A TWO-STAGE DESIGN FRAMEWORK FOR OPTIMAL SPATIAL PACKAGING OF INTERCONNECTED FLUID-THERMAL SYSTEMS

BY

SATYA RAVI TEJA PEDDADA

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Systems and Entrepreneurial Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Associate Professor James T Allison, Chair and Director of Research
Professor Nathan M Dunfield
Assistant Professor Kai A James
Associate Professor Pingfeng Wang
Dr. Lawrence E Zeidner, Raytheon Technologies Research Center

Abstract

Optimal spatial packaging of interconnected subsystems and components with coupled physical (thermal, hydraulic, or electromagnetic) interactions, or SPI2, plays a vital role in the functionality, operation, energy usage, and life cycle of practically all engineered systems, from chips to ships to aircraft. However, the highly nonlinear spatial packaging problem, governed by coupled physical phenomena transferring energy through highly complex and diverse geometric interconnects, has largely resisted automation, and quickly exceeds human cognitive abilities at moderate complexity levels. The current state-of-the-art in defining an arrangement of these functionally heterogeneous artifacts still largely relies on human intuition and manual spatial placement, limiting system sophistication and extending design timelines. Spatial packaging involves packing and routing, which are separately challenging NP-hard problems. Therefore, solving the coupled packing and routing (PR) problem simultaneously will require disruptive methods to better address pressing related challenges, such as system volume reduction, interconnect length reduction, ensuring non-intersection, and physics considerations.

This dissertation presents a novel automated two-stage sequential design framework to perform simultaneous physics-based packing and routing (PR) optimization of fluid-thermal systems. In Stage 1, unique spatially-feasible topologies (i.e., how interconnects and components pass around each other) are enumerated for a given fluid-thermal system architecture. It is important to guarantee a feasible initial graph as lumped-parameter physics analyses may fail if components and/or routing paths intersect. Stage 2 begins with a spatially-feasible layout, and optimizes physics-based system performance with respect to component locations, interconnect paths, and other continuous component or system variables (such as sizing or control). A bar-based

design representation enables the use of a differentiable geometric projection method (GPM), where gradient-based optimization is used with finite element analysis. In addition to geometric considerations, this method supports optimization based on system behavior by including physics-based (temperature, fluid pressure, head loss, etc.) objectives and constraints. In other words, stage 1 of the framework supports systematic navigation through discrete topology options utilized as initial designs that are then individually optimized in stage 2 using a continuous gradient-based topology optimization method. Thus, both the discrete and continuous design decisions are made sequentially in this framework. The design framework is successfully demonstrated using different 2D case studies such as a hybrid unmanned aerial vehicle (UAV) system, automotive fuel cell (AFC) packaging system, and other complex multi-loop systems.

The 3D problem is significantly more challenging than the 2D problem due to a vastly more expansive design space and potential features. A review of state-of-the-art methods, challenges, existing gaps, and opportunities are presented for the optimal design of the 3D PR problem. Stage 1 of the framework has been investigated thoroughly for 3D systems in this dissertation. An efficient design framework to represent and enumerate 3D system spatial topologies for a given system architecture is demonstrated using braid and spatial graph theories. After enumeration, the unique spatial topologies are identified by calculating the Yamada polynomials of all the generated spatial graphs. Spatial topologies that have the same Yamada polynomial are categorized together into equivalent classes. Finally, CAD-based 3D system models are generated from these unique topology classes. These 3D models can be utilized in stage 2 as initial designs for 3D multi-physics PR optimization. Current limitations and significantly impactful future directions for this work are outlined. In summary, this novel design automation framework integrates several elements together as a foundation toward a more comprehensive solution of 3D real-world packing and routing problems with both geometric and physics considerations.

To all my teachers, friends, and family members for their constant love and support.

Acknowledgments

First and foremost, I am very grateful to my research advisor, Professor James Allison for his constant encouragement and support for the past five years. Without his dedicated, patient, and thoughtful guidance this would have been an impossible task. Overall, it has been my honor and a great learning experience to have been advised by him.

I would also like to sincerely thank all the dissertation committee members: Prof. Kai James, Dr. Larry Zeidner, Prof. Nathan Dunfield, and Prof. Pingfeng Wang for their time and insightful feedback whenever I needed. I have greatly benefited from the many discussions we had on various dissertation-related topics over the past few years.

Furthermore, I would like to acknowledge the valuable feedback from a number of industry personnel, Dr. Joe Zimmerman from CU Aerospace, Dr. Mark Jennings, Dr. Sunil Katragadda, and Dr. Man Prakash Gupta from Ford Motor company, and Tim Deppen from PC Krause and Associates on my dissertation-related project topics. I also benefited a lot from the courses I took under a number of graduate faculty at the University of Illinois. I thank them for their insightful interactions and dedicated teaching that helped in my research.

I am also very thankful to the members of the Engineering Systems Design Laboratory (ESDL) for both their valuable suggestions and significant feedback on my research. In this regard, I would like to sincerely thank former ESDL graduates Dr. Anand Deshmukh, Dr. Daniel Herber, and Dr. Danny Lohan for their assistance and senior guidance in my graduate studies. I would also like to thank Dr. Yong Hoon Lee, Albert Patterson, Vedant, Tonghui Cui, and Sayeid Bayat for being such nice and resourceful colleagues

at work.

I sincerely thank Prof. Andrew Alleyne, Center Director for the NSF center for Power Optimization of Electro-Thermal Systems (POETS) along with his former graduate students, Dr. Pamela Tannous, and Dr. Herschel Pangborn, with whom I thoroughly enjoyed collaborating on several interesting research projects. I also thank Anurag Bhattacharya, Waheed Bello, and Samanta Rodriguez for their enthusiastic collaboration. I also enjoyed working with POETS education coordinators Dr. Jessica Perez and Joe Muskin, and thank them for their help in organizing several STEM outreach activities in and around the campus. I would also like to thank Jodi Gritten, POETS Office Manager and the Department of Industrial and Enterprise Systems Engineering (ISE) graduate coordinator Lauren Redman, former ISE staff Aleta Lynch, Barb Bohlen, and Holly Kizer for their timely administrative support whenever I needed during the graduate program.

I sincerely acknowledge the financial support I received from the NSF POETS center and the ISE Department which helped me carry out my research in a smooth fashion.

Last but not the least, I am personally indebted to my loving teacher, all my family members, and mentors for their constant encouragement and unwavering support from India. Without their blessings, understanding, and sacrifices, this was never happening.

Ultimately, I am grateful to my dear merciful Lord Sri Varaha Lakshmi Narasimha Swamy who gave me the strength, intelligence, protection, all the resources at the right time, and wonderfully orchestrated everything in my journey towards the Doctor of Philosophy degree.

TABLE OF CONTENTS

LIST OF TABLES ix

LIST OF FIGURES x

LIST OF ABBREVIATIONS xiv

LIST OF SYMBOLS xvi

CHAPTER 1 INTRODUCTION 1
- 1.1 Motivation 1
- 1.2 Some Important Terms 2
- 1.3 SPI2 Problem Definition 3
- 1.4 SPI2 Design Research: Stuck since Decades 6
- 1.5 Complexity of SPI2 vs. VLSI 7
- 1.6 Need for SPI2 Design Automation 7
- 1.7 Research Objectives 8
- 1.8 Dissertation Overview 10

CHAPTER 2 REVIEW: SPI2 DESIGN RESEARCH AREAS . 13
- 2.1 Four SPI2 Research Elements 13
- 2.2 3D Packing and Component Layout Design 14
- 2.3 3D Interconnect Routing 15
- 2.4 Physics-based Topology Optimization 17
- 2.5 SPI2 Mathematical Design Representations 18
- 2.6 Existing Critical Gaps 20
- 2.7 Associated Design Challenges 22
- 2.8 Summary 23

CHAPTER 3 THE SPI2 DESIGN FRAMEWORK: STAGE 1 . 25
- 3.1 Introduction 25
- 3.2 2D vs. 3D SPI2 Design Problems: Similarities and Differences 26
- 3.3 The Two-Stage SPI2 Design Framework 27
- 3.4 Chapter Aims and Contributions 31
- 3.5 2D SPI2 Problem Design Representation 32
- 3.6 2D Force-Directed Layout Method (FDLM) 33

3.7 2D Shortest-path based Layout Algorithms (SPA) 39
3.8 2D Unique Spatial Topology (UST) Enumeration Algorithm . 43
3.9 Stage 1: Case Studies 45
3.10 Discussion 50
3.11 Summary 54

CHAPTER 4 THE SPI2 DESIGN FRAMEWORK: STAGE 2 . 55
4.1 Introduction 55
4.2 Geometric Projection Method 56
4.3 Physics models 57
4.4 Optimization problem and sensitivity analysis 65
4.5 Stage 2 Illustration: UAV test platform 75
4.6 Case Studies 81
4.7 Discussion 85
4.8 Summary 86

CHAPTER 5 DESIGN REPRESENTATION AND SYSTEMATIC ENUMERATION OF 3D SYSTEM SPATIAL TOPOLOGIES 94
5.1 Introduction 94
5.2 Spatial Graphs 99
5.3 Yamada Polynomial Invariants 102
5.4 3D Spatial Topology Enumeration Framework 110
5.5 Case Studies 113
5.6 Discussion 121
5.7 Summary 122

CHAPTER 6 CONCLUSIONS AND FUTURE WORK 125
6.1 Dissertation summary 125
6.2 Contributions 128
6.3 Future Work 129

REFERENCES 134

List of Tables

4.1 Unmanned aerial vehicle's component properties. 78
4.2 Power electronics cooling system optimization results. Optimal objective function values are highlighted in yellow. . . . 78
4.3 Final component temperatures of optimized layouts shown in Fig. 4.14 are satisfying the physics-based temperature constraints. 83

5.1 Yamada polynomials of diagrams shown in Fig. 5.10 (case study 5.5.1). 115
5.2 Case study 2: Unique spatial topology classes with increasing number of components and crossings. 115

List of Figures

1.1 Diverse examples of typical manually designed systems presenting SPI2 spatial packing and routing complexity, subject to physics interactions, and exhibiting spatial accessibility challenges for life-cycle processes: A) the externals (components, wires, pipes and ducts interconnecting components and engine features) of a commercial turbofan engine covering the limited surface area of its core and fan case, B) the refrigeration unit for a truck trailer, C) an environmental control system providing pressurization and cooling to commercial aircraft cabin air, and D) helicopter avionics hardware, interconnected by wire harnesses and thermal management pipes and ducts to reject electronics heat, presenting accessibility challenges in the front avionics bay. Figure courtesy: Wikipedia public-domain (copyright-free) images. 5

1.2 Cabinet level (automotive electrical and cooling system, aeroengines, etc.), and device level (circuit layout board, chip scale designs, etc.) 5

