

© 2018 by Mayank Baranwal. All rights reserved.

ENTROPY-BASED FRAMEWORK FOR COMBINATORIAL OPTIMIZATION
PROBLEMS
AND
ENABLING THE GRID OF THE FUTURE

BY
MAYANK BARANWAL

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Professor Srinivasa Salapaka, Chair and Director of Research
Professor R. Srikant
Associate Professor Carolyn Beck
Assistant Professor Mohammed Ali Belabbas
Assistant Professor Subhonmesh Bose
Assistant Professor Karthekeyan Chandrasekaran

Abstract

This thesis is divided into two parts. In the first part, I describe efficient meta-heuristic algorithms for a series of combinatorially complex optimization problems, while the second part is concerned with robust and scalable control architecture for a network of paralleled converter/inverter systems (DC/AC microgrids).

Combinatorial optimization problems arise in many applications in various forms in seemingly unrelated areas such as data compression, pattern recognition, image segmentation, resource allocation, routing, and scheduling, graph aggregation, and graph partition problems. These optimization problems are characterized by a combinatorial number of configurations, where a cost value can be assigned to each configuration, and the goal is to find the configuration that minimizes the cost. Moreover, these optimization problems are largely non-convex, computationally complex and suffer from multiple local minima that riddle the cost surface. Most heuristics to these optimization problems are very sensitive to initial guess solutions, and efforts to make them robust to initializations typically come at significant computational costs such that the algorithms lose practicality in many applications. In our work, we are motivated by solutions that are employed by nature to similar combinatorial optimization problems; well described in terms of laws such as maximum entropy principle (MEP) in statistical physics literature. We propose to use MEP in solving a variety of combinatorial optimization problems.

Our main current contributions are threefold - (i) First we provide a clustering or resource allocation viewpoint to several combinatorial optimization problems: (a) data clustering, (b) graph partitioning (such as clustering of power networks), (c) traveling salesman problem (TSP) and its variants, and (d) hard problems on graphs, such as multiway k -cut. This viewpoint enables a unified approach to handle a broad class of problems, and therefore efficient MEP based heuristics can be leveraged to obtain high-quality solutions. (ii) Second, we explore MEP based ideas to clustering problems specified by pairwise distances. Many problems in graph theory are indeed specified in terms of the corresponding edge-weight matrices (and not in terms of the nodal locational coordinates). (iii) Finally, our framework allows for inclusion of several constraints in the clustering/resource allocation problems. These constraints may correspond to capacity constraints in case of resource allocations where capacity of each resource is limited, or minimum-tour

length constraints in case of traveling salesman problems (TSPs) and its variants.

In the second part of this thesis, we describe a novel distributed, robust and optimal control architecture for both DC as well as AC microgrids. Microgrids are grid systems that allow integration of local power sources, such as photovoltaics (PVs), wind, battery and other distributed energy resources (DERs) with local loads connected at the DC-link or the point of common coupling (PCC). Microgrids are hypothesized as viable alternatives to the traditional electric grid. In a microgrid, the main goals of the control design are to regulate voltage and frequency at the PCC and ensuring a prescribed sharing of power among different sources; for instance, economic considerations can dictate that power provided by the sources should be in a certain proportion or according to a prescribed priority. The main challenges arise from the uncertainties in the size and the schedules of loads, the complexity of a coupled multi-converter network, the uncertainties in the model parameters at each converter, and the adverse effects of interfacing DC power sources with AC loads, such as the 120Hz ripple that must be provided by the DC sources. A systematic control design that addresses all the challenges and objectives for the multi-converter/inverter control is still lacking in the existing literature. The main contribution of the control architecture proposed by us is its capability to addresses all the primary objectives - a) voltage and frequency regulation at the PCC with guaranteed robustness margins, b) prescribed time-varying power sharing in a network of parallel converters, c) controlling the tradeoff between 120Hz ripple on the total current provided by the power sources and the ripple on the DC-link voltage. An important contribution of our work is that our control architecture allows for closed-loop analysis and robust control synthesis for the entire grid network. We introduce a structure in the control architecture, whereby, we show that analysis of the entire multi-component microgrid can be simplified to that of an equivalent *single-component* system. Besides analysis, this simplification facilitates using robust and optimal control tools for achieving multiple objectives simultaneously; in contrast in existing architectures, closed-loop analysis for entire networks is typically difficult, and posing optimal control and robustness objectives for the entire network practically untenable.

To my family

Acknowledgments

First, I would like to sincerely thank my adviser Prof. Srinivasa Salapaka for his patience and guidance. Apart from the knowledge he has been imparting in me through various useful discussions on varied topics ranging all the way from research to the current state of politics, I always admire his humility and positive attitude towards the students. A student is a reflection of his/her mentor, and I sincerely hope that I am able to imbibe his qualities not just at the professional level, but also on the personal level.

I would also like to express my gratitude to my esteemed committee members - Dr. Carolyn Beck, Dr. Ali Belabbas, Dr. Subhonmesh Bose, Dr. Karthekeyan Chandrasekaran, and Dr. R. Srikant for agreeing to serve on my dissertation committee and providing useful insights that helped me to develop a well-grounded, holistic approach to my doctoral work. I would also like to thank my collaborators - Dr. Subhonmesh Bose, Dr. Murti Salapaka, Dr. Alexander Schwing, and Dr. Yurii Vlasov for providing me opportunities to explore interesting new avenues.

It would only be fair to acknowledge the love and support of my research colleagues - Alireza Askarian, Ram Sai Gorugantu, Sheikh Mashrafi, Amber Srivastava, and Sreenath Sundar. I also enjoyed my interactions with fellow CSL folks, particularly Thinh Doan, Shripad Gade, Jin Kim, Siddhardh Nadendla, Philip Pare, and Amirhossein Taghvaei.

All through my stay at UIUC, my family has always supported me through their love and encouragement. It's only fair to thank Almighty for having given me a wonderful family to whom I could always look up to for advice and encouragement.

I would like to extend my deepest regards to some of the greatest teachers and researchers at UIUC for instilling confidence and motivation in me to keep striving for excellence.

Graduate life in Urbana-Champaign has been a fun-filled journey and I owe this to my friends Shikha Aggarwal, Shashank Agrawal, Ravi Bhaduria, Ankita Bhutani, Pranav Garg, Dileep Kini, Vivek Kumar, Shikhar Mohan, Shishir Pandya, Advitya Patyal, Komal Singh, Piyush Singh, Manila Sarangi, Srikanthan Sridharan, Prakalp Srivastava, and Ankur Taneja.

Table of Contents

List of Tables	viii
List of Figures	ix
List of Abbreviations	xii
Part I, Combinatorial Optimization Problems: Maximum Entropy Principle Approach	1
Chapter 1 Introduction: Combinatorial Optimization Problems	2
1.1 Historical Overview	2
1.2 Main Contributions	3
1.3 Organization	5
Chapter 2 Maximum Entropy Principle: Overview	6
2.1 Deterministic Annealing (DA) Algorithm	7
Chapter 3 Applications of Maximum Entropy Principle	11
3.1 Traveling Salesman Problems and its Variants	11
3.1.1 TSP: Problem Formulation and Approach	12
3.1.2 Non-Returning mTSP: Problem Formulation and Approach	15
3.1.3 Returning mTSP: Problem Formulation and Approach	18
3.1.4 Single Depot mTSP: Problem Formulation and Approach	20
3.1.5 CETSP: Problem Formulation and Approach	21
3.1.6 Controlling the Lagrange Multiplier	23
3.1.7 Experimental Evaluation	25
3.2 Clustering of Power Networks	28
3.2.1 Quantification of Electrical Distance	30
3.2.2 Graph Clustering and Determining Zones in Electrical Network	32
3.2.3 Rule-Based Supervisory Voltage Control	35
3.2.4 Evaluation on IEEE Test Systems	37
3.2.5 Effect of perturbations at buses within the same cluster over the remainder of the network	39
3.2.6 Rule-based supervisory voltage control	41
3.3 Clustering Nonlinearly Separable Data	45
3.3.1 Motivation and Related Work	45
3.3.2 The Weighted-Kernel Deterministic Annealing (WKDA) Algorithm	47
3.3.3 Connection with Kernel k -Means and Spectral Clustering Algorithms	48
3.3.4 Experimental Evaluations of WKDA	50
3.4 Hard Problems on Graphs: Multiway k -Cut	53
3.4.1 Problem Description and Formulation	53

3.4.2	Convergence Analysis of the Proposed Algorithm	57
3.4.3	Illustrative Examples	60
Chapter 4	Conclusions and Future Directions	64
Part II, Grid and Microgrid of the Future: Robust Control Framework		67
Chapter 5	Introduction: Grid and Microgrid of the Future	68
5.1	Historical Overview	68
5.2	Main Contributions	71
5.3	Organization	72
Chapter 6	Facilitating Linear Models of Power Electronic Converters	73
6.1	Modeling of DC-DC Converters	73
6.1.1	Boost converter dynamics	74
6.1.2	Buck converter dynamics	75
6.1.3	Buck-boost converter dynamics	75
6.2	Modeling of DC-AC Inverter	76
Chapter 7	Robust Control Framework for Power Electronic Converters	78
7.1	Control of DC-DC Converters	78
7.1.1	Control of Single Converter	78
7.1.2	Extension to Multi-Converters System	83
7.2	Control of DC-AC Inverters	85
7.2.1	Control of Single Inverter	85
7.2.2	Extension to Multi-Inverters System	89
Chapter 8	Case Studies: Simulations and Discussions	92
8.1	DC-DC Converters	92
8.2	DC-AC Inverters	95
Chapter 9	Experimental Validation	98
Chapter 10	Conclusions and Future Directions	102
Appendix	Proofs of Theorem 1 & Theorem 2	104
References		107

List of Tables

3.1	Clustering Results For IEEE-14 Bus Data	39
3.2	Clustering Results For IEEE-30 Bus Data	40
3.3	Ranges and steps of control measures	41
3.4	Supervisory Control for IEEE-14 Bus Network	42
3.5	Control actions for 14-bus system	43
3.6	Supervisory Control for IEEE-30 Bus Network	43
3.7	Control actions for 30-bus system	44
3.8	Controlled islanding for IEEE-14 Bus Network	44
3.9	Final association matrix for the 4-cut problem shown in Fig. 3.16b	62
6.1	Generalized models for multiple converter topologies	76

List of Figures

3.1	Schematic of a (a) 9-nodes non-returning 3TSP. The dashed blue lines indicate the removal of links, which corresponds to $\alpha_{k_1=3} = 1, \alpha_{k_2=7} = 1$ (b) 9-nodes returning 3TSP. The dashed blue lines indicate the removal of links ($-d_{\sigma_{k_j}, \sigma_{k_{j+1}}}$), while solid blue lines indicate the addition of links ($d_{\sigma_{k_j}, \sigma_{k_{(j-1)+1}}}$). (c) 9-nodes single-depot returning 3TSP. (d) Single salesman returning CETSP. Each node x_i is provided with a radius parameter ρ_i . The orange dots indicate y_j such that $v_{ij} = 1$ for some node i .	12
3.2	Pictorial representation of the working methodology of the DA algorithm for routing. Note that the desired number of resources (depicted by triangles) are same as the number of cities (depicted by squares). (a) At very low values of β , each city is uniformly associated with all the resources - all the resources are located at the centroid of the customer locations. (b) At an intermediate value of β , the <i>fuzziness</i> in the associations decreases and as a result resources start developing affinity to unique customer locations. (c) When $\beta \rightarrow \infty$, the resource associations become <i>hard</i> resulting in hard-clustered solutions as desired. At each value of β , the resources are constrained to maintain tension in the loop connecting them.	15
3.3	Schematic of a (a) Non-returning 2TSP, with $\mathcal{R} = k$. (b) Returning 2TSP, with $\mathcal{R} = \{k, l\}$. (c) Returning 2TSP (with Depot), with $\mathcal{R} = k$. The dashed blue lines indicate the removal of links, while solid blue lines indicate the addition of links.	16
3.4	(a) Result for non-returning 2TSP for 59 nodes data. The dashed line indicates the removal of the link. (b) Result for returning 2TSP for 59 nodes data. The dashed lines indicate the removal of links, while the black solid lines indicate the addition of links. (c) Returning 2TSP version for cocentric rings.	25
3.5	(a) Single-depot returning 2TSP solution to a 59 nodes data. The depot location is denoted by <i>red</i> marker. (b) Solution to the same 59 nodes data with equality constraint. Note that the optimal length is only slightly bigger than the earlier case.	26
3.6	Comparison between MEP based deterministic annealing (DA) approach and simulated annealing (SA) approach for - (a) the usual traveling salesman problem (TSP), (b) non-returning 2TSP (NR2TSP), and (c) 2TSP returning to depot.	27
3.7	CETSP result for single salesman 10 nodes returning TSP. The <i>red</i> markers denote the node locations, while the <i>black</i> 'x' denote the facility locations. The <i>cyan</i> circles correspond to the radii ρ_i .	27
3.8	An example of a four-bus network. Bus 1 is a <i>Slack</i> bus whose voltage and angle are specified. Bus 2 is a <i>Generator</i> bus (<i>PV</i> bus) whose real output power P and commanded output voltage V are specified. Generators are set to regulate their own bus to a commanded voltage (by adjusting reactive power). Bus 3 is a <i>connecting</i> bus with zero load. The load power is known, and this is a <i>PQ</i> bus. Bus 4 is a <i>load</i> bus (<i>PQ</i> bus) with per unit (p.u.) active and reactive powers $P = 0 : 2; Q = 0 : 4$ both given. The buses are connected by line impedances Z_{ij} . The impedance matrix is denoted by \mathbf{Z}_{bus} . The inverse matrix $\mathbf{Y}_{\text{bus}} := \mathbf{Z}_{\text{bus}}^{-1}$ denotes the admittance matrix.	30
3.9	Network Configuration of (a) IEEE-14 bus, (b) IEEE-30 bus	38
3.10	Results of k -means clustering algorithm on (a) linearly separable input data, and (b) nonlinearly separable input data.	46
3.11	Implementation of the proposed WKDA algorithm on some interesting data sets. The data sets in examples (a), (b) and (c) are obtained from https://cs.joensuu.fi/sipu/datasets/ under the shape sets category. The example presented in (d) is an artificially generated data set, whereas (e) presents the constrained version of (d).	52

3.12 (a) Schematic of a minimum multiway k -cut problem with $k = 3$ and $S = \{s_1, s_2, s_3\}$. Here A_1, A_2 and A_3 are the disconnected components. (b) Illustrative example of a multiway 3-cut problem with $S = \{1, 6, 10\}$ described in Sec. 3.4.3.	54
3.13 Empirical computational complexity of the proposed algorithm for 3-cut problem on randomly generated graphs with varying number of nodes. (a) Average number of iterations per β value, (b) Computational complexity is cubic in the number of vertices.	60
3.14 Illustrative example of minimum 3-cut problem. The columns indicate the association matrices $p(i \in A_j)$ at different β values. The ‘bold’ numbers indicate the associations of terminals, i.e., $p(1 \in A_1) = 1, p(6 \in A_2) = 1$ and $p(10 \in A_3) = 1$. As the algorithm progresses, the associations of remaining nodes harden, thereby resulting in optimal cut.	60
3.15 Illustration of the proposed multiway k -cut approach to background-foreground segmentation. Here each pixel in the bounding box represents a node. The terminal nodes (corresponding to foreground and background) are interactively provided by the user once, and the algorithm successively evaluates minimum st -cut to refine foreground-background segmentation. Edge-weights are obtained as functions of pixels’ RGB values.	61
3.16 (a) Example of a 22 nodes non-planar graph with 6 terminals, $S = \{1, 4, 8, 11, 15, 19\}$. The proposed algorithm results in an optimal cut. (b) An 8 node graph with multiple permissible optimal 4-cuts. (c) Toy-example with $2k = 6$ vertices where k of the vertices form a cycle where each edge weight is equal to 1, and each other vertex is connected to exactly one vertex on the cycle with an edge weight of 1.98. The isolating cut heuristic [1] results in a cut value of 3.96, whereas our MEP-based algorithm results in optimal cut.	62
5.1 Energy consumption history and projection. (Source: U.S. Energy Information Administration, International Energy Outlook 2013.)	69
5.2 A schematic of a microgrid. An array of DC sources provide power for AC loads. Power sources provide power at DC-link, their common output bus, at a voltage that is regulated to a set-point. The control system at the respective DC-DC converter that interfaces with a source is responsible for regulating the voltage at the DC-link.	70
6.1 Circuit representing (a) Boost converter, (b) Buck converter, and (c) Buck-Boost converter. Note that i_{load} includes both the nominal load current, as well as ripple current. The converters are assumed to operate in continuous-conduction-mode (CCM). Boost converters step up the voltage at the output, while buck converters step down the voltage. A buck-boost converter can achieve both the objectives.	73
6.2 Block diagram representation of a boost-type converter. The control signal \hat{u} is converted to an equivalent PWM signal to command the gate of the transistor acting as a switch.	75
6.3 (a) Circuit representing a full-bridge inverter. The AC-side current is given by i_{load} . The switches s_1, s_2, s_3 and s_4 control the AC-side voltage $V(t)$. (b) Associated control model of the full-bridge inverter.	77
7.1 Block diagram representation of the inner-outer control design. The <i>regulated variables</i> z_1, z_2, z_3 and z_4 correspond to weighted - (a) tracking error in DC-link voltage, (b) mismatch between i_{ref} and i_{load} , (c) control effort \hat{u} , and (d) output voltage tracking, respectively.	79
7.2 Existence of 120 Hz ripple in interfacing DC-link with AC-side. The AC-side power pulsates at 120 Hz and needs to be accounted for by the DC sources.	80
7.3 Bode magnitude plots of the closed-loop plant \tilde{G}_c for various ζ_1 values. $\tilde{\omega}$ is chosen to be 600π rad/s. Note that a relatively larger value of $\tilde{\omega}$ is in accordance with choosing a fast inner-current controller.	80
7.4 Many-converters system with shaped inner plants \tilde{G}_c . In the proposed implementation, we adopt the same outer controller for different converters, i.e., $K_{v_1} = \dots = K_{v_m} = \frac{1}{m}K_v$ and $K_{r_1} = \dots = K_{r_m} = K_r$; γ_k represents the proportion of power demanded from the k^{th} source.	83
7.5 Inner-outer control architecture for a full-bridge inverter.	86
7.6 Proposed decentralized framework with <i>smart</i> choices of controller parameters. The inner-current controllers K_{c_k} are chosen such that the inner-shaped plant mimics $\tilde{G}_c(s)$. The outer-loop controllers are scalar multiples of the nominal outer-loop controller K_v ; the scalars γ_k govern the power-sharing requirements in a microgrid setup.	89

8.1	A parallel network comprising of a PV, a Li-ion battery and two generic sources. The operating voltages and associated converter parameters are described above. It is desired to regulate the DC-link voltage to 250V. The PV module is operated using MPPT algorithm [2]. Its output current, i_{PV} , is directly proportional to the (time-varying) irradiance and is included in our proposed formulation by regarding i_{PV} as part of the disturbance signal, with the net disturbance current modeled as $i_{load} - i_{PV}$. The DC-link can additionally be used to power complex AC loads via a DC-AC inverter. The inverter is tied to utility grid. Any control design must facilitate seamless integration of DC microgrid with the AC grid.	93
8.2	Simulation results representing centralized and decentralized control implementation for voltage regulation and time-varying power sharing.	94
8.3	Simulation results representing handling of complex AC loads.	94
8.4	(a) Voltage regulation through a network of three parallel inverters with heterogeneous power sources. (b) Power sharing among multiple inverters.	96
9.1	Experimental setup with (1) custom-designed boost-converter boards, (2) controllers implemented on TMS320F28335 Delfino MCUs, (3) variable load - two resistors, each of value 50Ω , (4) DC-sources with maximum rated output voltage of 30V, (5) PV simulator subjected to simulated noisy ramp profile with a peak power of 43W and controlled using MPPT algorithm , and (6) relay for load.	99
9.2	Experimental results demonstrating effectiveness of the proposed control design under perfectly decentralized implementation for several test scenarios: 1:1:1 sharing (PV off) - (a) and (b); 2:1:1 sharing (PV off) - (c) and (d); 1:1:1 sharing (PV on) - (e) and (f); Equal sharing in presence of abrupt failure in power generation - (g) and (h). Colors blue, red, green and purple indicate power outputs of DC sources 1, 2, 3 and PV emulator, respectively.	100

List of Abbreviations

AC	Alternating Current
APX	Approximable
CETSP	Close-Enough Traveling Salesman Problem
CVRPTW	Capacitated Vehicle Routing Problem with Time-Windows
DA	Deterministic Annealing
DC	Direct Current
DER	Distributed Energy Resource
DG	Distributed Generation
EV	Electric Vehicle
FLP	Facility Location Problem
GA	Genetic Algorithm
GM	Gain Margin
HV	High Voltage
IGBT	Insulated Gate Bipolar Transistor
LMI	Linear Matrix Inequality
LV	Low Voltage
MEP	Maximum Entropy Principle
MPPT	Maximum Power Point Tracking
mTSP	multiple Traveling Salesmen Problem
NP	Non-deterministic Polynomial-time
PCC	Point of Common
PID	Proportional Integral Derivative
PM	Phase Margin
PMU	Phase Measurement Unit
PSD	Positive Semi-Definite

PSO	Particle Swarm Optimization
p.u.	per unit
PV	Photo-Voltaic
SA	Simulated Annealing
$\text{sgn}(\cdot)$	Signum function
TSP	Traveling Salesman Problem
TV	Tertiary Voltage
VRPTW	Vehicle Routing Problem with Time-Windows
WKDA	Weighted Kernel Deterministic Annealing

Part I

Combinatorial Optimization Problems: Maximum Entropy Principle Approach

Chapter 1

Introduction: Combinatorial Optimization Problems

1.1 Historical Overview

Optimization problems are ubiquitous in science and engineering, and even in our daily life, for instance, in how we optimize our budget over different kinds of expenses, in how we allocate our time to various activities, or how we allocate our resources for future investments. A subset of these optimization problems, also referred as combinatorial optimization problems, deal with finding an optimal object from a *finite* set of objects. Combinatorial optimization problems operate on the domain of those optimization problems, in which the set of feasible solutions is discrete or can be reduced to discrete, and the aim is to find the best such solution. An example of combinatorial optimization problems is the *knapsack* problem, in which we are given a finite set of objects with non-negative values and weights and a knapsack of finite capacity. The goal is to choose items to be put in the knapsack such that the sum total capacity of all the items inside the knapsack does not exceed the capacity of the knapsack, and the total value of the items inside the knapsack is maximized. Note that if there are N items in a knapsack problem, there are 2^N ways in which a knapsack can be filled, i.e., there are combinatorial number of ways to arrange N items in a knapsack. Each such way defines a total value in the sack. Evaluating all possible ways to arrange N items in a knapsack is NP-hard, as is often the case with most combinatorial optimization problems, where exhaustive search is not feasible. In this thesis, we consider a more generalized view of combinatorial optimization problems, where we also encompass mixed-variable (both continuous and discrete variables) optimization problems, such as a facility location problem (FLP).

Combinatorial optimization problems find their relevance in many applications in various forms in seemingly unrelated areas such as data compression [3], pattern recognition [4], image segmentation [5], resource allocation [6], routing [7], and scheduling [8], graph aggregation [9], and graph partition problems [10]. While problems, such as, shortest paths, flows and circulations, spanning trees, matching, and matroid problems admit polynomial-time algorithms for finding an optimal solution, there are still a large variety of combinatorial optimization problems for which are provably non-solvable in polynomial time,

unless P=NP. These optimization problems are largely non convex, computationally complex and suffer from multiple local minima that riddle the cost surface [11]. There is a vast literature on commonly used approaches to efficiently recover high-quality solutions of combinatorial optimization problems. These approaches range from exact methods, such as, branch-and-bound [12] and branch-and-cut [13], which work well in practice, however, may require exponential number of evaluations depending on the problem instance, to inexact methods (heuristics). These heuristics include repeated optimization with random initialization (such as Lloyd’s algorithm [14]), single solution metaheuristics such as simulated annealing (SA) [15], and population-based approaches such as particle swarm optimization (PSO) [16] and genetic algorithms (GA) [17]. However, the algorithms resulting from these heuristics are very sensitive to initial guess solutions, and efforts to make them robust to initializations typically come at significant computational costs such that the algorithms lose practicality in many applications.

Our approach to solving combinatorial optimization problems is motivated by solutions that are employed by nature to similar problems; well described in terms of laws such as minimum free energy principle in statistical physics literature. E.T Jaynes [18] was a pioneer in this direction and made close analogies between information theory and statistical mechanics. In fact he expounded the maximum entropy principle (MEP), which is analogous to the minimum free-energy principle. In our work, we propose to use MEP in solving the underlying combinatorial optimization problems. The MEP based approach has two significant advantages: (a) the approach is not biased to a starting solution, in fact the approach does not start with a feasible solution, but rather assumes a uniform distribution over all possible solutions and converges to a feasible (favored) solution in the limiting case. (b) the algorithm makes very few computations during its exploration to converge to final sub-optimal solutions.

1.2 Main Contributions

The first part of this thesis develops a unifying framework for addressing a wide range of discrete or combinatorial optimization problems. The work exploits tools, such as, maximum entropy principle/ minimum free-energy principle and annealing, from the statistical physics literature [18] to convert a discrete optimization problem into continuous optimization problem. By appropriately modifying the Lagrange multiplier, the continuous variables are forced to mimic discrete (binary) variables in the limiting case. Our main contributions can be summarized as follows:

- (a) *Development of a unifying framework:* Combinatorial problems arise in several contexts, such as, data clustering, scheduling, vehicle routing, minimum cuts, unit commitment, etc. and are often NP-hard.

While there are some efficient heuristics for solving problems belonging to different domains, there is a lack of common viewing point among them. The absence of such a unifying framework limits the advantages a heuristic leverages in a specific problem to generalize to other problems. In this work, we have developed a common resource allocation viewpoint to a wide range of combinatorial optimization problems. This unification enables problems from different domains to be handled using the proposed approach.

For instance, this thesis presents a unifying resource allocation viewpoint to a classical traveling salesman problem (TSP) and its variants. These variants comprise of multiple traveling salesman problem (mTSP) with varying constraints, such as, mTSP with single depot, non-returning mTSP and close-enough traveling salesman problem (CETSP). Note that a solution to mTSP comprises of several routes, one for each salesman, and the optimal tour would be the set of routes such that the total distance traveled is minimized. On the other hand, a solution to CETSP consists of a route containing each city, where it suffices to reach within a certain radius of each city in the route. CETSPs have additional computational complexity as they consider a discrete optimization problem over continuous variables (points-of-visit). All the above variants of TSP are addressed in a resource allocation framework, where a resource is allocated for each city and the constrained optimization is performed over these resource locations. Thus, our approach provides a way to view discrete optimization problems as equivalent continuous optimization problems which are then addressed using tools from statistical physics, such as maximum entropy principle (MEP).

- (b) *Convergence proof:* The MEP based approach suggested in this work is shown to converge to a minimum. In fact, it is shown that the update rule in our algorithm boils down to gradient-descent on the *free-energy* function.
- (c) *Incorporating physics of the underlying problem:* Combinatorial optimization problems, such as, clustering of power networks should incorporate physics of electrical networks based on Kirchoff's laws for efficient partitioning. Most existing approaches on partitioning power networks completely ignore the physics of the underlying problem. We have proposed a novel metric for incorporating physics of the concerned problem.
- (d) *Statistical physics meets machine learning:* The original DA algorithm proposed by Rose [19, 3] fails to cluster points that are non-linearly separable in the input space. We overcome this limitation by incorporating *kernel-trick* [20] from the machine learning literature. We call the modified algorithm the weighed kernel deterministic annealing (WKDA) algorithm. In fact, several spectral clustering

algorithms, such as, normalized-cut [5], or ratio-cut [21] can be viewed as special cases of the WKDA objective function.

- (e) *Generalization to other problems:* As part of the ongoing and future work, we plan to extend the proposed approach to more general problem of mixed-integer optimization. Our initial evaluations suggest a great scope of the proposed MEP based approach.

1.3 Organization

The organization of the first part of the thesis is as follows. In chapter 2, we describe the maximum entropy principle (MEP) in the context of combinatorial optimization problems. In particular, we describe the deterministic annealing (DA) algorithm [3, 19] that was developed primarily to address a clustering or facility location problem. We then discuss the extension of the MEP idea to a wide range of discrete and combinatorial optimization problems - (a) multiple traveling salesman problem and its variants, (b) clustering of large power networks into loosely coupled smaller networks, (c) clustering with pairwise distances using kernel trick, and (d) multiway k -cut problem. Many theoretical and practical tools are developed in the treatment of each of these problems. A variety of experiments are considered and MEP based methods exhibit state-of-the-art results. We finally conclude our discussion with interesting directions to ongoing and future work.

Chapter 2

Maximum Entropy Principle: Overview

Maximum entropy principle (MEP) was first expounded by E. T. Jaynes [18] in 1957 as a natural correspondence between *statistical mechanics* and *information theory*. MEP states that of all the probability distributions that satisfy a given set of constraints on the expected value of a function of a random variable, the distribution with the largest entropy is the best representative of the current state of knowledge (constraints).

Let $X \in \mathcal{X}$ be a random variable and f be a function of that random variable. Let $P(X)$ be any probability distribution on X . Moreover, the expected value of $f(X)$ is known to be $f_0 \in \mathbb{R}$, i.e.,

$$\langle f(X) \rangle = \mathbb{E}[f(X)] = f_0. \quad (2.0.1)$$

There are possibly infinitely many probability distributions that satisfy (2.0.1). However according to MEP, the *fairest* distribution that satisfies (2.0.1) is the one that is obtained by maximizing entropy. The idea behind MEP is that since entropy is considered as *the* measure of randomness, by choosing another distribution one implicitly makes extra restrictive assumptions.

MEP considers the following optimization problem,

$$\max_{\{P(X)\}} \underbrace{- \sum_X P(X) \log P(X)}_{H(P(X))}, \quad \text{such that,} \quad \sum_X P(X)f(X) = f_0, \quad (2.0.2)$$

where $H(P(X))$ is Shannon-entropy term [22]. The probability distribution that maximizes the Shannon entropy term in (2.0.2) is equivalently obtained by considering the following optimization problem:

$$\min_{\{P(X)\}} \underbrace{\sum_X P(X)f(X) + \frac{1}{\beta} \sum_X P(X) \log P(X)}_{L(P(X))}, \quad (2.0.3)$$

where $L(P(X))$ is the associated Lagrangian and β is the Lagrange multiplier. Note that $\sum_X P(X) = 1$. The

probability distribution $\{P(X)\}$ that solves (2.0.3) is given by:

$$P(X) = \frac{e^{-\beta f(X)}}{\sum_{X'} e^{-\beta f(X')}}. \quad (2.0.4)$$

The probability distribution in (2.0.4) is also referred as *Gibbs* distribution. Note that at small values of β , i.e., $\beta \approx 0$, $P(X) = \frac{1}{|\mathcal{X}|}$, i.e., the optimizing distribution is uniform (completely random). This makes sense since optimizing the Lagrangian at small values of β is equivalent to maximizing the Shannon entropy, and the Shannon entropy is maximized when the distribution is random. On the other hand, as $\beta \rightarrow \infty$, the optimizing distribution becomes hard (0-1). This forms the basis for using MEP for solving combinatorial optimization problems.

Thus in our approach to combinatorial optimization problems, we aim to optimize the entropy constrained objective function at successively increased values of β . Maximization of entropy is a convex problem and there is a unique deterministic global optimum (i.e., uniform distribution) to this problem. Thus β defines homotopy between maximization of entropy and minimization of original (non-convex) objective function. This process is referred as *annealing* in the statistical physics literature [18]. As β is gradually increased, the aim is to keep track of optimizers at all β such that in the limiting case ($\beta \rightarrow \infty$) the approach recovers high-quality hard solutions. Rose [19] proposed a similar MEP-based algorithm, known as deterministic annealing (DA) algorithm, in the context of facility location problem (FLP). Below we describe the DA algorithm, which forms the basis of our approach to solving a variety of combinatorial optimization problems, in detail. Note that the MEP is easily generalizable to multi-variable optimization of the form:

$$\min_{\{X_i\}} f(X_1, X_2, \dots, X_n),$$

where a probability distribution of the form $P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i)$ is considered and application of MEP results in following Gibbs distribution:

$$P(X_i) = \frac{e^{-\beta f(X_i)}}{\sum_{X'_i} e^{-\beta f(X'_i)}} \quad \text{for all } i \in \{1, 2, \dots, n\}.$$

2.1 Deterministic Annealing (DA) Algorithm

The MEP based algorithm proposed in this thesis is an extension of the deterministic annealing algorithm [19, 3], which concerns with the facility location problem (FLP). Since the DA algorithm forms the basis for

the work discussed in this thesis, it is worthwhile to describe to present the DA algorithm in detail so as to gain a lot of intuitions for our proposed framework. In an FLP, we seek optimal placement of facilities to minimize the transportation costs from a given set of points to their nearest facilities. More precisely, given a set of $N \in \mathbb{N}$ points $\mathcal{X} = \{\mathbf{x}_i : \mathbf{x}_i \in \mathbb{R}^M, 1 \leq i \leq N\}$, the objective of an FLP is to find optimal locations of $k \in \mathbb{N}$ facilities denoted by $\mathcal{Y} = \{\mathbf{y}_j : \mathbf{y}_j \in \mathbb{R}^M, 1 \leq j \leq k\}$ such that the *aggregate weighted sum of distances of each point from its nearest facility location is minimized*. If p_i denotes the relative significance of point \mathbf{x}_i , then an FLP considers the following objective:

$$\min_{\substack{\mathcal{Y} = \{\mathbf{y}_j\} \\ \mathcal{T} = \{\chi_{ij}\}}} \underbrace{\sum_{j=1}^k \sum_{i=1}^N \chi_{ij} p_i d(\mathbf{x}_i, \mathbf{y}_j)}_{D(\mathcal{X}, \mathcal{Y})}, \quad (2.1.1)$$

where $\mathcal{T} = \{\chi_{ij} : \chi_{ij} \in \{0, 1\}\}$ is a set of associations with $\chi_{ij} = 1$ if facility \mathbf{y}_j is allocated to point \mathbf{x}_i , otherwise $\chi_{ij} = 0$, and $d(\mathbf{x}_i, \mathbf{y}_j) = \|\mathbf{x}_i - \mathbf{y}_j\|^2$. Borrowing from data compression literature [22], the quantity $D(\mathcal{X}, \mathcal{Y})$ in (2.1.1) is often referred as *distortion* between set of data points \mathcal{X} and facility locations \mathcal{Y} . Then the equivalent optimization problem is to minimize the distortion function. Solution to an FLP, for squared-Euclidean distance functions, results in a set of clusters, where facility j is located at the centroid \mathbf{y}_j of the j^{th} cluster, and each data point is associated only to its nearest facility (*Voronoi* partitions).

Most algorithms for FLP (such as Lloyd's [14]) start with some initial distribution of facility locations \mathcal{Y} and iteratively optimize over them as the algorithm proceeds. However, such approaches are sensitive to the choice of initial facility locations, primarily due to the distributed aspect of the FLPs, where any change in the location of \mathbf{x}_i affects $d(\mathbf{x}_i, \mathbf{y}_j)$ only with respect to the *nearest* facility located at \mathbf{y}_j . The DA algorithm suggested by Rose [3], overcomes this sensitivity by allowing *fuzzy* association of every data point to each facility through an association probabilities $\{p(j|i)\}$. This results in a modified *distortion* measure to reflect the weighted average distance of data points to *all* the facilities:

$$\bar{D}(\mathcal{X}, \mathcal{Y}) = \sum_{i=1}^N p_i \sum_{j=1}^k p(j|i) d(\mathbf{x}_i, \mathbf{y}_j). \quad (2.1.2)$$

The probability distribution $\{p(j|i)\}$ assesses the trade-off between decreasing the *local* influence and the deviation of the modified distortion \bar{D} from the original distortion measure D . The uncertainty in associating facility locations $\mathcal{Y} = \{\mathbf{y}_j\}$ to locations of data points $\mathcal{X} = \{\mathbf{x}_i\}$ is captured by Shannon *entropy*

term, widely used in data compression literature:

$$H(\mathcal{Y}|\mathcal{X}) = - \sum_{i=1}^N p_i \sum_{j=1}^k p(j|i) \log(p(j|i)). \quad (2.1.3)$$

Note that maximizing the entropy is commensurate with decreasing the *local* influence. The trade-off between maximizing the entropy and minimizing the modified distortion in (2.1.2) is addressed by seeking the probability distribution $\{(p(j|i))\}$ that minimize the *free-energy* function (or equivalent Lagrangian) given by

$$F \triangleq \bar{D}(\mathcal{X}, \mathcal{Y}) - \frac{1}{\beta} H(\mathcal{Y}|\mathcal{X}) + \sum_{i=1}^N \mu_i \left(\sum_{j=1}^k p(j|i) - 1 \right), \quad (2.1.4)$$

where the last term corresponds to $\{p(j|i)\}$ being a valid probability distribution. The Lagrange multiplier β bears a direct analogy to the inverse of the *temperature* variable in an annealing process [18]. Minimizing F at small values of β is equivalent to maximizing entropy H (a convex optimization problem). As β is increased gradually, minimization of F lays more emphasis on minimization of the distortion function. The association weights $\{p(j|i)\}$ that minimize the free-energy function are given by the *Gibbs* distribution

$$p(j|i) = \frac{e^{-\beta d(\mathbf{x}_i, \mathbf{y}_j)}}{\sum_{j'=1}^k e^{-\beta d(\mathbf{x}_i, \mathbf{y}_{j'})}}. \quad (2.1.5)$$

By substituting the Gibbs distribution into (2.1.4), the corresponding *free-energy* function is obtained as

$$F(\mathcal{Y}) = -\frac{1}{\beta} \sum_{i=1}^N p_i \log \left(\sum_{j=1}^k e^{-\beta d(\mathbf{x}_i, \mathbf{y}_j)} \right). \quad (2.1.6)$$

In the DA algorithm, the *free-energy* function is *deterministically* optimized at successively increased values of the annealing parameter β . The optimal facility locations \mathcal{Y} are obtained by setting the derivative of $F(\mathcal{Y})$ with respect to \mathbf{y}_j to zero, thereby resulting in following update equation

$$\mathbf{y}_j = \frac{\sum_{i=1}^N p_i p(j|i) \mathbf{x}_i}{\sum_{i=1}^N p_i p(j|i)} \quad \text{for } d(\mathbf{x}_i, \mathbf{y}_j) = \|\mathbf{x}_i - \mathbf{y}_j\|_2^2. \quad (2.1.7)$$

Note that the above equation has a form similar to computing centroids in k -means clustering algorithm. However in k -means clustering, the association between \mathbf{x}_i and \mathbf{y}_j are hard (0-1). The DA algorithm alternates between (2.1.5) and (2.1.7) at each β until convergence. In fact, the convergence of (2.1.6) is guaranteed as a consequence of coordinate descent on the free-energy function [23].

Since its inception, DA has been successfully applied to larger class of optimization problems such as,

pattern classification [24], image segmentation [25], graph aggregation [9], robust speech recognition [26], expectation-maximization [27], coverage control [28] and scheduling problems [8].

Shortcomings of the DA algorithm:

The DA algorithm is developed primarily in the context of FLPs, where the locations of site-points are required, and not the pairwise distances between them. On the other hand, many combinatorial optimization problems on graphs deal with instances where edge-weights between two nodes reflect the pairwise (dis)similarity between them. The DA algorithm proposed by Rose, lacks its generalization ability to handle problems with pairwise constraints. Moreover, the original DA algorithm does not allow for inclusion of general constraints, such as, capacity constraints, and it is often infeasible to derive convergence to sub-optimal solutions using MEP-based ideas. The first part of the thesis addresses these drawbacks for a wide range of combinatorial optimization problems, described in the next chapter.

