the reactive power output from the shunt-connected static reactive power compensators. These VAR-devices are applied for managing reactive-power balance, hence, are used for voltage regulation. The variable $T_{tap_{nt}}$ gives the tap position for transformers with tap-changing mechanisms. This technology allows the secondary-side adaptation of the actual number of used turns along a winding, enabling voltage regulation at the transformer output.

The traditional OPF formulation aims at finding a static solution $\overline{u}$ for a given situation, thus, determining concrete values for all control variables. However, in dynamic situations it would be more appropriate to have flexible control actions on hand that guarantee (near-) optimal power flow control, rather than static values for control variables.

3.4.3 Static and Dynamic Optimal Power Flow

The traditional OPF as stated above is a classical application of static optimization, where the solution $\overline{u}$ gives a vector of real-valued control variables for a certain steady-state situation. However, as discussed in Section 2.2.3, power flow control clearly exhibits situations where dynamic optimization is necessary, making OPF a fruitful benchmark problem for describing dynamic stochastic optimal control approaches. The following considerations will concentrate on power flow optimization, thus, constructing the related approaches based on this electrical engineering domain. Here, the formulations are based on the traditional statement of the optimal power flow (OPF) problem with the target of obtaining a scalable control method for dynamic stochastic optimal power flows, which will later be extended according to smart grid considerations.

Part II

Simulation-Based Dynamic And Stochastic Power Grid Optimization

Chapter 4

Optimization Methods

So far, simulation optimization seems to build a fruitful ground for optimization under uncertainty of complex systems. From this methodical fundament, methods shall be developed in order to provide stochastic and dynamic optimization in future electric power grids.

4.1 Static Optimization of Stochastic Problems

As already discussed, static optimization can be considered as the process of finding best-performing parameter values for a given problem, seeking a steady-state solution. Static optimization is probably the most investigated class (compared to dynamic optimization), also providing lots of methods as well as applications for simulation optimization in literature, a profound overview can be found in [36, 38].

4.1.1 Description of Static Power Flow Optimization

The principal functionality of simulation-optimization has already been shown in Figure 3.2, which holds both for static as well as dynamic problems. Referring to this Figure, for static problems the solution candidate is represented by a set of (discrete or continuous) values. These values directly serve as input parameters to the simulation model and yield the quality of the solution after simulation. Considering a practical example, the steady-state OPF as formulated above provides a typical static optimization problem, where the solution $\overline{u}$ is a parameter vector

$$u^T = [P_{G_2}...P_{G_{NG}}, V_{G_1}...V_{G_{NG}}, Q_{C_1}...Q_{C_{NC}}, T_{tap_1}...T_{tap_{NT}}]$$

of fixed size, yielding the resulting costs per hour of operation as quality.

Besides the traditional OPF, a plurality of other optimization problems exists in power grid research seeking static solutions, like placement decisions for infrastructure planning, unit commitment, or maintenance scheduling, all discussed in the introductory chapter. Additionally, very popular general problem formulations from operations research mostly aim at finding static solutions (at least in their original formulations), like so called backpacking problems, traveling salesman problems or quadratic assignment problems.

4.1.2 Limitations of Static Considerations

While static approaches only consider a system $J(t)$ in steady-state condition, they disregard possible dynamics of the system or even force a recomputation of the solution after a changing event. In such a case, it would be more appropriate to make a solution flexible for reacting to dynamic conditions, unchaining it from its steady-state limitation. Therefore, dynamic considerations have to be integrated into the OPF control, which can be realized with simulation optimization.

Detailed experimental illustrations of the limitations of static considerations will be provided in Chapter 6, while the general seek for flexible and dynamic solutions is obvious now.

4.2 Policy-Based Optimization of Dynamic Stochastic Problems

While the term dynamic optimization can generally be understood as the issue of determining optimal actions in a dynamic environment, related techniques find applications in manifold practical fields. Methods for this class of optimization have successfully been developed for applications like logistics or financial engineering [86], while considerations for power flow control build the core of this work.

4.2.1 Related Approaches in Literature

The operations research society early identified the need for dynamic optimization methods. Much like Dantzig formulated fundamental insights for linear optimization problems (both deterministic as well as stochastic) that shaped this optimization class ever since, quite at the same time Bellman specified sequential dynamic problems to be solved by dynamic programming [13], and formulated his famous optimality equation for describing sequential optimization. While in classic dynamic programming the integration of uncertainties into optimal sequential decision problems has been a challenging issue since then, the general conjunction of dynamic and stochastic optimization remained residual for some decades.

Arriving in the late 90s, operations researchers determined essential approaches for enabling this conjunction when unifying dynamic programming methods with approximation approaches.
Werbos [113] developed a class of methods to perform this kind of optimization in any engineering application, stating Approximate Dynamic Programming (ADP) as flexible and scalable technology being suitable for future smart grid issues as well. While classical Dynamic Programming [13] is an exact method being limited by the so called "curse of dimensionality", ADP comes up with concepts of finding - as the name already says - sufficiently accurate approximate solutions to control problems. Coming from ADP, a family of Adaptive Critic Designs (ACD) were proposed that combine ADP with reinforcement learning techniques for finding an optimal series of control actions that must be taken sequentially, not knowing their effects until the end of the sequence. While the standard ACD combines a set of neural networks within a neurocontroller for approximating the Hamilton-Jacobi-Bellman equation associated with optimal control theory, further concepts like Dual Heuristic Programming, Globalized Dual Heuristic Programming and Action-Dependent Heuristic Programming extend this formulation and build the family of ACDs.

Several researchers already picked up this idea of ACD, applying these concepts to electric power grid problems. Venayagamoorthy adapted it for building an automated voltage regulator of a generator system in [107, 108] or even for operating a 12-bus power grid with an ACD-enabled Dynamic Stochastic Optimal Power Flow (DSOPF) controller in [69]. The necessity of solving such a DSOPF in future power grid operation has been substantiated by Momoh [72, 74] sufficiently, stating its generality for being suitable to a plurality of problems. Xu [117] even applied a simpler variant of ADP for solving an electric vehicle charging control problem.

However all approaches related to ACD evolved from optimal control theory and thus try to approximate the Hamilton-Jacobi-Bellman equation with computational intelligence techniques. Building an alternative approach for optimization over time, the evolution of flexible control policies using metaheuristic algorithms as demonstrated within this work shows validity for electric power engineering problems as demonstrated further on. A differentiation from existing methods in the area of ACDs will be deepened later in Section 4.2.10.

4.2.2 Simulation-Based Evolutionary Policy Approximation

> *"Policy function approximations (PFAs) are used when the structure of the policy seems obvious. A PFA is an analytic function that returns an action given a state, without solving an imbedded optimization problem."* Warren B. Powell et al. [86]

When optimizing power flows over time in a given distribution system, the aim is to find actions $u(t)$ for controllable units that lead to optimal behavior over a given time interval $[t, (t+K)]$. In such a case, if all possible information of the system in near-future states $J(t+1)...J(t+K)$ could be predicted accurately, it would be possible to compute anticipatory solutions (actions) $u(t+1)...u(t+K)$ beforehand that are optimal to each near-future state, enabling optimal power grid operation in the near future (*deterministic lookahead optimization*). Here, steady-state optimization would be appropriate for finding an accurate solution to each predicted state separately. But, what if the system changes at time $t+k$ with $k < K$? This event would cause the necessity of computing a completely new solution (or at least the adaptation of a solution), making the usage of such a predictive steady-state optimization less attractive in dynamic systems. Here, it would be more appropriate to not compute all actions $u(t+1)...u(t+K)$ beforehand, but to make decisions online in a robust and quick way. Therefore, flexible policies can be used that are trained offline but lead to accurate decisions that react to new situations quickly, avoiding the need of re-optimization in each event while being able to consider full volatility and uncertainty of the system.
Figure 4.1 illustrates these two variants, namely the predictive computation of steady-state solutions in the upper part of the figure, versus the offline-training of a flexible policy, that takes system states at runtime and directly computes accurate actions online. While the first approach computes fixed solutions beforehand, that are unable to react to dynamic and uncertain conditions, the policy-based approach keeps full flexibility during operation. Equations 4.1 and 4.2 should formalize this distinction between predictive steady-state solutions and flexible (approximative) policies.

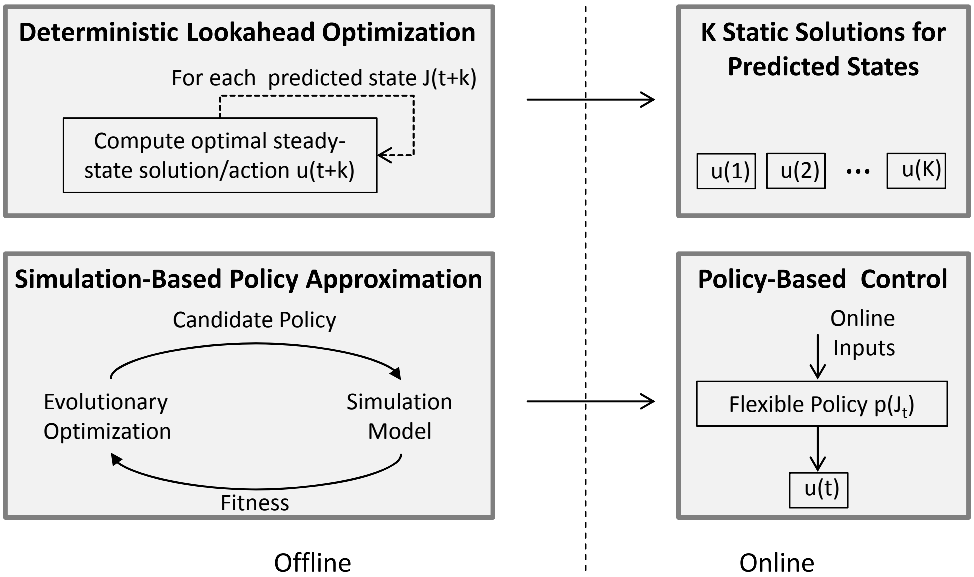

Figure 4.1: Policy-Based Control vs. Lookahead Optimization

$$\overline{u} = \arg\min_{\overline{u}\in\mathbb{R}}[E(\sum_{t=1}^{T} C(u_t))] \tag{4.1}$$

$$p = \arg\min_{p\in\mathbb{P}}[E(\sum_{t=1}^{T} C(p(J(t))))] \tag{4.2}$$

While in Equation 4.1 for each state at time $t = 1...T$ a deterministic action $u_t \in \overline{u}$ needs to be defined beforehand that minimizes the expected costs (objective function), using a flexible policy p in Equation 4.2 the specific action at time t does not directly depend on the value of t, rather than of the whole state $J(t)$ respectively the inputs that the policy receives from the system at this time.

As for control problems, actions (respectively control variables) are typically real-valued numbers, according to Equation 4.1 the vector $\overline{u}$ has to be optimized in the set of real numbers $\mathbb{R}$. When optimizing a policy p with regard to Equation 4.2, this issue is a bit more complex, since p takes the form of an analytic function, thus needs to be optimized in the space of potential policies $\mathbb{P}$. More discussions on this issue will done later when considering the mathematical structure of policies.

Considering the steady-state solution to the general optimal power flow, the vector $\overline{u}$ typically consists of generator real power outputs (except the slack bus), generator voltages (including slack bus), shunt VAR compensators' outputs as well as transformer tap settings. Hence,

$$\overline{u}^T = [P_2...P_{NG}, V_1...V_{NG}, Q_{C1}...Q_{CNC}, T_{tap_1}...T_{tap_{NT}}],$$

with number of generator buses NG, number of VAR compensators NC and the number of switchable transformers NT. Thus, for the predictive steady-state optimization, one would need to compute $\overline{u}$ for each future state, hence the solution would be

$$\overline{u} = [\overline{u_1}...\overline{u_K}],$$

for $t = 1...K$. When using flexible policies, each variable (P_G, V_G, Q_C, T_{tap}) is represented by a policy. This policy $p(J(t))$ is a function of the system's global state as well as the unit's local conditions and outputs the unit's actions $\overline{u(t)}$ in order to achieve the optimal power flow in the grid.
Now, after having defined the principal functionality of policy-based optimization, how can such policies be computed, that lead to (near-) optimal control actions at runtime?

4.2.3 Requirements for Policy Approximation

Before discussing the technology for obtaining policies, one will need to define the requirements that a desired policy approximation shall satisfy. When considering the defined dynamic OPF problem on the one hand, but also the general power flow control requirements in future smart electric grids (as provided in Section 2.2) on the other hand, the following requirements can be derived:

1. Policies need to be learned within a stochastic environment.
2. Robust control decisions will have to be derived from the policy under stochastic and volatile situations.
3. Learning/approximation and application can be performed on arbitrary large systems (i.e. number of state-variables).
4. Control needs to be scalable to numerous, distributed and interdependent devices.
5. Policies need to be able to identify complex, nonlinear relationships in real-world systems without a-priori knowledge of the their structure.

The first two requirements consider the general nature of control issues in dynamic and uncertain power flow environments. While on the one hand the robustness of derived actions at runtime is of high importance, during the approximation (learning) of policies this has to be integrated appropriately. Thus, the learning-phase needs to implement optimization under uncertainty. Regarding requirements number 3 and 4, the scalability issue of such applications is addressed. This scalability is twofold: while power flow control problems occur in differently sized systems ranging from small low-voltage distribution feeders to large-scale transmission networks, especially with regard to future decentralization in power grid operation as well as emerging distributed control devices an increase of controllable units needs to be tackled. Hence, both the scalability with respect to system size (i.e. number of system/state variables) as well as the scalability to control increasing amounts of devices are of importance.
Finally the last requirement is the most challenging one especially when considering existing approaches in literature. As stated in the quote at the beginning of this section very clearly, policy approximation is attractive when the structure of the policy p is obvious. But, what if we do not know about the outlook of this structure? Here, a technology will be needed that is able to evolve policies without a-priori knowledge of their mathematical structure.

All these requirements will need to be satisfied in order to build a technology for dynamic stochastic optimal power flow control based on policy approximation. In later sections, these requirements will be referred in order to judge the developed approaches on a qualitative level.
At this point, it is obvious to discuss about the information that a potential policy will need to take for deriving power flow control actions, while discussions on approaches for evolving such policies will be provided later.

4.2.4 Policy Design: System Variable Synthesis

Tentatively, we assume a fixed mathematical structure $p(\bar{i})$[1] that represents the policy, where the vector $\bar{i}$ contains all relevant input variables for decision making. This policy $p(\bar{i})$ serves as optimal power flow controller and provides the controllable unit respectively the grid operator with fast and robust actions.
In order to determine such a policy, one has to discuss the available variables when considering steady-state power flow computations. As defined previ-

[1] $\bar{i}$ has not been discussed yet, but shall be denoted as a generic vector that contains inputs to a control policy, without having specified these inputs so far.

ously when stating the OPF formulation, the nomenclature has evolved of defining $\bar{i}$ as the vector of input variables, $\overline{x}$ contains the (dependent) state variables and $\overline{u}$ comprises independent control variables. Thus, $\bar{i}$ addresses all variables that come from the power grid's environment and cannot be controlled (like usually customer demand and many others). The (dependent) state variables $\overline{x}$ concern those quantities that result from the values of $\bar{i}$ and $\overline{u}$ because of physical relations. The values of these variables are obtained from the power flow simulation.
Since a power flow controller obviously aims at providing control actions with respect to information from its environment, it would be obvious to define a desirable policy to be a function $p(\bar{i})$. In first considerations, $\bar{i}$ is assumed to comprise system variables from the power grid's model that are important for power flow decisions - this is the aim of so called *system variable synthesis*[2].

In order to make the idea of system variable synthesis (SVS) clearer, it should be related to the OPF problem within a specific power grid model, namely the IEEE 14-bus test case [21].
Figure 4.2 shows the layout of this grid model, where the 5 generator buses are indicated with G, arrows represent loads. The slack bus (needed for power flow simulation) is shown in black. For this model, we now want to define the input variables that a generator would need to consider in order to control its output variable P_G with the objective of system-wide cost minimization (as in traditional OPF), i.e. the real power injection of the unit. For making a valid action on P_G, a policy would need to take the following inputs:

- Real and reactive load values of all buses; $P_{L_1}...P_{L_{NB}}, Q_{L_1}...P_{L_{NB}}$ for $NB = 14$,
- Real and reactive generation limits of all generators; $P_{G_1}^{max}...P_{G_{NG}}^{max}$, $Q_{G_1}^{max}...Q_{G_{NG}}^{max}$ for $NG = 5$,
- Polynomial cost coefficients of all generators; $a_1...a_{NG}, b_1...b_{NG}, b_1...b_{NG}$ for $NG = 5$,
- Real power flow limits of all branches $P_{b_1}^{max}...P_{b_{Nb}}^{max}$ for $Nb = 20$.

Thus, the vector of needed input variables that the final policy $P_G(\bar{i})$ takes is:

[2]In this work, the term synthesis specifies the process of combining complex decision policies out of single information entities (such as system variables).

$$i^T = [P_{L_1}...P_{L_{NB}}, Q_{L_1}...Q_{L_{NB}}, P_{G_1}^{max}...P_{G_{NG}}^{max}, Q_{G_1}^{max}...Q_{G_{NG}}^{max}, P_{b_1}^{max}...P_{b_{Nb}}^{max},$$
$$a_1...a_{NG}, b_1...b_{NG}, c_1...c_{NG}].$$

In Figure 4.2, these variables are illustrated exemplarily for two buses, while one of these two is a generator bus.

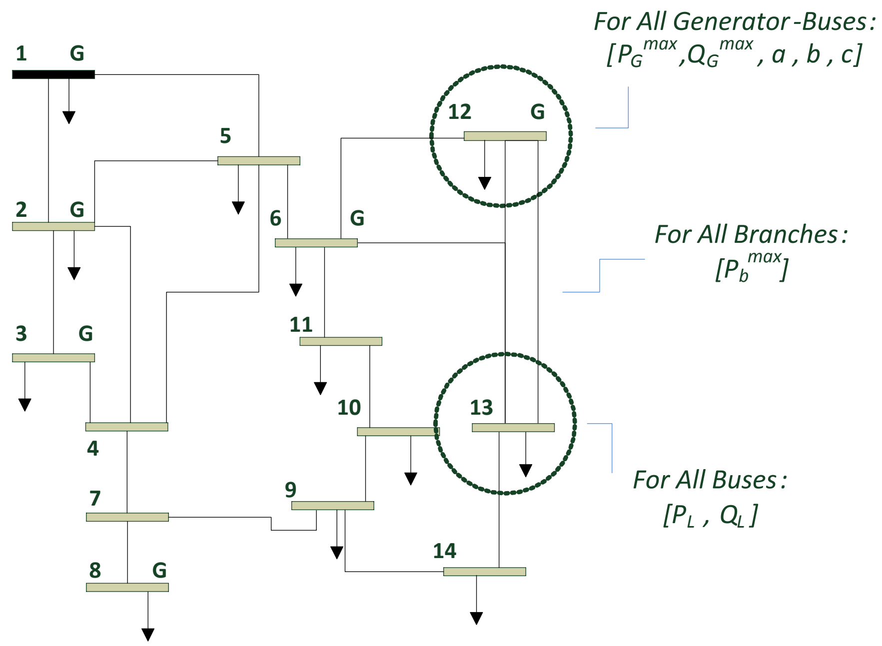

Figure 4.2: 14-Bus Test Case Layout for System Variable Synthesis

Finally, for system variable synthesis, the optimization problem aims at finding optimally performing policies $p(\bar{i})$ for all controllable units in the system. Considering real power injection as the only controllable unit for simplicity reasons, for all $NG - 1$[3] generators[4] a policy $P_G(\bar{i})$ needs to be computed.

[3]When doing power flow computation, usually NG-1 generators are controllable, while the slack-bus generator is non-controllable and gets adapted at the end of the (iterative) computation in order to meet the resulting power losses in the system.

[4]Note that lower case p is generally used for a policy, while P in this case addresses a policy for the variable P_G- real-power injection.

Deficiencies of System Variable Synthesis

Synthesizing policies directly out of system variables implies some disadvantages. On the one hand, a comparatively high amount of variables has to be considered when deriving the final policies, which makes the respective optimization process a quite hard computational problem during the learning phase. Taking variables from each branch and each bus in the system as input lets the dimensionality of the solution space grow with the size of the considered power grid (see also Section 5.5 for explicit discussions on that). On the other hand, each decision-making device in the system (i.e. generation unit, transformer, etc.) requires a proper policy that has to be learned, while all policies are indeed interrelated and have to be learned synchronously. Both facts increase the problem complexity proportionally with the size of the power grid model and thus handicap the scalability of this approach significantly.

Taking for example the considered 14-bus test case, $NG - 1 = 4$ generators need to be controlled each with two variables (P_G, V_G), and additionally 3 transformer tap changers and 1 VAR compensator are available as controllable units, yielding $(NG - 1$ [5]$)+NG + NQ + NT = 4 + 5 + 3 + 1 = 13$ policies that would need to be evolved synchronously[6]. Taking exemplarily a medium-sized 57-bus test case as used for experimental investigations in Chapter 5, for all controllable units in sum $6+7+3+17 = 33$ policies would need to be learned all at a time. For the hardest problem instance that will be treated, the 300-bus test case, even $68+69+8+107 = 252$ policies would be necessary.

Hence, it is more appropriate to build an approach that is more scalable, being less dependent on considered power grid's size. Such an approach can be built by abstracting information from the input variable vector $\bar{i}$ in such a way, that this information is invariant from the power grid's size while being generic to the decision-making units, but still provides the finally derived power flow controller with sufficient data.

[5]While the reference voltage V_G is controlled for all NG generators, P_G is controllable for all generators except the slack-bus.

[6]Since these variables are interrelated, it is not possible to optimize the policies one-by-one; hence, they have to be co-optimized in parallel.

4.2.5 Policy Design: Abstract Rule Synthesis

We assume that it is not necessary for the policy to consider the whole set of variables $\bar{i}$, but to only use abstract metrics that are important for power flow decisions. Thus, for these metrics so called "abstract rules" are introduced, that gather necessary information from the system's state, supposing that this information is sufficient for achieving valid and accurate actions. All abstract rules for dynamic OPF were generated based on domain knowledge in the context of this work [46, 55] and are shown in Table 4.1. Thus, some kind of feature processing is applied based on expert knowledge for reducing the set of input variables $\bar{i}$ to a set of abstract rules $\bar{r}$, where $|\bar{r}| << |\bar{i}|$ holds for most systems.
These rules take the general information $\bar{i}$ from the system's state and output a specific information quantity that is tailored to the considered unit (for instance a specific generator). For example, at a specific situation, the neighboring load factor NLF for the generator at bus number 12 is different from its value at bus number 6, as these buses have distinct neighborhoods. The same holds for the other rules as well.

This set of rules is considered to sufficiently describe the power grid's state in order to derive accurate actions for power flow control; the experimental validation will be performed later in Chapter 5. Here, the third column indicates which rules are necessary for learning a policy of a given variable. For example, the policy for variable P_G of a generator bus is a function

$$P_G(LLF, NLF, GLF, MARF, MERF, LCCF, QCCF).$$

For further details, the implementation of this set of rules is shown in Appendix A.1, written in Matlab notation based on MATPOWER data structures.
The application of abstract rules instead of using all input variables $\bar{i}$ is twofold: First, the cardinality of $\bar{r}$ is much lower for most power grid models than the number of relevant input variables which is advantageous to the optimization. Second, the rules are completely independent of the grid's topology, making this approach applicable to any distribution or transmission grid model in a generic way. For example, the rule GLF makes a controllable unit considering the global load situation, but instead of considering all load values $P_{L_1}...P_{L_{NB}}$ throughout all buses in the system as with system variable synthesis (SVS), only one single value (namely the output of the rule GLF) has to finally be integrated into the structure of the policy, making this approach scalable and generic.

Rule	**Explanation**	**Variable**
LLF	Local Load Factor: active load at bus divided by maximum active power injection at bus	P_G
NLF	Neighboring Load Factor: sum of active load at directly connected buses and their neighbors divided by maximum active power injection at those buses	P_G, V_G, Q_C, T_{tap}
GLF	Global Load Factor: sum of total active load in grid divided by sum of maximum active power generation	P_G, V_G, Q_C, T_{tap}
MARF	Max Rating Factor: maximum rating (power flow) of connected branches divided by maximum rating of all branches	P_G, Q_C
MERF	Mean Rating Factor: mean rating (power flow) of connected branches divided by maximum rating of all branches	P_G, Q_C
LCCF	Linear Cost Coefficient: linear cost coefficient of generator divided by maximum linear cost coefficient of all generators	P_G
QCCF	Quadratic Cost Coefficient: quadratic cost coefficient of generator divided by maximum quadratic cost coefficient of all generators	P_G
NRLF	Neighboring Reactive Load Factor: sum of reactive load at directly connected buses and their neighbours devided by maximum reactive power output at those buses	V_G, Q_C, T_{tap}
GRLF	Global Reactive Load Factor: sum of total reactive load in grid divided by sum of maximum reactive power output	V_G, Q_C, T_{tap}

Table 4.1: Abstract Rules

Having the set of abstract rules $\overline{r}$, the policy $p(\overline{r})$ has to be computed. Therefore, some mathematical structure needs to be assumed which gets optimized by evolutionary algorithms. Since rules are normalized (by division with the maximum value in each rule), they can be integrated into any given mathematical structure.
For each considered class of controllable units, a policy is principally the same (like for the real power injection P_G of all generators in the system), but using information about the unit's specific situation through its abstract rules, it derives individual actions for each unit. Referring to the 14-bus case as discussed before, instead of $NG - 1$ policies for variable P_G for all generators, only one flexible policy needs to be learned. Since this holds for all considered output variables, only 4 policies $(P_G(\overline{r}), V_G(\overline{r}), Q_C(\overline{r}), T_{tap}(\overline{r}))$ would need to be learned instead of 13. For bigger models, this number of needed policies keeps constant, making this approach suitable for scalable technologies.

Further discussions on abstract rule synthesis (ARS) will be performed when applying it to practical benchmark problems for building a dynamic and stochastic optimal power flow controller in Chapter 5. Before, some theoretic considerations have to be conducted concerning the policy approximation.

4.2.6 A Learning Approach

The determination of policies in order derive optimal power flow control clearly describes a learning procedure, where the aim is to learn the policy with the best estimated performance in future operation. This procedure is hard to fit into traditional learning schemes. On the one hand, it cannot be defined being related to reinforcement learning, since its aim is not to find a defined sequence of actions (from a finite set of actions) that have to be taken. The herein considered policies form continuous functions that output real-valued control actions over time, independent of a certain sequence. To a greater degree, the proposed policy learning scheme fits into traditional supervised learning considerations, where the finally best found policy optimizes some kind of reward (like financial costs of power supply) over a simulated time span, rather than any error function from a given set of samples/data points. Even if learning is based on the simulation-evaluation of candidate solutions rather than any error function from samples, basic principles from supervised machine learning hold as well and need to be considered.

Overfitting vs. Underfitting

Let us assume that we want to find a policy $p(\bar{i}, \overline{w})$, with $\bar{i}$ containing the input information from the system and $\overline{w}$ defining the set of optimal parameters to this policy. The aim of the learning procedure is to optimize the reward (or value of the cost function) $C(p(\bar{i}, \overline{w}))$ within the simulation model. Considering principles of supervised machine learning, the aim is not to optimize the reward with respect to the given training samples, but to optimize the estimated reward for future samples (see also empirical risk minimization vs. structural risk minimization [32, 105]). The analogue definition holds here as well, where a policy's reward has to be optimized for future operation (rather than for the actual simulation model), thus $C(p(\bar{i}', \overline{w}))$ needs to be optimized rather than $C(p(\bar{i}, \overline{w}))$, with $\bar{i}'$ describing the future inputs. Thus, one has to avoid the risk of "overfitting" the policy, i.e. build an accurate policy $p(\bar{i}, \overline{w})$, that only performs well within the actual simulation model but generalizes poor during future operation. This concern is strongly related to choosing an appropriate complexity of the policy (model complexity in traditional supervised learning), where a too high complexity tends to overfit while a too low complexity is not capable of identifying a system's complex interrelations for making accurate decisions and thus is "underfitted". Further discussions on supervised learning theory shall be obtained from literature [32, 41, 105], the described outline is assumed to be sufficient at this point.

4.2.7 Metamodel-Based Synthesis

So far we did not discuss the concrete structure of a control policy p. The issue of choosing an accurate structure is related to defining a suitable model class in supervised learning, which strongly influences the risk of overfitting/underfitting. Within this work, two principal approaches can be distinguished for deriving a policy with evolutionary computation.

First, fixed mathematical structures (i.e. metamodels) are used for combining input values (such as abstract rules) and thus synthesizing the final policy out of them, using fixed-order polynomials. Realizing for instance the linear combination of rules according to Equation 4.3 (exemplarily shown for variable P_G), the set of weights $\overline{w}$ has to be optimized when learning the policy for P_G, with NR being the cardinality of rules $\bar{r}$, n indicating the index of the controllable unit. c additionally represents a constant that is evolved during the optimization process as well[7]. Using the second representation as

[7]This constant is necessary for building accurate policies and can be interpreted as analogy to the *bias* within the simple linear perceptron model [91] respectively within

shown in Equation 4.4, policies are combined using polynomials of degree 2, thus, for each rule two weights ($nr, 1$ and $nr, 2$) are learned. This principle can be extended to arbitrary polynomials (or even other expressions).

$$P_{G_n} = \frac{\sum_{nr=1}^{NR} r_{nr,n} * w_{nr}}{NR} + c \tag{4.3}$$

$$P_{G_n} = \frac{\sum_{nr=1}^{NR} r_{nr,n} * w_{nr,1} + r_{nr,n}^2 * w_{nr,2}}{NR} + c \tag{4.4}$$

However, polynomials of arbitrary degree (or any other generic function) could be assumed to be the metamodel. But while increasing the polynomial's degree, the model-complexity increases too and leads to a higher risk of overfitting. Hence, low polynomial degrees need to be considered.

Assuming the following policies to be optimized for all variables:

$$\begin{aligned}
&P_G(w_{LLF,P}, w_{NLF,P}, w_{GLF,P}, w_{MARF,P}, w_{MERF,P}, w_{LCCF,P}, w_{QCCF,P}, c_P),\\
&V_G(w_{NLF,V}, w_{GLF,V}, w_{NRLF,V}, w_{GRLF,V}, c_V),\\
&Q_C(w_{NLF,Q}, w_{GLF,Q}, w_{MARF,Q}, w_{MERF,Q}, w_{NRLF,Q}, w_{GRLF,Q}, c_Q),\\
&T_{tap}(w_{NLF,T}, w_{GLF,T}, w_{NRLF,T}, w_{GRLF,T}, c_T),
\end{aligned}$$

21 real-valued weights contained by ($\overline{w}$) and 4 constants in ($\overline{c}$) would need to be optimized, yielding 25 control variables for the exemplary linear synthesis in case of OPF as defined before (46 for the polynomial synthesis). In order to satisfy bounding constraints for the variables P_G, V_G, Q_C and T_{tap}, the output of the respective policy is multiplied by the variable's maximum value while guaranteeing that this value is not exceeded. Therefore, each rule needs to be designed in order to output a value in the range $[0, 1]$.
The optimization of such metamodels requires a real-valued vector as solution representation, which is evaluated through simulation for expressing its fitness value. Figure 4.3 describes the decomposition of the real-valued solution vector into single policies for OPF control.
The solution vector in this case is created by concatenating all weights and constants of each policy accordingly, and can easily be decomposed when evaluating a solution.

---
artificial neural networks (multilayer perceptrons) [16].

$$[\overbrace{w_{LLF}\dots w_{QCCF}, c_P}^{P(\bar{r},\overline{w},c)}, \overbrace{w_{NLF}\dots w_{GRLF}, c_V}^{V(\bar{r},\overline{w},c)}, \overbrace{w_{NLF}\dots w_{GRLF}, c_Q}^{Q(\bar{r},\overline{w},c)}, \overbrace{w_{NLF}\dots w_{GRLF}, c_T}^{T(\bar{r},\overline{w},c)}]$$

Figure 4.3: Decomposition of the Solution Vector Into Policies

4.2.8 Genetic Programming: Metamodel-Less Synthesis

Even if the rule synthesis using a fixed-order polynomial is quite intuitive and will lead to competitive results for certain applications (like shown in Chapter 5), it seems to be inflexible, being unable to approximate any nonlinear behavior that may occur in real-world. Thus, this approach may tend to underfit the problem for very complex systems. Therefore, a second approach is introduced that allows a more flexible combination of abstract rules without the need of assuming a metamodel, namely genetic programming (GP) [6].

Introduction to Genetic Programming

Extending the principle concept of genetic algorithms, GP uses evolutionary-inspired concepts for the heuristic search process, but is able to evolve computer programs. Within this work, these computer programs take the appearance of trees, where leafs represent inputs (rules) as defined before, that are combined by arbitrary mathematical operators which are incorporated by inner nodes. This kind of solution representation allows arbitrary mathematical combinations of abstract rules, where the applied operators for inner nodes are defined by the used grammar to a final decision-making policy.

To give some overview on GP, finding first research activities in the 1980s, the computationally expensive concept of GP was pushed essentially by the steady increase of computational power in the last two decades. One of the most important publications in this field was provided by Koza [62], stating GP as automated invention machine for numerous practical applications like the artificial ant problem or later applications of symbolic regression [6], to name the most popular ones, [66] finally provides profound analysis of GP in the context of GA schema analysis. This ability of GP to automatically construct new solutions (programs) to a given problem is enabled by its special kind of solution representation, that is not restricted to a fixed structure (like fixed-length one-dimensional array as in standard GA), but forms a hierarchical computer program of variable length, consisting of functions and terminals.

Synthesizing Policies With Genetic Programming

Figure 4.4 gives an exemplary tree (GP solution) that could represent a policy for variable P within the OPF problem. Here, inner nodes that represent functions are indicated in dotted style, while terminal nodes using either rules or constants are plotted in solid style. In this case, the policy would consider the load situation at neighboring buses (NLF), the global load situation (GLF) as well as the linear costs of the respective generation unit (LCF). Out of this mathematical combination, finally a numerical value is derived that represents a decision for the unit's real power output.

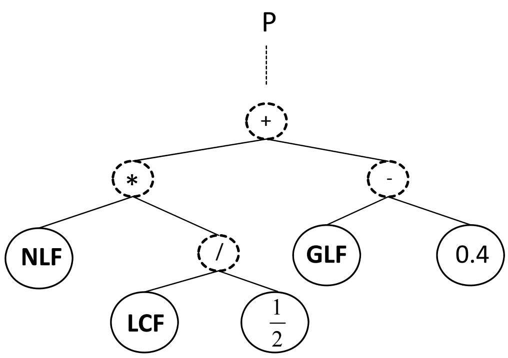

Figure 4.4: Exemplary Policy with GP

The great advantage of this kind of flexible solution representation compared to the application of a fixed-order polynomial is that GP is able to find complex mathematical coherences between rules with variable length. Hence, it enables the search for performant policies within a solution space of both variant shape and size. Using any arbitrary combination of mathematical operators as inner nodes (grammar), the degree of freedom for finding performant policies gets increased drastically. Moreover, since GP is not constrained to use the whole set of rules for solution creation, simpler policies can be found too. Here, an implicit feature selection happens during the genetic search, where specific rules that where used within successful solution candidates are more likely to finally build the best found solution (see schema theorem analysis [6]). In the end, the disadvantage of policy synthesis with

GP is that the possible solution space increases with the cardinality of the applied grammar. For overcoming this problem and pruning the solution space, the possible grammar is restricted to a set of operators. Grammars in this case may include arbitrary algebraic operators like arithmetic operators, exponential, sinoidal and other functions, and even logical and conditional operators. Concretely applied grammars will be discussed when performing experimental studies later.
Beside all these facts, one feature of applying GP remains fundamental: namely the ability of evolving policies independently from any predefined mathematical structure, hence, without the need of a-priori knowledge of the policy. This ability is a fundamental enabler for policy-based control in real-world problems and clearly distinguishes this work from related approaches in literature.

4.2.9 Co-Optimization of Interdependent Policies

When evolving policies for OPF control, the issue arises that multiple units will have to be controlled that need proper policies, while these are interrelated. Considering for instance the problem formulation of OPF policy optimization, the four policies P_G, V_G, Q_C and T_{tap} are indeed interrelated (for instance it is obvious, that the voltage control performed by the tap changer T_{tap} depends on the voltage control V_G of the generator units). Thus, it is not sufficient to evolve these policies one-by-one independently (i.e. treating their optimization as partial problems), but they have to be evolved all at a time, preserving their interrelationships during the optimization process.

When applying metamodel-based policy synthesis as discussed in Section 4.2.7, this issue is less problematic since the respective solution representation of a real-valued vector contains the control variables for all four policies, which gets decomposed into the single policies according to Figure 4.3 as discussed before.
For synthesis with GP, this issue is a little more tricky. Here, it is not useful to put all four policies (i.e. solutions in the form of trees) into one solution representation, since their grammar is different[8] which would cause genetic operators being highly inefficient. In such a case, it is more appropriate to consider the different policies as distinct species that have to be evolved while taking care of their relationships, necessitating a coevolution-related scheme[9].

[8]Note that these policies take different abstract rules, i.e. different terminals.

[9]Another valid approach would be to put all 4 trees into one representation (vector), but to perform some repair after each crossover in order to fix invalid branches. This alternative method is not considered herein.

Coevolution is often essential when considering real-world problems. Such problems naturally consist of distributed entities that have own behavior but a global goal needs to be achieved as result of their interactions, hence, coevolution is strongly related to multi-agent systems [43, 85]. Here, coevolutionary algorithms have emerged that use parallelization techniques for evolving disjunct subpopulations of different species, such as with coevolutionary genetic algorithms.
Originally, in evolutionary computation two main parallelization schemes have emerged, namely island models and diffusion models [6]. However, these models evolve populations of a single species, while individuals of this species exist within different subpopulations. Contrary, the aim of coevolutionary GA is to realize the coexistence of several species (in this case different control variables) that aim at a common goal (such as minimization of generation costs). Another distinction has to be made between cooperative and competitive coevolution [84, 92], while in this case clearly a cooperative scheme lies on hand.