2.1 Primary SPI2 design research problem elements and some practical SPI2 industry relevant applications. 14

3.1 2D geometric layout design representation with core problem features. 29

3.2 Two-stage design framework for 2D spatial packaging. 30

3.3 Illustration of equivalent and distinct spatial topologies. . . . 30

3.4 Initial and final graphs attained after force-directed layout implementation. 33

3.5 Force representations on system components. 35

3.6 2D Force-Directed Layout Representation. 35

3.7 Initial and final layouts for 10-component (**(a)** and **(b)**), and 15-component (**(c)** and **(d)**) systems respectively with FDLM described under Sec.3.6.2. 37

3.8 Initial and final layouts attained with FDLM for a 10-component system using fully connected and nearest neighbours graph networks. 38
3.9 Initial and final layouts attained with FDLM in Sec.3.6.3 for a 3-component and 5-component interconnected systems. . 39
3.10 Initial and final layouts attained with the shortest path A* method for a 4-component 2-port system. Sequence number of paths for each layout are shown. 42
3.11 4 Unique Spatial Topologies for a 3-component system. ST-A is a directly connected configuration. ST-B, -C, and -D are indirectly connected configurations. 42
3.12 Illustration of the shortest path A* algorithm on a denser mesh. 43
3.13 UST method illustration: (a) Five unique spatial topologies of a 4-component 2-port system; and (b) offset representation. 44
3.14 UST method demonstration to enumerate feasible layouts with distinct spatial topologies. 44
3.15 FDL-FETS method illustration with a random 8-component 2-port system. 47
3.16 A result from Case Study 1 with a 6-component 2-port system showing an **(a)** initial layout; and **(b)** final layout attained using FDL-NNG method; **(c)** shows SPA final layout; and **(d)** an indirect UST final layout. 48
3.17 Results for **(a)** Average computational times required to attain a single feasible layout for 200 simulations for nC-2P systems; and **(b)** Box plot of computational times taken by different algorithms to attain a feasible layout for a 10C-2P system. 49
3.18 Illustration of final layouts obtained using FDLM with arbitrarily-shaped bounding boxes without and with restricted regions respectively. 51
3.19 Illustration of final layouts obtained using SPA and UST methods respectively with arbitrarily-shaped bounding boxes *not containing restricted regions*. 51
3.20 Illustration of final layouts obtained using SPA and UST methods respectively with arbitrarily-shaped bounding boxes *containing restricted regions*. 52
3.21 Average success rates to a attain feasible layout over 500 simulations for nC-2P systems. 52

4.1 Bar projection 60
4.2 component projection 61
4.3 Pipe elbow geometry 64

4.4 Geometric constraints . . . 75
4.5 2D schematic of an unmanned aerial vehicle (UAV). . . . 78
4.6 Results of hybrid UAV power electronics cooling system illustrative study: (a) Initial layout; (b) final optimal layout for head loss objective function; (c) final optimal layout for bounding box objective function. . . . 79
4.7 Boundary conditions for the thermal finite element analysis. . 80
4.8 Objective function convergence history and first order optimality plots of (a) a pressure head loss function; and (b) bounding box objective function. . . . 88
4.9 Initial layout of a UAV power electronics cooling system. . . . 89
4.10 The 2D schematic diagram of the automotive fuel cell (AFC) system with all its components and their interconnections. . . 89
4.11 Initial and final optimized AFC layout with a four subsystems' layout. The corresponding subsystems have been labelled. . . . 90
4.12 Initial and final optimized AFC layout with a six subsystems' layout. The corresponding subsystems have been labelled. 90
4.13 Bounding box objective function values for nC-2P systems. . . 91
4.14 Corresponding initial and final optimal 6C-2P layouts of designs (a)-(d) indicated in Fig. 4.13 for bounding box objective function. . . . 91
4.15 Pressure (head loss) objective function values for nC-2P systems. . . . 92
4.16 Corresponding initial and final optimal 6C-2P layouts of designs (a)-(d) indicated in Fig. 4.15 for pressure head loss objective function. . . . 92
4.17 Initial and final optimal layouts for a 5C-2P interconnected multi-component system for four selected designs from the Pareto set. . . . 93
4.18 Bi-objective optimization Pareto plot for a 5C-2P interconnected multi-component system. . . . 93
4.19 Initial and final optimal layouts for a two-loop interconnected multi-component system. . . . 93

5.1 A simple illustration of 3D system architecture and spatial topology concepts. 3D systems A and B1 have different system architectures. Systems B1 and B2 have the same system architecture but are two distinct spatial topologies. . . 96
5.2 Figure **(a)** represents a five component interconnected 3D system; **(b)** is the shadow of the 3D diagram; and **(c)** shows three distinct spatial graph diagrams (SGD1, SGD2, and SGD3) . . . 101
5.3 Fundamental Reidemeister moves for spatial graphs. . . . 103

5.4 SGDs A and B are topologically equivalent θ_1 graphs under Reidemeister-I (RI) move. C is a θ_2 graph and is topologically distinct from A and B under any fundamental R moves. 103
5.5 At left is a typical flat vertex graph. The Reidemeister $R5$ move on this kind of spatial graph is shown at right. 103
5.6 At left is a typical ribbon graph; notice the twists in the topmost band. The upper-right shows how a SGD encodes a ribbon graph via the blackboard framing, with the vertical isotopy showing how loops in the SGD can describe twists in the ribbons. 104
5.7 Class of spins for a crossing. 104
5.8 Identifying connected planar spatial diagrams from the combinatorially enumerated set is performed using this procedure. 111
5.9 The six sequential steps of the proposed spatial graph-based topology enumeration framework. 112
5.10 Results from case study 5.5.1 for a given system architecture for maximum crossing numbers from 0 through 3. 114
5.11 Results from case study 5.5.3 for a given system architecture for maximum crossing numbers from 0 through 3 with components having different valencies. 116
5.12 Implementation of the circular spatial graph representation technique in Sec. 5.5.4 to avoid unnecessary or extra twists between any two edges of a diagram. 117
5.13 Demonstration of spatial graph diagram decomposition approach discussed in case study 5.5.5. 119
5.14 First 30 braid words for a set of 4-braid with maximum 3-crossing representations. 121
5.15 Braid-closure based spatial graph representations for $N = 5$ strands and $K = 6$ crossings system with two bivalent and trivalent components. 124

LIST OF ABBREVIATIONS

PR	Packing and Routing
SE	Systems Engineering
SPI2	Spatial Packaging of Interconnected Systems with Physics Interactions (SPI2)
HEV	Hybrid Electric Vehicle
UAV	Unmanned Aerial Vehicle
AFC	Automotive Fuel Cell
GPM	Geometric Projection Method
CAD	Computer Aided Design
ST	Spatial Topology
SGD	Spatial Graph Diagram
SA	System Architecture
GPM	Geometric Projection Method
UST	Unique Spatial Topology
FDLM	Force Directed Layout Method
SPA	Shortest Path Algorithm
SPM	Shortest Path Method
HBB	Hard Bounding Box
IC	Interconnect
LHS	Latin Hypercube Sampling
FPGA	Field-programmable Gate Array

VLSI	Very Large-Scale Integrated
AFC	Automotive Fuel Cell
PD	Planar Diagram
PC	Planarity Check
RAM	Random Access Memory
CS	Case Study
OSPP	Optimal Spatial Packaging Problem
AC/DC	Alternating Current/Direct Current
PES	Power Electronic Systems
A^*	A-Star
FDA	Force-directed Algorithm
FCG	Fully Connected Graph
NNG	Nearest Neighbours Graph
DP	Dynamic Programming
FETS	Free Expansion Then Shrinking
MAPF	Multi-agent Path Finding
MDTO	Material Distribution Topology Optimization
EMI	Electromagnetic Inference
DA	Design Automation
FVG	Flat Vertex Graph
APF	Artificial Potential Field
TRL	Technology Readiness Level

LIST OF SYMBOLS

A^*	A-star algorithm
v	Node (vertex) of a force-directed graph
f_a	Attractive force factor
f_r	Repulsive force factor
d_{uv}	Distance between any two graph vertices
l	Natural spring length
l_{uv}	Spring length between two vertices
$k_{uv}(1)$	Spring stiffness
$k_{uv}(2)$	Repulsive force strength
F	Total spring network force
$l_{v_i}l_{v_j}(1)$	Desired distance between two spring network vertices v_i and v_j
$k_{v_i}l_{v_j}(1)$	spring stiffness between two vertices
r_i	radius of a bounding circle in a force-direct graph
$f(n)$	Total cost of path in shortest path algorithm
$g(n)$	Exact cost of the path from source to any cell n
$h(n)$	Heuristic estimated cost from empty cell n to the target
$\boldsymbol{\kappa}$	The matrix of thermal conduction coefficients
$T(\boldsymbol{x})$	The temperature solution field,
Q	Heat flux per unit volume in the domain,
$\boldsymbol{n}$	Unit normal to the domain boundary

T^*	Temperature of the domain
q^*	Domain heat flux
Γ_T	Domain temperature boundary condition
Γ_q	Domain heat flux boundary condition
$\mathbf{x}$	Design vector
$\mathbf{T}$	Temperature field vector
$\mathbf{K}$	Global thermal stiffness matrix
$\mathbf{k}_{\text{el}}$	Element stiffness matrix
$\mathbf{P}$	Global load vector
$\mathbf{p}_{\text{el}}$	Element load vector
$\boldsymbol{N}$	Element shape function vector
$\boldsymbol{B}$	Shape function gradient vector
h	Convection coefficient
T_{env}	Temperature of the convective fluid
ρ_i	Material density parameter with a value between zero and one
p	Penalization parameter used to penalize intermediate densities between 0 and 1
$\rho_{\min}$	A minimum density, that is enforced to prevent singularity in the global stiffness matrix
$\boldsymbol{x}_0$	Bar segment start points
$\boldsymbol{x}_f$	Bar segment end points
w	Bar width
d_q	Distance between the segment q and point $\boldsymbol{p}$
$\boldsymbol{c}_d$	Reference point for each component
$\boldsymbol{b}_i$	Set of vectors pointing from the reference point to polygon vertices
$\tilde{\rho}_e$	Edge density
$\boldsymbol{x}_{e_0}$	Edge segments start points vector

$\boldsymbol{x}_{e_f}$	Edge segments end points vector
Q_d	Internal heat generation of each component
H_L	Head loss within a pipe section
V	Velocity of fluid between two points in a pipe
P	Fluid pressure
ρ_w	Weight density
g	Gravitational acceleration
Q	Volumetric flow rate
A_i	Cross sectional area of a pipe at location i
K	Head loss coefficient
l_i	Length of pipe
d_i	Diameter of pipe
α_i	Bend angle at an elbow junction
$\dot{w}$	Weight flow rate
r_i	Bend radius at an elbow junction
ε_s	Small perturbation
l_c	Clipped length - to account for the length of straight pipe that is overlapped by an elbow
θ	Supplementary angle
f_i	Friction factor
Re	Reynolds number
ρ_m	Mass density
μ	Fluid viscosity
z	Reduced design vector
$\boldsymbol{M}$	Binary mapping (or selection) matrix
$\boldsymbol{P}$	Vector represents fixed components that are not in the reduced design variable vector, but have a fixed value throughout the optimization