Chapter 3

Applications of Maximum Entropy Principle

3.1 Traveling Salesman Problems and its Variants

The traveling salesman problem (TSP) [29, 30, 31] is one of the most extensively studied involving combinatorial optimization. TSP is an NP-hard problem and appears in operations research, computer science and manufacturing. A TSP is defined by a set of nodes or cities, and the edges connecting them which define the cost of travel between each city. Each solution to a TSP is referred as a *tour* and is made up of a combination of edges such that each node is visited sequentially. The optimal tour is the combination of edges that minimizes the total cost to the salesman.

A generalization of the TSP in which more than one salesman is allowed is referred as the Multiple Traveling Salesman Problem (mTSP). Note that a solution to mTSP comprises of several routes, one for each salesman, and the optimal tour would be the set of routes such that the total distance traveled is minimized. Additional constraints may be imposed on the system such as requiring each salesman to start at the same point, representing a warehouse or depot. These variants find applications in job-shop scheduling, such as, scheduling of orders at a steel rolling company [32], printing press scheduling [33], and vehicle routing problems [7].

Another important variant of the TSP is close-enough TSP (CETSP) [34], which is a variant where the salesman must only come within a certain radius of each city on the tour. This adds great complexity to the problem. Because of the significant increase in the number of edges, many conventional heuristics are unable to address this variant. CETSPs appear in the context of aerial reconnaissance, or monitoring wireless electric meters where it suffices to visit anywhere within the communication range of a meter. Unlike other variants of TSPs, CETSPs admit a continuum of feasible solutions and therefore are relatively complex to be addressed efficiently. Some special formulations are adopted to address CETSPs [34, 35]. Fig. 3.1 shows the schematic of variants of mTSP and CETSP.

In this thesis, we present a resource allocation viewpoint to TSP. This viewpoint enables to employ a modified version of the previously described deterministic annealing algorithm to TSP and its variants. In

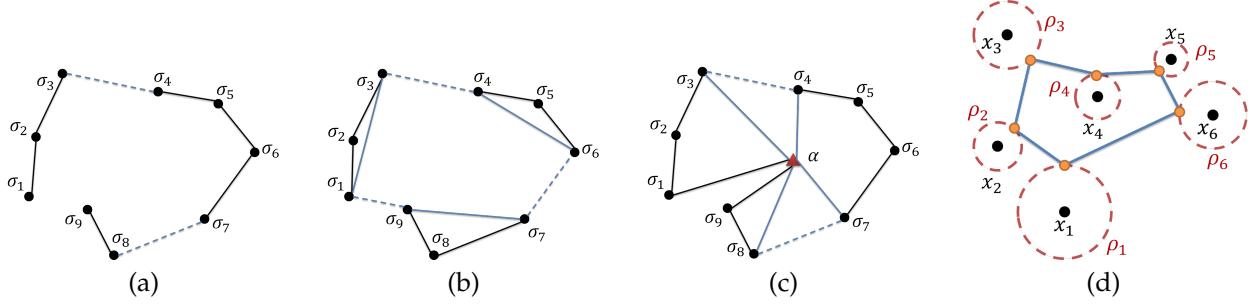


Figure 3.1: Schematic of a (a) 9-nodes non-returning 3TSP. The dashed blue lines indicate the removal of links, which corresponds to $\alpha_{k_1=3} = 1, \alpha_{k_2=7} = 1$ (b) 9-nodes returning 3TSP. The dashed blue lines indicate the removal of links $(-d_{\sigma_{k_j}, \sigma_{k_{j+1}}})$, while solid blue lines indicate the addition of links $(d_{\sigma_{k_j}, \sigma_{k_{(j-1)+1}}})$. (c) 9-nodes single-depot returning 3TSP. (d) Single salesman returning CETSP. Each node x_i is provided with a radius parameter ρ_i . The orange dots indicate y_j such that $v_{ij} = 1$ for some node i .

this viewpoint we consider an ordered set of facilities, where each facility is regarded as a *copy* of one of the cities. The MEP is used to consider every potential tour of the cities, and through the optimization process of DA the shortest tour through every city is determined. While the original DA algorithm was developed in the context of clustering, it was later adapted to the basic TSP as a case of constrained clustering [19], which serves as the foundation for our proposed extension to variants, such as mTSP and CETSP. Below we describe our approach to TSP and its variants in detail.

A primer on notations used in this section: The set of node (city) locations is denoted by $\mathcal{X} = \{\mathbf{x}_i : \mathbf{x}_i \in \mathbb{R}^2, 1 \leq i \leq n\}$, where n is the number of cities. We use $\alpha \in \mathbb{R}^2$ to denote the location of depot. The distance between any two nodes (cities) i and j is denoted by $d_{i,j}$. Unless stated otherwise, we consider squared-Euclidean distance between two nodes, i.e., $d_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$. We use S_n to denote the set of all permutations of $\{1, \dots, n\}$. An element of S_n is denoted by μ . An element $\sigma = [\sigma_1, \dots, \sigma_n, \sigma_{n+1}]$ in $\tilde{S}_n := S_n \cup \{n+1\}$ is given by:

$$\sigma_i := \sigma(i) = \begin{cases} \mu(i) & \text{if } i \in \{1, 2, \dots, n\} \\ \mu(1) & \text{if } i = n+1. \end{cases}$$

3.1.1 TSP: Problem Formulation and Approach

Given a set \mathcal{X} of n cities, the objective of TSP is to minimize the total length of a tour connecting the cities such that each city is visited just once and the salesman must return to his starting location on the

completion of tour. A TSP is mathematically described as:

$$\min_{\sigma \in S_n} \sum_{i=1}^n d_{\sigma_i, \sigma_{i+1}}.$$

The idea behind MEP based approach to classical TSP is that if we employ the same number of facilities as the number of nodes for clustering, i.e., if the number of facilities K in the DA algorithm are set to number of nodes n , then in the limiting case ($\beta \rightarrow \infty$), each node becomes a potential cluster, and the facility locations $\{\mathbf{y}_j\}$ coincide with node locations $\{\mathbf{x}_i\}$. The distortion function in the DA algorithm is modified to include the tour length. This requires then a second Lagrange multiplier θ for the tour length component of the distortion function, in addition to the first Lagrange multiplier β for the original component of the distortion function. Solving through the gradual change in both Lagrange multipliers leads to a solution of the TSP (and its variants). We now describe the proposed modification to the DA algorithm for solving a TSP. As before, we use $\mathcal{Y} = \{\mathbf{y}_j\}_{j=1}^n$ and $\mathcal{T} = \{\chi_{ij}\}_{i,j=1}^n$ to denote the set of facility locations and the set of associates, respectively.

Recall that an instance of an FLP is given by the tuple $(\mathcal{Y}, \mathcal{T})$. Following Equation 2.1.1, to every instance we associate a *distortion*-measure given by:

$$D_{\text{cluster}}(\mathcal{Y}, \mathcal{T}) \triangleq \sum_{i=1}^n \sum_{j=1}^n \chi_{ij} d(\mathbf{x}_i, \mathbf{y}_j), \quad (3.1.1)$$

which is the distortion of specific hard-clustering solution. To this distortion measure, we must augment another term that penalizes the current length of the tour, i.e., we consider a second distortion measure of the form:

$$D_{\text{TSP}}(\mathcal{Y}) \triangleq \theta \sum_{j=1}^n d(\mathbf{y}_j, \mathbf{y}_{j+1}). \quad (3.1.2)$$

Thus, the total distortion of a TSP is given by:

$$D(\mathcal{Y}, \mathcal{T}) \triangleq D_{\text{cluster}}(\mathcal{Y}, \mathcal{T}) + D_{\text{TSP}}(\mathcal{Y}) = \sum_{i=1}^n \sum_{j=1}^n \chi_{ij} d(\mathbf{x}_i, \mathbf{y}_j) + \theta \sum_{j=1}^n d(\mathbf{y}_j, \mathbf{y}_{j+1}). \quad (3.1.3)$$

Note that (3.1.3) is the distortion associated with an instance $(\mathcal{Y}, \mathcal{T})$. However, we aim to recover the instance with least distortion by ascribing a probability distribution over the set of all feasible instances. Let $P(\mathcal{Y}, \mathcal{T})$ denote the probability of an instance $(\mathcal{Y}, \mathcal{T})$. In the spirit of MEP formulation, we therefore

consider the expected distortion measure:

$$\bar{D} \triangleq \langle D(\mathcal{Y}, \mathcal{T}) \rangle = \sum_{\mathcal{Y}, \mathcal{T}} P(\mathcal{Y}, \mathcal{T}) D(\mathcal{Y}, \mathcal{T}). \quad (3.1.4)$$

Since we have no prior knowledge on $P(\mathcal{Y}, \mathcal{T})$, we use MEP to estimate them. The MEP objective can thus be formulated as:

$$\min_{\mathcal{Y}, \{P(\mathcal{Y}, \mathcal{T})\}} \sum_{\mathcal{Y}, \mathcal{T}} P(\mathcal{Y}, \mathcal{T}) D(\mathcal{Y}, \mathcal{T}) + \underbrace{\frac{1}{\beta} \sum_{\mathcal{Y}, \mathcal{T}} P(\mathcal{Y}, \mathcal{T}) \log P(\mathcal{Y}, \mathcal{T})}_{-H(P(\mathcal{Y}, \mathcal{T}))}, \quad \text{s.t. } \sum_{\mathcal{Y}, \mathcal{T}} P(\mathcal{Y}, \mathcal{T}) = 1, \quad (3.1.5)$$

where $H(P(\mathcal{Y}, \mathcal{T}))$ is the Shannon entropy term and β is the annealing parameter. The probability distribution (Gibbs distribution) that optimizes the objective function in (3.1.5) is given as:

$$P(\mathcal{Y}, \mathcal{T}) = \frac{e^{-\beta D(\mathcal{Y}, \mathcal{T})}}{\sum_{\mathcal{Y}', \mathcal{T}'} e^{-\beta D(\mathcal{Y}', \mathcal{T}')}}. \quad (3.1.6)$$

Since we are interested in estimating the most probable set of facility locations, we consider the marginal probability, given by:

$$P(\mathcal{Y}) = \sum_{\mathcal{T}} P(\mathcal{Y}, \mathcal{T}) = \frac{e^{-\beta F(\mathcal{Y})}}{\sum_{\mathcal{Y}'} e^{-\beta F(\mathcal{Y}')}}, \quad (3.1.7)$$

where $F(\mathcal{Y})$ is the analog of statistical *free-energy* and is given as:

$$F(\mathcal{Y}) = -\frac{1}{\beta} \sum_{i=1}^n \log \left(\sum_{j=1}^n e^{-\beta d(\mathbf{x}_i, \mathbf{y}_j)} \right) + \theta \sum_{j=1}^n d(\mathbf{y}_j, \mathbf{y}_{j+1}), \quad \text{with } \mathbf{y}_{n+1} = \mathbf{y}_1. \quad (3.1.8)$$

In order to obtain the optimal facility location \mathcal{Y} , we must set the derivative of the free-energy function to zero, i.e.,

$$\frac{\partial F}{\partial \mathbf{y}_j} = \mathbf{0} \Rightarrow \sum_{i=1}^n p(j|i)(\mathbf{y}_j - \mathbf{x}_i) + \theta(2\mathbf{y}_j - \mathbf{y}_{j+1} - \mathbf{y}_{j-1}) = 0, \quad (3.1.9)$$

where $p(j|i) = \frac{e^{-\beta d(\mathbf{x}_i, \mathbf{y}_j)}}{\sum_j e^{-\beta d(\mathbf{x}_i, \mathbf{y}_j)}}$ denote the association probabilities. Thus the update equations for facility

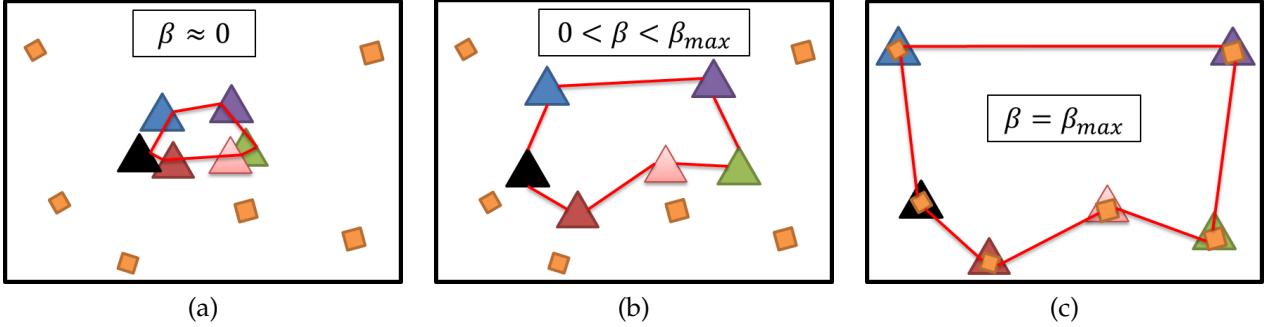


Figure 3.2: Pictorial representation of the working methodology of the DA algorithm for routing. Note that the desired number of resources (depicted by triangles) are same as the number of cities (depicted by squares). (a) At very low values of β , each city is uniformly associated with all the resources - all the resources are located at the centroid of the customer locations. (b) At an intermediate value of β , the *fuzziness* in the associations decreases and as a result resources start developing affinity to unique customer locations. (c) When $\beta \rightarrow \infty$, the resource associations become *hard* resulting in hard-clustered solutions as desired. At each value of β , the resources are constrained to maintain tension in the loop connecting them.

locations are given as:

$$\mathbf{y}_j = \frac{\sum_{i=1}^n p(j|i) \mathbf{x}_i + \theta(\mathbf{y}_{j-1} + \mathbf{y}_{j+1})}{\sum_{i=1}^n p(j|i) + 2\theta}. \quad (3.1.10)$$

In the proposed approach, the association probabilities and update equations for $\{\mathbf{y}_j\}$ are iterated at each β until convergence. The second Lagrange multiplier θ is related to the tour-length.

It is desirable to have a consistent and repeatable method for varying the Lagrange multipliers β and θ . In our proposed approach, the β multiplier is considered as the main driver, and the θ multiplier is secondary. As such, the θ parameter is decreased according to an exponential function until a stable tour length is reached, at which point β is increased according to an exponential function. This process is repeated until a sufficiently high β value and sufficiently low θ value are both reached, leaving the final solution. This approach is discussed in detail in Sec. 3.1.6.

Fig. 3.2 represents a snapshot of the working methodology of the enhanced DA-algorithm for TSPs. Note that at each value of β , the resources are constrained to maintain tension in the loop connecting them.

3.1.2 Non-Returning mTSP: Problem Formulation and Approach

In a non-returning mTSP (NR-mTSP), we are given a set of n cities and m salesmen to traverse these cities. The objective is to minimize the total tour-length such that each city is visited just once by only one sales-

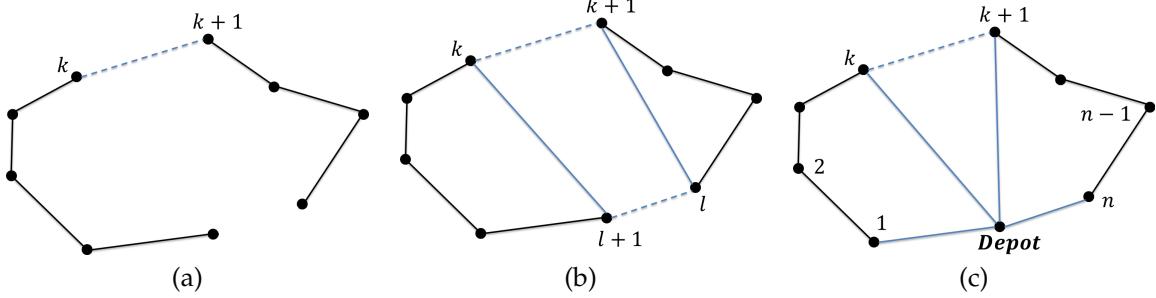


Figure 3.3: Schematic of a (a) Non-returning 2TSP, with $\mathcal{R} = k$. (b) Returning 2TSP, with $\mathcal{R} = \{k, l\}$. (c) Returning 2TSP (with Depot), with $\mathcal{R} = k$. The dashed blue lines indicate the removal of links, while solid blue lines indicate the addition of links.

man. The starting and ending node locations of each salesman do not coincide. This formulation is applicable to problems pertaining to non-recurring events, such as the scheduling of orders at a steel rolling company [32]. The non-returning m TSP can be mathematically described as:

$$\begin{aligned} & \min_{\substack{\sigma \in \mathcal{S}_n \\ \{\alpha_{k_j}\}}} \left\{ \sum_{i=1}^{n-1} d_{\sigma_i, \sigma_{i+1}} - \sum_{j=1}^{m-1} \alpha_{k_j} d_{\sigma_{k_j}, \sigma_{k_j+1}} \right\} \\ \text{s.t. } & \forall j, \sum_{k_j} \alpha_{k_j} = 1, \quad \alpha_{k_j} \in \{0, 1\}, \quad 1 \leq k_j \leq n-1. \end{aligned}$$

Unlike the classical TSP, in a NR-mTSP an instance is defined by three parameters, \mathcal{Y} , \mathcal{T} and \mathcal{R} . Here \mathcal{Y} corresponds to the set of facility locations \mathbf{y}_j , $\mathcal{T} = \{\chi_{ij}\}$ is a set of associates and describes the membership of node \mathbf{x}_i to facility \mathbf{y}_j , and \mathcal{R} is a set of locations of the partition representing the breaks between subsequent salesmen in the chain of consecutive facilities (see Fig. 3.3a). We first consider the case for $m = 2$ salesmen. Therefore an instance \mathcal{R} is mathematically described as:

$$\mathcal{R} = k; \quad \text{if there is no link b/w } \mathbf{y}_k \text{ and } \mathbf{y}_{k+1}.$$

As this is the non-returning version of the mTSP, there is no connection between \mathbf{y}_1 and \mathbf{y}_n . Therefore $\mathbf{y}_0 = \mathbf{y}_{n+1} = \mathbf{0}$ when they do appear in the equations. For a given instance of the problem $(\mathcal{Y}, \mathcal{T}, \mathcal{R})$, the distortion function in (3.1.3) is modified as:

$$D(\mathcal{Y}, \mathcal{T}, \mathcal{R}) = D_{\text{cluster}}(\mathcal{Y}, \mathcal{T}) + D_{\text{TSP}}(\mathcal{Y}) + D_{\text{NR}}(\mathcal{Y}, \mathcal{R}), \quad (3.1.11)$$

where the new component $D_{\text{NR}}(\mathcal{Y}, \mathcal{R})$ represents the partition of facilities for the independent salesmen and subtracts the distance at the partition between the facility locations \mathbf{y}_k and \mathbf{y}_{k+1} from the original

distortion function.

$$D_{\text{NR}}(\mathcal{Y}, \mathcal{R}) = -\theta d(\mathbf{y}_k, \mathbf{y}_{k+1}) \quad (3.1.12)$$

Following our approach to classical TSP, we ascribe a probability distribution $P(\mathcal{Y}, \mathcal{T}, \mathcal{R})$ over the set of all instances and then use MEP to estimate them. The associated free-energy function in this case is given by:

$$F = -\frac{1}{\beta} \sum_{i=1}^n \log \left(\sum_{j=1}^n e^{-\beta d(\mathbf{x}_i, \mathbf{y}_j)} \right) + \theta \sum_{j=1}^{n-1} d(\mathbf{y}_j, \mathbf{y}_{j+1}) - \frac{1}{\beta} \log \left(\sum_{k=1}^{n-1} e^{\beta \theta d(\mathbf{y}_k, \mathbf{y}_{k+1})} \right). \quad (3.1.13)$$

Setting the derivative of (3.1.13) with respect to each \mathbf{y}_j allows determination of the set of facility locations that maximize entropy in the system.

$$\frac{\partial F}{\partial \mathbf{y}_j} = \sum_{i=1}^n p(j|i)(\mathbf{y}_j - \mathbf{x}_i) + \theta(2\mathbf{y}_j - \mathbf{y}_{j+1} - \mathbf{y}_{j-1}) + \theta(\mathbf{y}_{j+1} - \mathbf{y}_j)\mathcal{P}(j) + \theta(\mathbf{y}_{j-1} - \mathbf{y}_j)\mathcal{P}(j-1) = 0,$$

where $p(j|i)$ is same as with classical TSP formulation. $\mathcal{P}(j)$ represents the probability that the partition occurs at facility j (i.e. $\mathcal{R} = j$) and is given by:

$$\mathcal{P}(j) = \frac{e^{\beta \theta d(\mathbf{y}_j, \mathbf{y}_{j+1})}}{\sum_{j=1}^{n-1} e^{\beta \theta d(\mathbf{y}_j, \mathbf{y}_{j+1})}}. \quad (3.1.14)$$

Solving for each y_j provides the solution to the system at this pair of β and θ values, so that for every facility location

$$\mathbf{y}_j = \frac{\sum_{i=1}^n p(j|i)\mathbf{x}_i + \theta\mathbf{y}_{j+1}(1-\mathcal{P}(j)) + \theta\mathbf{y}_{j-1}(1-\mathcal{P}(j-1))}{\sum_{i=1}^n p(j|i) + \theta(2-\mathcal{P}(j)-\mathcal{P}(j-1))} \quad (3.1.15)$$

Note that the (3.1.15) is only slightly more complex than the basic TSP formulation in (3.1.10). In fact, setting $\mathcal{P}(j) = 0, \forall j$ transforms the 2TSP into the basic TSP formulation.

Extension to m TSP for a general $m \geq 2$

Now that our approach to non-returning 2-TSP is established, we aim to generalize this approach to any general m . In this case, the partition set \mathcal{R} contains $m-1$ points, i.e., $\mathcal{R} = \{k_1, k_2, \dots, k_{m-1}\}$. The distortion function $D_{\text{NR}}(\mathcal{Y}, \mathcal{R})$ in (3.1.12) and the probability of a partition $\mathcal{P}(k_1, \dots, k_{m-1})$ in (3.1.14) are thus obtained

as:

$$D_{\text{NR}}(\mathcal{Y}, \mathcal{R}) = -\theta \sum_{i=1}^{m-1} d(\mathbf{y}_{k_i}, \mathbf{y}_{k_{i+1}})$$

$$\mathcal{P}(k_1, \dots, k_{m-1}) = \frac{e^{\beta \theta \sum_{i=1}^{m-1} d(\mathbf{y}_{k_i}, \mathbf{y}_{k_{i+1}})}}{\sum_{\tilde{\mathcal{R}}} e^{\beta \theta \sum_{i=1}^{m-1} d(\mathbf{y}_{k_i}, \mathbf{y}_{k_{i+1}})}}.$$

Note that the probability $\mathcal{P}(k_1, \dots, k_{m-1})$ is symmetric in its arguments, i.e.,

$$\mathcal{P}(k_1, k_2, \dots, k_{m-1}) = \dots = \mathcal{P}(k_{m-1}, k_1, \dots, k_{m-2})$$

Let us define $\tilde{\mathcal{R}} = \mathcal{R} \setminus \{k_1\}$ and denote $\mathcal{P}(j, k_2, \dots, k_m)$ by $\mathcal{P}(j, \tilde{\mathcal{R}})$. Then the characteristic equation for each facility location is given by:

$$y_j = \frac{\sum_{i=1}^n p(j|i) x_i + \theta y_{j+1} \left(1 - (m-1) \sum_{\tilde{\mathcal{R}}} \mathcal{P}(j, \tilde{\mathcal{R}})\right) + \theta y_{j-1} \left(1 - (m-1) \sum_{\tilde{\mathcal{R}}} \mathcal{P}(j-1, \tilde{\mathcal{R}})\right)}{\sum_{i=1}^n p(j|i) + \theta \left(2 - (m-1) \sum_{\tilde{\mathcal{R}}} (\mathcal{P}(j, \tilde{\mathcal{R}}) + \mathcal{P}(j-1, \tilde{\mathcal{R}}))\right)}$$

3.1.3 Returning mTSP: Problem Formulation and Approach

Fig. 3.1b shows a schematic of a returning m TSP. In a returning m -TSP (R-mTSP), we are given a set of n nodes ($\{\mathbf{x}_i\}$) and m salesmen, and the objective is to minimize the total tour length such that each node is visited by only one salesman, and the start and end positions of each salesman must be coincident. Many recurring events, such as job scheduling [33] fall under this category. The returning m TSP can be mathematically formulated as:

$$\min_{\substack{\sigma \in S_n \\ \{\alpha_{k_j}\}}} \left\{ \sum_{i=1}^n d_{\sigma_i, \sigma_{i+1}} + \sum_{j=1}^m \alpha_{k_j} \left[-d_{\sigma_{k_j}, \sigma_{k_{j+1}}} + d_{\sigma_{k_j}, \sigma_{k_{(j-1)+1}}} \right] \right\}$$

$$\text{s.t. } \forall j, \sum_{k_j} \alpha_{k_j} = 1, \quad \alpha_{k_j} \in \{0, 1\}, \quad 1 \leq k_j \leq n; k_0 = k_m.$$

Following on our approach to NR-mTSP, the distortion corresponding to partition function $D_R(\mathcal{Y}, \mathcal{R})$ must not only consider the distance between the facility locations where the partition occurs, it must also account for the distance incurred in completing the continuous tour by reconnecting to the other end of the loop (see Fig. 3.3b). Similar to the non-returning m TSP, we first derive the results for $m = 2$ salesmen and

then later extend it for a general m . The partition parameter \mathcal{R} in this case is described by two parameters.

$$\mathcal{R} = k, l \quad \text{if} \begin{cases} \text{no links b/w } \mathbf{y}_k \text{ and } \mathbf{y}_{k+1}, \mathbf{y}_l \text{ and } \mathbf{y}_{l+1}; \\ \text{links b/w } \mathbf{y}_k \text{ and } \mathbf{y}_{l+1}, \mathbf{y}_l \text{ and } \mathbf{y}_{k+1}; \end{cases}$$

It should be noted that the facility locations \mathbf{y}_1 and \mathbf{y}_n are considered to be adjacent, i.e. $\mathbf{y}_0 = \mathbf{y}_n$ and $\mathbf{y}_{n+1} = \mathbf{y}_1$. The tour-length distortion function is given by $D_{\text{TSP}}(\mathcal{Y}) = \sum_{j=1}^n d(\mathbf{y}_j, \mathbf{y}_{j+1})$. The distortion function $D_R(\mathcal{Y}, \mathcal{R})$ pertaining to the partition parameter is defined as:

$$D_R = \theta(-d(\mathbf{y}_k, \mathbf{y}_{k+1}) - d(\mathbf{y}_l, \mathbf{y}_{l+1}) + d(\mathbf{y}_k, \mathbf{y}_{l+1}) + d(\mathbf{y}_l, \mathbf{y}_{k+1})).$$

An important consequence of this framework is that if $k = l$, then the problem reduces to the classical returning TSP. Thus, this framework allows automatic determination of the optimal number of salesmen. As before, we ascribe the probability distribution $P(\mathcal{Y}, \mathcal{T}, \mathcal{R})$ over the space of all instances and use MEP to estimate them. MEP results in following update equations for facility locations:

$$\mathbf{y}_j = \frac{\sum_{i=1}^n p(j|i) \mathbf{x}_i + 2\theta \left(\sum_{l \neq j-1} \mathcal{P}(j, l) \mathbf{y}_{l+1} + \sum_{l \neq j} \mathcal{P}(j-1, l) \mathbf{y}_l \right) + \theta \left[(1 - 2 \sum_l \mathcal{P}(j, l)) \mathbf{y}_{j+1} + (1 - 2 \sum_l \mathcal{P}(j-1, l)) \mathbf{y}_{j-1} \right]}{\sum_{i=1}^n p(j|i) + 2\theta(1 - 2\mathcal{P}(j, j-1))} \quad (3.1.16)$$

Extension to m TSP for a general $m \geq 2$

We now extend our formulation for returning m TSP to a general $m \geq 2$. The partition set \mathcal{R} is defined as:

$$\mathcal{R} = \{k_1, k_2 \dots k_m\} \begin{cases} \text{no links b/w } \mathbf{y}_{k_i} \text{ and } \mathbf{y}_{k_{i+1}}; \\ \text{links b/w } \mathbf{y}_{k_i} \text{ and } \mathbf{y}_{k_{(i-1)+1}}; \end{cases}$$

with the understanding that $k_0 = k_m$. The distortion function and the corresponding probability distribution pertaining to the partition parameter are given by:

$$D_R(\mathcal{Y}, \mathcal{R}) = \theta \sum_{\mathcal{R}} \left\{ d(\mathbf{y}_{k_i}, \mathbf{y}_{k_{(i-1)+1}}) - d(\mathbf{y}_{k_i}, \mathbf{y}_{k_{i+1}}) \right\}$$

$$\mathcal{P}(\mathcal{R}) = \frac{\sum_{i=1}^m e^{-\beta\theta \left\{ d(\mathbf{y}_{k_i}, \mathbf{y}_{k_{(i-1)+1}}) - d(\mathbf{y}_{k_i}, \mathbf{y}_{k_{i+1}}) \right\}}}{\sum_{\mathcal{R}} \sum_{i=1}^m e^{-\beta\theta \left\{ d(\mathbf{y}_{k_i}, \mathbf{y}_{k_{(i-1)+1}}) - d(\mathbf{y}_{k_i}, \mathbf{y}_{k_{i+1}}) \right\}}}.$$

As before, it must be noted that the probability distribution $\mathcal{P}(k_1, \dots, k_m)$ is symmetric in its arguments. Let us define $\tilde{\mathcal{R}} = \mathcal{R} \setminus \{k_1, k_2\}$. Then the corresponding update equation for each facility location y_j is given by:

$$y_j = \frac{m\theta \sum_{\tilde{\mathcal{R}}} \mathcal{P}(j, j-1, \tilde{\mathcal{R}})(y_{j+1} + y_{j-1}) + \sum_{i=1}^n p(j|i)x_i + m\theta \sum_{k \neq m-1} \sum_{\tilde{\mathcal{R}}} \mathcal{P}(j, k, \tilde{\mathcal{R}})(y_{k+1} - y_{j+1}) + m\theta \sum_{k \neq m} \sum_{\tilde{\mathcal{R}}} \mathcal{P}(j-1, k, \tilde{\mathcal{R}})(y_k - y_{j-1})}{\sum_{i=1}^n p(j|i) + \theta \{1 - 2m [\sum_{\tilde{\mathcal{R}}} \mathcal{P}(j, j-1, \tilde{\mathcal{R}})]\}}.$$

3.1.4 Single Depot mTSP: Problem Formulation and Approach

Fig. 3.1c shows a schematic of a single-depot returning m TSP. In a single depot m -TSP (SD-mTSP), we are given a set of n nodes ($\{x_i\}$), a depot (α) and m salesmen, and the objective is to determine the optimal tour such that each node is visited by only one salesman. Each salesman must start and end at the depot. The total distance traveled by all salesmen is minimized. Real-world problems such as vehicle routing problem (VRP) [7, 8] with single-depot fall under this category. If $x_0 = \alpha$ denotes the location of depot and \mathcal{I} is the indexed-set of nodes, then the single-depot m TSP is mathematically described as:

$$\begin{aligned} & \min_{\sigma \in \tilde{\mathcal{S}}_n, \{\alpha_{k_j}\}} \left\{ \sum_{i=1}^{n-1} d_{\sigma_i, \sigma_{i+1}} + d_{\sigma_1, \alpha} + d_{\sigma_n, \alpha} + \sum_{j=1}^{m-1} \alpha_{k_j} \left[-d_{\sigma_{k_j}, \sigma_{k_j+1}} + d_{\sigma_{k_j}, \alpha} + d_{\sigma_{k_j+1}, \alpha} \right] \right\} \\ & \text{s.t. } \forall j, \sum_{k_j} \alpha_{k_j} = 1, \quad \alpha_{k_j} \in \{0, 1\}, \quad 1 \leq k_j \leq n-1, \end{aligned}$$

where $d_{\sigma_l, \alpha}$ denotes the distance between the node x_{σ_l} and depot α .

Fig. 3.3c shows the schematic of the proposed framework for the returning version of a single depot multiple salesmen problem. We first formulate the framework for $m = 2$ salesmen. The partition parameter \mathcal{R} in this case is defined as:

$$\mathcal{R} = k, \quad \begin{cases} \text{no link b/w } y_k \text{ and } y_{k+1}; \\ \text{links b/w } y_k \text{ and } \alpha, y_{k+1} \text{ and } \alpha. \end{cases}$$

The distortion function $D_{\text{depot}}(\mathcal{Y}, \mathcal{R})$ associated with partitioning parameter and the corresponding probability distribution $\mathcal{P}(\mathcal{R} = k)$ pertaining to the partition parameter are given by:

$$\begin{aligned} D_{\text{depot}}(\mathcal{Y}, \mathcal{R}) &= \theta (-d(y_k, y_{k+1}) + d(y_k, \alpha) + d(y_{k+1}, \alpha)) \\ P(\mathcal{Y}) &= \frac{e^{-\beta\theta \{-d(y_k, y_{k+1}) + d(y_k, \alpha) + d(y_{k+1}, \alpha)\}}}{\sum_{k=1}^{n-1} e^{-\beta\theta \{-d(y_k, y_{k+1}) + d(y_k, \alpha) + d(y_{k+1}, \alpha)\}}}. \end{aligned}$$

The distortion function corresponding to the tour-length constraint is modified to include the links be-

tween \mathbf{y}_1 and α , and between \mathbf{y}_n and α , i.e., $D_{\text{TSP}}(\mathcal{Y}) = \theta \left\{ \sum_{j=1}^{n-1} d(\mathbf{y}_j, \mathbf{y}_{j+1}) + d(\mathbf{y}_1, \alpha) + d(\mathbf{y}_n, \alpha) \right\}$. If we define $\mathcal{P}(0) = \mathcal{P}(n) = 1$, then the corresponding update equation for each facility location \mathbf{y}_j is given by:

$$\mathbf{y}_j = \frac{\sum_{i=1}^n p(j|i) \mathbf{x}_i + \theta \{ \mathcal{P}(j) + \mathcal{P}(j-1) \} \alpha + \theta \{ 1 - \mathcal{P}(j) \} \mathbf{y}_{j+1} + \theta \{ 1 - \mathcal{P}(j-1) \} \mathbf{y}_{j-1}}{2\theta + \sum_{i=1}^n p(j|i)}.$$

Extension to m TSP for general $m \geq 2$

In this case, the partition set \mathcal{R} consists of $m-1$ points, i.e., $\mathcal{R} = \{k_1, \dots, k_{m-1}\}$. The distortion and the probability distribution pertaining to the partition parameter are modified as:

$$D_{\text{depot}}(\mathcal{Y}, \mathcal{R}) = \theta \sum_{i=1}^{m-1} \{ -d(\mathbf{y}_{k_i}, \mathbf{y}_{k_{i+1}}) + d(\mathbf{y}_{k_i}, \alpha) + d(\mathbf{y}_{k_{i+1}}, \alpha) \}$$

$$\mathcal{P} = \frac{e^{-\beta\theta \sum_{i=1}^{m-1} \{ -d(\mathbf{y}_{k_i}, \mathbf{y}_{k_{i+1}}) + d(\mathbf{y}_{k_i}, \alpha) + d(\mathbf{y}_{k_{i+1}}, \alpha) \}}}{\sum_{\mathcal{R}} e^{-\beta\theta \sum_{i=1}^{m-1} \{ -d(\mathbf{y}_{k_i}, \mathbf{y}_{k_{i+1}}) + d(\mathbf{y}_{k_i}, \alpha) + d(\mathbf{y}_{k_{i+1}}, \alpha) \}}}$$

Note that the probability distribution $\mathcal{P}(k_1, \dots, k_{m-1})$ is symmetric in its arguments. If we define $\tilde{\mathcal{R}}$ by $\tilde{\mathcal{R}} = \mathcal{R} \setminus k_1$, then the corresponding update equation for each facility location \mathbf{y}_j is given by:

$$\mathbf{y}_j = \frac{\sum_{i=1}^n p(j|i) \mathbf{x}_i + (m-1)\theta \sum_{\tilde{\mathcal{R}}} (\mathcal{P}(j, \tilde{\mathcal{R}}) + \mathcal{P}(j-1, \tilde{\mathcal{R}})) \alpha + \theta \{ 1 - (m-1) \sum_{\tilde{\mathcal{R}}} \mathcal{P}(j, \tilde{\mathcal{R}}) \} \mathbf{y}_{j+1} + \theta \{ 1 - (m-1) \sum_{\tilde{\mathcal{R}}} \mathcal{P}(j-1, \tilde{\mathcal{R}}) \} \mathbf{y}_{j-1}}{\sum_{i=1}^n p(j|i) + 2\theta}.$$

3.1.5 CETSP: Problem Formulation and Approach

Fig. 3.1d shows a schematic of a single salesman returning CETSP. In a CETSP, we are given a set of n nodes ($\{\mathbf{x}_i\}$), each with a specified radius ($\{\rho_i\}$), and set of m salesmen with the objective of determining the optimal tour such that at least one salesman comes within ρ_i distance from each node \mathbf{x}_i . CETSPs are used to represent problems such as aerial reconnaissance [35] and establishing a wireless meter reader [34]. The CETSP variant may be applied to any of the TSP class of problems. The most significant difference between point-based TSPs and the CETSP is that due to the radius associated with each node, the CETSP does not define a specific edge between a pair of nodes, rather there is a continuum of possible edges between a pair of nodes. As a result, there are infinitely many possible solutions to this problem. A single

salesman returning CETSP is mathematically described as:

$$\begin{aligned} & \min_{\{\chi_{ij}\}, \{\mathbf{y}_j\}} \sum_{j=1}^n \left\{ \sum_{i=1}^n \chi_{ij} d_{CE}(\mathbf{x}_i, \mathbf{y}_j) + d(\mathbf{y}_j, \mathbf{y}_{j+1}) \right\}; \mathbf{y}_{n+1} = \mathbf{y}_1 \\ \text{s.t. } & \chi_{ij} \in \{0, 1\}, \quad \sum_{i=1}^n \chi_{ij} = 1 \forall j, \quad \sum_{j=1}^n \chi_{ij} = 1, \forall i \\ \text{where, } & d_{CE}(\mathbf{x}_i, \mathbf{y}_j) = \begin{cases} 0 & \text{if } \|\mathbf{y}_j - \mathbf{x}_i\| < \rho_i \\ (\|\mathbf{y}_j - \mathbf{x}_i\| - \rho_i)^2 & \text{else} \end{cases} \end{aligned}$$

For ease of exposition, we consider a single salesman CETSP in this work, however, the framework can be modified to additionally incorporate any of the aforementioned variants. Note that there are no partition parameters for a single salesman returning TSP. The distance between the node and the facility location pairs is modified as:

$$d_{CE}(\mathbf{x}_i, \mathbf{y}_j, \rho_i) = (\|\mathbf{y}_j - \mathbf{x}_i\| - \rho_i)^2 \quad (3.1.17)$$

The distortion functions corresponding to the node-facility distances and the tour-length constraints are respectively given by:

$$D_{\text{cluster}}(\mathcal{Y}, \mathcal{V}) = \sum_{i=1}^n \sum_{j=1}^n v_{ij} d_{CE}(\mathbf{x}_i, \mathbf{y}_j, \rho_i) \quad (3.1.18)$$

$$D_{\text{TSP}}(\mathcal{Y}) = \theta \sum_{j=1}^n d(\mathbf{y}_j, \mathbf{y}_{j+1}) \quad (3.1.19)$$

The free energy of this system is obtained as:

$$F = -\frac{1}{\beta} \sum_{i=1}^n \log \left(\sum_{j=1}^n e^{-\beta d_{CE}(\mathbf{x}_i, \mathbf{y}_j, \rho_i)} \right) + \theta \sum_{j=1}^n d(\mathbf{y}_j, \mathbf{y}_{j+1}).$$

Taking derivative of the free-energy term and setting it to 0, we obtain the update equation for each facility location given by:

$$\mathbf{y}_j = \frac{\sum_{i=1}^n p(j|i)(\mathbf{x}_i + \rho_i \text{sgn}(\mathbf{y}_j - \mathbf{x}_i)) + \theta(\mathbf{y}_{j+1} + \mathbf{y}_{j-1})}{2\theta + \sum_{i=1}^n p(j|i)},$$

where, the association probability distribution is now given by $p(j|i) = \left(\frac{e^{-\beta d_{CE}(\mathbf{x}_i, \mathbf{y}_j, \rho_i)}}{\sum_{k=1}^n e^{-\beta d_{CE}(\mathbf{x}_i, \mathbf{y}_k, \rho_i)}} \right)$ and $\text{sgn}(\cdot)$ is a vector-valued *signum* function.