Coevolutionary Genetic Programming

Numerous approaches for coevolutionary genetic algorithms have been developed in recent years, many of them realizing multi-agent or game-theoretic approaches for matching parallelized populations [35]. In the context of this work, a method has been developed (also proposed in [55]) using a more straight-forward coevolutionary scheme, where a global fitness function is shared along parallely executed genetic programming processes. The idea is that q subpopulations $X_1...X_q$ are evolved within separate processes of same population size z, where evolutionary operations (mutation, parent selection, recombination) are applied separately and independently from each other. Each process evolves one of m policies using its individually defined grammar. The only information that the processes share is the common fitness of individuals belonging to the same solution. Thus, a complete solution X consists of q partial solutions (policies), i.e. $X = \{X_1, X_2, ...X_q\}$. This principle is shown Figure 4.5. When evaluating a solution candidate X, all policies that belong to X serve as input to the simulation model. After computing the fitness function value, it is shared along all partial solutions.
While all other genetic operators are executed locally, survivor selection happens globally. Being realized by a proportional selection scheme, complete solutions (such as a row $\{X_{1.1}, X_{2.1}, ..., X_{q.1}\}$ in Figure 4.5) are selected for replacement, rather than making this selection within each process indepen-

dently. This is important regarding the nature of coevolutionary systems, where the fitness of a partial solution is always an objective (global) measure depending on other partial solutions, rather than a subjective measure.

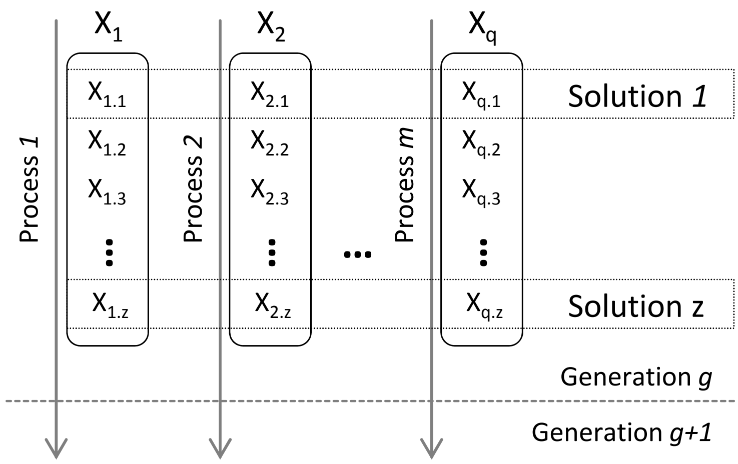

Figure 4.5: Process Model for Coevolutionary GP

In the herein case, $q = 4$ holds for the OPF policies, hence, $X = \{P_G, V_G, Q_C, T_{tap}\}$. For realization, HeuristicLab is chosen offering sophisticated implementations for genetic programming. Here, 4 separate processes are executed where fitness values are shared using MPI (Message Passing Interface) in Windows OS.

4.2.10 Discussions on Simulation-Based Evolutionary Policy Approximation

Section 4.2.1 already provided some overview on existing approaches for the optimization in dynamic stochastic problems, namely a class of methods called *approximate dynamic programming* (ADP). Now, some comparative discussion on the proposed simulation-based evolutionary policy approximation shall be provided both with respect to the general theory on optimal control with dynamic programming according to Bellman's optimality equation as well as especially in the context of ADP.

Dynamic programming applies the fundamental idea of Bellman's optimality principle, that a dynamic optimization problem is considered as a multi-stage problem with $t = 1...T$ time steps. While a state $J(t+1)$ is reached via applying an action u_t to the state $J(t)$, the sequence $\overline{u} = [u_1...u_T]$ needs to be found that optimizes the (estimated) objective function value at the end. Here, Bellman's optimality principle states that an optimal sequence of actions - whatever the initial state $J(1)$ and initial decision u_1 are - has to constitute an optimal sequence regarding the state resulting from the first decision [13]. Taking the herein notations for policy approximation, Bellman's optimality equation in form of a recursive definition of the policy function can be stated as:

$$p(J_t) = \max_{\overline{u} \in \mathbb{R}} \left(C_t(J_t, u_t) + E[p(J_{t+1})|J_t]\right), \tag{4.5}$$

where the costs of a policy's action at the beginning $C_t(J_t, u_t)$ are separated from the estimated costs of the policy in the future $E[p(J_{t+1})|J_t]$.

When handling such sequential decision problems via dynamic programming in practice, the common approach is to apply some state-space discretization (as described before in Section 3.2) where the system is discretized through a set of states that are linked with both actions as well as transition probabilities. Throughout this set of states, the aim is finally to obtain the sequence of actions that fulfills the bellman optimality equation. This discrete-state modeling via so called Markov Decision Processes [38] and the subsequent search for an optimal sequence of actions within the Markov model is capable of tackling dynamic and uncertain systems, but raises a major problem: the *curses of dimensionality* [38, 87].
This issue is twofold: on the one hand, the number of states may get very high for real-world applications (no matter if a combinatorial or a real-valued decision problem is considered). On the other hand, the number of possible actions that link the interconnected states grows fast as well. Hence, the *curses of dimensionality* concern both the dimension of the state-space as well as the dimension of the action-space.

In this context, the previously discussed class of approximate dynamic programming (ADP) methods has evolved, major achievements in this field have been mentioned in Section 4.2.1. ADP aims at overcoming the curses of dimensionality by not considering a complete model of the state-action space, but rely on approximations of it, making this technology attractive to real-world dynamic stochastic optimization problems. However, even if performing approximations, ADP still necessitates the description of the system via state-action pairs and related transition probabilities, hence, still needs the transformation of the real-world system to some approximated Markov Decision Process where dynamic programming can be applied to for finding an optimal sequence of actions.

Contrary to this idea, simulation-based evolutionary policy approximation proposes a more straight-forward way of tackling such problems. While the optimization is performed directly in a multi-stage simulation model, where the discrete states of this model result from probabilistic simulation, the formulation of some Markov model is not needed. This principle makes it more suitable to real-world problems, where a simulation model is often already available or even can be built with reasonable effort.
Additionally, contrary to ADP the aim is not to find a defined sequence of actions that has to be taken starting at some fixed time step t, but to evolve a general policy that outputs near-optimal actions in whatever state of the system it is applied to. Hence, instead of traversing some action-space for searching an optimal sequence of actions, the policy-space $\mathbb{P}$ gets traversed. While $\mathbb{P}$ (as indicated in Equation 4.2) may indeed get very complex for real-world problems, a directed search via evolutionary algorithms is applied that finds good policies in reasonable time.

4.2.11 Summarizing the Developed Techniques

Manifold scenarios in smart power grid engineering challenge optimization methods that are capable of handling stochastic systems successfully. While stochasticity is one feature that characterizes power grid control and planning problems, the necessity of dealing with dynamic conditions and making control actions in an anticipatory manner is of higher interest.
Especially for such dynamic stochastic optimal power flow (DSOPF) problems, the actual literature provides only few approaches, while the general interest in solving such problems is high. Policy approximation has been identified herein as potential concept, where a set of requirements was defined that would need to be satisfied by a technology that makes this concept suitable to DSOPF problems in future smart grids. Simulation-based evolutionary optimization of flexible control policies was developed that is able to fulfill these requirements. Here, the application of simulation as system representation allows to fully integrate the complex as well as stochastic behavior of power grids into the optimization process. Furthermore, the method of learning/approximating control policies offline that later provide flexible and valid control actions during operation online builds a suitable class of techniques for approximate optimal control, which avoids the need of any

reoptimization at runtime.
Building a technique for dynamic optimization, contrary to related approaches in literature, the application of simulation-optimization avoids the need of building some system representation such as Markov Decisions Processes. In this way, the curses of dimensionality of dynamic programming problems get overcome and eases the application to real-world problems. Especially the application of GP allows the evolution of policies without a-priori knowledge of their structure, which is a powerful ability when treating complex systems. It is important to mention at this point that the herein used mathematical structures for policy synthesis (i.e. metamodel-based synthesis with polynomials and metamodel-less synthesis with GP) only build a subset of possible methods. Any other generic function approximators like for instance artificial neural networks could be fruitful as well, but will not be treated herein. The next section presents experimental applications of the developed techniques to practical problems in power grid planning and control. Both static as well as dynamic problems will be treated with respect to stochastic considerations, which founds core findings for later applications to smart grid control.

Chapter 5

Simulation Optimization in Electric Power Grid Operation and Planning

In the previous chapter, fundamentals on simulation optimization have been discussed, both for static as well as dynamic stochastic optimization problems. A new technique has been developed - namely the simulation-based evolutionary approximation of power flow control policies - which is capable of enabling dynamic stochastic optimal power flows. The reader shall now be provided with practical show cases in order to foster the understanding of evolutionary simulation optimization. Experimental studies will show the validity of the developed techniques, both for static as well as dynamic problem classes.

5.1 Static Deterministic Benchmark Optimization - Proof of Concept

While all following empirical studies will treat multi-period power flow considerations, benchmarks exist in literature for the class of steady-state optimal power flow problems. These benchmarks provide deterministic models and are used in literature since many years for validation of optimization methods. In order to experimentally show the principal capabilities of metaheuristic simulation optimization, these benchmarks shall now be treated with comparison to reference solutions.

5.1.1 Formal Problem Description

The definition of the general OPF problem has already been stated in Section 3.4.2, respectively in Equations 3.2 - 3.13. This definition remains the same, where the objective is to minimize costs of power supply over all available generation units while satisfying constraints of power grid operation. Therefore, the vector of control variables $\overline{u}$ will be computed statically for a steady-state situation that is given by the benchmark descriptions. As in the OPF definition, the solution contains values for generator voltages V_G, generator real power outputs P_G, transformer tap settings T_{tap} and the output of shunt VAR compensators Q_C. Since constraints are considered within the heuristic search through penalization, Equation 5.1 gives the final fitness function. Here the objective function $C(P_G(\overline{u}))$ denotes the supply costs from the generation units that in turn depend on the control variables. While constraint violation is penalized by adding a penalty term to the objective function value, the vector $\overline{w_{CV}}$ gives the relative weight of each constraint, where its violation is contained by $\overline{CV}$.

$$F(\overline{u}) = C(P_G(\overline{u})) + \overline{w_{CV}} * \overline{CV(\overline{u})} \tag{5.1}$$

5.1.2 Experiments Description

Throughout the different benchmarks, both the model's size (in means of dependent variables) as well as the amount of control variables vary.
A set of five different quasi-standard OPF benchmarks exists, where distribution grid models as well as generator cost data are given in the IEEE test case archive [21], published by Richard D. Christie in IEEE Common Data Format [20, 39] notation.

Distribution Grid Model Data

Table 5.1 characterizes these steady-state load flow models that build the fundament of later experimental investigations within this work by showing their key figures. Concerning structural properties of these test cases, not only various complexities in means of system size (and thus computation time for load flow simulations), but also different sets of control variables exist (see lines 3-6 in Table 5.1), building a suitable set for also proving the scalability of the developed techniques in subsequent discussions.

Property	14-Bus	30-Bus	57-Bus	118-Bus	300-Bus
# Buses	14	30	57	118	300
# Branches	20	41	80	186	411
# Generators	5	6	7	54	69
# Tap Changers	3	2	3	9	8
# VAR Compensators	1	4	17	12	107

Table 5.1: IEEE Power Flow Benchmark Problems

The smallest two cases are very well investigated, providing lots of solutions to their steady-state OPF in literature. Very popular works are published by Alsac and Stott [8] when solving the traditional OPF for the 30-Bus system in an analytic way. Many other researchers did various extensions in means of different objective functions or constraint formulations when applying metaheuristic algorithms [1, 64]. All used constraints data not comprised by the test case archive can be obtained from Appendix A.2, given in MATPOWER data format.

Algorithmic Settings

For optimizing the real-valued solution vector $\overline{u}$, evolution strategies (ES) according to [14] will be used, being proved to be performant metaheuristic optimization algorithms for real-valued optimization problems. In principle, ES is a nature-inspired population-based optimization algorithm, that tries to improve a set of solution candidates until a certain stopping criterion is reached. Contrary to other related evolutionary algorithms, ES selects (deterministically) the best μ individuals within each generation out of λ offspring individuals and tries to improve them where mutation is the main evolutionary operator, hence the primary source of genetic variation. ES is generally proven to be a powerful and efficient metaheuristic algorithm for real-valued optimization problems, supplied by its special ability of self-adaptiveness within the search process.
For the herein experiments, $(\mu + \lambda)$-ES is applied with one endogenous strategy parameter τ that adapts the mutation strength dynamically (as implemented in HeuristicLab).

The following settings given in Table 5.2 are used when treating the different benchmark instances.

Parameter	Value
ES Type	14-Bus and 30-Bus Cases: (5+20) 57-, 118- and 300-Bus Cases: (10+50)
Manipulator	SelfAdaptiveNormalAllPositions- Manipulator
Strategy Parameter	Learning Rate $\tau = 0.5$
Recombinator	Discrete Crossover
Parents per Child	2
Stopping Criterium	Maximum Generations: 10000

Table 5.2: Parameters for Benchmark Cases

5.1.3 Experimental Results & Conclusion

Within these 5 benchmark instances, the OPF solution gets computed with evolutionary simulation optimization. The distribution grid models are implemented in MATPOWER. When evaluating a solution candidate, power flow simulation is performed within MATPOWER in order to derive the solutions' fitness function value (consisting of both objective function value and constraints penalization), see Algorithm 1 for further details.
At this point it is important to mention, that these benchmark instances represent steady-state problems that are given deterministically. Hence, neither stochastic nor dynamic optimal power flows are treated. Rather, these experiments shall proof the general capability of evolutionary simulation optimization for the general OPF computation, since deterministic exact solvers are available here for comparison reasons.

Algorithm 1 Calculate fitness of steady-state OPF solution $\overline{u}$

$initializeGridModel()$;
$setControlVariables(solutionCandidate)$; {// for V_G, P_G, T_{tap} and Q_C}
$performPowerFlowComputation()$;
$fitness \leftarrow (powerLosses + constraintsPenalty)$;

In order to validate reachable solution qualities, a primal dual interior point method as implemented in MATPOWER is taken for creating reference solutions. This method can be seen as standard solver for static OPF problems both in industry as well as in academia.

Figure 5.1 illustrates boxplots[1] where for each instance the 10 best found

[1]The red line gives the median, the borders of the box represent the 25th and 75th percentiles, the whiskers reach to most extreme data points (not considered outliers),

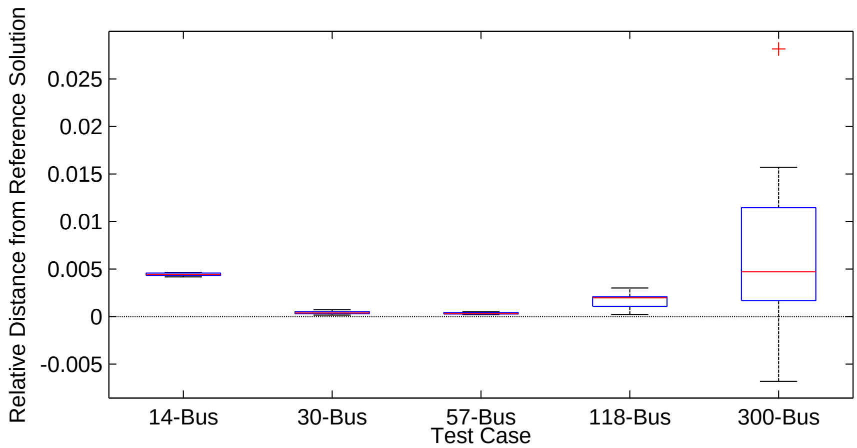

Figure 5.1: Simulation-Based OPF: Distribution of Best Found Solutions' Fitness Values

solutions' fitnesses are given (by their relative distance to the reference solution), in order to not only highlight the best obtained solution for these problems, but also the heuristic optimization's robustness.

For most cases, the evolutionary optimization robustly finds near-optimal solutions that are below 0.5% worse in means of fitness function from the reference solution. Only for the largest case - the 300-bus case - the heuristic search is less robust, but even finds solutions that are better than the reference. This is possible because of the implementation of the the interior-point method in MATPOWER, which uses a linearized approximate model of the non-linear power flow equations for optimization, while the simulation optimization uses the actual power flow simulation model without approximations. Hence, the simulation-based method even is able to find better solutions, since it does not need for any simplification of the problem. Especially for very complex power flow models this is an important advantage, where linearized approximations may describe the system insufficiently.

The results from this boxplot are concluded numerically in Table 5.3. Here, the median relative distance corresponds to the distance from the reference solution to the median fitness of the 10 best found ones, while the

while outliers are indicated individually (red plus symbols). The whisker length is defined to be 1.5, which corresponds to immediately $\pm 2.7\sigma$ and 99.3% coverage if the data is normally distributed.

minimum relative distance addresses the single best found solution.

	14-Bus	**30-Bus**	**57-Bus**	**118-Bus**	**300-Bus**
Best Found Solution's Fitness	8185.4	802.77	41781.0	129690.0	535070.0
Reference Solution's Fitness	8151.46	802.66	41772.40	129661.0	538743.0
Median Relative Difference	0.0044	0.00036	0.00030	0.0020	0.0047
Min Relative Difference	0.0042	0.00014	0.00021	0.00022	-0.0068

Table 5.3: Results to Static IEEE OPF Benchmark Optimization, Fitness Function Values in [$/h] Operation Costs

Since - as mentioned above - for the two smallest test cases (14-Bus and 30-Bus) lots of solutions to the OPF exist in literature, the best found solutions herein are given in Appendix A.3. The interested reader may compare these with related solutions, for example from [1, 8, 64].

Concluding Remarks

These experiments were conducted in order to provide some proof-of-concept that evolutionary simulation optimization is able to supply competitive solutions compared to state-of-the-art OPF solvers. While for most cases the evolutionary simulation optimization finds near-optimal solutions that are only some 0.1% worse than the reference solutions, it is even possible to find better solutions than the state-of-the-art OPF solver for the largest test case. In these experiments, still the originally formulated steady-state OPF problem was treated, being a central optimization problem in power grid engineering and research since many decades. In the following chapters, the OPF problem will be extended in order to consider real-world relevant issues in future power grids. These extensions will consider multiperiod-formulations, the inclusion of uncertainty as well as the integration of numerous distributed controllable units. While applying herein developed optimization concepts to these extended formulations, new methods for power flow optimization in smart electric grids get discussed.

5.2 Static Stochastic Optimization Problems - Plant Placement Under Uncertainty

Coming from the general OPF as a static (parametric) and deterministic optimization problem, an extension of this problem shall now enable the treatment of a probabilistic planning scenario in power grids. Therefore, a combinatorial problem will be handled, that concerns the placement of distributed renewable power plants under uncertainty - a challenging issue in smart grid engineering.

One of the most popular planning problems in actual developments considers the placement of renewable sources like wind power and photovoltaic plants [7, 56, 59]. While these sources have the advantage of delivering zero-emission electric power, they may be dangerous to existing grid infrastructures because of their intermittent and non-deterministic behavior, jeopardizing security constraints of power grid operation. As placement of these sources influences the reachable penetration of renewable supply, optimization problems are formulated for obtaining placement decisions. A thorough overview on this problem, possible objectives as well as respective approaches for solution in literature can be found in [59, 75].
The formulation of this infrastructure planning issue is based on the well known IEEE 118-bus test-case (see Table 5.1). The test-case is downscaled by a factor of 10 in means of generation as well as load values. Within this distribution grid, 10% of the installed generation capacity should be substituted by renewable supply from wind power and photovoltaic plants. These distributed power sources shall be placed optimally in a way that efficient as well as secure power grid operation is guaranteed. A combinatorial optimization problem is defined where at 54 potential generation buses[2] any combination of four different plants (photovoltaic and wind power, each in two different sizes) or none may be placed (i.e. 16 possible states per bus). Thus, the resulting solution space is of size 16^{54}. As the simulation of a single solution candidate (= plant locations) takes up to 1 second[3], intelligent optimization algorithms are required for obtaining good solutions in reasonable time (exhaustive search would take obviously around 16^{54} seconds). Since a plant placement has to be valid throughout different system states that may occur over time, its performance is evaluated within a probabilistic multi-stage simulated.

[2]As being defined in the test case description.
[3]In Matlab 7.12 (R2011a, 64 bit) under Windows 7, on an Intel Core i7-2620M CPU at 2.7 GHz, 8 GB RAM.

5.2.1 Formal Problem Description

Equations 3.8 - 3.11 as defined previously in the general OPF formulation build the set of standard load flow constraints in power grids and will be applied here as well.
While satisfying the defined constraints, the objective function is defined of minimizing real power system losses

$$min \sum_{t=1}^{T} L_t(\overline{p_{PL}}), \tag{5.2}$$

with t being a discrete time step during the simulated time span of length T. $\overline{p_{PL}}$ in this case represents the binary vector defining the plant placement, using 4 bits per generator bus describing the existence of the 4 available plants (photovoltaic and wind power, each of two possible sizes).
To avoid infeasible solutions, penalty terms are added to the objective function when evaluating a candidate's fitness. The final fitness function is defined as:

$$F(\overline{p_{PL}}) = \sum_{t=1}^{T} [L_t(\overline{p_{PL}}) + \overline{w_{CV}} * \overline{CV_t(\overline{p_{PL}})}], \tag{5.3}$$

where the objective function $L_t(\overline{p_{PL}})$ is penalized according to the respective degree of constraint violations $CV_t(\overline{p_{PL}})$ within each simulated time step, where $\overline{w_{CV}}$ contains each constraint's (Equations 3.8 - 3.11) individual weight. T is chosen to be 24, hence, a candidate solution is evaluated within 24 discrete equidistant time steps over a simulated day. By this means, the general OPF gets extended in direction of a multiperiod consideration, where the solution $\overline{p_{PL}}$ itself is a static (parametric) one, but has to perform within a dynamic and uncertain environment.

5.2.2 Experiments Description

The optimal placement of renewable power plants forms a probabilistic combinatorial problem, since a possible placement of plants has to guarantee validity in manifold system states that could appear in power grid operation, which are mainly influenced by nondeterministic behavior like weather conditions or uncertain demand situations. The great advantages of using simulation optimization for this problem are obvious: On the one hand, the modeling of the system through simulation enables the full integration of uncertain and intermittent characteristics of sources as well as their effects through probabilistic models. On the other hand, the usage of evolutionary optimization is able to overcome the effects of state space explosion of such combinatorial problems.

Since the resulting optimal placement of renewable plants needs to guarantee valid and secure power grid operation under uncertainty, the evaluation of a solution candidate is performed within different simulated system states (along a time horizon of length T). For imagination of the present simulation optimization issue, its architecture is illustrated in Figure 5.2.

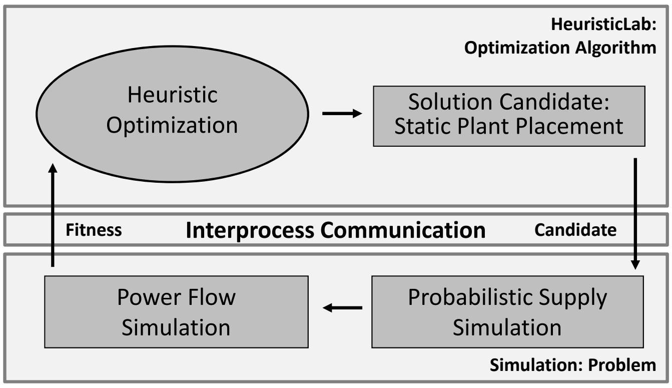

Figure 5.2: Infrastructure Optimization Architecture

As introduced before, HeuristicLab is chosen as suitable framework for heuristic optimization, where Matlab is used for external solution evaluation with probabilistic simulation.
The simulation problem consists of probabilistic models for describing the uncertainty of renewable supply plants connected to a power flow computation. Finally, a sufficient description of the system's operation states is provided that constitutes the respective fitness of the solution. For wind power modeling, the corresponding wind speed values at the plant sites are sampled from a Weibull-distribution as described in [109]. The plants' power curves are assumed such that each site reaches its maximum rated output at a windspeed of 16 m/s. With the sampled wind speed values, the resulting power output of the plant can be modeled using the plant's power curve. Photovoltaic-plants follow a typical daily generation profile that is randomized in each time step with a standard deviation of 10%, considering a typical uncertainty in photovoltaic-generation forecasting. The renewable plants are assumed to have two possible sizes in means of nominal injection, namely 5 MW and 2.5 MW for wind power plants, as well as 0.5 MW and 0.25 MW

for photovoltaic plants respectively.
As loadflow-algorithm, Newton's method is chosen as implemented in MATPOWER. The demand within this system follows typical daily load profiles that are distributed throughout the buses, while the concrete demand value at each bus is randomized normally distributed within each simulation run, using a standard deviation of 4%. For modeling reasons, a constant power factor is assumed for both load and injection values, which is taken from the testcase description.
The evaluation procedure is indicated in Algorithm 2 in pseudocode-notation. Each evaluation is performed $S = 6$ times in order to overcome the uncertainty of the evaluation.

Algorithm 2 Calculate fitness of candidate renewables' placement $\overline{p_{PL}}$

for all $s = 1...6$ **do**
  $initializeGridModel()$;
  $setRenewablePlantPlacements(solutionCandidate)$;
  for all $t = 1...T$ **do**
    $sample(renewableSupply)$;
    $sample(demandValues)$;
    $performPowerFlowComputation()$;
    $fitnessTimeStep[t] \leftarrow (powerLosses + constraintsPenalty)$;
  end for
  $sampledFitness[s] \leftarrow sum(fitnessTimeStep)$;
end for
$fitness \leftarrow mean(sampledFitness)$;

For illustrating the simulation optimization principle, the evaluation of a randomly chosen solution candidate (plant placement) over a simulated day is described in Figure 5.3 as it happens during the optimization process.
At the upper part of the figure, the simulated variables of renewable supply (black bars: photovoltaic plants, gray bars: wind power plants) as well as mean demand (dotted line) are shown, each scaled to its maximum value. The lower part indicates the cumulated system losses with white bars, as well as the penalization caused by operational constraint violations (line overloading, voltage deviation,...) with dashed bars. All values are scaled again in order to allow a qualitative imagination on the simulated system behavior. One can clearly see, that a possible placement decision has to guarantee valid power grid operation throughout manifold and differing system states, where e.g. in the early time steps where demand is low, significant constraint violations occur, making this exemplary placement invalid. Thus, probabilistic simulation throughout multiple periods is necessary for estimating a candi-

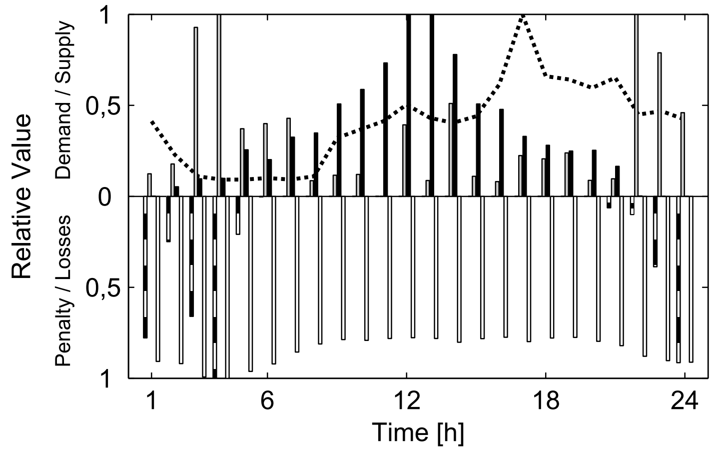

Figure 5.3: Simulation-Based Evaluation of a Solution Candidate

date's performance within later uncertain real-world operation.
The simulation not only allows to fully integrate the system's uncertainty within the optimization process, it further enables to gather information on the resulting power flows using load flow computation based on renewable supply. Thus, a complete picture of the power grid's state can be formed and allows to evaluate a placement's performance. Here, minimizing system losses while satisfying power flow security constraints over a simulated day characterize the optimization problem.

5.2.3 Experimental Results & Conclusion

The full experimental details as well as algorithm configurations for applied genetic algorithms can be obtained from a published paper [54]. Some comparisons shall be taken from this publication and will be extended.
Figure 5.4 shows some algorithmic comparisons of reachable fitness values for this optimization problem, also being illustrative for showing the interplay of objective function $L(\overline{p_{PL}})$ and the penalty for constraints violations $\overline{w_{CV}} * \overline{CV_t(\overline{p_{PL}})}$. Here, two selection schemes are compared, each paired with single-point crossover (SPX), two-point crossover (NPX) and uniform crossover (UX). While a best selection scheme is not even capable of finding valid solutions (notice that fitness values above 600 mostly result because of constraint violations), best found solutions reach a fitness value of around 458. This value corresponds to the objective function of cumulated losses in

MW over all buses over all 24 evaluated discrete time steps, with zero constraint violations. For these solutions, a plant placement is found that allows a partial substitution of the system's supply with renewables (10%), while minimizing system losses and still guaranteeing security constraints satisfaction under stochastic influences.

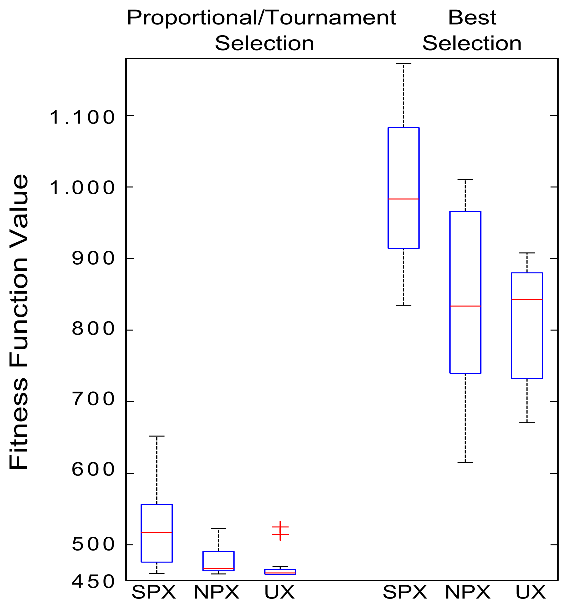

Figure 5.4: Experimental Results to Plant Placement Problem

To get an estimate of this placement issue, 1000 random solutions have been created by adding random photovoltaic or wind power plants of given sizes to random buses until the required 10% substitution is fulfilled. Figure 5.5 shows that these random solutions (regardless of the system losses they would cause) violate secure power grid operation, hence, are invalid. The ordinate shows the mean penalty violation over all random solutions when simulating their performance. The violation of the branch flow constraint (see Equation 3.11) indicates the squared error from P_b^{max} in MW , while the voltage deviation contraint (see Equation 3.10) shows the violation[4] of V_j^{min} in per unit (p.u.) notation. The squared errors are cumulated over all branches $b = 1, ..., NB$ respectively all buses $j = 1, ..., J$. While this Figure only illustrates the mean penalty violation, it has to be mentioned that none of the 1000 random solutions showed to be valid (i.e. yielded zero constraint

[4]An oversupply leads to voltage drops, i.e. violates V_j^{min}.

violations). Thus, an optimization procedure in this case is necessary for deriving a placement that both guarantees secure power grid operation within a stochastic environment (constraint satisfaction) as well as minimizes expected power losses.

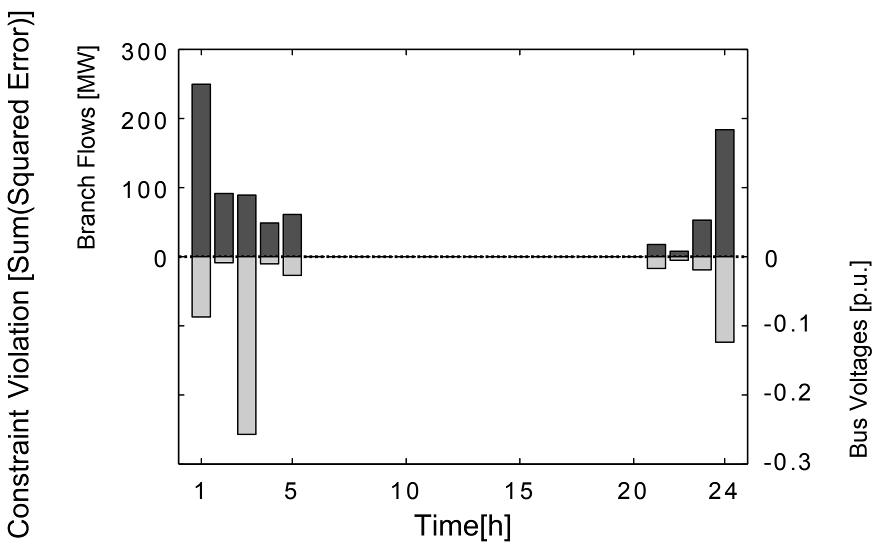

Figure 5.5: Constraint Violations of Random Solutions

The handled problem describes a typical application, where stochastic power flows need to be considered for an optimal planning task. Even if it is not about power flow control, it is strongly related to SOPF (*stochastic* OPF, see introductory chapter). Through simulation, the static solution is evaluated within a dynamic multi-period environment - notice that the simulation is performed along a time span of T steps - that is governed by stochastic variables (uncertain demand as well as uncertain renewable supply). Since the placement of plants cannot change during operation, a static solution representation is suitable, even if the system itself is dynamic as well as stochastic.

Next, a DSOPF problem shall be treated experimentally, showing the validity of the developed technique of learning flexible power flow control policies within a dynamic stochastic environment.

5.3 Dynamic Optimization - Dynamic Stochastic Optimal Power Flow

In Chapter 3, based on the mission statement of this work, the developed policy optimization approach has been discussed in order to enable dynamic stochastic optimal power flow (DSOPF) control. For showing the validity of the policy-based OPF approach as well as for fostering the reader's understanding of this method, it shall be demonstrated by solving benchmark steady-state OPF problems with respect to synthetic dynamic environments. Several publications of the author already treated parts of the subsequent explanations [46, 55].

5.3.1 Formal Problem Description

Within this show case, optimal policies for dynamic power flow control shall be approximated. Once more, for a considered power grid model, the dynamic behavior along a specific time horizon of T discrete time steps is simulated. But rather than optimizing any inflexible parameters (like placement decisions), a dynamic problem is considered where flexible control policies shall be optimized that take system states as input and derive (near-) optimal power flow control actions within each time step in order to guarantee economic power grid operation at runtime.

This application treats a dynamic stochastic optimal power flow problem as discussed in the previous chapter, where policies for the control variables P_G, Q_C, V_G and T_{tap} have to be evolved that perform approximate optimal power flow control both in the simulation model for learning, as well as in estimated real-world operation. For learning such policies, a multi-period simulation model is set-up that simulates the dynamics of a given power system along a time interval of T discrete time steps. This simulation model is used for evaluating a policy's performance within a synthetic dynamic as well as stochastic environment.

While within each time step standard steady-state security constraints (see Equations 3.8 - 3.11) need to be satisfied, the objective function shall be defined as minimizing financial costs of power generation along the simulated time span over all generation units, i.e.:

$$min \sum_{t=1}^{T} \sum_{n=1}^{NG} C(P_{G_n})_t$$

Since a dynamic case is considered where the power injection at a generation site may vary over time, additional ramping constraints need to be defined for variable P_G in order to ensure that this variation is within certain physical limits. Thus,

$$\forall n \forall t : |P_{G_{n,t+1}} - P_{G_{n,t}}| \leq \Delta P_{G_{n,max}}, \tag{5.4}$$

needs to be satisfied additionally.

Considering proportional penalization for constraint violation, the final fitness function is defined as:

$$min \sum_{t=1}^{T} \sum_{n=1}^{NG} [C(P_{G_n})_t + \overline{w_{CV}} * \overline{CV(P_{G_n})_t}], \tag{5.5}$$

where the cardinality of $\overline{w}$ equals the number of considered constraints. The real power injections of the controllable generators $P_{G_2}...P_{G_{NG}}$ directly result from the policies, while the slack generator's injection value P_{G_1} is implicitly derived from power flow simulation.

5.3.2 Experiments Description

Within these experiments, the same distribution grid models are taken as before (specified in Table 5.1), while creating dynamic multi-stage optimal power flow problems out of them.
For modeling the power flow control in a dynamic and uncertain environment based on these test cases, the power grid is simulated using MATPOWER along a time-horizon of $T = 96$ discrete time steps ($\Delta t = 15$ minutes, i.e. time horizon equals one day) in order to build a multi-period dynamic environment out of steady-state situations[5]. Here the load changes over time according to statistical profiles from power grid operation, but is additionally randomized in order to simulate real-world uncertain conditions. At each time-step, steady-state power flow computation is performed, where the stochastic variables (i.e. load values) are sampled from respective distributions. Notice that steady-state power flow computation is a sufficient simulation method within the discrete time steps even if a dynamic environment is created out of them, since the time domain of decisions is high enough (in the order of minutes) - as discussed in the introductory chapter when considering steady-state versus transient power flow simulation.

[5]Some researchers already identified the application of creating a dynamic OPF problem out of steady-state scenarios, where the general term of so called multiperiod-OPF problems has evolved in recent years [65].

Training: Learning Optimal Policies

Within this simulation, policies are trained in order to lead to (near-) optimal power flows within uncertain future system states. For this reason, each solution is evaluated within 20 simulated power grid states that are randomly (equally distributed) spread along the considered time horizon within each evaluation. Additionally, in order to avoid overfitting the learned policies to specific demand situations, one out of four load profiles is selected randomly (equally distributed) at the beginning of each evaluation. These profiles as shown in Figure 5.6 are used for power grid operation in Austria within the APG regulation zone.They represent typical load trends for domestic (*H0*) as well as commercial (*G0,G6*) and agricultural (*L0*) customers, based on APCS[6] load profiles for the 2.7.2012. A fifth profile is indicated as well being a randomly generated one for enabling the later testing of the policies to arbitrary volatile situations. All profiles are scaled to the interval $[0, 1]$.