$\boldsymbol{p}_i$	Port locations vector
$\boldsymbol{T}$	Temperature vector (state variables)
$f(\boldsymbol{x}, \boldsymbol{T})$	Objective function
$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{T})$	Constraint function
$\boldsymbol{x}$	Design variable vector
$\boldsymbol{g}_{\text{phys}}(\boldsymbol{x}, \boldsymbol{T})$	Physics-based constraints - they depend both on design and on solutions to physics models
$\boldsymbol{g}_{\text{dd}}(\boldsymbol{x})$	Interference constraints between two components
$\boldsymbol{g}_{\text{sd}}(\boldsymbol{x})$	Interference constraints between one routing segment and one component
$\boldsymbol{g}_{\text{ss}}(\boldsymbol{x})$	Interference constraints between two routing segments
$\boldsymbol{\Psi}$	The adjoint vector
$\boldsymbol{R}$	The residual vector
$\boldsymbol{P}^p$	Flux at nodes where prescribed boundary conditions are applied
$\boldsymbol{T}^f$	Temperatures at all remaining nodes
$\boldsymbol{T}_{\text{el}}$	Vector of element nodal temperatures
N_e	Number of finite elements in the mesh
ρ_e	Element density
ρ_{de}	Component element density
$\boldsymbol{\Psi_e}$	Adjoint vector corresponding to element degrees of freedom
$\boldsymbol{T}_e$	Temperature vector corresponding to element degrees of freedom
r^b	The radius of a bounding circle at the component reference point
$A(x, y)$	Bounding box area objective function
$T_{\boldsymbol{d}_{max}}$	Maximum component (device) temperature constraint vector
$T_{\boldsymbol{f}_{max}}$	Maximum fluid temperature constraint vector
H_{l_c}	Maximum head loss constraint
$\boldsymbol{g}_{\text{geo}}(\cdot)$	Vector of geometric constraints (includes all types of interference constraints)

δ_x	Prescribed design vector tolerance
δ_f	Prescribed objective function tolerance
G	Abstract graph
A	A is an indeterminate (arbitrary independent variable)
$H(G)(A)$	Polynomial invariant of an abstract graph G
$G_1 \sqcup G_2$	Denotes the disjoint union of graphs G_1 and G_2
$G_1 \vee G_2$	Denotes a wedge at a vertex of two graphs G_1 and G_2
e	Non-loop edge of a graph G
g	Spatial graph diagram
c	Crossing in a spatial graph
s_+, s_-, and s_0	Graph reduction with class of spin +1, -1 and 0 respectively.
S	Planar graph - state on g
$U(g)$	Set of states on g obtained by applying all possible reductions in its crossings
$R\{g\}(A)$	Yamada polynomial invariant
$\tilde{G}$	Spatial graph
k	crossing number of a spatial graph diagram
k_m	Maximum crossing number

CHAPTER 1

INTRODUCTION

1.1 Motivation

High power density electric machines [1,2], automotive cooling systems [3], electric drives [4], aircraft engines [5] (and many other types of other engineering systems) are composed of components that exchange energy, as well as routing (interconnects) that facilitate energy transfer. One reason these systems are difficult to design is that they have many requirements, including: performance, cost, geometry, and volume restrictions. Identification of feasible designs can be exceptionally difficult in applications where the available physical design space is limited, components and interconnects involve complicated geometries, and system performance depends on spatial relationships and multiple physics couplings. Current practice relies largely upon human expertise, design rules, modification of existing designs, and manual adjustments to solve these problems. This limits both the complexity of systems that can be designed involving non-trivial packing and routing (PR) decisions, as well as the realization of potentially improved functionality or performance. Many previous efforts have focused on creating design automation methods to address elements of the integrated packing and routing problem described here, but not the combined problem. In this dissertation a novel 2-stage design automation approach is presented that integrates several of these elements as a step toward a more comprehensive solution of 3D packing and routing problems with both geometric and physics considerations. Development of such methods for the formalization, automation, and prioritization of topological layouts for complex, multi-domain systems will offer benefits across several industry sectors. This chapter introduces the spatial packaging of interconnected systems with physics interactions (SPI2), various other related terms, and motivates the need for systematic design au-

tomation methods in this field of research. Furthermore, the major research objectives of this dissertation are outlined, and a thesis overview is presented that briefly discusses the role of various chapters in this dissertation.

1.2 Some Important Terms

This section provides descriptions to several terms that will be widely used in various sections of this dissertation. A *system* can be defined as the combination of heterogeneous elements such as components, interconnections, equipment, etc., that function together to produce the capability required to meet a specific need. Parts of a system are referred to as *subsystems*, they can be classified based on functionality, spatial proximity, physics-type, etc. The performance of a system as a whole, beyond that contributed independently by its parts, is primarily created by the relationship among the parts; that depends both on how they are interconnected to and interacting with each other in a given physical environment. *Systems engineering (SE)* is a methodical, multi-disciplinary process that supports the design and development of an optimized system capable of meeting several performance requirements within multiple, often conflicting constraints. In other words, SE is an integrated discipline where different engineering domains' decisions are combined in a holistic and balanced manner to achieve stakeholder functional, physical, and operational performance requirements in the intended use environment over the planned life-cycle of the system within cost, schedule, and other constraints. *Systems engineers* are personnel skilled in the art and science of system engineering. Their key job is to lead in the development of the concept of system operations, define boundaries, integrate sub-systems, analyze interfaces, evaluate design trade-offs, and perform verification and validation for final product design. A *systems approach* in design considers the attributes of the entire system to achieve certain system-level performance objective(s). This approach is very challenging because it involves understanding the functionalities and interaction between sub-systems in a synthetic view while simultaneously analyzing individual sub-subsystems.
Design representation provides a way to describe the different features of an engineering system design problem. This representation is generally per-

formed using a model that embodies the functionality and relationships of various components in a system. Design representations can be mathematical, graphical, physics-based, conceptual or analytical. *Design methods* refer to the procedures, steps, or tools to aid the development of systems. Design methods can be developed either to address a unique design application or could be generic to support a *class of systems* that have similar functionalities. A *design framework* is a collection of design methods that are either implemented sequentially, simultaneously or in a nested manner for achieving final system designs. *Design theory* describes the set of design guidelines and principles on which the design practice is based. In other words, design theory, when combined with a purpose (function) or problem to solve, results in effective design solutions.

1.3 SPI2 Problem Definition

The Spatial Packaging of Interconnected Systems with Physics Interactions (SPI2) problem can be defined as optimal spatial arrangement of heterogeneous geometric components and interconnects of often non-trivial sizes inside irregular three-dimensional volumes, along with the consideration of their physics-based behavior, life-cycle processes, and system operating conditions. These design problems cut across a wide swath of engineering domains that are vital to society (e.g., medical devices, transportation, and computing hardware), and entail especially complex design spaces (combining complex combinatorial/topological, geometric, parametric, and time-dependent decisions) that are difficult to navigate either via expert human cognition or computational search. These have resisted holistic treatment by potentially powerful design automation methods, and still rely largely on manual spatial placement by designers supported by computer-aided design (CAD) tools. Solving the SPI2 problem requires highly-skilled engineers who understand the engineering application, manufacturing, testing, assembly, maintenance, and repair requirements. Moreover, design and maintenance of large-scale systems such as aircraft and ships requires thousands of man hours, and any advancement to overcome this bottleneck has potential for significant technical and economic impact. The SPI2 design problem attributes

are outlined as follows:

1. Fundamentally 3-dimensional; involves interconnected components with complex spatial geometries and often complex, irregularly shaped enclosing volumes;

2. Interconnects of various types (pipes, ducts and/or wires, etc.), sizes, shapes, and requirements (curvature, proximity, temperature, electromagnetic interference (EMI), etc.)

3. Strongly-coupled physics interactions (thermal, hydraulic pressure, electromagnetic, etc.) and influence of spatial arrangement on performance

4. Interconnect complexity (both topological and spatial), as illustrated in Fig.1.1

5. Unlike 2D systems, the 3D-SPI2 problem contains objects (components, casings, bays, etc.) that are either solid or have holes or spatial-access ports. This makes the topological problem more interesting. For example, an interconnect may pass through a hole in a component, or bypass the hole and be routed between components.

6. SPI2 designs can be characterized based on scale of the relevant applications. For example, Fig. 1.2 shows examples of SPI2 design applications at the *cabinet* level (automotive system electric layout, aero engine, etc.) and *device* level (3D chips design).

7. Value metrics: packaging density (volumetric), product life-cycle and maintenance costs, efficiency measures, system reliability, and physics-based objectives such as maximum system temperature variance, pressure head loss, electrical energy/power efficiency, etc.

8. Constraints: geometric (for ensuring both feasibility and connectivity), physics-based (device/fluid temperature, fluid velocity, etc.), failure modes, etc.

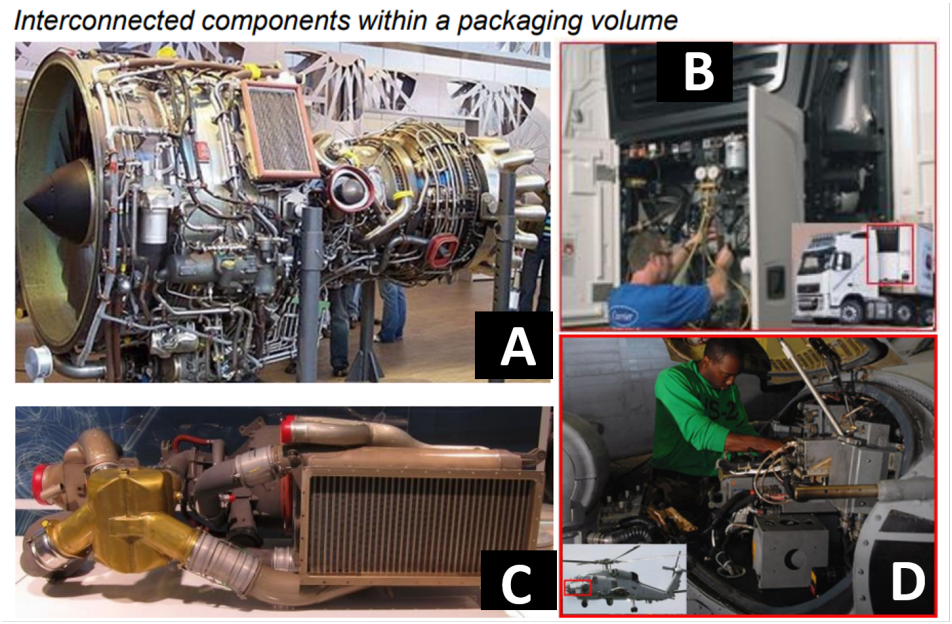


Figure 1.1: Diverse examples of typical manually designed systems presenting SPI2 spatial packing and routing complexity, subject to physics interactions, and exhibiting spatial accessibility challenges for life-cycle processes: A) the externals (components, wires, pipes and ducts interconnecting components and engine features) of a commercial turbofan engine covering the limited surface area of its core and fan case, B) the refrigeration unit for a truck trailer, C) an environmental control system providing pressurization and cooling to commercial aircraft cabin air, and D) helicopter avionics hardware, interconnected by wire harnesses and thermal management pipes and ducts to reject electronics heat, presenting accessibility challenges in the front avionics bay. Figure courtesy: Wikipedia public-domain (copyright-free) images.