3.1.6 Controlling the Lagrange Multiplier

Our approach to TSP involves a cost function with two Lagrange multipliers β and θ . While the primary Lagrange multiplier β is varied geometrically and is the primary driver for the proposed optimization algorithm, the secondary Lagrange multiplier θ is varied as a function of β . Many suggestions for varying the second Lagrange multiplier are suggested in the existing literature, such as, $\theta \propto 1/\sqrt{\beta}$ [36] or $\theta = \text{const.}$ [37].

Note that the secondary Lagrange multiplier θ is related to the tour length. We aim to constrain the tour-length at each β in order to avoid obtaining just the clustering solution. Since it is easier to work the Lagrange multiplier θ than the tour-length directly, the method adopted in this thesis uses a two-step optimization procedure [19]: (a) At a given β , gradually reduce θ until θ reaches some θ_{\min} , which maintains some “tension” in the net connecting the cities, (b) Keeping tour-length constant, update β and simultaneously obtain a new initial value of θ , and repeat (a).

We now describe how to determine a new initial value of θ during each β update, such that the free tour-length is kept constant. For the sake of brevity, the update procedure is derived only for the non-returning m TSP. The approach is quite general and is easily extended to address any of the variants of the classical TSP. For notational convenience, we use θ^* , \mathcal{Y}^* and L to denote the optimum value of secondary Lagrange parameter, optimal set of facility locations and optimal free tour-length at a given value of β , respectively. Finally, let $F^* \triangleq F(\mathcal{Y}^*, \theta^*) = F(\mathcal{Y}^*)$ denote the optimal value of the free-energy function in (3.1.13).

It can be shown [38] that for such constrained optimization, we have:

$$\theta^* = -\frac{\partial F^*}{\partial L}. \quad (3.1.20)$$

Therefore from (3.1.13) and (3.1.20), we have:

$$\begin{aligned} \frac{\partial \theta^*}{\partial \beta} &= -\frac{\partial}{\partial \beta} \left[\frac{\partial F^*}{\partial L} \right] \\ &= -\frac{\partial}{\partial L} \left[\frac{\partial F^*}{\partial \beta} + \sum_k \frac{\partial F^*}{\partial \mathbf{y}_k} \frac{\partial \mathbf{y}_k}{\partial \beta} \Big|_{\mathbf{y}_k=\mathbf{y}_k^*} \right] \\ &= -\frac{\partial}{\partial L} \left[\frac{\partial F^*}{\partial \beta} \right], \end{aligned} \quad (3.1.21)$$

where the last statement is a consequence of the fact that $\frac{\partial F}{\partial \mathbf{y}_j} = 0$ for all j at the optimum. Using the

function form of F from (3.1.13), $\frac{\partial F}{\partial \beta}$ evaluates to:

$$\begin{aligned} \frac{\partial F}{\partial \beta} &= \frac{1}{\beta^2} \sum_{i=1}^n \log \left(\sum_{j=1}^n e^{-\beta d(\mathbf{x}_i, \mathbf{y}_j)} \right) + \frac{1}{\beta} \sum_{i=1}^n \sum_{j=1}^n \frac{e^{-\beta d(\mathbf{x}_i, \mathbf{y}_j)}}{\sum_{j'=1}^n e^{-\beta d(\mathbf{x}_i, \mathbf{y}_{j'})}} d(\mathbf{x}_i, \mathbf{y}_j) \\ &\quad + \frac{1}{\beta^2} \log \left(\sum_{j=1}^{n-1} e^{\beta \theta d(\mathbf{y}_j, \mathbf{y}_{j+1})} \right) - \frac{\theta}{\beta} \frac{e^{\beta \theta d(\mathbf{y}_j, \mathbf{y}_{j+1})}}{\sum_{j'=1}^{n-1} e^{\beta \theta d(\mathbf{y}_{j'}, \mathbf{y}_{j'+1})}} d(\mathbf{y}_j, \mathbf{y}_{j+1}), \end{aligned} \quad (3.1.22)$$

which can be re-written as:

$$\frac{\partial F}{\partial \beta} = -\frac{F}{\beta} + \frac{1}{\beta} \underbrace{\left[\theta \sum_{j=1}^{n-1} d(\mathbf{y}_j, \mathbf{y}_{j+1}) + \sum_{i=1}^n \sum_{j=1}^n p(j|i) d(\mathbf{x}_i, \mathbf{y}_j) - \theta \sum_{j=1}^{n-1} \mathcal{P}(j) d(\mathbf{y}_j, \mathbf{y}_{j+1}) \right]}_{\triangleq E}, \quad (3.1.23)$$

where $\mathcal{P}(j)$ and $p(j|i)$ are as defined in (3.1.14). Note that the expression E in (3.1.23) is basically related to the free-energy F by:

$$E = \frac{\partial}{\partial \beta} (\beta F). \quad (3.1.24)$$

Thus from (3.1.21), (3.1.23) and (3.1.24), we have:

$$\begin{aligned} \frac{\partial \theta^*}{\partial \beta} &= -\frac{\partial}{\partial L} \left(\frac{E^* - F^*}{\beta} \right) \\ &= \frac{1}{\beta} \left(-\frac{\partial E^*}{\partial L} + \underbrace{\frac{\partial F^*}{\partial L}}_{=-\theta^*} \right) \quad [\text{from (3.1.20)}] \end{aligned} \quad (3.1.25)$$

Therefore, we consider the following first-order approximation for θ update:

$$\begin{aligned} \theta' &\approx \theta^* + \frac{\partial \theta^*}{\partial \beta} \Delta \beta \\ \Rightarrow \theta' &= \theta^* + \frac{\Delta \beta}{\beta} \left(-\frac{\Delta E^*}{\Delta L} - \theta^* \right) \quad [\text{from (3.1.25)}], \end{aligned} \quad (3.1.26)$$

where $\Delta E^* / \Delta L$ is estimated using the last two iterations in θ (before the moment to update β arrives). Note that (3.1.26) holds true for any TSP variant, however, the formulations of free-energy function F and free tour-length L are different for each variant. For instance, in a classical TSP, the free tour-length is simply the cumulative distance between consecutive facility locations, whereas in an m TSP the free tour-length

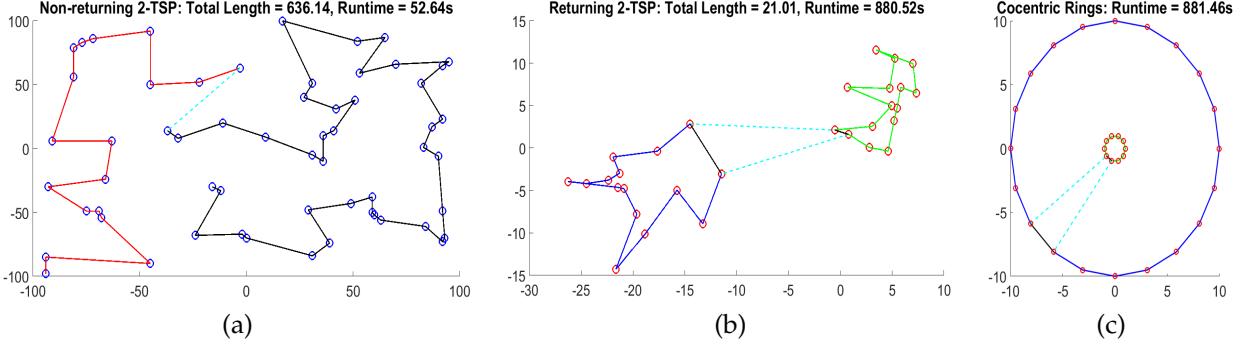


Figure 3.4: (a) Result for non-returning 2TSP for 59 nodes data. The dashed line indicates the removal of the link. (b) Result for returning 2TSP for 59 nodes data. The dashed lines indicate the removal of links, while the black solid lines indicate the addition of links. (c) Returning 2TSP version for cocentric rings.

includes the total length of all individual tours.

3.1.7 Experimental Evaluation

This section provides an overview of the results of the MATLAB implementations (on an Intel *i5 – 4200U @ 2.30GHz* machine) of the proposed heuristic. As yet, the MATLAB code used for this implementation has not been optimized for minimum computation time, so valid comparisons on the basis on run time are not currently available, however the heuristic is shown to achieve high quality results based on tour lengths in fairly reasonable amount of time. Tests are performed on both the *mTSP* heuristic and the *CETSP* variation. The heuristics are evaluated on synthetic data. We also compare the proposed MEP based approach against the optimized simulated annealing implementation ([39]) on a synthetic dataset of 30 different instances, each comprising of total number of nodes ranging from 100 to 200 nodes uniformly spanned in an area of $[-30, 30] \times [-30, 30] \in \mathbb{R}^2$.

Non-returning 2TSP: Fig. 3.4a shows the non-returning 2TSP result for a synthetic 59 nodes data. The tours of the two salesmen are shown in *red* and *black* respectively, with *cyan* dashed line indicating the partition.

Returning 2TSP: Fig. 3.4b shows the returning 2TSP result for a randomly generated 30 nodes data. The heuristic is able to find the two largest links to be removed from the sequence of facility locations. Fig. 3.4c shows the returning 2TSP result for a 30 nodes cocentric rings arrangement. The DA based heuristic finds the two most optimal routes for this configuration. Note that this dataset is particularly challenging for heuristics such as *cluster-first route-second*, where clustering the data first will either result in two symmetric subsets or the only cluster identified will be at the origin and when the two salesmen are allocated to the nodes, there is no way to effectively partition the set into two distinct subsets based on the information

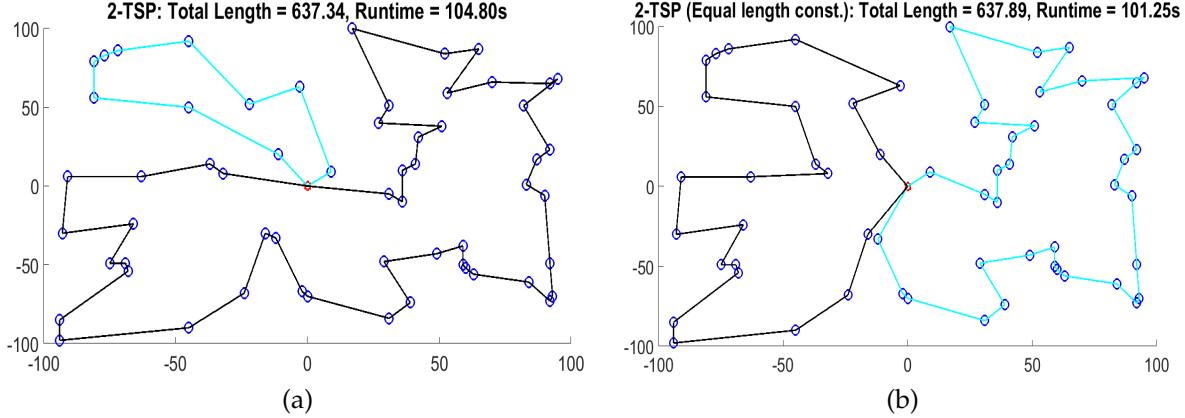


Figure 3.5: (a) Single-depot returning 2TSP solution to a 59 nodes data. The depot location is denoted by *red* marker. (b) Solution to the same 59 nodes data with equality constraint. Note that the optimal length is only slightly bigger than the earlier case.

provided by the clustering solution.

Single depot returning 2TSP: Fig. 3.5a shows the implementation results for the 59 nodes (and a depot) data. The two tours are shown in *cyan* and *black* colors respectively. Fig. 3.5b shows the result with an additional equality constraint. Note that the optimal length in this case is only marginally greater than the optimal length in the previous case.

Returning CETSP: Fig. 3.7 shows the implementation results for the CETSP on a randomly generated 10 nodes data with the additional radius parameter. It is difficult to determine whether the algorithm arrives at an optimal solution because this is much more difficult to check manually and unlike the standard TSP, there is no database of optimal tours for the CETSP. We have compared the heuristic against one of the 100 node sets (kroD100 from TSPLIB [40]) tested by Mennell for equal radii of 11.697[35]. Mennell achieves a tour length of 58.54 units with a 0.3 overlap ratio on the data. However, there are no details on the calculation time. The MEP based heuristic finds an optimal tour length of 64.99 units in 949 seconds. Note that in the current formulation, there is a penalty for a facility location existing either inside or outside of the circle. However, according to the problem formulation, there should be no penalty when the facility location exists within the radius of the node. This can be addressed by setting the derivative of the distance function $d_{CE}(x_i, y_j, \rho_i)$ with respect to y_j to zero whenever y_j exists within ρ_i distance from the node x_i . This negates the penalty incurred for placing a facility location within the radius of a node and should help this heuristic identify more accurate solutions.

Comparison with SA: Fig. 3.6 shows the comparison of the proposed MEP based approach against the simulated annealing (SA) based approach for 30 randomly generated instances for the multiple scenarios.

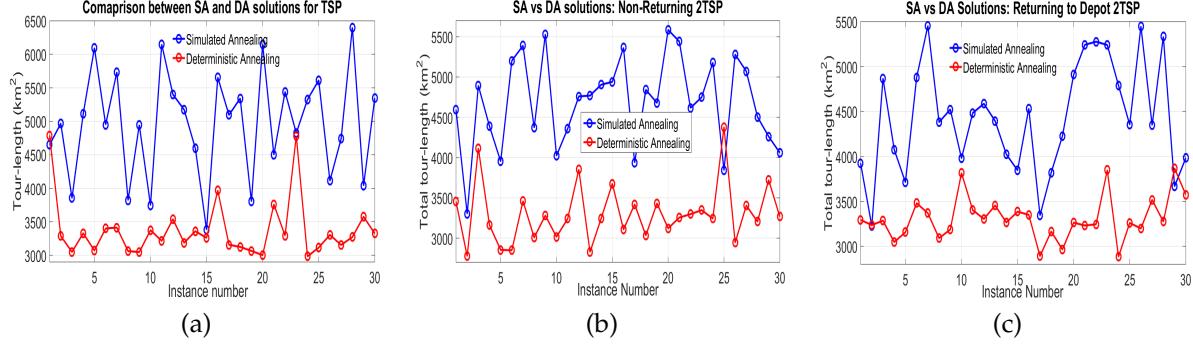


Figure 3.6: Comparison between MEP based deterministic annealing (DA) approach and simulated annealing (SA) approach for - (a) the usual traveling salesman problem (TSP), (b) non-returning 2TSP (NR2TSP), and (c) 2TSP returning to depot.

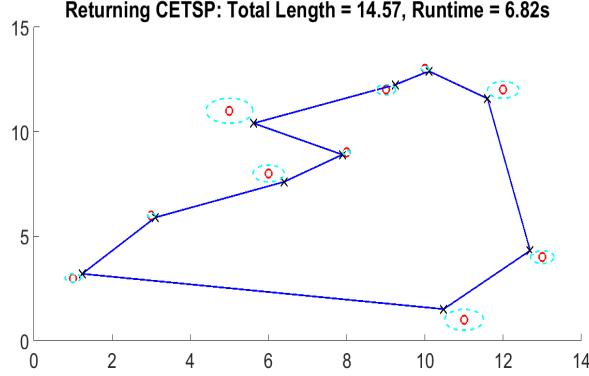


Figure 3.7: CETSP result for single salesman 10 nodes returning TSP. The red markers denote the node locations, while the black 'x' denote the facility locations. The cyan circles correspond to the radii ρ_i .

It should be remarked that both the algorithms require similar average computational time for each of the scenarios. We plot the total tour-lengths for each of the instances for the two approaches. Clearly the proposed MEP algorithm outperforms the most widely used simulated annealing algorithm with marginal increase in run-time.

3.2 Clustering of Power Networks

The North American electrical grid is regarded as the most significant engineering achievement of the 20th century [41], and yet the modern power transmission system faces major challenges due to ever increasing complex interconnections among multiple elements in the grid. Existence of strong links between underlying topological structure and performance in electrical networks have motivated for better strategies for managing and mitigating risks related to network failures. An electrical power transmission system can be represented as a network with nodes and links representing buses and impedance between the buses, respectively. Decomposing a large interconnected power network into smaller loosely-coupled groups facilitates easy and flexible management of the power transmission systems by allowing secondary voltage control at regional levels [42], controlled islanding that aims to prevent the spreading of large-area blackouts, and making the network robust to power and load fluctuations [43].

In this context, it is required to develop interpretable classifications of a given power network. More specifically, the aim of this project is to identify mutually decoupled (or loosely coupled) *clusters* (or zones) in a network such that in any unforeseen event of blackout or catastrophic failure, it is possible to control the spread of power outage and simultaneously identify the nodes that are most affected by the failure. In fact, we propose a clustering metric and approach such that any given node in a network is *tightly* coupled to the nodes within its cluster, while bearing *loose* coupling with nodes in other clusters. Thus the proposed approach reveals the underlying topological structure in the network by decomposing the large network into small number of tractable sub-networks.

Several recent works have looked at the problem of partitioning of electrical networks using varied approaches. From an abstract viewpoint, an electrical network can be represented by a directed-weighted graph where nodes represent electrical buses in the network, edges representing some notion of electrical connectivity, and weights representing the corresponding *strength* of connectivity. An important element of any graph-clustering approach is the quantification of the notion of *similarity* between any two nodes in a network. These quantifications include but not limited to - (1) *structural similarity*: based on quantities such as degree distribution of nodes and degree assortativity [44], graph diameter [45] and characteristic path length [46], (2) *topological similarity*: Here *electrical* distance is derived either from offline (non realtime) quantities such as nodal conductance matrix [47, 48] or power flow matrix [42] and online quantities, such as derived time-series phase angle data from phase measurement units (PMUs). While the measures of structural similarity are useful for comparing power grids with other graph structures, the absence of any underlying dynamics (arising from Kirchhoff's laws) fails to capture any electrical coupling among nodes of the network. Topological similarity measures alleviate this problem by introducing notion of *electrical*

distance obtained using circuit laws and network theorems. Furthermore, offline measures of similarity are preferred since the online methods rely on the observed data after the disturbance has occurred.

This thesis quantifies the electrical *similarity* between any two nodes based on the first-order perturbation matrix obtained by solving power flow equations [42]. We aim to cluster an electrical network such that nodes within each cluster have *similar* influence over the entire network. The proposed approach is general in the sense that different notions that quantify *similarity* and that quantify *influence* can be used. For ease of exposition, the approach is presented for a particular practical notion of influence; more precisely the influence of one node on another is characterized in terms sensitivity of voltage fluctuations at one node due to reactive power perturbations at the other node. This notion of influence is particularly useful since it encompasses electrical connectivity rather than only the network structure; for instance, two nodes that are strongly electrically coupled through the network even though not directly physically connected to each other will be considered similar in this notion, since voltage variation at one node brings about similar variation at the other node. Note that sensitivity to reactive perturbations strongly assesses the electrical coupling between buses, and not other features such as the amount of power being generated or consumed in the network. The proposed notion of similarity gives a measure of the electrical coupling between buses for a given reactive power circulation in the network. The approach proposed in this thesis is general and can easily accommodate other notions of similarity that depend both on active and reactive powers.

Furthermore, we show that the grouping of nodes (buses) achieved after clustering using this notion of influence is such that the voltage fluctuations at a node due to perturbations at nodes within the same cluster are more than voltage fluctuations due to perturbations at nodes from other clusters. That is, not only that perturbations at two nodes in the same cluster have similar effects on the entire network, the resulting voltage fluctuations at buses from other clusters are much smaller than the voltage fluctuations at the buses from the same cluster. Therefore, the algorithm partitions the electrical network into clusters or zones that are weakly coupled.

The algorithm to cluster nodes of a network is based on vector quantization (similar to DA), however, the algorithm deals with clustering of graphs and not the individual data points. Finally, a rule-based approach for decentralized voltage control is proposed, which exploits mutual decoupling of the partitioned network and requires only local control actions only at the local level. Similar rule-based expert system for voltage control is proposed in [49], albeit the control actions were not confined at local level.

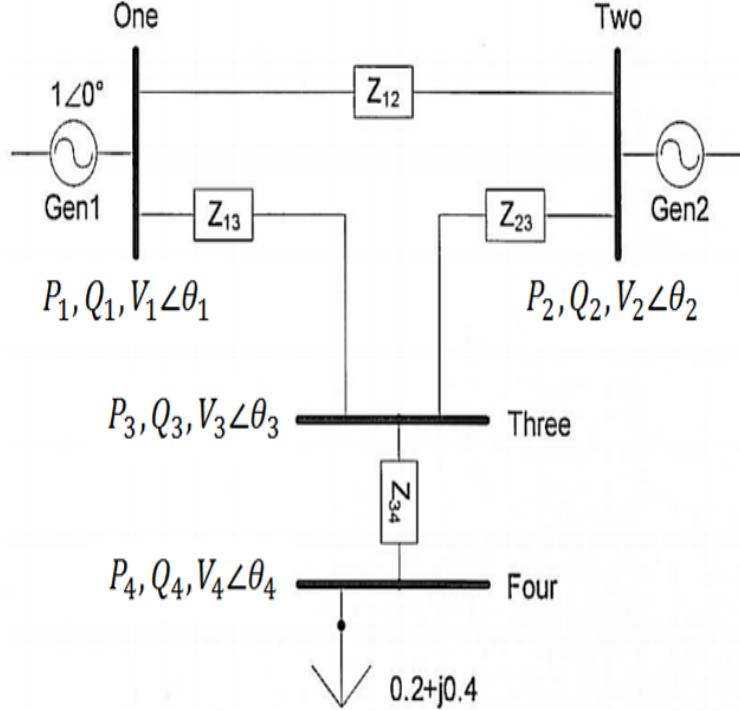


Figure 3.8: An example of a four-bus network. Bus 1 is a *Slack bus* whose voltage and angle are specified. Bus 2 is a *Generator bus* (*PV bus*) whose real output power P and commanded output voltage V are specified. Generators are set to regulate their own bus to a commanded voltage (by adjusting reactive power). Bus 3 is a *connecting bus* with zero load. The load power is known, and this is a *PQ bus*. Bus 4 is a *load bus* (*PQ bus*) with per unit (p.u.) active and reactive powers $P = 0 : 2; Q = 0 : 4$ both given. The buses are connected by line impedances Z_{ij} . The impedance matrix is denoted by \mathbf{Z}_{bus} . The inverse matrix $\mathbf{Y}_{\text{bus}} := \mathbf{Z}_{\text{bus}}^{-1}$ denotes the admittance matrix.

3.2.1 Quantification of Electrical Distance

Our approach for the quantification of electrical proximity of any two nodes is based on computation of the Jacobian matrix obtained by solving power flow equations [42]. In this context, we first describe the fundamental electrical quantities and matrix equations linking them, followed by quantification of electrical distances. Fig. 3.8 shows a four-bus electrical network, an example network with nodes and links representing buses and corresponding electrical connections, respectively. The buses can be of different types - Slack bus, Generator bus (or *PV bus*) and Load bus (or *PQ bus*). Each node i is completely specified by four physical variables - voltage magnitude V_i , phase θ , real power flow P_i , reactive power flow Q_i . The links are specified by line impedances Z_{ij} . \mathbf{Z}_{bus} denotes the impedance matrix of the network. The inverse matrix $\mathbf{Y}_{\text{bus}} := [Y_{ij}]$ is the admittance matrix of the network. The current injection at node i is given by I_i . \mathbf{V} and \mathbf{I} are the column vectors of voltage and current magnitudes, respectively. Similarly, $\mathbf{P}, \mathbf{Q}, \Theta$ are the column vectors depicting real power flows, reactive power flows and the voltage phase angles at the buses of an electrical network.

Recall that the admittance Y_{ij} is generally complex with real part (conductance) G_{ij} and imaginary part

(susceptance) B_{ij} , i.e. $Y_{ij} = G_{ij} + jB_{ij}$. These physical variables are related by the following governing equations

$$\begin{aligned}\mathbf{I} &= \mathbf{Y}_{\text{bus}} \mathbf{V}, \quad \mathbf{V} = \mathbf{Z}_{\text{bus}} \mathbf{I}, \\ P_j &= \sum_{k=1}^N V_k V_j \left(G_{kj} \cos(\theta_k - \theta_j) + B_{kj} \sin(\theta_k - \theta_j) \right), \\ Q_j &= \sum_{k=1}^N V_k V_j \left(G_{kj} \sin(\theta_k - \theta_j) - B_{kj} \cos(\theta_k - \theta_j) \right),\end{aligned}\tag{3.2.1}$$

where, N is the number of nodes (buses) in the network and $j \in \{1, \dots, N\}$. The last two equations are called the power flow equations, and they are necessary to address the *power flow problem* [50]. In a power flow problem, the voltage magnitudes and angles for one set of buses are desired when voltage magnitudes and power levels for another set of buses are known and when a model of the network configuration is available. In order to quantify *electrical distance* between any two nodes of a network, we consider small variations around an operating point (a power flow solution). The first order perturbations in the above electrical quantities are given by,

$$\begin{aligned}\Delta \mathbf{I} &= \mathbf{Y}_{\text{bus}} \Delta \mathbf{V}, \quad \Delta \mathbf{Q} = [\partial \mathbf{Q} / \partial \mathbf{V}] \Delta \mathbf{V}, \\ \Delta \mathbf{V} &= \mathbf{Z}_{\text{bus}} \Delta \mathbf{I}, \quad \Delta \mathbf{V} = [\partial \mathbf{V} / \partial \mathbf{Q}] \Delta \mathbf{Q},\end{aligned}\tag{3.2.2}$$

where matrices $[\partial \mathbf{Q} / \partial \mathbf{V}]$ and $[\partial \mathbf{V} / \partial \mathbf{Q}] \in \mathbb{R}^{N \times N}$ are inverses of each other. While the former matrix appears as a Jacobian during a load-flow computation, the elements of the latter matrix (also known as *sensitivity matrix*) reflect the propagation of voltage variations due to reactive power injection at a node throughout the electrical transmission system.

Note that (3.2.2) not only represents the dynamical behavior of an electrical system, it also captures the couplings between different nodes of the grid. Using these equations, it is possible to study the sensitivity of an electrical variable (\mathbf{V} , \mathbf{I} , \mathbf{P} , or \mathbf{Q}) or any combination of them to perturbations of electrical variables at other nodes. Grouping of nodes based on such sensitivities prove very useful for subsequent resource allocation or power network management problems. For instance, (3.2.2) can be used to study the effect of injecting power at a particular node on the voltage magnitudes at the remainder of the network. Alternatively since reactive power management is critical to voltage control for inductive grids, one can study the effect of perturbations of reactive power at a node on the voltages at different nodes in the network. The proposed methodology in this thesis can address grouping of nodes based on a combination of sensitivity measures such as above. For ease of illustration, in this thesis we investigate the case where we are in-

terested in studying the sensitivity of voltage fluctuations caused at a node with respect to reactive power fluctuations at another node; in particular we consider inductive power networks where the effect of phase-angle perturbations on reactive power at each node is negligible. Most high power electrical networks are indeed largely inductive and therefore exhibit active-reactive decoupling, i.e., \mathbf{P} primarily depends on Θ and is almost independent of \mathbf{V} and similarly \mathbf{Q} depends primarily on \mathbf{V} and is independent of Θ [51].

The *influence* of one node on another node is given by the magnitude of voltage coupling between the two nodes, which is quantified in terms of *matrix of attenuation* $[\alpha_{ij}] \in \mathbb{R}^{N \times N}$, that is

$$\Delta V_i = \alpha_{ij} \Delta V_j, \quad \text{where } \alpha_{ij} := \left[\frac{\partial V_i}{\partial Q_j} \right] / \left[\frac{\partial V_j}{\partial Q_j} \right], \quad (3.2.3)$$

which quantifies the voltage fluctuation at node i per unit voltage fluctuation at j th node, when reactive perturbations are applied at node j . Note that the normalization in the definition of α_{ij} has two distinct advantages - (i) making the quantities dimensionless, (ii) assigning equal importance to all the nodes (i.e. $\alpha_{ii} = 1$ for all i). If $\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j$ denote the i^{th} and j^{th} columns of the matrix of attenuation, respectively, then the *electrical distance* between nodes i and j is defined as

$$d(i, j) = \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j\|_2^2 = \sum_{k=1}^N (\alpha_{ki} - \alpha_{kj})^2; \quad (3.2.4)$$

Qualitatively, two nodes i and j are *close*, when the *influence* of these nodes on the network (including the nodes i and j themselves) are commensurate with one another. Note that from the definition (3.2.3), the diagonal terms of the attenuation matrix satisfy $\alpha_{kk} = 1$, for all $1 \leq k \leq N$, and therefore for any $\epsilon > 0$, if

$$d(i, j) < \epsilon \Rightarrow |\alpha_{ii} - \alpha_{ij}| = |1 - \alpha_{ij}| < \epsilon$$

Similarly, we have $|1 - \alpha_{ji}| < \epsilon$. Therefore $|\alpha_{ij} - \alpha_{ji}| < 2\epsilon$. Therefore, if two nodes i and j are *close*, then as a consequence the influence of perturbations at nodes i and j on each other are *similar*. This observation implies that if we partition the nodes of a network in terms of how *similar* they are in influencing the network, then the influence of nodes on each other from the same cell in a partition will be large, that is close to 1.

3.2.2 Graph Clustering and Determining Zones in Electrical Network

With the notion of distance between buses proposed in the previous section, we view an electrical network as a weighted directed graph (digraph), where buses represent the nodes, the elements α_{ij} represent the

edge weights. This makes it amenable to a graph aggregation method developed in [9], where a large weighted directed graph \mathcal{G}_x with N nodes is approximated by a smaller weighted directed graph \mathcal{G}_y (with $K \ll N$ nodes) such that the smaller graph is the *best representation* of the larger graph; the extent of representation is quantified in terms a *dissimilarity* measure. In the resulting smaller graph, each node of \mathcal{G}_y can be viewed as representative of a set of nodes on the larger graph \mathcal{G}_x ; in fact, the algorithm explicitly gives the set of nodes in \mathcal{G}_x that each node of \mathcal{G}_y represents. Thus this graph aggregation can be used to cluster nodes in \mathcal{G}_x into K clusters, for a given notion of distance between two nodes. Accordingly we use the graph aggregation method to group the buses in the electrical network into clusters for the above notion of electrical distance. In the next section, we briefly present this graph aggregation algorithm and present its important features. A more rigorous and exhaustive treatment can be found in [9]. An important aspect of this thesis is that we reinterpret this algorithm in terms of a specific information theoretic view point. This reinterpretation enables answering questions such as identifying the most influential edges or couplings in the electrical network; disrupting which can cause the maximum change to the behavior of the electrical network.

A digraph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ is described in terms of $\mathcal{V}, \mathcal{E} \in \mathcal{V} \times \mathcal{V}$ and $\mathbf{W} \in \mathbb{R}_+^{|\mathcal{V}| \times |\mathcal{V}|}$ which represent the set of *nodes*, *edges* and the *edge-weight matrix*, respectively. Furthermore, $|\mathcal{V}| = N \in \mathbb{N}$ and the relative node weights are denoted by $\{p_i\}, i \in \{1, \dots, N\}$, which satisfy $p_i \geq 0$ with $\sum_i p_i = 1$. The *incoming vector* of the i^{th} node is described by the weights of its incoming edges and is denoted by $\mathbf{W}_i \triangleq [W_{1i}, \dots, W_{Ni}]^T$, the i^{th} column of the matrix \mathbf{W} . We consider a distance between two nodes i and j based on *edge connectivity* given by $d(\mathbf{W}_i, \mathbf{W}_j)$. Note that this distance measures *similarity* between nodes; for example, small value of $d(\mathbf{W}_i, \mathbf{W}_j)$ implies that nodes i and j have similar connectivity in the graph.

In graph clustering problems, a small representative graph \mathcal{G}_y with $|\mathcal{V}_y| = K$ is obtained from a large graph \mathcal{G}_x with $|\mathcal{V}_x| = N \gg K$ by aggregating *similar* nodes in \mathcal{V}_x into K *supernodes* and then determining the resulting connections among these supernodes. This partition of the nodes \mathcal{V}_x into K clusters, where each cluster is represented by a supernode in \mathcal{V}_y is represented by *partition function* $\phi : \mathcal{V}_x \rightarrow \mathcal{V}_y$ which is such that for any $1 \leq j \neq l \leq K$, $\phi^{-1}(j) \subset \mathcal{V}_x$ is non-empty, $\phi^{-1}(j) \cap \phi^{-1}(l) = \emptyset$ and $\cup_{j=1}^K \phi^{-1}(j) = \mathcal{V}_x$. Each partition function ϕ defines an *aggregation matrix* $\Phi \in \{0, 1\}^{N \times K}$ as:

$$\Phi_{ij} := [\Phi]_{i,j} = \begin{cases} 1 & \text{if } \phi(i) = j, \\ 0 & \text{otherwise.} \end{cases} \quad (3.2.5)$$

Before we state the graph aggregation problem precisely, we present an example for ease of exposition of the subsequent concepts. Consider the graph \mathcal{G}_x shown in Fig. ?? with $\mathcal{V}_x = \{1, 2, 3, 4\}$ with $|\mathcal{V}_x| = N = 4$

nodes. The corresponding edge-weight matrix is given by

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 1 \\ 2 & 2 & 2 & 0 \end{bmatrix}.$$

Suppose we want to determine a graph \mathcal{G}_y with two supernodes ($|\mathcal{V}_y| = K = 2$), that is $\mathcal{V}_y = \{1', 2'\}$, which aggregates the graph \mathcal{G}_x . In this example, note that \mathbf{X} contains duplicated columns, which indicates $\{1, 2, 3\}$ are similar; in fact have identical connectivities. Therefore it is easy to see that the a supernode (say $1'$) should correspond to the three nodes 1, 2, and 3 and another ($2'$) should correspond to the node 4; that is we have the partition function given by $\phi : \{1, 2, 3, 4\} \rightarrow \{1', 2'\}$ with $\phi^{-1}(1') = \{1, 2, 3\}$ and $\phi^{-1}(2') = \{4\}$. Therefore the corresponding aggregation matrix Φ is given by:

$$\Phi = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

In this construct, we also define a weight matrix $\mathbf{Z} \in \mathbb{R}^{N \times K}$ given by:

$$\mathbf{Z} = \begin{bmatrix} 0 & 0.5 \\ 0 & 1.5 \\ 0 & 1 \\ 2 & 0 \end{bmatrix}.$$

Note here that the column $\mathbf{Z}_{\phi(i)}$ approximates the i th column of \mathbf{X} ; in fact, in this example they are exactly the same. The element Z_{kl} in this weight matrix can be interpreted as a directed weight from the k th supernode to l th node. Since the first three rows correspond to the first supernode, this matrix can again be aggregated to obtain the weight matrix \mathbf{Y} of \mathcal{G}_y , that is

$$\mathbf{Y} = \Phi^T \mathbf{Z} = \begin{bmatrix} 0 & 3 \\ 2 & 0 \end{bmatrix},$$

which defines the aggregated graph. In this example, the graph aggregation essentially required aggre-

gation of the columns of the matrix \mathbf{X} ; more precisely, it required finding a partition matrix Φ and a corresponding weight matrix \mathbf{Z} such that the cost function $\min_{\Phi, \mathbf{Z}} d(\mathbf{X}_i, \mathbf{Z}_{\phi(i)})$ is minimized. Once the optimal double (Φ, \mathbf{Z}) are obtained the aggregated graph weight matrix is given by $\mathbf{Y} = \Phi^T \mathbf{Z}$. Accordingly a general problem of aggregating a large graph $\mathcal{G}_x(\mathcal{V}_x, \mathcal{E}_x, \mathbf{X})$ with $|\mathcal{V}_x| = N$ into a graph $\mathcal{G}_y(\mathcal{V}_y, \mathcal{E}_y, \mathbf{Y})$ with $|\mathcal{V}_y| = K < N$ is given by:

$$\min_{\Phi \in \chi, \mathbf{Z} \in \mathbb{R}^{N \times K}} p_i d(\mathbf{X}_i, \mathbf{Z}_{\phi(i)}), \quad (3.2.6)$$

where χ represents the set of all $\{0, 1\}^{N \times K}$ aggregation matrices; the edge-weight matrix \mathbf{Y} is then given by $\mathbf{Y} = \Phi^T \mathbf{Z}$. Here $\{p_i\}$, with $\sum_i p_i = 1$ have been added in the problem formulation to represent relative weights of the nodes of \mathcal{G}_x , which are known a priori; in the case where all nodes are equally important, we can choose $p_i = \frac{1}{N}$ for $1 \leq i \leq N$.

Interestingly, the cost function in the aggregation problem (3.2.6) is algebraically the same as the cost function that arises in source coding problem from information theory that we now describe. By making this connection, it becomes possible to avail the existing methods for the source coding problem to solve (3.2.6).

In the context of clustering of power networks, the matrix of attenuation $\alpha = [\alpha_{i,j}]$ in (3.2.3) is regarded as the edge-weight matrix of a given graph \mathcal{G}_x , i.e., $\mathbf{X} = \alpha$. Application of the DA algorithm results in determination of the weight matrix \mathbf{Z} and the aggregation matrix Φ , which result in a smaller representative graph \mathcal{G}_y with $\mathbf{Y} = \Phi^T \mathbf{Z}$. The aggregation matrix Φ in this case determines the *zones* (or underlying partition) in the electrical network. Note that from (3.2.5), the aggregation matrix Φ is defined through a corresponding partition function $\phi : \mathcal{V}_x \rightarrow \mathcal{V}_y$. The inverse map $\phi^{-1}(j) \subset \mathcal{V}_x$ for all $j \in \mathcal{V}_y$ defines a *zone* (or set of buses aggregated in a cluster) in the network.

3.2.3 Rule-Based Supervisory Voltage Control

The proposed clustering algorithm provides a classification of a power network intended for easy and flexible management of bus voltages in the network. The “local” voltage control is achieved using a rule-based expert system, similar to the work proposed in [49]. However in our work, we rely only on local (comprising of buses belonging to the same zone in a network) measurements and control actions to achieve desired voltage control. Thus the task of a large network-wide voltage control is reduced to control of many sub-networks with very few number of buses. The overall rule-based strategy is implemented as a set of IF-THEN rules, as described below:

Rule 1 : IF	There are no voltage violations,
THEN	Exit.
Rule 2 : IF	There are voltage violations,
THEN	Identify the bus with the largest violations as the target bus.
Rule 3 : IF	Target bus has been identified,
THEN	Identify zone to which target bus belongs.
Rule 4 : IF	PV bus in the zone with largest control margin has been identified,
THEN	Adjust PV bus generation voltage until *reactive generation limits are reached *OR voltage limit on PV bus is reached *OR bus voltage has been corrected
Rule 5 : IF	Target bus voltage has been corrected,
THEN	Start over from Rule 1.
Rule 6 : IF	PV generation/voltage limit is reached,
THEN	Identify next available generator in the zone with the largest control margin.
Rule 7 : IF	Next PV bus has been identified,
THEN	Start over from Rule 4.
Rule 8 : IF	PV bus hasn't been identified,
THEN	Identify shunt bus with largest margin.
Rule 9 : IF	Shunt bus has been identified,
THEN	Adjust shunt capacitance until *shunt capacitance limit is reached, *OR bus voltage has been corrected

Rule 10 : IF	Target bus voltage has been corrected,
THEN	Start over from Rule 1.
Rule 11 : IF	Shunt capacitance limit is reached,
THEN	Identify next available shunt bus.
Rule 12 : IF	Next shunt bus has been identified,
THEN	Start over from Rule 9.
Rule 13 : IF	Shunt bus hasn't been identified,
THEN	Identify zonal tap-changer corresponding to largest sensitivity for target bus.
Rule 14 : IF	Tap-changer has been identified,
THEN	Adjust tap-ratio until *limit on tap-ratio is reached *OR bus voltage has been corrected
Rule 15 : IF	Target bus voltage has been corrected,
THEN	Start over from Rule 1.
Rule 16 : IF	Tap-ratio limit is reached,
THEN	Identify next available tap-changer.
Rule 17 : IF	Next tap-changer has been identified,
THEN	Start over from Rule 14.