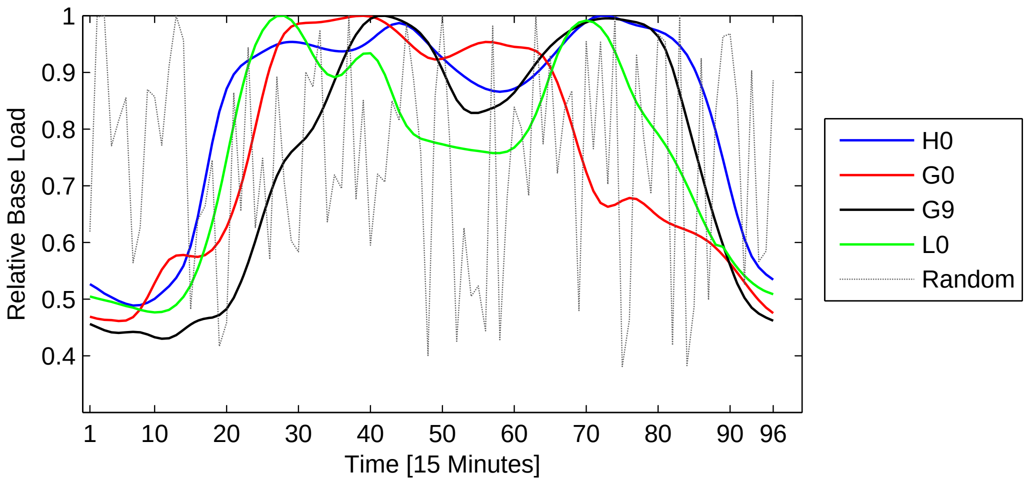

Figure 5.6: Applied Load Profiles

Notice that - when considering the cumulated demands throughout a power grid - such high load variations induced with the applied load profiles are not realistic. But in order to show the capability of the approach to handle highly volatile demand situations for power flow control, an appropriate variance of higher order is assumed herein. To additionally model the

[6]Austrian Power Clearing & Settlement, http://www.apcs.at/

uncertainty of load forecasts in power grid operation, at each time step the load value at each bus is randomized normally distributed with $\sigma = 0.04$. The procedure of evaluating a solution candidate (= policies for all required variables) is depicted in Figure 5.7. After randomizing the demand situation, the computed policies are evaluated for all output variables (i.e. elements of $\overline{u}$) which yields - after power flow computation - the resulting constraints violations as well as the objective function value. Performing this evaluation for 20 randomly selected time steps along the simulated time horizon, the fitness of a solution candidate is obtained.

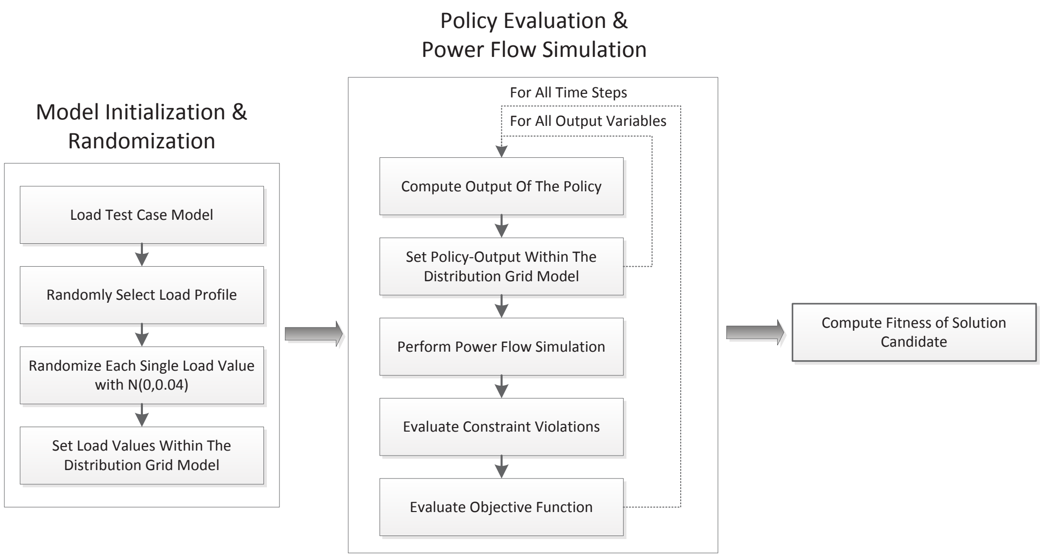

Figure 5.7: OPF Policy Evaluation Procedure

At this point, the reader shall be made aware, that this application still treats the traditional OPF, where the generation is scheduled for meeting a given (stochastic) demand. Thus, *load-dependent generation* is operated.

Testing: Evaluating the Learned Policies

In order to evaluate the validity of the learned flexible policies on a separate test scenario, a proper load profile is applied which is randomly generated (see Figure 5.6) in order to test the found policies on arbitrary dynamic situations that are independent from the training environment.
For making the results comparable, when simulating the distribution grid

with the random load profile along the time horizon, ten discrete (static) system states equally distributed over time are expressed from the simulation. For these ten test states (i.e. 10 chosen time steps), the exact steady-state solution to the OPF is computed with primal dual interior point method (same solver as before in Section 5.1), implemented in MATPOWER. These steady-state solutions are then compared to the solutions that the policies would lead to in each state (i.e. time step).

Algorithmic Settings

The following parameter settings have finally been applied for the optimization experiments. When optimizing policies of fixed structure (i.e. using a metamodel), the control variables comprise real valued weights given by the vector $\overline{w}$ as defined before (see the concept of metamodel-based synthesis, Section 4.2.7). Therefore, real-valued optimization is performed, where evolution strategies are applied as for the steady-state benchmark systems optimization in Section 5.1.
When applying ES to uncertain optimization problems (where only an estimate $f'(\overline{w})$ of the true fitness function value $f(\overline{w})$ can be provided by the solution evaluation), it is important to use a comma selection scheme in order to avoid that overestimated solutions of poor quality remain in the population. Full parameter settings are given in Tables 5.4 and 5.5.

When applying genetic programming for metamodel-less synthesis of policies, another class of metaheuristic algorithms is applied; namely genetic algorithms (GA). Contrary to ES, GA's major operator for solution space exploration is crossover. Especially when being applied for computing symbolic expressions with genetic programming (as herein), high population sizes as well as high mutation rates are needed in order to overwhelm the huge solution space. Respective settings that are used for all test cases are given

Parameter	**Value**
ES Type	(5,20)
Manipulator	SelfAdaptiveNormalAllPositions- Manipulator
Strategy Parameter	Learning Rate $\tau = 0.5$
Recombinator	Discrete Crossover
Parents per Child	2
Stopping Criterium	Maximum Generations: 1000

Table 5.4: Parameters for Test Cases: IEEE 14 Bus, IEEE 30 Bus

Parameter	Value
ES Type	(10,40)
Manipulator	SelfAdaptiveNormalAllPositions- Manipulator
Strategy Parameter	Learning Rate $\tau = 0.5$
Recombinator	Discrete Crossover
Parents per Child	2
Stopping Criterium	Maximum Generations: 2000

Table 5.5: Parameters for Test Cases: IEEE 57 Bus, IEEE 118 Bus, IEEE 300 Bus

in Table 5.6. Similar to using comma-selection in ES, with GA no elitism is applied in order to avoid that (probabilistically) overestimated solutions of poor quality become elites and thus remain in the population.

For better understanding, the resulting trees that get evolved by the genetic programming process can be formalized as follows:

$$tree(L, root, R) =< L, root, R >,$$

with left and right subtree L and R as well as the root node $root$. Here, the following holds:

$$\begin{aligned}
root &= innerNode \\
L &= tree|terminalSymbol \\
R &= tree|terminalSymbol \\
innerNode &= \text{“}+\text{”}|\text{“}-\text{”}|\text{“}*\text{”}|\text{“}/\text{”} \\
terminalSymbol &= constant|rule \\
rule &\in \overline{r} \\
constant &\in \mathbb{R}
\end{aligned}$$

Parameter	Value
Maximum Generations	200
Population Size	400
Mutation Probability [%]	15
Crossover Probability [%]	100
Continued on next page	

Parameter	Value
Selector	Tournament Selector Group Size: 5
Mutator	Multi Symbolic Expression Tree Manipulator[7]: Replace Branch Manipulation, Change Node Type Manipulation, Full Tree Shaker, One Point Shaker
Crossover	Subtree Swapping Crossover
Elitism	No Elitism
Maximum Tree Depth	8
Maximum Tree Length [nodes]	80
Tree Grammar	Arithmetic Operators Real-Valued Constants

Table 5.6: GP Parameters: All Test Cases

5.3.3 Experimental Results & Conclusion

Techniques for OPF policy learning have been introduced in Chapter 3 and shall now be applied to the defined test cases. Comparisons will be conducted throughout these different test cases (power grid models) in order to provide experimental results underlining the abilities of the respective techniques.

System Variable Policy Synthesis

As introduced previously in Section 4.2.4, system variable synthesis (SVS) is probably the most obvious approach for evolving policies out of system information from an engineering point of view. Therefore, system variables $\bar{i}$ are directly taken as policy inputs without the step of deriving abstract rules. Within this experiment, SVS shall be applied primarily to the 14-bus test case, which is the smallest one within the given set of models in means of needed system variables. Within this scenario, all the information needed for building the policy for the variable P_G , i.e. $P_G(\bar{i})$, is contained by the input vector $\bar{i}$ defined as follows (this issue has already been discussed in 4.2.4):

[7]Multi Symbolic Expression Tree Manipulator as implemented in HeuristicLab: Randomly applies one of the defined operators with equal probability.

$$i^T = [P_{L_1}...P_{L_{NB}}, Q_{L_1}...Q_{L_{NB}}, P_{G_1}^{min}...P_{G_{NG}}^{min}, Q_{G_1}^{min}...Q_{G_{NG}}^{min}, P_{G_1}^{max}...P_{G_{NG}}^{max}, Q_{G_1}^{max}...Q_{G_{NG}}^{max}, P_{b_1}^{max}...P_{b_{Nb}}^{max}, a_1...a_{NG}, b_1...b_{NG}, c_1...c_{NG}],$$

with all respective load values P_L and Q_L, generation limits of all buses ${P_G}^{max}$ and ${Q_G}^{max}$, branch flow limitations ${P_b}^{max}$ as well as all generator cost coefficients a_g, b_g and c_g. In order to maintain the voltage deviation constraint (Equation 3.10), this information on voltage boundaries would be needed as well. As it is assumed that V^{max} is equal for all buses (in p.u.[8]), this information is not needed additionally, while this constraint is fixed to $V^{min} = 0.94 p.u.$ and $V^{max} = 1.06 p.u.$. Thus, for the 14-bus test instance, the cardinality of $\bar{i}$ equals[9] 49. Since policies for 4 generators need to be evolved ($NG = 5$ generators minus the slack bus which cannot be controlled directly), $4 * 49 = 196$ control variables (i.e. weights) would be needed for linear synthesis only for the variable P_G (according to Equation 4.3)!

Within these experiments for SVS, only the generators' real power injection (variable P_G) will be used for policy optimization, while this variable indeed has the highest impact on power flow control and thus on the considered cost function. Variables V_G, T_{tap} and Q_C are fixed to scalar values given in the model data for keeping the solution space manageable within these experiments, but will be considered later.
Building policies out of the defined variables $\bar{i}$, polynomial synthesis with linear as well as quadratic functions (as used later) has shown to be invalid in this case, which is obvious since the powerflow within a distribution grid is strongly nonlinear with respect to $\bar{i}$ (the interested reader could take a look at general power flow formulations [116] at this point). Thus, synthesis with GP is the only remaining choice herein, since it enables the identification of even nonlinear policies. The GP parameter settings are given above, while arithmetic operators are used for combining the input variables in $\bar{i}$ to the final policies. The best found solutions are given in Appendix A.4, Equations A.1-A.4. While for each generator $P_{G_2}...P_{G_{NG}}$ a proper policy has to be learned, a coevolutionary optimization according to Section 4.2.9 has been applied as all these generators are indeed interrelated.

[8]The per unit (p.u.) notation system is used in the power engineering field for expressing quantities as fractions of defined base-unit quantities. For modeling issues, normally base-quantities are defined (mostly power and voltage) so that all other quantities in the model are defined as a multiple of them. In the case of the voltage deviation this means that ±6% from the reference voltage are allowed.

[9]$NB + Nb + 3 * NG = 49$

For exemplary reasons, a representative illustration is given in Figure 5.8 showing the performance of an arbitrary solution taken during the training process (policies $P_{G,2}...P_{G,5}$) with SVS. Additionally, along all time steps within this evaluation, the exact steady-state OPF solution is computed with interior point method implemented in MATPOWER. The fitness of this exact static solution in each discrete state (dotted blue line, right hand ordinate) is plotted with respect to the fitness of the policies' outputs within these states (solid blue line, right hand ordinate), where the relative difference is indicated (red line, left hand ordinate) over a simulated day. This evaluation is performed on a simulated scenario using H0 load profile as appearing during the optimization process for training.

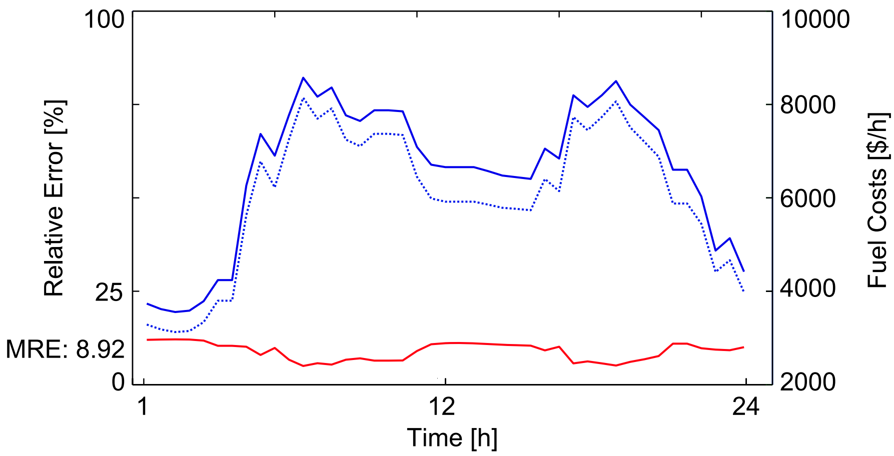

Figure 5.8: 14-Bus Case: Illustration of SVS Solution Evaluation

In this figure, one can see that along this simulated day the power flow actions generated by the obtained SVS policy are mostly only around 9% worse in means of resulting fitness value compared to the deterministic interior point solution within each discrete state. This gap from the fitness of the deterministic solution (which is assumed to be the best one within each state but which disregards dynamic near-future behavior) shall further be entitled as error.
This relative error is defined as follows:

$$RE_t = \frac{f(p)_t - f(\overline{u}_{exact})_t}{f(\overline{u}_{exact})_t}, \tag{5.6}$$

with $f(p)_t$ representing the fitness of the policy's output and $f(\overline{u}_{exact})_t$ giving the exact solution's fitness at time t.

Note that for final comparisons (in Table 5.7) a proper test set is used based on a random load profile. During the optimization, the interior point solution is not known to the process, while the quality (i.e. fitness function value in fuel costs per hour) of the solution candidate is the only known information for the metaheuristic search.

Building a Test Set

Since within each discrete time step (state) the static OPF solution can be computed exactly with interior point method, this introduced "error" (i.e. the dynamic solution's relative distance from the exact static solution within a state) is suitable for validating the found policies. As discussed before, a proper random load profile is applied to the power grid model for building a separate test scenario of arbitrary volatility. Out of the simulation of this scenario along 24 hours, 10 randomly chosen discrete states are expressed that build the test set. For these 10 states, the error value is computed.

Table 5.7 shows the final validation of the trained best found solutions on the test set, where the reachable qualities (i.e. relative errors as defined in Equation 5.6 within ten chosen discrete time steps of the test scenario) are compared for the best found solutions of system variable synthesis with GP. Additionally, for comparing system variable synthesis (SVS) with abstract rule synthesis (ARS), a policy for variable P_G has been learned with ARS as well, both with polynomial metamodels (linear and quadratic) as well as with GP.

Time Step	SVS GP	ARS linear	ARS quadratic	ARS GP
1	0.0013	0.1008	0.0156	0.0023
2	0.0050	0.0387	0.0312	0.0013
3	0.0103	0.1528	0.1700	0.0246
4	0.0070	0.0929	0.1064	0.0093
5	0.0068	0.0870	0.1001	0.0082
6	0.0044	0.0825	0.0185	0.0022
7	0.0044	0.0825	0.0185	0.0022
8	0.0047	0.0599	0.0230	0.0017
9	0.0060	0.0554	0.0666	0.0032
10	0.0055	0.0351	0.0448	0.0014
MRE	**0.0055**	**0.0788**	**0.0595**	**0.0056**

Table 5.7: Results 14-Bus P System Variable Synthesis

These results show that it is possible to derive comparable policies for the variable P_G both with SVS as well as with ARS, that are in average only 0.55% worse than reachable solutions with deterministic optimization in each state. Compared to metamodel-based approximation with linear and quadratic models, these results are superior. Linear and quadratic policies are nevertheless capable of approximating optimal power flow actions with an error of 7.88% and 5.95% respectively due to the application of abstract rules. Additionally, the qualitative facts substantiate ARS, where the number of needed control variables (and thus the solution space complexity of the optimization problem) is huge for SVS and grows with the size of the treated power grid, while it can be reduced significantly when building abstract rules[10]. Additionally, this application to a relatively small scenario illustrates, that all the needed information for deriving power flow control actions is contained within the defined set of abstract rules (since both SVS and ARS lead to similar results), making the use of all system input variables unnecessary and unattractive.
Further experiments shall now be conducted, where ARS will be used for evolving policies for all other control variables as well.

The following experiments shall demonstrate the capabilities of power flow control policy approximation with Abstract Rule Synthesis. While scalability of this technology is an important issue, experiments will be conducted on all IEEE testcases defined above. For all these cases, the complete set of policies $\{P_G, Q_C, V_G, T_{tap}\}$ will be learned both using metamodels as well as applying metamodel-less approximation with GP. At this point it is important to mention, that with SVS, policies would need to be learned for every controllable unit in a system (like for all $NG-1$ generator real-power outputs). At the same time, with ARS only one policy needs to be evolved for all units of same type, like for all generator real-power outputs, generator voltages, VAR compensators or transformer taps. This fundamental feature is enabled through the usage of abstract rules that provide unit-specific information to the policy, making a policy's output individual to each unit even if the policy itself is generic.

[10] This scalability issue has already been treated in Section 4.2.5 and will be deepened in Section 5.5.

Abstract Rule Synthesis - Polynomial Metamodels

Based on the abstract rules defined in Table 4.1, policies will be learned with a given fixed mathematical structure (i.e. metamodel), namely polynomials of degree $\{1, 2\}$, i.e. linear and quadratic functions. Therefore, real-valued optimization is necessary for finding performant values of the weights-vector $\overline{w}$, which takes 25 respectively 46 elements (independent of the power grid's size!).
Table 5.8 shows the results when validating the best found solutions on the test set, where once more the relative error in means of fitness is shown between the policies' actions and the deterministic interior point solutions for ten randomly chosen discrete time steps. Comparisons are performed for all previously mentioned IEEE distribution grid test cases.

Time Step	**14-Bus**	**30-Bus**	**57-Bus**	**118-Bus**	**300-Bus**
1	0.0248	0.0085	0.0311	0.0603	-
2	0.0363	0.0071	0.0663	0.0527	-
3	0.1405	0.0061	0.0311	0.0588	-
4	0.0923	0.0071	0.0663	0.0539	-
5	0.0876	0.0057	0.0415	0.0784	-
6	0.0270	0.0074	0.0459	0.0687	-
7	0.0270	0.0059	0.0439	0.0542	-
8	0.0303	0.0098	0.0399	0.0709	-
9	0.0625	0.0062	0.0603	0.0573	-
10	0.0463	0.0097	0.0647	0.0803	-
MRE	**0.0575**	**0.0074**	**0.0491**	**0.0636**	**-**
Model Class	quadratic	linear	quadratic	quadratic	-

Table 5.8: Results Metamodel-Based Synthesis

In contrast to the other cases, for the 30-bus test case, the linear synthesis reached an equivalent solution quality compared the synthesis with a quadratic function. While the best found solutions for all cases are provided in Appendix A.5, the found policies for the 30-bus case shall be given here for illustrative reasons in Equations 5.7-5.10, where the policies are synthesized according to Equation 4.3.

$$P_G(\overline{r}) = (LLF + 0.7859 * NLFS + GLF + 0.4303 * MRF + 0.5329 * MERF + 0.3474 * LCCF + QCCF)/7 + 0 \quad (5.7)$$

$$V_G(\overline{r}) = (0.5312 * NLFS + 0.5624 * GLF + 0.6094 * NRLF + 0.728 * GRLF)/4 + 0.7976 \quad (5.8)$$

$$Q_C(\overline{r}) = (0.9679 * NLF + 0 * GLF + 0.9492 * MRF + 0.6566 * MERF + 0 * NRLF + 0.3444 * GRLF)/6 + 0.1996 \quad (5.9)$$

$$T_{tap}(\overline{r}) = (0 * NLF + 0.1743 * GLF + 0.814 * NRLF + 0.3958 * GRLF)/4 + 0.147 \quad (5.10)$$

For most cases, the mean relative error in means of fitness when comparing the policies' actions to exact steady-state OPF solutions on a defined test set is in the range of 5% to 6%, and even below 1% for the 30-bus case. However, for the largest instance, the 300-bus test system, this method has not been able to find valid policies. For this system, a polynomial metamodel seems not to be capable of approximating optimal decisisons appropriately. As visualized in the illustrated solution to the 30-bus instance, some rules are weighted with 0, which clearly shows that their information is not necessary in this case respectively is contained by other rules. Thus, some kind of feature selection could be of advantage, reducing the complete set of rules $\overline{r}$ to only a necessary subset.

The results clearly demonstrate the power of the policy-based approach throughout a given set of test instances, where a flexible policy is learned offline and avoids any re-optimization during the operation of a volatile and uncertain power grid. Using abstract information (ARS) rather than a huge set of system input variables (SVS) as before, it can be shown for this set of standard test instances, that relatively simple (and even linear!) mathematical structures are sufficient for building policies which derive accurate as well as valid power flow decisions at runtime. However, a more sophisticated method may be needed especially for the 300-bus case, which is able to identify more complex relationships for synthesizing more powerful policies out of the set of rules.

Abstract Rule Synthesis - Genetic Programming

Genetic programming - as discussed before - provides a method for synthesizing policies free from a predefined mathematical structure (metamodel). Here, within an evolutionary optimization process, policies are built out of a set of input variables (abstract rules) and a given grammar (set of mathematical operators) being represented by tree structures.

HeuristicLab offers implementations for GP, algorithmic settings have already been given. Once more, when applying GP for evolving multiple interrelated policies (such as $P_G(\overline{r}), Q_C(\overline{r}), V_G(\overline{r})$ and $T_{tap}(\overline{r})$), cooptimization as proposed in Section 4.2.9 was necessary.
Experiments are conducted for the whole set of distribution grid test cases as before, where the test set performances of best found solutions are compared in Table 5.9. As indicated before in Table 5.6, only arithmetic operators are comprised by the used grammar for the sake of search space limitation, while numerous additional operators would be possible such as sinoid or exponential functions.

Time Step	14-Bus	30-Bus	57-Bus	118-Bus	300-Bus
1	0.0074	0.0051	0.0368	0.0083	0.0160
2	0.0053	0.0049	0.0142	0.0048	0.0201
3	0.0105	0.0056	0.0368	0.0075	-0.0151
4	0.0045	0.0049	0.0142	0.0079	0.0033
5	0.0041	0.0058	0.0081	0.0173	0.0030
6	0.0071	0.0060	0.0060	0.0127	0.0046
7	0.0071	0.0059	0.0067	0.0051	0.0048
8	0.0064	0.0060	0.0467	0.0137	0.0033
9	0.0034	0.0052	0.0103	0.0067	0.0030
10	0.0042	0.0057	0.0222	0.0181	0.0082
MRE	**0.0060**	**0.0055**	**0.0202**	**0.0102**	**0.0051**

Table 5.9: Results of Metamodel-Less Synthesis with GP

The great advantage of synthesizing policies with GP is that it allows arbitrary mathematical combinations of abstract rules and thus is able to identify even complex (non-linear) coherences between rules without a-priori knowledge of these relationships. Additionally, contrary to applying a fixed mathematical structure, GP enables an implicit feature selection mechanism and thus builds policies that do not necessarily take all $|\overline{r}|$ abstract rules into account for the final solution, but is able to perform with only a subset of $\overline{r}$

if successful during the genetic search.

Overall, when performing the test set comparisons, throughout all test cases impressive solutions have been achieved with approximate policy-based control, with errors of even below 1% for three of five instances. Even in one test state (300-bus instance, time step 3), the approximate policy-based solution leads to a better quality than the exact one. The reason for this is the same as before, namely that the interior-point method in MATPOWER uses a linearized approximate model of the non-linear power flow equations for optimization, while the simulation optimization of policies uses the actual power flow simulation model without approximations.

Equations 5.11-5.14 give the best found solution for the 30-bus case. Since within this work GP solutions are evolved represented by trees with arithmetic operators as inner nodes, these solutions provide algebraic equations. For making them more readable, real-valued constants are extracted (by the variables rc_x) and listed in Table 5.10.

$$\begin{aligned} P_G(\overline{r}) &= rc_1 * (rc_2 * merf + rc_3 * glf - rc_4 * lcf) \\ &* (\frac{rc_5 * glf * (rc_6 * glf - rc_7 * lcf)}{pcf^2} + rc_8 * glf * mrf) + rc_9 \end{aligned} \tag{5.11}$$

$$\begin{aligned} Q_C(\overline{r}) &= \frac{-rc_{10} * nlfsq}{merf * glf} + rc_{11} * nlfs + rc_{12} * glf - rc_{13} * merf \\ &- rc_{14} * glf * mrf * nlfsq \end{aligned} \tag{5.12}$$

$$\begin{aligned} V_G(\overline{r}) &= rc_{15} * nlfs - rc_{16} * nlfsq \\ &* (rc_{17} * nlfsq + \frac{rc_{18} * glf}{-rc_{19} * glfq + rc_{20} * nlfsq}) + rc_{21} \end{aligned} \tag{5.13}$$

$$T_{tap}(\overline{r}) = \frac{rc_{22} * nlfsq}{glfq} \tag{5.14}$$

While these policies tend to be more complex than those provided by synthesis with fixed-order polynomials, they lead to significantly better performing solutions. Anyway, because of the implicit feature selection in GP, it is possible to derive simpler policies as well. The policy for the transformer

Constant	Value	Constant	Value
rc_1	0.022584392	rc_{12}	0.193319217
rc_2	0.300369793	rc_{13}	0.127834428
rc_3	1.156898808	rc_{14}	0.643195687
rc_4	0.482939198	rc_{15}	0.029170953
rc_5	3.209932224	rc_{16}	0.043287062
rc_6	1.241398697	rc_{17}	1.672301840
rc_7	0.482939198	rc_{18}	0.814501832
rc_8	8.870819778	rc_{19}	0.757900514
rc_9	0.281666168	rc_{20}	1.468518450
rc_{10}	0.077008964	rc_{21}	0.955877454
rc_{11}	0.238363556	rc_{22}	6.724411552

Table 5.10: Real-Valued Constants Assignment 30-Bus ARS

tap setting $T_{tap}(\overline{r})$ in Equation 5.11 represents an illustrative example therefore.

The obtained results clearly demonstrate the power of GP for this application, where its ability of finding more complex policies without being restricted to a predefined metamodel leads to superior solution qualities on the test states.
Especially for the largest instance, the 300-bus test case with $(NG-1)+NG+NQ+NT=252$ control variables (that are controlled by only 4 distinct policies P_G, V_G, Q_C and T_{tap}), GP was still able to produce an accurate set of policies, while no valid solution has been achieved with polynomial synthesis.

At this point, it is important to mention that only 4 policies have been learned here in each instance for numerous control variables (like 252 variables in the largest case). If another distribution grid would be considered with for example a higher number of controllable generation units, still only 4 policies would be needed for dynamic OPF control on this power grid. Thus, the technology of learning flexible control policies with abstract rules is highly scalable to applications with lots of distributed control devices. Especially the synthesis with GP allows the identification of complex nonlinear relationships between the abstract rules while providing an implicit feature selection, enabling the optimization of powerful control policies within dynamic and stochastic environments.

5.4 Complexity of Approximated Policies

From prior experiments, it was concluded that it is possible to approximate policies of high quality (i.e. low error values) while keeping their mathematical complexity low. The reason for this is twofold: on the one hand, abstract rules are able to aggregate information from the power grid's state in a highly compressed manner, enabling to build simple (even linear) policies that are still able to approximate optimal power flow decisions. On the other hand, GP enables an implicit feature selection that is able to identify those input variables (no matter if $\bar{i}$ or $\bar{r}$) that are important to the decision process. Thus, not only the best obtainable quality of a policy is of interest, but also the relation of a policy's quality and complexity.

GP provides a suitable technology for performing experiments that analyze this relation. Here, mathematical expressions of arbitrary structure can be evolved, where the only restriction is the maximum allowable tree size as well as the provided grammar. While in above experiments this tree size has been fixed in the way that the GP process has a sufficiently large degree of freedom for finding best performing expressions, the relation between quality and complexity shall be investigated in following experiments.
Therefore, optimization experiments are designed where the solution complexity in means of allowable tree length is increased incrementally while computing the best obtainable solutions for each length value. Beside the tree size, the GP configuration remains the same as above.

Figure 5.9 illustrates the results to this experiments performed on the 14-bus test case. Here, policies for variable P_G are evolved both with SVS as well as with ARS (variables V_G,Q_C and T_{tap} are fixed to discrete values for easing experiments[11]). The obtainable solution quality in means of error on the test case is analyzed for the 5 best found solutions within each fixed maximum tree length, where the length value is increased starting with small steps and increasing step size for estimating the potential of very large expressions.
In this figure, the left side shows the analysis for SVS, while ARS results are given on the right hand side. Here, two main outcomes can be derived: first, the correlation of achievable quality and solution complexity emerges clearly, where the improvement in quality stagnates with large trees. Second, this improvement is significantly steeper for ARS than with SVS and stagnates

[11]The decision on variable P_G has the most significant impact on the power grid operation, making this approach valid.

earlier. This is due to the fact that abstract rules compress information from $\bar{i}$ and thus enable derived policies to come along with smaller complexities. This observation additionally substantiates the application of abstract rules that are able to simplify control policies while still preserving high quality values. Especially for ARS a maximum tree length of 24 nodes seems to be promising, not only leading to best results, but also to a most robust heuristic optimization process. Higher allowed length values make the genetic search more difficult due to the increased solution space size, while not promising better solution qualities.

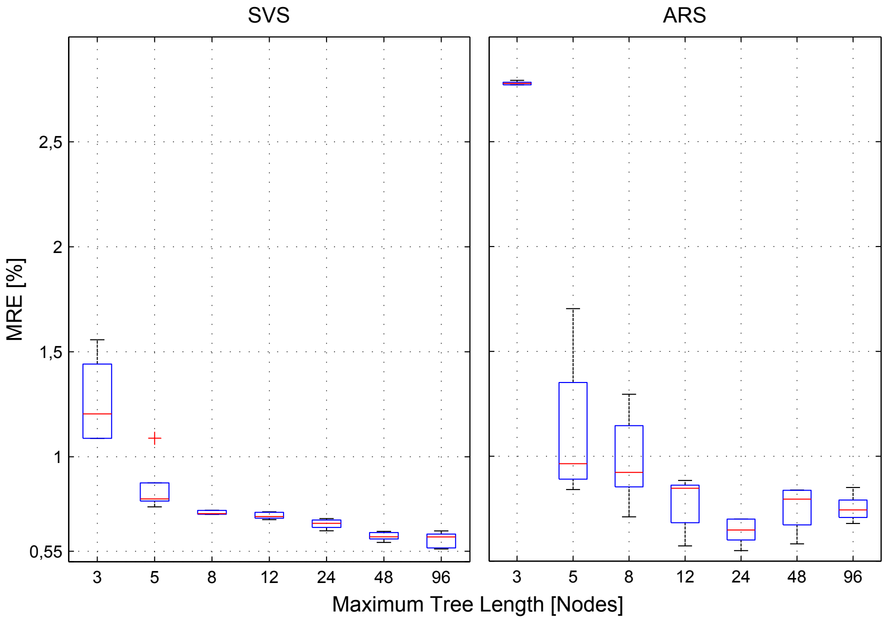

Figure 5.9: Policy Complexity Analysis for 14-Bus Case

The same analysis is performed for a larger test instance as well, namely the 30-bus test case illustrated in Figure 5.10, where once more policies for variables $P_{G2}...P_{GNG}$ are evolved, while V_G, Q_C and T_{tap} are fixed. Here, the observed facts remain the same, but the quality improvement is flatter along the tree length steps especially for SVS. This is obvious since this test case is larger and exhibits a more complex behavior as well as a higher amount of input variables $\bar{i}$, demanding policies of higher complexity. Once more, ARS seems to perform well with policies of lower size than approximation with SVS. Additionally, for those experiments in a more complex system, ARS in general leads to slightly better results regarding the best achieved solutions.

For both analysis, the best achievable solution quality is comparable for SVS and ARS. This is an important indicator that the set of rules $\overline{r}$ contains all necessary information from $\overline{i}$, but is able to compress this information and enables the policy approximation with mathematical expressions of lower complexity. Besides this complexity issue, another essential advantage of using abstract rules is the scalability in order to control high amounts of distributed devices. This fact shall be considered more in detail now.

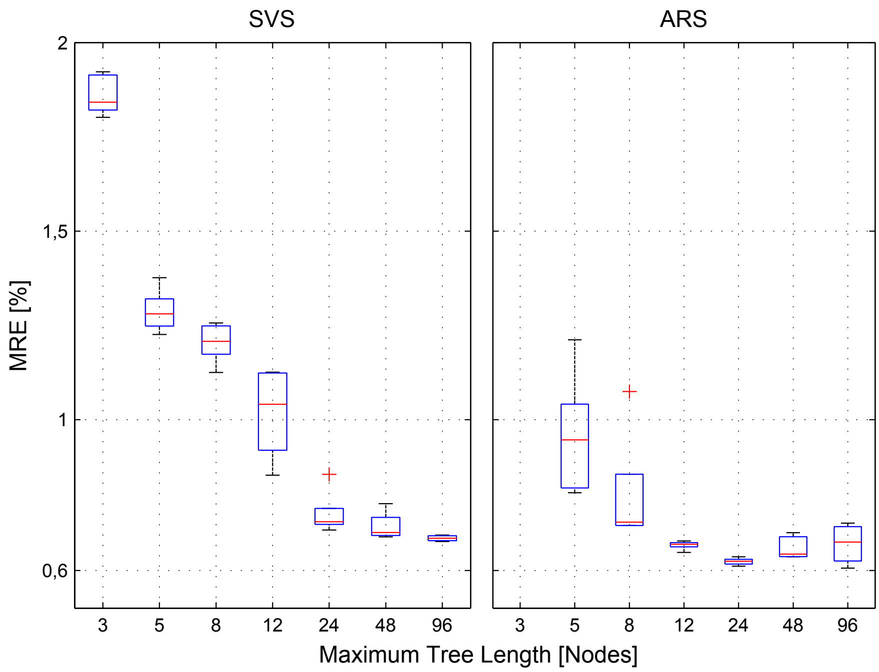

Figure 5.10: Policy Complexity Analysis for 30-Bus Case

5.5 Scalability Issues with SVS and ARS

The scalability of a control technology has already been stated as central requirement, which is important especially with regard to future decentral and distributed devices in electric power grids. Abstract rules have been shown to be promising therefore, since they reduce the amount of needed input variables, making policy-based control attractive to high-dimensional decision problems. Additionally, abstract rules deliver unit-specific information, enabling policies to provide a general decision system across multiple devices. While with SVS, for example a specific policy for real-power injection P_{Gn} would need to be derived for each of NG generation units, with ARS it is possible to compute only one general policy $P_G(\bar{r})$ which is valid for all units. Table 5.11 illustrates this issue for all power flow test cases.

Property	14-Bus	30-Bus	57-Bus	118-Bus	300-Bus
# Buses	14	30	57	118	300
# Branches	20	41	80	186	411
# Generators	5	6	7	54	69
# Tap Changers	3	2	3	9	8
# VAR Compensators	1	4	17	12	107
# Control Variables	13	17	33	128	252
# Policies $p(\bar{i})$	13	17	33	128	252
# Policies $p(\bar{r})$	4	4	4	4	4
# Variables $\|\bar{i}\|$	49	89	158	466	918
# Rules $\|\bar{r}\|$	7	7	7	7	7

Table 5.11: Scaling of Test Cases

For each test case, 4 types of controllable units need to be considered, namely P_G, Q_C, V_G and T_{tap}. For each type of unit, a number of controllable devices exist, yielding the number of needed control variables which varies between 13 and 252 for the smallest and the largest test case respectively. While with SVS a specific policy would need to be approximated for each control variable (yielding 252 needed policies in the largest case), with ARS still only 4 policies are needed since they provide device-specific decisions. Hence, policy approximation with ARS is highly fruitful for controlling huge amounts of distributed devices.
A second consideration needs to be done with regard to the number of variables that a policy needs to include: when modeling policies out of all input variables $\bar{i}$ with SVS, in the largest case a policy would need to take up to 918 variables. Using ARS instead, the number of rules remains constant and the relation $|\bar{r}| << |\bar{i}|$ gives an important advantage to the optimization process in general.

5.6 Findings & Preliminary Statement

Future power flow control tasks in smart electric grids require optimization methods that are capable of deriving fast and robust control actions in dy-

namic and uncertain environments for potentially high amounts of controllable devices. Thus, the need for scalable technologies enabling dynamic stochastic optimal power flows is fundamental.
A new approach for dynamic approximate power flow optimization has been proposed, where flexible control policies are learned offline that later provide (near-) optimal control actions at runtime. One special ability of this approach is that respective policies are learned out of abstract information entities - so called abstract rules - that make it highly scalable. The learning procedure is realized using evolutionary simulation optimization. Since uncertainties of a complex power grid have to be taken into account for evolving valid policies for real-world optimization, simulation is applied as system representation that allows to fully integrate the probabilistic system behavior into the optimization process.
Finally, the proposed technology has been validated on well-known benchmark systems in the power grid engineering society, the IEEE distribution grid test cases. Out of these benchmarks, dynamic multi-period power flow problems have been simulated for evolving policies. It has been shown, that for randomly chosen sets of discrete steady-state test states, the policies' outputs lead to competitive power flow control actions when comparing them to statically optimized exact solutions within these states. Thus, a technology is available for accurate and scalable dynamic stochastic optimal power flow control.

Here, the application to the general OPF has been demonstrated for IEEE benchmark systems for illustrating its validity, but numerous additional applications could be of interest like shown in [50], that potentially benefit from using the discussed principles. For example, with the same approach policies could be evolved for intelligent and autonomous control of distributed smart appliances like controllable plug-in electric vehicle charging infrastructures. Thus, a generic, flexible and scalable approach was presented being capable of representing a fundamental tool for future smart electric grids.
The next chapter shall motivate this application to future power grid engineering, where a generic and illustrative smart-grid scenario with numerous distributed devices will be developed. Applying the discussed policy optimization techniques finally shows its capabilities for large-scale distributed control issues.