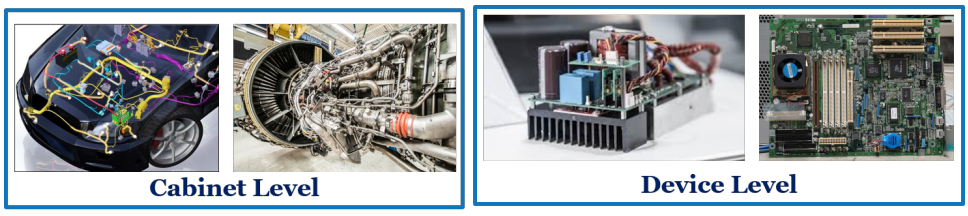


Figure 1.2: Cabinet level (automotive electrical and cooling system, aero-engines, etc.), and device level (circuit layout board, chip scale designs, etc.)

1.4 SPI2 Design Research: Stuck since Decades

A high demand exists in present society to create systems that are increasingly compact, while providing enhanced technical capability to realize benefits such as reducing emissions [6], increasing energy efficiency [7], and improving economic competitiveness. New SPI2 design automation methods are needed that can reduce the size of complex systems considerably, impacting applications such as power-dense smart batteries [6], spacecraft cooling systems [8], minimally-invasive medical wearables [9–11], strategies for improving vehicle spaciousness [12], and compact avionics and military electronic systems [13,14]. Engineers have labored for decades to improve spatial packing density across diverse domains such as in avionics [15–18], spacecraft systems [19], automotive packaging [20], vehicle electrification [3], and spacesuit design [21,22]. These advances, however, have largely been incremental, and have depended heavily on expert human ingenuity.

Current practice relies heavily on design heuristics and time-intensive human activities. Computer-Aided Design (CAD) tools provide support, but packing and routing decisions still rely primarily on manual design techniques as described in Refs. [3,18,20]. Product design engineers and skilled industry personnel who understand the engineering applications, manufacturing, testing, assembly, maintenance, and repair requirements use legacy design methods for product development. In most cases, they start with an existing system design and adapt it to meet new requirements. This evolutionary process is not only very time-consuming, but usually leaves available performance enhancements on the table, such as improved density, system performance, or reliability. The associated design space involves many interrelated geometric constraints and physics-based considerations, which are cognitively difficult to navigate effectively. For these and related reasons (such as those articulated in Refs. [11,13,14]), SPI2 design practice has been stuck for decades. As demand for higher-complexity systems grows, SPI2 design is further complicated in that the number of possible arrangements increases exponentially with the number of components and interconnects. Several inefficiencies, including biased human errors, lead to inferior designs, and these inefficiencies may increase as product requirements grow more complex and required packaging volumes shrink.

1.5 Complexity of SPI2 vs. VLSI

Significant work has been performed in related areas, such as 2D VLSI (Very-large-scale integration) circuit component layout, design and routing optimization [23, 24] for several board-based electronic applications. The VLSI design problem that has been automated successfully has limited degrees of freedom compared to more general SPI2 design problems. VLSI problems, in general, are primarily 2D applications with planar rectangular or square simple geometric surfaces [25] and do not involve extensive physics coupling between components and/or interconnect networks. Moreover, manually it is possible to design VLSI circuits both intuitively and by experience using existing design tools, and is possible to estimate their efficiency or other related performance metrics. In contrast, real-world SPI2 design problems are 3-dimensional (which adds a another layer of complexity) and have multiple diverse attributes such as components with complex spatial geometries (either concave or convex), restrictive domains, arbitrarily-sized, irregularly-shaped bounding volumes, interconnects of various types (pipes, ducts and/or wires, etc.) and radii, possible topological network configurations, strongly-coupled physics interactions (thermal, hydraulic pressure, electromagnetic, etc.), often large scale, and frequently encompass several other design challenges. To make accurate SPI2 design decisions based on human judgement, or even using limited software tools, one cannot easily attain superior optimal designs or quantify their performance. Another consideration is that SPI2 systems need to be designed with spatial accessibility to support safe and efficient manufacture, assembly, maintenance, diagnosis, overhaul, repair, upgrade, replacement, and other complex requirements.

1.6 Need for SPI2 Design Automation

Based on the discussion presented in the previous sections, it is evident that no single industry or application domain can afford to tackle this large common packaging problem. It requires diverse and deep expertise and tight, interdisciplinary engagement. For example, new mathematical SPI2 representations must integrate with physics considerations, detailed geometric analysis, as well as navigation of formidable spatial topology (ST) decision

spaces. Spatial topologies involve how interconnects (such as ducts, pipes or wires) pass through or around other elements in a space, and these decisions represent one of the most difficult elements of SPI2 problems.

Effective design automation (DA) strategies are key to meeting the demands of present and future needs for SPI2 packing and routing problems. Automated, systematic, flexible, and efficient design methods with the ability to explore and access new system configurations are essential for achieving better designs across different engineering industries. Effective methods will support adjustments that can be made easily if the product requirements change over time. An important potential benefit of realizing such methods is the reduction in design time and resources required to solve packing and routing problems, enabling greater tailoring of designs to enhance performance for unique applications, while greatly reducing design effort. Applications and potential impact of SPI2 design automation are expansive, and associated practical design methods stand to impact industry and society through: 1) drastically reducing engineering effort and time to design new SPI2s, 2) opening the door to the design of SPI2s with new levels of complexity, with the potential for new functionality and performance.

1.7 Research Objectives

The goal of this dissertation is to create a strong scientific foundation to solve the grand SPI2 design problem with a systems approach. However, to create such a foundation requires two significant efforts: 1) an in-depth study and development of an automated design framework to address core SPI2 related topics such has component packing, interconnect routing, physics-based optimization, etc., in a combined manner, and 2) to simultaneously understand the broader SPI2 design problem, its attributes, existing critical gaps, associated design challenges, and practical application areas. To fulfill the above needs, this dissertation aims to accomplish the following major research objectives:

RO1. To define the optimal spatial packaging problem, identify its core research domains, and their interfaces that exist between them.

RO2. To develop a systematic, tractable, and an automated design frame-

work to solve the combined packing, interconnect routing, and physics performance problem.

RO3. To create new SPI2 design representations that can capture the SPI2 problem attributes, and are simple, flexible, and scalable to handle larger-complex SPI2 application problems.

RO4. To develop efficient enumeration techniques to generate unique spatial topologies of both 2D and 3D SPI2 design problems. Furthermore, investigate techniques to help classify these topologies according to system complexity.

RO5. Identify a collection of layout generation methods that can be utilized for generating feasible (interference-free) layouts for different SPI2 problem scenarios.

RO6. Develop a design optimization framework to search design space effectively, aiding both discrete and continuous decision making.

RO7. Develop a design method that can simultaneously perform physics-based performance optimization while satisfying spatial feasibility, packing density, and geometric connectivity constraints.

RO8. Comprehensive evaluation of the 2D SPI2 design problem and automation methods as a means to develop insights and a foundation for tackling the more complex 3D SPI2 problem.

RO9. To generate new design knowledge that 1) aids systems engineers to understand the SPI2 problems comprehensively, 2)supports industry practice, and 3) creates SPI2 software design tools that generate various SPI2 system geometric models.

The broader goal of these above research objectives is to not only create methods to solve the SPI2 design problem, but also attempt to answer some larger SPI2 design research questions, identified as follows:

1. Characterization of the SPI2 design space: What are its boundaries? How do the feasible and infeasible design space regions compare with each other?

2. Is the SPI2 design space generic, or does it need to be classified according to system size (e.g., number of components), complexity, physics performance requirements, product life-cycle cost value metrics, or other dimensions?

3. How does SPI2 design difficulty scale with increase in the number of components, constraints, complexity, etc?

4. What kinds of design tools does industry require to adopt SPI2 design automation methods?

5. What unified design parameterizations/representations are needed to solve the SPI2 optimization problems efficiently?

6. How might various product-life cycle value metrics such as maintenance, upgrade, overhaul, repair, and accessibility costs be incorporated as part of the SPI2 problem formulation and automated solution?

Based on the key findings of this dissertation, a detailed discussion in the conclusion (Chapter 6) is presented that addresses some of these above questions.

1.8 Dissertation Overview

The introduction motivates the need to systematically solve the optimal SPI2 problem to design complex engineering systems in an efficient manner and perhaps attain much greater system-level performances than already existing design solutions. This chapter has provided the definitions of some important terms that will be used throughout this dissertation. The SPI2 design problem was then defined, and its diverse problem attributes were outlined. A detailed discussion was presented on how the SPI2 design research has been effectively stuck for decades, impacting myriad practical industry applications. Furthermore, the complexity of the SPI2 problems was compared with the traditional VLSI circuit design problems that have well-established design automation methods. The need for design automation strategies for efficiently solving the SPI2 design problems was then discussed. Finally, the main research objectives of this dissertation were outlined. Considering the

diverse design elements of the SPI2 design problem solved in this dissertation, a number of design methods addressing different aspects of this design process are presented in the following chapters. Chapter 2 provides a broad and in-depth review of existing SPI2 design methods in related research domains. The focus in Chapters 3-5 is both on the generality of the proposed design methods and on their implementation to tailored case studies that fit within typical SPI2 design problem categories. The remainder of this dissertation is organized as follows.

- Chapter 2 presents a state-of-the-art review in four fundamental SPI2 design related research topics: component packing, interconnect routing, physics-based topology optimization, and mathematical design representations. A list of existing critical gaps across the interfaces of these core research topics with respect to SPI2 design are discussed. The later chapters of this dissertation address a majority of those gaps. Furthermore, the associated design challenges in SPI2 design at the intersections of these areas are also presented in Chapter 2.