The above rule-based expert system is successfully applied to IEEE-14 and IEEE-30 bus systems for voltage correction and is discussed below.

3.2.4 Evaluation on IEEE Test Systems

The graph clustering algorithm in combination with rule-based control 3.2.3 is tested on some standard network configurations - IEEE-14 bus system and IEEE-30 bus system. Figs. 3.9a and 3.9b show the network configuration of the IEEE-14 bus and IEEE-30 bus test cases, respectively. The IEEE-14 bus test case represents a portion of the American Electric Power System (in the Midwestern US) as of February, 1962. The test case includes all different kinds of buses - Slack, *PV* and *PQ* comprising of 5 generator buses, 3

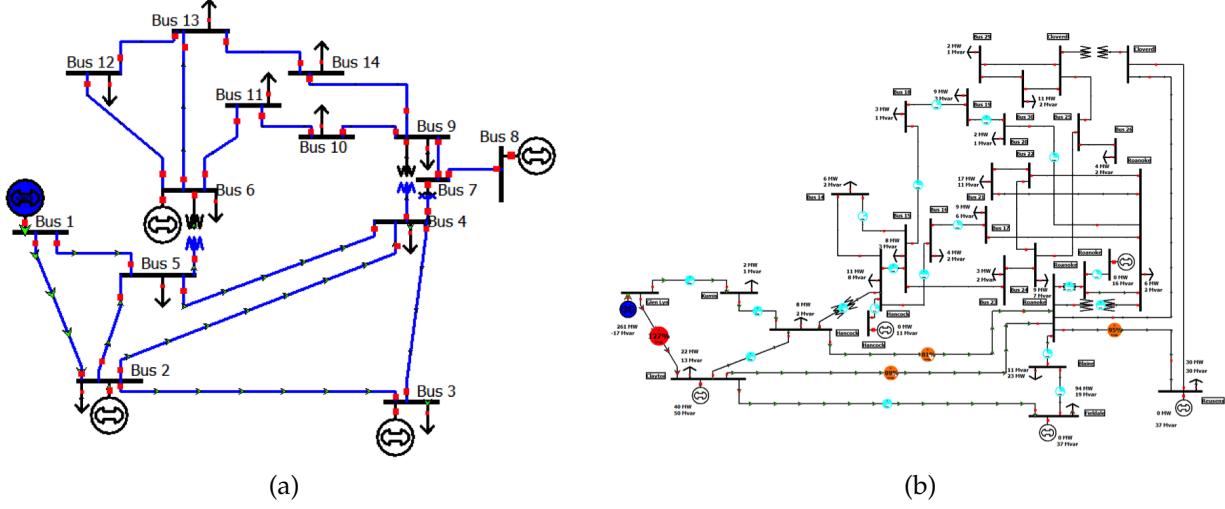


Figure 3.9: Network Configuration of (a) IEEE-14 bus, (b) IEEE-30 bus

tap changers and 1 shunt capacitor. The IEEE-30 bus test case represents a simple approximation of the American Electric Power system as it was in December 1961, and comprises of 8 generators, 4 tap changers and 2 shunt capacitors. The matrices of attenuation $[\alpha_{ij}]$ are first obtained by load-flow computations using Newton-Raphson method for the two test cases. The obtained matrices are then clustered into 3 partitions for the IEEE-14 bus test case and 2 partitions for the IEEE-30 bus test case. Tables 3.1 and 3.2 denote the clustering results for the two test systems. These partitions are marked by different colors in the 'Bus Type' columns and also indicated by corresponding initials. The results for various overloading and islanding scenarios are summarized below.

Effect of perturbations on inter and intra-cluster elements

The power-flow solutions in per unit (p.u.) at nominal loading conditions for the two test systems are indicated in column 3 of Tables 3.1 and 3.2, whereas columns 4, 5 and 6 indicate the effects of perturbing generator voltages at different buses. It is observed that the influence of these perturbations is larger at the buses belonging to the same group (cluster) where the perturbations originate. For instance, in the IEEE-14 bus system, doubling the generator voltage at bus 2 results in change in voltage magnitudes at buses 4 and 5 by about 0.4 p.u. The effect of this perturbation is less severe at other buses, which do not belong to the group formed by the buses 2, 3, 4 and 5. Note that bus 3 is a generator bus (*PV* bus) where voltage is set a priori, and hence there is no change in its voltage magnitude. Similar effects are seen in columns 5 and 6 when perturbing the generator voltages at buses 6 and 3, respectively. Interestingly, buses {6, 9, 10, 11, 12, 13, 14} in the IEEE-14 bus system are labeled as low-voltage (LV) buses, whereas buses 7

Table 3.1: Clustering Results For IEEE-14 Bus Data

Bus #	Bus Type	Volt mag. Oper- ating pt	Volt mag. ($2 \times V_2$)	Volt mag. ($1.5 \times V_6$)	Volt mag. ($2 \times V_3$)
1	Slack (B)	1.060	1.060	1.060	1.060
2	PV (Y)	1.045	2.090	1.045	1.045
3	PV (Y)	1.010	1.010	1.010	2.020
4	PQ (Y)	1.018	1.408	1.107	1.241
5	PQ (Y)	1.020	1.411	1.130	1.152
6	PV (G)	1.070	1.070	1.605	1.070
7	PQ (B)	1.062	1.238	1.188	1.162
8	PV (B)	1.090	1.090	1.090	1.090
9	PQ (G)	1.056	1.226	1.280	1.151
10	PQ (G)	1.051	1.192	1.331	1.130
11	PQ (G)	1.057	1.129	1.462	1.097
12	PQ (G)	1.055	1.068	1.572	1.063
13	PQ (G)	1.050	1.075	1.546	1.064
14	PQ (G)	1.036	1.144	1.382	1.096

and 8 are marked as tertiary-voltage (TV) buses. Remaining buses are indicated as high-voltage (HV) buses. This underlying electrical structure is naturally captured by the proposed clustering algorithm.

The proposed approach generalizes to larger bus systems too. Similar to the 14-bus test system, perturbing generator buses in the IEEE-30 bus test system result in large perturbations in buses belonging to the same cluster where the perturbations originate. While the algorithm was also tested for the IEEE-300 bus system, the details of it are excluded in this manuscript for the sake of brevity.

3.2.5 Effect of perturbations at buses within the same cluster over the remainder of the network

By construction, we have that two buses are considered close (electrically) when they have similar influence over the entire network. This is very well captured in the resulting partitions for the two test systems. We perturb generator voltages at buses 2 and 3 separately in the IEEE-14 bus system. Both these buses belong to the same cluster and result in similar perturbations over the entire network. For instance, both these buses have a very small influence on bus 12, affecting the voltage magnitudes by 0.013 p.u. and 0.008 p.u., respectively. However, the effect is large on buses such as bus 7, where the changes in voltage magnitudes are 0.176 p.u. and 0.100 p.u., respectively.

Similar conclusions can be drawn for the IEEE-30 bus test system. Doubling generator voltages at buses

Table 3.2: Clustering Results For IEEE-30 Bus Data

Bus #	Bus Type	Volt mag. Operat-ing pt	Volt mag. ($2 \times V_2$)	Volt mag. ($2 \times V_{11}$)	Volt mag. ($2 \times V_{13}$)
1	Slack (G)	1.060	1.060	1.060	1.060
2	PV (Y)	1.045	2.090	1.045	1.045
3	PQ (Y)	1.021	1.234	1.057	1.094
4	PQ (Y)	1.012	1.286	1.057	1.103
5	PV (Y)	1.010	1.010	1.010	1.010
6	PQ (Y)	1.011	1.203	1.066	1.063
7	PQ (Y)	1.003	1.117	1.036	1.034
8	PV (Y)	1.010	1.010	1.010	1.010
9	PQ (G)	1.051	1.165	1.488	1.187
10	PQ (G)	1.045	1.174	1.343	1.296
11	PV (Y)	1.082	1.082	2.164	1.082
12	PQ (G)	1.057	1.070	1.150	1.654
13	PV (G)	1.071	1.071	1.071	2.142
14	PQ (G)	1.043	1.159	1.160	1.597
15	PQ (G)	1.038	1.157	1.177	1.550
16	PQ (G)	1.045	1.165	1.224	1.497
17	PQ (G)	1.040	1.167	1.303	1.354
18	PQ (G)	1.028	1.153	1.225	1.452
19	PQ (G)	1.026	1.152	1.256	1.396
20	PQ (G)	1.030	1.157	1.277	1.371
21	PQ (G)	1.033	1.164	1.318	1.295
22	PQ (G)	1.034	1.164	1.314	1.299
23	PQ (G)	1.027	1.154	1.202	1.451
24	PQ (G)	1.022	1.156	1.242	1.325
25	PQ (G)	1.018	1.161	1.177	1.230
26	PQ (G)	1.000	1.146	1.162	1.215
27	PQ (G)	1.024	1.172	1.144	1.177
28	PQ (Y)	1.007	1.155	1.059	1.061
29	PQ (G)	1.004	1.155	1.126	1.160
30	PQ (G)	0.992	1.145	1.116	1.150

Table 3.3: Ranges and steps of control measures

Serial #	Control measure	Range of control (p.u.)
1	Generator voltage	0.9 - 1.1
2	Tap changer	0.9 - 1.1
3	Shunt capacitor	0.0 - 0.5

2 and 11, both of which belong to the same cluster, have similar effect over the entire network. Their effects are large on buses such as bus 28, where the changes in voltage magnitudes are 0.148 p.u. and 0.052 p.u., respectively.

3.2.6 Rule-based supervisory voltage control

Each test system consists of generators, tap changers and shunt capacitors. The ranges and steps in which these control actions are varied is shown in Table 3.3.

CASE 1: IEEE-14 bus system with 3.5 times the active load and 4.2 times the reactive load

Desired Voltage: 0.90 p.u. to 1.10 p.u.

Fault: Voltages at buses 9, 10 and 14 are 0.886 p.u., 0.884 p.u. and 0.833 p.u., respectively and are below the lower limit of 0.90 p.u. (shown in column 3 of Table 3.4)

Action: Control actions and their steps are shown in Table 3.5.

Results: The bus voltages after correction are within the desired range and the corresponding magnitudes are shown in column 4 of Table 3.4.

As seen in Table 3.4, a sudden increase in active and reactive loads results in large violations in the bus voltages. In particular, voltage magnitudes at buses 9, 10 and 14 fall below the allowable limit of 0.9 p.u.. These violations are subsequently corrected through a set of control actions indicated in Table 3.5. Note that the *faulty* buses belong to green (G) zone. The operating voltage of generator at bus 6, which lies in the fault zone, is increased to compensate for low voltages at buses 9, 10 and 14. Once the generator bus voltage reaches the maximum allowable limit of 1.1 p.u., the shunt capacitor at bus 9 is increased in steps of 0.1 p.u. until the capacitance can no longer be increased further (max 0.5 p.u.). Finally, the tap-changer between buses 4

Table 3.4: Supervisory Control for IEEE-14 Bus Network

Bus #	Bus Type	Volt magnitude (Before correction)	Volt magnitude (After correction)
1	Slack (B)	1.060	1.060
2	PV (Y)	1.045	1.045
3	PV (Y)	1.010	1.010
4	PQ (Y)	0.945	0.956
5	PQ (Y)	0.954	0.965
6	PV (G)	1.070	1.100
7	PQ (B)	0.956	0.995
8	PV (B)	1.090	1.090
9	PQ (G)	0.886	0.960
10	PQ (G)	0.884	0.952
11	PQ (G)	0.960	1.011
12	PQ (G)	0.998	1.034
13	PQ (G)	0.966	1.007
14	PQ (G)	0.833	0.900

and 9 is adjusted until all the bus voltages are within the allowable limits. Thus, the proposed rule-based scheme relies only on “local” inputs and control actions for voltage correction and achieves the desired performance in fewer number of steps as compared to similar rule-based schemes proposed in [49, 52].

CASE 2: IEEE-30 bus system with 2.1 times the active load and 3.1 times the reactive load

Desired Voltage: 0.90 p.u. to 1.10 p.u.

Fault: Voltages at buses 18-27 and 29-30 are below the lower limit of 0.90 p.u.
(shown in column 3 of Table 3.6)

Action: Control actions and their steps are shown in Table 3.7.

Results: The bus voltages after correction are within the desired range and the corresponding magnitudes are shown in column 4 of Table 3.6.

Similar to the 14-bus system, increase in active and reactive loads result in large violations in voltage magnitudes at buses 18-27, 29 and 30, all of which belong to green (G) zone. As before, corrective action can be localized to green (G) zone for voltage control in the IEEE-30 bus system using fewer number of control actions. These control actions are indicated in Table 3.7.

Table 3.5: Control actions for 14-bus system

Serial #	Control action	Steps
1	Generator at 6	3
2	Shunt Capacitor at 9	4
3	Transformer between 4 and 9	4

Table 3.6: Supervisory Control for IEEE-30 Bus Network

Bus #	Bus Type	Volt magnitude (Before correction)	Volt magnitude (After correction)
1	Slack (G)	1.060	1.060
2	PV (Y)	1.045	1.045
3	PQ (Y)	0.966	0.978
4	PQ (Y)	0.956	0.970
5	PV (Y)	1.010	1.010
6	PQ (Y)	0.969	0.978
7	PQ (Y)	0.961	0.967
8	PV (Y)	1.010	1.010
9	PQ (G)	0.973	1.057
10	PQ (G)	0.914	1.034
11	PV (Y)	1.082	1.082
12	PQ (G)	0.972	1.034
13	PV (G)	1.071	1.100
14	PQ (G)	0.925	1.003
15	PQ (G)	0.908	0.999
16	PQ (G)	0.924	1.013
17	PQ (G)	0.901	1.013
18	PQ (G)	0.873	0.979
19	PQ (G)	0.862	0.975
20	PQ (G)	0.872	0.988
21	PQ (G)	0.874	1.009
22	PQ (G)	0.875	1.013
23	PQ (G)	0.866	0.997
24	PQ (G)	0.837	1.019
25	PQ (G)	0.844	0.992
26	PQ (G)	0.785	0.943
27	PQ (G)	0.877	0.999
28	PV (Y)	0.958	0.974
29	PQ (G)	0.815	0.947
30	PQ (G)	0.780	0.918

Table 3.7: Control actions for 30-bus system

Serial #	Control action	Steps
1	Generator at 13	3
2	Shunt Capacitor at 10	4
3	Shunt Capacitor at 24	5
4	Transformer between 6 and 9	4
5	Transformer between 28 and 27	2

Table 3.8: Controlled islanding for IEEE-14 Bus Network

Bus #	Bus Type	Volt mag. before islanding	Volt mag. after islanding
1	Slack (B)	1.060	1.060
2	PV (Y)	1.045	1.045
3	PV (Y)	1.010	1.010
4	PQ (Y)	1.018	1.037
5	PQ (Y)	1.020	1.041
7	PQ (B)	1.062	1.076
8	PV (B)	1.090	1.090

Controlled Islanding

Islanding is required whenever there is a fault and whenever the maintenance is required in a power network. A controlled islanding can not only prevent damage to customer equipment due to continued excessive generation, but also avoid widespread blackouts. Our classification approach is very well suited for operations such as controlled islanding. We consider a simulated scenario, where there is a fault at one of the load buses in green (G) zone in the IEEE-14 bus system and it is desired to avoid any cascading failure by appropriately isolating a major part of the network. The identification obtained using the proposed clustering algorithm provides a natural way to prevent such cascading failure. Since, the clustering algorithm partitions a network into mutually decoupled (loosely coupled) zones, it is natural to isolate the zones not containing the faulty bus. As a consequence we isolate the zones (B) and (Y) in the IEEE-14 bus system from zone (G). The effects of isolating the buses belonging to zones (B) and (Y) are shown in Table 3.8. Even though the 14-bus system is practically reduced to a 7-bus system, the effect of such an isolation is *minimal* in terms of changes in p.u. bus voltages. In fact, the largest such increase in p.u. voltage is imperceptible (only 0.021 p.u. at bus 5).

3.3 Clustering Nonlinearly Separable Data

The basic DA algorithm by Rose [19, 3] lacks the ability to cluster data points that are not linearly separable in the Euclidean space. Kernel k -means and spectral clustering methods have both been used extensively to cluster data that are non-linearly separable in input space. While there has been significant research since their inceptions, both the methods have some drawbacks. Similar to the basic k -means algorithm, the Kernel k -means algorithm is sensitive to initialization [53]. On the other hand, the spectral methods are based on finding eigenvectors and can be computationally prohibitive [5, 21]. We thus propose a novel, MEP-inspired algorithm that enjoys the best of both worlds. On one hand, similar to spectral clustering algorithms, the algorithm is independent of initialization; and on the other hand, the proposed algorithm does not require computation of eigenvectors. We refer to this algorithm as the weighted-kernel deterministic annealing (WKDA) algorithm. The WKDA algorithm has the ability to avoid poor local minima. Additionally, we show that the WKDA approach reduces to Kernel k -means approach as a special case. Finally, we extend the proposed algorithm to include constrained-clustering and present the results for a variety of interesting data sets.

3.3.1 Motivation and Related Work

Cluster analysis or *clustering* is a key element of unsupervised learning and has emerged as one of the fundamental problems in data mining in the recent years. It is used for exploratory data analysis to find hidden patterns in data, where the clusters are modeled using similarity measures based upon metrics such as Euclidean, Manhattan and Bergman divergences. These similarity measures represent distances of data points from their corresponding cluster centroids, or pairwise distances between any two data points in the input space.

A major drawback of most commonly used k -means or more evolved DA algorithms is their incapability to separate clusters that are non-linearly separable in input space. While the data points in Fig. 3.10a can be separated using a hyperplane in \mathbb{R}^2 (line), there is no such line that can separate data points distributed along two concentric circles in Fig. 3.10b. Thus, while k -means algorithm finds optimal linear separation of data points in Fig. 3.10b, such separations are indeed not natural and often undesired. Several approaches are proposed to tackle such a problem - (a) Agglomerative (or hierarchical) clustering [54], which uses linkage functions and distance thresholding on resulting dendograms [55], (b) Spectral clustering [5, 56], which requires computing eigenvectors of the associated graph Laplacian, and (c) Kernel k -means [53], which uses kernel-trick to map data points to higher-dimensional space and then clusters data points using

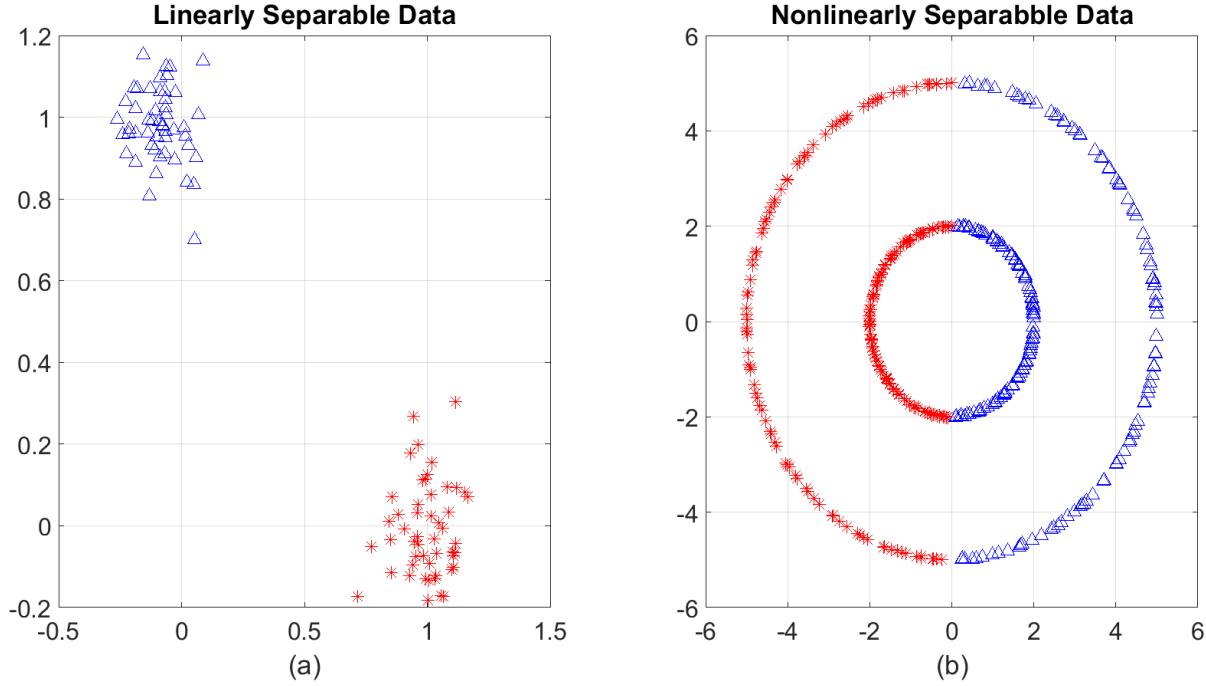


Figure 3.10: Results of k -means clustering algorithm on (a) linearly separable input data, and (b) nonlinearly separable input data.

linear separators in the new space.

While performance of agglomerative clustering is sensitive to choice of linkage functions and thresholds on cutting the resulting dendograms, computation of eigenvectors of large sparse matrices in spectral clustering can have substantial computational overheads, especially when a large number of eigenvectors are to be computed. On the other hand, similar to the basic k -means algorithm, the kernel k -means algorithm is sensitive to initialization and a poor initialization may result in undesirable clustering performance.

In order to overcome these limitations, we describe a novel, MEP-inspired weighted-kernel deterministic annealing (WKDA) algorithm below. The WKDA algorithm enjoys the best of both worlds. On one hand, the algorithm is independent of initialization much similar to the basic DA algorithm; and on the other hand, WKDA does not require computing eigenvectors. Furthermore similar to kernel k -means, by choosing the weights in particular ways, the WKDA objective function is identical to the normalized cut. Thus we can use WKDA-like iterative algorithms for directly minimizing the normalized-cut of a graph.

A word on notations: We use capital letters such as X, Y to denote matrices; and lower case bold letters such as \mathbf{x}, \mathbf{y} to denote column vectors. $N, k \in \mathbb{N}$ denote the number of data points and number of desired clusters, respectively. M denotes the number of attributes (or dimensions) of input data point. Script letters such as \mathcal{X}, \mathcal{Y} represent sets; $\|\mathbf{x}\|$ denotes the L^2 -norm of \mathbf{x} ; and $\|X\|_F$ denotes the Frobenius norm of matrix X , and is given by $\|X\|_F = \left(\sum_{i,j} X_{ij}^2\right)^{1/2}$.

3.3.2 The Weighted-Kernel Deterministic Annealing (WKDA) Algorithm

In the WKDA algorithm, we map the data \mathcal{X} in the input space to a higher-dimensional feature space through an appropriate choice of *kernel* functions. This approach is referred as “kernel trick” [20] and enables learning algorithms to operate in higher-dimensional feature space without ever explicitly computing the coordinates of data points in that space. The mapping allows to use linear separators to extract clusters in the *implicit* feature space.

Note that the squared Euclidean distance $d(\mathbf{x}_i, \mathbf{y}_j)$ can be expressed using inner-products as

$$d(\mathbf{x}_i, \mathbf{y}_j) = \langle \mathbf{x}_i, \mathbf{x}_i \rangle + \langle \mathbf{y}_j, \mathbf{y}_j \rangle - 2 \langle \mathbf{x}_i, \mathbf{y}_j \rangle, \quad (3.3.1)$$

where \mathbf{y}_j is defined in (2.1.7). For all \mathbf{x}_i and $\mathbf{x}_{i'}$ in the input space \mathcal{X} , *kernel* functions $\kappa(\mathbf{x}_i, \mathbf{x}_{i'})$ can be expressed as an inner product in higher-dimensional, *implicit* feature space \mathcal{H} using non-linear feature maps $\phi : \mathcal{X} \rightarrow \mathcal{H}$ which satisfies

$$\kappa(\mathbf{x}_i, \mathbf{x}_{i'}) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_{i'}) \rangle_{\mathcal{H}}. \quad (3.3.2)$$

While explicit representation of ϕ is not necessary, its existence is guaranteed as long as κ satisfies *Mercer’s* condition [57]. For a given set of data points $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ in the input space, a *kernel matrix* $K \in \mathbb{R}^{N \times N}$ is given by $K_{ii'} = \kappa(\mathbf{x}_i, \mathbf{x}_{i'})$. Mercer’s condition requires that K must be positive semi-definite (PSD) [57, 58]. Empirically, for kernel-based algorithms, choices of kernel function κ that do not satisfy Mercer’s condition may still perform reasonably if κ at least approximates the intuitive idea of similarity [59]. Many popular choices of κ exist, such as Gaussian, polynomial or radial basis function kernels. Using the non-linear distortion function ϕ , the distance between data point $\phi(\mathbf{x}_i)$ and facility location \mathbf{y}_j in the *implicit* feature space is given as

$$\underbrace{\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_i) \rangle + \langle \mathbf{y}_j, \mathbf{y}_j \rangle - 2 \langle \phi(\mathbf{x}_i), \mathbf{y}_j \rangle}_{d(\phi(\mathbf{x}_i), \mathbf{y}_j)}, \quad (3.3.3)$$

with

$$\mathbf{y}_j = \frac{\sum_i p(\mathbf{x}_i) p(\mathbf{y}_j | \mathbf{x}_i) \phi(\mathbf{x}_i)}{\sum_i p(\mathbf{x}_i) p(\mathbf{y}_j | \mathbf{x}_i)}, \quad p(\mathbf{y}_j | \mathbf{x}_i) = \frac{e^{-\beta d(\phi(\mathbf{x}_i), \mathbf{y}_j)}}{\sum_{j'} e^{-\beta d(\phi(\mathbf{x}_i), \mathbf{y}_{j'})}}. \quad (3.3.4)$$

Here (3.3.4) is a consequence of the DA algorithm with modified distance function $d(\phi(\mathbf{x}_i), \mathbf{y}_{j'})$. All computations in (3.3.3) are in the form of inner products, hence we can replace all inner products by entries of

the kernel matrix, i.e.,

$$d(\phi(\mathbf{x}_i), \mathbf{y}_j) = K_{ii} - 2 \frac{\sum_l p(\mathbf{x}_l) p(\mathbf{y}_j | \mathbf{x}_l) K_{il}}{\sum_l p(\mathbf{x}_l) p(\mathbf{y}_j | \mathbf{x}_l)} + \frac{\sum_{l,m} p(\mathbf{x}_l) p(\mathbf{x}_m) p(\mathbf{y}_j | \mathbf{x}_l) p(\mathbf{y}_j | \mathbf{x}_m) K_{lm}}{(\sum_l p(\mathbf{x}_l) p(\mathbf{y}_j | \mathbf{x}_l))^2}. \quad (3.3.5)$$

In the WKDA algorithm, the Euclidean distance in (3.3.5) is iteratively computed until convergence at each β value.

Algorithm 1 WKDA Algorithm

Input: $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$; No. clusters: k ; Kernel Matrix K ; Weight Matrix $W \triangleq \text{diag}\{p(\mathbf{x}_1), \dots, p(\mathbf{x}_N)\}$
Output: Cluster associations : $\{p(\mathbf{y}_j | \mathbf{x}_i)\}$

Initialization:

$$p(\mathbf{y}_j | \mathbf{x}_i) \leftarrow \frac{1}{k} \quad \forall \mathbf{x}_i \in \mathcal{X}, \mathbf{y}_j \in \mathcal{Y}$$

$$\beta \leftarrow \beta_{\min}$$

Annealing Process

while $\beta < \beta_{\max}$ **do**

β Iterations

while until convergence **do**

 Evaluate $d(\phi(\mathbf{x}_i), \mathbf{y}_j)$ as in (3.3.5) $\forall i, j$

 Evaluate $p(\mathbf{y}_j | \mathbf{x}_i)$ as in (3.3.4) $\forall i, j$

end while

 Increment β

end while

return $\{p(\mathbf{y}_j | \mathbf{x}_i)\}$

3.3.3 Connection with Kernel k -Means and Spectral Clustering Algorithms

The WKDA algorithm (Algorithm 1) shares many properties with the *kernel k-means* algorithm described in [53]. For instance, when the association probabilities $\{p(j|i)\}$ are hard (0-1), the distance function in (3.3.5) reduces to distance function for kernel k -means algorithm. Moreover similar to the DA algorithm, the WKDA algorithm decreases the objective function (modified free-energy function) in each β iteration.

For implementing the WKDA algorithm, we must compute the distance matrix $[d(\phi(\mathbf{x}_i), \mathbf{y}_j)]$ during each iteration. The complexity of the WKDA algorithm can be analyzed using (3.3.5). The main complexity arises from computing the numerator of the third term in (3.3.5). The complexity is $O(N^4k)$ scalar operations per iteration of computing the distance matrix. Thus if the total number of iterations is τ , then the complexity of the WKDA algorithm is $O(\tau N^4k)$. The complexity can be significantly reduced using scalable implementation of the WKDA [60]. Such scalable implementation uses thresholding on association weights to minimize the number of scalar computations arising from associating every data point in the

input space to *all* the clusters.

Similar to kernel k -means algorithm, the WKDA algorithm too exhibits connections with spectral clustering clustering algorithms such as normalized cut and ratio cut methods. Note that the WKDA algorithm aims to optimize the expected distortion $\bar{D}(\phi(\mathcal{X}), \mathcal{Y})$ given by

$$\bar{D}(\phi(\mathcal{X}), \mathcal{Y}) = \sum_{i=1}^N p_i \sum_{j=1}^k p(j|i) d(\phi(\mathbf{x}_i), \mathbf{y}_j). \quad (3.3.6)$$

Let W_j be the diagonal matrix of all the p_i weights in the j^{th} cluster C_j , i.e., $W_j \triangleq \text{diag} \left\{ p_{i_1}, \dots, p_{i_{|C_j|}} \right\} \forall i_l \in C_j$ and $W \triangleq \text{diag}\{W_1, \dots, W_k\}$, then the minimization of total-distortion $\bar{D}(\phi(\mathcal{X}), \mathcal{Y})$ in the limiting case (i.e. $p(j|i) \in \{0, 1\}$) is equivalent to the following trace maximization problem [53]

$$\min_{\mathcal{Y}, \{C_j\}} \bar{D}(\phi(\mathcal{X}), \mathcal{Y}) \equiv \max_{U \in \mathbb{R}^{N \times k}} \text{Tr} \left(U^T W^{1/2} \underbrace{\Phi^T \Phi}_K W^{1/2} U \right), \quad (3.3.7)$$

where $\Phi = [\Phi_1, \dots, \Phi_k]^T$ and Φ_j is a matrix of points of the form $\phi(\mathbf{x}_i)$ associated with cluster C_j , i.e., $\Phi_j \triangleq [\phi(\mathbf{x}_i)] \forall i \in C_j$. The matrix U is of the form given by

$$U = \begin{bmatrix} \frac{W_1^{1/2} \mathbf{e}_1}{\sqrt{s_1}} \\ \vdots \\ \frac{W_k^{1/2} \mathbf{e}_k}{\sqrt{s_k}} \end{bmatrix}, \quad (3.3.8)$$

where $s_j = \sum_{i \in C_j} p_i$ and \mathbf{e}_j is a vector of ones of appropriate dimension. Note that U is an orthonormal matrix, i.e., $U^T U = I$. This discrete optimization problem is relaxed by allowing U to be an arbitrary orthonormal matrix. Using Rayleigh-Ritz theorem, the optimal U for the relaxed problem is obtained by taking the top k eigenvectors of $W^{1/2} K W^{1/2}$. Similarly, the sum of the top k eigenvalues of $W^{1/2} K W^{1/2}$ gives the optimal trace value.

On the other hand, for a graph \mathcal{G} with edge-weight matrix A and degree-matrix D , the optimization problem for the relaxed normalized cut problem is given by

$$\max_{U \in \mathbb{R}^{N \times k}} \text{Tr} \left(U^T D^{-1/2} A D^{-1/2} U \right) \quad \text{s.t.} \quad U^T U = I. \quad (3.3.9)$$

Thus if we consider WKDA with $W = D$ and $K = D^{-1/2} A D^{-1/2}$, then the optimization problem in (3.3.9) is identical to the one in (3.3.7). Similarly, the optimization problem for the relaxed ratio cut problem is

given by

$$\max_{U \in \mathbb{R}^{N \times k}} \text{Tr}(U^T A U) \quad \text{s.t.} \quad U^T U = I. \quad (3.3.10)$$

Choosing $W = D^{1/2}$ and $K = D^{-1/2}AD^{-1/2}$ establishes the equivalence between the WKDA algorithm and the ratio cut approach. Thus, if the affinity matrix K is positive definite, we can use the WKDA procedure in order to minimize the normalized (or ratio) cut, without the need to compute eigenvectors.

Remark (Semi-supervised shape clustering): Semi-supervised clustering methods aim to improve clustering results using pairwise constraints, such as *must-link* and *cannot-link* constraints. These constraints can be incorporated into our framework through an appropriate modification of the kernel matrix. For every *cannot-link* constraint between \mathbf{x}_i and $\mathbf{x}_{i'}$, the corresponding entry in the kernel matrix is set to zero, i.e., $K(i, i') = 0$. This can be understood as follows. The WKDA algorithm replicates normalized cut (or ratio cut) and aims to minimize the associated cut value. Setting $K(i, i') = 0$ is equivalent to setting the edge-weight between i i' to zero in the associated graph. Thus any cut separating i and i' incurs zero cost. Hence, such a choice of kernel matrix favors viability of *cannot-link* constraints. Note that we still require positive definiteness of the kernel matrix in order to guarantee the existence of a kernel function.

Must-link constraints are relatively straight forward to handle. For every pair i, i' with *must-link* constraint between them, we require that the two points must be associated to the same cluster. This can be easily addressed in our framework by enforcing $p(j|i') = p(j|i)$ during each β iteration of the WKDA algorithm.

3.3.4 Experimental Evaluations of WKDA

We now provide experimental results to validate the usefulness of the proposed WKDA algorithm. Our WKDA algorithm is implemented in MATLAB and all experiments are done on a PC (Windows, Intel i7-4790 CPU @ 3.6GHz processor, 8 GB RAM). Note that a geometric scheduling rate of β update (i.e. $\beta_{t+1} = 1.05\beta_t$) is employed and thus results in fast clustering performance. The kernel matrices are generated using Gaussian kernels.

We first present the results on standard shape data sets - 'flame', 'pathbased' and 'R15', with 2, 3 and 15 natural clusters, respectively. These examples are downloaded from <https://cs.joensuu.fi/sipu/datasets/> under the shape sets category. The results are shown in Figs. 3.11a, 3.11b and 3.11c, respectively. Our WKDA algorithm successfully finds the underlying natural clusters in each of these examples. Similar performances are obtained for other standard shape data sets, too. However, the corresponding results are excluded for the sake of brevity. Fig. 3.11d presents an artificially generated generated data set composed of the string -

'ICC2018'. The example contains 7 natural clusters in the form of individual characters of the string. Our WKDA algorithm correctly finds the underlying clusters in this example. Moreover, we further require to cluster 'ICC2018' into 6 such natural constraints, such that some of pixels in the two 'C's belong to the same cluster (must-link constraint). The algorithm correctly recovers the clusters with both 'C's belonging to the same cluster (see Fig. 3.11e).

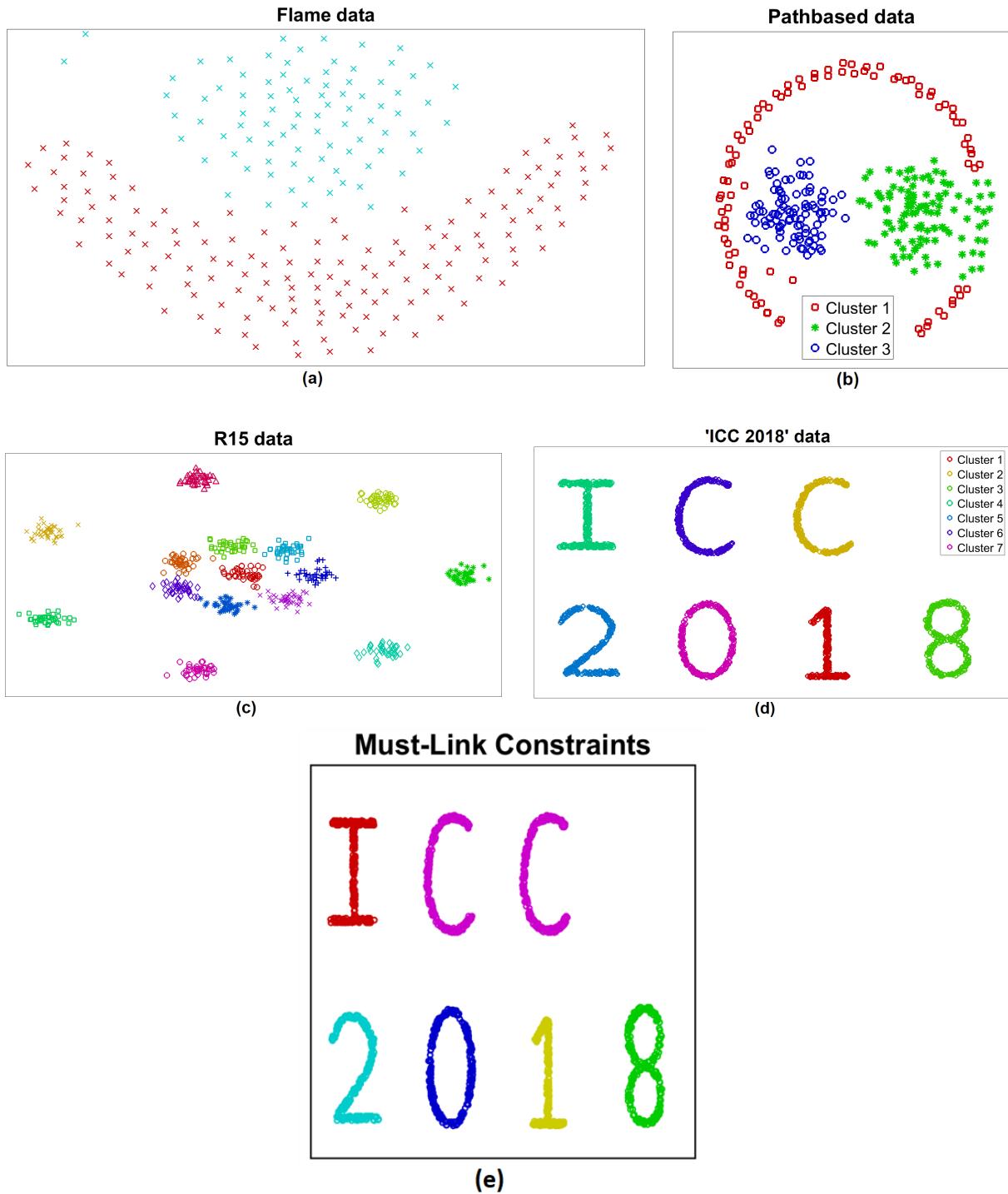


Figure 3.11: Implementation of the proposed WKDA algorithm on some interesting data sets. The data sets in examples (a), (b) and (c) are obtained from <https://cs.joensuu.fi/sipu/datasets/> under the shape sets category. The example presented in (d) is an artificially generated data set, whereas (e) presents the constrained version of (d).