Part III

Smart Electric Grids

Chapter 6

Scalable Dynamic Stochastic Optimal Power Flow Control in Smart Grids

Having developed a policy approximation approach as scalable technology for dynamic stochastic optimal power flow control, its application has been shown to benchmark systems that represent traditional power grid control schemes. This approach seems to be capable of handling numerous distributed control units within uncertain as well as volatile environments, hence, creates a fruitful ground for enabling optimal power flow control in future smart electric grids. This chapter is now about demonstrating this issue, where a typical smart grids scenario is developed that is representative for multiple future technologies. By applying the simulation-based evolutionary policy approximation to this scenario, the utilization of this developed technology to power flow control in smart grids will be illustrated.

6.1 Developing a Smart Grids Scenario

In Section 2.2 the general features of this desired technological change to smart grids have been discussed. Beside many other challenges, the issue of guaranteeing robust as well as fast power flow control in volatile environments is of central concern.

Regarding power flows and related control tasks, important changes that come along with smart electric grids are a decentralization of the distribution grid operation on the one hand (i.e. small and distributed devices for supply, storage as well as controllable demand), but an inversion of the traditional *load-dependent generation scheme* on the other hand. While in traditional operation the controllable generation units are scheduled in order to meet uncontrollable demand, in smart power grid operation, controllable load devices shall be operated for enabling efficient usage of available generation resources (i.e. uncontrollable supply from renewables). Thus, a new scheme of *generation-dependent load* comes up in smart power grid operation, which has to be enabled through respective computational methods for operating optimal power flow control. Here, the efficient interplay of load control with renewable non-deterministic supply is a common objective [58, 95].

6.1.1 Generic Power Flow Control Formulation for Smart Grids

Section 2.2 stated the major challenges that come up with smart electric grids, leading to new requirements for power flow control. However, even if future smart grid implementations will come up with different shapes and characteristics, common and generic requirements can be formulated for developing smart power flow control technologies. Their basic motivation has already been discussed in Section 2.2 and shall now be considered in more detail referring to the OPF problem.

Increasing Quantity of Control Variables
Due to the change to *generation-dependent load* as well as the implementation of numerous distributed controllable devices, the quantity of control variables for power flow actions increases significantly, demanding for scalable computational methods that are capable of handling high amounts of variables.

Higher Volatility
On the one hand, control will be performed on lower power grid levels, where dynamics of single decisions are higher than on upright levels. On the other hand, the progressive penetration of fluctuating renewable small-scale power plants further increases the volatility of power grid behavior on these lower levels, necessitating control methods that provide power flow decisions quickly.

Need for Incorporation of Uncertainty
Similar reasons hold when considering uncertainty. Both the supply-side as well as the demand-side within the power grid cause probabilistic influences. This issue needs to be considered when deriving power flow actions in order

to guarantee robust decisions under uncertainty that preserve secure as well as reliable operation.

To discuss these issues in a formal way, a more detailed consideration shall be given with respect to the general optimal power flow formulation, more especially the set of dependent (state-) and independent (control-) variables as defined in Equations 3.5 and 3.6.

With the change from *load-dependent generation* to *generation-dependent load*, the definition of these sets changes. While load devices become partly controllable, a distinction has to be made between still dependent load variables P_L belonging to $\overline{x}$ and independent controllable load variables $P_{L,C}$ within $\overline{u}$. Thus, the cardinality of $\overline{x}$ decreases while the cardinality of control variables $\overline{u}$ increases. These vectors have to be reformulated for the smart grids scenario accordingly:

$$x_s^T = [P_{G_1}, V_{L_1}...V_{L_{NL}}, \boldsymbol{P_{L_1}...P_{L_{NLNC}}}, Q_{G_1}...Q_{G_{NG}}, P_{B_1}...P_{B_{NB}}] \tag{6.1}$$

$$u_s^T = [V_{G_1}...V_{G_{NG}}, P_{G_2}...P_{G_{NG}}, \boldsymbol{P_{L,C_1}...P_{L,C_{NLC}}}, T_1...T_{NT}, Q_{C_1}...Q_{C_{NC}}], \tag{6.2}$$

with NLC being the number of controllable load units and $NLNC$ giving the number of non-controllable load units respectively.
However, with such formal considerations, power flow control within smart grids can be described in a generic way. Referring to Section 2.2, major technologies that will influence power flow decisions are distributed storages, distributed switchable loads (like for instance cooling devices/air conditioners), and finally controllable electric vehicle charging infrastructures. No matter which of these technologies will finally be considered for power flow control, common to them is the property that they are abstracted through the variable $P_{L,C}$. While switchable load devices are obviously described by this variable, distributed storages are nothing else than a positive load when being charged and a negative one when injecting stored energy to the grid. For controllable electric vehicles the same applies. Thus, manifold technologies can be modeled by only assigning the appropriate value to $P_{L,C}$, but the defined OPF formulation is still valid. Finally, smart controllers for optimal power flow need to be able to handle the increasing cardinality of u_s^T, while being able to cope with dynamic as well as stochastic conditions.

Being now able to integrate the *generation-dependent* load scheme into the OPF formulation, the remaining issues such as higher volatility, increasing

uncertainty as well as high quantities of control variables can be handled by the discussed approach of policy function approximation when being applied to OPF. Here, robust (near-) optimal control can be enabled under dynamic environments, being highly scalable as shown in previous experiments.
Hence, a generic technology for dynamic stochastic OPF is available with the developed simulation-based methods for evolutionary policy approximation and shall now be applied to a smart grid scenario. Therefore, an electric vehicle charging control problem with consideration of fluctuating renewables will be defined, being a representative issue for smart power flow control.

6.1.2 A Generic Scenario: Electric Vehicle Charging Control

Controlling charging processes of electric vehicles - as already discussed in Section 2.2 - is highly promising to the smart grid concept and already relevant to practical implementations. Within this work, based on this idea a scenario for power flow control shall be developed that is representative for other smart grid applications too, hence, provides a suitable basis for testing the developed policy approximation techniques to smart grid optimization tasks.
While the general aims of controlled charging have already been shown, it is a representative scenario due to multiple reasons:

- Electric vehicles - plugged to charging infrastructures - represent spatially distributed and controllable load devices scattered throughout distribution grid areas.
- Like other controllable load appliances (e.g. thermal devices), charging load is governed by human individual behavior and needs, hence exhibits probabilistic situations.
- Even if being governed by such uncertain influences, individual behavior typically follows statistical patterns to a certain degree which enables their consideration through probabilistic models.
- Both high charging power values as well as long parking intervals at charging lots allow promising degrees of freedom for control aims.
- Being interrelated with non-deterministic and volatile renewable supply, complex power flow control scenarios can be constructed.

Hence, optimal electric vehicle charging control can be applied as representative issue, unifying manifold characteristics of future smart power flow control problems.

6.2 Controlled EV Charging - A Survey

6.2.1 General Aim

The mentioned integration of plug-in electric vehicles can be clearly stated as one of the hot-spots in power grid engineering nowadays. Being a possible chance for balancing fluctuating supply from decentral renewables, possibly critical effects may come up as well with the revolution to electric mobility caused by the non-deterministic nature as well as the comparably high load-values associated to charging processes.
Various researchers examine the problem of integrating electric vehicles optimally into power grids. The main idea is that some kind of central or decentral control influences the individual charging behavior in order to avoid critical peak loads on the one hand, and to flatten the overall load curve on the other hand [22, 23, 24, 57]. While these approaches generally have in common that they try to control the resulting charging power of each EV directly, a second trend needs to be mentioned that employs indirect control. Here, various works have been performed [29, 71, 99], using an indirect market-based approach for influencing EV charging. Final charging decisions are left to the end users, but electricity price tariffs are designed to encourage them to make charging decisions similar to what direct control of charging would aim at. Therefore a time-of-use (TOU) rate would charge a lower price for electricity during defined hours of the day. This could provide users with an incentive to delay charging, for example from peak time in the evening, when they arrive at home, to off-peak hours at night. Furthermore, real-time pricing (RTP) is a second common issue, which dynamically allocates prices based on the real-time electricity market and could provide EV users with even finer grained price information.
However, purely price-based control is not able to ensure secure power grid operation at all (i.e. enable guaranteed constraints satisfaction) since it still allows uncontrolled behavior, even if it is generally capable of globally shifting charging to off-peak hours. Hence, at the point where the charging decision is finally made, the actual physical condition of the power grid has to be taken into account for example in order to avoid overloading of devices, causing the fundamental need for direct control schemes.

6.2.2 Direct Control Strategies

Direct charging control manages the charging power of controllable EVs directly during operation. Several researchers already identified the abilities of direct control and therefore formulated manifold optimization problems, that can be grouped to a set of general classes when considering their objectives as well as the modeling assumptions. A holistic overview on existing approaches and common assumptions for optimal charging control shall be provided as follows:

Clement et al. [23] investigated influences on the distribution grid when proposing an optimal charging control that minimizes power losses. While scheduling the load of EV charging to off-peak time periods of the base load, the system peak demand as well as the overall power losses are reduced. With similar motivation, Sortomme et al. [100] defined an EV charging problem as a quadratic program, also considering a given distribution grid area, relationships among power losses, load factor and load variance. Popular works have also been published by Saber and Venayagamoorthy [93], that used EV charging control with particle swarm optimization for unit commitment. Sousa et al. [101] published a thorough overview of different approaches when applying simulated annealing for matching renewable supply with charging demand, also considering power flow computation. However, they assumed both consumer as well as renewables to behave according to deterministic patterns, disregarding uncertainties. All these works apply steady-state power flow computation, which is valid for power flow decisions respectively charging control actions on a time scale of some minutes and more.
With the deployment of smart grid technologies, controlling and scheduling of EV charging on a lower time scale may become possible as well, making charging control a fruitful ground for services such as frequency regulation. Using a V2G (Vehicle-to-Grid) aggregator model, Han et al. [40] proposed the scheduling of EVs for frequency regulation, where an optimization procedure was applied for deciding between charging and regulating with the target of optimal costs for the customer. With similar intention, [98] formulated a quadratic program for the optimization of charging schedules. Given these schedules, frequency regulation was performed in distribution feeders. All these works principally consider the power grid operation as primary motivation for optimizing charging decisions, building one general class of direct control approaches.
A much simpler but also common approach is to disregard power flow computation, but only consider the resulting load profile as grid-side metric to be sufficient. [51] and [119] for instance used charging control in order to

flatten peak-load in the overall resulting load profile..
Another class considers purely supply-oriented charging as central aim, with the objective of optimally using energy from renewables. Comprehensive discussions can be found in [10] [94] and [109]. Their common feature is the general aim of considering grid-wide renewable supply plants in order to optimally use their power output for charging EVs. Besides grid-wide control, similar works consider local objectives (like for instance at a single charging infrastructure that is connected to a small-scale supply plant) for scheduling the EV charging. A thorough overview can be found in [68], that also formulated a linear program for optimally matching photovoltaic supply with EV charging load at single grid nodes.
While this class of approaches is very promising for both enabling higher penetration rates of renewable supply on the one hand, but making electric individual traffic more ecologically attractive on the other hand, the general difficulty is the uncertainty that comes up with both renewable supply as well as EV user behavior, that both have to be considered for making valid control decisions but get disregarded in most existing approaches.

No matter if direct or indirect control, the mentioned approaches for optimization of charging decisions can generally be assigned to three classes as shown in Figure 6.1. While these classes are generally not disjunct (some approaches may consider two of these motivations together), they generally provide a rough overview of existing issues. However, for a real-world application of large-scale charging control, all these aspects have to be considered in common. For example, an end-user would aim at charging with lowest possible financial costs while at the same time the distribution system operator needs to maintain secure operation (i.e. has to guarantee various grid-parameters within certain ranges) under probabilistic supply as well as demand conditions. Thus, a desired technological solution should be able to consider a holistic optimization procedure unifying all these aspects.

Besides the formulation of the optimization problem itself, another major challenge is the modeling of the EV usage that characterizes electric vehicle charging load. Respective approaches typically assume usage patterns that are considered as being sufficient for obtaining time and duration of charging demand. Different approaches try to tackle this task using load profiles from statistical investigations [23], representations of individual behavior using Queuing Theory [109] or simulation via Monte Carlo Methods [100]. All these works generally have in common that they apply some mathematical procedure in order to compute deterministic profiles of electric vehicle charging load that are later used within certain optimization methods. A richer

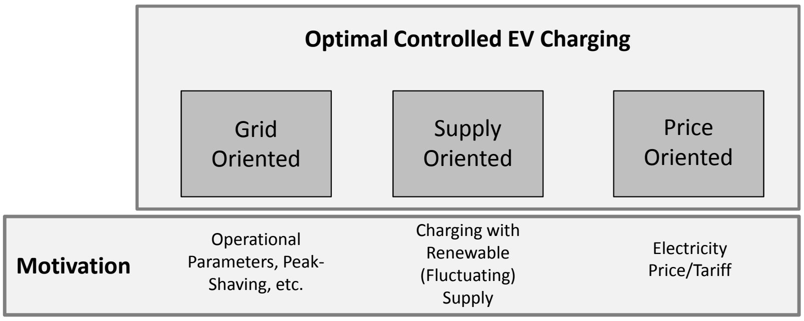

Figure 6.1: Classes of EV Charging Control Optimization

approach would be to directly integrate the EV users' uncertain behavior into the optimization process for deriving actions of both high quality as well as high robustness under dynamic and stochastic environments.

Using common assumptions as in related works, an optimization problem shall now be treated that characterizes most developments in literature.

6.2.3 Optimizing Deterministic Models

The optimization problem that will now be formulated represents typical assumptions of most works in literature, where a fleet of EVs is given with known driving patterns that should be charged with respect to some objective function, having a joint power limit. This power restriction represents for instance the limiting power capacity over all charging spots of a parking area, or the maximum free capacity that a power grid is able to use for charging. The objective function may be the minimization of charging costs (given some tariff data) or the optimal usage of renewable energy (given some forecast of for example photovoltaic injection). Thus, the optimization problem can be defined as follows:

Let $P_{EV_{t,n}}$ be the charging power of EV n at time step t. Considering a time horizon of length T, the aim is to minimize some utility (cost-) function throughout this time horizon, where at each time step the value of this function is cf_t. Knowing that all N EVs have a common power constraint p_{MAX_t} that provides the maximum power capacity that is available for charging at time step t, the control variables represent the "share" of this power capacity $u_{t,n}$ that each EV receives for charging. Thus,

$$P_{EV\,t,n} = u_{t,n} * p_{MAX\,t}. \tag{6.3}$$

While all EVs together are restricted to this common power capacity, each EV has a proper maximum charging rate $cr_{MAX\,t,n}$ that gives its local power constraint[1] which may be distinct throughout different EVs.
Since each EV needs to satisfy a specific minimum state-of-charge (SOC) when leaving the charging point, $E_{MIN\,n}$ gives the minimum charged energy at the end of the time horizon for each vehicle. Hence, all constraints can be formulated as:

$$\forall t \forall n : u_{t,n} * p_{MAX\,t} \leq cr_{MAX\,t,n}, \tag{6.4}$$

$$\forall t : \sum_{n=1}^{N} u_{t,n} \leq 1, \tag{6.5}$$

$$\forall n : \sum_{t=1}^{T_n} P_{EV\,t,n} * \Delta t \leq E_{MIN\,n}, \tag{6.6}$$

with Δt being the length of a time step t in hours, and T_n the time horizon (i.e. time steps until leaving) of each EV n. While Equations 6.4 and 6.5 give the local (of each EV) and global (throughout all EVs) power capacity constraints, Equation 6.6 ensures a minimum charged amount of energy when leaving.

Furthermore, the objective function is stated as:

$$F(\overline{u}) = \sum_{n=1}^{N} \sum_{t=1}^{T_n} [u_{t,n} * p_{MAX\,t} * \Delta t * cf_t], \tag{6.7}$$

where $\overline{u}$ gives all the control variables. While the objective function aims at minimizing costs of charging, the actually charged energy is multiplicated by the utility function value within each time step and summarized over all EVs and all time steps.

Since both objective function as well as all constraints are linear, exact methods can be applied for solving the optimization problem. As in most works in literature, all data is assumed to be given deterministically. This

[1] The maximum charging rate of an EV may be constrained due to the installed charging infrastructure (i.e. the technical limitations of the charging point) or the charging controller of the EV itself. In the latter case, the maximum charging rate may vary along time.

means that for each vehicle its time of departure, its arrival-SOC (and so the needed amount of energy until departure), the common power capacity p_{MAX_t} as well as the individual maximum charging rate $cr_{MAX_{t,n}}$ are known. As utility function, the electricity tariff can be taken that varies over time.

For following analysis, the defined optimization problem is implemented in Matlab using Simplex solver within the *linprog()*- function. A more detailed discussion on the implementation can be obtained in Appendix A.7. Now, optimization experiments shall be performed for illustrating the defined problem.

Experiments' Data

As underlying motivation, a charging scenario shall be treated that may occur at a company with a car pool of $N = 8$ EVs that are used by its employees for commercial reasons. In actual developments, such scenarios are seen as primary applications for first-generation electric vehicles, as being investigated in an Austrian demonstration project [80].
In such a scenario, the situation may occur that all EVs come back from their morning-trips around noon and need to be charged until the next trip begins (such situations actually occur with high probability as shown in the CMO project [80]). Thus, the following arrival- and departure-times result as shown in Table 6.1.

EV #	**Arrival**	**Departure**	cr_{MAX} **[kW]**	E_{MIN} **[kWh]**
1	12:00	16:00	10	8
2	10:30	13:30	10	8
3	12:30	15:30	10	6
4	11:00	15:00	10	7
5	11:00	14:30	10	9
6	13:00	17:00	10	5
7	12:30	16:45	10	8
8	12:30	17:30	10	8

Table 6.1: Vehicle Data for Demonstration Scenario

For simplicity reasons, it is assumed that all charging points as well as all EVs are equipped with the same technology, i.e. the maximum charging rate of $cr_{MAX} = 10kW$ is constant[2]. After arrival, the users prefer to fully recharge their batteries, hence, E_{MIN} gives the consumed energy from the

[2]Batteries are assumed to be ideal, taking a constant charging current.

last trip (which took between 20 and 32 km assuming a consumption of 0.25 kWh/km as being typical for actual market-available EVs).

The common power capacity p_{MAX} is defined to be 25kW (installed power for charging infrastructure), which is reduced to 17kW during office hours (8:00-19:00). Within this scenario, the objective shall be to minimize financial costs for charging. For doing so, a dynamic electricity tariff is assumed as utility function cf with regard to electricity spot market prices. Both global power capacity p_{MAX} (red) and utility function cf (electricity price, blue) are shown in Figure 6.2.

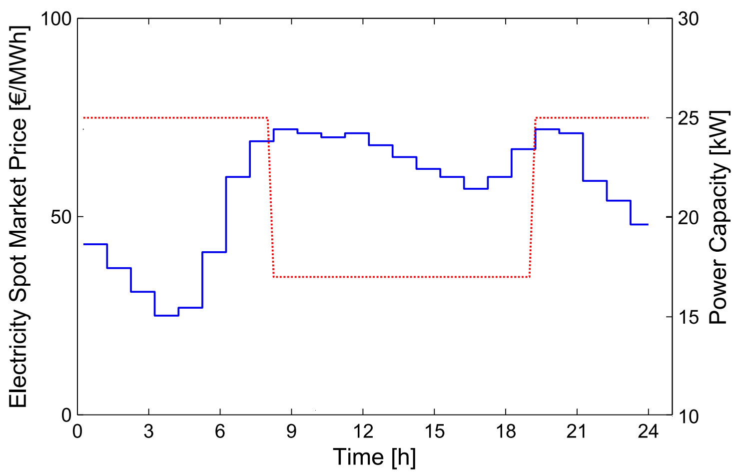

Figure 6.2: Electricity Price Utility Function and Common Power Capacity Constraint

For the given situation, the optimization problem can be solved exactly according to the definition above. Δt is chosen to be 15 minutes, which is a common time granularity for control in low voltage distribution networks.

In this show case, the consideration of local renewable supply is neglected. However, this can easily be integrated by extending the objective function with some incentive-term that favors charging with renewable energy. The resulting objective function would result to:

$$F(\overline{u}) = \sum_{n=1}^{N} \sum_{t=1}^{T_n} [u_{t,n} * p_{MAX_t} * \Delta t * (cf_t - p_t)], \qquad (6.8)$$

where p_t gives the profit at time step t for favoring charging with renewable energy. The unit of p_t needs to be the same as that of cf_t, thus, $[\frac{Euro}{kWh}]$.

Experimental Results

The solution to this optimization problem shall now be discussed. For comparison reasons, Figure 6.3 gives the resulting charging power that would result when using a fair scheduler for charging control instead of an optimization procedure. This fair scheduler considers the common power capacity p_{MAX_t} at time step t and distributes this capacity fairly to all EVs that are demanding energy (i.e. are plugged to the charging infrastructure and are not fully charged yet) at this moment. The parking durations (i.e. the potential time intervals for charging) are indicated for each EV with a solid blue line, where the charging power is given in the range $[0, cr_{MAX}]$. These charging schedules resulting from a fair scheduler tend to charge at the earliest possible time, disregarding the electricity tariff.

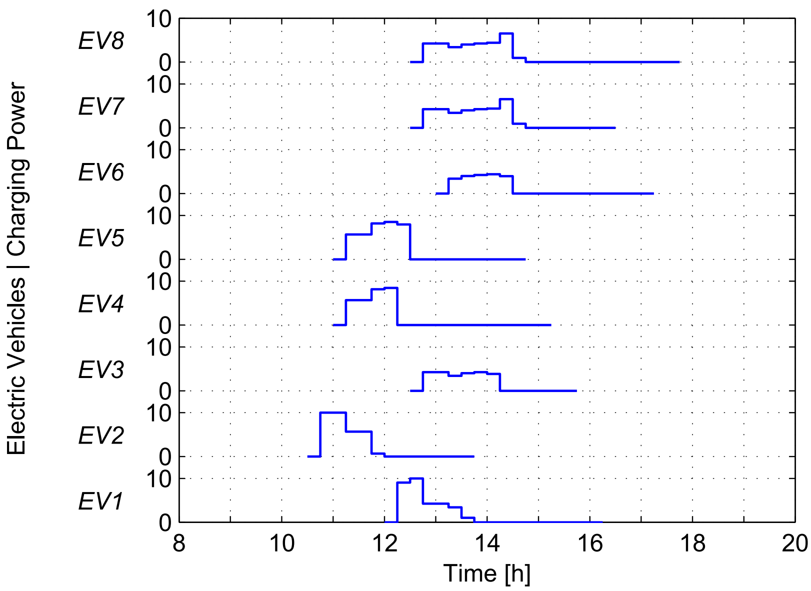

Figure 6.3: Results of Fairly Scheduled Charging Control

Figure 6.4 gives the optimized charging control according to above problem definition, obtained with Simplex solver in Matlab. Similar to the fairly scheduled control, the charging demand is satisfied with respect to all constraints, but the utility function (i.e. price for charging) is minimized. Comparing the resulting costs for charging all EVs, the fair solution comes to 3.98 Euro while the optimal solution leads to costs of 3.64 Euro[3]. Since the time

[3]Notice that stock market prices are used as utility function; real-world consumer prices

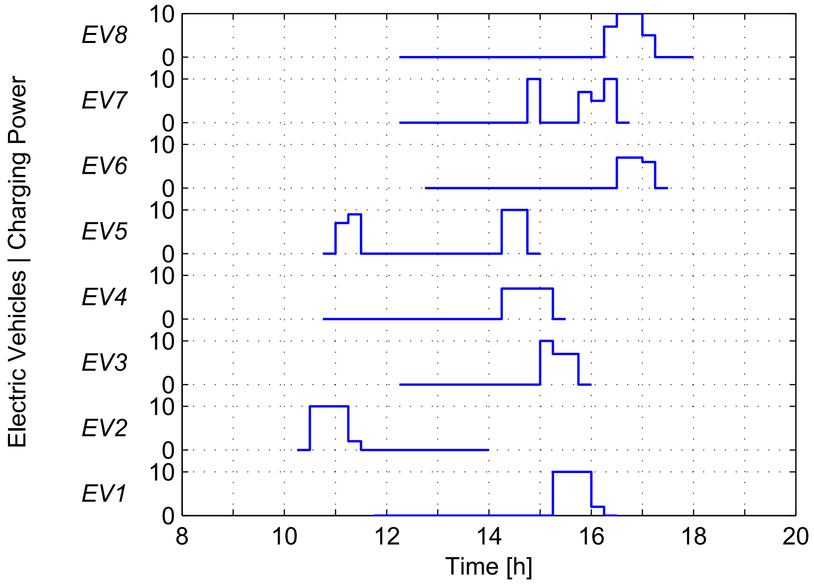

Figure 6.4: Optimal Charging Control Solution

intervals for charging are situated at times of high costs, this difference is quite small (around 9%) and would be higher when considering for example parking at night.
However, the optimized solution strictly uses "cheap" time steps for charging (which is obvious), thus, it may occur that charging is shifted to time steps immediately before departure (consider for example EVs nr. 1,3,4,5,6,7 in Figure 6.4). Hence, resulting schedules are only valid as long as all departure times can be predicted exactly (which is the assumption in most works in literature). In the case that certain EV users would leave the parking lots earlier than predicted, their EVs' batteries would not be charged fully, violating the E_{MIN}-constraint (Equation 6.6).
Hence, in an uncertain environment where EV users behave according to individual decisions, such a deterministic optimization is not appropriate. With such a scenario, it can illustratively be shown that the integration of uncertainty as well as of the system's dynamics is of fundamental necessity for deriving valid control decisions.

are a multiple higher depending on taxes and grid operation costs in the respective region.

Acknowledgements

Many of the prior considerations are related to the Clean Motion Offensive demonstration project [80], where manifold strategies for controlled charging as well as its real-world applicability are evaluated. The numerical results together with underlying scenario data have been collected as described in [48]. A more detailed discussion on the Matlab-implementation of this optimization problem can be found in Appendix A.7.

6.2.4 The Need for Power Flow Consideration

Another fundamental issue is the consideration of the physical power grid model in order to not strain distribution equipment and maintain secure grid operation when making charging decisions. Many researchers therefore incorporate constraints into the optimization process to restrict for instance the common charging power capacity of all EVs to a maximum value (such as p_{MAX} before). However, when optimizing a system like a distribution grid with complex interrelations between numerous instances of demand, supply as well as distribution equipment, a more thorough approach is needed that considers the actually resulting (local) power flows throughout the entire grid for each possible operation point. Here, a general power constraint (such as p_{MAX}) that neglects local conditions in the grid may be too unprecise and would even lead to local constraint violations. Moreover, probabilistic conditions need to be analyzed in more detail in order to find an efficient incorporation of EVs, domestic and commercial loads, distribution equipment and renewable supply sites.

A simple demonstration model should substantiate the idea that both thorough power flow modeling of the network as well as the consideration of uncertain conditions are essential for computing valid charging decisions. A synthetic model is set up as shown in Figure 6.5 that describes a quite simple form of a distribution feeder. Detailed model data is given in Appendix A.8 of this document and is not necessary now for the following demonstration. There are 3 load buses in the system (3,4,5) that are connected to a higher power grid level via Bus 1. Bus 3 and 4 serve domestic areas, each of around 40-50 households. At bus 3 there are additionally charging stations installed for EVs that are used by its residents. At bus 5, a large commercial area is connected to the grid that is equipped with charging infrastructure and an additional photovoltaic plant with peak power of 25 kW. Since in the near future mainly mode 2 charging with up to around 10kW will be applied as charging mode, therefore, a charging power of maximum 10kW is assumed

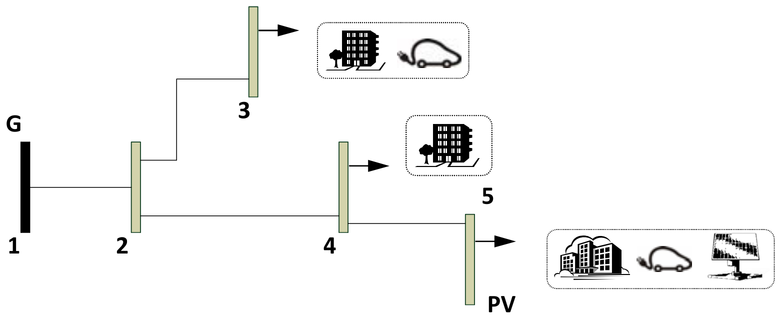

Figure 6.5: Experimental Distribution Feeder

Variable	**Constraint**	**Variable**	**Constraint**
Power Flow Through Line 1-2	$\leq 500kW$	Power Flow Through Line 2-4	$\leq 300kW$
Power Flow Through Line 2-3	$\leq 200kW$	Power Flow Through Line 4-5	$\leq 200kW$
Charging Power per EV	$\leq 10kW$	Cumulated Charging Power all EVs	$\leq 150kW$

Table 6.2: Constraints Demonstration Scenario

for each EV in the system. This specification is related to a three-phase charging process with 400V and 16A, as exemplarily possible when using a Mennekes VDE (Type 2) plug connector[4]. 20 EVs are existing within this system and behave individually, demanding electric power when getting charged. In order not to overload the electric distribution equipment, a maximum cumulated charging power of p_{MAX} = 150kW is defined for all EVs which forms a possible constraint to optimization algorithms. For guaranteeing secure distribution grid operation, the following simple constraints are formulated in Table 6.2 for restricting power flows over cables.
Three experiments will be discussed now in order to get an idea of different states of this system during a synthetic day. For reasons of simplification, all exact power flow results are listed in the respective appendix, only the necessary key-values are used now.

Experiment 1: Weekday, 8:00 p.m.
This experiment should show a typical weekday at around 8:00 p.m., where 15 of the 20 EVs have already left the owner's working location at bus 5. 10

[4]This configuration, as existing for example when charging the well known Tesla Roadstar or many other actually market-available EVs, is certainly one of the most important technical specifications in this field in actual developments.

of them already arrived at home, were plugged to the charging infrastructure immediately after arrival and are still charging. 5 EVs are on the streets or already reached a full battery SOC, thus do not demand electric power at this time. The remaining 5 EVs belong to owners that are at bus 5 and charge their cars for later departure. Since 8:00 p.m. is peak-hour in most distribution grids, the base load caused by domestic consumers is high. The photovoltaic plant supplies only a quite small amount of power, 5 kW are assumed. The total load data is given in Table 6.3.

Bus Nr.	Base Load [kW]	EV Charging load [kW]
3	90	100 (10 EVs at 10 kW each)
4	91	
5	60	50 (5 EVs at 10 kW each)

Table 6.3: Load Data Experiment 1

The power flow calculation provides the following results for line flows in Table 6.4.

Cable	Power Flow [kW]	Cable	Power Flow [kW]
1-2	388.0 (of 500)	2-4	197.1 (of 300)
2-3	190.2 (of 200)	4-5	105.5 (of 200)

Table 6.4: Power Flow Result Experiment 1

As a result, no violations occur when considering the constrained active power flows over distribution cables. Hence, all plugged-in EVs are allowed to charge with their maximum power of 10kW from a power grid point of view. The constraint that all EVs together have a cumulated charging power of maximum 150kW holds.

Experiment 2: Weekday, 1:00 p.m.

The second experiment shows another typical weekday, but around 1:00 p.m., where some part-time workers already arrived at home or are on the streets, the rest remain at work. Since even if it is cloudy, supply from the photovoltaic plant is quite high at this time (15 kW), all residing EVs at the commercial center are charging. 13 EVs are assumed to remain at bus 5 at this time and are plugged-in for charging. 4 EVs are on the streets or already fully charged at bus 5, 3 EVs belong to part-time workers, immediately arrived at home and plugged in at bus 3, 1 of them is already at home and fully charged. The resulting load data is given in Table 6.5. The base-load

Bus Nr.	**Base Load [kW]**	**EV Charging load [kW]**
3	65	20 (2 EVs at 10 kW each)
4	91	
5	93	130 (13 EVs at 10 kW each)

Table 6.5: Load Data Experiment 2

around noon is quite at peak both for the commercial center, as well as for domestic customers.
Executing power flow calculation, the resulting line flows are listed in Table 6.6:

Cable	**Power Flow [kW]**	**Cable**	**Power Flow [kW]**
1-2	387.0 (of 500)	2-4	301.0 (of 300)
2-3	85.0 (of 200)	4-5	209.0 (of 200)

Table 6.6: Power Flow Result Experiment 2

It can be seen clearly that even if only in sum 150 kW are used for charging the EVs (i.e. the p_{MAX} constraint is satisfied), cables 4-5 and 2-4 are overloaded. Thus, a simple global constraint p_{MAX} on maximum cumulated charging power for all EVs in the system is not sufficient, a thorough consideration of the resulting power flows in the grid is necessary for evaluating the system state. In this case, intelligent charging decisions are necessary for decreasing the charging power of single EVs in order to not stress distribution equipment.

Experiment 3: Weekday, 1:00 p.m., Probabilistic Consideration

The last simple experiment should show the influence of uncertain behavior within the system. Experiment 2 is used with equal load data, but uncertainty is added at the supply side. What if, for example, the atmospheric condition is very volatile at this time and solar irradiance increases rapidly because of decreasing cloudiness, yielding higher power supply from the photovoltaic plant of for instance 25 kW peak injection? The resulting line flows come up to the values in Table 6.7.
Now, due to increased supply from the fluctuating renewable plant at bus 5, line flow constraints are all satisfied, even if the load data is equal to experiment 2. Thus, it is clear that the degree of uncertainty in a given system has to be added for finding valid charging decisions. In this case, uncertainty is only given at the supply side, but in real world it additionally occurs at the demand side as well.

Cable	Power Flow [kW]	Cable	Power Flow [kW]
1-2	376.8 (of 500)	2-4	290.9 (of 300)
2-3	85.0 (of 200)	4-5	198.9 (of 200)

Table 6.7: Power Flow Results Experiment 3

Concluding these three simple experiments within a synthetic test model, it can be clearly stated that modeling the power flows in a given grid thoroughly is essential in order to find valid charging decisions. Interrelations in an electric network are too complex as to just form a constraint like restricting the cumulated charging power of all vehicles to a maximum value p_{MAX}. Additionally, variabilities of uncertain conditions in the system like the supply of renewables have to be taken into account in order to preserve robust system operation.
Since it has been shown that accurate simulation of the distribution grid's state with all its uncertainties is fundamental for validating the situation, an optimization approach is needed that is able to incorporate this into the computation of charging decisions.

6.2.5 Conclusion: Lack of Existing Work

As stated, plenty of research has already been performed to optimize EV charging decisions, but there are still open issues that have to be addressed in more detail. One problem is the uncertain behavior of the system, which is caused both by probabilistically behaving EV end-users at the demand side as well as fluctuating and non-deterministic power plants on the supply side. Many approaches aim at integrating both renewables' supply as well as EV users' power demand into optimization problems, but they mostly derive some deterministic patterns of them for building the problem-model, disregarding uncertainties as well as the dynamics of these influences. Therefore, an optimization method is needed that incorporates both the uncertainty as well as the dynamics of the system when computing optimal charging decisions.
Another important issue is the consideration of actually resulting power flows as well as local conditions throughout the entire system. Even if integrating power flow computation into the optimization problem formulation is a complex issue, it is necessary for deriving valid control actions under consideration of secure and reliable power grid operation.

Hence, a more holistic optimization problem formulation needs to be performed, which is capable of integrating both the demand side's as well as the supply side's uncertainties. The ability of deriving decisions with respect to actually resulting power flows in the system is an additional requirement in order to maintain reliable power grid operation during charging control. Such a problem will have to be tackled with an appropriate method for stochastic and dynamic optimization, which is provided by the developed simulation-based evolutionary policy approximation scheme that shall later be applied.

6.3 EV Charging Control: Holistic Optimization Problem

Until now, EV charging control has been identified as generic problem scenario for future smart grid operation and optimization tasks. While reviewing existing approaches in literature and investigating some representative scenarios, the main requirements have been worked out that need to be satisfied for making valid charging decisions, which cannot be fulfilled all together with existing approaches:

- Inclusion of power flow computation for maintaining reliable power grid operation during charging control.
- Providing a holistic system representation, containing appropriate representations of the EV user side, the physical power grid as well as the probabilistic demand side.
- Consideration of systems' uncertainties both existing at the demand as well as at the supply side.
- Dynamic optimization for providing quick and robust control actions.

Therefore, the EV charging control optimization problem from Equations 6.3 - 6.7 needs to be reformulated and extended as follows.

6.3.1 Formal Definition

A formal description should clarify the statement of the holistic EV charging control problem. Test scenarios derived from this description will be provided with additional data later in order to build the base for experimental investigations.

Given a fleet of n EVs within some distribution grid area, over a future time interval of $t = 1...T$ time steps optimal values for $P_{EV\,t,n}$ shall be computed that give the active charging power of each EV. Assuming that the system's behavior can be predicted to a certain degree within the considered time interval, this prediction is related to uncertainty of various random system variables (i.e. photovoltaic injection, EV departure times, etc.). Thus, the aim is to optimize control variables $\overline{u_{EV}}$ with the objective of minimizing the expected utility function value, hence:

$$min \sum_{t=1}^{T} E(F(\overline{u_{EV\,t}})). \tag{6.9}$$

At the end of the considered time interval, each EV must have received a specific amount of energy for satisfying its daily demand. As additional load caused by charging of electric vehicles could endanger power grid security, operational power flow constraints have to be satisfied that ensure secure distribution grid operation within each discrete time step[5] t. Steady- state security constraints need to be considered as already defined in the general OPF formulation (see Equations 3.8 - 3.11) for ensuring lower and upper bounds for generator real and reactive power output, maximal power flows over lines/branches as well as admissible voltage deviations within each time step. These constraints need to be considered by performing power flow computations in order to integrate the distribution grid operation into charging control optimization as postulated before.

In order to guarantee feasibility of solutions from the EV fleet point of view, different restrictions have to be satisfied as well which can be stated as inequality constraints: at the end of the scheduling horizon, the energy demand of each single EV has to be satisfied as before, thus the constraints

$$\forall t \forall n : P_{EV\,t,n} \leq cr_{MAX\,t,n}, \tag{6.10}$$

$$\forall n : \sum_{t=1}^{T} P_{EV\,t,n} * \Delta t \leq E_{MIN\,n}, \tag{6.11}$$

still hold for each EV. Therefore, the sum of charged energy over all discrete time steps t has to reach a minimum to satisfy user's needs as well as the charging infrastructure's physical restrictions.

[5] Notice that charging decisions will be made on a time scale of more than some minutes, thus, once more steady-state power flow considerations are sufficient.