- Chapter 3 concentrates on solving the 2D SPI2 design problem. The 2D SPI2 design problem is aptly framed to attain two major outcomes: 1) to comprehensively solve the real-world planar layout SPI2 design problems, and 2) create a foundation to solve the 3D SPI2 design problem based on the insights attained from the 2D SPI2 design process. In this chapter the similarities and differences between the 2D and the 3D SPI2 design problems are presented. Then the proposed two-stage SPI2 design automation framework is introduced. The bulk of Chapter 3 is focused on the implementation of stage 1 of the two-stage design framework to 2D SPI2 design problems. Three efficient feasible 2D layout generation methods are presented, each of them having their own unique characteristics. A number of case studies are performed to both compare and perform a thorough analysis of the three methods for different kinds of 2D SPI2 problems. The results of the case studies are used to understand the general trends of these layout generation methods as the scale and system complexity increase. Finally, a set of practical guidelines are provided for design engineers on how to unitize these methods for generating feasible layouts for different 2D SPI2 problems.

- Chapter 4 focuses on stage 2 of the two-stage design framework. The geometric projection method (GPM) is applied on 2D SPI2 layouts (attained from stage 1) for performing multi-physics packing and routing optimization. A bulk of this chapter discusses the 2D layout modeling in detail. A number of case studies are presented that address different aspects of the 2D SPI2 problem. Stage 2 demonstrates the incorporation of physics-based objectives and constraints on multi-loop layouts and industry-relevant applications. This chapter shows the efficacy of the two-stage design framework when systematically applied to 2D SPI2 design problems.

- Chapter 5 focuses on the 3D SPI2 design problem. Stage 1 of the design problem is rigorously investigated in this chapter. Hence, a bulk of this chapter presents a novel design framework to represent, enumerate, and identify unique 3D spatial topologies of a given 3D system. Spatial graph and braid theories are extensively utilized to attain spatially-feasible 3D system designs. A number of case studies are provided along with design guidelines to tackle the challenging 3D SPI2 design problems. Stage 2 of the two-stage framework for the 3D SPI2 problem is considered out of the scope for this dissertation.

- Chapter 6 presents the dissertation summary and outlines a list of potentially impactful future research directions.

CHAPTER 2

REVIEW: SPI2 DESIGN RESEARCH AREAS

The SPI2 design problem consists of different intricately-related research elements that are individually very challenging themselves. For example, both the component packing and interconnect routing are two separate NP-hard problems. Existing design methods treat the pieces of this challenging SPI2 problem separately without a fundamental systems approach and are sometimes too slow to evaluate a practical set of possible designs. Hence, there exists an emergent need to develop efficient SPI2 design automation frameworks for two reasons: 1) to enable the rapid generation and evaluation of candidate SPI2 design solutions; and 2) for the development of newer complex engineering systems. In Chapter 1, the holistic 3D-SPI2 design problem with various its attributes was defined. In this chapter[1], previous research efforts in four fundamental SPI2 design research areas are reviewed, existing critical gaps are outlined, and associated challenges are identified.

2.1 Four SPI2 Research Elements

The most important areas that are an integral part of SPI2 design research are shown in Fig. 2.1. The SPI2 problem can be subdivided into four basic sub-problems: 1) component packing, 2) interconnect routing, 3) physics-based topology optimization, and 4) SPI2 mathematical design representations. It should be noted, however, that there could be other similar domains that might interact with these areas and could impact SPI2 design research. This dissertation focuses primarily on these above four research areas as they fundamentally address the SPI2 problem directly. State-of-the-art-methods and work done in each of these areas related to SPI2 design are reviewed in the following sections.

[1]Some elements of this chapter are published in Ref. [26]

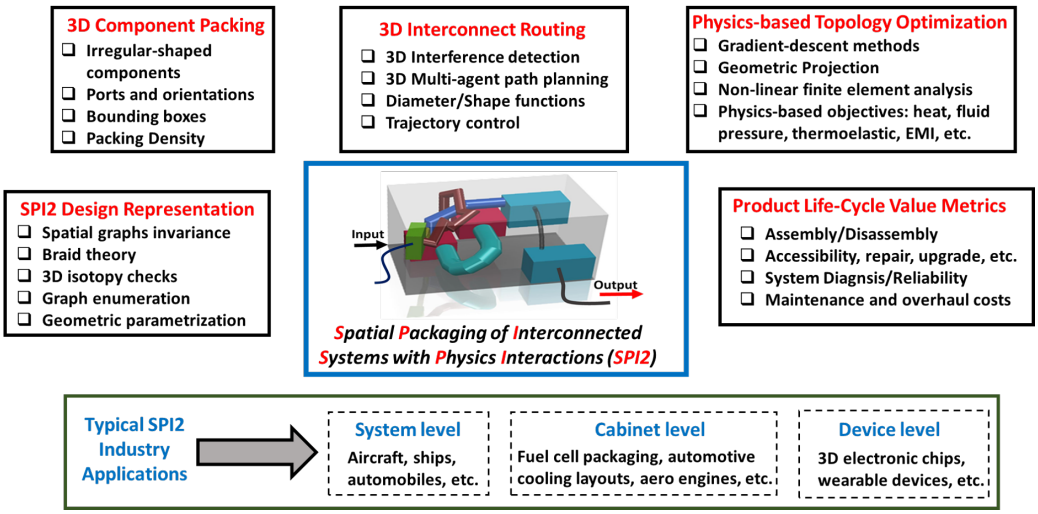


Figure 2.1: Primary SPI2 design research problem elements and some practical SPI2 industry relevant applications.

2.2 3D Packing and Component Layout Design

3D component layout is a 3D bin-packing problem that comes under the class of optimization problems in mathematics that involves optimally placing and orienting objects within a given 3D volume, or optimally reducing the volume within which they can fit. Packing problems are NP-hard. Typical engineering systems products are a combination of functionally and geometrically interrelated components. The spatial location and orientation of these components affect a number of physical quantities of interest to the designer, engineer, manufacturer, and the end user of the product. The 3D component packing framework concerns itself with determining the optimal spatial location and orientation of a set of components given some design objectives and constraints. It models the layout problem as a volume minimization problem with the objective function being a weighted sum of the design objectives and penalties for constraint violation. The design objectives can include a quantification of a variety of measures such as the amount of cable used in the engine compartment of a car, or the component packing density in an electric drill, or the center of mass of a space vehicle. The most significant constraint is the non-intersection of components and non-protrusion of components outside the design space. Other constraints include spatial relationships between components (e.g., co-axially mounted on a shaft) and between a component and the design space (e.g., gravity-based orientation

of fluid reservoirs).

The 3D packing problem in most practical cases is quite general [27], where component geometries can be arbitrary, and multiple types of design goals and spatial constraint satisfactions could exist. For practical purposes, the minimization of layout cost functions is done under certain constraints imposed by design, fabrication, and operational requirements. Most layout algorithms are restricted to a certain class of systems and are as a whole intractable due to their combinatorial nature. Problem variants differ by the particular definition of their packing constraints (presence of guillotine cuts, balancing and stability of the packing, possible overlapping of certain items, forbidden rotations of the items, etc.) and objective function, going by the well-known names of knapsack, bin packing, strip packing, variable-sized pellet packing, container loading, etc. Design automation methods for solving the optimal spatial packing problem have been developed and studied previously in the context of many applications, such as vehicle assembly [28], electronic module layout design [29], 3D container loading [30], bin packing [31], computer animation [32], the layout of components in additive manufacturing [33], and automotive transmission design [34]. Solution algorithms used in previous 3D component packing research can be generally classified under four categories: genetic algorithms [35], heuristic methods [36], gradient descent algorithms [37], and simulated-annealing algorithms [38].

2.3 3D Interconnect Routing

The 3D interconnect or pipe routing problem is a common industrial problem that is solved for designing layouts in chemical process plants, oil and natural gas refineries, water treatment and distribution plants, hydroelectric power plants, and other applications. Designing a 3D pipe layout involves two major tasks. First is equipment allocation, i.e., finding the 3D coordinate locations of all equipment to minimize total cost and satisfy all the equipment constraints such as maximum distances and maintenance access requirements. In this task, conventional design strategies employ a rough measure to evaluate the total cost based on Manhattan distances. The second task is to find 3D routes for all the pipes connecting the equipment and

to ensure that they are not colliding with each other.

Since the 1970s, interconnect routing design has been studied in various industrial fields, such as transportation, large-scale integrated circuits, and computing hardware. It is one of the most important processes for system integration. However, due to the complexity of routing systems and the diversity of constraints involved, it is quite time-consuming and difficult to achieve a feasible layout using both manual experience and CAD-based design tools. Systematic studies in route path planning have been carried out by researchers for several decades. Dijkstra's algorithm, proposed in 1959 [39], is a well-known algorithm for path optimization with shortest length. Based on Dijkstra's algorithm, another heuristic algorithm was developed to improve search efficiency, and is defined in Ref. [40]. In 1961, Lee [41] proposed a maze algorithm to solve the problem of connecting two points. Further search efficiency developments were presented in Refs. [42, 43]. Recently, research on route path planning has been promoted by the development of modern optimization algorithms such as genetic algorithms [44], ant colony algorithms [45, 46], and particle swarm optimization [47, 48]. For example, a genetic algorithm was used to optimize routing in three-dimensional space [44]. In addition, CAD-design based routing algorithms [49–51] were also applied in 3D pipe routing design.

A key observation here is that many efforts have addressed the interconnect routing problem alone, where the component layout is fixed, and do not holistically incorporate other SPI2 problem elements. Especially in the electrical engineering domain, many examples of 2D routing algorithms were developed for VLSI circuit layouts based on Manhattan rules and its variants [52]. Other 3D routing applications include aero-engine externals routing [53], ship pipe routing [54], chemical plant pipe routing [55], electrical wire routing in buildings [56], in developing CAD-based FPGA (field-programmable gate array) design tools [57], unmanned aerial vehicle navigation [58], and robotic path planning [59]. Optimization approaches have incorporated metrics such as packaging volume and mass properties [60], and have utilized solution methods such as simulated annealing [61, 62], pattern search [63, 64], genetic algorithms [65], ant colony optimization [66], and several other heuristic methods [67].

Finally, it is interesting to note that the 3D pipe routing problem, which aims at placing non-intersecting pipes from start locations to given target locations in a known 3D system environment, is very similar to the multi-agent path finding (MAPF) problem, which is a topic of active study in robotics research. 3D MAPF methods have the potential for application to the 3D interconnect routing problem, with the distinction that the former is dependent on dynamics of agents, whereas the latter is a static problem.