3.4 Hard Problems on Graphs: Multiway k -Cut

Apart from clustering using pairwise distances, there are more general problems on graphs that are also NP-hard. Some of these problems include finding minimum multiway k -cut, maximum independent set, graph coloring, and maximum clique. In this section, we consider the problem of finding minimum multiway k -cut using MEP in detail and describe the convergence properties. The approach presented here is quite general and applicable to a broad class of combinatorial optimization problems on graphs. The iterative algorithm described in this thesis is designed to avoid poor local minima and its run-time complexity is only $\sim O(kN^3)$, where N is the number of vertices. Simulations on practical examples of interactive foreground-background segmentation and minimum multiway k -cut optimization for non-planar graphs that demonstrate the efficacy of the algorithm are presented.

3.4.1 Problem Description and Formulation

The *Multiway k -Cut* Problem [61] is a generalization of minimum weight cut problem and has applications in parallel and distributed computing [62], as well as in chip design. Multiway cut also finds applications in several other problems of related interest, such as extending a partial k -coloring of a graph [63]. Given a graph, $G = (V, E)$ with vertex set $V, |V| = N \in \mathbb{N}$, set of edges E , edge weights $w : E \rightarrow \mathbb{R}^+$, and a set of terminals $S = \{s_1, s_2, \dots, s_k\} \subseteq V$, a *multiway k -cut* is a set of edges whose removal disconnects the terminals from each other. The goal of the minimum multiway k -cut problem is to find a minimum weight set of edges $E' \subseteq E$ such that removing E' from G separates all terminals. Fig. 3.12a shows a schematic of a minimum multiway 3-cut problem with set of terminals denoted by $S = \{s_1, s_2, s_3\}$. The objective is to obtain a partition of the vertex set V into disconnected components $\{A_j : A_j \subset V, 1 \leq j \leq 3\}$, such that $s_j \in A_j$ for all j , and the total cut size, $\frac{1}{2} \sum_{j=1}^3 w(A_j, \bar{A}_j)$ is minimized.

While the problem of computing a minimum s - t cut (i.e., $k = 2$) is solvable in polynomial time, it is shown in [1] that the minimum multiway k -cut problem is not just NP-hard, but also APX-hard, i.e., there is a constant $\delta > 1$ such that it is NP-hard to even approximate the solution to within a ratio of less than δ to the optimal cost even when restricted to instances with three terminals ($k = 3$) and unit edge costs. The special case of the problem on planar graphs is also NP-hard if k can be arbitrarily large, but can be solved in polynomial time for every fixed k [64]. The complexity of multiway k -cut problem arises from the *combinatorial* number of ways in which the vertex set V can be partitioned into k sets.

In this thesis, we propose an MEP-based algorithm for the minimum multiway k -cut problem. We exploit the algebraic structure of the relaxed cost function and non-negativity of the Kullback-Leibler diver-

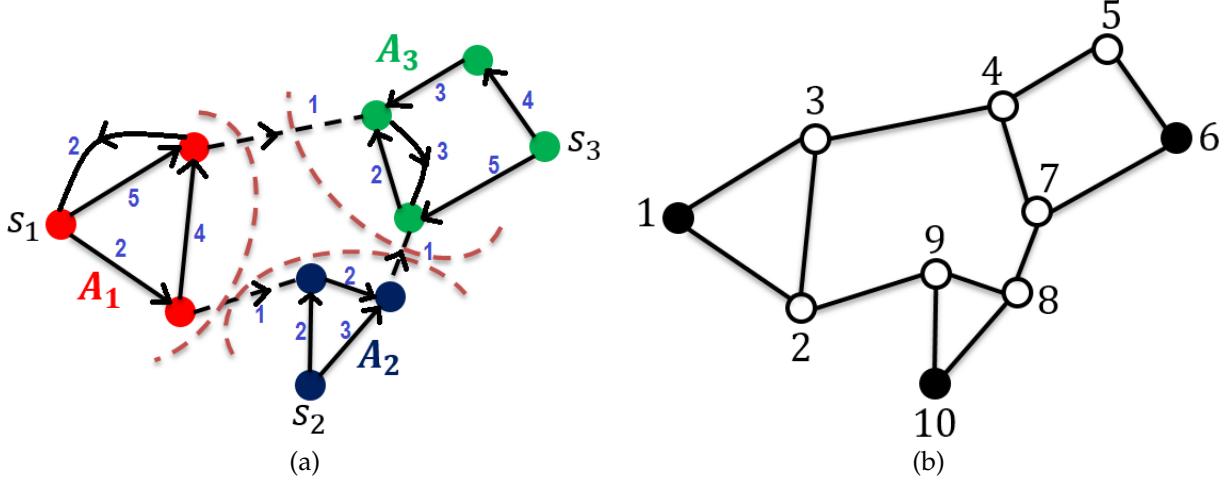


Figure 3.12: (a) Schematic of a minimum multiway k -cut problem with $k = 3$ and $S = \{s_1, s_2, s_3\}$. Here A_1, A_2 and A_3 are the disconnected components. (b) Illustrative example of a multiway 3-cut problem with $S = \{1, 6, 10\}$ described in Sec. 3.4.3.

gence to prove the convergence of the proposed algorithm to a minimum. Simulation results demonstrate that this algorithm outperforms the approximation algorithms described in [1, 65, 66]; for instance in the example shown in Fig. 3.16c, where the *isolating cut* heuristic achieves 132% of the optimal value - in fact, it is shown to be the best achievable value for the algorithm [1], whereas our MEP based algorithm achieves the optimal value. In particular, the graph in Fig. 3.16c can be generalized to any number of nodes, and the isolating cut heuristic results in a solution with approximation ratio of $2(1 - 1/k)$. Furthermore, for the st -min cut problem, which is solvable in polynomial time, the proposed algorithm has a better run-time complexity than the most commonly used Edmonds-Karp [67] or Dinic's [68] algorithm. We also demonstrate the effectiveness of our algorithm on very large graphs for interactive foreground-background segmentation.

For a given weighted directed graph, $G = (V, E, W)$ with vertex set $V, |V| = N \in \mathbb{N}$, set of edges $E \subseteq V \times V$, edge weights $w : E \rightarrow \mathbb{R}^+$ with $w(l, m) = W_{lm}$, and a set of terminals $S = \{s_1, s_2, \dots, s_k\} \subseteq V$, the minimum multiway k -cut problem is defined as:

$$\begin{aligned} & \min_{\{A_1, A_2, \dots, A_k\}, A_i \subset V} \frac{1}{2} \sum_{j_1=1}^k \sum_{j_2=1}^k \sum_{\substack{l \in A_{j_1} \\ j_2 \neq j_1 \\ m \in A_{j_2}}} w(l, m) \\ \text{s.t. } & s_1 \in A_1, \dots, s_k \in A_k, \quad \bigcup_{i=1}^k A_i = V, \text{ and} \quad A_j \cap A_l = \emptyset, \text{ for all } j \neq l. \end{aligned} \quad (3.4.1)$$

Note that a partition $\{A_1, \dots, A_k\}$ of V results in k disconnected components (subgraphs) of graph G (see Fig. 3.12a), and the above optimization problem seeks a partition that minimizes the cumulative weight of all the edges whose vertices belong to different components.

We reformulate the optimization problem in (3.4.1) by introducing $N \times k$ soft decision variables $\{p(i \in A_j)\}$. The variable $p(i \in A_j) \in [0, 1]$ denotes a probabilistic association of the vertex i with component A_j . We also require that for each vertex i , $\sum_j p(i \in A_j) = 1$, i.e., $\{p(i \in A_j)\}$ ascribes a probability distribution over all feasible associations over vertex set V . A relaxation of the optimization problem (3.4.1) is given by:

$$\begin{aligned} & \min_{\substack{\{p(i \in A_j)\} \\ 1 \leq i \leq N, 1 \leq j \leq k}} D \triangleq \frac{1}{2} \sum_{j_1=1}^k \sum_{\substack{j_2=1 \\ j_2 \neq j_1}}^k \sum_{l=1}^N \sum_{m=1}^N p(l \in A_{j_1}) p(m \in A_{j_2}) w_\lambda(l, m) \\ & \text{i.e., } \min \frac{1}{2} \sum_{\substack{j_1, j_2 \\ j_2 \neq j_1}} \sum_{l, m} p(l \in A_{j_1}) p(m \in A_{j_2}) w(l, m) - \frac{\lambda}{2} \sum_j \sum_l p(l \in A_j)^2 \\ & \text{s.t. } p(s_1 \in A_{j_1}) = \dots = p(s_k \in A_{j_k}) = 1, \sum_j p(i \in A_j) = 1 \forall i, \text{ and } p(i \in A_j) \in [0, 1], \end{aligned} \quad (3.4.2)$$

where $w_\lambda(l, m) = W_{lm}$, $W_\lambda \triangleq W + \lambda \mathbb{I}_N$. The inclusion of a constant parameter λ or equivalently the *regularizer* term in D is explained as follows. Note that in formulation (3.4.1), self-loop edges with node weights $w(l, l)$ can not be in a multiway k -cut, and therefore the solution to this problem is independent of these weights, that is, independent of the diagonal entries of the matrix W . However in the proposed relaxation, since each vertex has partial membership over different components A_k , the self-loop edges can become a part of the multiway k -cut; so we replace the edge-weight matrix W by $W + \lambda I_N$, where λ is large enough to make sure that self-loop edges are not included in the multiway k -cut; the choice of λ is discussed in Section 3.4.2.

Following our usual approach that employs maximizing entropy at successively increased values of distortion D , instead of directly solving (3.4.2), we solve for a closely related problem where we seek the distributions $\{p(i \in A_j)\}$ that ensure the cost function $D \leq D_0$ for some $D_0 > 0$. Accordingly in our case, the MEP would solve $\max H(\{p(i \in A_j)\})$ under the constraint that $D \leq D_0$, where D is given in (3.4.2) and the Shannon entropy term is given by:

$$H \triangleq - \sum_{i=1}^N \sum_{j=1}^k p(i \in A_j) \log p(i \in A_j). \quad (3.4.3)$$

The equivalent Lagrangian is thus defined as:

$$L \triangleq D - D_0 - \frac{1}{\beta} H + \sum_{i=1}^N \mu_i \left(\sum_{j=1}^k p(i \in A_j) - 1 \right), \quad (3.4.4)$$

where Lagrange multiplier β controls the trade-off between minimizing distortion and maximizing entropy,

and Lagrange multipliers $\{\mu_i\}$ correspond to equality constraints $\sum_j p(i \in A_j) = 1$ for all $1 \leq i \leq N$ that are not in $\{s_1, \dots, s_k\}$. Apart from ensuring that the solution approach does not make extra restrictive assumptions, the entropy term also ensures that the probabilities lie in the interval $[0, 1]$.

Using the fact that $\sum_{j_2 \neq j_1}^k p(i \in A_{j_2}) = 1 - p(i \in A_{j_1})$, the effective Lagrangian (*free-energy*) in (3.4.4) is given by:

$$\begin{aligned} F &\triangleq -D_0 + \frac{1}{2} \sum_{l,m=1}^N w_\lambda(l, m) - \frac{1}{2} \sum_{l,m=1}^N \sum_{j=1}^k p(l \in A_j) p(m \in A_j) w_\lambda(l, m) \\ &+ \frac{1}{\beta} \sum_{i=1}^N \sum_{j=1}^k p(i \in A_j) \log p(i \in A_j) + \sum_{i=1}^N \mu_i \left(\sum_{j=1}^k p(i \in A_j) - 1 \right). \end{aligned} \quad (3.4.5)$$

Let $\tilde{W}_\lambda = [\tilde{w}_{lm}^{(\lambda)}]$ be a symmetric matrix defined as $\tilde{W}_\lambda = \frac{W+W^T}{2} + \lambda \mathbb{I}_N$. Note that for a vector p and a matrix W of appropriate dimensions, the scalar $p^T W p = 0.5 p^T (W + W^T) p$. Moreover, the first two terms in the above expression are constants and therefore ignored in the subsequent analysis. Thus the *free-energy* term in (3.4.5) reduces to:

$$F \triangleq -\frac{1}{2} \sum_{l,m=1}^N \sum_{j=1}^k p(l \in A_j) p(m \in A_j) w_\lambda(l, m) + \frac{1}{\beta} \sum_{i=1}^N \sum_{j=1}^k p(i \in A_j) \log p(i \in A_j) + \sum_{i=1}^N \mu_i \left(\sum_{j=1}^k p(i \in A_j) - 1 \right). \quad (3.4.6)$$

The above reformulation is critical to the convergence analysis described in Section 3.4.2. By setting $\frac{\partial F}{\partial p(i \in A_j)} = 0$ toward optimizing F with respect to $p(i \in A_j)$ yields:

$$-\sum_{m=1}^N p(m \in A_j) \tilde{w}_{im}^{(\lambda)} + \frac{1}{\beta} (1 + \log p(i \in A_j)) + \mu_i = 0, \quad (3.4.7)$$

which results in following implicit equation for *Gibbs* distribution by using the equality constraint $\sum_j p(i \in A_j) = 1$,

$$p(i \in A_j) = \frac{\exp \left\{ \beta \left(\sum_{m=1}^N p(m \in A_j) \tilde{w}_{im}^{(\lambda)} \right) \right\}}{Z_i}, \quad (3.4.8)$$

where $Z_i = \sum_j \exp \left\{ \beta \left(\sum_{m=1}^N p(m \in A_j) \tilde{w}_{im}^{(\lambda)} \right) \right\}$ is the normalization constant corresponding to node i . The essence of the MEP-based approach lies in successive evaluation of Gibbs distribution in (3.4.8).

In our proposed algorithm, the *free-energy* is deterministically optimized through fixed-point iterations in (3.4.8) at successively increased values of the annealing parameter β . The algorithm is summarized in Algorithm 2. Note that at each value of the annealing parameter β , the algorithm executes the following

two steps until convergence is obtained:

$$\begin{aligned}
 \textbf{Step 1:} \quad \sigma_{lj}^+ &\leftarrow p(l \in A_j), \\
 \textbf{Step 2:} \quad p^+(l \in A_j) &\leftarrow \frac{\exp \left\{ \beta \left(\sum_{m=1}^N \sigma_{mj}^+ \tilde{w}_{lm}^{(\lambda)} \right) \right\}}{\sum_j \exp \left\{ \beta \left(\sum_{m=1}^N \sigma_{mj}^+ \tilde{w}_{lm}^{(\lambda)} \right) \right\}}, \\
 \forall l \in \{1, \dots, N\} \setminus S, j \in \{1, \dots, k\}
 \end{aligned} \tag{3.4.9}$$

Algorithm 2 Algorithm for minimum MultiwayCut

Input: $G = (V, E)$, $w : E \rightarrow \mathbb{R}^+$, $S = \{s_1, \dots, s_k\}$

Output: $\{p(i \in A_j)\}$

Initialization:

$$\begin{aligned}
 p(s_j \in A_j) &\leftarrow 1 \quad \forall s_j \in S \\
 p(i \in A_j) &\leftarrow \frac{1}{k} \quad \forall i \notin S, \forall j \\
 \beta &\leftarrow \beta_{\min}
 \end{aligned}$$

Annealing Process

while $\beta < \beta_{\max}$ **do**

Fixed-Point Iterations

while until convergence **do**

Update $\{p(i \in A_j)\}$ as in (3.4.8) $\forall i \notin S$

end while

Increment β

end while

return $\{p(i \in A_j)\}$

3.4.2 Convergence Analysis of the Proposed Algorithm

We now provide a convergence proof of the two-step iterations in (3.4.9). The convergence analysis is quite general and applicable to both directed and undirected graphs. The proof is based on appropriate algebraic transformation of the *free-energy* function in (3.4.6) and on the non-negativity of the Kullback-Leibler (KL) distance [69]. In particular, it is shown that an appropriately designed energy functional decreases during successive iterations of the implicit equations in (3.4.8). We use $P \in \mathbb{R}^{N \times k}$ to denote the matrix of associations $[p(i \in A_j)]$.

Claim 1 *The two-step iterations described in (3.4.9) converge for every value of the Lagrange multiplier β .*

Proof. Our proof is based on construction of a function $\Gamma(\zeta, \eta) : \mathbb{R}^{N \times k} \times \mathbb{R}^{N \times k} \rightarrow \mathbb{R}$ such that $\Gamma(P, P)$ is equal to $F(P)$ in (3.4.6). Consider the objective function Γ as:

$$\Gamma(\zeta, \eta) \triangleq \frac{1}{2} \sum_{l,m=1}^N \sum_{j=1}^k \eta_{lj} \eta_{mj} \tilde{w}_{lm}^{(\lambda)} - \sum_{l,m=1}^N \sum_{j=1}^k \zeta_{lj} \eta_{mj} \tilde{w}_{lm}^{(\lambda)} + \frac{1}{\beta} \sum_{i=1}^N \sum_{j=1}^k \zeta_{ij} \log \zeta_{ij}, \quad (3.4.10)$$

where the choice of Γ is motivated from the following observation:

$$-\frac{p^2}{2} \rightarrow \min_{\tau} \left(\frac{\tau^2}{2} - p\tau \right), \quad (3.4.11)$$

which is minimized at $\tau = p$. (3.4.10) is a consequence of the application of the vectorized form of the above algebraic transformation to the objective function in (3.4.6). We now show that, $\Delta \triangleq F(P) - F(P^+) = \Gamma(P, P) - \Gamma(P^+, P^+) \geq 0$, where P^+ is obtained after successive executions of steps 1 and 2 in (3.4.9). Note that,

$$\Delta = \underbrace{\Gamma(P^+, P) - \Gamma(P^+, P^+)}_{\Delta_1} + \underbrace{\Gamma(P, P) - \Gamma(P^+, P)}_{\Delta_2}. \quad (3.4.12)$$

We show that $\Delta_1 \geq 0$ by showing that $\Gamma(P^+, \sigma)$ (as a function of σ with fixed P^+) achieves its minimum when $\sigma = P^+$. The minimum is obtained by setting $\frac{\partial \Gamma}{\partial \sigma_{ij}}|_{(P^+, \sigma)} = 0$ in (3.4.10), which yields:

$$\sum_{m=1}^N \tilde{w}_{im}^{(\lambda)} \sigma_{mj} = \sum_{m=1}^N \tilde{w}_{im}^{(\lambda)} p^+(m \in A_j) \Rightarrow \sigma_{mj} = p^+(m \in A_j). \quad (3.4.13)$$

This solution $\sigma = P^+$ is a minimum if the Hessian $\frac{\partial^2 \Gamma(P^+, \sigma)}{\partial \sigma^2}|_{\sigma=P^+} = \tilde{W}_\lambda$ is positive-definite. Choosing $\lambda = d_{\max}$, maximum degree of the graph, ensures positive-definiteness of \tilde{W}_λ matrix. However, we must remark that $\lambda = d_{\max}$ is conservative, since in that case \tilde{W}_λ is positive-definite for all $\sigma \in \mathbb{R}^{N \times k}$. On the other hand, σ has a special structure where columns of σ sum up to one. In practice, we run the proposed algorithm directly without the regularizer term, i.e., $\lambda = 0$ and the algorithm generally monotonically converges in our experiments. Since P^+ is a minimizer, setting $\sigma = P \neq P^+$ results in $\Delta_1 \geq 0$.

To show that $\Delta_2 \geq 0$, note that from (3.4.10) we have

$$\begin{aligned} \Gamma(P, P) - \Gamma(P^+, P) &= - \sum_{lmj} (p(l \in A_j) - p^+(l \in A_j)) p(m \in A_j) \tilde{w}_{lm}^{(\lambda)} \\ &\quad + \frac{1}{\beta} \sum_{lj} p(l \in A_j) \log p(l \in A_j) + \frac{1}{\beta} \sum_{lj} p^+(l \in A_j) \log p^+(l \in A_j). \end{aligned} \quad (3.4.14)$$

Also from taking logarithm on both sides of (3.4.9), we have:

$$\sum_m p(m \in A_j) \tilde{w}_{lm}^{(\lambda)} = \frac{1}{\beta} \log Z_l + \frac{1}{\beta} \log p(l \in A_j) \quad (3.4.15)$$

(here Z_l is given by (3.4.8), which implies

$$\sum_{lj} (p(l \in A_j) - p^+(l \in A_j)) \sum_m p(m \in A_j) \tilde{w}_{lm}^{(\lambda)} = \frac{1}{\beta} \sum_{lj} (p(l \in A_j) - p^+(l \in A_j)) \log p^+(l \in A_j). \quad (3.4.16)$$

On substituting (3.4.16) in (3.4.14), we obtain

$$\Gamma(P, P) - \Gamma(P^+, P) = \frac{1}{\beta} \sum_{l=1}^N \sum_{j=1}^k P_{lj} \log \frac{P_{lj}}{p^+(l \in A_j)} = \frac{1}{\beta} \sum_{l=1}^N D_{KL}(P_l || P_l^+) \geq 0, \quad (3.4.17)$$

where p_l and p_l^+ are the l th rows of P and P^+ respectively and since Kullback-Leibler (KL) measure $D_{KL}(p_l || p_l^+)$ between two probability distributions is non-negative. Therefore, we have $\Delta_2 \geq 0$. Consequently, $\Delta \geq 0$.

Since by construction, $F(P) = \Gamma(P, P)$, we have that $F(P)$ decreases as a result of the two-step iteration in (3.4.9), and since F is bounded from below, the implicit equations in (3.4.8) converge to a local minimum of F . ■

Time-complexity of the proposed algorithm

The main complexity of the proposed MEP-based algorithm for the minimum multiway k -cut problem stems from the matrix multiplication in the fixed point iteration scheme. For a graph $G = (V, E)$ with $|V| = N$, there is a total Nk association probability parameters $\{p_i \in A_j\}$, that need to be estimated at each β iteration. Note that the *batch* update equation in (3.4.8) requires multiplying the two matrices $[w(i, j)]$ and $[p_i \in A_j]_{N \times k}$. This multiplication operation runs in $O(N|E|k)$ time (total of $N|E|$ operations for each partition associated with multiplying non-zero elements of edge-weight matrix). For each β , the fixed-point iteration converges in about 2-3 iterations (see Fig. 3.13a, which indicates average number of iterations per β for convergence for 3-cut problem on randomly generated graphs with varying number of nodes). If χ denotes the number of β iterations, then the runtime of the proposed MEP approach is bounded by $O(\chi N|E|k)$. Remark that for the minimum $s-t$ cut problem, the bound on the runtime of the proposed algorithm is $O(\chi N|E|)$ (experimentally verified in Fig. 3.13b for 3-cut problem on randomly generated graphs with varying number of nodes). Thus compared to the standard algorithms, such as Dinic's algorithm ($O(N^2|E|)$) and Edmonds-Karp algorithm ($O(N|E|^2)$), the proposed statistical physics based algorithm is

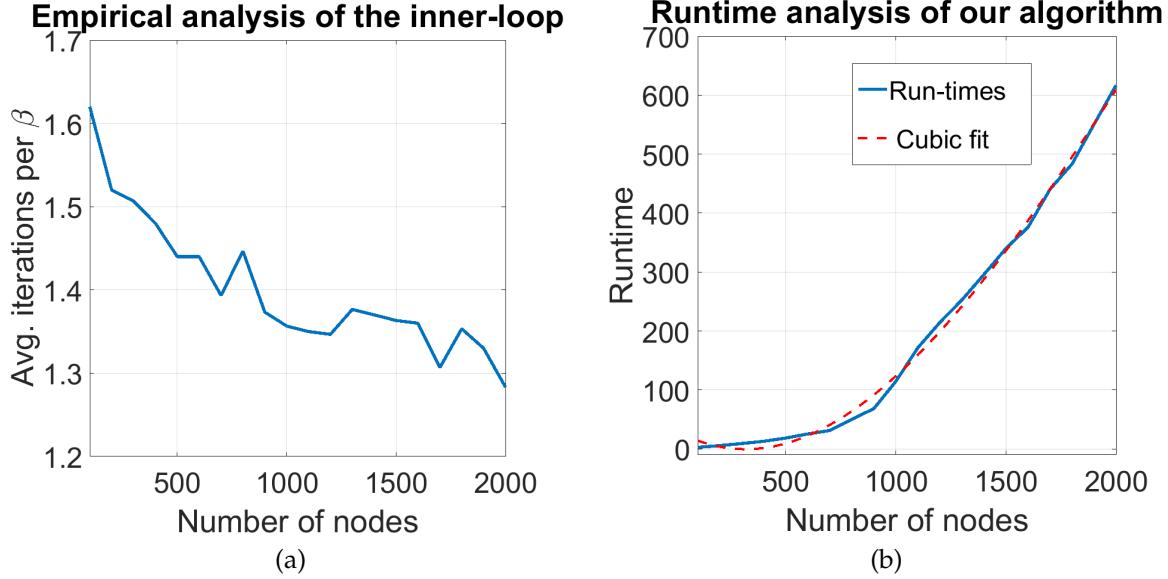


Figure 3.13: Empirical computational complexity of the proposed algorithm for 3-cut problem on randomly generated graphs with varying number of nodes. (a) Average number of iterations per β value, (b) Computational complexity is cubic in the number of vertices.

computationally more efficient for the minimum $s-t$ cut problem.

3.4.3 Illustrative Examples

Fig. 3.14 shows a typical run of our algorithm at increasing β values for the example shown in Fig. 3.12b. The set of terminals are identified as, $S = \{1, 6, 10\}$. Moreover, in this example it is easy to verify that the optimal cut set comprises of the edges $\{e_{2-9}, e_{3-4}, e_{7-8}\}$.

As seen in Fig. 3.14, at very low values of β , the association probability $p(i \in A_j) \equiv 1/3$ for every

Illustrative example of the proposed approach											
$\beta \approx 0$			$\beta = 0.2412$			$\beta = 0.9623$			$\beta = 38.55$		
1.0000	0	0	1.0000	0	0	1.0000	0	0	1.0000	0	0
0.3333	0.3333	0.3333	0.3918	0.3008	0.3074	0.6410	0.1405	0.2185	1.0000	0	0
0.3333	0.3333	0.3333	0.3939	0.3033	0.3028	0.6728	0.1751	0.1521	1.0000	0	0
0.3333	0.3333	0.3333	0.3335	0.3398	0.3267	0.3410	0.4414	0.2176	0	1.0000	0
0.3333	0.3333	0.3333	0.3055	0.3894	0.3051	0.2095	0.6044	0.1861	0	1.0000	0
0	1.0000	0	0	1.0000	0	0	1.0000	0	0	1.0000	0
0.3333	0.3333	0.3333	0.3033	0.3871	0.3095	0.1836	0.5560	0.2604	0	1.0000	0
0.3333	0.3333	0.3333	0.3012	0.3069	0.3918	0.1544	0.2046	0.6411	0	0	1.0000
0.3333	0.3333	0.3333	0.3073	0.3010	0.3917	0.2255	0.1463	0.6281	0	0	1.0000
0	0	1.0000	0	0	1.0000	0	0	1.0000	0	0	1.0000

Figure 3.14: Illustrative example of minimum 3-cut problem. The columns indicate the association matrices $p(i \in A_j)$ at different β values. The 'bold' numbers indicate the associations of terminals, i.e., $p(1 \in A_1) = 1, p(6 \in A_2) = 1$ and $p(10 \in A_3) = 1$. As the algorithm progresses, the associations of remaining nodes harden, thereby resulting in optimal cut.

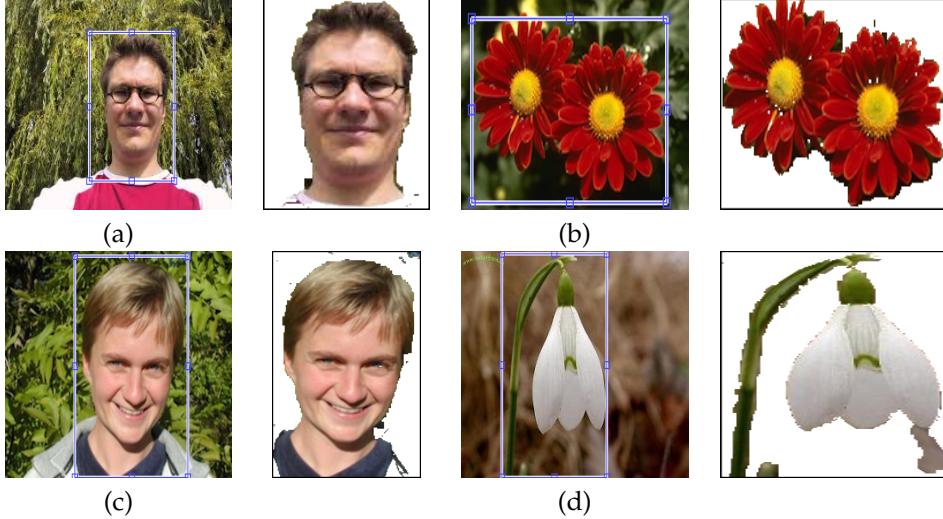


Figure 3.15: Illustration of the proposed multiway k -cut approach to background-foreground segmentation. Here each pixel in the bounding box represents a node. The terminal nodes (corresponding to foreground and background) are interactively provided by the user once, and the algorithm successively evaluates minimum st -cut to refine foreground-background segmentation. Edge-weights are obtained as functions of pixels' RGB values.

$i \notin S$, which in turn corresponds to maximizing randomness (Shannon entropy) of the solution. However, as the randomness is gradually decreased by increasing β , the probabilities start becoming *non-uniform* and exhibit preferential association to a specific cluster A_j . In the limiting case, i.e., at large values of β , the algorithm results in hardened probabilities (0 – 1 associations). Thus an optimal cut is obtained. Note that in this example, β is increased *geometrically* from 0.01 to 40, i.e., the algorithm provides for very fast β scheduling. We also verified the correctness and efficacy of our algorithm on polynomial time solvable minimum $s-t$ cut problem for 100 instances of 1000 nodes randomly generated graphs; our algorithm successfully obtained optimal solutions in each of these instances. As described earlier, the run-time was cubic in the number of vertices with average number of iterations per β value < 4 .

We also test our approach on very large graphs with number of nodes as large as ~ 25000 (corresponding to the size of the bounding box 150×160 pixels). Figure 3.15 shows the results of our implementation of the interactive foreground-background segmentation (GrabCut [70]) using the proposed MEP approach. The implementation results in effective segmentation.

Minimum Multiway Cut for a Non-Planar Graph

We now consider an instance of a 22-node *non planar* graph, shown in Fig. 3.16a. In this example we consider a 6-cut problem, whose set of terminals is specified as $S = \{1, 4, 8, 11, 15, 19\}$. Our MEP algorithm results in a partition of the underlying graph, given by $\{1, 2, 3\}, \{4, 5, 6, 7\}, \{8, 9, 10\}, \{11, 12, 13, 14\}$,

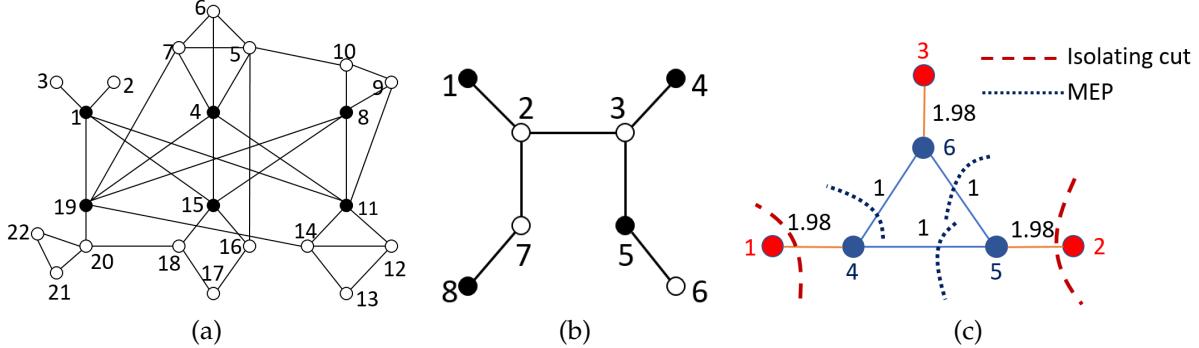


Figure 3.16: (a) Example of a 22 nodes non-planar graph with 6 terminals, $S = \{1, 4, 8, 11, 15, 19\}$. The proposed algorithm results in an optimal cut. (b) An 8 node graph with multiple permissible optimal 4-cuts. (c) Toy-example with $2k = 6$ vertices where k of the vertices form a cycle where each edge weight is equal to 1, and each other vertex is connected to exactly one vertex on the cycle with an edge weight of 1.98. The isolating cut heuristic [1] results in a cut value of 3.96, whereas our MEP-based algorithm results in optimal cut.

$\{15, 16, 17, 18\}$ and $\{19, 20, 21, 22\}$ with a cut value of 15; which is indeed optimal and can be easily verified. On the other hand, the isolating cut heuristic [1] for this randomly generated instance results in a cut solution with a value of 16. A similar observation is made on other randomly generated instances, where our algorithm results in optimal cut values (whenever verifiable). Moreover, the total run-time for the example in Fig. 3.16a for a naive implementation of the proposed approach in MATLAB is < 1s on an Intel i7-4790 CPU @ 3.60 GHz.

Non-unique Optimal Cuts

In both the examples described above, the resulting optimal cuts are indeed *unique*. We now therefore consider a scenario with more than one permissible optimal cuts. Our algorithm identifies the multiplicity of optimal cuts, which is reflected in the final association matrix $\{p_i \in A_j\}$. Fig. 3.16b shows an example of an 8-node graph with multiple permissible optimal 4-cuts. Executing the proposed MEP-based algorithm results in the following association matrix. As shown in Table 3.9, node 3 can be included in either A_2 or

$\{p_i \in A_1\}$	$\{p_i \in A_2\}$	$\{p_i \in A_3\}$	$\{p_i \in A_4\}$
1	0	0	0
1	0	0	0
0	0.5	0.5	0
0	1	0	0
0	0	1	0
0	0	1	0
0	0	0	1
0	0	0	1

Table 3.9: Final association matrix for the 4-cut problem shown in Fig. 3.16b

A_3 without affecting the value of the optimal cut, and therefore the association probabilities $p(3 \in A_2) = p(3 \in A_3) = 0.5$. Assigning node 3 to any of the two partitions still results in a feasible, yet optimal cut.

Comparison on Challenging Examples

Fig. 3.16c shows the performance of our MEP based approach on a toy-example, where heuristic such as isolating cut [1], fails to identify an optimal cut for the simplest such scenario. In fact, the graph in Fig. 3.16c can be generalized to any number of nodes, and the isolating cut heuristic results in a solution with approximation ratio of $2(1 - 1/k)$. However, our algorithm correctly finds the optimal multiway k -cut.

Chapter 4

Conclusions and Future Directions

In this thesis, we develop an efficient heuristic and a unifying framework for obtaining high quality solutions to a broad class of combinatorial optimization problems. The approach is derived from fundamental principles from the statistical physics literature, such as maximum entropy principle or minimum free-energy principle. Unlike other meta-heuristics, the proposed algorithm is independent of initialization, and the solution evolves deterministically as the algorithm progresses. A broad class of combinatorial optimization problems ranging from TSP and its variants, clustering non-linearly separable data, partitioning of electrical networks to solving hard problems on graphs, such as obtaining minimum multiway k -cut are considered. Several theoretical and practical tools are developed in the treatment of each of these problems.

In particular, we have reformulated problems from different application areas as equivalent resource allocation problems and used MEP-based approach to efficiently handle the combinatorial constraints. These problems include:

- *Traveling salesman problem and its variants:* It is shown that the proposed resource allocation framework can easily handle a large number of variants of classical TSP, including the CETSP where a feasible solution set admits a continuum of possible edges between the cities. Consequently it is shown that our MEP-based approach presents some viable opportunities as a heuristic for TSP and its variants. Because our approach is independent of the edges between nodes, it offers more flexibility to address these variants without compromising on computational complexity.
- *Clustering of power networks:* Our approach to this problem is a classical example of how a properly chosen metric, along with an efficient clustering algorithm, can potentially reduce the complexity of analyzing a large interconnected power network into analysis of small decoupled (loosely-coupled) networks. Unlike conventional metrics to graph clustering, such as degree of nodes, we have used the underlying physics of power systems in designing a suitable metric that captures sensitivity of bus voltages to other electrical quantities, such as reactive or active power perturbations. An MEP-based graph clustering algorithm is then employed to obtain natural decoupled sub-networks. This decou-

pling is then exploited to implement local (decentralized) controllers for maintaining bus voltages within the permissible limits over the entire network. The performance of the proposed clustering methodology, along with the supervisory rule-based voltage control scheme is benchmarked on standard IEEE networks and exhibits state-of-the-art results in terms of overcoming faults/perturbations within a power network.

- *Clustering nonlinearly separable data:* One of the earliest application domains of MEP-inspired approaches is clustering of linearly separable data points. However, these formulations do not address cluster analysis for nonlinearly separable data. We address this issue by marrying machine learning (kernel-trick) to statistical physics (deterministic annealing). We show that the kernel trick efficiently reduces clustering of nonlinearly separable data points in the input domain to clustering in abstract higher-dimensional spaces where these data points are linearly separable. In doing so, we do not require eigenvector computations that are often computationally prohibitive (as is the case with spectral clustering algorithms), while still working with the objective functions of spectral clustering algorithms through appropriately chosen kernel matrices. Constraints, such as, must-link or cannot-link are easily addressed in the proposed framework. Results on benchmark and synthetic datasets demonstrate the effectiveness of the proposed approach.
- *Multiway k -cut:* It is further shown that several hard-problems on graphs are easily handled using the proposed MEP-based framework. In particular, we consider the minimum multiway k -cut problem, which is APX-hard for $k \geq 3$. We also describe a convergence proof for the proposed algorithm and show that the fixed-point iteration scheme converges. Moreover, the proposed algorithm is computationally efficient than the Dinic's or Edmonds-Karp algorithm for the minimum $s - t$ cut problem.

Many possible future directions exist in extending the proposed approach to even broader class of combinatorial optimization problems. One of the problems of interest here that we have already started looking into is a general mixed binary integer linear program (MBILP). An MBILP is mathematically described as:

$$\begin{aligned} & \min_{\substack{\mathbf{x} \in \mathbb{R}^n \\ \mathbf{z} \in \{0,1\}^m}} \mathbf{a}^T \mathbf{x} + \mathbf{b}^T \mathbf{z}, \\ & \text{subject to, } G\mathbf{x} + H\mathbf{z} \leq c, \text{ and } \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{4.0.1}$$

where $G \in \mathbb{R}^{k \times n}$ and $H \in \mathbb{R}^{k \times m}$. MBILPs occur in variety of domains, such as scheduling, production planning, telecommunication networks, unit commitment and economic dispatch, and cellular networks. MBILPs are NP-complete and in some instances, the optimal integral solutions are far from the optimal

solutions to corresponding relaxed linear programs (LPs), which are solvable in polynomial times. Our approach to MBILPs is to replace the binary variables with their continuous counterparts - i.e., with probabilities of them being one and then use Shannon-entropy to constrain the probabilities to maintain maximum randomness to the solutions (i.e., making least restrictive assumptions in the nature of final solutions). The Shannon entropy term is then gradually relaxed in an annealing process to obtain hard (0-1) solutions to the binary variables. Thus instead of directly solving an MBILP, we work on a series of LPs that are easily solvable in polynomial times, thereby significantly reducing the computational complexity of solving the original MBILP. In our initial evaluations using this approach, we seem to be arriving at solutions to randomly generated instances with 2000 binary variables within 2-3% of the optimal solutions and our naive MATLAB implementation takes only a little over a minute on an average.

One possible shortcoming of the proposed MEP-based approach is the lack of parallel implementation in its current form. This is primarily due to the functional form of Gibbs distribution which requires computation of distances of facility locations from each data point. While number of such computations are significantly less with the MEP based approach ($O(NK)$ instead of $O(N^2)$, where N, K are number of data points and facilities, respectively), the algorithm lacks parallelization capability of k -means. There remains significant opportunities to optimize the code implementation of this framework to achieve more favorable computation times, at which point this algorithm can be run on benchmark mTSP and multiway k -cut cases and compared against many of the conventional heuristics.