Once more, the set of constraints is considered through penalizing the constraint violation CV with a weight w_{CV}. Assuming that the vector of control variables contains the charging power value for each EV at the specific time steps, i.e. $P_{EVt,n} = \overline{u_{EVt}}[n]$, the final objective function can be stated as follows:

$$min \sum_{t=1}^{T} [F(u_{EVt}) + \overline{w_{CV}} * \overline{CV(u_{EVt})}], \tag{6.12}$$

Since this optimization task is related to uncertainty, the aim is to optimize the expected fitness function value (as well as the expected constraint violation) rather than the exact one (which indeed is not observable), hence Equation 6.12 becomes to:

$$min \sum_{t=1}^{T} E[F(u_{EVt}) + \overline{w_{CV}} * \overline{CV(u_{EVt})}], \tag{6.13}$$

In the end, this holistic optimization problem formulation considers both the EV charging problem formulation as before, but additionally integrates power system operation aspects. Since uncertain system variables have to be integrated into the optimization of control decisions over time, it can be expressed as a dynamic stochastic optimal power flow (DSOPF) problem. Here, simulation-based evolutionary policy approximation is applied as being the core technology of this work, where the estimated performance of a solution candidate (Equation 6.13) is computed through simulation.

6.3.2 Simulation Model

Now, the simulation model of the system shall be described in detail, consisting of the distribution grid model together with renewable supply plants, and the traffic model that represents the electric power demand over time of EVs.

Distribution Grid Power Flow

One great aim of integrating the distribution grid model into the holistic simulation model is to derive the resulting state of the physical power grid for a solution candidate. Therefore, power flow simulation is applied for computing the electric voltage and current magnitudes throughout the entire grid for ensuring appropriate operation of distribution equipment with respect to defined OPF security constraints (Equations 3.8 - 3.11). Additionally,

this simulation is necessary in order to completely evaluate the total power consumption (including losses) within a given state that indeed affects the objective function value.
The mathematical models applied for representing the distribution grid are presented now in order to provide a complete formulation of the problem:

Branches are modeled using the standard "single-phase line model", with complex series impedance and line charging capacitance. The complex current injection at the beginning and the end of each branch is computed with Equation 6.14, using the branch admittance matrix Y_b that takes real- and reactive part of the impedance as well as the line charging capacitance. In case of an ideal phase-shifting transformer at the beginning of a branch, phase shifting angle and tap ratio are added for computing the branch admittance matrix.

$$\begin{bmatrix} I_{be} \\ I_e \end{bmatrix} = Y_b \times \begin{bmatrix} V_{be} \\ V_e \end{bmatrix} \tag{6.14}$$

Generators and loads are modeled as complex power injections at specific buses. For instance, for generator j, the injection is given by

$$S_{G,j} = P_{G,j} + j \times Q_{G,j}.^{6} \tag{6.15}$$

A complex load is a negative constant generator. Both for loads and generators, a constant power factor is assumed during all experimental investigations.
The network equations can be obtained by formulating a general branch admittance matrix over all branches b as well as a general bus admittance matrix over all nodes (buses) j in the system. Using these matrices, resulting nodal injections S_j can be derived, such that a general AC-power balance equation for each bus j can be stated, split into its real and reactive parts:

$$P_j(\Theta, V) + P_L - P_G = 0 \tag{6.16}$$

$$Q_j(\Theta, V) + Q_L - Q_G = 0 \tag{6.17}$$

For solving the AC load flow, slack bus method is applied according to [116], where load-buses are represented as so called PQ-nodes with fixed real- and reactive injection. Generator-buses are PV-nodes with fixed real-power injection and voltage magnitude, furthermore, a single slack bus is defined with

[6] Notice that the index j gives the bus number, while the other j indicates the imaginary part.

fixed voltage angle Θ and real power slack. Newton-Raphson method is applied for solving the $nPV+2nPQ$ dimensional system of nonlinear equations. Further information about solving the steady-state AC load flow can be obtained from the literature [116] in general and in the MATPOWER-manual considering the specifically used implementation [120].

Probabilistic Renewable Power Supply

As stated before, the combination of EV-charging as controllable load with fluctuating supply from renewable plants is a promising future scenario for smart electric grids. Therefore, these renewable power plants have to be integrated into the distribution grid model appropriately, in order to represent their influence on the power flows in the grid and the resulting charging decisions respectively. Each plant is modeled as a generator according to Equation 6.17, where its real-power value depends on a probabilistic distribution. Since supply from renewables can be forecasted with partly high accuracy, this distribution describes the uncertainty of the forecast.

Wind Power Plants

Prediction of available power is a major concern for many stakeholders in the electric power supply sector like for instance power grid operators or electricity market retailers. Therefore, plenty of effort has been invested in building accurate methods for forecasting. Especially the modeling of wind power is a highly challenging task, since wind speed depends both on atmospheric and meteorologic features as well as geographic properties of the surface around wind power plants. Therefore, forecasting techniques and features used for building models depend on the modeling horizon. While short-term prediction of wind power is primarily applied using purely data-based time-series methods for modeling wind speed, mid-term and long-term forecasting is being performed by using numerical weather prediction with meteorologic data and analytical modeling techniques [15] [70]. Generally for both short as well as medium and longterm forecasting, accuracies around 10% to 15% are reachable, depending on the specific site's characteristics.

Within probabilistic power flow studies, intermittent wind power supply is generally modeled using Weibull distributions together with a physical model of the wind power plant [109]. For deriving the resulting power output of the plant in herein experiments, within each simulation run the forecasted wind speed value is sampled from a Weibull-distribution. With this resulting wind speed value, the corresponding power output can be computed using the plant's power curve. The power curve describes a simplified relation between the wind speed and the electric power output of a specific wind power plant considering its physical characteristics within a simple two-dimensional function [70]. For most applications, this simplification is a sufficient representation. The power curve of a plant is usually available with its data sheet and is valid for the herein performed system modeling, Figure 6.6 gives an exemplary one. Here, the so called cut-in speed (where the mechanical torque exerted by the wind is high enough to make the turbine blades rotate) is around 2 m/s. There is an exponential increase in power output (which primarily makes the integration of wind power plants critical for grid operation) until the plant reaches its rated output at around 16 m/s, where it reaches the maximum capacity of the generator (40 kW in this case). For higher wind speed values, the plant limits the output which is most often performed by adjusting the blade angles (but other techniques exist as well). At around 25 m/s, most plant technologies cut-down the generation. Since the mechanical torque exceeds critical limits and could endanger the construction, this is called cut-off speed.

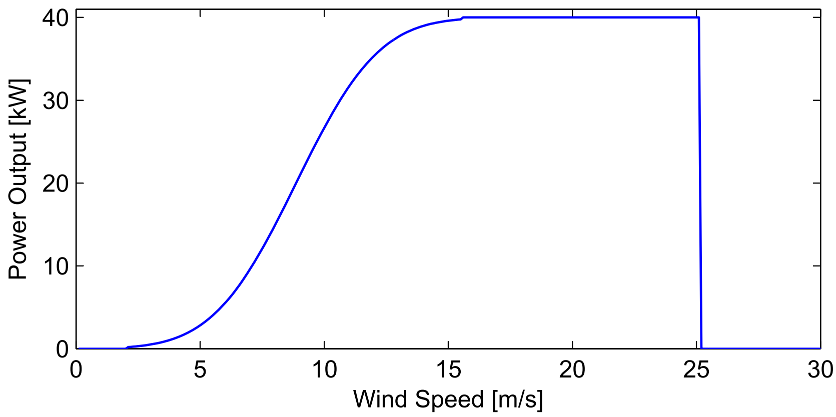

Figure 6.6: Exemplary Power Curve

Hence, for simulating the output of a power plant, a predicted wind speed value (sampled value from the Weibull distribution) yields the real power output by applying it to the power curve.

Photovoltaics

In principle, wind power plants are the more critical issue in electric power grid operation, which is caused by the facts that wind speed is more fluctuating (both due to meteorologic reasons as well as because of the exponential increase of wind speed as illustrated above) than solar irradiance on the one hand, but their installed capacities are much higher than for photovoltaic (PV) plants on the other hand. Nevertheless, PV plants have to be considered as well since they form the second important renewable source beside wind power and provide well predictable but still uncertain power output.

Since modern PV stations are typically equipped with power point tracking controls, their output can be assumed to be linear with the solar irradiance, i.e. no mapping is needed such as a power curve. Usually reachable accuracies for PV generation prediction lie around 10% as well, hence, a probability distribution has to be added for describing this uncertainty. For PV output modeling, a normal distribution can be applied similar to [34] for describing the accuracy of the forecasted power supply, where the assumed mean value represents the predicted power output.

All these small-scale distributed power plants are integrated into the distribution grid power flow model by just assuming them to be a complex injection according to Equation 6.15 and adding this injection to an appropriate bus. Similar to loads where a constant power factor is assumed, for renewable generation units as well a constant relation between active- and reactive-power injection is defined. This relation is taken from the original model definition (i.e. the IEEE 33-bus test feeder in later experiments).

Electric Individual Traffic Charging Demand

The consideration of individual traffic is necessary in order to model at which time single EVs will be parked and ready for charging on the one hand, but further for evaluating if each single vehicle received the demanded energy on the other hand. This model has to be accurate to specify location and charging demand of each single EV, but has to be efficient for supporting low computational effort for solution candidate evaluation. In [51], discrete event simulation has been proposed for traffic behavior simulation. The advantage of using this simulation method is that with probabilistic queues an exact model of waiting EVs can be built that queue at the same charging spot. Thus, interrelations between single entities (EVs) can be described very accurately. Since actual surveys on near-future electric mobility show that the lions' share of charging processes will take place at domestic charging spots and that charging infrastructure will be available in higher quantity than EVs exist, it is valid to assume that each EV has it's own charging spot once it reaches a location where charging infrastructure is available. Hence, a simpler simulation approach can be applied.

In [44] and [52] the simulation of randomized driving patterns has been introduced where for each EV a separate driving profile is generated according to its usage pattern. Such a synthetic driving profile is simulated at each run for each EV that represents time interval and location of being parked and plugged to a charging infrastructure. An exemplary profile of a vehicle over a day is shown in Figure 6.7, which is either being parked at home, at any location at work or free time, or is currently on the way.

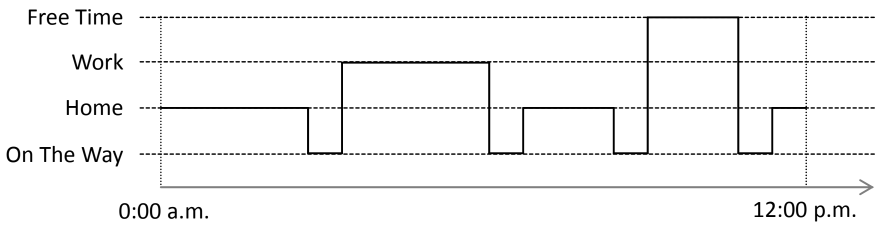

Figure 6.7: Driving Pattern Representation

For each vehicle, at the beginning of the simulation the necessary data for generating these profiles needs to be available, namely the general pattern that gives information such as estimated departure time in the morning, typical work duration, or duration for driving from work to home location. During simulation, all these figures get randomized in order to obtain real-world uncertainty of individual decisions and influences from the environment.

If the vehicle is parked at any location, it is potentially ready for charging as far as charging infrastructure is available. Therefore, additional probabilities are assumed that model the availability of charging equipment at potential locations, namely p_{home} for parking at home, p_{work} for parking at work as well as p_{free} for locations in free time.

Real-world data about driving behavior is used as essential information for the simulation model that will be described in more detail when setting up scenarios for experimental investigations. A thorough discussion on the traffic simulation as well as the applied assumptions for later experiments can be found in Appendix A.10.

Interaction of the Simulation Model Components

The so defined simulation model components aggregate to the system representation as shown in Figure 6.8.

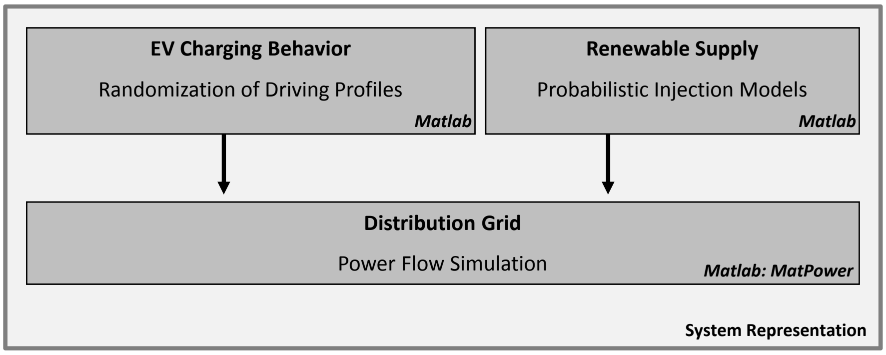

Figure 6.8: Holistic Simulation Model

From the electric vehicle charging behavior simulation, the resulting charging load is sampled and inserted into the electric power grid model. Sampling solar irradiance and wind speed from the probabilistic models of renewable supply, the respective power generation is computed before updating the distribution grid simulation model and computing the resulting power flows. Thus, a complete state-description of the entire grid and its distribution components is obtained through the power flow solution, which is the only deterministic component in the system representation. Since the complete simulation is governed by random variables, i.e. delivers varying outputs for same inputs, each solution candidate's evaluation has to be sampled a sufficient number of times in order to get a valid estimate of its performance. A complete illustration on the optimization procedure is provided in Figure 6.9, where the metaheuristic algorithm is abstracted to a single block in order to highlight the fact, that this procedure is independent from the concretely applied search procedure.

So far, only the system representation has been discussed without consideration of the concrete outlook of a solution candidate. The following section shall discuss possible solutions in more detail when finally applying policy function approximation for charging control.

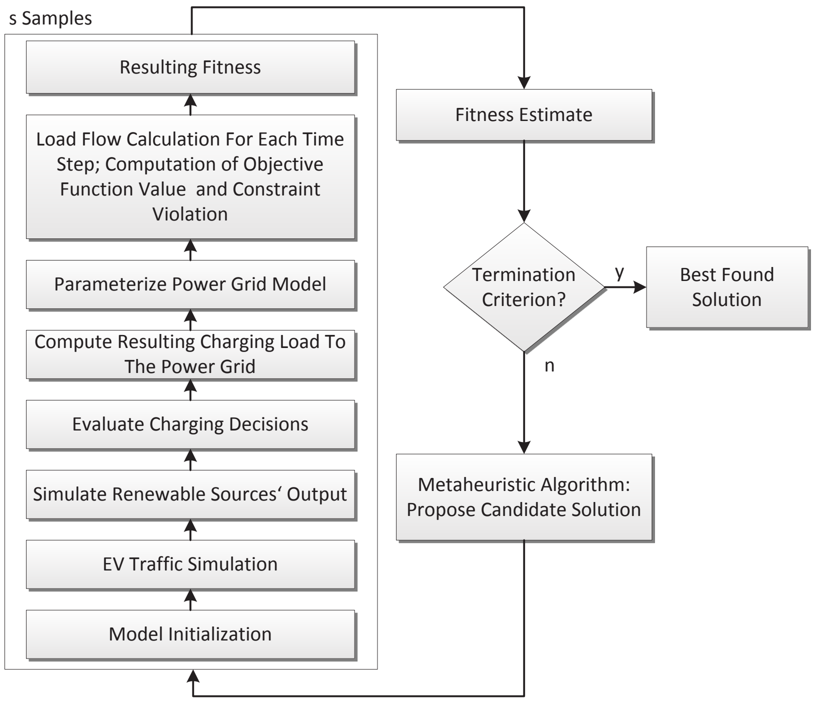

Figure 6.9: Flowchart Simulation Optimization for EV Charging Control

6.4 Solving the Holistic Charging Control Optimization Problem

The defined optimization problem shall now be treated using evolutionary simulation optimization. While real-valued optimization in the form of deterministic lookahead optimization will be applied for comparative reasons, simulation-based policy function approximation as technique for dynamic stochastic optimization once more comes into play as promising technology.

6.4.1 Simulation-Based Lookahead Optimization

As in the optimization of a simpler deterministic model on EV charging with exact methods (see Section 6.2.3), the aim is to directly use the charging power values $P_{EV1,1}...P_{EVT,N}$ as control variables which are optimized over a predicted time interval T. Similar approaches as discussed in the introductory chapters on smart grids as well as on optimization under uncertainty (see Sections 2.2 and 3) that perform kinds of lookahead optimization ensure optimal behavior beforehand, considering forecasted states of the system (see Figure 4.1) deterministically. This approach seems to be the easiest and most

intuitive way for tackling this problem, but may easily lead to solution space explosion and lacks of flexibility for making adaptive decisions to volatile situations. When for example handling the $N = 300$ EVs within a given distribution grid area with $T = 24$ time steps to be controlled in a predictive manner, $300 * 24$ control variables for $P_{EV\,t,n}$ will be needed. For suchlike or even bigger scenarios, the dimension of the control variables vector may exceed a manageable problem size for evolutionary algorithms. In this case, clustering can applied as the author described in [44], where agents (EVs) with similar behavior and similar local appearance in the power grid get clustered into groups of size m using the same solution, thus reducing the solution space drastically.
Beside the exploding problem size, deterministic lookahead optimization additionally shows the disadvantage that it considers volatile behavior of the stochastic system in advance, being very inflexible to dynamic conditions. Thus, such an optimization approach is incapable if incorporating anticipatory and flexible behavior into charging decisions, which would be necessary when considering a stochastic and dynamic system. However, this method provides promising reference solutions and shall be applied for comparison reasons.

Hence, the solution in form of a vector of control variables as well as the final fitness function come up to:

$$min \sum_{t=1}^{T} E[F(\overline{u_{EV}})_t + \overline{w_{CV}} * \overline{CV(\overline{u_{EV}})_t}], with \tag{6.18}$$

$$\overline{u}_{EV}^T = [P_{EV\,1,1} ... P_{EV\,T,N_c}], \tag{6.19}$$

where N_c gives the number of clusters with $N_c = N/m$.
Finally, a real-valued optimization problem needs to be solved, with a $T * N_c$-dimensional solution vector.

6.4.2 Simulation-Based Policy Function Approximation

At the very beginning of this Chapter 6, the general outlook of smart power flow control tasks has been discussed, deriving that manifold future smart grid technologies just lead to extensions of the OPF problem. Hence, besides using lookahead optimization for directly computing the solution vector $\overline{u_{EV}}$ of charging decisions for EVs, simulation-based evolutionary policy approximation can be applied as technology for DSOPF-related charging control.

In this application, the aim is to find some flexible policy $p_{EV}(\bar{i})$ that derives (near-) optimal charging decisions (=actions) at runtime, avoiding the need of computing these decisions beforehand. The author already published several considerations on this application domain [47, 50]; their common feature is once more the idea of creating a set of abstract rules $\bar{r}$ instead of using the extensive set of input variables $\bar{i}$.

Figure 6.10 shall demonstrate the functionality when applying policy-based control to the EV charging problem, where the policy evaluation is indicated for a given EV that arrives at an arbitrary location which is equipped with charging infrastructure.

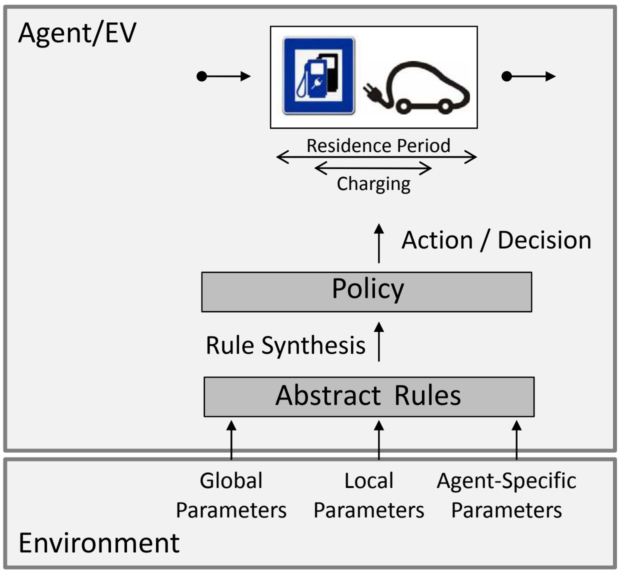

Figure 6.10: Illustration of Policy-Based Control Principle for EV Charging Control

Principally, the optimized policy which finally decides the EV's charging power at a given time step t is synthesized from abstract rules that consider parameters from its environment. Out of these parameters, abstract rules derive agent-specific information for evaluating the EVs power demand as well as the state of its environment. Here, three different parameter classes can be distinguished from each other:

- Agent-specific parameters concern the EV's driving behavior and charging demand, including its residence time at the actual charging station or its likelihood of getting parked at another charging spot later on.

- Local parameters consider other EVs immediately affecting the local situation in the power grid. For example, if the power grid is stressed locally because of a high amount of EVs charging at the same bus, their charging power may has to be reduced in the next time step in order to avoid critical power flow conditions.

- Global parameters consider information describing the whole system's state, such as the total load to the distribution grid, totally expected supply from renewables or financial aspects considering costs of electrical power supply.

Using these parameters as input for the abstract rules, each rule delivers a numeric result in the interval [0,1] that defines the agent's priority for charging. 0 would indicate that the corresponding EV should not charge at the actual time step, 1 advises it to charge with maximum power. Since a variety of criteria has to be taken into account for computing the optimal charging power of an EV, as can be estimated from the parameter classes defined above, multiple rules have to be defined that finally get combined in order to synthesize a final policy.
In this case, each abstract rule indeed gives indication on the priority and is not only an aggregated information of the environment. Thus, the term "rule" is significant herein.
Table 6.8 gives the set of defined abstract rules according to [50] that will be used within this application.

Rule	**Acronym**	**Description**
Residence Time So Far	RT	Total residence time during all previous time steps
Estimated Time to Departure	ETTD	Remaining residence time at actual charging station
Passed Residence Time	PRT	Passed residence time at actual charging station
Estimated Remaining Time	ERT	Estimated remaining residence throughout the time horizon
Actual Irradiance	AI	Actual solar irradiance relative to known maximum
Past Irradiance	PI	Past mean solar irradiance during PRT
Estimated Irradiance	EI	Estimated mean solar irradiance during ETTD

Continued on next page

Rule	Acronym	Description
Actual Wind Speed	AWS	Actual wind speed relative to known maximum
Past Wind Speed	PWS	Past mean wind speed during PRT
Estimated Wind Speed	EWS	Estimated mean wind speed during ETTD
Actual Base Load	ABL	Actual base load relative to peak load value
Past Base Load	PBL	Past mean base load during PRT
Estimated Base Load	EBL	Estimated mean base load during ETTD
Actual Price	AP	Actual price relative to peak price value
Past Price	PP	Past mean price during PRT
Estimated Price	EP	Estimated mean price during ETTD
Distance to Peak Load	DTP	Absolute temporal distance from time of peak load
Mean MVA Rating	MMVA	Mean rating (maximum allowed power flow) of connected branches relative to maximum ratings in the grid
Number of EVs at Bus	NREVB	Actual number of EVs remaining at bus
Mean Number of EVs during PRT	MNREVB	Mean number of EVs remaining at bus at each time step during PRT
State Dependent Rules		
Number of EVs Charging	NREVC	Total number of EVs charging during last time step
Number of EVs Charging, Same Bus	NREVCB	Total number of EVs charging during last time step at same bus
Mean Charging Rate	MCR	Mean charging rate (relative to maximal charging power) per EV during last time step over all EVs
Mean Charging Rate, Same Bus	MCRB	Mean charging rate (relative to maximal charging power) per EV during last time step over all EVs at same bus

Continued on next page

Rule	**Acronym**	**Description**
Already Charged Energy	ACE	Sum of already charged energy during simulated time span related to minimum needed amount of energy

Table 6.8: Formulation of Abstract Rules

While the first three rules consider agent-specific information of the considered EV, additional rules using local (at the same bus) and global (in the entire grid) information are formulated for regarding the power grid's operation point. Irradiance as well as wind speed data during the whole residence period of an agent is taken into account since using renewable power for charging batteries should be forced. Base load data is used with the intention of shifting charging to off-peak hours. Since the objective function of the optimization problem aims at minimizing financial costs, price data is used as well. For representing local distribution grid aspects, data from connected branches is included as well as information about how many EVs are remaining at the same bus during the considered time steps, using local information.
The lion's share of rules can be seen as absolute values that do not depend on the states before, i.e. on the evaluation of the policy in step $t-1$, while the last 5 rules consider state-dependent rules. Here, the evaluation of the rules depends on the evaluation (i.e. the value of the policy) at the steps before, where the actual charging rate is taken into account that resulted from policy evaluation. Many rules (including MMVA to MCRB) additionally incorporate the interrelation between agents implicitly, that influence the resulting charging rate of an agent according to the actual behavior of the others. Thus, competitive behavior is integrated.
An additional information is important considering each rule: namely whether this rule leads to a higher or lower charging rate. Taking for example ETTD: an EV that has a relatively high remaining time left at the charging station should get a lower charging rate at the actual time step. This is intrinsically clear, since there is much time left to get charged, probably more than for other agents. Therefore, this rule is getting inverted.
Appendix A.9 provides the formal definition plus implementation of the EV charging rules for more detailed understanding of their mathematical outlook.

Figure 6.11 shows the solution evaluation within the optimization process with specific detail to policy-based control.
Once more, the metaheuristic algorithm is abstracted into one box. Moreover, the outlook of the policy $p_{EV}(\overline{r})$ as well as the synthesis scheme are not

content of discussion right now, but will be considered later on.

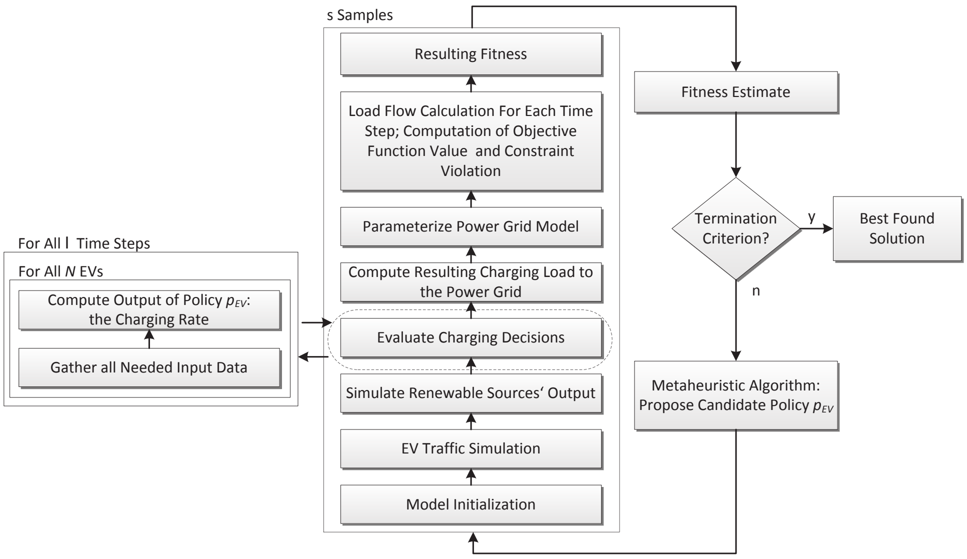

Figure 6.11: Solution Evaluation Using Policy-Based EV Charging Control

Realizing Intrinsic Memory with Abstract Rules

For detailed analysis of the provided rules, one has to do a more general consideration of function approximation in computational intelligence with respect to dynamic models:

Since the early beginnings of machine learning, the integration of some kind of memory to intelligent systems represented a challenging task. Manifold approaches have been identified, however, the developments in the field of artificial neural networks represent some very general image of what this issue is about. Here, so called *Recurrent Neural Networks (RNNs)* have evolved that implement directed cycles between units to realize some internal memory or intrinsic state that allows to not only consider inputs from one time level, but from many different ones (such as inputs or neural network states from time steps before). Hence, sequences of inputs can be processed by this class of networks rather than only stationary inputs. This development was crucial to function approximation tasks in dynamic systems like time-series modeling or issues of dynamic optimization (like for instance the mentioned techniques for Adaptive Dynamic Programming). In the meanwhile there exist various architectures of RNNs, the most popular one is probably the Elman network [28] which introduces cycles from the hidden layer back to

the input layer for generating an inner state of the network.
Coming from the recurrence principle, a special peculiarity was the implementation of feedback loops that propagate the neural network's output back to the input layer, like for instance the Jordan network [28]. Thus, the output layer at time t both depends on the inputs of time t as well as the network's output of $t-1$, hence, implementing an additional memory-ability. Especially for dynamic systems this has been an essential achievement.

In this context of implementing an intrinsic memory to artificial neural networks, this important idea which is fundamental to function approximation in dynamic systems needs consideration within the herein evolutionary policy optimization as well. While the herein discussed schemes for policy synthesis do not allow for the implementation of loops - hence would be unable to consider recurrence as well as feedback principles - these ideas have to be realized by designing appropriate abstract rules. Therefore, within the given rules from Table 6.8, recurrence as well as feedback techniques are integrated:

- Dynamics are implicitly considered by all rules that take information from past time steps into account, namely *RT, PRT, PI, PWS, PBL, PP, MNREVB, NREVC, NREVCB, MCR, MCRB* and *ACE*. Hence, information from previous time steps is integrated directly into the rule computation and does not need for consideration in the final policy synthesis via, for instance, loops.

- Recurrence/feedback is realized by a subset of the mentioned rules, namely the state-dependent rules *MNREVB, NREVC, NREVCB, MCR, MCRB* and *ACE*. They consider the charging rates (and thus the outputs of the policies' evaluations) of previous time steps, hence, implement feedback-enabled memory into policies' actions without the need of designing loops.

In this way, recurrence enables the policies to not only derive actions based on the actual system state $J(t)$ as indicated by Equation 6.20, but to integrate knowledge of the system's past states - illustrated in Equation 6.21.

$$u_t = p(J(t)) \tag{6.20}$$

$$u_t = p(J(t), J(t-1), ..., u_{t-1}, u_{t-2}, ...) \tag{6.21}$$

Summing up, with the implementation of abstract rules, important schemes of memory-integration into policy function approximation are realized, substantiating the power of the herein developed technology for dynamic optimization issues.

6.5 Experimental Evaluation

Having specified the holistic optimization problem for EV charging control, the experimental section shall now demonstrate the application to a real-world power grid scenario.

6.5.1 System Description

For this show case a power grid of lower voltage-level will be considered in order to show the application to distributed smart control issues in distribution feeders. As discussed before, especially the application of controllable small-scale devices at lower levels challenges new technologies in smart electric grid developments.
The 33-bus radial test feeder represents a well known[7] real-world application to primary distribution and is set-up respectively extended as follows: the distribution network is modeled according to [5], where 1000 domestic customers are assumed to be served. The base voltage is 12.66kV, while at each node a transformer to a secondary feeder respectively to a larger customer can be assumed. Hence, each node is a load-bus (indicated with arrows), as shown in the layout in Figure 6.12. 8 photovoltaic power plants (indicated with PV) as well as 2 wind power plants (indicated with W) are added to the grid in a distributed manner. The renewable generation capacity is assumed to be around 10% of the total supply. 300 EVs are modeled to exist within the system and get charged at different buses according to traffic patterns. The black bar shows the slack bus, which is necessary for power flow calculation [116].

In this system, the utility function shall be defined of minimizing total cost of energy supply, i.e. minimizing generator-costs of the slack bus. While the total supply considers both the (non-controllable) base load as well as the (controllable) EV charging load, choosing appropriate charging decisions aims at minimizing costs for charging.

[7] This grid model has also been used for several studies on EV charging control optimization, like in [94], [97], [98] and [101].

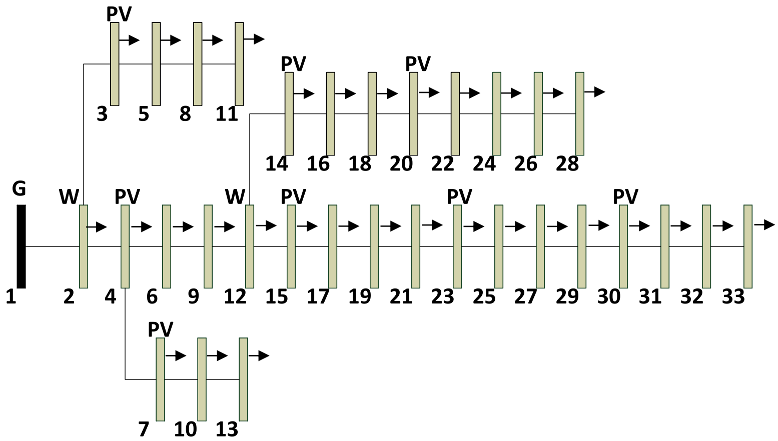

Figure 6.12: Layout of the 33-Bus Test Feeder

Constraints Weighting

Since for obtaining the fitness of a solution candidate according to Equation 6.13 (no matter if using a policy or a real-valued vector as solution representation) the constraint violation is added to the objective function value via a weighted penalty term, these weights $\overline{w_{CV}}$ have to be defined. The choice of the weights is crucial to the evolutionary search, where too low values would lead to negligence of the constraints, while too high values increase the ruggedness of the fitness landscape and complicate the evolutionary search. The violation of a constraint is measured as squared difference from the boundary value. While the objective function is given in [Euro/MWh] the following weights are chosen:

- Real-power supply capacity exceedance at slack bus (Equation 3.8): $1MW \mathrel{\hat{=}} (1\ Euro)^2$
- Reactive-power supply capacity exceedance at slack bus (Equation 3.9): $1MVAr \mathrel{\hat{=}} (1\ Euro)^2$
- Voltage deviation per bus (Equation 3.10): $1\%\ deviation \mathrel{\hat{=}} (2\ Euro)^2$
- Real-power flow overload per branch (Equation 3.11): $1MW \mathrel{\hat{=}} (1\ Euro)^2$
- Charging rate exceedance per EV (Equation 6.10): $1kW \mathrel{\hat{=}} (1\ Euro)^2$
- Minimum charged energy not reached per EV (Equation 6.11): $1kWh \mathrel{\hat{=}} 10\ Euro$

Traffic Simulation

Real-world data on driving behavior is used as essential information for the simulation model that is taken from an official survey by the Austrian ministry [17]. This survey - among other things - identifies patterns of driving behavior. Therefore, two most relevant driving patterns have been extracted from the survey, namely the patterns of full-time and half-time workers. During a weekday, these two patterns comprise 74% of all driven journeys, thus, provide a very descriptive power for modeling issues. For these patterns, the mean values and the deviations for specific arrival and departure times as well as driving distances are taken which finally are used for building the driving profiles as discussed before.
Within each pattern, three different locations are modeled for parking at home, at work and at any location in free time (shopping, education, entertainment). For each location, different probabilities for the existence of a charging infrastructure are modeled (i.e. p_{home}, p_{work} and p_{free}), describing a possible future infrastructure from an actual point of view: at home, each EV user has an own charging station ($p_{home} = 1$). At work, there is a probability of $p_{work} = 0.5$ that an appropriate infrastructure is available. For locations where potential users remain in free time, this probability is assumed to be $p_{free} = 0.25$. The resulting charging load at a specific location is then being correlated to a corresponding bus within the power grid model. Within each simulation run, synthetic driving profiles are computed from prototype-profiles of both patterns, being randomized in terms of driving time and remain time at specific locations. Thus, the probabilistic behavior of individual traffic can be simulated based on real-world data and incorporated into the metaheuristic optimization process.
Each of the 300 EVs is assumed to be equipped with a charging controller with $cr_{MAX} = 10$kW (as before). In reality, various configurations may occur, but using this single specification is sufficient for the underlying simulation model.

6.5.2 Training and Test Scenarios

Similar to experiments in Chapter 5, for learning and testing of policies, proper scenarios need to be defined to avoid overfitting of found solutions. Therefore, the environment that typically shows volatile and uncertain behavior needs to be provided with appropriate uncertainty within the simulation when solution candidates get evaluated. This environment to the power system is given by the base-load profile (i.e. the system demand that is caused by domestic and commercial customers), the electricity-tariff data (which may vary dynamically according to the stock market), as well as the wind speed and solar irradiance forecasts that estimate renewables' supply.

Training Scenarios

When evaluating a solution candidate in an uncertain system, the simulation model is randomized properly for each evaluation. For the learning procedure, a set of different environmental behavior data needs to be defined that mimics real-world uncertainty that the policies will have to cope with during operation. Similar to previous experiments in Chapter 5, these data are given in the form of profiles that vary along the simulation horizon.

For base-load data, same profiles are used as in Section 5.3, namely H0, G0, G6 and L0 profiles from the Austrian APG regulation zone that have already been defined in Figure 5.6. For dynamic electricity prices that vary along time, official stock-market prices from the European Energy Exchange are taken. Therefore, 4 different 24h-profiles of hourly ELIX contracts are expressed, shown in Figure 6.13 being scaled to [0,1].

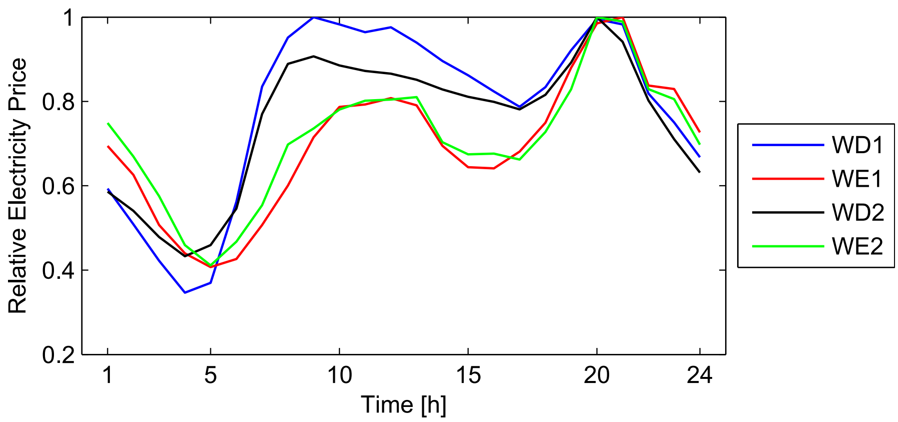

Figure 6.13: Electricity Price Profiles from the European Energy Exchange

For probabilistic training, diverse price data from weekend (WE) and weekdays (WD) is chosen. WD1 gives the mean prices from Monday, 19.9.2011 until Friday 23.9.2011. WE1 gives the mean weekend prices from Saturday, 24.9.2011 and Sunday, 25.9.2011. Similar holds for WD2 (26.9.2011-30.9.2011) and WE2 (1.10.2011 and 2.10.2011). At the beginning of each evaluation, one of these profiles is chosen equally distributed.

For simulating the probabilistic supply from renewables, solar irradiance as well as wind speed profiles are taken representing typical forecasts utilized for power grid operation. These profiles are provided by statistical investigations, online available from the National Solar Radiation Data Base[8] by the National Renewable Energy Laboratory.
This database provides various meteorological data for numerous sites in the US. For training, profiles are taken for solar irradiance (SI) in Figure 6.14 and wind speed (WS) in Figure 6.15 respectively.