2.4 Physics-based Topology Optimization

As mentioned earlier, an important aspect of the SPI2 design research is to integrate the physics interactions between the various components, interconnect flow passages, etc., as part of the geometric packing and routing optimization problem. Topology optimization, defined here as the optimal placement of material in a 2D or 3D geometric domain, does take into account models of physical behavior. This method class has been used across an extensive range of engineering domains, including to design structures for maximum stiffness [68], multi-material properties [69], or component geometries for optimal heat conduction properties [70, 71]. Problems that include multiple distinct physics domains have also been studied. De Kruijf *et al.*, Takezawa *et al.* and Kang & James performed optimization studies which included both structural and thermal conduction requirements [72–74]. The aerodynamic shape and internal structure of a wing have been optimized simultaneously [75–77] considering the interaction between aerodynamic loading and structural wing response. Topology optimization has also been used to optimize the placement of components and their supporting structure [78, 79]. This allows sections of specific geometry, such as a pattern of bolt holes, to be distributed optimally within a structure. Designs produced by topology optimization are often infeasible for traditional manufacturing methods (subtractive, formative), but often can be made using additive manufacturing [80]. The design of components that are more easily manufactured using traditional methods motivates the development of methods that optimize designs made from standard material sizes and shapes, typically using ground structure methods [81, 82]. The geometric projection

methods in Refs. [83, 84] have also been suggested to optimize structures made from stock materials. This greatly helps in preserving the rigid system geometry while performing optimization. Previous work in manufacturing constraint mapping [85] could complement geometric projection methods in optimizing systems to meet process-driven manufacturing requirements in product design.

Recent developments made in geometric projection method are highly relevant to SPI2 design research. An in-depth investigation in this dissertation involves using projection methods for 2D SPI2 design problems, and was first presented in Ref. [86]. The simultaneous physics-based packing and routing approach utilized in Ref. [87] makes significant system volume reduction possible. The projection method of Norato et *al.* [83] is extended in this dissertation to allow devices of arbitrary polygonal shape to be projected. Sensitivity analysis for this projection is provided to allow the efficient use of gradient-based optimization methods. These methods could be extended to model various combinations of physics; for example, fluid-thermal, thermal-electric or structural-fluid systems.

2.5 SPI2 Mathematical Design Representations

Another very important aspect of the SPI2 engineering design optimization problem is the design representation used for system modeling and design decision-making. Design representations must be accurate and compatible with other SPI2 problem elements. For example, SPI2 representations must integrate with physics considerations, detailed geometric analysis, as well as navigation of formidable spatial topology decision spaces via graph-based enumeration. Spatial topologies involve how interconnects (such as ducts, pipes or wires) pass through or around other elements in a space, and these decisions represent one of the most difficult elements of SPI2 problems due to their combinatorial nature. Unlike 2D systems, 3D systems contain crossings and it is important to have representations that can cater to this need directly.

It must be noted that the 3D spatial packaging problem, even without con-

sidering physics aspects, is exceptionally difficult. In solving complex design optimization problems, much depends on the mathematical representations that are used to describe the various features of this system and system classes. This dissertation identifies three important mathematical representations 1) spatial graphs, 2) braids, and 3) homotopy classes that can be used for representing 3D SPI2 design problems in a way that facilitates enumeration of different initial SPI2 designs for optimal selection processes. These representations are summarized below:

1. **Spatial graph theory:** 3D engineering system networks can represented as spatial graphs. Spatial graphs are graphs in three-dimensional space projected on a two dimensional plane. They are a natural extension of knot theory, which is the study of circles embedded in $\mathbf{R}^3$, as placing vertices on a knot can transform it into a spatial graph. While the study of knot theory has its origin in the physics of the late 19th century [88], spatial graph theory has its roots in chemistry [89, 90] and is different from graph theory because graph theory studies abstract graphs while spatial graph theory studies embeddings of graphs in $\mathbf{R}^3$ or even in other 3-manifolds [91–93]. This theory was used in polymer stereochemistry [89, 94] and molecular biology (e.g., protein folding) to distinguish different topological isomers. A 3D system can be represented as a spatial graph where components are the nodes, interconnected are the edges, and the ports as node valencies.

2. **Braid theory:** Mathematical braid theory [95] can be utilized to represent the interconnect network within a 3D system. This allows efficient enumeration of various braid-based representations of the interconnect network, thus supporting the exploration of discrete topological system configurations. Braid and knot equivalence methods are leveraged to weed out redundant topologies. Braid and knot theory representations have been successfully used in other applications such as protein folding [96], and very recently in multi-agent motion planning [97, 98], etc.

3. **Homotopy classes:** There are many applications in robot motion planning [99] where it is important to consider and distinguish between different homotopy classes of trajectories (paths followed by robots).

Two trajectories are homotopic if one trajectory can be continuously deformed into another without passing through an obstacle, and a homotopy class is a collection of homotopic trajectories. Classification of homotopy classes in two-dimensional work spaces has been studied in robotics literature using geometric methods [100], probabilistic road-map construction techniques [101], and triangulation-based path planning [102]. Reference [99] introduces a strategy for classifying and representing homotopy classes in a 3-dimensional configuration space using theorems from electromagnetism. BiotSavart's Law and Ampere's Law were used to define a differential 1-form, the integration of which along trajectories gives an invariant for the homotopy classes of trajectories. This concept of homotopy classes has been extended to defining different classes of 3D SPI2 problems into two categories: 1) system containing only solid components (closing infinite or unbounded objects), and 2) systems containing both solid and hollow components (decomposing objects with genus > 1). Such mathematical extensions are very valuable in improving design richness and flexibility.

2.6 Existing Critical Gaps

In this section, some of the most important gaps lying at the interfaces of the four main research topics related to SPI2 design are outlined:

1. **Missing holistic treatment:** The main limitation with methods used in component packing, interconnect routing, and physics-based topology optimization is that existing efforts address these problems separately, instead of in a combined manner that accounts for inherent coupling. In addition, most of the methods consider only geometric aspects of the problem, neglecting important physical system properties such as operating temperature, pressure drop, thermal loading, aerodynamic, and electromagnetic effects. Thus, existing methods may not extend well to the general coupled problem.

2. **Limits of manual design methods:** In addition, the performance evaluation of the designs obtained from existing systems is left to human designers. The amount of time required for a designer to generate

a feasible design and analyze its performance limits the ability of engineers to explore these complex design spaces within a constrained project timeline. Existing strategies can produce feasible designs, but they may not be optimal when considering all the system requirements and design couplings, and the complexity of systems that can be considered is limited. In current practice, many aspects of layout and routing problems are solved manually, which severely limits design capabilities for systems involving complex packing and routing tasks (especially in cases with strong physics interactions). This begs the need to develop efficient design automation strategies through a human-centered design approach.

3. **Systematic enumeration and search of 3D topological design space:** A critical gap is the lack of methods to exhaustively search a SPI2 design space, such as those that have recently become available for system architecture enumeration as in Refs [103–105].

   An efficient enumeration technique for navigating through the discrete 3D topology options possible for SPI2 design is required.

4. **Handling continuous and discrete elements together:** It is observed that this problem contains both continuous (spatial locations, interconnect diameter, trajectory, etc.) and discrete elements (topology options, number of components, ports, interconnects, crossings, etc.). This is very challenging for optimization problems and therefore there is a great need to have design optimization techniques that can efficiently navigate the combined space. In addition, unified geometric parameterization of both discrete and continuous variables should be performed to enhance optimization process efficiency and to aid improved problem formulations.

5. **Common design language:** SPI2 design research is along the interfaces of several engineering domains and applications. To effectively communicate design knowledge between various practitioners and domain experts, there is need for common terminology and constructs to address these problem elements.

6. **Visualization tools:** 3D SPI2 design problems have heterogeneous elements and this makes it challenging to conceptualize the design space.

Both CAD-based and virtual reality based tools are helpful for viewing and conceptualizing numerical simulations. Current tools are very limited with respect to SPI2 demands, and mostly serve the purposes of manual design activities.

7. **Human-centered design:** Human-informed design is required for developing better quality solutions to provide competitive advantage, improving end-user experience, increasing productivity and operational efficiency of product design pipelines, and for achieving greater system sustainability. Current SPI2 related research methods rarely bring human aspects to design which sometimes causes practical setbacks in industry adoption.

8. **Need for flexible design representations:** Existing design representations are developed to support specific applications and cannot be utilized for creating general design methods. Therefore, there is a need to develop more unified representations that are both compatible for modeling and can capture the various SPI2 problem features accurately.

9. **Tailored SPI2 routing algorithms:** Existing approaches widely use Manhattan rules for pipe routing, but improved SPI2 system performance requires flexible, and maybe deformable, pipe-shape routing representations depending on applications. In addition, optimal trajectory control of pipe routing has not yet been performed in the existing interconnect routing research. Achieving this capability could help satisfy several practical SPI2 constraints. For example, if a pipe should pass through some way points on its path for several practical reasons such as cooling a hot component, fluid transfer, etc., then optimal trajectory control plays a vital role.

2.7 Associated Design Challenges

The associated challenges related to SPI2 design research are identified below:

1. Both packing and routing are NP-hard problems. Therefore, as the scale and complexity of the system increases, the number of possible

solutions combinatorially explodes, increasing decision-making cost significantly.

2. The 3D topological space is vast and challenging to navigate as there can be infinite design options depending on the tuning parameters. Therefore, it is essential to have sampling strategies that can cover the design space thoroughly and efficiently.

3. The 3D-SPI problem is a highly nonlinear optimization problem that must simultaneously address packing, routing, and physics performance evaluation. Therefore, incorporating continuous spatial or parameter tuning may increase the occurrence of local optima in the design landscape.

4. One key challenge in using gradient-based solution methods, such as the geometric projection method [83, 87], is that changes in interconnect spatial topology may impact the lumped-parameter system models (such as fluid loops) in ways that either prevent simulation of certain designs, or, at a minimum, introduce non-smoothness.

5. Creating design representations that can support topology, geometry, and physics aspects of 3D SPI2 problem in a unified way is one of the most challenging aspects. Current methods address at most a pair of them while solving multi-physics optimization problems.

6. SPI2 design automation tools should also consider the human perspective in all steps of the problem-solving process. In particular, industry practitioners who have vast experience in handling these complex systems possess valuable design knowledge that can be applied while developing SPI2 design automation frameworks. Incorporating human expertise into SPI2 automation methods, however, may introduce human biases or errors.