Other immediate extensions of the proposed algorithm are to consider other related (to minimum multiway k -cut) problems on graphs, such as, finding maximum independent set or graph coloring. The quadratic cost function in minimum multiway k -cut can be easily modified to incorporate problem specific objectives.

While it appears to be a really difficult problem to obtain sub-optimality guarantees with MEP-based algorithms, it would be worthwhile to consider them as well as part of the future directions. Obtaining performance bounds for statistical physics based approaches has been an open problem since its inception and guarantees are provided only for a few special cases, such as, linear assignment problems [71]. Though only little is known in terms of sub-optimality guarantees, it must be noted that these algorithms work very well in practice and are empirically shown to converge to very high-quality optima.

Part II

Grid and Microgrid of the Future: Robust Control Framework

Chapter 5

Introduction: Grid and Microgrid of the Future

5.1 Historical Overview

The North American electrical grid is regarded as the most significant engineering achievement of the 20th century [41], however, the infrastructure that defines the U.S. electric grid is based largely on pre-digital technologies and is ill-equipped to serve smaller, innovative solar or wind facilities. Throughout its evolution, the utility grid mainly relied primarily on centralized power plants and developed protocols to provide system reliability based on that model. On the other hand, the International Energy Outlook 2013 (IEO2013) [72] projects that world energy consumption will grow by 56 percent between 2010 and 2040 (see Fig. 5.1). The study examines the potential role of photo-voltaics (PVs), wind turbines and other distributed energy resources (DERs) in meeting the growing energy demands while achieving security of supply and minimizing carbon dioxide emissions. However, the increasing use of renewable generation and distributed energy resources (DER), such as residential solar and home energy storage, along with customers' changing energy usage patterns lead to greater uncertainty and variability in the electric grid. New tools are required to create a flexible and modern electric grid that can meet this increase in renewable generation and DERs, while providing the quality of service, resiliency, and reliability that customers expect.

In this regard, *microgrids* are hypothesized as viable alternatives for supporting a flexible and efficient electric grid by enabling the integration of growing deployments of distributed energy resources such as renewables like solar and wind. In addition, the use of local sources of energy to serve local loads helps reduce energy losses in transmission and distribution, further increasing efficiency of the electric delivery system. Microgrids are localized grid systems that are capable of operating in parallel with, or independently from, the existing traditional grid [73]. Fig. 5.2 shows a schematic of a microgrid with multiple DC sources providing power for AC loads. Existing control architectures for traditional grids, which are designed for relatively large conventional sources (power plants) of predictable and dispatchable electric power, cannot adequately manage uncertain power sources such as solar or wind generations. Limited

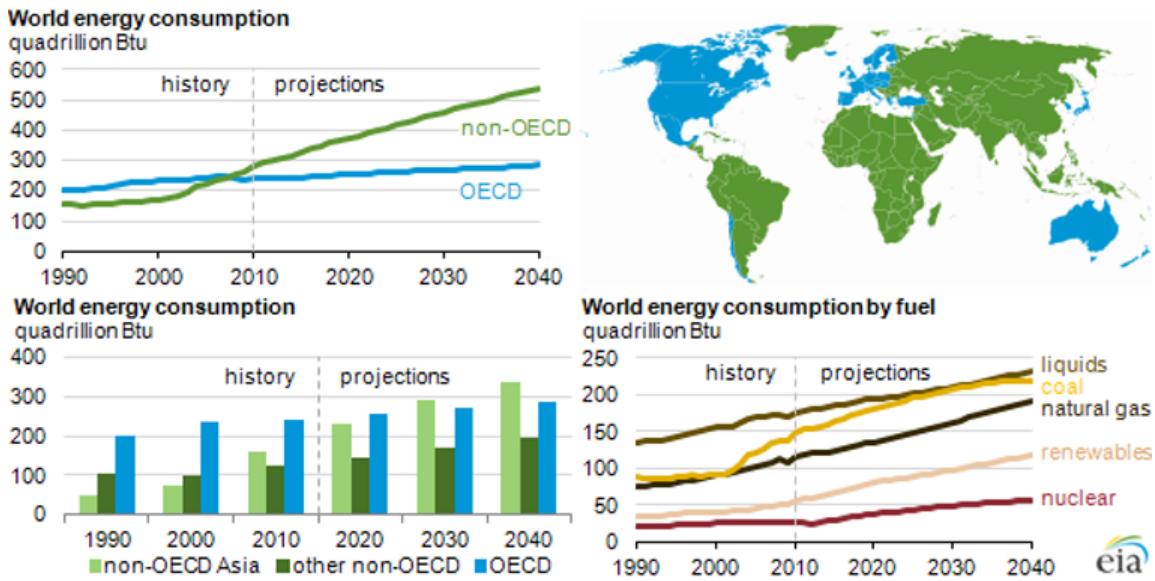


Figure 5.1: Energy consumption history and projection. (Source: U.S. Energy Information Administration, International Energy Outlook 2013.)

predictability with such resources result in intermittent power generation; moreover time-varying loads, practicability and economics factors pose additional challenges in efficient operation of microgrids. Thus it is required to develop efficient distributed control technologies for reliable operation of smart microgrids [74].

In such micro grids, multiple DC power sources connected in parallel, each interfaced with *DC-DC converter*, provide power at their common output, the DC-link, at a regulated voltage; this power can directly feed DC loads or be used by a *DC-AC inverter* to interface with AC loads (see Fig. 5.2). Both converters and inverters are switched-mode power electronic devices. By appropriately controlling the switch duty-cycle of these power-electronic converters at each power source, it becomes feasible to manipulate electrical quantities such as the power output by each power source, voltage at the DC-link, power transactions between utility grid and localized power sources (in grid-tied mode), and voltage and frequency at the AC-link (also known as point of common coupling (PCC)).

The main challenges arise from the uncertainties in the size and the schedules of loads, the complexity of a coupled multi-converter network, the uncertainties in the model parameters at each converter, and the adverse effects of interfacing DC power sources with AC loads, such as the 120 Hz ripple that has to be provided by the DC sources. In view of these challenges, a robust and distributed control technology is needed for reliable operation of smart microgrids. In the multiple-input multiple-output setting necessitated by the need to control multiple generation sources, it is difficult to address robustness and

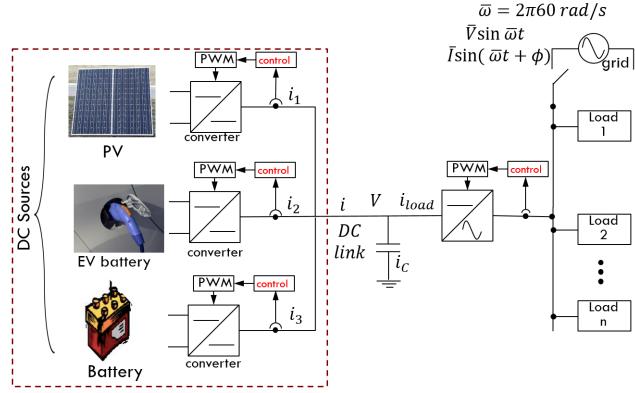


Figure 5.2: A schematic of a microgrid. An array of DC sources provide power for AC loads. Power sources provide power at DC-link, their common output bus, at a voltage that is regulated to a set-point. The control system at the respective DC-DC converter that interfaces with a source is responsible for regulating the voltage at the DC-link.

performance criteria in the conventional PID-based control synthesis framework. Recently robust and optimal control methodologies have received attention. In [75], a linear-matrix-inequality (LMI) based robust control design is presented for boost converters which demonstrates significant improvements in voltage regulation over PID based control designs. In [76, 77, 78] robust \mathcal{H}_∞ control framework is employed in the context of inverter systems. While the issue of current sharing is extensively studied (see [79] and [80]), most prior methods reported assume a single power source. Furthermore, the sharing requirements can be time-varying and are often dictated by the availability and relative costs of different power sources; for instance, economic considerations can dictate that power provided by the sources should be in a certain proportion or according to a prescribed priority (e.g. PV provides the maximum power it can to satisfy load demand, and the deficit is provided by battery).

These control objectives can be summarized as: (a) regulating voltage at the DC-link with guaranteed robustness, (b) prescribed time-varying power sharing in a network of parallel converters, (c) controlling the trade-off between 120 Hz ripple on the total current provided by the power sources and the ripple on the DC-link voltage, (d) regulating voltage and frequency at the PCC for islanded microgrids, and (e) power sharing among multiple microgrids interfaced at the PCC. While the control designs for the DC and the AC side can be decoupled, a systematic control design that addresses all the above objectives simultaneously is missing. Furthermore, if multiple converter units are interfaced together, it often becomes infeasible to analyze the performance and stability of the closed-loop system.

5.2 Main Contributions

This thesis develops a comprehensive approach that addresses the challenges to system reliability and power quality presented by widespread renewable power generation. By developing techniques for both centralized cloud-based and distributed peer-to-peer networks, the proposed control system enables co-ordinated response of many local units to adjust consumption and generation of energy, satisfy physical constraints, and provide ancillary services requested by a grid operator. The proposed work uses tools from robust control theory [81, 82] simultaneously addresses the aforementioned control objectives pertaining to managing multiple generation sources and regulating electrical quantities of interest. Our main contributions can be summarized as follows:

- (a) *Robust regulation and sharing performance*: We identify appropriate maps of converter/inverter duty-cycles to facilitate a common framework for analyzing and synthesizing controllers for different types of converters while rendering models that are linear. Modern robust control tools are employed to address multiple objectives that include regulation of the DC-link voltage to a desired set-point reference, and a prescribed sharing of power among different DERs. Apart from meeting performance guarantees including prioritization, our control design also addresses the challenges of interfacing AC loads, including the 120 Hz ripple that has to be provided by the DC sources.
- (b) *Modular and structured architecture*: The control architecture presented in this work is modular and facilitates plug and play operation. Here, new converter module can be added or removed from the network, without any need to redesign controllers and without compromising the voltage regulation performance of the network. Furthermore, adding a module, which is agnostic to sharing ratios of other modules, to the networked system does not affect the overall performance of the networked system. The intramodule and inter-module control is structured in such a way that it allows easy multi-converter network analysis and synthesis. In the framework developed, the network of parallel converters can be analyzed, and the corresponding control systems synthesized, in terms of an *equivalent single-converter system*.
- (c) *Robustness to communication uncertainties*: The synthesis procedure results in a single controller which functions for the entire range of communication capabilities; from decentralized to centralized. Here, it guarantees precise regulation of the DC-link voltage and power sharing specifications when communication allows for a centralized operation, and meets gracefully degraded specifications with lessened communication capabilities among converters.

5.3 Organization

The organization of this part of the thesis is as follows. In chapter 6, we describe cycle averaged dynamics of power electronic converters. Through an appropriate choice of maps between converter duty-cycles and control signals, the dynamics of converters can be made linear. We then introduce a robust control theoretic framework for controlling a network of multiple converter units in chapter 7. In chapter 8, we demonstrate the effectiveness of the proposed control design through some practical simulation scenarios. These scenarios consider DERs of different types and highly uncertain loads. We then conclude our discussion with practical experimental scenarios in chapter 9.

Chapter 6

Facilitating Linear Models of Power Electronic Converters

In this chapter, we describe amenable dynamical models of power electronic converters. These dynamical models are essential to control design approach described in chapter 7. Below we discuss these cycle-averaged dynamical models for both DC-DC converters as well as DC-AC inverter. In fact, we identify appropriate maps of converters/inverters duty-cycles to facilitate a common framework for analyzing and synthesizing controllers for different topologies of power electronic converters while rendering models that are linear.

6.1 Modeling of DC-DC Converters

A DC-DC converter is an electronic circuit that converts a source of direct current from one voltage level to another. These converters belong a class of switched-mode power electronics, where semiconductor based high-frequency switching mechanism connected to a DC power source enables changing voltage and current characteristics at its output [83, 84]. Fig. 6.1 shows the schematic of common DC-DC converter topologies. In essence, a DC-DC converter comprises of a power switch, a diode, an inductor and a capacitor. These can be arranged in a variety of ways to realize *boost*, *buck*, and *buck-boost* converters. Below

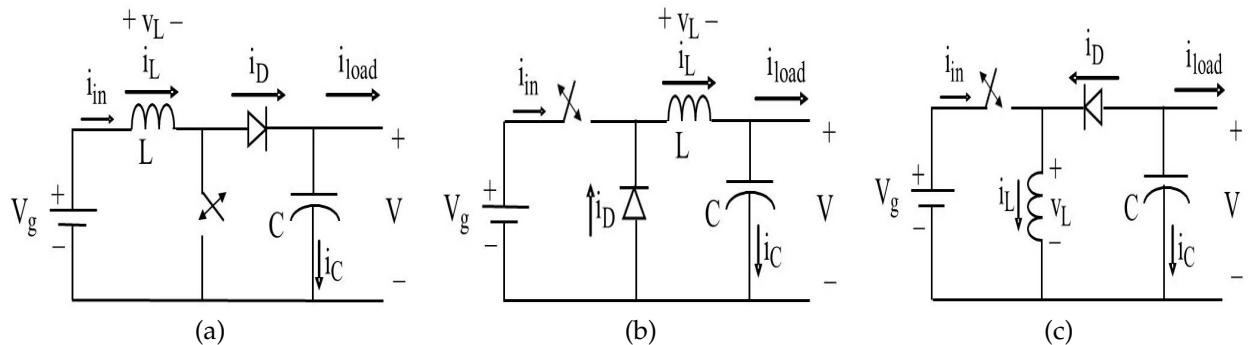


Figure 6.1: Circuit representing (a) Boost converter, (b) Buck converter, and (c) Buck-Boost converter. Note that i_{load} includes both the nominal load current, as well as ripple current. The converters are assumed to operate in continuous-conduction-mode (CCM). Boost converters step up the voltage at the output, while buck converters step down the voltage. A buck-boost converter can achieve both the objectives.

we describe the dynamic model of each converter topology for signals (inductor current, capacitor voltage) that are averaged over a switched cycle.

Notations: We use $V_g(t)$ and $V(t)$ to represent voltages at input and output, respectively. L and C denote the converter inductance and DC-link capacitance. i_L and i_{load} denote the inductor and load currents respectively. V_{ref} denotes the desired reference voltage at the DC-link. Quantities $d(t)$ and $d'(t) \triangleq 1 - d(t)$ denote the instantaneous *duty-cycle* and *complementary duty-cycle* of semiconductor switches respectively, while D and D' denote the corresponding steady-state quantities.

6.1.1 Boost converter dynamics

A boost converter (also known as *step-up* converter), steps up the input voltage V_g so that the output voltage V is higher than the input (see Fig. 6.1a). The dynamical equations of a boost converter are described as:

$$\begin{aligned} L \frac{di_L}{dt} &= V_g - V \\ C \frac{dV}{dt} &= i_L - i_{\text{load}}, \end{aligned} \quad (\text{when switch is OFF}) \quad (6.1.1)$$

$$\begin{aligned} L \frac{di_L}{dt} &= V_g \\ C \frac{dV}{dt} &= -i_{\text{load}}. \end{aligned} \quad (\text{when switch is ON}) \quad (6.1.2)$$

If the switch is ON for $d(t)$ proportion of time during a cycle, then the averaged dynamic model of a boost converter (averaged over one-cycle) is given by:

$$\begin{aligned} L \frac{d \langle i_L(t) \rangle}{dt} &= -(1 - d(t)) \langle V(t) \rangle + V_g \\ C \frac{d \langle V(t) \rangle}{dt} &= (1 - d(t)) \langle i_L(t) \rangle - \langle i_{\text{load}}(t) \rangle, \end{aligned} \quad (6.1.3)$$

where $\langle \cdot \rangle$ represents average over a switching cycle. Note that the switching occurs at a very fast time-scale as compared to converter current-voltage dynamics. We use slight abuse of notation to denote the averaged current and voltages by i_L and V , respectively. The

$$L \frac{di_L(t)}{dt} = \underbrace{V_g - d'(t)V(t)}_{\tilde{u}(t) := V_g - u(t)} \quad (6.1.4)$$

$$C \frac{dV(t)}{dt} = \underbrace{(D' + \hat{d}(t))}_{\approx D'} i_L(t) - i_{\text{load}}(t), \quad (6.1.5)$$

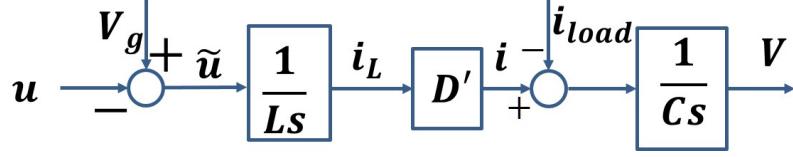


Figure 6.2: Block diagram representation of a boost-type converter. The control signal \tilde{u} is converted to an equivalent PWM signal to command the gate of the transistor acting as a switch.

where $D' := (V_g/V_{\text{ref}})$ is the steady-state complementary duty-cycle. Note that $\hat{d}(t) = d'(t) - D'$ is typically very small, and therefore allows for a linear approximation around the nominal duty-cycle, $D = 1 - D'$.

The corresponding block-diagram representation of the above set of equations is shown in Fig. 6.2. The control objectives are to design u (equivalently \tilde{u}) such that the voltage regulation error $V - V_{\text{ref}}$ is made small irrespective of load disturbances i_{load} and variations in parameters L and C . Note that the equivalent duty-cycle $d(t)$ can be obtained from \tilde{u} via $d(t) = 1 - \frac{V_g - \tilde{u}(t)}{V(t)}$.

6.1.2 Buck converter dynamics

Fig. 6.1b shows the schematic of a buck converter (*step-down* converter). A buck converter steps down the voltage at its output. Proceeding as before, the averaged dynamics of a buck converter can be derived as:

$$\boxed{\begin{aligned} L \frac{di_L(t)}{dt} &= \underbrace{d(t)V_g - V(t)}_{\tilde{u}(t)} \\ C \frac{dV(t)}{dt} &= i_L(t) - i_{\text{load}}(t). \end{aligned}} \quad (6.1.6)$$

6.1.3 Buck-boost converter dynamics

As the name suggests, a buck-boost converter can both step-up or step-down the voltage at its output. The averaged dynamical equations for a buck-boost converter are given as:

$$\boxed{\begin{aligned} L \frac{di_L(t)}{dt} &= \underbrace{d(t)V_g + (1 - d(t))V(t)}_{\tilde{u}(t)} \\ C \frac{dV(t)}{dt} &= -D'i_L(t) - i_{\text{load}}(t). \end{aligned}} \quad (6.1.7)$$

Note: An important consequence of the above modeling approach is the equivalence among dynamical models of multiple converter topologies. In fact, the dynamic models of DC-DC converters can be

succinctly written as:

$$\boxed{\begin{aligned} L \frac{di_L(t)}{dt} &= \tilde{u}(t) \\ C \frac{dV(t)}{dt} &= \alpha i_L(t) - i_{\text{load}}(t), \end{aligned}} \quad (6.1.8)$$

where corresponding duty-cycles $d(t)$ and α are as shown in Table 6.1:

Converter topology	Duty-cycle $d(t)$	α
Boost converter	$\left(1 - \frac{V_g - \tilde{u}(t)}{V(t)}\right)$	D'
Buck converter	$\left(\frac{\tilde{u}(t) + V(t)}{V_g}\right)$	1
Buck-boost converter	$\left(\frac{\tilde{u}(t) - V(t)}{V_g - V(t)}\right)$	$-D'$

Table 6.1: Generalized models for multiple converter topologies

6.2 Modeling of DC-AC Inverter

Unlike DC-DC converters, DC-AC inverters transform a source of direct current to an equivalent source of alternating current using semiconductor switches. Fig. 6.3a shows the schematic of a *full-bridge* DC-AC inverter. A full-bridge inverter [83] comprises of two legs each containing two switches each - (a) s_1 and s_2 , (b) s_3 and s_4 . The full-bridge inverter is interfaced with the AC-side load through an interface reactor represented by a series RL branch. L and R respectively, represent the inductance and internal resistance of the interface reactor. The interface reactor acts as a low pass filter and ensures low-ripple AC-side current i_L resulting from switching operations. The voltage at terminal a is controlled by periodically switching ON/OFF the switches s_1 and s_2 . Similarly, switches s_3 and s_4 control the voltage at terminal b . The quantities $V(t)$ and $i_{\text{load}}(t)$ represent the AC-side voltage (or output voltage), and load current, respectively. Let $d_a(t)$ represent the proportion of ON time of switch s_1 (or OFF time of switch s_2), also known as the duty-cycle of switch s_1 . Therefore the average voltage at terminal a , $V_a(t)$ is given by $V_a(t) = d_a(t)V_{\text{dc}}$, where V_{dc} is the voltage of the DC source. Similarly, the voltage V_b at terminal b , averaged over switching cycles, is given by $V_b(t) = d_b(t)V_{\text{dc}}$. By combining the two states of operation, dynamic model

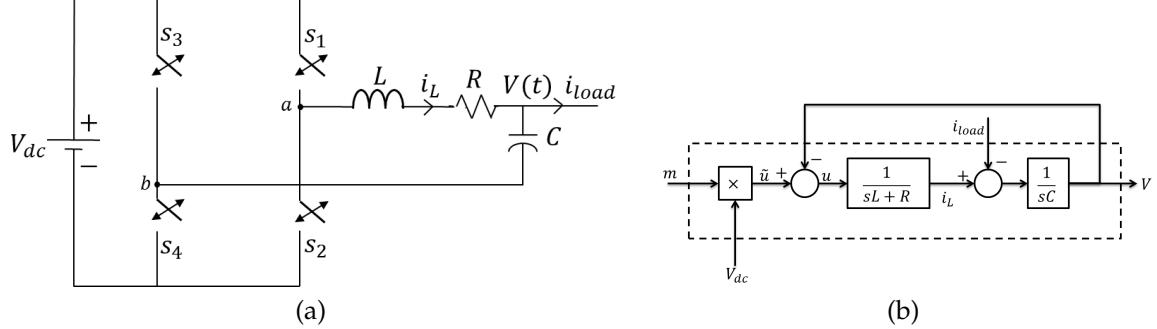


Figure 6.3: (a) Circuit representing a full-bridge inverter. The AC-side current is given by i_{load} . The switches s_1, s_2, s_3 and s_4 control the AC-side voltage $V(t)$. (b) Associated control model of the full-bridge inverter.

of a full-bridge inverter (averaged over switching cycles) is given as:

$$\boxed{\begin{aligned} L \frac{di_L(t)}{dt} + Ri_L(t) &= \underbrace{(d_a(t) - d_b(t)) V_{dc} - V(t)}_{\tilde{u}(t) := m(t)V_{dc} - V(t)} \\ C \frac{dV(t)}{dt} &= i_L(t) - i_{load}(t). \end{aligned}} \quad (6.2.1)$$

The average voltage between terminals a and b , $\tilde{u}(t) := m(t)V_{dc}$ is proportional to, and can be controlled by the modulating signal $m(t) \in [-1, 1]$. Fig. 6.3b shows a control block diagram of the system described by (6.2.1), for which the next chapter presents a closed-loop structure to regulate $V(t)$ at its reference value. Note that for a given value of modulating signal $m(t) := d_a(t) - d_b(t)$, there are infinitely many choices for the duty-cycles $d_a(t)$ and $d_b(t)$. This issue of non-uniqueness is addressed by considering the following scheme:

	$m(t) \geq 0$	$m(t) < 0$
$d_a(t)$	$m(t)$	0
$d_b(t)$	0	$-m(t)$

An important contribution of this thesis is that it introduces a control architecture for a paralleled network of inverters for which performance analysis becomes feasible. The proposed architecture is scalable and extends to any number of inverters in the network. In the next chapter, we discuss the control design scheme of a network of multiple inverters.

Chapter 7

Robust Control Framework for Power Electronic Converters

In this chapter, we develop a comprehensive control approach that addresses the challenges to system reliability and power quality presented by widespread renewable power generation. By developing techniques for both centralized cloud-based and distributed peer-to-peer networks, the proposed control design enables coordinated response of many local units to adjust consumption and generation of energy, satisfy physical constraints, and provide ancillary services requested by a grid operator. We apply concepts from nonlinear and robust control theory to design self-organizing power systems that effectively respond to the grid events and variability. A key feature enabled by the proposed methodology is a flexible plug-and-play architecture wherein devices and small power networks can easily engage or disengage from other power networks or the grid. The main idea is to apply concepts from robust control theory to design coordination-friendly power systems that effectively respond to the grid events and variability. The control architecture is aimed to help with (a) electrical requirements at the PCC, (b) integrating communication among power-electronic converters, and (c) performance analysis of microgrids. The proposed control architecture addresses these objectives by formulating the control problems in a *disturbance rejection* framework, developing a unified architecture for both centralized and decentralized implementations, and posing the voltage regulation problem in a robust optimal control framework. Below we discuss control design approach for both DC-DC converters, as well as DC-AC inverters.

7.1 Control of DC-DC Converters

7.1.1 Control of Single Converter

In this section, we describe the *inner-outer* control design for a single boost converter system. This design for a single converter forms the basis for the analysis and design of control architecture for multiple converters presented in Sec. 7.1.2. While the design is easily extendable to include other converter types such as buck and buck-boost, the discussion has been confined to boost converters only for the sake of brevity. Note that in the proposed control architecture (see Fig. 7.1), the inputs to outer feedback controller include i_{ref}

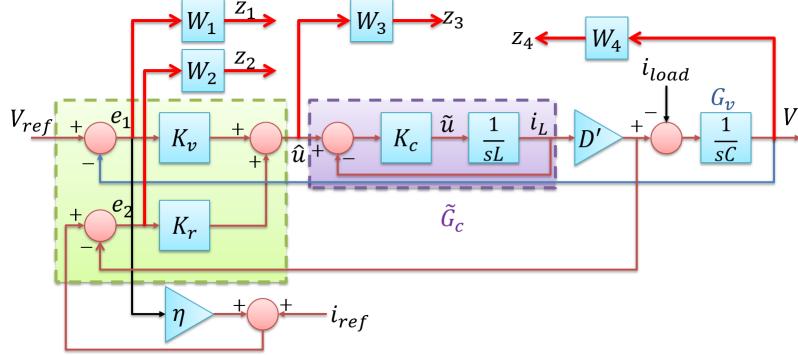


Figure 7.1: Block diagram representation of the inner-outer control design. The *regulated variables* z_1, z_2, z_3 and z_4 correspond to weighted - (a) tracking error in DC-link voltage, (b) mismatch between i_{ref} and i_{load} , (c) control effort \hat{u} , and (d) output voltage tracking, respectively.

in addition to the typical V_{ref} and the measured DC-link voltage V . The requirements on current sharing are imposed through this additional i_{ref} signal (explained in Sec. 7.1.2) by setting i_{ref} to measured (or communicated) load current i_{load} in the centralized case, and setting i_{ref} to estimated (or prespecified) signals in the decentralized case. Below we describe the proposed inner-outer control architecture [85, 86].

Design of the inner-loop controller

The main objective for designing the inner-loop controller K_c is to decide the trade-off between the 120 Hz ripple on the capacitor current i_C (equivalently on the output voltage V) and the inductor current i_L of the converter. The 120 Hz ripple arises as the AC-side instantaneous power pulsates at 120 Hz (see Fig. 7.2). When the DC-link is interfaced with AC-link, the 120 Hz component is eventually supplied by the DC sources. DC sources often may not sustain large amount of high-frequency fluctuations in the sourced currents and thus a trade-off must be made between the 120 Hz ripple on the capacitor current i_C and the inductor current i_L of the converter. Accordingly K_c is designed such that the inner-closed loop plant \tilde{G}_c assumes a low-order transfer function, i.e., we choose K_c such that

$$\tilde{G}_c(s) = \left(\frac{\tilde{\omega}}{s + \tilde{\omega}} \right) \left(\frac{s^2 + 2\zeta_1\omega_0 s + \omega_0^2}{s^2 + 2\zeta_2\omega_0 s + \omega_0^2} \right), \quad (7.1.1)$$

where $\omega_0 = 2\pi 120 \text{ rad/s}$. $\tilde{\omega}, \zeta_1$ and ζ_2 are design parameters. Here the parameter $\tilde{\omega} > \omega_0$ is simply chosen to implement a low-pass filter that attenuates undesirable frequency content in i_L beyond $\tilde{\omega}$. $\tilde{\omega}$ is chosen large enough to provide good steady-state tracking, however, it should be small enough to not allow the switching frequency ripples to affect the output. The ratio ζ_1/ζ_2 determines the size of the notch in \tilde{G}_c at ω_0 (see Fig. 7.3). Lower values of this ratio correspond to larger notches, which in turn imply smaller 120

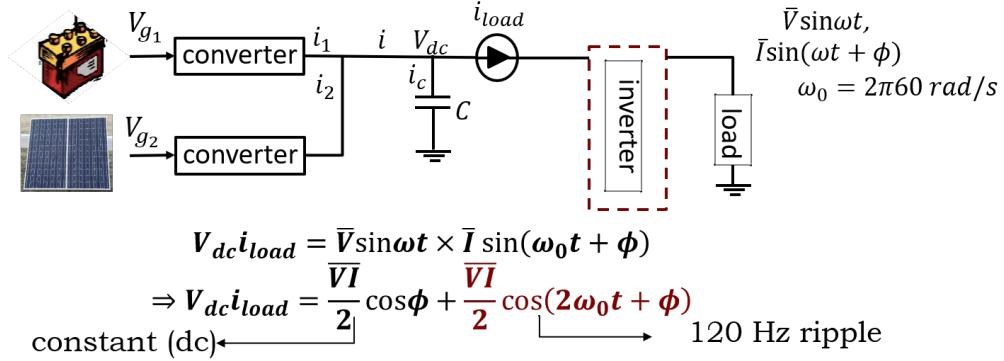


Figure 7.2: Existence of 120 Hz ripple in interfacing DC-link with AC-side. The AC-side power pulsates at 120 Hz and needs to be accounted for by the DC sources.

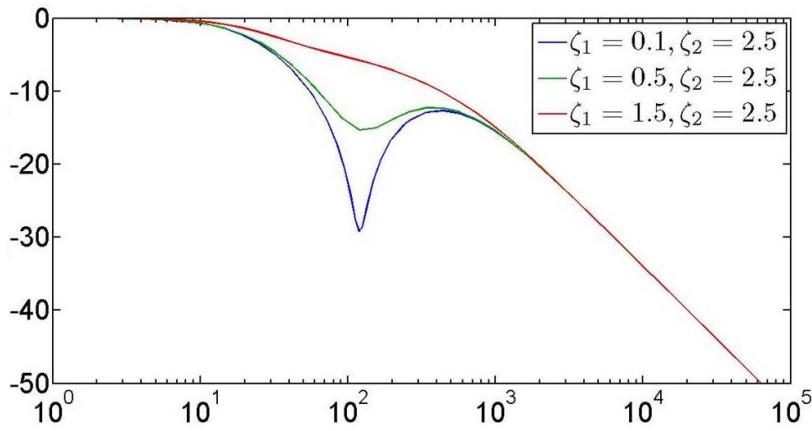


Figure 7.3: Bode magnitude plots of the closed-loop plant \tilde{G}_c for various ζ_1 values. $\tilde{\omega}$ is chosen to be 600π rad/s. Note that a relatively larger value of $\tilde{\omega}$ is in accordance with choosing a fast inner-current controller.

Hz component in i_L , since \tilde{G}_c represents the inner closed-loop transfer function from reference current to i_L . Furthermore since $i_C = i_{load} - i_L$, this in turn implies higher ripples in i_C . Thus the ratio ζ_1/ζ_2 can be appropriately designed to achieve a specified tradeoff between 120 Hz ripple on i_C and i_L . The stabilizing second-order controller K_c that yields the above closed-loop plant \tilde{G}_c is given by,

$$\tilde{K}_c(s) = L\tilde{\omega} \frac{(s^2 + 2\zeta_1\omega_0 s + \omega_0^2)}{(s^2 + 2\zeta_2\omega_0 s + 2(\zeta_2 - \zeta_1)\omega_0\tilde{\omega} + \omega_0^2)}, \quad (7.1.2)$$

which is again a low-order (second-order) controller design.

Design of the outer-loop controller

Let $G_c := 1/sL$ and $G_v := 1/sC$ denote the inner (current) and outer (voltage) open-loop plant transfer functions of the dynamics of a boost converter. For a given choice of K_c , Fig. 7.1 shows the block-diagram

representation of the proposed inner-outer control design. \tilde{G}_c represents the inner-shaped plant as before. The outer controllers K_v and K_r are designed to regulate the output DC voltage V to the desired reference voltage V_{ref} and the output current $D'i_L$ to the reference current i_{ref} , respectively. Note that from (6.1.4), $D'i_L$ is equal to i_{load} in the steady-state. The augmentation of controller K_r forms the basis for time-varying power sharing and is explained in the section on control of multiple converters. It should be remarked that the proposed design has a feature that if the load current measurement is available, i.e., $i_{\text{ref}} = i_{\text{load}}$, then the steady-state DC output voltage is maintained at V_{ref} . However in the absence of i_{load} measurement, the outer controller K_r regulates the output current $D'i_L$ to $i_{\text{ref}} \neq i_{\text{load}}$ resulting in an output voltage $V \neq V_{\text{ref}}$. The mismatch in voltage tracking is captured by a pre-specified *droop*-like coefficient η in a controlled manner, the notable difference here being the application of droop to the *faster* current loop when compared with the conventional droop-based design acting on the *slower* voltage loop. This feature is mathematically quantified in the following discussion on the proposed control design.

The main objective for the design of the controllers K_v and K_r is to make the tracking errors small and simultaneously attenuate measurement noise to achieve high resolution. This is achieved by posing a *model-based* multi-objective optimization problem, where the required objectives are described in terms of norms of the corresponding transfer functions, as described below. Note that the regulated variables z_1, z_2, z_3 and z_4 correspond to weighted - (a) tracking error in DC-link voltage, (b) mismatch between i_{ref} and i_{load} , (c) control effort \hat{u} , and (d) output voltage tracking, respectively. From Fig. 7.1, the transfer function from exogenous inputs and auxiliary control input $w = [V_{\text{ref}}, i_{\text{ref}}, i_{\text{load}}, \hat{u}]^T$ to regulated output $z = [z_1, z_2, z_3, z_4, e_1, e_2]$ is given by:

$$\underbrace{\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ e_1 \\ e_2 \end{bmatrix}}_z = \underbrace{\begin{bmatrix} W_1 & 0 & W_1 G_v & -D' W_1 G_v \tilde{G}_c \\ \eta W_2 & W_2 & \eta W_2 G_v & -D'(1 + \eta G_v) W_2 \tilde{G}_c \\ 0 & 0 & 0 & W_3 \\ 0 & 0 & -W_4 G_v & D' W_4 G_v \tilde{G}_c \\ 1 & 0 & G_v & -D' G_v \tilde{G}_c \\ \eta & 1 & \eta G_v & -D'(1 + \eta G_v) \tilde{G}_c \end{bmatrix}}_{T_{wz}} \underbrace{\begin{bmatrix} V_{\text{ref}} \\ i_{\text{ref}} \\ i_{\text{load}} \\ \hat{u} \end{bmatrix}}_w. \quad (7.1.3)$$

The optimization problem is to find stabilizing controllers $K_{\text{outer}} = [K_v, K_r]^T$ such that the \mathcal{H}_{∞} -norm of the above transfer function from w to z is minimized. Here the weights W_1, W_2, W_3 and W_4 are chosen to reflect the design specifications of robustness to parametric uncertainties, tracking bandwidth, and saturation limits on the control signal [81, 82]. More specifically, the weight functions $W_1(j\omega)$ and $W_2(j\omega)$ are chosen to be large in frequency range $[0, \omega_{BW}]$ to ensure small tracking errors $e_1 = V_{\text{ref}} - V$ and $e_2 = i_{\text{ref}} + \eta e_1 - D'i_L$ in this frequency range. The design of weight function $W_3(j\omega)$ entails ensuring that the control effort lies

within saturation limits. The weight function W_4 is designed as a high-pass filter to ensure that the transfer function from i_{load} to V is small at high frequencies to provide mitigation to measurement noise.

Note that for the system shown in Fig. 7.1, the voltage V at the DC-link is given by:

$$V = G_v (-i_{\text{load}} + D' \tilde{G}_c (K_v e_1 + K_r e_2)). \quad (7.1.4)$$

Using the fact that $e_1 = V_{\text{ref}} - V$ and $e_2 = i_{\text{ref}} + \eta e_1 - D' \tilde{G}_c (K_v e_1 + K_r e_2)$, the DC-link voltage in terms of exogenous quantities V_{ref} , i_{ref} and i_{load} is given by:

$$\begin{aligned} V(s) &= \left[\frac{D' \tilde{G}_c G_v (K_v + \eta K_r)}{1 + D' \tilde{G}_c K_r + D' \tilde{G}_c G_v (K_v + \eta K_r)} \right] V_{\text{ref}}(s) \\ &+ \left[\frac{D' \tilde{G}_c G_v K_r}{1 + D' \tilde{G}_c K_r + D' \tilde{G}_c G_v (K_v + \eta K_r)} \right] (i_{\text{ref}}(s) - i_{\text{load}}(s)) - \left[\frac{G_v}{1 + D' \tilde{G}_c K_r + D' \tilde{G}_c G_v (K_v + \eta K_r)} \right] i_{\text{load}}(s). \end{aligned} \quad (7.1.5)$$

Let $S(s)$, $T_{V_{\text{ref}}V}$ and $T_{i_{\text{ref}}V}$ denote the closed-loop sensitivity transfer function and complementary sensitivity transfer functions from V_{ref} to V and i_{ref} to V , respectively. Then (7.1.5) can be rewritten as:

$$V(s) = T_{V_{\text{ref}}V} V_{\text{ref}}(s) + T_{i_{\text{ref}}V} (i_{\text{ref}}(s) - i_{\text{load}}(s)) - G_v S i_{\text{load}}(s).$$

The DC gains of above closed-loop transfer functions are given by (since $G_v = 1/sC$ has an infinite DC gain),

$$|T_{V_{\text{ref}}V}(j0)| = 1, \quad |T_{i_{\text{ref}}V}(j0)| = \frac{|K_r(j0)|}{|K_v(j0) + \eta K_r(j0)|} \quad \text{and} \quad |(G_v S)(j0)| = \frac{1}{D' (|K_v(j0) + \eta K_r(j0)|)}.$$

We now provide a sketch of the proposed design concept. Since K_v and K_r are chosen as high DC-gain controllers (obtained by solving the \mathcal{H}_∞ optimization problem), we have $|G_v S((j\omega))| \approx 0$ at low-frequencies. Thus the effect of disturbance signal i_{load} is insignificant at low frequencies. Similarly $T_{V_{\text{ref}}V}(j\omega)$ has unity gain at low frequencies. Furthermore, if the load current i_{load} measurement is available (i.e. $i_{\text{ref}} = i_{\text{load}}$), then the Boost converter tracks the reference voltage with almost unity gain. However in the absence of i_{load} measurement, the tracking error depends on the mismatch between i_{ref} and i_{load} , i.e., the bound on the steady-state tracking error becomes proportional to $\frac{|K_r(j0)|}{|K_v(j0) + \eta K_r(j0)|}$ multiplied by the mismatch value $|i_{\text{ref}}(j0) - i_{\text{load}}(j0)|$. By choosing appropriate controllers K_v and K_r (i.e., $|T_{i_{\text{ref}}V}(j0)| \ll 1$), the tracking error can be made small.