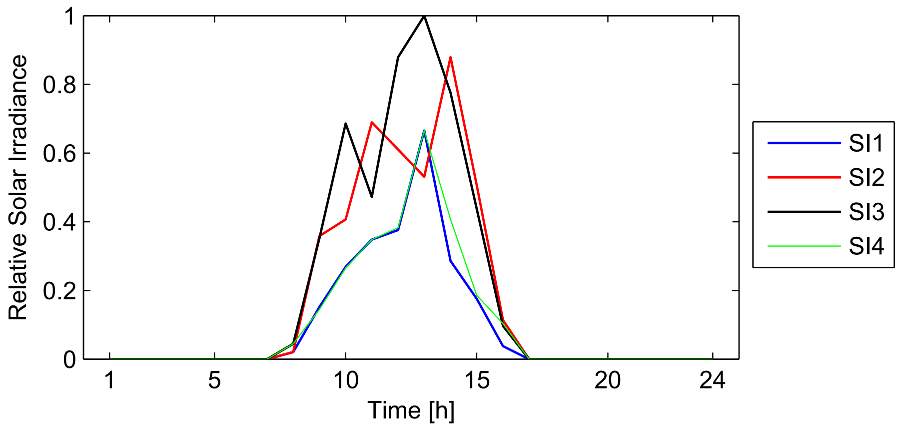

Figure 6.14: Solar Irradiance Profiles for Photovoltaic Supply Simulation

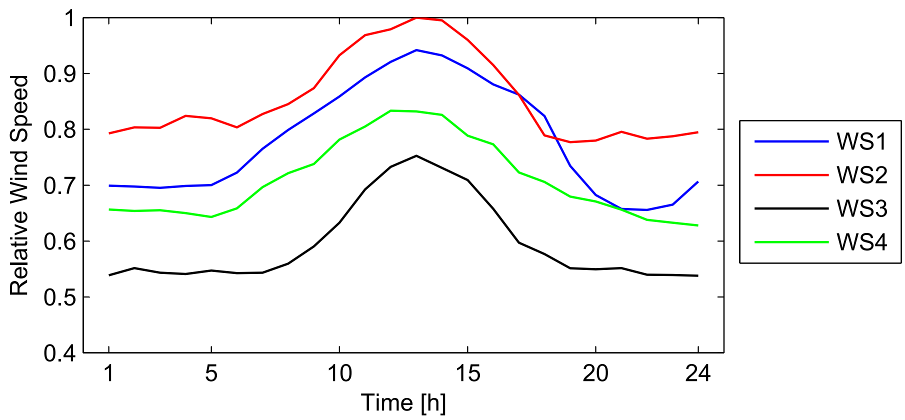

Figure 6.15: Wind Speed Profiles for Wind Power Supply Simulation

[8]National Solar Radiation Database, http://rredc.nrel.gov/solar/old_data/nsrdb/1991-2005/tmy3/by_state_and_city.html

In order to evaluate a candidate policy on uncertain scenarios during the training phase, at the beginning of each evaluation these profiles are sampled equally distributed. Redarding the general processflow when evaluating a solution candidate as discussed previously in Figure 6.11, the following Figure 6.16 presents this issue.
In this illustration the rest of the processflow is disregarded (indicated by the dashed line), while the model initialization is given in more detail, where at the very beginning of each evaluation the simulation model gets initialized with randomly chosen environment-data enabling the training of the solution to uncertain conditions.

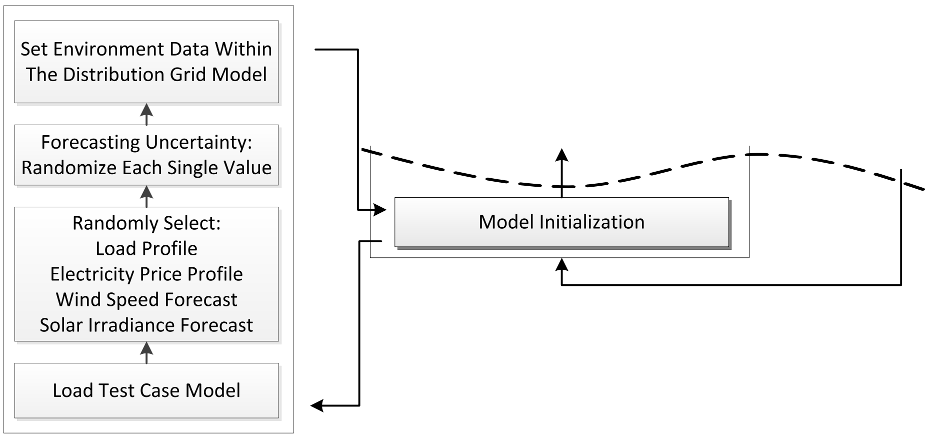

Figure 6.16: Policy Evaluation With Uncertain Environment

Test Scenarios

For evaluating evolved policies on independent test scenarios, a deterministic simulation is built using environment data (i.e. load, price, and supply profiles) that is distinct from training. For EVs, from the traffic simulation for each agent 2 fixed patterns are expressed, i.e. 2 different deterministic EV scenarios are defined. Figure 6.17 illustrates the assumed behavioral patterns. Here, over all $N = 300$ EVs, the mean charging availability (i.e. the EV is being parked and charging infrastructure is available) is shown for the two scenarios. This availability approximates the value 1 at night for both cases, where most EVs are at home (where charging infrastructure is assumed to be always available). Before as well as immediately after noon this availability peaks as well, caused by users that have charging spots available at work or half-time workers that arrive at home after midday.

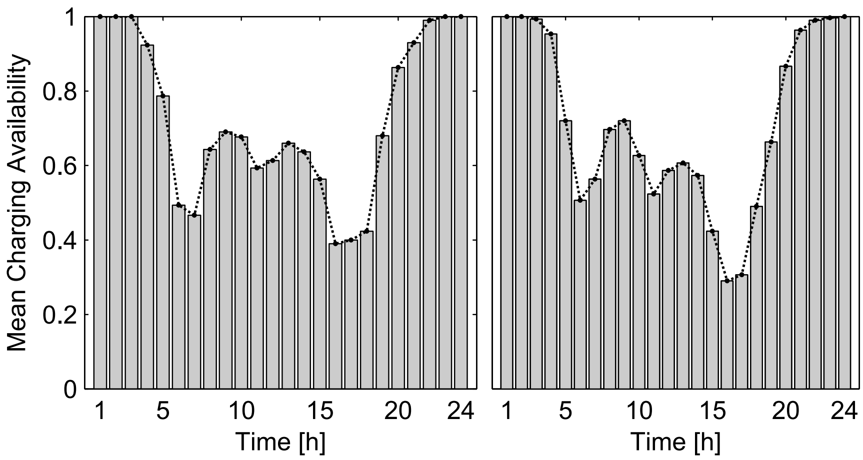

Figure 6.17: Relative Probability Over all EVs Being Parked and Charging Infrastructure is Available.

Especially during the day, the average charging availability between the two scenarios shows clear differences, which illustrates the volatility that needs to be treated successfully for making valid charging decisions. Additionally, this figure only illustrates the average charging availability over all N EVs, hence, does not illustrate the individual differences within the fleet. For showing this individuality of EV agents, Figure 6.18 gives the standard deviation over all N EVs within each time step for both scenarios.

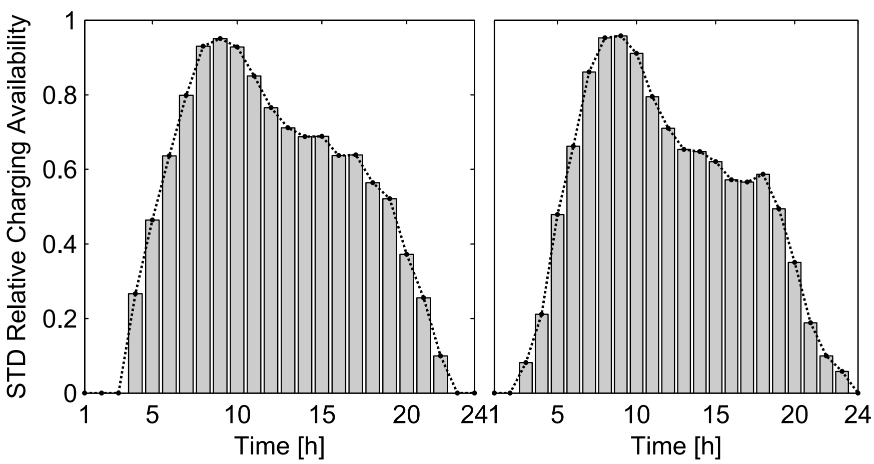

Figure 6.18: Standard Deviation of Charging Availability Among all 300 EVs.

While the standard deviation values look quite similar for both scenarios, they clearly show the individuality of single EVs within the fleets. Especially before noon, the standard deviation over all EVs' charging availabilities peaks nearly to one.
Having defined two distinct EV-behavior scenarios for testing, the environment data needs to be defined as well.

From the same data sources as used for generating the training data (i.e. the national solar irradiation data base for renewables' forecasts, price data from the EEX and load data from the APCS), but using measurement data from distinct time intervals, the independent profiles for test-scenarios are generated. While both load data as well as electricity tariff (price) data are one source of volatility during real-world operation, especially the uncertain supply from renewables is of high interest when deriving charging decisions. As illustrated in Figure 6.19, for both solar irradiance as well as wind speed, two different profiles are defined, also giving one profile for both electricity price and base load.

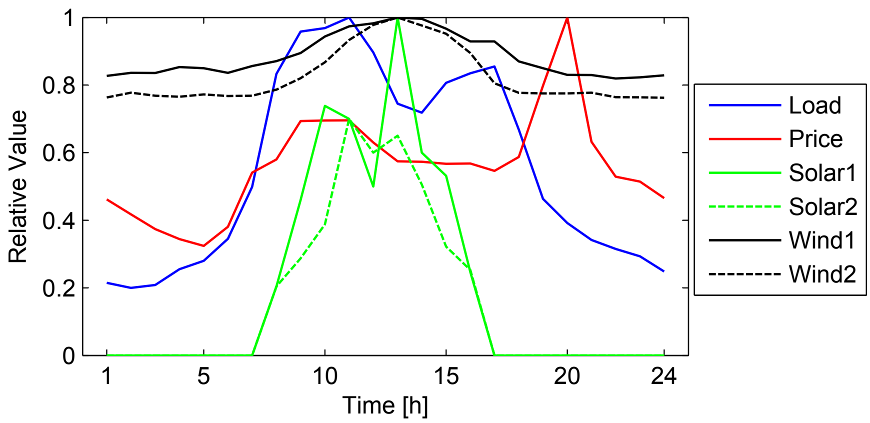

Figure 6.19: Environment Data for Test Scenario

These profiles (that are independent from the training simulation) shall be used to extract the test set: while for EV behavior 2 scenarios are defined, both for irradiance as well as wind speed two profiles are given as well. Therefore, $2^3 = 8$ test cases can be fixed (price and load profiles keep constant) which will later be used for testing charging control strategies. On these scenarios, the optimized policies' performances are evaluated with respect to reference-strategies for charging control.

6.5.3 Simplistic Charging Policies as Reference

In the introductory section on controlled EV charging (see Section 6.2), the different objectives have been stated such as control for peak-shaving (grid-oriented), charging with renewable energy (supply-oriented) or charging at lowest possible costs (price-oriented), that were summarized in Figure 6.1. While the control with optimized policies according to above defined holistic EV charging optimization problem unifies all these objectives, simplistic charging policies p_s shall be defined for comparison reasons. These simplistic policies are designed to make individual charging decisions to each EV while considering one of the defined objectives. These simplistic policies as well provide policy-based distributed control, but rely on simple heuristics for deriving charging decisions. Additionally, a simplistic policy for random charging is defined that should mimic the behavior of randomly distributing charging decisions along time. These policies build a suitable reference-technology being used for comparisons later and shall be defined as follows (some policies are given in pseudocode-notation for better understanding):

Uncontrolled Charging: $p_{S,UC}$

This heuristic disregards any kind of grid-wide interests and makes the EV charging with cr_{MAX} immediately after being plugged to a charging infrastructure until it reaches a full SOC. Such charging behavior is the reference scenario if no charging control would be installed.

Randomly Distributed Charging: $p_{S,R}$

Randomness is natural source for distributing charging decisions equally. Therefore, in each time step if an EV is available for charging (i.e. it is being plugged to charging infrastructure and its $SOC < 1$), for this EV a dice is thrown that decides whether to charge now or not. If it is allowed to, it charges with full power cr_{MAX} within this time step, see Algorithm 3. This heuristic is an obvious approach for distributing charging along time. A 3-sided dice is assumed for the herein scenario which showed to be most performant on the test evaluation.

Demand-Oriented Charging: $p_{S,D}$

This heuristic scales the allowed charging power of each available EV inversely proportional to the actual base load in time step t (see Algorithm 4), hence, leads to the objective of peak-shaving.

Algorithm 3 Calculate charging rates P_{EV} as output of $p_{S,R}$

```
cr ← cr_MAX;
for all t = 1...T do
  for all n = 1...N do
    r ← random(1, 3);
    if r == 1 && (available(i, n)) then
      P_EV t,n ← cr;
    else
      P_EV t,n ← 0;
    end if
  end for
end for
```

Algorithm 4 Calculate charging rates P_{EV} as output of $p_{S,D}$

```
cr ← cr_MAX;
minBL ← min(loadProfile); maxBL ← max(loadProfile);
for all t = 1...T do
  scale ← (loadProfile[t] − minBL)/(maxBL − minBL);
  for all n = 1...N do
    if available(t, n) then
      P_EV t,n ← cr ∗ (1 − scale);
    else
      P_EV t,n ← 0;
    end if
  end for
end for
```

Price-Oriented Charging 2: $p_{S,P}$

Similar as $p_{S,D}$, but the charging power of each EV is scaled according to the actual electricity tariff rather than using the actual base load. This heuristic mimics locally cost-optimized charging as different works in literature aim at.

Supply-Oriented Charging: $p_{S,S}$

$p_{S,S}$ scales the charging power of each available EV proportional to the actual supply from renewables (both wind and photovoltaic) in order to force charging with renewable energy.

Algorithm 5 Calculate charging rates P_{EV} as output of $p_{S,S}$

```
cr ← cr_MAX;
minPV ← min(irradiance); maxPV ← max(irradiance);
minWind ← min(windSpeed); maxWind ← max(windSpeed);
for all t = 1...T do
  scalePV ← (irradiance[t] − minPV)/(maxPV − minPV);
  scaleWind ← (windSpeed[t] − minWind)/(maxWind − minWind);
  scale ← mean(scalePV, scaleWind);
  for all n = 1...N do
    if available(t, n) then
      P_EV t,n ← cr ∗ scale;
    else
      P_EV t,n ← 0;
    end if
  end for
end for
```

These simple policies (heuristics) are designed to fulfill the charging requirement of each EV (i.e. satisfy the E_{MIN}-constraint), but indeed are not able to directly consider global behavior respectively make holistic decisions.

6.5.4 Experimental Results

Validating the Performance of Simplistic Strategies

The simplistic control strategies offer quite obvious approaches for charging control, which realize some kind of distributed control that could be implemented with low effort to real-world applications. In order to build a baseline, these strategies shall now be applied to the defined test scenarios in order to provide an estimate on the difficulty of deriving valid charging decisions. In the same way as evaluating a complex charging policy, the simplistic strategies can be evaluated by computing their respective objective function values as well as constraint violations (respectively the fitness function value when summing both according to Equation 6.12) when being applied to test scenarios. Table 6.9 shows the resulting fitness function values (according to the fitness function definition in Equation 6.12) of all simplistic strategies applied to the test set.

Test Case #	$p_{S,UC}$	$p_{S,D}$	$p_{S,S}$	$p_{S,P}$	$p_{S,r}$
1	3448.90	2856.60	6151.60	8232.70	3392.10
2	3497.00	2895.20	8957.90	8372.10	3456.40
3	3519.00	2916.10	5705.30	8494.90	3371.30
4	3470.90	2877.50	7373.20	8355.10	3404.40
5	3875.30	2737.20	5195.80	7498.00	3593.30
6	3933.70	2768.90	9036.70	7626.40	3908.80
7	3956.40	2790.20	4298.80	7730.50	3696.40
8	3898.00	2758.50	6453.00	7601.40	3808.70

Table 6.9: Fitness Function Values of Simplistic Strategies' Test Evaluation

Both the demand-oriented $p_{S,D}$ and the randomized $p_{S,R}$ charging strategies lead to lowest values in means of fitness function. $p_{S,D}$ leads to charging at off-peak periods where the electricity tariff is low and the distribution equipment is far from being overloaded. The other strategies lead to significantly higher fitness values that cannot be caused solely by the objective function (notice that these values are in [$]), but occur because of constraint violations. Therefore, Table 6.10 gives the values of constraint violations (also mapped to [$] as defined before).

An important point is that none of the provided simplistic strategies leads to valid charging behavior, since they all cause significant constraint violations throughout all test scenarios. Hence, more sophisticated policies may be needed that are able to combine different information from the EVs' environment in order to derive charging decisions that both avoid constraint violations and lead to low-cost charging.

Test Case #	$p_{S,UC}$	$p_{S,D}$	$p_{S,S}$	$p_{S,P}$	$p_{S,R}$
1	715.21	483.58	3068.50	5294.50	23.69
2	755.39	514.34	5474.20	5425.30	182.70
3	757.09	515.69	2518.10	5524.90	1153.40
4	716.90	484.92	4192.20	5393.90	155.02
5	1070.60	383.18	2133.00	4577.30	281.93
6	1121.00	407.10	5538.80	4697.10	125.23
7	1123.40	408.72	1167.50	4778.10	461.25
8	1073.00	384.80	3267.50	4657.60	21.87

Table 6.10: Constraint Violations of Simplistic Strategies' Test Evaluation

Comparing Lookahead Optimization

Since in the test scenarios all behavior of demand, renewable supply and EV users is given deterministically, lookahead optimization can be applied for computing the set of optimal charging control variables $\overline{u}_{EV} = [P_{EV1,1}...P_{EVT,N}]$ directly for building a reference solution. In this case, the $T \times N = 7200$ control variables are computed directly using simulation-based real-valued optimization.
Island genetic algorithms (IGA) implemented in HeuristicLab have shown to be capable of tackling this high-dimensional problem, the finally applied configuration is given in Table 6.11. For both mutation as well as crossover, operators originally stemming from evolution strategies [14] are applied in order to treat the real-valued solution vector successfully.

Having defined the algorithmic settings, deterministic lookahead solutions are optimized for all test scenarios. The solution consists of $N * T = 7200$ real values that give the desired charging power for each EV at each single time step for satisfying both the users' charging demand as well as the operational power grid constraints. Figure 6.20 shows the 5 best found solutions' fitness values within each test scenario with boxplots.

While for all scenarios this deterministic optimization seems to find quite robust results with a fitness function value around 2500 - 2600 \$, compared to the aforementioned simplistic strategies, the demand-oriented charging $p_{S,D}$ has lead to similar results (around 2700 - 2800 \$). However, these optimized solutions not only lead to better fitness function values, but especially satisfy all constraints. Thus, finding valid charging decisions is principally possible for the defined scenarios.

Parameter	**Value**
Maximum Generations	1000
Population Size Per Island	80
Number of Islands	5
Mutation Probability [%]	7
Crossover Probability [%]	100
Selector	Tournament Selector Group Size: 2
Mutator	Self Adaptive Normal All Positions Manipulator
Crossover	Discrete Crossover
Elites	1
Emigrants Selection	Best Selection
Migration Rate [%]	15

Table 6.11: IGA Setting for Lookahead Optimization

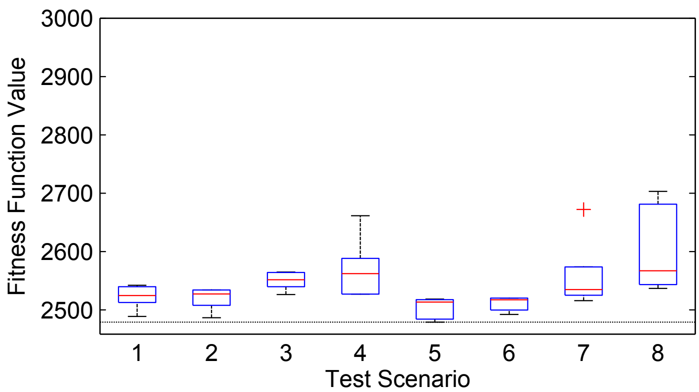

Figure 6.20: Best Found Deterministic Solutions for Test Scenarios

In order to visualize the resulting optimized charging behavior in Figure 6.21, exemplary for the test scenario # 1 the mean charging power over all $N = 300$ EVs is plotted along time for the best found solution $\overline{u_{EV}}$.
According to this illustration, optimal EV control tends to charge batteries in the late evening and at night, which is obvious regarding manifold reasons. On the one hand, the electricity price decreases in the late evening. Additionally, base load is low at night, making charging efficient from a grid's power flow point of view. Moreover, at night most EVs remain at home, where charging infrastructure is assumed to be always available.
However, even if around noon the base load as well as the electricity prices

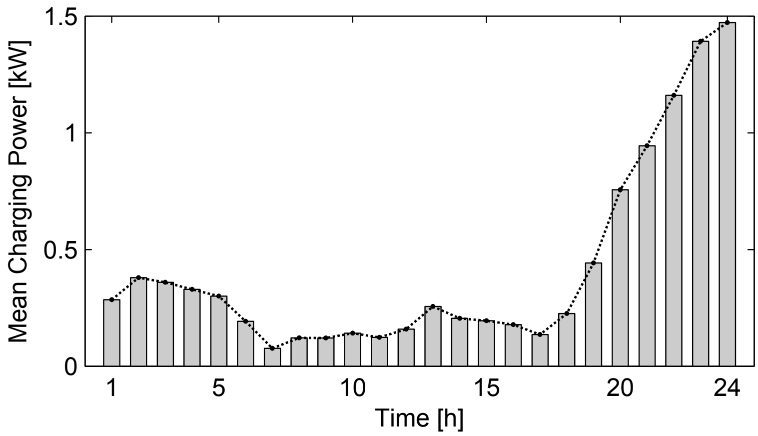

Figure 6.21: Mean Charging Power for Lookahead Optimization

are high, there occurs some charging as well. This is mainly because of the usage of renewable supply which injects with a high probability at this time, which is assumed to have zero generation costs. Thus, EVs that are available for charging at this time are scheduled for using the injected power from renewables. Nevertheless, especially around midday this is a difficult issue since base load is high and power grid constraints have to be considered thoroughly in order to not overload distribution equipment.

It has to be mentioned once more, that these solutions are optimized to exactly one specific deterministic system's behavior (i.e. one out of 8 test scenarios) and thus are unable to consider the volatile and uncertain system behavior of the real world. Nevertheless, these solutions build a suitable reference for comparing policy-based optimization on the test set.

Best Found Policies for EV Charging Control

The evolution of policies $p_{EV}(\overline{r})$ with genetic programming allows the arbitrary synthesis of complex decisions out of the set of rules $\overline{r}$. In order to evolve policies based on the training simulation, the following algorithm configuration (given in Table 6.12) is used. After training, they get evaluated on the test set in order to validate their performance.

Parameter	**Value**
Maximum Generations	200
Population Size	800
Mutation Probability [%]	20
Crossover Probability [%]	100
Selector	Tournament Selector Group Size: 5
Mutator	Multi Symbolic Expression Tree Manipulator: Replace Branch Manipulation, Change Node Type Manipulation, Full Tree Shaker, One Point Shaker
Crossover	Subtree Swapping Crossover
Elitism	No Elitism
Maximum Tree Depth	10
Maximum Tree Length [nodes]	60
Tree Grammar	Arithmetic Operators Real-Valued Constants

Table 6.12: GP Parameters for Holistic EV Charging Control Problem

From the set of evolved policies, the best found 10 solutions shall be taken for further analysis. In order to provide a fair selection, they are evaluated on the training simulation (which is probabilistic indeed) with a sample size of 100 and averaging their fitness values over all 100 samples. Figure 6.22 shows boxplots for the best policies' evaluation on the training simulation (the very left box) as well as on the test scenarios (boxes 1-8). The boxes give the distribution of the policies' fitness values for each scenario, while for the test evaluations black dots indicate the best found deterministic lookahead solution.

Throughout all test cases, the policies' performances are near the deterministically optimized best reference solutions (a more detailed view on this follows). Additionally, the best policies seem to be robust over all cases, even if they perform within different environments (i.e. different EV behavior and different environmental data profiles), which understates their application to

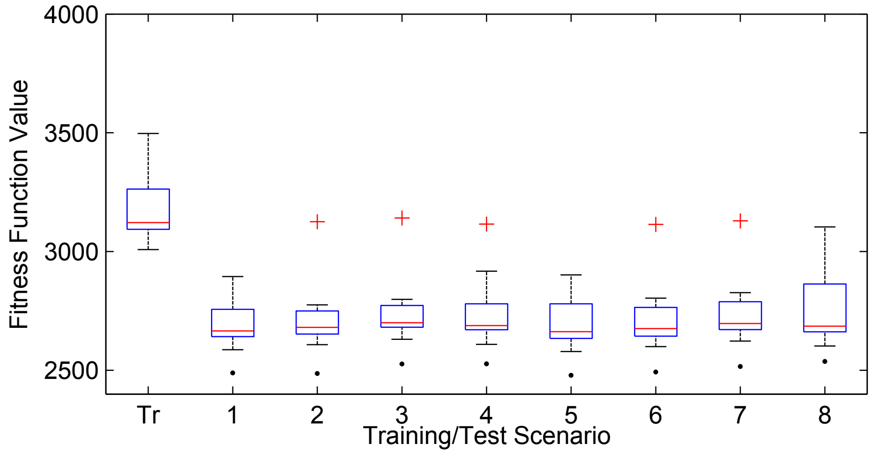

Figure 6.22: Evaluation of Policies: Training and Test

uncertain and volatile systems. Notice, that each of 10 policies is evaluated within all test scenarios, whereas the reference solutions are deterministically optimized to each single scenario.
The first box represents significantly higher fitness values, which is caused by the fact that the price profile taken for the test scenarios is "cheaper" than the average price profile that results in the training simulation.
In order to analyze the comparison between the policy-based control and deterministic lookahead reference solutions in more detail, Figure 6.23 gives the boxplots for each type of optimization approach over all test scenarios. While the filled boxes indicate the fitness function values of the best found deterministic lookahead solutions for each test scenario, the boxes indicating the policies have the same style as before.

An important outcome is, that for some cases the policies' boxes overlap with the reference solutions' boxes. This fact indicates that the policy-based approximate optimal control for some situations even outperforms deterministically optimized lookahead solutions that are computed individually for the respective scenario. In average, there is a clear tendency that the policies are comparable to reference solutions, even if they are general and are able to cope with different situations (i.e. states of the simulated system). Table 6.13 shows the median fitness function values of the best policies and the best selected reference solutions for comparison reasons.
Hence, one can easily deduce that the charging decisions derived from the approximated policies lead to competitive control, where a relative error in the range of $4.5 - 6\%$ results over the test scenarios.

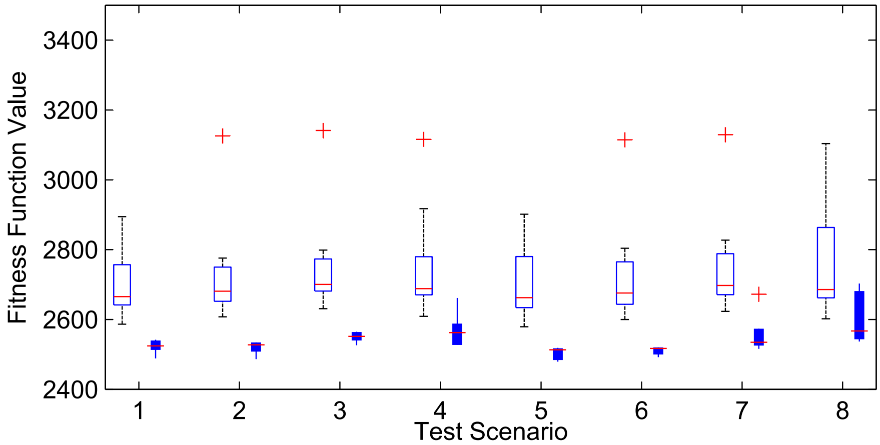

Figure 6.23: Evaluation of Policies: Comparison to Reference Solution

Scenario	1	2	3	4	5	6	7	8
Policy Control	2665.5	2681.1	2700.4	2688.2	2662.7	2675.7	2697.2	2685.8
Reference Solutions	2524.6	2527.5	2551.8	2562.3	2513.4	2517.3	2534.8	2567.1
Relative Error	0.0558	0.0608	0.0582	0.0491	0.0594	0.0629	0.0641	0.0462

Table 6.13: Policies: Comparison of Median Fitness Function Values

For better illustration, one of these policies is chosen that gets analyzed in more detail. This charging policy is depicted in Equation 6.22, its real-valued constants are given in Table 6.14.

$$
\begin{aligned}
p_{EV} =& \\
& rc_1 * -(rc_2 - rc_3 * PI - rc_4 * NREVCB - rc_5 * ACE * MCR * AWS \\
& + \frac{rc_6 * ERT^3 * PP * ACE^2 * (-rc_7 * ACE + rc_8 + rc_9 * ACE * MCR)}{-rc_{10} - rc_{11} * PBL}) \\
& * (rc_{12} * NREVCB + rc_{13} * ERT * ACE^4 * MCR) + rc_{14} \qquad (6.22)
\end{aligned}
$$

Constant	Value	Constant	Value
rc_1	0.0332934964364087	rc_8	13.98276461
rc_2	0.2505188541	rc_9	0.3959326284
rc_3	1.50256421037195	rc_{10}	19.9638199970861
rc_4	1.23114366202972	rc_{11}	0.914340006671468
rc_5	0.3959326284	rc_{12}	1.23114366202972
rc_6	36.79503836	rc_{13}	6.776484929
rc_7	31.68359083	rc_{14}	0.76845871679191

Table 6.14: Real-Valued Constants Assignment for EV Charging Policy

It has been symbolically simplified for obtaining this representation (which is obvious, since the defined grammar of arithmetic functions is not able to evolve expressions with x^3 directly, but in an indirect manner via many multiplied branches). Here, the intrinsic feature selection of GP comes into play, where only 8 abstract rules are used (out of 25 defined in Table 6.8). This is an important ability, regarding that the defined rules are partly strongly correlated.
However, this policy considers all different aspects that need to be taken into consideration for deriving flexible control actions suited to volatile situations: namely all environment data (base load, price, wind speed and solar irradiance via PBL, PP, AWS, and PI), the behavior of the EV agent itself (via ERT - estimated remaining time and ACE - already charged energy) as well as the behavior of other agents (via MCR - mean charging rate over all EVs during last time step, and $NREVCB$ that gives the number of EVs that have been charging at the same bus). These rules seem to be sufficient in order to make valid and flexible charging control actions according to the holistic problem definition. Among the test scenarios, this policy yields a mean relative error of 0.0474 [9].

This policy illustratively demonstrates the ability of GP to identify nonlinear relationships between abstract rules (free from any predefined metamodel), which is a fundamental enabler of policy-function approximation in herein applications. Having tested a set of found policies to various test scenarios, their power in means of valid and near-optimal charging control has been demonstrated when being compared to deterministically optimized reference solutions. An important fact is that policies are able to react to volatile and uncertain conditions when deriving control actions at runtime,

[9] In means of difference from the median value of the reference solutions, as depicted before.

since they have been trained within a probabilistic simulation. Hence, policies feature extrapolation abilities in order to make valid decisions in system states that may have not been covered during the training phase.
The special application of abstract rules additionally enables policies to deduce unit-specific control actions. Thus, even if a charging policy is the same for all EV agents, it leads to individual decisions that are tailored to the respective agent's environment.

Finally, a simulated smart grid scenario has been provided where holistic charging control is necessary for secure power grid operation (consider simplistic policies respectively uncontrolled charging which lead to power flow constraint violations). Here, the evolved policies are able to perform approximate-optimal control of distributed charging decisions, fulfilling the holistic EV charging control problem formulation. While being learned within an uncertain training simulation, policies result in equivalent performances on deterministic test scenarios (with 4.5-6% error) when being compared to deterministically optimized charging decisions. Even if an optimized policy is the same for all 300 EV agents, it leads to individual behavior through its abstract rules and makes the agents consider system-wide objectives of cost minimization and constraints satisfaction.

6.5.5 Concluding Remarks

Within this chapter, it has been shown that many future smart grid technologies yield an extension of the OPF problem formulation. In this context, the charging control of EV fleets has been formulated as representative issue, that unifies manifold requirements for future load control technologies in smart electric grids. Among them the most important requirements concern the abilities of deriving quick and robust control actions under volatile and uncertain conditions, which consider both the customers' needs and behavior while incorporating power grid operation aspects. Additionally, such a technology needs to be scalable to be applied to numerous distributed devices which aim at common system-wide goals.
While existing approaches for EV charging control mostly concentrate on isolated aspects of this technology, various lacks of related works have been identified. Based on this fundament, a holistic EV charging control problem was formulated that comprises both the customers' as well as the power grid's aims of charging control, while incorporating fluctuating supply from renewables. This formulation has been applied to a practical scenario where the IEEE 33-bus distribution feeder was extended to form a suitable test-bed. Here, simulation-based evolutionary policy approximation with abstract rules

has been demonstrated to evolve powerful policies for multi-agent charging control. While comparing the best found policies to both simpler charging strategies as well as to deterministically optimized charging decisions tailored to specific situations, the best found policies showed to be competitive on a real-world test scenario while have been learned within a distinct training simulation.
For approximating such a policy, a comprehensive set of rules has been developed that unifies all important data that is needed for deriving appropriate charging decisions. While many of the provided rules are correlated, policy approximation with GP showed to evolve powerful policies that rely on only a subset of all possible rules using its intrinsic feature selection process during the genetic search.

Part IV

Conclusion

Chapter 7

Summary & Outlook

Optimization is a central tool in power grid engineering, being an essential enabler of both actual and future technologies in planning as well as operation of power grids. This work presented a thorough overview on existing technologies, while having developed promising approaches that meet requirements for increasingly stochastic and dynamic optimization problems in this domain.

7.1 Main Achievements

As stated in the synopsis of this thesis, the scientific challenges it is related to regard on the one hand the definition of innovative problem formulations, but on the other hand the development of appropriate methods for treating them. These aspects shall now be considered in more detail.

7.1.1 Problem Formulation

The general OPF formulation represents an essential framework for optimization in electric power grids since many decades. In recent developments, extensions to this general problem formulation become necessary, which have been provided within this work.
By this means, the traditional OPF problem (in case of generation-unit control) has been extended to a multiperiod formulation in Chapter 5 in order to provide a formulation for optimal power flow control over time within a dynamic an uncertain environment. While having developed a suitable problem definition, respective methods for successfully treating it have been given as well.

At the same time where multiperiod considerations get important for power grid optimization, smart grid technologies emerge that raise new control possibilities as well as optimization issues. Considering different promising future technologies for smart electric grids, their requirements regarding power flow optimization have been analyzed for deriving an extended OPF formulation which is capable of satisfying them. A load control scenario based on controllable electric vehicle charging processes has been defined in Chapter 6, where based on open issues from the literature an OPF-based holistic optimization problem was formulated which is representative for multiple future smart grid technologies.

7.1.2 Methodical Developments

While the problem formulation represents one innovative aspect of this work, the development of appropriate methods for treating these problems is an essential scientific achievement.
Uncertainty is a main issue along all problems presented herein. Therefore, metaheuristic simulation optimization was presented which builds a fruitful technological ground since it enables the integration of a complex system's uncertainties into the optimization process for finding solutions of both high quality as well as high robustness. While the general functionality of simulation optimization has already been presented in the literature for different applications, its adaptation and extension to power flow related optimization aspects has been introduced in Chapter 4, where a clear distinction between static and dynamic optimization got important.
Especially for dynamic optimization under uncertainty, the development of respective methods is a hot spot in the research literature at this time, where policy function approximation represents a promising general approach. Several requirements have been specified which would need to be satisfied in order to provide dynamic stochastic optimal power flow control. In consequence, simulation-based evolutionary policy function approximation has been developed herein as being capable of meeting all these requirements. Especially when applying genetic programming for function approximation, the important feature is provided that (even nonlinear) policies for optimal control can be approximated without a-priori knowledge of their structure, which is an important advantage when handling complex real-world applications. Additionally, the idea of providing expert-generated abstract rules that are processed out of a model's state variables came into play, which enables the compression of a system's state into few information entities and thus makes policy function approximation suitable also for large-scale systems.

Since in real-world problems most often the control of multiple interrelated but distinct control units needs to be handled, additionally a method for coevolutionary genetic programming was introduced in Section 4.2.9. It enables the evolution of policies of distinct grammar for multiple devices that aim at a common objective.
Finally, the developed methods have been applied to both existing benchmarks as well as innovative problem formulations in order to validate their capabilities.

7.2 Runtime Assessment

The time needed for deriving (near-) optimal decisions is of important interest when applying optimization algorithms to real-world problems. Especially for control applications, this is an important issue since decisions most often need to be available within a defined time horizon. While having discussed diverse simulation-based optimization concepts within this work, a broad view on their runtime assessment can now be provided.
When having introduced the basic principles of simulation optimization in Section 3.3, the critical issue has already been discussed that metaheuristic optimization requires the computational intensive evaluation of a huge number of points in solution space. Especially the application of simulation for solution evaluation additionally strengthens this aspect. Further, the evaluation under uncertainty may necessitate multiple simulation runs for each solution in order to increase the objective function value's estimate's quality. While the time consumption for performing evolutionary operations (i.e. selection, mutation and crossover) and so the computational cost for solution manipulation are marginal compared to solution evaluation, this creates the bottleneck for the concepts herein and shall now be treated in more detail.
The following considerations are based on executing the simulations in Matlab 7.12 (R2011a, 64 bit) under Windows 7, on an Intel Core i7-2620M CPU at 2.7 GHz, 8 GB RAM.

7.2.1 Static Stochastic Infrastructure Planning

In Section 5.2, static simulation optimization has been proposed for handling probabilistic planning issues. The special ability of integrating uncertainty of eventual future system states into the optimization process is an attractive feature for deriving solutions that robustify the technical/economic validity of decisions. As already mentioned when describing this application, the time for evaluating a solution candidate takes around 1 second within the

respective experiments. With a population containing 120 individuals and a maximum number of 1000 generations, approximately $12 * 10^4$ seconds respectively 1.39 days are needed for performing an optimization run.
At first glance, this runtime appears to be quite long. But for static planning problems a decision needs to be computed only once during the planning process (notice that plant placements obviously do not change during operation), hence, the runtime requirement is negligible for such applications. However, for control problems this is a more important issue.