2.8 Summary

This chapter presented an extensive literature search of the major SPI2 related areas: component packing, interconnect routing, physics-based topology optimization, and SPI2 mathematical design representations. Existing

critical gaps that prevent the creation and successful application of design automation methods to industry-relevant holistic SPI2 problems are outlined, and associated challenges are addressed. Some of the key takeaways from this chapter are discussed here. The component packing problem has solved by experts for decades in a way that involves packing both regular and irregular objects within restricted volumes. There are several approaches to perform packing, such as sequential sorting, shrinking an initial unpacked arrangement, and heuristic component assembly, that could be used for addressing the holistic SPI2 problem. Promising solution techniques in the interconnect routing area are shortest path algorithms (SPA) with optimal trajectory control utilized in aerial drone navigation and robotic arm manipulations. SPAs are efficient, create optimal pipe routes, and can be applied either sequentially or simultaneously. Geometric projection-based topology optimization methods offer great potential as they support both complex SPI2 system geometry and physics-based modeling capability together. The mathematical representations identified in this chapter have well established theories in the field of low-dimensional topology which can be leveraged for searching both the 2D and 3D SPI2 topological design spaces effectively. This dissertation focuses on utilizing these above techniques among many alternatives for several reasons that will be explained in the later chapters. However, there could be other methods and approaches that might be powerful and highly relevant when combined together to solve the holistic SPI2 design problem and are considered aspects of future work.

CHAPTER 3

THE SPI2 DESIGN FRAMEWORK: STAGE 1

3.1 Introduction

This chapter[1] presents in detail the first stage of a two-stage design framework for performing simultaneous 2D packing and routing optimization previously developed by the author and collaborators. Stage 1 comprises methods for generation of spatially-feasible initial layouts where there is no overlap between components and/or interconnect network. Three newly applied, computationally efficient, and practical methods are demonstrated here to produce automatically interference-free 2D layouts. First, a 2D force-directed layout method (FDLM) is proposed that implicitly ensures non-interference between components and/or the interconnect network by utilizing spring force theory. Second, the A* algorithm, a well-established 2D shortest path algorithm (SPA), is modified significantly to perform efficient routing of complex interconnect systems. Third, a unique 2D spatial topology (UST) enumeration algorithm is presented that produces all unique interconnect routing configurations for a given multi-component system. These three methods are analyzed with respect to average computational efficiencies and average success rates in attaining feasible layouts for a restricted class of topologies, including evaluation of how they scale to larger complex problems. Practical guidelines are provided to make a choice among the three methods for application to different layout scenarios. Limitations of these methods will be addressed in future work. This chapter serves as an important step in the two-stage design framework for efficient navigation through discrete spatial topology options that could be fed into continuous design optimization methods as initial layouts in stage 2, in addition to scaling to more complex problems.

[1] Some elements of this chapter are published in Ref. [86, 106]

3.2 2D vs. 3D SPI2 Design Problems: Similarities and Differences

Before introducing the proposed two-stage design framework, it is important to understand the similarities and differences between the 2D and 3D SPI2 design problems. Both the problem domains can support similar:

1. complex component geometries (convex, concave, irregularly-shaped, etc.), orientations, restricted regions (obstacles) etc.

2. interconnect features (curvy, straight, L/T-joints, diameter, bend radii, etc.),

3. physics objectives/constraints (temperature, hydraulic pressure, electromagnetic effects, material stress/fatigue, etc.), and

4. spatio-geometric features: packing fraction, arbitrary bounding area/volumes, system interconnectivity constraints, etc.

However, there are several differences between them, as follows:

1. 3D problems have crossings between interconnects. This exponentially expands the spatial topology space and design options possible,

2. Accessibility, assembly/disassembly, maintainability, repair, etc. are non-trivial problems in 3D, but are much less challenging in 2D due to its planar layout features.

3. In 3D, the port locations, and how the interconnects interact with components at their ports (at angles, with offsets, etc.) is a unique problem aspect,

4. In 2D, objects can have voids or holes, but these features are not of much physical significance. In 3D, however, interconnects can either go through hollow objects or bypass the holes, thus expanding a number of spatial design options.

5. The 3D problem is harder to conceptualize, whereas the 2D layout problem is easier to graphically visualize which helps make optimization search and decision making a much easier process.

The above mentioned differences and similarities between the 2D and 3D problem cases motivate the need to solve 2D SPI2 problem initially before attempting to solve the 3D problem. Three major reasons are mentioned below:

1. The 2D SPI2 design problem, although challenging, is relatively easier to conceptualize, represent, and solve when compared to the 3D SPI2 design problem. Moreover, there are some direct 2D SPI2 applications where this methods can be implemented.

2. The proposed sequential design framework can be thoroughly tested on the 2D SPI2 problem. In particular, solving the combined packing, routing, and physics-based problems in 2D can help develop valuable insights regarding some of the challenges that may arise when attempting to solve the 3D SPI2 problem.

3. The 2D SPI problem is fundamentally different from the 3D SPI2 problem, however, there are many similarities that can be exploited for developing the two-stage design optimization framework. This provides a platform to transition from the 2D SPI2 design methods to 3D where there are similar design features.

Thus, the 2D SPI2 problem plays a vital role in creating a strong foundation for building 3D design methods can be built. Hence, investigation into the 2D SPI2 design problems is one of the major contributions of this work.

3.3 The Two-Stage SPI2 Design Framework

Although discrete formulations have dominated earlier research efforts to solve the optimal spatial packing problem, these approaches quickly become intractable, as both the packing problem and the routing problem are NP-hard. Rather than make incremental progress on established methods, this dissertation proposes a novel strategy for SPI2 design using a continuous representation applied to distinct spatial topologies. This enables the use of gradient-based methods to efficiently search the packing and routing design space. This methodology is centered on the use of simple geometric bars to approximately model both the component geometry and routing paths

(see Fig. 3.1). Bars have favorable geometric properties for both the packing and routing problems, enabling simultaneous solution. This simple geometric representation with physics-based boundary conditions (heat source, heat sink and forced convection) will be used as part of the two-stage framework. Specifically, the combined physics-based packing and routing problem can be subdivided into two important sequential stages as shown in Fig. 3.2: 1) Generate spatially feasible geometric layouts without any interference between the components and interconnect routing (for example, pipes or ducts), and 2) to utilize these feasible layouts as initial designs for performing multi-physics optimization to meet system-level requirements such as packaging volume, operating temperature, fluid pressure, and cost. It is assumed here that the continuous optimization strategy cannot make topological changes, two classes of which are described below.

Spatial topology (ST) refers to the mathematical study of the properties that are preserved through continuous deformations, such as stretching and twisting, but not tearing or gluing. For instance, a circle is topologically equivalent to an ellipse, but a circle and a hollow ring have different topologies. For example, if an interconnect links ports $P1$ and $P2$, many options exist for how this interconnect passes around various other interconnects and components in the system. Two spatial topologies are *equivalent* when there is a continuous deformation of component locations and interconnect trajectories that takes one topology to the other. *System architecture (SA)* specifies what components comprise a system, and how ports on components are connected to specific ports on other components which represent specific technology options to perform each function, and the flows of material, energy, and/or information from one component to another. System architecture enumeration problems were previously studied for engineering design examples such as hybrid electric power trains [107–109], automotive vehicle suspension systems [105], and optimal cooling system layouts [103]. However, for each system architecture, many spatial topologies (STs) may exist, each with its own optimal geometry. Hence, the geometric optimization problem can be simplified by decomposing the problem into identification of unique STs, and then geometrically optimizing each one, subject to all relevant physical interactions. Figure 3.3 shows three layout options for a system with three components that have two ports each. All three have the same system archi-

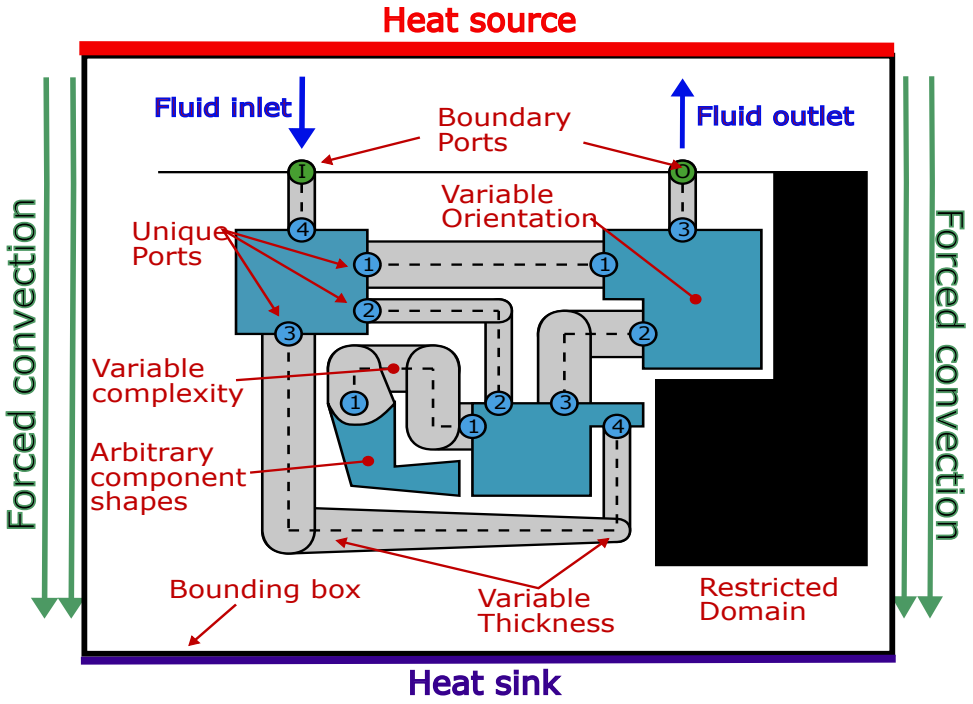


Figure 3.1: 2D geometric layout design representation with core problem features.

tecture (the component connectivity matrix is the same), which is one aspect of system topology. Layouts A and B differ only in the interconnect path; they have the same spatial topology. Layout C has a distinct spatial topology; i.e., the red-colored interconnect cannot be continuously morphed on a plane to achieve the spatial topology exhibited in layouts A and B without either leaving the plane or requiring cuts or reconnections.

Continuous optimization algorithms in stage 2 require initial designs without spatial interference. Specifically, feasible 2D spatial layouts here are planar layouts where both the individual components and interconnects do not overlap and/or intersect with each other. Such layouts can be obtained by applying non-overlap constraints, also known as interference detection tests. Interference detection between fixed polygons is a fundamental geometric consideration that arises in many engineering applications, such as robotics [110, 111], computer animation graphics [112, 113], virtual reality games [114], UAV navigation [115], and multi-agent coordination theory [116].

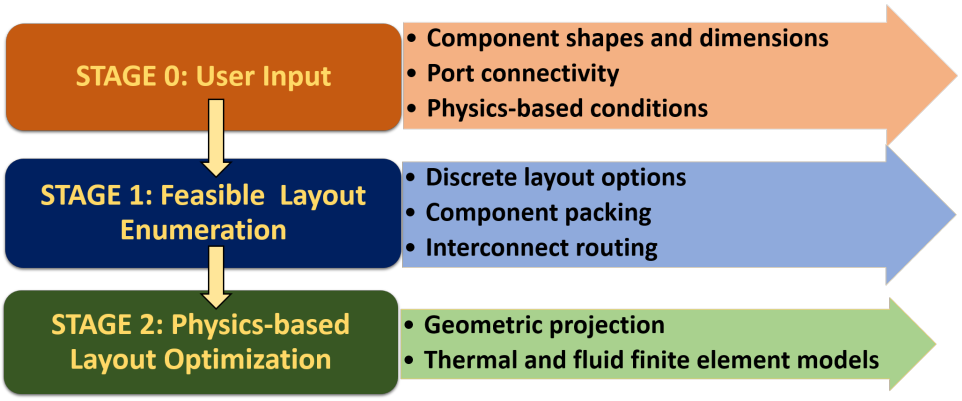


Figure 3.2: Two-stage design framework for 2D spatial packaging.