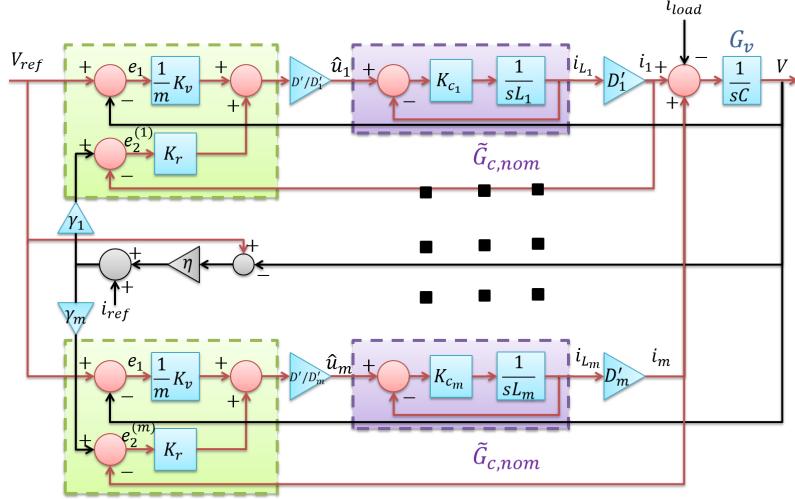


Figure 7.4: Many-converter system with shaped inner plants \tilde{G}_c . In the proposed implementation, we adopt the same outer controller for different converters, i.e., $K_{v_1} = \dots = K_{v_m} = \frac{1}{m} K_v$ and $K_{r_1} = \dots = K_{r_m} = K_r$; γ_k represents the proportion of power demanded from the k^{th} source.

7.1.2 Extension to Multi-Converters System

In this section we extend our control framework for a single converter to a system of DC-DC converters connected in parallel in the context of power sharing, keeping in mind the practicability and robustness to modeling and load uncertainties. In particular, we analyze the multi-converter system in Fig. 7.4 through an *equivalent* single-converter system (similar to the system shown in Fig. 7.1), where the multi-converter system inherits the performance and robustness achieved by a design for the single-converter system. Note that the control design for a network of parallel converters is particularly challenging because of inter-converter couplings. Moreover, analysis of closed-loop stability and performance of a parallel networked system is often infeasible unless a structured control architecture is imposed. Majority of existing control design methodologies fail to address this important issue of analyzability and can be evaluated only empirically.

Note that instead of feeding the reference current i_{ref_k} directly to the k^{th} outer controller K_{r_k} in the proposed architecture in Fig. 7.4, the reference signal $i_{\text{ref}} + \eta(V_{\text{ref}} - V)$ is prescaled by a time-varying multiplier $\gamma_k, 0 \leq \gamma_k \leq 1$. The choice of γ_k dictates the power sharing requirements on the k^{th} converter. In fact, we later show that the proposed implementation distributes the output power in the ratios $\gamma_1 : \gamma_2 : \dots : \gamma_m$. After noting that the voltage-regulation and current reference tracking is common to all the outer controllers, in our architecture, we impose the same design for outer-controllers for all the converters, i.e., $K_{v_1} = K_{v_2} = \dots = K_{v_m}$ and $K_{r_1} = K_{r_2} = \dots = K_{r_m}$. This imposition enables significant reduction in the overall complexity of the distributed control design for a parallel network of converters and power sources, thus

ensuring the practicability of the proposed design which allows integration of power sources of different types and values.

We design inner-controllers K_{c_k} such that the inner-shaped plants from \tilde{u}_k to i_{L_k} are same and given by,

$$\tilde{G}_{c,nom}(s) = \left(\frac{\tilde{\omega}}{s + \tilde{\omega}} \right) \left(\frac{s^2 + 2\zeta_{1,nom}\omega_0 s + \omega_0^2}{s^2 + 2\zeta_{2,nom}\omega_0 s + \omega_0^2} \right), \quad (7.1.6)$$

where the ratio $\zeta_{1,nom}/\zeta_{2,nom}$ determines the trade-off of 120Hz ripple between the total output current $D'i_L = \sum_{k=1}^m D'_k i_{L_k}$ and the capacitor current i_C . Note that for given values of $\zeta_{1,nom}, \zeta_{2,nom}$ and inductance L_k , explicit design of K_{c_k} exists and is given by (7.1.2). Furthermore, we impose that $K_{v_k} = \frac{1}{m}K_v$ and $K_{r_k} = K_r$.

Indeed, by our choice of inner and outer controllers, the transfer functions from external references V_{ref} , i_{ref} and i_{load} to the desired output V are identical for all converters. Hence the entire network of parallel converters can be analyzed in the context of an equivalent single converter system. This implies that K_{v_k} and K_{r_k} can be computed by solving \mathcal{H}_∞ -optimization problem (as discussed in the previous section) similar to the *single* converter case. We make these design specifications more precise and bring out the equivalence of the control design for the single and multiple converter systems in the following theorem.

We say that the system representation in Fig. 7.1 is *equivalent* to that in Fig. 7.4, when the transfer functions from the reference voltage V_{ref} , reference current i_{ref} and load current i_{load} to the DC-link voltage V in Fig. 7.1 are identical to the corresponding transfer functions in Fig. 7.4.

Theorem 1 Consider the single-converter system in Fig. 7.1 with inner-shaped plant $\tilde{G}_{c,nom}(s)$ as given in (7.1.6), outer controllers K_v, K_r , droop-coefficient η , and external references $V_{ref}, i_{load}, i_{ref}$; and the multi-converter system described in Fig. 7.4 with inner-shaped plants $\tilde{G}_{c_k} = \tilde{G}_{c,nom}(s)$ and outer controllers $K_{v_k} = \frac{1}{m}K_v; K_{r_k} = K_r$, droop-coefficient η , and same external references V_{ref}, i_{load} and reference current i_{ref} prescaled by time-varying scalars $\gamma_k > 0$ for $1 \leq k \leq m$.

1. [System Equivalence]: If $\sum_{k=1}^m \gamma_k = 1$, then the system representation in Fig. 7.1 is equivalent to the system representation in Fig. 7.4.

2. [Power Sharing]: For any two converters k and l , $k, l \in \{1, \dots, m\}$ in a multi-converter system shown in Fig. 7.4, the difference in the corresponding steady-state scaled output currents is given by:

$$\left| \frac{D'_k i_{L_k}(j0)}{\gamma_k} - \frac{D'_l i_{L_l}(j0)}{\gamma_l} \right| \leq \left(\eta |\tilde{T}_1(j0)| + \left| \frac{1}{\gamma_k} - \frac{1}{\gamma_l} \right| |\tilde{T}_2(j0)| \right) |e_1(j0)|, \quad (7.1.7)$$

where, $\tilde{T}_1 := \frac{D' \tilde{G}_{c,nom} K_r}{(1+D' \tilde{G}_{c,nom} K_r)}$ and $\tilde{T}_2 := \frac{D' \tilde{G}_{c,nom} K_v}{m(1+D' \tilde{G}_{c,nom} K_r)}$. Furthermore, the steady-state tracking error $e_1 \triangleq V_{ref} - V$ in DC-link voltage is upper bounded by,

Centralized case: $i_{ref} = i_{load}$

$$|e_1(j0)| \leq \frac{1}{D'(|K_v(j0) + \eta K_r(j0)|)} |i_{ref}(j0)|$$

Decentralized case: $i_{ref} \neq i_{load}$

$$|e_1(j0)| \leq \frac{|K_r(j0)|}{D'(|K_v(j0) + \eta K_r(j0)|)} |i_{ref}(j0)| + \frac{D' |K_r(j0)| + 1}{D'(|K_v(j0) + \eta K_r(j0)|)} |i_{load}(j0)|$$

Remark 1: If the steady-state tracking error in DC-link voltage is zero, i.e., $|e_1(j0)| = 0$, then (7.1.7) reduces to the following constraint:

$$\frac{|D'_k i_{L_k}(j0)|}{|D'_l i_{L_l}(j0)|} = \frac{\gamma_k}{\gamma_l},$$

i.e., the closed-loop multi-converter system achieves output power sharing given by $|D'_1 i_{L_1}(j0)| : \dots : |D'_m i_{L_m}(j0)| = \gamma_1 : \dots : \gamma_l$. In practice the tracking error e_1 is never exactly zero, however, the tracking error is made practically non-existent through an appropriate choice of large DC-gain controllers K_v, K_r resulting from the \mathcal{H}_∞ optimization problem in (7.1.3). Moreover, the design of the controllers is such that $|K_v(j0)| < |K_r(j0)|$ resulting in $|\tilde{T}_1(j0)| \leq 1$ and $|\tilde{T}_2(j0)| \leq 1$.

Remark 2: For the *decentralized* implementation, it is required that each converter can measure its own inductor current i_{L_k} and DC-link voltage V only.

Proof: See Appendix.

7.2 Control of DC-AC Inverters

For controlling a full-bridge inverter, we propose a simple lead-lag based inner-outer control design for voltage regulation. Power sharing is achieved through a smart choice of inner controllers and distributing the output of outer-voltage controllers appropriately. We now describe the explicit choice of inner and outer controllers for a DC-AC inverter.

7.2.1 Control of Single Inverter

Any controller that is required to regulate the output voltage $V(t)$ at its reference value, must do so through an equivalent control of the modulating signal $m(t)$. However for the purpose of control design and implementation, one must focus directly on the control input $u(t)$. The objective of voltage regulation is achieved using a nested inner-current outer-voltage control architecture, shown in Fig. 7.5. The outer-voltage controller K_v generates a current reference i_{ref} for the inner-loop. The inner-current controller K_c regulates the

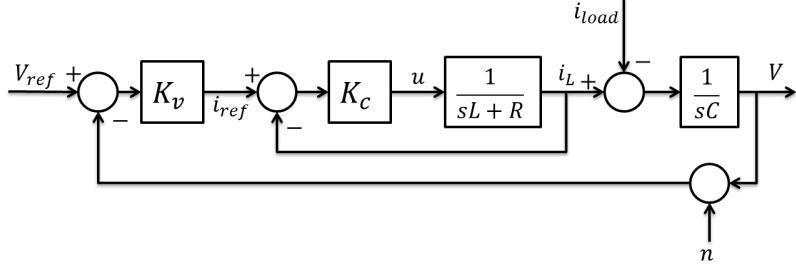


Figure 7.5: Inner-outer control architecture for a full-bridge inverter.

inductor current i_L to the desired i_{ref} . The quantity V_{ref} represents the desired reference voltage signal. For the purpose of control design, it is assumed that the signals $i_L(t)$ and $V(t)$ are available for measurements. The total current drawn at the PCC, which comprises of load current and any unmodeled disturbances, is denoted by i_{load} . For simplicity, we use $G_c := \frac{1}{sL + R}$ and $G_v := \frac{1}{sC}$ to denote the plant transfer functions in the inner and outer loops.

Design of the inner-loop controller

In our architecture, the main objective of the inner-controller K_c is to ensure regulation of inductor current i_L to the reference i_{ref} generated by K_v . Since the AC signal pulsates at $\omega_n = 60\text{Hz}$, K_c must ensure robust tracking at frequencies at least till ω_n . Additionally, it is preferred to have a relatively low-order controller K_c . These objectives are achieved through an appropriate *loop shaping* using a lead-lag controller as described below.

We assume that i_{ref} is required to be tracked with a *closed-loop* bandwidth of $\omega_b (\sim 10\omega_n)$; ω_b is chosen sufficiently large such that the inner closed-loop system from i_{ref} to i_L has unity gain till ω_n with zero-phase delay, and is small enough to ensure that the switching ripple content of the control signal u is low. In the loop shaping procedure, the desired performance objectives are specified in terms of the properties of the loop transfer function $l(s) = G_c(s)K_c(s)$. For achieving zero steady-state error at ω_n , the controller must be equipped with a pair of complex-conjugate poles at $s = \pm j\omega_n$, also referred as resonant controller in the literature [87]. Furthermore the open-loop plant $G_c(s)$ contains a pole at $s = -R/L$, which in turn introduces a -90° phase delay for frequencies larger than $10R/L$. Therefore, to improve the loop-gain phase, a zero at $s = -R/L$ is introduced. This pole-zero cancellation is admissible since the pole is on the left half plane (LHP). Thus the modified inner-loop controller assumes the form,

$$K_c(s) = \left(\frac{sL + R}{s^2 + \omega_n^2} \right) H(s). \quad (7.2.1)$$

Note that the resonant controller $1/(s^2 + \omega_n^2)$ introduces a -180° phase delay at and beyond ω_n . However to achieve a stable closed-loop system, the loop-gain phase at the gain crossover frequency ω_c must be larger than -180° by a value that is referred to as the phase-margin. For robust stability, a phase-margin of about 60° is required, i.e. $\angle l(j\omega_c) = 60^\circ$. The gain crossover frequency ω_c and the closed-loop -3dB bandwidth ω_b are correlated and in general satisfy the inequality $\omega_c < \omega_b < 2\omega_c$, resulting in following approximation $\omega_b \approx 1.5\omega_c$. A phase-margin of about 60° can be achieved by using a lead filter of the form

$$F_{\text{lead}}(s) = \left(\frac{s + \omega_c/\sqrt{\alpha}}{s + \omega_c\sqrt{\alpha}} \right), \quad (7.2.2)$$

where $\alpha > 1, \alpha \in \mathbb{R}$. The maximum phase of this lead filter occurs at ω_c and is given by

$$\delta_{\max} = \sin^{-1} \left(\frac{\alpha - 1}{\alpha + 1} \right). \quad (7.2.3)$$

For a phase-margin of 60° , δ_{\max} is chosen to be 60° which results in $\alpha = 13.93$. For achieving zero steady-state error at DC, the magnitude of loop-gain must be large ($\sim 50\text{dB}$) at low frequencies, i.e. $|l(j0)| = 10^{2.5}$. The loop-gain magnitude is increased at low frequencies if the following lag-filter is introduced

$$F_{\text{lag}}(s) = \left(\frac{s + \beta}{s + \delta\beta} \right), \quad (7.2.4)$$

where $\beta \sim 2 \text{ rad/s}$ and $\delta < 1, \delta \in \mathbb{R}$. Note that F_{lag} has the property that $F_{\text{lag}}(j\omega) \approx 1$ for frequencies larger than about 20 rad/s. Therefore, it does not change the phase or magnitude of the loop-gain around the crossover frequency ω_c . Thus the transfer function $H(s)$ in Eq. (7.2.1) is expressed as:

$$H(s) = h \underbrace{\left(\frac{s + \beta}{s + \delta\beta} \right)}_{F_{\text{lag}}(s)} \underbrace{\left(\frac{s + \omega_c/\sqrt{\alpha}}{s + \omega_c\sqrt{\alpha}} \right)}_{F_{\text{lead}}(s)}, \quad (7.2.5)$$

where $h \in \mathbb{R}$ is added to ensure that $|l(j\omega_c)| = 1$. Therefore, the inner-loop controller K_c assumes the following modified form

$$K_c(s) = h \left(\frac{sL + R}{s^2 + \omega_n^2} \right) \left(\frac{s + \beta}{s + \delta\beta} \right) \left(\frac{s + \omega_c/\sqrt{\alpha}}{s + \omega_c\sqrt{\alpha}} \right), \quad (7.2.6)$$

where h and δ satisfy the following equations

$$\begin{aligned} h &= \sqrt{\alpha} (\omega_c^2 - \omega_n^2) \\ \delta &= \frac{h}{10^{2.5}\alpha\omega_n^2}. \end{aligned} \quad (7.2.7)$$

Thus the inner-loop controller $K_c(s)$ described in (7.2.6) ensures robust (60° phase-margin) tracking with large loop-gains at low-frequencies and ω_n .

Remark: If the current reference i_{ref} is constant, the inverter is just controlled to be a current source, which is the same with the traditional grid-tied inverter. Thus a grid-tied operation of inverters does not require an outer-voltage loop, since the PCC voltage is fixed by the utility grid.

Design of the outer-loop controller

The inner-loop controller K_c ensures that the inductor current tracks the current reference i_{ref} . However, i_{ref} is an internal signal in the closed-loop system of Fig. 7.5 and is produced by the outer-voltage controller K_v which is designed to regulate the AC-side voltage V at its reference value V_{ref} . Since the objectives of the outer-loop are similar to those of the inner-loop (albeit in terms of the voltage signal), we use similar design methodology.

Note that the outer-loop plant $G_v(s) = 1/sC$ introduces large gains in the loop-transfer function $\tilde{l}(s) = G_v(s)K_v(s)$ at low-frequencies. Thus a lag-controller is not required for the outer-loop control design. In an inner-outer cascaded control design, the outer-loop controller is the primary controller that regulates the primary controlled variable (V) at the desired reference, whereas the inner-loop (secondary) controller rejects any input disturbance locally before it propagates to the outer-loop plant (G_v). Thus for a cascaded design to function properly, the inner-loop must respond much faster than the outer-loop. This is achieved by ascribing the outer-loop controller K_v such that the outer loop-gain crossover frequency $\tilde{\omega}_c \sim 0.4 - 0.5\omega_c$. Thus $K_v(s)$ assumes the following functional form:

$$K_v(s) = \tilde{h}C \left(\frac{s + \beta}{s^2 + \omega_n^2} \right) \left(\frac{s + \tilde{\omega}_c/\sqrt{\alpha}}{s + \tilde{\omega}_c\sqrt{\alpha}} \right), \quad (7.2.8)$$

where $\tilde{h} \in \mathbb{R}$ is chosen such that $|\tilde{l}(j\tilde{\omega}_c)| = 1$ and is given by

$$\tilde{h} = \tilde{\omega}_c \sqrt{\frac{\alpha}{\beta^2 + \tilde{\omega}_c^2}} (\tilde{\omega}_c^2 - \omega_n^2). \quad (7.2.9)$$

Remark: The inner-loop controller K_c depends on the inverter parameters L and R , while the outer-

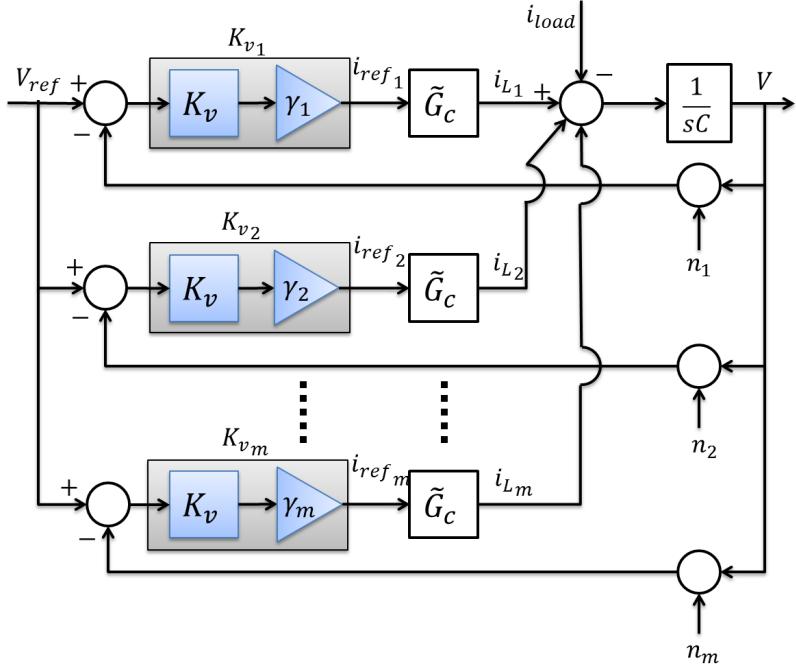


Figure 7.6: Proposed decentralized framework with *smart* choices of controller parameters. The inner-current controllers K_{c_k} are chosen such that the inner-shaped plant mimics $\tilde{G}_c(s)$. The outer-loop controllers are scalar multiples of the nominal outer-loop controller K_v ; the scalars γ_k govern the power-sharing requirements in a microgrid setup.

loop controller K_v depends on the capacitance C at the output. This parametric dependence (and inner-outer decoupling) is exploited in the next section to extend the control design to multiple parallel inverter system.

7.2.2 Extension to Multi-Inverters System

A microgrid facilitates integration of multiple parallel full-bridge VSIs at the PCC. In this section, we describe a new non-droop based decentralized power sharing scheme through an extension of the proposed single inverter control. Such an extension achieves the objective of power sharing and voltage regulation in the context of DC/DC converters too and is reported in our prior work [86]. An *isochronous* operation is assumed for the parallel inverters case, i.e., we assume existence of common time reference among inverters. This assumption is needed to ensure that all the inverters have access to the same time-domain voltage reference signal V_{ref} (see [78] for more details on isochronous operation).

Since the multi-inverter system is highly coupled with individual controllers having access only to *local* current measurements i_{L_k} , any arbitrary choice of controller transfer functions $\{K_{v_k}, K_{c_k}\}_{k=1}^m$ renders the stability and performance analysis of the multi-inverter system intractable. However, the decentralized framework is easily simplified by a *smart* choice of inner-outer controllers. For given desired gain-crossover

frequencies ω_c and $\tilde{\omega}_c$ for the inner and outer-loops, respectively, we make the following two important observations -

1) The inner-controllers $K_{c_k}, k \in \{1, \dots, m\}$ are parameterized by the corresponding coupling impedances (see Eq. (7.2.6)), and therefore the respective loop-gains $l_k(s) = \frac{1}{(sL_k + R_k)}K_{c_k}(s)$ are independent of the parameters L_k and R_k as a consequence of admissible pole-zero cancellations, i.e.,

$$l_k(s) = \left(\frac{h}{s^2 + \omega_n^2} \right) \left(\frac{s + \beta}{s + \delta\beta} \right) \left(\frac{s + \omega_c/\sqrt{\alpha}}{s + \omega_c\sqrt{\alpha}} \right), \forall k. \quad (7.2.10)$$

Thus $l_1(s) = \dots = l_m(s) =: l(s)$ and therefore the closed inner-loops $\tilde{G}_{c_k} = l_k(s)/(1 + l_k(s))$ are identical, i.e., $\tilde{G}_{c_1}(s) = \dots = \tilde{G}_{c_m}(s) =: \tilde{G}_c(s)$.

2) The objective of voltage regulation at the PCC is common to all the inverters. Thus in the proposed decentralized architecture, we impose *similar* structure for the outer-voltage controllers $K_{v_k}, k \in \{1, \dots, m\}$, i.e.,

$$K_{v_k}(s) = \underbrace{\gamma_k \tilde{h}C \left(\frac{s + \beta}{s^2 + \omega_n^2} \right) \left(\frac{s + \tilde{\omega}_c/\sqrt{\alpha}}{s + \tilde{\omega}_c\sqrt{\alpha}} \right)}_{:= K_v(s)}, \forall k, \quad (7.2.11)$$

where $\gamma_k \in \mathbb{R}, \gamma_k \in [0, 1] \quad \forall k \in \{1, \dots, m\}$ and satisfy $\sum_{k=1}^m \gamma_k = 1$. The parameters γ_k are chosen to apportion power among parallel sources. We make these design specifications more precise and bring out the equivalence of the control design for the single and multiple parallel VSIs in the following theorem.

Theorem 2 Consider a single VSI system described in Fig. 7.5 with parameters L, R and C , and controllers K_c and K_v described by Eqs. (7.2.1) and (7.2.8), respectively; and a parallel inverter system in Fig. 7.6 with same output capacitance C , but distinct inverter system parameters $\{L_k, R_k\}_{k=1}^m$ with inner and outer controllers K_{c_k} and $K_{v_k} = \gamma_k K_v$ as described in Eqs. (7.2.10) and (7.2.11) such that $\sum_{k=1}^m \gamma_k = 1$.

1. [Performance Equivalence]: The controllers K_{v_k} and K_{c_k} yield identical (to single VSI control) performance for a network of multiple parallel inverters connected at the PCC; more precisely, for the exogenous inputs - the reference V_{ref} , the load disturbance current i_{load} , and measurement-noise $n = \sum_{k=1}^m \gamma_k n_k$, the steady-state regulated signals $(V_{ref} - V, i_L, V)$ for the single-inverter system are same as the regulated signals $(V_{ref} - V, \sum_{k=1}^m i_{L_k}, V)$ for the multi-inverter system.

2. [Power Sharing]: The steady-state output currents at the PCC get divided in the ratio $\gamma_1 : \dots : \gamma_m$ up to the measurement noises; more precisely, if the measurement noise is bounded above, i.e., $(|n_k(j\omega)| < \epsilon(\omega)), \forall k$ then,

$$\left| \frac{i_{L_j}(j\omega)}{\gamma_j} - \frac{i_{L_k}(j\omega)}{\gamma_k} \right| \leq |\tilde{G}_c(j\omega)| \cdot |K_v(j\omega)| \epsilon(\omega).$$

Consequently, if $\epsilon(\omega) = 0$, then the above condition reduces to $|i_{L_1}(j\omega)| : \dots : |i_{L_m}(j\omega)| = \gamma_1 : \dots : \gamma_m$ for all ω . Thus in case of perfect measurements, the proportions γ_k capture the power-sharing requirements exactly.

Remark: By design, the inner-closed loop plant $\tilde{G}_c(j\omega)$ has unity gain till bandwidth and rolls-off at higher frequencies. Similarly the outer controller $K_v(j\omega)$, given by (7.2.8), rolls-off at higher frequencies. Thus the effect of high-frequency noise is mitigated by the choice of control design and the output current is apportioned according to the prescribed sharing requirements.

Proof: See appendix.

Remark: The inner-loop controller K_c depends on the inverter parameters L and R , while the outer-loop controller K_v depends on the capacitance C at the output. This parametric dependence (and inner-outer decoupling) is exploited in the next section to extend the control design to multiple parallel inverter system.

Chapter 8

Case Studies: Simulations and Discussions

This chapter describes simulation studies that cover different aspects of the proposed distributed control design. All simulations are performed in MATLAB/Simulink using SimPower/SimElectronics library, which use *non-ideal* components (such as diodes with non-zero breakdown voltage, IGBT switches, stray capacitances, parametric uncertainties) and switched level implementation to include nonlinearities associated with real-world experiments. Here, we consider the setup shown in Fig. 8.1.

Note that the photovoltaics are technically treated as *current* sources. In a microgrid setup, a PV module is interfaced with the DC-link through a boost converter and is controlled using the maximum power point tracking (MPPT) algorithm. The output current of PV i_{PV} is directly proportional to the (time-varying) irradiance and is included in our proposed formulation by regarding i_{PV} as part of the disturbance signal, i.e., the net disturbance current is modeled as $i_{\text{load}} - i_{\text{PV}}$. In this simulation study, we squeeze worth 8 hours of insolation data into a total duration of 19.5s amounting to rapidly varying irradiance (and hence the disturbance current i_{PV}).

8.1 DC-DC Converters

In order to illustrate the robustness of the proposed approach, the control design assumes nominal (or equivalent single converter) inductance, capacitance and steady-state complementary duty-cycle given by $L = 0.12\text{mH}$, $C = 500\mu\text{F}$ and $D' = V_g/V_{\text{ref}} = 0.5$, whereas the simulated system has non-identical inductances and steady-state complementary duty-cycles. The mismatch (or uncertainty) in L and C parameters is large ($\sim 20\%$). The design parameters for the inner-controller K_c are: damping factors $\zeta_1 = 0.7$, $\zeta_2 = 2.2$, and bandwidth $\tilde{\omega} = 2\pi 300 \text{rad/s}$. The outer controllers K_v and K_r are obtained by solving the stacked \mathcal{H}_∞ optimization problem (see Equation 7.1.3)[81] using appropriate weighting functions.

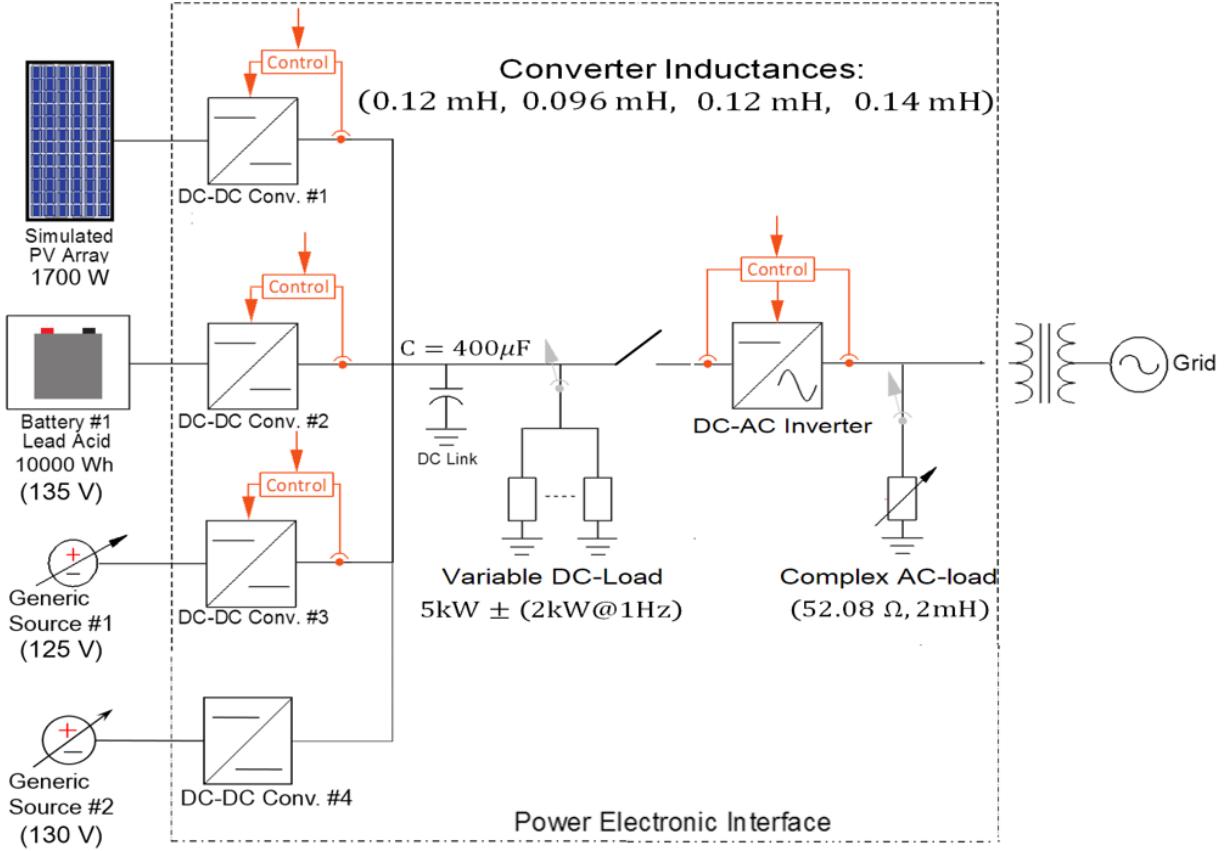


Figure 8.1: A parallel network comprising of a PV, a Li-ion battery and two generic sources. The operating voltages and associated converter parameters are described above. It is desired to regulate the DC-link voltage to 250V. The PV module is operated using MPPT algorithm [2]. Its output current, i_{PV} , is directly proportional to the (time-varying) irradiance and is included in our proposed formulation by regarding i_{PV} as part of the disturbance signal, with the net disturbance current modeled as $i_{load} - i_{PV}$. The DC-link can additionally be used to power complex AC loads via a DC-AC inverter. The inverter is tied to utility grid. Any control design must facilitate seamless integration of DC microgrid with the AC grid.

Results

The controllers derived for the nominal single converter system are used to derive controller parameters for a parallel multi-converter system as described in section 7.1. (by setting for each converter $K_{v_k} = \frac{1}{m}K_v$ and $K_{r_k} = K_r$ for all $k = 1, \dots, m$). Fig. 8.2 shows the (i) voltage regulation at the DC-link to the reference $V_{ref} = 250V$ for the centralized (i_{load} measurement available) and decentralized implementations, (ii) *time-varying* current sharing among power sources. The sources are initially required to provide power in equal proportion, followed by a proportion of 5 : 2 : 3 from $t = 2s$ onwards. For ease of illustration, the scaled output currents $D'i_L/\gamma$ are plotted. Overlapping values of scaled currents depict excellent sharing performance. The DC-link load changes by 4kW every second (3kW to 7kW, and 7kW to 3kW). The reference current is considered as $i_{ref} = 5kW/250V = 20A$.

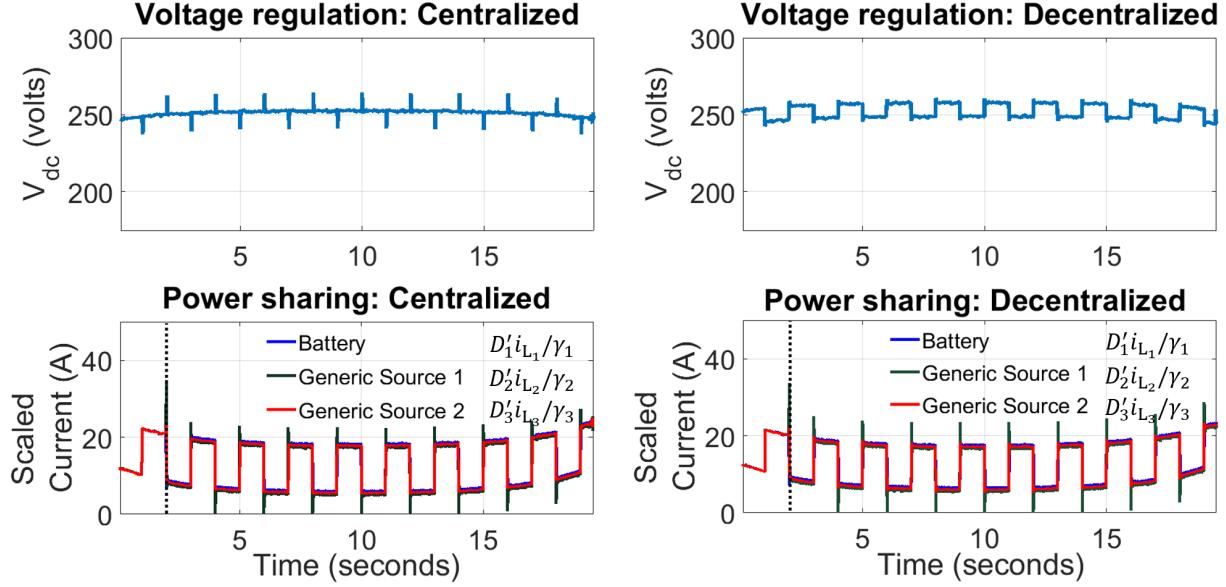


Figure 8.2: Simulation results representing centralized and decentralized control implementation for voltage regulation and time-varying power sharing.

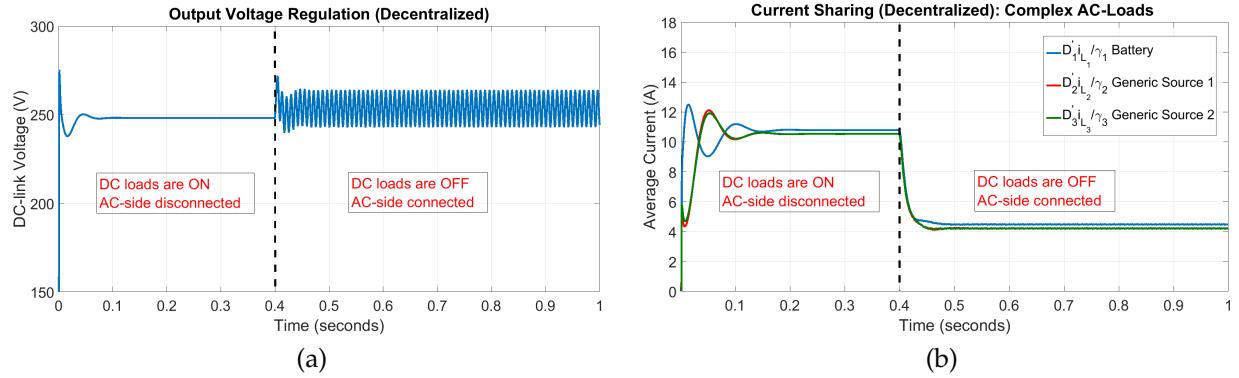


Figure 8.3: Simulation results representing handling of complex AC loads.

Fig. 8.3a and Fig. 8.3b show the results of adding complex AC loads through a DC-AC inverter. The converter system is required to operate in “DC-only” mode until 0.4s. The three DC-sources are required to share their output power in the ratio of 4 : 3 : 3. The DC-load considered in this test case has a resistance of 20Ω . Subjected to these conditions, the DC-sources regulate the DC-link voltage at 250V (see Fig. 8.3a), while ensuring desired sharing performance (see Fig. 8.3b). At $t = 0.4$, the DC-load is dropped and the networked system is interfaced with a complex AC-load ($R,L = (52.08\Omega, 2mH)$) through a grid-tied DC-AC inverter. Despite this sudden interconnection, the proposed control design facilitates seamless integration to ensure that the average DC-link voltage is regulated at desired 250V, while ensuring the same sharing capabilities. The transient response to grid interconnection remains well within acceptable limits.

8.2 DC-AC Inverters

We now describe simulation case studies for a network of multiple DC-AC inverters using the control design proposed in section 7.2. All simulations are done using Simulink/Simscape [88] components, which incorporate dynamical models of batteries and generic DC sources. The customized converter/source library is available for download at [89].

Voltage regulation in presence of parametric uncertainties

The inner-outer controllers in Equation 7.2.6 and Equation 7.2.8 are designed with high phase-margins (60°) which imparts robustness to modeling and parametric uncertainties. The simulation parameters are given below:

$$V_{\text{ref}} = A \sin(2\pi\omega_n t), \quad \text{where, } A = \begin{cases} 400V & \text{if } t < 0.2s; \\ 500V & \text{if } t \geq 0.2s. \end{cases}$$

AC-load unknown to controller,

$$(R_l, L_l) = \begin{cases} (83m\Omega, 137\mu H) & \text{if } t < 0.4s; \\ (41.5m\Omega, 68.5\mu H) & \text{if } t \geq 0.4s. \end{cases}$$

Nominal parameters : $L = 100\mu H$, $R = 0.88m\Omega$.

Nominal PCC Capacitance : $C = 2500\mu F$.

The controllers are designed with the nominal parametric values, while the simulations are performed with 20% uncertainty in L and C values. The desired inner-loop bandwidth is chosen to be $\omega_b = 11,100 \text{ rad/s}$. The resulting inner-outer controllers are:

$$\begin{aligned} K_c &= \frac{20384(s + 1983)(s + 16.3)(s + 2)}{(s + 2.76e4)(s + 0.65)(s^2 + (377)^2)} \\ K_v &= \frac{80421(s + 793.1)(s + 2)}{(s + 1.105e4)(s^2 + (377)^2)}. \end{aligned} \quad (8.2.1)$$

Fig. 8.4 shows the result of voltage regulation for 20% uncertainty in L and C . The voltage at the PCC gets regulated at its reference V_{ref} within one cycle.