7.2.2 Dynamic Stochastic Optimal Power Flow Control

For dynamic optimization problems, policy evolution has been proposed. Here, the great advantage is that no optimization is necessary during operation, while the runtime consumption for interpreting a policy and deriving a (near-) optimal control decision from it is marginal. This idea makes simulation optimization an attractive technology for control problems, where the computationally intensive task of finding policies (with computational cost T_{EVO}) is separated from their application at runtime. This aspect shall be depicted in Figure 7.1.

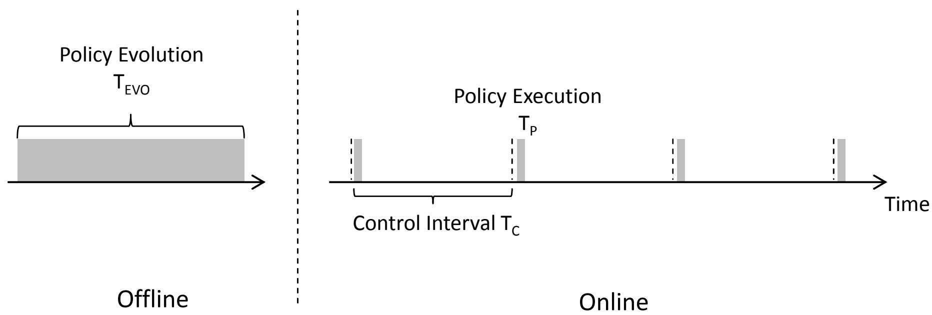

Figure 7.1: Runtime Assessment of Policy-Based Control

In many cases (especially in those treated herein), the time scale for making control decisions (in intervals of length T_C) is high compared to the time consumption when interpreting a policy and deriving a decision from it (T_P). Having proposed two different applications for policy-based dynamic optimization herein, respective runtime considerations shall be conducted as follows.

OPF-Based Generation Unit Scheduling

The conventional OPF formulation is a well established optimization problem that characterizes power grid planning and operation since many decades. By this means, appropriate solvers already exist and build an integrated part of state-of-the-art analysis tools that represent an important instrument for electric power utilities. While such a solver has been used within this work (namely interior point solver implemented in MATPOWER) for creating OPF reference solutions, it shall be applied for runtime assessment too.
Executing this reference solver for computing an OPF solution takes (depending on the problem's size) from 0.1 to approximately 2 seconds for herein treated benchmarks. In real-world power grid operation, such computations are performed at comparably high time scales (such as once per hour (i.e. $T_C = 1$h), depending on the application), thus this runtime is marginal.
For evolving OPF control policies in herein experiments according to algorithmic settings given in Tables 5.4, 5.5 and 5.6, the resulting computational costs are summarized in Table 7.1. Similar to the previous application, simulation-based optimization is very computationally expensive and takes up to $T_{EVO} = 48.44$ hours for the largest benchmark herein.

Computational Cost	**14-Bus**	**30-Bus**	**57-Bus**	**118-Bus**	**300-Bus**
# Evaluations for Polynomial Synthesis	$2 * 10^4$	$2 * 10^4$	$8 * 10^4$	$8 * 10^4$	$8 * 10^4$
# Evaluations for GP-Based Synthesis	$8 * 10^4$	$8 * 10^4$	$8 * 10^4$	$8 * 10^4$	$8 * 10^4$
Time / Evaluation [s]	0.13	0.17	0.29	1.04	2.18
Total Time for Policy Evolution: Polynomial Synthesis [h]	0.72	0.94	6.44	23.11	48.44
Total Time for Policy Evolution: GP-Based Synthesis [h]	2.88	3.78	6.44	23.11	48.44

Table 7.1: Computational Cost for OPF Policy Evolution

However, the computation of a control policy is separated from online-operation. At runtime, only the interpretation of the found policies is of importance. Since within respective experiments the herein found policies are analytical functions of quite simple mathematical structure (even only use arithmetic operations), the interpretation of a policy only takes less than a thousandth of a second (T_P). Hence, similar to the time consumption of the reference OPF solver, this time is negligible in real-world operation of power grids where OPF decisions are made at much higher time scales (T_C). Hence, with simulation-based policy evolution it is possible to make use of the advantages of simulation optimization, but disable its issues on computational cost by separating the (offline) optimization from the (online) operation stage.

OPF-Based EV Charging Control

Regarding the essential optimization concept, this application is similar to OPF-based generation unit scheduling. When considering its runtime assessment, as well two different stages need to be discussed, namely the evolution of the policies on the one hand, and their application to online operation on the other hand. Contrary to generation-unit scheduling, no state-of-the-art solvers exist for suchlike problems that can be used for optimization at runtime. Therefore, a simulation-based optimization approach has been proposed for creating static (parametric) reference solutions for specific system states, namely lookahead optimization.
Table 7.2 presents the resulting computational cost for solution computation.

Computational Cost	**Computing Static Reference Solutions**	**Policy Evolution**
Algorithm Type	IGA	GP
# Evaluations	$4 * 10^5$	$1.6 * 10^5$
Time / Evaluation [s]	0.11	0.58
Total Time for Solution Evolution [h]	12.22	25.78

Table 7.2: Computational Cost for EV Control Policy Evolution

When computing static reference solutions with simulation-based lookahead optimization, the vector $\overline{u}_{EV}$ is computed directly that gives each EV's charging power. By this means the solution evaluation is much faster than with GP-based policy evolution, where in each evaluation run the policy's output has to be derived for each agent separately over all time steps. While the interpretation of a policy itself is quite fast (less than a thousandth of a second), deriving $N * T = 7200$ charging decisions within a complete simulation run leads to a significant increase of computation time for solution evaluation.

Nevertheless, the optimization of policies is decoupled from their application in real-time operation. Since a charging decision can be derived from a policy at runtime in less than a thousandth of a second (i.e. $T_P << T_C$), this approach is suitable to real world control applications.

7.2.3 Conclusions on Runtime Assessment

While evolutionary simulation optimization is faced with critical runtime issues due to computationally expensive solution evaluations, the simulation runtime builds the critical bottleneck. Since time consumption for evolutionary operations (selection, mutation, crossover) becomes marginal, analysis have been performed mainly based on computational costs for simulation.
Especially the concept of policy-based near-optimal control brings the advantage that the simulation optimization process itself can be shifted to some offline stage, where during operation only the interpretation of the policies represented by analytical functions is necessary. While in real-world applications most often the costs for policy interpretation (T_P) are small compared to necessary control intervals (T_C), this concept builds a promising approach.
For herein experiments, the simulation models and solvers have been implemented in Matlab without special emphasis on runtime minimization. Sure, using parallelization techniques and more efficient algorithm implementations would bring the potential of speeding up simulation, reducing the derived values for T_{EVO}. Nevertheless, the principal order of T_{EVO}, T_C and T_P would remain the same.

7.3 Outlook

From this essential work herein, potential investigations can be carried on into multiple directions, both concerning the application as well as the applied methods themselves. Before discussing potential future work, a borderline should be drawn that characterizes the spectrum of this thesis.

7.3.1 Spectrum of the Thesis

System Dynamics
Dynamic optimization with policies has the great advantage that it avoids the necessity of computing a specific solution to each state the dynamic system exhibits over time. Hence, dynamic adaptation of solutions seems to be not necessary, which is a major challenge in dynamic evolutionary optimization [78, 79]. However, this advantage is only true in a restricted area: A policy is

able to make accurate decisions within situations that are sufficiently similar to those situations it has been trained to (i.e. similar to the training simulation). For other situations, its extrapolation-ability is necessary to still make good decisions. As soon as specific situations are too different from the training simulation, obviously the policy-based control becomes useless. In such a case, the simulation model would need to be adapted in order to correspond to such situations, and respective policies would need to be adapted / relearned. Figure 7.2 illustrates this issue. Here, the gray-colored region indicates the predictable volatility of the system, which gets approximated by the simulation model where finally the policy gets trained to. Real-world systems are often faced by some kind of drifts that change the system in way that is not predictable and finally force its model to be adapted and necessitate the adaptation of control policies too.

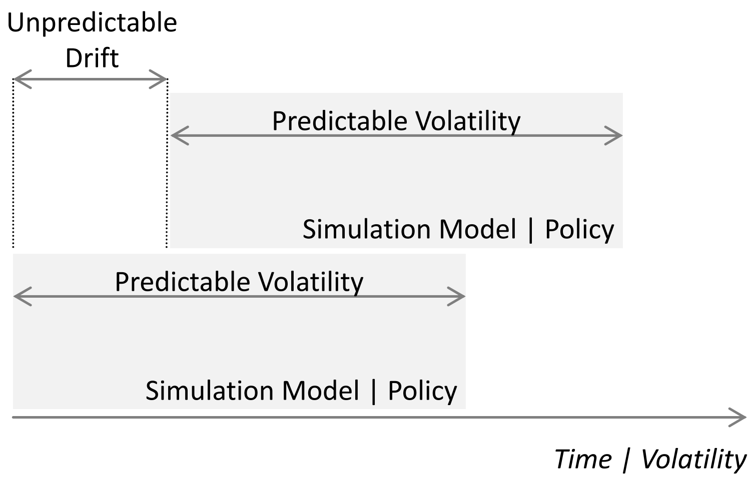

Figure 7.2: System Dynamics: Drift of Real-World System

Hence, if being learned accurately, policy-based control is valid for systems where their behavior, dynamics and uncertainty are adequately predictable. If such a system changes over long time, and the simulation no more matches the real-world, special techniques will need to be applied for learning a new policy or adapting the existing one. Numerous approaches already exist in literature, using memories of already evaluated solutions, sub-populations or immigration methodologies in order to adapt the evolutionary search to changing positions of the desired optima over time. A comprehensive overview on such concepts can be found in [118]. Hence, with policy function approximation, solution adaptation is avoided on a short-time scale, where the policy is able to derive decisions for uncertain and dynamic states. On a long-time scale, solution adaptation is still necessary in order to meet potential drifts of the system (that cause a mismatch between simulation model and real-world).

Function Approximation
As already mentioned, the used function approximation technique is a crucial choice for policy-based control. Within this work, two distinct approximation techniques were used, namely metamodel-based approximation with polynomials as well as metamodel-less approximation with GP. Especially the latter is highly promising, since with GP one is able to evolve any executable expression without the need of a-priori assumptions on the policy's structure. Since the descriptive power of GP is high and nearly any behavior can be approximated theoretically, it is a valid choice for investigations herein. However, many other function approximations could be applied, being more or less suitable to herein treated problems.

Information Technology - Availability of Data
Since with policy-based approximate optimal dynamic control decisions are executed in a distributed manner, appropriate information needed for evaluating abstract rules is needed at the point where the policy is applied. Within this work, it has been assumed that all data that is needed for making control decisions is always available. When handling the classical OPF problem (i.e. power grid operation with generation unit scheduling) in Chapter 5, this issue can be neglected since all data that is needed here for policy-based control is also available in traditional power grid operation. Hence, all data is available.
When treating smart grid related OPF control issues such as the proposed holistic EV charging problem, much more information is needed for distributed control. Here, the assumption that future smart grid architectures are able to provide these data is essential. In [61], researchers have already shown the sufficiency of existing IT infrastructures for the general aim of distributed demand response in smart grids, also substantiating this assumption.

Time Scale
In the introductory chapter on power grid operation, an important statement characterizes this work: namely that steady-state representation of the system is sufficient when making decisions on a time scale of minutes and

more. In literature on future power grid control when optimizing decisions for e.g. distributed generation units or controllable load devices, most often such high time scales are considered. In the same way, this work assumes similar time scales for decision making. Hence, even if dynamic situations are regarded, the consideration of multi-step problem definitions where each step is modeled by a steady-state formulation is sufficient. However, lower time-scales necessitating the application of transient representations could be promising as well, but are not being treated herein.

Electric Vehicles - Load Modeling

Within Chapter 6, electric vehicle charging control has been stated as generic problem description for OPF related power flow control in smart grids. Here, this technology was defined to control time and (positive) magnitude of charging power, while different attempts are made in literature in order to enable also to discharge electric vehicles and provide regulation services to the power grid (V2G concept). This discharging has not been considered due to two reasons: from the optimization problem formulation point of view, it makes little difference since the only change is to allow a negative value for the charging power. Additionally, more accurate battery models would need to be integrated into the simulation. The second reason is that a potential discharging technology will not be practically relevant in coming developments, since battery costs are too high [89].

7.3.2 Potential Future Work

Having defined the spectrum as well as the borderlines of performed investigations, potential future research work can be derived.

From the applications' point of view, various additional developments could be performed. For example, instead of learning policies for load control, a set of policies for power system restoration after an outage could be evolved using a simulation model that describes system faults. Additionally, the time-scale plays an important role on the application's side: While in this thesis decisions with a time range of 15 minutes and more have been treated, decisions on a lower time scale (i.e. seconds or less than seconds) could be promising for enabling frequency regulation, for example to learn control policies for a device that provides spinning reserves and thus participates in the primary reserve market. In such a case, transient power flow modeling would be needed within the simulation.

Another potential research direction from an applications' point of view could address the question on how to implement policy-based control into modern/future power grid architectures. While in this work the complete availability of data for decision making has been assumed, the question of implementability could be of great concern. Also a possible reduction of needed data (i.e. a reduction of abstract rules) could be of interest for implementation reasons.

Considering methodical research extensions, an important issue concerns system dynamics as discussed above, so the adaptation of learned policies over time. So far, policies are learned in order to perform accurately in a real-world that is described through the simulation model. It has been assumed that the simulation model sufficiently accurately describes the real-world. But, what if the real-world system drifts (for example because of seasonal changes to the power system)? In such a case, the policies would need to be adapted or even completely relearned. As mentioned above, many techniques already exist for dynamic evolutionary optimization in order to adapt/relearn solutions. Here, research questions could address the task of adapting existing policies from the evolutionary computation point of view, or the issue on how to recognize a system's drift respectively the inadequacy of the simulation model.

Another methodical extension would be the investigation of using other function approximators as metamodel. Numerous fruitful approximation techniques exist such as ANNs, decision trees or fuzzy logic, all of them with specific features that could or could not be advantageous to the herein problem domain. In case of genetic programming, the application of extended grammars could be promising. For example, the integration of conditional operators could enable the evolution of more complex policies that are able to distinguish between different situations via a higher-level conditional query (at a higher node in the tree structure) and apply accurate policies to them that are subjacent to this query.

Summing up, the developed techniques provide a fruitful ground for manifold future investigations in the field of smart electric grid engineering as well as policy function approximation in general.

7.4 Publications

The author published several works in the context of this thesis, that will now be listed and briefly discussed with respect to their main achievements. Their sequence fits well to the structure of this thesis.

Conference Proceedings and Journal Papers

To conclude the very beginning of the related research work, [**53**] has been published for stating the principal idea of applying evolutionary simulation optimization to power grid operation applications. While explaining the fundamentals of this technique, the special advantages have been discussed with respect to power flow optimization in future smart grids.

[**49**] extended this work by applying simulation optimization for solving the general OPF for a benchmark instance. By unifying HeuristicLab with Matlab for power flow simulation in PSAT (Power System Analysis Toolbox), achievable solution qualities have been discussed. However, this publication still treated the steady-state OPF, i.e. the computation of a static solution within one discrete state.

Taking the general OPF as suitable benchmark problem, [**46**] presented for the first time the idea of creating a dynamic (multiperiod) problem out of the steady-state OPF for testing simulation-based policy function approximation on it. This publication built the fundament for Chapter 5 herein which got extended both in terms of the methodology as well as considering the implemented problem instances.

[**55**] extended this work and uses argumentations quite at the same level as herein when introducing abstract rules in comparison with system variable synthesis. Also, the idea of creating separate training and test scenarios as well as the technique of coevolving multiple policies in parallel has been discussed.

A little earlier in [**51**], the framework of aggregating power grid considerations with traffic simulation has been presented, where discrete-event simulation was used for describing fleets of electric vehicles as well as their interrelated behavior. Even if a dynamic control issue has been treated, still a real-valued solution representation was used according to lookahead optimization, while directly computing optimal values for P_{EV} (respectively $\overline{u_{EV}}$).

The existing problem description has been extended to a holistic problem formulation of EV charging control in [**52**], which was able to integrate both the traffic simulation together with a power flow simulation of a given distribution grid. While incorporating uncertainties of the traffic simulation as well as fluctuating stochastic supply models from renewables, the general fundament for herein considerations in Chapter 6 was provided. Using a real-valued solution representation as before, some experiments demonstrated the validity of this approach to optimal EV charging control.

Taking the developed simulation optimization framework and extending existing experiments, the author completed a first journal-paper submission [**44**] to the International Journal of Energy Optimization and Engineering at the very end of 2011. After revisioning the submission, the paper is scheduled for publication in Volume 2, Issue 3, 2013. In addition to the previous papers, this work presented an adaptive sampling scheme based on offspring selection genetic algorithms (OSGA) that automatically adapts the number of simulation samples when estimating the fitness of a solution.

Since the very beginning of his work the author identified the strong potential of simulation optimization for the engineering society. Followingly, he decided to publish a journal article that gives profound discussions of simulation optimization with HeuristicLab, highlighting the benefits that this technology is able to provide to engineers that work in the optimization field. This article shall now be appearing in the IEEE Transactions on Industrial Informatics (5-year impact: 3.14) [**54**].

The author recognized the potential of dynamic optimization in the context of charging control. After formulating policy-based approaches, [**45**] first presented the idea of formulating abstract rules for synthesizing flexible charging policies out of them. The author additionally provided more detailed discussions on the synthesis with and without metamodels, which lead to major insights forcing the application of genetic programming for policy approximation.

On top of that, [**50**] presented comparisons to lookahead optimization and reachable solution qualities with both deterministic as well as policy-based control optimization.

In order to put the evolved technology of simulation-based evolutionary policy function approximation more in the context of near-future smart grid implementations, **[47]** stated it as suitable technology to be implemented to e-mobility aggregators and discussed both its advantages as well as implementation-related abilities.

Summing up, the author strongly intended to bring his work and ideas to the research society. Besides the journal articles, he is proud of having presented them at highly established scientific conferences such as the *Genetic and Evolutionary Computation Conference (GECCO)*, the *EvoStar*, the *IEEE Conference on Probabilistic Methods Applied to Power Systems (PMAPS)* or the *IEEE Symposium Series on Computational Intelligence (SSCI)*.

Talks

Several talks have been used for discussing the actual developments in the context of this thesis: in May 2011, an invited talk named *Metaheuristic Open Source Power System Optimization and Analysis using HeuristicLab* at KTH in Stockholm was done discussing both the simulation-based optimization of charging decisions as well as the general application of metaheuristic techniques to the power engineering domain.
In recent years, an Austrian research group from TU Vienna established the annual national conference on smart grids *ComForEn*[1], where the author took the chance to discuss his work in front of the national scientific community annually since 2010.
At the EvoStar-conference in April 2013 within the EvoTransfer-workshop, the author debated the general potential of applying soft computing techniques like metaheuristics to manifold issues in smart grid engineering, such as probabilistic infrastructure planning, demand forecasting or demand-side dynamic control optimization.

[1] ComForEn - Kommunikation für Energiesysteme

Bibliography

[1] M. A. Abido. Optimal power flow using particle swarm optimization. *Electrical Power And Energy Systems*, 24(1):563–571, 2002.

[2] M. A. Abido. Environmental/economic power dispatch using multiobjective evolutionary algorithms. *IEEE Transactions on Power Systems*, 18(4):1529–1537, 2003.

[3] M. A. Abido. Multiobjective evolutionary algorithms for electric power dispatch problem. *IEEE Transactions on Evolutionary Computations*, 10(3):315–329, 2006.

[4] A. A. Abou El Ela, M. A. Abido, and S. R. Spea. Optimal power flow using differential evolution algorithm. *Electric Power Systems Research*, 80(7):878–885, 2010.

[5] N. Acharya, P. Mahat, and N. Mithulananthan. An analytical approach for DG allocation in primary distribution network. *Electrical Power and Energy Systems*, 28(10):669–678, 2006.

[6] M. Affenzeller, S. Wagner, S. Winkler, and A. Beham. *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. CRC Press, 2009.

[7] Y. Alinejad-Beromi, M. Sedighizadeh, M. R. Bayat, and M. E. Khodayar. Using genetic alghoritm for distributed generation allocation to reduce losses and improve voltage profile. In *Proceedings of the 42nd International Universities Power Engineering Conference*, 2007.

[8] O. Alsac and B. Stott. Optimal load flow with steady-state security. *IEEE Transactions on Power Apparatus and Systems*, PAS-93(2):745 – 751, 1974.

[9] S. M. Amin and B.F. Wollenberg. Toward a smart grid: Power delivery for the 21st century. *IEEE Power and Energy Magazine*, 3(5):34–41, 2005.

[10] P. H. Andersen, J. A. Mathews, and M. Raska. Integrating private transport into renewable energy policy: The strategy of creating intelligent recharging grids for electric vehicles. *Energy Policy*, 37(7):2481–2486, 2009.

[11] S. Andradottir. A review of simulation optimization techniques. In *Proceedings of the Winter Simulation Conference*, pages 151–158, 1998.

[12] L. Bam and W. Jewell. Review: Power system analysis software tools. In *IEEE Power & Energy Society General Meeting*, 2005.

[13] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1957.

[14] H. G. Beyer and H. P. Schwefel. Evolution strategies - a comprehensive introduction. *Springer Natural Computing*, 1(1):3–52, 2002.

[15] M. Bhaskar, A. Jain, and V. Srinath. Wind speed forecasting: Present status. In *Proceedings of the International Conference on Power System Technology*, pages 1–6, 2010.

[16] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.

[17] BMVIT. Federal Ministry for Transport, Innovation and Technology. Verkehr in Zahlen 2011. http://www.bmvit.gv.at/verkehr/gesamtverkehr/statistik/viz11/, 2013.

[18] J. Branke and C. Schmidt. Selection in the presence of noise. *Lecture notes in computer science*, 2723:204–215, 2003.

[19] Y. Carson and A. Maria. Simulation optimization: methods and applications. In *Proceedings of the Winter Simulation Conference*, pages 118–126, 1997.

[20] R. D. Christie. Partial description of the ieee common data format for the exchange of solved load flow data, http://www.ee.washington.edu/research/pstca/formats/cdf.txt, 2013.

[21] R. D. Christie. Power systems test case archive, http://www.ee.washington.edu/research/pstca/, March 2013.

[22] K. Clement, S. De Breucker, and J. Driesen. The analysis of objective functions applied to the coordination problem of phevs to minimize the grid impact. In *Proceedings of the EVS-25, Shenzen, China*, 2010.

[23] K. Clement, E. Haesen, and J. Driesen. The impact of uncontrolled and controlled charging of plug-in hybrid electric vehicles on the distribution grid. In *Proceedings of the European Ele-Drive Conference*. EET-2008, 2008.

[24] K. Clement, E. Haesen, and J. Driesen. Stochastic analysis of the impact of plug-in- hybrid electric vehicles on the distribution grid. In *Proceedings of the 20th International Conference on Electricity Distribution*. CIRED, 2009.

[25] European Commission. EU energy in figures - statistical pocketbook 2012. Technical report, European Union, 2013.

[26] International Electrotechnical Commission. IEC 61850 - communication networks and systems in substations. Switzerland, 2003.

[27] International Electrotechnical Commission. IEC 61851 - electric vehicle conductive charging system. Switzerland, 2010.

[28] H. Cruse. *Neural Networks as Cybernetic Systems, 2nd Edition*. Brains, Minds & Media, Bielefeld, Germany, 2006.

[29] D. Dallinger, D. Nestle, and J. Ringelstein. Indirect control of plug-in hybrid vehicles with variable tariffs. In *Proceedings of the European Conference Smart Grids and Mobility*, 2009.

[30] G. B. Dantzig. Application of the simplex method to a transportation problem. In T. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 359–373. John Wiley and Sons, New York, 1951.

[31] G. B. Dantzig. The allocation of aircrafts to routes: An example of linear programming under uncertain demand. *Management Science*, 2(1):45–73, 1956.

[32] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification, Second edition*. John Wiley & Sons, 2001.

[33] M. Eldred, A. Giunta, J.S.F. Wojtkiewicz, and T. Trucano. Formulations for surrogate-based optimization under uncertainty. In *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2002.

[34] M. Fan, V. Vittal, G. T. Heydt, and R. Ayyanar. Probabilistic power flow studies for transmission systems with photovoltaic generation using cumulants. *IEEE Transactions on Power Systems*, 27(4):2251 – 2261, 2012.

[35] S. Ficici and J. Pollack. A game-theoretic approach to the simple coevolutionary algorithm. In *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI)*, pages 467–476. Springer, 2000.

[36] M. C. Fu. Feature article: Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14(3):192–215, 2002.

[37] M. C. Fu, F.W. Glover, and J. April. Simulation optimization: A review, new developments, and applications. In *Proceedings of the Winter Simulation Conference*, pages 83–95, 2005.

[38] A. Gosavi. *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Springer, Berlin, 2003.

[39] W. Group. Common format for exchange of solved load flow data. *IEEE Transactions on Power Apparatus and Systems*, PAS-92(6):1916 – 1925, 1973.

[40] S. Han and K. Sezaki. Development of an optimal vehicle-togrid aggregator for frequency regulation. *IEEE Transactions on Smart Grid*, 1(1):65–72, 2010.

[41] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[42] N. Higgins, V. Vyatkin, N. Nair, and K. Schwarz. Intelligent decentralised power distribution automation with IEC 61850, IEC 61499 and holonic control. *IEEE Transactions on Systems, Machine and Cybernetics*, 41(1):81–92, 2011.

[43] D. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D: Nonlinear Phenomena*, 42(1):228 – 234, 1990.

[44] S. Hutterer and M. Affenzeller. Probabilistic electric vehicle charging optimized with genetic algorithms and a two-stage sampling scheme. *International Journal of Energy Optimization and Engineering (accepted for publication)*, 2013.

[45] S. Hutterer, M. Affenzeller, and F. Auinger. Policy evolution for large scale electric vehicle charging control. In *Proceedings of the International Conference on Harbor, Maritime & Multimodal Logistics Modelling and Simulation (HMS 2013)*, 2012.

[46] S. Hutterer, M. Affenzeller, and F. Auinger. Evolutionary algorithm based control policies for flexible optimal power flow over time. In *Proceedings of the EvoApplications 2013, Lecture Notes in Computer Science (LNCS)*, volume 7835, pages 152–161, 2013.

[47] S. Hutterer, M. Affenzeller, and F. Auinger. Evolutionary computation enabled controlled charging for e-mobility aggregators. In *Proceedings of the IEEE Symposium Series on Computational Intelligence, Workshop on Computational Intelligence Applications in Smart Grid (IEEE CIASG 2013)*, 2013.

[48] S. Hutterer and D. Auinger, F.and Rumetshofer. Simulation und Optimierung von Ladevorgängen an steuerbaren E-Ladesäulen. In *Proceedings of the ComForEn 2012*, pages 65–72, 2012.

[49] S. Hutterer, F. Auinger, and M. Affenzeller. Comparison of metaheuristic algorithms for simulation based opf computation. In *12th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE)*, pages 1–4. IEEE, 2011.

[50] S. Hutterer, F. Auinger, and M. Affenzeller. Evolutionary optimization of multi-agent control strategies for electric vehicle charging. In *Companion Publication of the 2012 Genetic and Evolutionary Computation Conference*, pages 3–10, 2012.

[51] S. Hutterer, F. Auinger, and M. Affenzeller. Heuristic power scheduling of electric vehicle battery charging based on discrete event simulation. In *Lecture Notes in Computer Science*, volume 6927, pages 311–318, 2012.

[52] S. Hutterer, F. Auinger, and M. Affenzeller. Metaheuristic optimization of electric vehicle charging strategies in uncertain environment. In *Proceedings of the 12th International Conference on Probabilistic Methods Applied to Power Systems*, 2012.

[53] S. Hutterer, F. Auinger, M. Affenzeller, and G. Steinmaurer. Overview: A simulation based metaheuristic optimization approach to optimal

power dispatch related to smart electric grids. In *Life System Modeling and Intelligent Computing (LNCS 6329)*, pages 368–378, 2010.

[54] S. Hutterer, A. Beham, F. Auinger, M. Affenzeller, and S. Wagner. Software-enabled investigation in metaheuristic power grid optimization. *IEEE Transactions on Industrial Informatics (accepted for publication)*, 2013.

[55] S. Hutterer, S. Vonolfen, and M. Affenzeller. Genetic programming enabled evolution of control policies for dynamic stochastic optimal power flow. In *Companion Publication of the 2013 Genetic and Evolutionary Computation Conference*, pages 1529–1536, 2013.

[56] A. Keane and M. O'Malley. Optimal allocation of embedded generation on distribution networks. *IEEE Transactions on Power Systems*, 20(3):1640–1646, 2005.

[57] W. Kempton and J. Tomic. Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy. *Journal of Power Sources*, 144:280 – 294, 2005.

[58] A. Keyhani and M. Albaijat. *Smart Power Grids 2011.* Springer, Berlin, 2012.

[59] D. K. Khatod, V. Pant, and J. Sharma. Evolutionary programming based optimal placement of renewable distributed generators. *IEEE Transactions on Power Systems*, 28(2):683–695, 2012.

[60] T. T. Kim and H. V. Poor. Scheduling power consumption with price uncertainty. *IEEE Transactions on Smart Grid*, 2(3):519–527, 2011.

[61] E. Koch and M. A. Piette. Architecture concepts and technical issues for an open, interoperable automated demand response infrastructure. Technical report, Lawrence Berkeley National Laboratory, 2008.

[62] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* The MIT Press, 1992.

[63] P. Kundur, J. Paserba, and V. Ajjarapu. Definition and classification of power system stability. *IEEE Transactions on Power Systems*, 19(3):1387–1401, 2004.

[64] L. L. Lai, J. T. Ma, R. Yokoyama, and M. Zhao. Improved genetic algorithms for optimal power flow under both normal and contingent

operation states. *Electrical Power And Energy Systems*, 19(5):287–292, 1997.

[65] A.J. Lamadrid, T. Mount, R. Zimmerman, C.E. Murillo-Sanchez, and L. Anderson. Alternate mechanisms for integrating renewable sources of energy into electricity markets. In *Proceedings of the 2012 IEEE Power and Energy Society General Meeting*, pages 1–8. IEEE, 2012.

[66] W. B. Langdon and R. Poli. *Foundations of Genetic Programming*. Springer, 2002.

[67] A. M. Law and McComas M. G. Simulation-based optimization. In *Proceedings of the Winter Simulation Conference*, pages 41–44, 2002.

[68] C. Leitinger. *Netzintegration von solarer elektrischer Mobilität - Auswirkungen auf das elektrische Energiesystem*. PhD thesis, Vienna Technical University, Institute of Energy Systems and Electrical Drives, 2011.

[69] Jiaqi Liang, G. K. Venayagamoorthy, and R. G. Harley. Wide-area measurement based dynamic stochastic optimal power flow control for smart grids with high variability and uncertainty. *IEEE Transactions on Smart Grid*, 3(1):59–69, 2012.

[70] M. Lydia and S. S. Kumar. A comprehensive overview on wind power forecasting. In *Proceedings of the International Power and Energy Conference*, pages 268–273, 2010.

[71] P. Mohseni and R. G. Stevie. Electric vehicles: Holy grail or fools gold. In *IEEE Power & Energy Society General Meeting*, 2009.

[72] J. A. Momoh. Toward dynamic stochastic optimal power flow. In J. Si, A. Barto, W. Powell, and D. Wunsch, editors, *Handbook of Learning and Approximate Dynamic Programming*, pages 561–598. Wiley-Interscience, 2004.

[73] J. A. Momoh. *Electric Power System Applications of Optimization*. 2nd Edition, CRC / Taylor & Francis, 2009.

[74] J. A. Momoh and E. Zivi. Control, optimization, security, and self-healing of benchmark power systems. In J. Si, A. Barto, W. Powell, and D. Wunsch, editors, *Handbook of Learning and Approximate Dynamic Programming*, pages 599–637. Wiley-Interscience, 2004.

[75] K. Nekooei, M. M. Farsangi, H. Nezamabadi-Pour, and K. Y. Lee. An improved multi-objective harmony search for optimal placement of DGs in distribution systems. *IEEE Transactions on Smart Grid*, 4(1):557–567, 2013.

[76] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7(15):308–313, 1965.

[77] A. Newell and H. Simon. *Human Problem-Solving*. Prentice-Hall, Englewood Cliffs, N.J., 1972.

[78] T. T. Nguyen, S. Yang, and J. Branke. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, 6:1 – 24, 2012.

[79] T. T. Nguyen and X. Yao. Continuous dynamic constrained optimization - the challenges. *IEEE Transactions on Evolutionary Computation*, 16(6):769–786, 2012.

[80] Clusterland Oberösterreich. Clean motion offensive, http://www.cleanmotion.at/, 2013.

[81] V. S. Pappala. Power system optimization under uncertainties: A PSO approach. In *Proceedings of the IEEE Swarm Intelligence Symposium*, pages 1–8, 2008.

[82] J. Parejo, A. Ruiz-Cortés, S. Lozano, and P. Fernandez. Metaheuristic optimization frameworks: a survey and benchmarking. *Soft Computing*, 16(3):527–561, 2012.

[83] C. W. Potter, A. Archambault, and K. Westrick. Building a smarter grid through better renewable energy information. In *Power Systems Conference and Exposition*, pages 1–5, 2009.

[84] M. Potter and K. De Jong. A cooperative coevolutionary approach to function optimization. In *Proceedings of the Third International Conference on Parallel Problem Solving from Nature (PPSN III)*, pages 249–257. Springer-Verlag, 1994.

[85] M. Potter and K. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000.

[86] W. B. Powell, H. P. Simao, and B. Bouzaiene-Ayari. Approximate dynamic programming in transportation and logistics: A unified framework. *European Journal of Transportation and Logistics*, 1(3):237–284, 2012.

[87] Warren B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality, Second Edition*. Wiley-Interscience, 2011.

[88] H. Raiffa. *Decision Analysis: Introductory Lectures on Choices Under Uncertainty*. Addison-Wesley, Reading, 1968.

[89] R. Rezania. *Integration of Electric Vehicles in the Austrian Electricity System*. PhD thesis, Vienna Technical University, Institute of Energy Systems and Electrical Drives, 2013.

[90] Y. Rinott. On two-stage selection procedures and related probability inequalities. *Communications in Statistics - Theory and Methods*, 7(8):799–811, 1978.

[91] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[92] C. Rosin and R. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5(1):1–29, 1996.

[93] A. Saber and G. Venayagamoorthy. Optimization of vehicle-to-grid scheduling in constrained parking lots. In *Proceedings of the IEEE PES Power and Energy Society General Meeting*, 2009.

[94] A. Saber and G. K. Venayagamoorthy. Plug-in vehicles and renewable energy sources for cost and emission reductions. *IEEE Transactions on Industrial Electronics*, 58(4):1229–1238, 2011.

[95] J.-C. Sabonnadire and N. Hadjsaid. *SmartGrids*. Wiley & Sons, London, 2011.

[96] A. J. Schwab. *Elektroenergiesysteme, 3rd Edition*. Springer, Berlin Heidelberg, 2011.

[97] G. Shuang, K.T. Chau, C.C. Chan, and Diyun Wu. Loss analysis of vehictle-to-grid operation. In *Proceedings of the IEEE Vehicle Power and Propulsion Conference (VPPC)*, pages 1–6, 2010.

[98] G. Shuang, K.T. Chau, L. Chunhua, and W. Diyun. SMES control for power grid integrating renewable generation and electric vehicles. *IEEE Transactions on Applied Superconductivity*, 22(3), 2012.

[99] R. Sioshansi. Modeling the impacts of electricity tariffs on plug-in hybrid electric vehicle charging, costs, and emissions. *OPERATIONS RESEARCH*, 60(2):1–11, 2012.

[100] E. Sortomme, M. M. Hindi, S. D. J. McPherson, and Venkata M. Coordinated charging of plug-in hybrid electric vehicles to minimize distribution system losses. *IEEE Transactions on Smart Grid*, 2(1):198 – 205, 2011.

[101] T. Sousa, H. Morais, Z. Vale, P. Faria, and J. Soares. Intelligent energy resource management considering vehicle-to-grid: a simulated annealing approach. *IEEE Transactions on Smart Grid*, 3(1):535–542, 2012.

[102] P. Stagge. Averaging efficiently in the presence of noise. *Lecture notes in computer science*, 1498:188–197, 1998.

[103] W. Su and M.-Y. Chow. Performance evaluation of an EDA-based large-scale plug-in hybrid electric vehicle charging algorithm. *IEEE Transactions on Smart Grid*, 3(1):308–315, 2012.

[104] J.R. Swisher, S.H. Jacobson, P.D. Hyden, and L.W. Schruben. A survey of simulation and optimization techniques and procedures. In *Proceedings of the Winter Simulation Conference*, pages 119–128, 2000.

[105] V. N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, 1998.

[106] G. K. Venayagamoorthy. Dynamic, stochastic, computational, and scalable technologies for smart grids. *IEEE Computational Intelligence Magazine*, 6(3):22–35, 2011.

[107] G. K. Venayagamoorthy, G. Harley, and D. Wunsch. Applications of approximate dynamic programming in power systems control. In J. Si, A. Barto, W. Powell, and D. Wunsch, editors, *Handbook of Learning and Approximate Dynamic Programming*, pages 479–515. Wiley-Interscience, 2004.

[108] G. K. Venayagamoorthy and R. G. Harley. Adaptive critic designs for optimal control of power systems. In *Proceedings of the 13th Conference on Intelligent Systems Application to Power Systems*, pages 136 – 148, 2005.

[109] J. G. Vlachogiannis. Probabilistic constrained load flow considering integration of wind power generation and electric vehicles. *IEEE Transactions On Power Systems*, 24(4), 2009.

[110] Z. Wan and T. Igusa. Adaptive sampling for optimization under uncertainty. In *Proceedings of the 4th International Symposium on Uncertainty Modelling and Analysis*, pages 423–428, 2003.

[111] H. Wang, C. E. Murillo-Snchez, R. D. Zimmerman, and R. J. Thomas. On computational issues of market-based optimal power flow. *IEEE Transactions on Power Systems*, 22(3):1185–1193, 2007.

[112] P. J. Werbos. Optimization methods for brain-like intelligent control. In *Proceedings of the 34th Conference on Decision & Control*, pages 579–584, 1995.