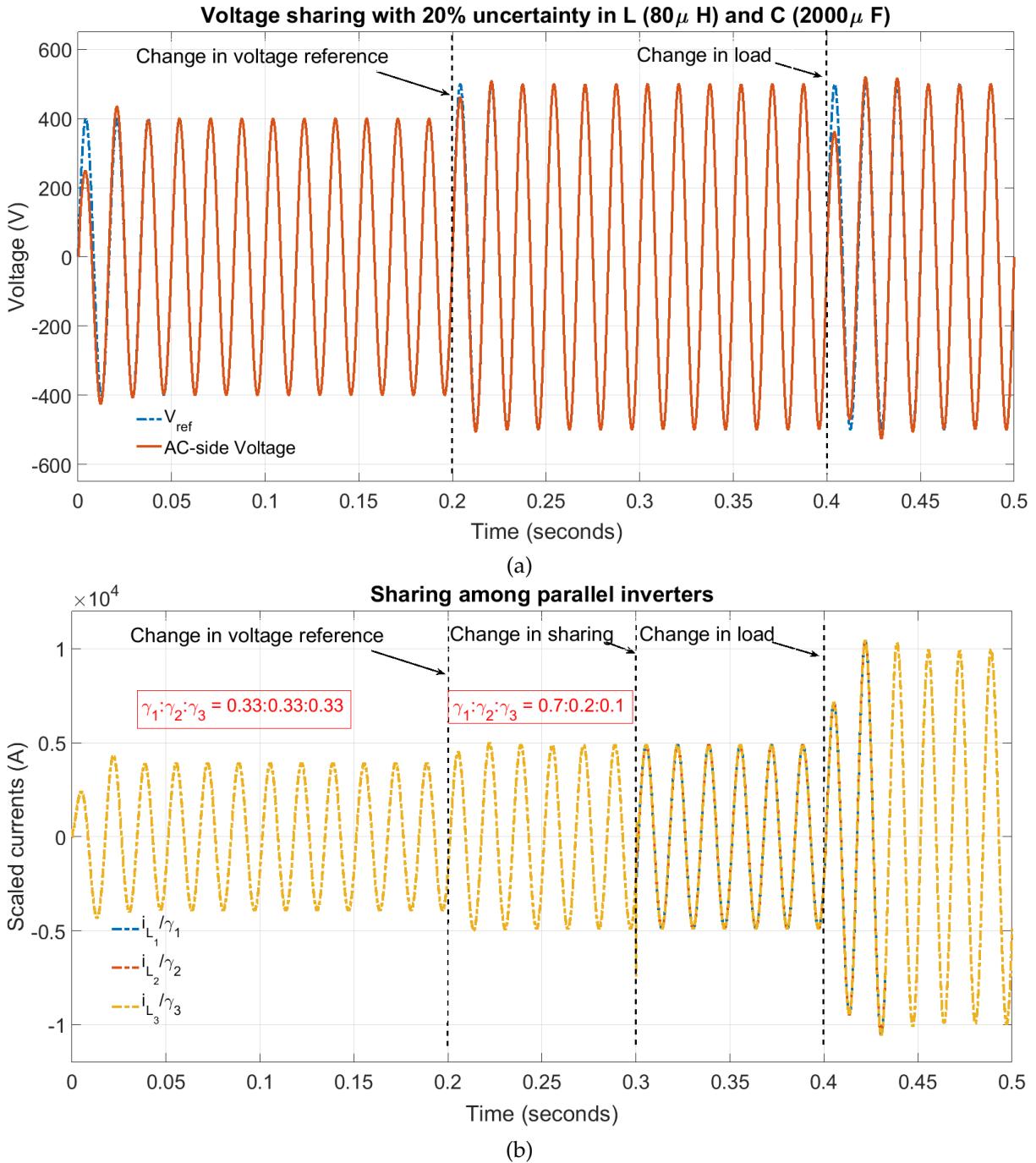


Figure 8.4: (a) Voltage regulation through a network of three parallel inverters with heterogeneous power sources. (b) Power sharing among multiple inverters.

Power sharing among three inverters

We now substantiate the proposed non-droop based sharing for a three-inverter system. We consider three heterogeneous power sources - 1) Lithium-ion battery (Nominal voltage: 1500V, Initial State-Of-Charge: 120%), 2) Generic Source-1 (DC Voltage: 1500V), 3) Generic Source-2 (DC Voltage: 1200V). The other simulation parameters are chosen as before. The power sharing requirements are:

$$\gamma_1 : \gamma_2 : \gamma_3 = \begin{cases} 0.33 : 0.33 : 0.33 & \text{if } t < 0.3\text{s;} \\ 0.70 : 0.20 : 0.10 & \text{if } t \geq 0.3\text{s.} \end{cases}$$

Fig. 8.4a presents the voltage regulation through a network of three parallel inverters with heterogeneous DC sources under - 1) change in reference voltage, V_{ref} , 2) change in power sharing requirements, and 3) change in AC-load. Fig. 8.4b shows the scaled values of inverter currents ($i_{L_1}/\gamma_1 : i_{L_2}/\gamma_2 : i_{L_3}/\gamma_3$). As can be seen from the figure, the resultant scaled currents overlap with each other, thereby establishing that the power gets divided in the ratio $\gamma_1 : \gamma_2 : \gamma_3$. Note that the sharing ratios are being maintained even during the transients and therefore provide for faster sharing than the droop-based methods, and even more so with very small voltage tracking error.

Chapter 9

Experimental Validation

To verify the effectiveness of the proposed approach, a test rig with three parallel operated DC sources and a parallel PV simulator PVS60085MR is built (see Fig. 9.1) [90]. It is desired to regulate the DC-link voltage to $V_{\text{ref}} = 60\text{V}$.

In order to illustrate the robustness of the proposed approach, the control design assumes nominal (or equivalent single converter) inductance, capacitance and steady-state complementary duty-cycle given by $L = 0.12\text{mH}$, $C = 500\mu\text{F}$ and $D' = V_g / V_{\text{ref}} = 0.5$, whereas the simulated system has non-identical inductances and steady-state complementary duty-cycles. The mismatch (or uncertainty) in L and C parameters is large ($\sim 20\%$). The design parameters for the inner-controller K_c are: damping factors $\zeta_1 = 0.7$, $\zeta_2 = 2.2$, and bandwidth $\tilde{\omega} = 2\pi 300\text{rad/s}$. The outer controllers K_v and K_r are obtained by solving the stacked \mathcal{H}_∞ optimization problem (see Eq. (Equation 7.1.3))[81] using following weighting functions.

$$W_1 = 0.4167 \frac{(s + 452.4)}{(s + 1.885)}, \quad W_2 = 0.4167 \frac{(s + 1206)}{(s + 5.027)}, \quad W_3 = 0.4, \quad W_4 = 37.037 \frac{(s + 314.2)}{(s + 3.142e4)}. \quad (9.0.1)$$

Here W_1 is chosen to be large in the frequency range $[0, 30]$ Hz so that the sensitivity transfer function corresponding to error in voltage tracking is small in that frequency. Similarly, W_2 is chosen to be large in the frequency range $[0, 80]$ Hz so that the transfer function from mismatch between the sourced output current and the reference current to the voltage regulation error is small. The bandwidth of W_2 is chosen to be large than the bandwidth of W_1 , primarily to allow for faster dynamics in the inner current loop since the change in capacitor voltage occurs at a relatively slower timescale than a sudden change in the loading conditions. By satisfying this condition, the reference value of the inner loop which is the output of the outer controller can be considered constant (see Fig. Figure 7.1). W_3 is chosen to be constant and is designed to make the control effort lie within the limits at all frequencies. Finally W_4 is designed as high-pass filter to ensure that the transfer function from i_{load} to V is small at high frequencies, which mitigates effects of high-frequency measurement noise. The corresponding outer controllers K_v and K_r are obtained by solving a multi-objective \mathcal{H}_∞ -optimization problem in Equation 7.1.3. The outer controller orders are

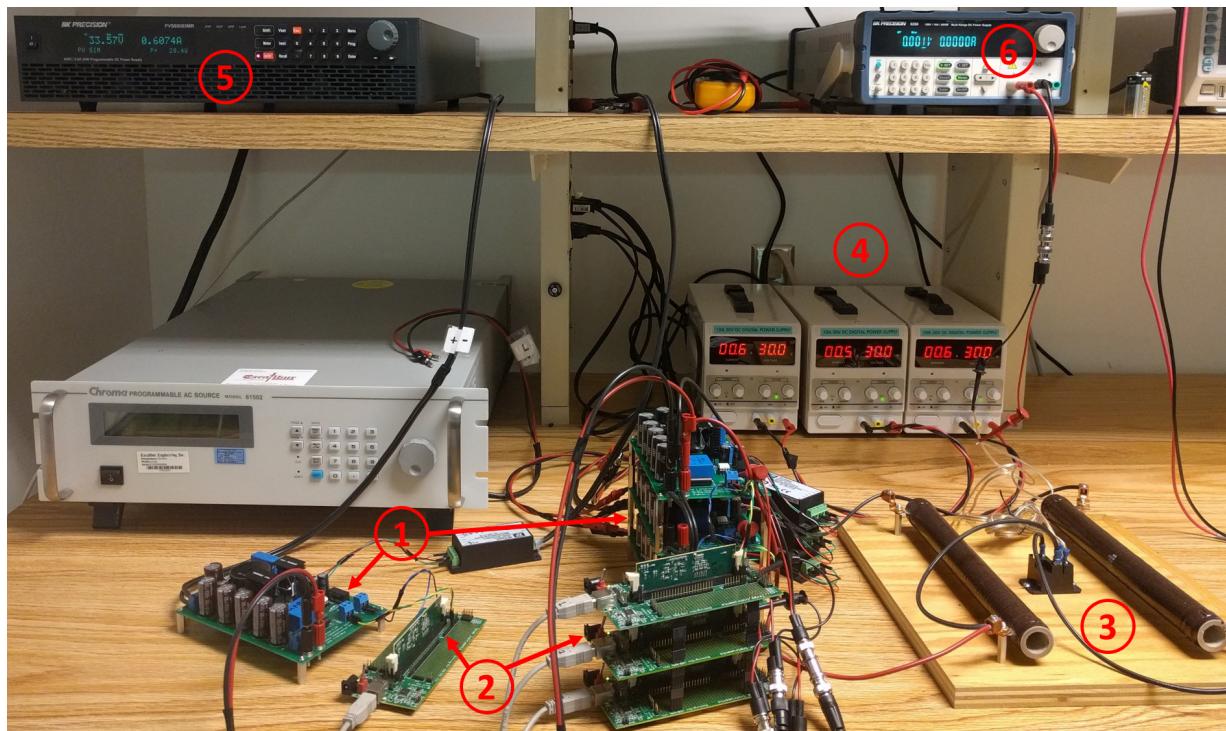


Figure 9.1: Experimental setup with (1) custom-designed boost-converter boards, (2) controllers implemented on TMS320F28335 Delfino MCUs, (3) variable load - two resistors, each of value 50Ω , (4) DC-sources with maximum rated output voltage of 30V, (5) PV simulator subjected to simulated noisy ramp profile with a peak power of 43W and controlled using MPPT algorithm , and (6) relay for load.

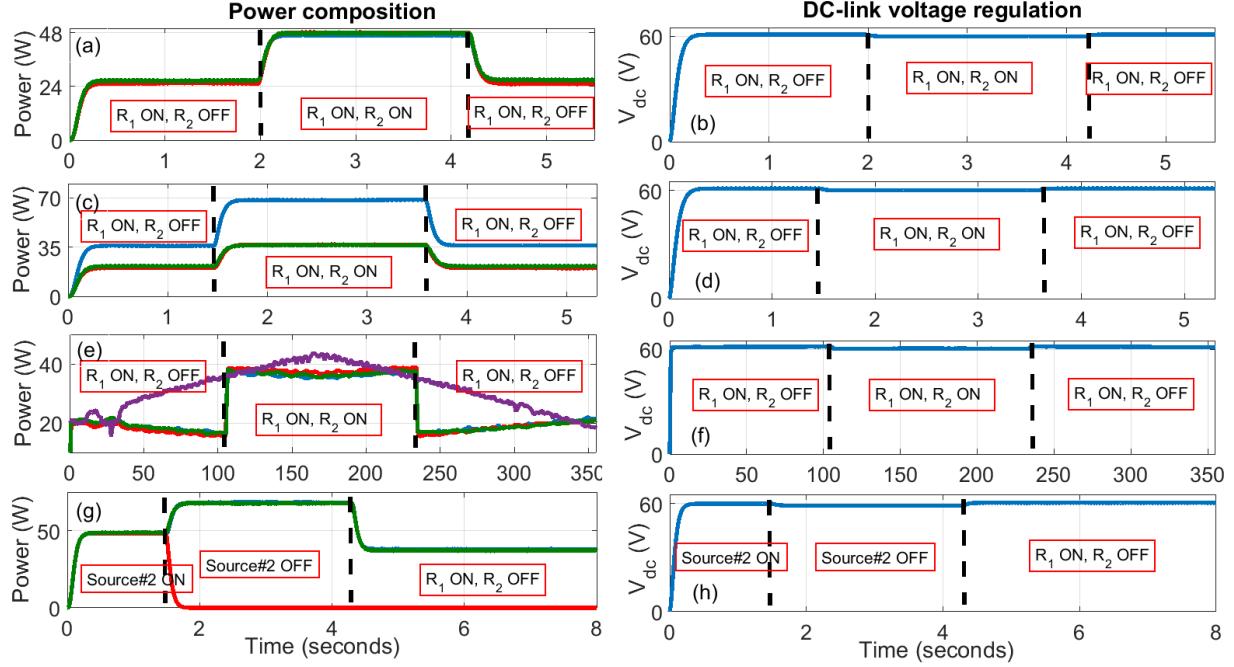


Figure 9.2: Experimental results demonstrating effectiveness of the proposed control design under perfectly decentralized implementation for several test scenarios: 1:1:1 sharing (PV off) - (a) and (b); 2:1:1 sharing (PV off) - (c) and (d); 1:1:1 sharing (PV on) - (e) and (f); Equal sharing in presence of abrupt failure in power generation - (g) and (h). Colors blue, red, green and purple indicate power outputs of DC sources 1, 2, 3 and PV emulator, respectively.

then reduced using balanced truncation [82] for efficient implementation:

$$K_v = 0.69 \frac{(s + 4.42 \times 10^6)(s + 167)(s^2 + 3930s + 1.75 \times 10^7)}{(s + 4891)(s + 719.2)(s^2 + 7.21 \times 10^4s + 2.51 \times 10^9)}$$

$$K_r = -0.12 \frac{(s - 4.56 \times 10^5)(s + 1.12 \times 10^4)(s + 355.7)(s + 248.9)}{(s + 4.64 \times 10^5)(s + 4.96)(s^2 + 714.9s + 2.66 \times 10^5)}. \quad (9.0.2)$$

The controllers for a *nominal* single-converter system are designed using the multi-objective robust optimal control framework described in chapter 7 and is extended to a three-converter system using the methodology described. In order to analyze robustness to modeling uncertainties, a 50% uncertainty in capacitance is considered. For brevity, case studies pertaining only to more challenging decentralized scenario are reported; where total load current i_{load} is *unknown* and there is no communication among the controllers. Furthermore, PV is regarded as a current source and injects power directly at the DC-link, as described in chapter 8. Since, the load current is unknown, constant $i_{\text{ref}} = 2\text{A}$ is used.

Several scenarios are considered for the purpose of experimental validation and are shown in Fig. 9.2. These are:

Case A: Power sharing when PV is off

Figures 9.2a and 9.2c show that power from the DC sources get distributed respectively in ratios 1 : 1 : 1 and 2 : 1 : 1, irrespective of the load at the DC-link; even when there are load changes as high as 100%. Figures 9.2b and 9.2d illustrate excellent DC-link voltage regulation at $V_{\text{ref}} = 60V$ in absence of communication between controllers about load. The regulation error is within 1V even when load is changed by 100%.

Case B: Power sharing with PV on

We now evaluate the performance of our control design under additional uncertainty in power generation, that is, a PV source under simulated noisy ramp irradiance profile is connected at the DC-link. The converter controllers to generic DC-sources are agnostic to PV output. The inclusion of PV also tests the robustness of the system to load disturbances since PV current can be viewed as time varying uncertain load at the DC-link for the rest of the power sources. Fig. 9.2e shows that DC sources adequately compensate for the PV disturbance, that is, they exhibit power profile *complementary* to PV profile, even though the loading conditions are not communicated to the controllers; also DC-link voltage is well regulated (see Fig. 9.2f).

Case C: Resilient to unforeseen failure in power generation in an agnostic setup

Robust performance of the networked system is now evaluated for the scenario when one of the generic DC sources is abruptly turned OFF (mimicking a power source failure in a network). Furthermore, this information is not communicated to the network.

If this information were communicated, our architecture in Sec. subsection 7.2.2, will make the following changes - 1) The outer controllers $K_{v_k} = (1/m)K_v$ will be updated to $K_{v_k} = K_v/(m-1)$, and 2) $\sum \gamma_k$ will be readjusted to sum up to 1 for the active sources. However, even without this communication and edits, Fig. 9.2g shows that as the DC-source #2 is abruptly turned OFF, the net power output from other DC sources auto-adjusts to loss in power generation from DC-source #2, and ensures DC-link voltage regulation (Fig. 9.2h) and equal power sharing (Fig. 9.2g). Furthermore, at $t = 4.3s$, load R_2 is shed, while DC-source #2 is still inactive. Despite the generation and load uncertainties, DC-link voltage is maintained within the viable limits and the load power is shared equally by the active sources.

Thus with the experimental evidence, the robustness and reliability of the design methodology is now established.

Chapter 10

Conclusions and Future Directions

In this thesis, we propose a scalable, decentralized, robust, plug-and-play control architecture for a network of parallel DC-DC converters and DC-AC inverters. The networked system resulting from our architecture is robust to communication failures, uncertainties in load demands and schedules, system parameters, and noise in measurement signals. Moreover, the entire networked system is analyzable in terms of an equivalent single converter/inverter system - which to the best of our knowledge is state-of-the-art as the conventional control methods that do not exploit structural properties of the network often come with almost no guarantees on closed-loop performance and stability characteristics.

Furthermore, it is shown that the same control architecture with exactly same controllers can be employed for both centralized as well as distributed implementations, that is, the controller architecture demonstrates quantifiably better performance with communication while guaranteeing viability in presence of only local measurements in case of distributed implementation. The modular control framework coupled with robust control design facilitates negligible performance degradation, even when a DC source is agnostically engaged/disengaged from the network during nominal operation. Uncertain renewable sources, such as PV are prioritized to provide maximum available power (unknown to the network), and the remaining sources distributedly adjust their power outputs to meet the load requirements in prescribed time-varying proportions.

Simulation case studies comprising of battery, photovoltaic (PV) and generic sources are presented and demonstrate the enhanced performance of prescribed optimal controllers for voltage regulation and power sharing. These simulations are then validated through experiments on custom-designed converter boards. While some of the experimental results are not discussed in this thesis due to IP-related issues, the proposed control methodology was also evaluated on a large scale test bed at the National Renewable Energy Laboratory (NREL) and the results corroborate the effectiveness of the proposed design.

Microgrids have significant potentials in rural electrification and sustainable development, and the bottom-up approach described in this thesis is a step in this direction. Many small microgrids, arising from locally aggregated DERs, can be interfaced together to set up a high-capacity microgrid. This would

require a modular, scalable control architecture that forms the basis for control design for individual low-power microgrids. We exploit the inter and intra-modular structure within a network, such that a common control methodology is applicable across all units within a large network. Such a significant reduction in complexity of analyzing a large network enables integrating a large number of DERs.

As part of the future directions, we aim to implement the proposed control design methodology on a high-power, large scale experimental test bed at the Dynapower [91], a Vermont based leading independent provider of power conversion solutions. The design approach will be tested across many different scenarios while using more than 100 actual physical devices such as photovoltaics, battery storage inverters, and home appliances. If successful, the proposed design could potentially replace 4.5 GW of spinning reserves (i.e. generation capacity on stand-by in case of outages and unforeseen intermittency), a value of \$3.3 billion per year. A more efficient and reliable grid would help protect U.S. businesses from costly power outages and brownouts.

The current work does not incorporate the economic aspect of electricity market in its control approach. Inclusion of market economy would entail some finite-time consensus based protocols over the communication layer, such that demand and generation requirements in a network-side setting are updated in a distributed manner. There are immense avenues of exploration in microgrid research and in our opinion the proposed work is a key enabler towards this vision.

Appendix

Proofs of Theorem 1 & Theorem 2

Proof of Theorem 1: System Equivalence

Proof 1 The equivalence is a direct consequence of cleverly chosen architecture. Note that for the single converter system in Figure 7.1 with $\tilde{G}_c(s) = \tilde{G}_{c,nom}(s)$, the error signal e_2 (input to controller K_r) is given by

$$\begin{aligned} e_2 &= [i_{ref} + \eta e_1 - D' \tilde{G}_{c,nom} \hat{u}] \\ \Rightarrow e_2 &= i_{ref} + (\eta - D' \tilde{G}_{c,nom}) e_1 - D' \tilde{G}_{c,nom} K_r e_2. \end{aligned} \quad (.0.1)$$

For the multi-converter system in Figure 7.4, we have $e_1^{(k)} = V_{ref} - V := e_1$. Let us denote the total error in current mismatch $\sum_{k=1}^m e_2^{(k)}$ by e_2 . Therefore, from Fig. Figure 7.4,

$$\begin{aligned} e_2^{(k)} &= \gamma_k [i_{ref} + \eta e_1] - D' \tilde{G}_{c,nom} \left(\frac{1}{m} K_v e_1 + K_r e_2^{(k)} \right) \\ \underbrace{\sum_{k=1}^m e_2^{(k)}}_{e_2} &= \sum_{k=1}^m \gamma_k [i_{ref} + \eta e_1] - D' \tilde{G}_{c,nom} \left(K_v e_1 - K_r \underbrace{\sum_{k=1}^m e_2^{(k)}}_{e_2} \right). \end{aligned} \quad (.0.2)$$

Using the fact that $\sum_{k=1}^m \gamma_k = 1$, the above equation reduces to (.0.1). Similarly, the expression for tracking error in voltage $V_{ref} - V$ is identical for the single and multiple converters case. Moreover for the multi-converter system, the output voltage at the DC-link is given by $V = G_v(-i_{load} + D' \tilde{G}_{c,nom}(K_v e_1 + K_r e_2))$. Since the expressions for e_1 and e_2 are identical for the single and multiple converters case and are written in terms of the exogenous variables $V_{ref}, i_{ref}, i_{load}$, the corresponding transfer functions from the exogenous variables to the DC-link voltage V are also identical, and hence establishes the required equivalence. Similar conclusions can be drawn for other signals, such as DC-link voltage V , and hence establishes the required equivalence.

Proof of Theorem 1: Power Sharing

Proof 2 From (.0.2), the error in current reference for the k^{th} -converter is given by

$$e_2^{(k)} = \left(\frac{\gamma_k}{1 + D' \tilde{G}_{c,nom} K_r} \right) i_{ref} + \left(\frac{\gamma_k \eta - \frac{D'}{m} \tilde{G}_{c,nom} K_v}{1 + D' \tilde{G}_{c,nom} K_r} \right) e_1. \quad (.0.3)$$

From Fig. Figure 7.4, the output current $i_k = D'_k i_{L_k}$ of the k^{th} converter is given by

$$i_k = D' \tilde{G}_{c,nom} \left[\frac{1}{m} K_v e_1 + K_r e_2^{(k)} \right]. \quad (.0.4)$$

Thus from (.0.3) and (.0.4), we obtain

$$i_k = D' \tilde{G}_{c,nom} \left[\left(\frac{\gamma_k K_r}{1 + D' \tilde{G}_{c,nom} K_r} \right) i_{ref} + \left(\frac{\frac{1}{m} K_v + \eta \gamma_k K_r}{1 + D' \tilde{G}_{c,nom} K_r} \right) e_1 \right].$$

Therefore, we have

$$\left| \frac{i_k(j0)}{\gamma_k} - \frac{i_l(j0)}{\gamma_l} \right| \leq \left(\eta |\tilde{T}_1(j0)| + \left| \frac{1}{\gamma_k} - \frac{1}{\gamma_l} \right| |\tilde{T}_2(j0)| \right) |e_1(j0)|$$

The expressions for the bounds on the tracking error for the two scenarios is directly obtained from Equation 7.1.5 and the system equivalence described earlier.

Proof of Theorem 2: System Equivalence

Proof 3 Let \tilde{G}_c denote the inner-shaped plant in Equation 7.2.10. For the single inverter system described in Figure 7.5, the AC-side voltage at the PCC is given by

$$V = \underbrace{\left(\frac{G_v \tilde{G}_c K_v}{1 + G_v \tilde{G}_c K_v} \right)}_{:=T} (V_{ref} - n) - G_v \underbrace{\left(\frac{1}{1 + G_v \tilde{G}_c K_v} \right)}_{:=S} i_{load}. \quad (.0.5)$$

Thus the tracking error $(V_{ref} - V)$ is given by

$$V_{ref} - V = SV_{ref} + Tn - G_v Si_{load} \quad (.0.6)$$

For the parallel inverter system in Figure 7.6, the AC-side voltage V is given by

$$V = G_v \left(-i_{load} + \sum_{k=1}^m \gamma_k \tilde{G}_c K_v (V_{ref} - V - n_k) \right). \quad (.0.7)$$

Using the fact that $\sum_{k=1}^m \gamma_k = 1$ and $\sum_{k=1}^m \gamma_k n_k = n$, and from Eq. (.0.7) one obtains

$$V = T (V_{ref} - n) - G_v Si_{load}, \quad (.0.8)$$

which is identical to Eq. (.0.5) and thus yields identical expression for $V_{ref} - V$. Similarly, the inductor current i_L in Figure 7.5 is given by

$$i_L = \tilde{G}_c K_v (V_{ref} - V - n). \quad (.0.9)$$

The inductor current in the k^{th} inverter in Figure 7.6 is given by $i_{L_k} = \gamma_k \tilde{G}_c K_v (V_{\text{ref}} - V - n_k)$. Summing it over k yields

$$\sum_{k=1}^m i_{L_k} = \tilde{G}_c K_v (V_{\text{ref}} - V - n) = i_L, \quad (0.10)$$

which establishes the required equivalence.

Proof of Theorem 2: Power Sharing

Proof 4 The power sharing scheme follows directly from the construction. Note that the inductor current $i_{L_j} = \gamma_j \tilde{G}_c K_v (V_{\text{ref}} - V - n_j)$. Therefore for inverters j and k , we have

$$\frac{i_{L_j}}{\gamma_j} - \frac{i_{L_k}}{\gamma_k} = \tilde{G}_c K_v (n_k - n_j). \quad (0.11)$$

Since the measurement-noises are bounded, i.e., $|n_k(j\omega)| \leq \epsilon(\omega) \forall k \in \{1, \dots, m\}$, from (0.11) we conclude that

$$\left| \frac{i_{L_j}(j\omega)}{\gamma_j} - \frac{i_{L_k}(j\omega)}{\gamma_k} \right| \leq |\tilde{G}_c(j\omega)| \cdot |K_v(j\omega)| \epsilon(\omega). \quad (0.12)$$

References

- [1] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, "The complexity of multiterminal cuts," *SIAM Journal on Computing*, vol. 23, no. 4, pp. 864–894, 1994.
- [2] A. Askarian, M. Baranwal, and S. Salapaka, "Dc bus voltage regulation using photovoltaic module: A non-iterative method," in *2017 American Control Conference (ACC)*, pp. 4099–4104, May 2017.
- [3] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2210–2239, 1998.
- [4] N. M. Nasrabadi, "Pattern recognition and machine learning," *Journal of electronic imaging*, vol. 16, no. 4, p. 049901, 2007.
- [5] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [6] Z.-J. Lee and C.-Y. Lee, "A hybrid search algorithm with heuristics for resource allocation problem," *Information sciences*, vol. 173, no. 1-3, pp. 155–167, 2005.
- [7] P. Toth and D. Vigo, *The vehicle routing problem*. SIAM, 2002.
- [8] M. Baranwal, P. M. Parekh, L. Marla, S. M. Salapaka, and C. L. Beck, "f problem with time windows: A deterministic annealing approach," in *American Control Conference (ACC), 2016*, pp. 790–795, IEEE, 2016.
- [9] Y. Xu, S. M. Salapaka, and C. L. Beck, "Aggregation of graph models and markov chains by deterministic annealing," *Automatic Control, IEEE Transactions on*, vol. 59, no. 10, pp. 2807–2812, 2014.
- [10] C. H. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A min-max cut algorithm for graph partitioning and data clustering," in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pp. 107–114, IEEE, 2001.
- [11] R. Gray and E. Karnin, "Multiple local optima in vector quantizers (corresp.)," *IEEE transactions on Information Theory*, vol. 28, no. 2, pp. 256–261, 1982.
- [12] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations research*, vol. 14, no. 4, pp. 699–719, 1966.
- [13] M. Tawarmalani and N. V. Sahinidis, "A polyhedral branch-and-cut approach to global optimization," *Mathematical Programming*, vol. 103, no. 2, pp. 225–249, 2005.
- [14] S. P. Lloyd, "Least squares quantization in pcm," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, 1982.
- [15] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, *et al.*, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [16] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*, pp. 760–766, Springer, 2011.

- [17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [18] E. T. Jaynes, "Information theory and statistical mechanics," *Physical review*, vol. 106, no. 4, p. 620, 1957.
- [19] K. Rose, E. Gurewitz, and G. Fox, "A deterministic annealing approach to clustering," *Pattern Recognition Letters*, vol. 11, no. 9, pp. 589–594, 1990.
- [20] B. Schölkopf, "The kernel trick for distances," in *Advances in neural information processing systems*, pp. 301–307, 2001.
- [21] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 11, no. 9, pp. 1074–1085, 1992.
- [22] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.
- [23] M. Baranwal and S. M. Salapaka, "Clustering with capacity and size constraints: A deterministic approach," in *Control Conference (ICC), 2017 Indian*, pp. 251–256, IEEE, 2017.
- [24] P. V. Gehler and O. Chapelle, "Deterministic annealing for multiple-instance learning," in *Artificial Intelligence and Statistics*, pp. 123–130, 2007.
- [25] S. Mitra, R. Castellanos, and S. Joshi, "An adaptive deterministic annealing approach for medical image segmentation," in *Fuzzy Information Processing Society, 2000. NAFIPS. 19th International Conference of the North American*, pp. 82–84, IEEE, 2000.
- [26] A. Rao, K. Rose, and A. Gersho, "Design of robust hmm speech recognizers using deterministic annealing," in *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pp. 466–473, IEEE, 1997.
- [27] N. Ueda and R. Nakano, "Deterministic annealing variant of the em algorithm," in *Advances in neural information processing systems*, pp. 545–552, 1995.
- [28] Y. Xu, S. Salapaka, and C. L. Beck, "Dynamic maximum entropy algorithms for clustering and coverage control," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 1836–1841, IEEE, 2010.
- [29] J. D. Little, K. G. Murty, D. W. Sweeney, and C. Karel, "An algorithm for the traveling salesman problem," *Operations research*, vol. 11, no. 6, pp. 972–989, 1963.
- [30] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Operations research*, vol. 21, no. 2, pp. 498–516, 1973.
- [31] K. L. Hoffman, M. Padberg, and G. Rinaldi, "Traveling salesman problem," in *Encyclopedia of operations research and management science*, pp. 1573–1578, Springer, 2013.
- [32] L. Tang, J. Liu, A. Rong, and Z. Yang, "A multiple traveling salesman problem model for hot rolling scheduling in shanghai baoshan iron & steel complex," *European Journal of Operational Research*, vol. 124, no. 2, pp. 267–282, 2000.
- [33] S. Gorenstein, "Printing press scheduling for multi-edition periodicals," *Management Science*, vol. 16, no. 6, pp. B–373, 1970.
- [34] D. J. Gulczynski, J. W. Heath, and C. C. Price, "The close enough traveling salesman problem: A discussion of several heuristics," in *Perspectives in Operations Research*, pp. 271–283, Springer, 2006.
- [35] W. K. Mennell, *Heuristics for solving three routing problems: Close-enough traveling salesman problem, close-enough vehicle routing problem, sequence-dependent team orienteering problem*. PhD thesis, 2009.
- [36] R. Durbin and D. Willshaw, "An analogue approach to the travelling salesman problem using an elastic net method," *Nature*, vol. 326, no. 6114, p. 689, 1987.

- [37] A. L. Yuille, "Generalized deformable models, statistical physics, and matching problems," *Neural Computation*, vol. 2, no. 1, pp. 1–24, 1990.
- [38] A. Ravindran, G. V. Reklaitis, and K. M. Ragsdell, *Engineering optimization: methods and applications*. John Wiley & Sons, 2006.
- [39] "Simulated annealing in matlab." <http://yarpiz.com/223/ypea105-simulated-annealing/>.
- [40] G. Reinelt, "Tsplib traveling salesman problem library," *ORSA journal on computing*, vol. 3, no. 4, pp. 376–384, 1991.
- [41] W. A. Wulf, "Great achievements and grand challenges," *The Bridge*, vol. 30, no. 3, p. 4, 2000.
- [42] P. Lagonotte, J. Sabonnadiere, J.-Y. Leost, and J.-P. Paul, "Structural analysis of the electrical system: Application to secondary voltage control in france," *IEEE transactions on power systems*, vol. 4, no. 2, pp. 479–486, 1989.
- [43] T. Van Cutsem and C. Vournas, *Voltage stability of electric power systems*, vol. 441. Springer Science & Business Media, 1998.
- [44] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [45] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.
- [46] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *nature*, vol. 393, no. 6684, p. 440, 1998.
- [47] E. Hogan, E. Cotilla-Sánchez, M. Halappanavar, S. Wang, P. Mackey, P. Hines, and Z. Huang, "Towards effective clustering techniques for the analysis of electric power grids," in *Proceedings of the 3rd International Workshop on High Performance Computing, Networking and Analytics for the Power Grid*, p. 1, ACM, 2013.
- [48] E. Cotilla-Sánchez, P. D. Hines, C. Barrows, S. Blumsack, and M. Patel, "Multi-attribute partitioning of power networks based on electrical distance," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4979–4987, 2013.
- [49] W. R. Wagner, A. Keyhani, S. Hao, and T. C. Wong, "A rule-based approach to decentralized voltage control," in *Power Industry Computer Application Conference, 1989. PICA'89, Conference Papers*, pp. 163–171, IEEE, 1989.
- [50] A. Monticelli, *State estimation in electric power systems: a generalized approach*, vol. 507. Springer Science & Business Media, 1999.
- [51] G. Andersson, "Modelling and analysis of electric power systems," *EEH-Power Systems Laboratory, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland*, pp. 141–147, 2004.
- [52] S. Singh, G. Raju, and A. Gupta, "Sensitivity based expert system for voltage control in power system," *International Journal of Electrical Power & Energy Systems*, vol. 15, no. 3, pp. 131–136, 1993.
- [53] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 551–556, ACM, 2004.
- [54] W. H. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of classification*, vol. 1, no. 1, pp. 7–24, 1984.
- [55] M. Ackerman, S. Ben-David, and D. Loker, "Characterization of linkage-based clustering," in *COLT*, pp. 270–281, Citeseer, 2010.

- [56] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [57] C. Carmeli, E. De Vito, and A. Toigo, "Reproducing kernel hilbert spaces and mercer theorem," *arXiv preprint math/0504071*, 2005.
- [58] H. Q. Minh, P. Niyogi, and Y. Yao, "Mercers theorem, feature maps, and smoothing," in *International Conference on Computational Learning Theory*, pp. 154–168, Springer, 2006.
- [59] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-supervised graph clustering: a kernel approach," *Machine learning*, vol. 74, no. 1, pp. 1–22, 2009.
- [60] P. Sharma, S. Salapaka, and C. Beck, "A scalable deterministic annealing algorithm for resource allocation problems," in *American Control Conference, 2006*, pp. 6–pp, IEEE, 2006.
- [61] S. Lowen, "Multiway cut," 2011.
- [62] H. S. Stone, "Multiprocessor scheduling with the aid of network flow algorithms," *IEEE transactions on Software Engineering*, no. 1, pp. 85–93, 1977.
- [63] P. L. Erdős and L. A. Székely, "On weighted multiway cuts in trees," *Mathematical programming*, vol. 65, no. 1-3, pp. 93–105, 1994.
- [64] D. Marx, "A tight lower bound for planar multiway cut with fixed number of terminals," in *International Colloquium on Automata, Languages, and Programming*, pp. 677–688, Springer, 2012.
- [65] G. Călinescu, H. Karloff, and Y. Rabani, "An improved approximation algorithm for multiway cut," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 48–52, ACM, 1998.
- [66] J. Naor and L. Zosin, "A 2-approximation algorithm for the directed multiway cut problem," *SIAM Journal on Computing*, vol. 31, no. 2, pp. 477–482, 2001.
- [67] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the ACM (JACM)*, vol. 19, no. 2, pp. 248–264, 1972.
- [68] Y. Dinitz, "Dinitzalgorithm: The original version and evens version," in *Theoretical computer science*, pp. 218–240, Springer, 2006.
- [69] A. Rangarajan, A. L. Yuille, S. Gold, and E. Mjolsness, "A convergence proof for the softassign quadratic assignment algorithm," in *Advances in neural information processing systems*, pp. 620–626, 1997.
- [70] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM transactions on graphics (TOG)*, vol. 23, pp. 309–314, ACM, 2004.
- [71] A. L. Yuille and J. Kosowsky, "Statistical physics algorithms that converge," *Neural computation*, vol. 6, no. 3, pp. 341–356, 1994.
- [72] U. E. I. Administration, "International energy outlook 2013," 2013.
- [73] N. Hatziargyriou, H. Asano, R. Iravani, and C. Marnay, "Microgrids," *IEEE power and energy magazine*, vol. 5, no. 4, pp. 78–94, 2007.
- [74] J. P. Lopes, C. Moreira, and A. Madureira, "Defining control strategies for microgrids islanded operation," *IEEE Transactions on power systems*, vol. 21, no. 2, pp. 916–924, 2006.
- [75] C. Olalla, R. Leyva, A. El Aroudi, P. Garces, and I. Queinnec, "Lmi robust control design for boost pwm converters," *IET Power Electronics*, vol. 3, no. 1, pp. 75–85, 2010.
- [76] G. Weiss, Q.-C. Zhong, T. C. Green, and J. Liang, "H ∞ repetitive control of dc-ac converters in microgrids," *IEEE Transactions on Power Electronics*, vol. 19, no. 1, pp. 219–230, 2004.

- [77] T. Hornik and Q.-C. Zhong, "A current-control strategy for voltage-source inverters in microgrids based on $h \infty$ and repetitive control," *IEEE Transactions on Power Electronics*, vol. 26, no. 3, pp. 943–952, 2011.
- [78] S. Salapaka, B. Johnson, B. Lundstrom, S. Kim, S. Collyer, and M. Salapaka, "Viability and analysis of implementing only voltage-power droop for parallel inverter systems," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp. 3246–3251, IEEE, 2014.
- [79] Y. Panov, J. Rajagopalan, and F. C. Lee, "Analysis and design of n paralleled dc-dc converters with master-slave current-sharing control," in *Applied Power Electronics Conference and Exposition, 1997. APEC'97 Conference Proceedings 1997., Twelfth Annual*, vol. 1, pp. 436–442, IEEE, 1997.
- [80] R.-H. Wu, T. Kohama, Y. Kodera, T. Ninomiya, and F. Ihara, "Load-current-sharing control for parallel operation of dc-to-dc converters," in *Power Electronics Specialists Conference, 1993. PESC'93 Record., 24th Annual IEEE*, pp. 101–107, IEEE, 1993.
- [81] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*, vol. 2. Wiley New York, 2007.
- [82] G. E. Dullerud and F. Paganini, *A course in robust control theory: a convex approach*, vol. 36. Springer Science & Business Media, 2013.
- [83] R. W. Erickson and D. Maksimovic, *Fundamentals of power electronics*. Springer Science & Business Media, 2007.
- [84] P. T. Krein, *Elements of power electronics*, vol. 126. Oxford University Press New York, 1998.
- [85] M. Baranwal, S. M. Salapaka, and M. V. Salapaka, "Robust decentralized voltage control of dc-dc converters with applications to power sharing and ripple sharing," in *American Control Conference (ACC), 2016*, pp. 7444–7449, IEEE, 2016.
- [86] M. Baranwal, A. Askarian, S. M. Salapaka, and M. V. Salapaka, "A robust scheme for distributed control of power converters in dc microgrids with time-varying power sharing," in *American Control Conference (ACC), 2017*, pp. 1413–1418, IEEE, 2017.
- [87] R. Teodorescu, M. Liserre, and P. Rodriguez, *Grid converters for photovoltaic and wind power systems*, vol. 29. John Wiley & Sons, 2011.
- [88] M. U. Guide, "The mathworks," *Inc., Natick, MA*, vol. 5, p. 333, 1998.
- [89] http://web.engr.illinois.edu/~baranwa2/Converter_Lib.slx.
- [90] M. Baranwal, A. Askarian, S. Salapaka, and M. Salapaka, "Distributed architecture for robust and optimal control of dc microgrids," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2018.
- [91] <https://www.dynapower.com/>.