[113] P. J. Werbos. ADP: Goals, opportunities and principles. In J. Si, A. Barto, W. Powell, and D. Wunsch, editors, *Handbook of Learning and Approximate Dynamic Programming*, pages 3–44. Wiley-Interscience, 2004.

[114] P. J. Werbos. Computational intelligence for the smart grid - history, challenges, and opportunities. *IEEE Computational Intelligence Magazine*, 6(3):14–21, 2011.

[115] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[116] A. J. Wood and B. F. Wollenberg. *Power Generation, Operation, and Control, 2nd Edition*. Wiley-Interscience, 1996.

[117] J. Xu and V. W. S. Wong. An approximate dynamic programming approach for coordinated charging control at vehicle-to-grid aggregator. In *Proceedings of the IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 279–284, 2011.

[118] S. Yang and X. Yao. *Evolutionary Computation for Dynamic Optimization Problems*, volume 490 of *Studies in Computational Intelligence*. Springer Berlin Heidelberg, 2013.

[119] M. Zhongjing, D.S. Callaway, and I.A. Hiskens. Decentralized charging control of large populations of plug-in electric vehicles. *IEEE Transactions on Control Systems Technology*, 21(1):67–78, 2013.

[120] R. D. Zimmerman, C. E. Murillo-Sanchez, and D. Gan. Matpower - a matlab power system simulation package; http://www.pserc.cornell.edu/matpower/#docs, March 2013.

Appendix A

Appendix

A.1 Abstract Rules for OPF: Matlab Code

Some explanation: *casex* is a structure containing the MATPOWER model data, see the MATPOWER manual [120]. *casex.bus* returns a matrix with the bus data, *casex.branch* returns the branch data, and *casex.gen* delivers generator data. An extra matrix is used for the cost coefficients of generating units, namely *casex.gencost*.

```
% busnNr gives the identifier of the actual bus

% LLF - local load factor of supply unit
% casex.bus(busNr,3): the load buses' active power values
% casex.gen(casex.gen(:,1)==busNr,9): maximum generation capacity

llf=casex.bus(busNr,3)/casex.gen(casex.gen(:,1)==busNr,9);


% NLF - neighboring load lactor
% sum of active load at connected buses and their neighbors divided
% by the maximum active power output at those buses
% neighborLoad returns the active power load values of neighbored
% buses, neighborPower addresses their generation capacity

nlf=sum(neighborLoad)/sum(neighborPower);

% NRLF - neighbouring reactive load factor
% sum of reactive load at directly connected buses and their
% neighbors divided by maximum reactive power output at those buses
nrlf=sum(neighborReactiveLoad)/sum(neighborReactivePower);
```

```
% GLF - global load factor
% sum of total active power load within grid divided by sum of
% maximum active power generation

glf=sum(casex.bus(:,3))/sum(casex.gen(:,9));

% GRLF - global reactive load factor
% sum of total reactive load within grid divided by sum of maximum
% reactive power output

grlf=sum(casex.bus(:,4))/sum(casex.gen(:,4));

% MARF - maximum rating factor
% maximum MVAR rating of connected branches divided by maximum MVAR
% rating of all branches
% neighborBranches: the MVAR ratings of the neighboring branches;
% casex.branch(:,6): the MVAR ratings of all branches;

marf=max(neighborBranches)/max(casex.branch(:,6));

% MERF - mean rating factor
% mean MVAR rating of connected branches divided by maximum MVAR
% rating of all branches

merf=mean(neighborBranches)/max(casex.branch(:,6));

% LCCF - linear cost coefficient
% linear cost coefficient of generator divided by maximum linear
% cost coefficients of all generators
% index is the identifier of the generation unit
lccf=1-casex.gencost(index,6)/max(casex.gencost(:,6));

% PCCF - polynomial cost coefficient
% polynomial cost coefficient of generator divided by maximum
% polynomial cost coefficients of all generators
pccf=1-casex.gencost(index,5)/max(casex.gencost(:,5));
```

A.2 IEEE Test Case Constraints Data

MATPOWER Case Format: Version 2
All values are give in per unit (p.u.) notation, with base-MVA = 100, in ascending order with the branch Nr. The code can be inserted into MATPOWER as is for setting the branch flow constraint values (branch rating).

IEEE 14-Bus Branch Rating

```
casexx.branch(:,6)=[120 65 36 65 50 65 45 55 32 45 18 32 32 32 32 32 32 12
12 12];
```

IEEE 30-Bus Branch Rating

```
casexx.branch(:,6)=[
130 130 65 130 130 65 90 70 130 32 65 32 65 65 65 65 32 32 32 16 16 16 16
32 32 32 32 32 32 16 16 16 16 16 16 65 16 16 16 32 32
];
```

IEEE 57-Bus Branch Rating

```
casexx.branch(:,6)=[
170 120 120 40 40 40 60 250 40 40 40 40 40 60 250 120 120 60 40 40 40 120
40 40 120 60 60 120 40 40 40 40 40 40 40 40 40 40 40 40 120 40 40 40 40
40 40 40 40 40 40 40 40 40 40 40 40 60 120 120 40 40 40 40 60 60 40 40 40
40 40 60 40 40 40 40 40 40 40 40
];
```

IEEE 118-Bus Branch Rating

```
casexx.branch(:,6)=[
175 175 500 175 175 175 500 500 500 175 175 175 175 175 175 175 175 175 175
175 500 175 175 175 175 175 175 175 175 175 500 500 500 175 175 500 175 500
175 175 140 175 175 175 175 175 175 175 175 500 500 175 175 175 175 175 175
175 175 175 175 175 175 175 175 175 175 175 175 175 175 175 175 175 175 175
175 175 175 175 175 175 175 175 175 175 175 175 175 500 175 175 500 500 500
500 500 500 500 175 175 500 175 500 175 175 500 500 175 175 175 175 175 175
175 500 175 175 175 175 175 175 500 500 175 500 500 200 200 175 175 175 500
500 175 175 500 500 500 175 500 500 175 175 175 175 175 175 175 175 175 175
200 175 175 175 175 175 175 175 175 175 500 175 175 175 175 175 175 175 175
175 175 175 175 175 175 175 500 175 175 175 500 175 175 175
];
```

A.3 OPF Benchmark Solutions for IEEE 14-Bus and 30-Bus Cases

The following table gives the control variables ($\overline{u}$) of the best found solutions for the IEEE 14-Bus and 30-Bus test cases. The subscripted numbers indicate the bus numbers of the respective controlled units. While units for variables P_G, V_G and Q_C are modeled directly at buses, transformers (T_{tap}) are modeled to be between two buses in MATPOWER. While the units for P_G, V_G and Q_C are MW, p.u., and MVAR respectively, T_{tap} represents a unit-less ratio.

14-Bus Case		30-Bus Case	
Variable	Value	Variable	Value
V_{G1}	1.1	V_{G1}	1.1
P_{G2}	28.35	P_{G2}	43.98
V_{G2}	1.1	V_{G2}	1.1
P_{G3}	59.88	P_{G5}	19.37
V_{G3}	1.08	V_{G5}	1.1
P_{G6}	10	P_{G8}	19.59
V_{G6}	1.1	V_{G8}	1.1
P_{G8}	10	P_{G11}	10
V_{G8}	1.1	V_{G11}	1.1
Q_{C9}	19	P_{G13}	12
T_{tap4_7}	0.98	V_{G13}	1.04
T_{tap4_9}	0.97	Q_{C10}	0.41
T_{tap5_6}	0.93	Q_{C24}	2.98
		T_{tap6_9}	1.06
		T_{tap6_10}	1.1
		T_{tap4_12}	1.1
		T_{tap28_27}	1.1

Table A.1: Best Found Solutions for 14-Bus And 30-Bus Cases

A.4 Results to IEEE 14-Bus SVS

Solutions $P_{G,n}$ for $n = 2...5$ ($P_{G,1}$ = non-controllable slack bus). In order to increase the readability, the constants are extracted and shown in subsequent tables.

$$P_{G_2} = \frac{rc_1 * P_{B_9} * (rc_2 - rc_3 * P_{B_{13}} - rc_4 * P_{B_2} - rc_5 * P_{B_6} + rc_6 * c1 + rc_7 * P_{B_4})}{rc_8 * P_{B_6} + rc_9 * P_{B_{13}} - \frac{rc_{10}}{P_{B_{12}}} + rc_{11} * P_{B_2} + rc_{12} * P_{G_5}^{max} + rc_{13} * c1} + rc_{14} + \frac{rc_{15} * P_{B_2}}{P_{b_5}^{max}} + \frac{rc_{16} * (rc_{17} + \frac{rc_{18}}{P_{B_{12}}} - rc_{19} * P_{B_2} - rc_{20} * P_{B_{13}} - rc_{21} * P_{B_6} + rc_{22} * c_1)}{(rc_{23} * P_{B_{14}} - rc_{24})} + \frac{rc_{25} * P_{b_{19}}^{max}}{rc_{26} * P_{B_1} - (\frac{rc_{27}*(rc_{28}*P_{B_6} - rc_{29}*c_1 + rc_{30}*P_{B_2})}{P_{b_5}^{max}}} \tag{A.1}$$

$$P_{G_3} = rc_{31} * P_{b_5}^{max} + rc_{32} * P_{B_6} + rc_{33} * P_{B_4} - rc_{34} * P_{b_4}^{max} + \frac{rc_{35} * (rc_{36} * P_{B_4} - rc_{37} * P_{B_2})}{rc_{38} * P_{B_4} + rc_{39} - rc_{40} * P_{b_4}^{max} + rc_{41} * c_3} + rc_{42} * P_{B_3} + \frac{rc_{43} * (rc_{44} * P_{B_4} - rc_{45} * c_3 - rc_{46} * P_{G_1}^{max} + rc_{47} * P_{B_3})}{\frac{rc_{48}*P_{b_5}^{max}}{P_{B_2}} - rc_{49} * P_{B_9}} - rc_{50} * P_{b_1}^{max} + rc_{51} * P_{B_9} - [(rc_{52} * (rc_{53} * P_{B_4} - rc_{54} * P_{b_4}^{max} + rc_{55} * c_3 - rc_{56} * P_{B_9} - rc_{57} * P_{b_{17}}^{max} - rc_{58} * P_{G_1}^{max} + rc_{59} * P_{B_3}))]/P_{b_4}^{max} - rc_{60} \tag{A.2}$$

$$P_{G_4} = rc_{61} * P_{B_{11}} * P_{B_5} - rc_{62} \tag{A.3}$$

$$P_{G_5} = rc_{63} * P_{G_3}^{max} + rc_{64} * P_{B_5} - rc_{65} \tag{A.4}$$

Some observations shall be stated when regarding these equations in more detail:

P_{G_4} & P_{G_5} seem to deliver quite low values ($<< 1$), which shows that the power production of those plants is too expensive to be used. However, a value > 0 is necessary since plants - even if nearly no power is produced - provide the reference voltage and are still important to the power flow control. Within P_{G_2} & P_{G_3}, even if various variables throughout the grid are influencing the policies, especially load-variables of directly connected nodes are used with high frequency (buses $1-6$), which seems to be reasonable.

Generally, load values occur most frequently, which is clear since they give the most crucial information on needed real-power production.

Constant	Value	Constant	Value
P_{G_2}			
rc_1	0.009174748632	rc_{16}	0.008994590782620
rc_2	1.49283300	rc_{17}	6.69564385
rc_3	1.78953483388612	rc_{18}	34.23717341
rc_4	.624093771901573	rc_{19}	1.248187544
rc_5	1.27154724912624	rc_{20}	1.78953483388612
rc_6	.264211771833534	rc_{21}	1.27154724912624
rc_7	2.5832983635531	rc_{22}	0.264211771833534
rc_8	1.27154724912624	rc_{23}	0.192005552542291
rc_9	5.505926961	rc_{24}	10.0365276640446
rc_{10}	127.0386212	rc_{25}	0.00010716
rc_{11}	1.296078252	rc_{26}	2.36263171335547
rc_{12}	5.325613134	rc_{27}	0.5294977006
rc_{13}	.264211771833534	rc_{28}	1.27154724912624
rc_{14}	.1281947642	rc_{29}	0.264211771833534
rc_{15}	0.005191038225	rc_{30}	0.624093771901573
P_{G_3}			
rc_{31}	0.001856568614	rc_{46}	0.84055468176719
rc_{32}	0.005946704804	rc_{47}	5.127725118
rc_{33}	0.004250986442	rc_{48}	0.8327927662
rc_{34}	0.002232502556	rc_{49}	0.767252976260522
rc_{35}	0.002182532036	rc_{50}	0.002642510637
rc_{36}	1.94773152065585	rc_{51}	0.001674554200
rc_{37}	1.02144134490299	rc_{52}	0.002133680015
rc_{38}	5.843194563	rc_{53}	6.800371227
rc_{39}	4.50401344	rc_{54}	1.02289566351344
rc_{40}	2.045791328	rc_{55}	1.27770086526315
rc_{41}	1.27770086526315	rc_{56}	0.767252976260522
rc_{42}	0.01119142434	rc_{57}	0.821366567046215
rc_{43}	0.002182532036	rc_{58}	0.84055468176719
rc_{44}	3.895463042	rc_{59}	5.127725118
rc_{45}	1.27770086526315	rc_{60}	0.296274606521816
P_{G_4}			
rc_{61}	0.000055316726	rc_{62}	0.00072239522234

Continued on next page

Constant	Value	Constant	Value
P_{G_5}			
rc_{63}	$1.671046042 * 10^{-8}$	rc_{65}	$4.09932847184168 * 10^{-7}$
rc_{64}	$7.367354603 * 10^{-8}$		

Table A.2: Real-Valued Constants Assignment 14-Bus SVS

A.5 Results to Polynomial ARS

Table A.5 lists the best found solution to each case, i.e. weights $\overline{w}$ of the abstract rules within polynomial synthesis for each case. For the linear solution (30-bus case), one weight is necessary per rule, while for the quadratic synthesis two weights are associated to each rule (consider Equations 4.3 and 4.4). For each policy, one constant is being optimized too, entitled by $c_1 ... c_4$. The decomposition of the solution vector to the single policies has been explained in Section 4.2.7 respectively Figure 4.3.

Variable	14-Bus quadratic	30-Bus linear	57-Bus quadratic	118-Bus quadratic
w1.1	1	1	1	0,4347
w1.2	1		1	0
w2.1	1	0,7859	0,9326	0
w2.2	0		1	0
w3.1	1	1	1	0,4714
w3.2	1		0,9883	0
w4.1	0	0,4303	0,5431	0,6771
w4.2	0		0,7185	0,2149
w5.1	0,564	0,5329	0	0
w5.2	0,3851		0	0,278
w6.1	0,648	0,3474	0,6977	0,0557
w6.2	0		1	0
w7.1	0	1	0,0619	0,8411
w7.2	0,6646		0,8545	1
w8.1	0,8657	0,5312	0,2941	1
w8.2	0,4557		0,7338	0
w9.1	1	0,5624	1	0,189

Continued on next page

Variable	14-Bus quadratic	30-Bus linear	57-Bus quadratic	118-Bus quadratic
w9.2	0,6659		1	0
w10.1	0,5453	0,6094	0	0,1863
w10.2	1		0,4361	0,2982
w11.1	1	0,728	1	1
w11.2	0,4071		0,5507	0,9654
w12.1	1	0,8679	0,5722	0,0048
w12.2	0,3432		0	0
w13.1	0,7817	0	0,8435	1
w13.2	0,2475		0,418	1
w14.1	0,5243	0,9492	0,7358	0,5991
w14.2	0,8827		0,1793	0
w15.1	1	0,6566	0	1
w15.2	0		0,1269	1
w16.1	1	0	1	0,369
w16.2	0,0609		0,9806	0,975
w17.1	0,5694	0,3444	0,0576	0,4947
w17.2	0		1	0,8072
w18.1	0	0	0,0758	0,1034
w18.2	1		0,9689	0,9732
w19.1	0	0,1743	0	0
w19.2	0,7827		0,0043	0,0104
w20.1	0,4012	0,814	0,1714	0,1023
w20.2	0,7461		0,1343	0,1551
w21.1	0,9254	0,3958	0,8927	0,9034
w21.2	0,6832		0,1016	0,0012
c1	1	0	0,1532	0,1043
c2	0,1142	0,7976	0,3803	0,22
c3	0	0,1996	0,1711	0,1478
c4	0,1648	0,147	0,0007	0,1918

Table A.3: Best Found Solutions with Polynomial Synthesis of Abstract Rules

A.6 Results to GP Synthesis with ARS

14-Bus Test Case

$$P_G(\overline{r}) = rc_1 * llf * (rc_2 * pcf + rc_3 * llf - rc_4) * \\ (rc_5 * llf + rc_6 * pcf - \frac{rc_7 * nlfs}{lcf} - rc_8) \quad \text{(A.5)}$$

$$Q_C(\overline{r}) = rc_9 * mrf + \frac{rc_{10} * merf}{glf * mrf * nlfsq * (-\frac{rc_{11} * nlfs}{merf^2} + rc_{12} * glf} \\ + 2 * nlfsq - rc_{13} * nlfs \quad \text{(A.6)}$$

$$V_G(\overline{r}) = \frac{rc_{14} * nlfsq * (rc_{15} * nlfsq - rc_{16} * glfq)}{(rc_{17} * nlfsq - rc_{18} * glfq)} + rc_{19} \quad \text{(A.7)}$$

$$T(\overline{r}) = rc_{20} * nlfsq \quad \text{(A.8)}$$

Constant	**Value**	**Constant**	**Value**
rc_1	0.5023134612	rc_{11}	12.69856671
rc_2	1.9310445684	rc_{12}	0.36013402
rc_3	1.650124450238	rc_{13}	4.337768712
rc_4	0.9613345034545	rc_{14}	36.16479623
rc_5	1.569803923546	rc_{15}	1.760117633
rc_6	1.930135952	rc_{16}	0.32989912
rc_7	0.7782317101	rc_{17}	189.39932432001
rc_8	0.25994357	rc_{18}	33.65009731235
rc_9	1.278829124	rc_{19}	0.92003212
rc_{10}	0.08737413221	rc_{20}	0.6501342678

Table A.4: Real-Valued Constants Assignment 14-Bus ARS

30-Bus Test Case

$$P_G(\overline{r}) = (rc_1 * (rc_2 * merf + rc_3 * glf - rc_4 * lcf))* \\ (\frac{rc_5 * glf * (rc_6 * glf - rc_7 * lcf)}{pcf^2} + rc_8 * glf * mrf) + rc_9 \tag{A.9}$$

$$Q_C(\overline{r}) = \frac{-rc_{10} * nlfsq}{merf * glf} + rc_{11} * nlfs + rc_{12} * glf - rc_{13} * merf \\ - rc_{14} * glf * mrf * nlfsq \tag{A.10}$$

$$V_G(\overline{r}) = rc_{15} * nlfs - rc_{16} * nlfsq* \\ (rc_{17} * nlfsq + \frac{rc_{18} * glf}{-rc_{19} * glfq + rc_{20} * nlfsq}) + rc_{21} \tag{A.11}$$

$$T(\overline{r}) = \frac{rc_{22} * nlfsq}{glfq} \tag{A.12}$$

Constant	Value	Constant	Value
rc_1	0.022584391622329	rc_{12}	0.19331921668476
rc_2	0.300369793090626	rc_{13}	0.127834427886384
rc_3	1.1568988081503	rc_{14}	0.6431956875
rc_4	0.482939198125127	rc_{15}	0.0291709528742042
rc_5	3.209932224	rc_{16}	0.0432870615113212
rc_6	1.24139869653752	rc_{17}	1.67230183977234
rc_7	0.482939198125127	rc_{18}	0.814501831722179
rc_8	8.87081977769297	rc_{19}	0.757900514291291
rc_9	0.28166616774456	rc_{20}	1.46851844985904
rc_{10}	0.07700896430	rc_{21}	0.955877454221944
rc_{11}	0.238363555569058	rc_{22}	6.724411552

Table A.5: Real-Valued Constants Assignment 30-Bus ARS

57-Bus Test Case

$$P_G(\overline{r}) = -rc_1 * pcf - rc_2 * lcf + \frac{rc_3 * lcf}{glf} - rc_4 * lcf * (-rc_5 * glf + rc_6) + rc_7 \quad \text{(A.13)}$$

$$Q_C(\overline{r}) = \frac{rc_8 * glfq}{nlfs * merf} \quad \text{(A.14)}$$

$$V_G(\overline{r}) = \frac{(rc_9 * (rc_{10} * glf - rc_{11} * nlfs)) * (\frac{rc_{12}}{nlfsq - rc_{13}})}{nlfsq} + rc_{14} \quad \text{(A.15)}$$

$$T(\overline{r}) = rc_{15} * glfq \quad \text{(A.16)}$$

Constant	**Value**	**Constant**	**Value**
rc_1	0.54593114482147	rc_9	0.0002605700484
rc_2	1.12370271944248	rc_{10}	1.01521391610751
rc_3	0.3086082691	rc_{11}	0.858157571660497
rc_4	0.274738839917123	rc_{12}	2.835516782
rc_5	15.1013679394346	rc_{13}	9.06989821567111
rc_6	8.61861396367215	rc_{14}	0.90344395654115
rc_7	1.17393665385405	rc_{15}	2.7650945133127
rc_8	0.8703370953		

Table A.6: Real-Valued Constants Assignment 57-Bus ARS

118-Bus Test Case

$$P_G(\overline{r}) = -rc_1 * lcf - \frac{rc_2}{(-rc_3 * pcf - rc_4 * lcf + rc_5 * glf)} - rc_6 \quad (A.17)$$

$$Q_C(\overline{r}) = rc_7 * glfq \quad (A.18)$$

$$V_G(\overline{r}) = \frac{-rc_8 * nlfsq * (rc_9 * nlfsq + \frac{rc_{10} * glfq}{nlfs} /)}{(rc_{11} * glfq - rc_{12} * nlfsq + \frac{rc_{13} * glfq}{nlfsq} - rc_{14})} + rc_{15} \quad (A.19)$$

$$T(\overline{r}) = \frac{-rc_{16} * glf}{glfq * nlfs} \quad (A.20)$$

Constant	**Value**	**Constant**	**Value**
rc_1	0.155548842697486	rc_9	1.26833120819581
rc_2	0.1145891097	rc_{10}	0.7491700156
rc_3	1.41024219036674	rc_{11}	0.382813938306131
rc_4	0.759401256221478	rc_{12}	17.6598906339669
rc_5	0.609241551352816	rc_{13}	0.5360855740
rc_6	0.0120148594281784	rc_{14}	3.0095511328716
rc_7	1.01466994729926	rc_{15}	0.945918940336494
rc_8	0.303452661919591	rc_{16}	0.2416044647

Table A.7: Real-Valued Constants Assignment 118-Bus ARS

300-Bus Test Case

$$P_G(\bar{r}) = \frac{-rc_1 * glf^2}{lcf^4 * (-rc_2 * glf - \frac{rc_3 * pcf}{mrf}} + rc_4 \tag{A.21}$$

$$Q_C(\bar{r}) = -(rc_5 * (-rc_6 * nlfsq - rc_7 * merf - rc_8 * mrf * glfq - rc_9 * glf)) * glf * (-rc_{10} * glf + rc_{11} * nlfsq) \tag{A.22}$$

$$V_G(\bar{r}) = (rc_{12} * ((rc_{13} * nlfsq - rc_{14} * glf) * (rc_{15} * nlfs - rc_{16} * nlfsq) + rc_{17})) * (-rc_{18} * nlfs + rc_{19} * glf)^2 + rc_{20} \tag{A.23}$$

$$T(\bar{r}) = \frac{-rc_{21}}{glfq} \tag{A.24}$$

Constant	**Value**	**Constant**	**Value**
rc_1	0.06120832765	rc_{12}	0.0177062258262658
rc_2	0.770697086326539	rc_{13}	2.31463007650868
rc_3	5.845319934	rc_{14}	1.21332053072351
rc_4	0.0644296181630367	rc_{15}	0.270119269747214
rc_5	0.109923212165849	rc_{16}	2.13833923606081
rc_6	1.077264126	rc_{17}	0.835234255602339
rc_7	0.231704707004565	rc_{18}	0.252995468453396
rc_8	0.1701531960	rc_{19}	2.18512229723492
rc_9	2.88931259346142	rc_{20}	0.916811401302349
rc_{10}	0.447585038452153	rc_{21}	0.2002247196
rc_{11}	2.941347444		

Table A.8: Real-Valued Constants Assignment 300-Bus ARS

A.7 Matlab Implementation of Linear EV Charging Optimization

The definition of the optimization problem in Matlab is performed by formulating both the objective function as well as the constraints with matrices. The *linprog*-function solves a minimization of $f^T * u$ such that $A * u \leq b$ and $lb \leq u \leq ub$ (equality constraints are not needed for the defined problem).

Referring to Equations 6.4-6.6, the constraints for cr_{MAX}, p_{MAX}, and E_{MIN} are expressed by building their coefficient-matrices such that $A * u \leq b$ holds.

The lower and upper bounds for the control variables are defined as $lb = 0$ and $ub = 1$, which ensures that none of the EVs is allowed of charging with a power value of more than p_{MAX}. Additionally, Equations 6.4 and 6.5 ensure the individual as well as the joint power restriction for each single EV and the total fleet respectively. The objective function from Equation 6.7 respectively its coefficients for all time steps are stored within a vector f. Finally the optimization problem is solved with:

```
[u,fval,exitflag,output,lambda] = linprog(f,A,b,[],[],lb,ub,[],options);
```

The resulting charging power $P_{EV\,t,n}$ of EV n at time step T is obtained by multiplying the control variable $u_{t,n}$ with the respective power capacity $p_{MAX\,t}$.

A.8 Small-Scale Demonstration Grid for EV Charging

This demonstration grid has been set up for illustrative reasons and does not stem from literature nor is taken from real-world. The grid data in MATPOWER data-format are given as follows. For more information on the data structures, see MATPOWER manual.

```
%% MATPOWER Case Format : Version 2
mpc.version = '2';
% Base MVA = 10 MVA
mpc.baseMVA = 10;
% Base kV = 12.66 kV.
mpc.baseKV=12.660;
%% Bus data
mpc.bus = [
%bus_i type Pd    Qd     Gs  Bs  area Vm  Va  baseKV zone Vmax  Vmin
% ==== === ===== =====  === === ===  === === ====== ==== ==== ======
     1  3   0     0       0   0   1    1   0   12.66  1    1.05   0.95;
     2  1   0.0   0.00    0   0   1    1   0   12.66  1    1.05   0.95;
     3  1   0.09  0.04    0   0   1    1   0   12.66  1    1.05   0.95;
     4  1   0.12  0.03    0   0   1    1   0   12.66  1    1.05   0.95;
     5  2   0.1   0.08    0   0   1    1   0   12.66  1    1.05   0.95;
```

```
% ===================================================================
];
%% Generator data
mpc.gen = [
 %  bus  Pg  Qg  Qmax Qmin Vg   mBase   status  Pmax   Pmin
 %  ===  === === === ===  ===  ====    =====   ====   ====
     1   0    0   10  -10  1    10       1      10      0;
     5   0    0   10  -10  1    10       1      4       0;
 % ===================================================
];
%% Branch data
mpc.branch = [
% fbus  tbus  r         x       b   rateA rateB rateC ratio angle status
% ====  ====  ===       ===     === ===== ===== ===== ===== ===== ======
  1     2     0.0353    0.0300  0   10    10    10      0     0       1;
  2     3     0.0421    0.0257  0   10    10    10      0     0       1;
  2     4     0.0409    0.0278  0   10    10    10      0     0       1;
  4     5     0.0427    0.0301  0   10    10    10      0     0       1;
% ===================================================================
];
```

The input data to the model as well as the power flow computation are given in Matlab-code for better understanding. The code for all three simple experiments is presented below:

Experiment 1

```
%sampleCase.m contains the model data as above
casexx=eval('sampleCase');

%set Domestic Load
%column 3 of the bus-structure contains the load values
casexx.bus(3,3)=0.090;
casexx.bus(4,3)=0.091;
casexx.bus(5,3)=0.06;


%set PEV Load
casexx.bus(3,3)=casexx.bus(3,3)+0.1;
casexx.bus(5,3)=casexx.bus(5,3)+0.05;

%set Generation
%the gen-structure contains the real-power injection in column 2
casexx.gen(2,2)=0.005;
%obtain results
%runpf performs power flor calculation with Newton-Raphson solver;
```

```
%results.branch(:,14) gives the resulting active power flows
results=runpf(casexx);
results.branch(:,14)
```

Experiment 2

```
%set Domestic Load
casexx.bus(3,3)=0.065;
casexx.bus(5,3)=0.093;
casexx.bus(4,3)=0.091;

%set PEV Load
casexx.bus(3,3)=casexx.bus(3,3)+0.02;
casexx.bus(5,3)=casexx.bus(5,3)+0.13;

%set Generation
casexx.gen(2,2)=0.015;

%get PF results
results=runpf(casexx);
results.branch(:,14)
```

Experiment 3

```
%set Domestic Load
casexx.bus(3,3)=0.065;
casexx.bus(5,3)=0.093;
casexx.bus(4,3)=0.091;

%set PEV Load
casexx.bus(3,3)=casexx.bus(3,3)+0.02;
casexx.bus(5,3)=casexx.bus(5,3)+0.13;

%set Generation
casexx.gen(2,2)=0.025;

%get PF results
results=runpf(casexx);
results.branch(:,14)
```

A.9 Abstract Rules for EV Charging: Matlab Code

Some explanation: *casex* is a structure containing the MATPOWER model data, see the MATPOWER manual [120]. *casex.bus* returns a matrix with the bus data, *casex.branch* returns the bus branch data. *schedules* is a $N \times T$-dimensional matrix containing the estimated driving patterns of all EVs. *buses* is a $N \times T$-dimensional matrix containing the locations of EVs, where *buses(n,t)* gives the bus number of EV n at time step t.
tempSchedules provides all previous charging decisions during the simulation and is a $N \times T$-dimensional matrix containing the charging rates of EVs

```

%total remain Time so far of EV 'n' at time step 't'
%schedules(n,i)=1 if EV 'n' is being parket at time 't'; 0 otherwise
remainTime=sum(schedules(n,1:t));
RT=1-(remainTime/t);

%estimated remaining remain Time until the end
estRemTime=sum(schedules(n,t:T));
ERT=1-(estRemTime/T);

%estimated time to departure from actual place
estDep=0;
if(schedules(n,t)>0)
    j=t;
        while(schedules(n,j)>0)
            if(j>=T)
                break;
            end
            estDep=estDep+1;
            j=j+1;
        end
end
ETTD=1-(estDep/T);

% actual time from arrival at this place until now
estArrival=0;
if(schedules(n,t)>0)
    j=t;
        while(schedules(n,j)>0)
            estArrival=estArrival+1;
            j=j-1;
            if(j<=0)
                break;
            end
        end
```

```
end
TFA=1-(estArrival/t);

% generation & distribution metrics (past, future, actual)
% scheduleIrradiance is a 1-dim vector of length 'T' containing the
% predicted solar irradiance, normalized to [0,1]
% scheduleBaseLoad is a 1-dim vector of length 'T' containing the
% predicted base load in the grid, normalized to [0,1]
% scheduleWindSpeed is a 1-dim vector of length 'T' containing the
% predicted wind speed, normalized to [0,1]
% schedulePrices is a 1-dim vector of length 'T' containing the
% predicted stock market prices, normalized to [0,1]

%actual irradiation
AI=scheduleIrradiance(t);

% actual base load
ABL=scheduleBaseLoad(t);

%actual stock market electricity price
AP=schedulePrices(t);

%actual wind speed
AWS=scheduleWindSpeed(t);

% horizon gives the estimated time interval until departure
horizon=i+estDep-1;

% past irrediation
PI=mean(scheduleIrradiance((t-estArrival+1):t));

% estimated mean irradiation during remain time
EI=mean(scheduleIrradiance(t:(horizon)));

% past base load
PBL=mean(scheduleBaseLoad((t-estArrival+1):t));

% estimated mean base load during remain time
EBL=mean(scheduleBaseLoad(t:horizon));

% past stock market electricity price
PP=mean(schedulePrices((t-estArrival+1):t));

% estimated mean price during remain time
EP=mean(schedulePrices(t:(horizon)));

% past wind speed
PWS=mean(scheduleWindSpeed((t-estArrival+1):t));
```

```

% estimated mean wind speed during remain time
EWS=mean(scheduleWindSpeed(t:horizon));

% distance to peak load; tpeak gives the time step if peak load
DTP=abs((tpeak-t)/tpeak);


% BNR contains the location of EV 'n' at time step 't' in form of the bus
% number; is necessary for location-related rules
BNR=buses(n,t);

% mean branch rating (maximum allowed power flow) of directly connected
% branches
branches=[find(casexx.branch(:,1)==BNR);find(casexx.branch(:,2)==BNR)];
MMVA(BNR)=mean(maxbranchMVA(branches))/max(maxbranchMVA);

% nr. of vehicles actually remaining at same bus; with exemplarily 33 buses
NREVB=1-size(find(buses(:,j)==BNR),1)/(N/33);

% nr. of vehicles actually charging at last time step all buses
NREVC=1-(size(find(tempSchedules(:,t-1)),1)/N);

% nr. of vehicles actually charging at last time step at same bus
NREVCB=1-size(find(sameBus.*tempSchedules(:,t-1)),1)/N;

% mean charging rate at last time step all buses with maximum charging
% rate of 10kW
MCR=1-sum(tempSchedules(:,t-1))/(N*10);

% mean charging rate at last time step at same bus
MCRB=1-sum(sameBus.*tempSchedules(:,t-1))/NREVB*10;

% mean nr. of cars remaing during remaing time at same bus; with
% exemplarily 33 buses
MNREVB=1-(size(find(buses(:,(t-estArrival+1):t)==BNR),1)/estArrival)
      /(N/33);

%already charged energy related to battery-capacity
%stepSize is the length of a discrete time step in hours
ACE=sum(tempSchedules(n,1:t))*stepSize/batteryCapacity;
```

A.10 EV Traffic Model

The finally applied EV-traffic simulation is designed to have low execution time while still describing the relevant behavior sufficiently. Therefore, for each EV a driving pattern is simulated through randomizing "prototype"-patterns from statistical investigations. Figure A.1 illustrates this issue, where an exemplary driving pattern is shown and exemplary Gauss-curves indicate the randomization process.

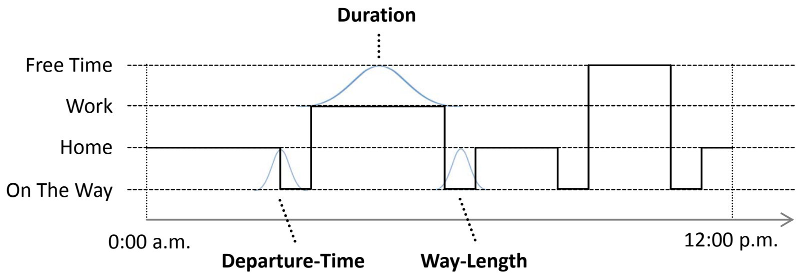

Figure A.1: Exemplary Driving Pattern

When randomizing a pattern, each element (i.e. departure times, way-lengths as well as durations) is drawn from a normal distribution with given μ and σ value. In this way, individual behavior is simulated even if EVs behave according to known patterns.

As already discussed, two distinct prototype-profiles are used herein, namely full-time and half-time employees. The respective parameters for simulating them are given in Table A.9 in ascending order according to their temporal occurrence along a day.

Element	Acronym	μ [h]	σ [h]
Departure Time Home	T_{home}	6.5	1
Way-Length	WL	0.66	0.25
Duration Work Halftime	$T_{work_{PT}}$	4.5	0.5
Duration Home Afterwork	$T_{home_{AW}}$	4.5	0.5
Duration Work Fulltime	$T_{work_{HT}}$	9	0.5
Way-Length Evening	WLE	2	0.5
Duration Free Time	T_{free}	2	0.5

Table A.9: Parameters for Driving Pattern Randomization

Most of these assumptions base on an Austrian survey on traffic data [17]. According to these parameters, the final driving pattern for each EV is sampled. The pattern for full-time employees results from sampling the elements in the sequence $[T_{home}, WL, T_{work_{FT}}, WLE]$, while for part-time employees this sequence is $[T_{home}, WL, T_{work_{PT}}, WL, T_{home_{AW}}, WL, T_{free}, WL]$. At the end of each sequence, a specific remain time at home results implicitly after the last journey.

As already given at the experiments description, at each location where an EV remains a specific probability is assumed that models the availability of a charging infrastructure. They are given as follows: $p_{home} = 1$ for parking at home, $p_{work} = 0.5$ for parking at work as well as $p_{free} = 0.25$ for locations in free time. Hence, for instance an exemplary part-time employee that visits some free-time location in the afternoon has a chance of 25% that a charging infrastructure is available there.

List of Figures

List of Tables

Curriculum Vitae

Stephan HUTTERER

Personal Data

PLACE AND DATE OF BIRTH:	4810 Gmunden, Austria December 6th, 1986
ADDRESS:	Leonfeldnerstraße 91b, 4040 Linz, Austria
EMAIL:	hutterer.stephan@gmail.com

Work Experience

Current 2012	Research Associate UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA, Campus Hagenberg *HEAL-Research Group Member*
MAI 2009	Research Associate, UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA, Campus Wels Research Interests: Smart Grids, Metaheuristic Optimization, Machine Learning, Electric Mobility, Probabilistic Modeling & Optimization
SEP 2008 - APR 2009	Engineer Robotics Software, PROMOT AUTOMATION, Roitham Internship and following employment; Software & requirements engineering in the field of PLC control for robotics.
MAR - JUN 2008	Teaching Assistant, UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA, Wels Control engineering and process visualisation.
JUL 2006 - JUN 2009	Tutor in Mathematics, at Various Organisations such as ABACUS, SCHÜLERHILFE

Education

2009 - 2013	Doctoral Studies: Technical Sciences/ Informatics, Johannes Kepler University Linz, Linz
2005 - 2009	Diploma Studies in Automation Engineering, Upper Austria University of Applied Sciences , School of Engineering/ Environmental Sciences, Wels
1997 - 2005	Grammar School, Höhere Internatsschule des Bundes, Schloss Traunsee, Altmünster

Extraordinary Achievements — Grants

2012	Research Associate of the Year, Upper Austria University of Applied Sciences , School of Engineering/ Environmental Sciences
July 2012	1st Place GECCO Industrial Data Mining Challenge, Genetic And Evolutionary Computation Conference
Academic Year 2006/2007	Achievement Grant, Upper Austria University of Applied Sciences , School of Engineering/ Environmental Sciences