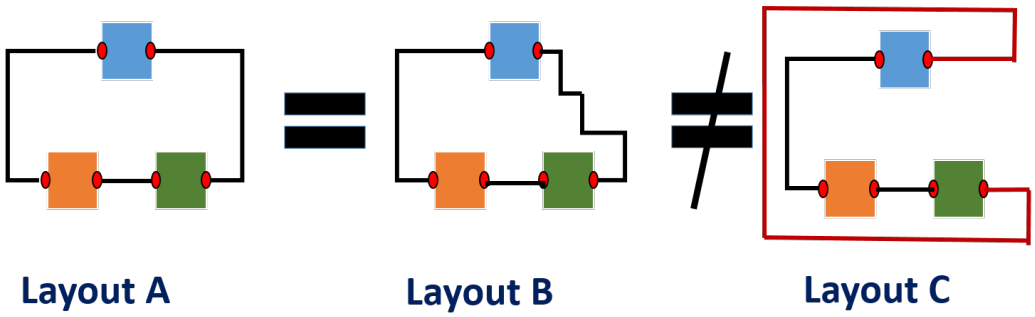


Figure 3.3: Illustration of equivalent and distinct spatial topologies.

Spatial packing problems have been solved previously using interference detection constraints across a range of applications, including 2D bin packing [117], aircraft container loading [118], non-overlapping rectangles [119], vehicle routing and assembly [120], transportation networks [121], and interconnected thermal systems [122]. While sophisticated computational geometry algorithms exist for interference detection, these approaches are usually slow for multiple objects. A primary disadvantage of using explicit non-overlap constraints is scaling, especially with respect to handling many simultaneous constraints. In this chapter, three strategies are investigated that avoid the need for explicit interference constraints: 1) a force-directed layout method (FDLM) based on spring forces to implicitly ensure non-interference, 2) an extension of shortest-path planning algorithms (SPAs), and 3) explicit enumeration of unique 2D spatial topologies.

3.4 Chapter Aims and Contributions

The primary aim of this chapter is to demonstrate design methods that automatically produce crossing-free 2D layouts for packing and routing problems. These methods can lay a foundation for further developments that will support real-world 2D and 3D packaging and routing problems where it is possible to navigate the fundamentally discrete options for interconnect topology, as well as scaling to more complex problems. The core contributions of this chapter are as follows:

1. Introduction of a novel force-directed layout method (FDLM) utilizing spring-force theory inspired by planar graph drawing algorithms [123, 124]. This method implicitly satisfies interference constraints.
2. Extension of the well-established A* (A-star) shortest path planning algorithm (SPA) to perform sequential interconnect routing between components while ensuring non-interference.
3. A systematic enumeration strategy for generating all unique 2D spatial topologies for an interconnected multi-component system, for given system architecture. This method serves as a baseline for design space coverage benchmarking but does suffer from a combinatorial explosion.
4. These three methods (FDLM, SPA and UST) are separately analyzed with respect to computational efficiency, layout design space coverage,

and performance in arbitrarily-shaped bounding areas using relevant case studies.

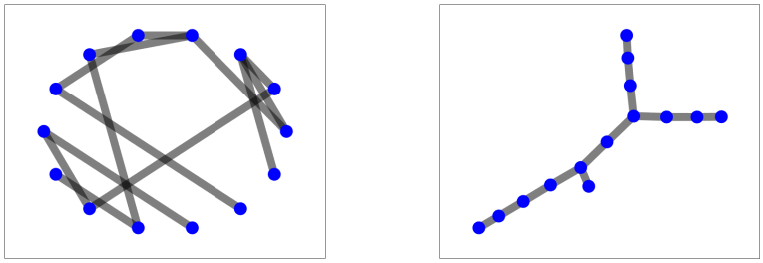
5. Practical design guidelines are provided for the application of the three proposed layout generation algorithms to attain various feasible 2D system layouts. The FDLM and SPA methods can be applicable from small to medium scale systems (approximately upto 50 various sized and shaped components). The UST at present can handle upto systems with 20 components only because it involves an exhaustive enumeration process.

The methods were developed keeping the ubiquitous nature of packing and routing problems in mind. If the current method limitations are overcome, they have the potential to be extended to produce 2D planar layouts of various sizes and scales for applications ranging from system-level (aircraft, automobile, marine, building architecture, water distribution networks), to cabinet-level (automotive engines, electric machines, cooling systems, motors, turbines, etc.), and device-level (VLSI circuit board) designs.

The remainder of this chapter is organized as follows. Section 3.5 describes the 2D layout design representation shown in Fig. 3.1. Section 3.6 demonstrates the Force-Directed Layout Method (FDLM). The shortest path-based layout algorithms (SPAs) utilized to perform complex interconnect routing are presented in Sec. 3.7. Section 3.8 presents a new unique layout topology generation algorithm to produce unique layouts. The comparative case studies for the 3 methods are presented in Sec. 3.9 with the discussion of results in Sec. 3.10. Finally, the chapter summary is provided in Sec. 3.11.

3.5 2D SPI2 Problem Design Representation

Figure 3.1 represents a 2D layout problem with different geometric design features: components with arbitrary shapes, interconnects (pipes/ducts, etc.) represented as bars, and restricted regions. The aim of the layout generation methods is to generate layouts for a given system architecture that are feasible with respect to spatial interference. It is desirable to minimize the inputs required from the user. The following are assumed to be specified by the user: 1) The bounding box shape and area (a region which all compo-



(a) Initial graph (b) Final crossing-free graph

Figure 3.4: Initial and final graphs attained after force-directed layout implementation.

nents and interconnects must fit within), 2) all component dimensions, port locations, and initial positions, and 3) the component connectivity matrix and interconnect geometric requirements (e.g., thickness).

3.6 2D Force-Directed Layout Method (FDLM)

Force-directed algorithms are dynamic graph-based algorithms for producing layouts of simple undirected graphs used in several applications, such as information networks [125], social networks [126], gene networks [127], and bio-molecular interaction networks [128]. A class of force-directed algorithms is based on mechanical spring force models, also known as spring embedders. Layouts drawn with these algorithms tend to exhibit symmetries, are aesthetically pleasing, and can produce crossing-free (nodes and edges do not intersect) planar graphs. One of the first force-directed graph drawing methods [129] is based on barycentric representations that ensure crossing-free layouts. The spring layout method was introduced in 1989 by Kamada et *al.* [123], and then improved by Fruchtermann et *al.* [124] in 1991. Figure 3.4 shows a final graph obtained from an initial graph with intersecting edges using a spring-based force-direction algorithm. In this method, adjacent graph nodes have attractive forces between them and repulsive forces with remaining nodes. Here the entire graph is considered to be a system of particles (nodes) with spring forces acting between them. The initial random graph layout has high overall energy (stored in virtual springs), and is unstable because of the forces between the nodes. The end goal is to position nodes

to obtain locally-minimal potential energy. In other words, force-directed techniques seek an equilibrium state where the sum of the forces acting on each particle (node) is zero. Thus, any multi-component physical system can be represented as a force-based model with its various system elements as shown in Fig. 3.5 such as components as nodes. In this method, if v is the node, the force on a node v, attractive factor f_a, and repulsive factor f_r are calculated using the following equations, respectively:

$$F(v) = \sum_{(u,v)\in E} f_a(d_{uv})\frac{(u-v)}{d_{uv}} - \sum_{(u,v)\in V} f_r(d_{uv})\frac{(u-v)}{d_{uv}}, \tag{3.1}$$

$$f_a(d_{uv}) = k_{uv}^{(1)}(d_{uv} - l_{uv}), \tag{3.2}$$

$$f_r(d_{uv}) = \frac{k_{uv}^{(2)}}{d_{uv}^2}, \tag{3.3}$$

where d_{uv} is the distance between any two graph nodes (say, u and v), and l_{uv}, $k_{uv}^{(1)}$, and $k_{uv}^{(2)}$ are the natural spring length, spring stiffness, and repulsive force strength, respectively. The total energy of the system is then given by:

$$E = \sum_{(i=1)}^{n-1} \sum_{(j=i+1)}^{n} \frac{1}{2} k_{v_i v_j}(d_{v_i v_j} - l_{v_i v_j})^2, \tag{3.4}$$

where $l_{v_i v_j}$ corresponds to the desired distance between graph nodes (vertices) v_i and v_j, and $k_{v_i v_j}$ is the spring stiffness between the two vertices. Increasing or decreasing the spring stiffness affects how rapidly or slowly the particles move. The position of each node is expressed by x and y coordinate axes. Taking partial derivatives of energy (E) defined in Eqn. (4) with respect to x and y, this algorithm computes a local minimum of the energy (E) using the regular Newton-Raphson method from an initial node position. At the stationary point, the dynamic graph network attains an equilibrium state where all the forces involved in Eqn. (1) are balanced at each node.

3.6.1 Force-Directed Layout Representation

To support the objective of finding feasible layouts for packing and routing problems, modifications are made to the basic FD method described above. Figure 3.6 illustrates a detailed representation used for 2D force-directed layout. The components are represented as polygonal shapes and

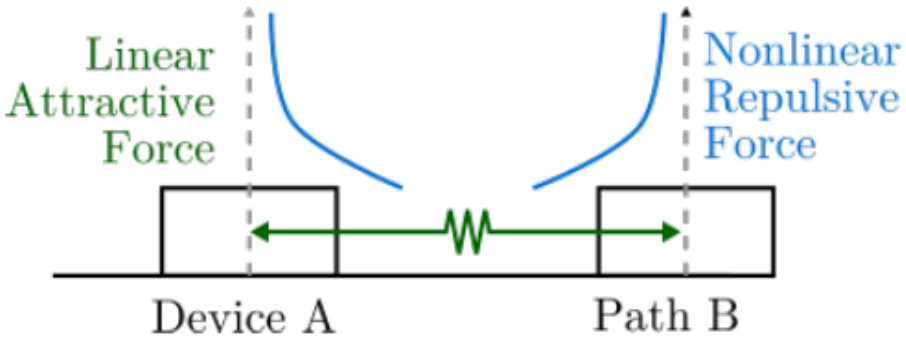


Figure 3.5: Force representations on system components.

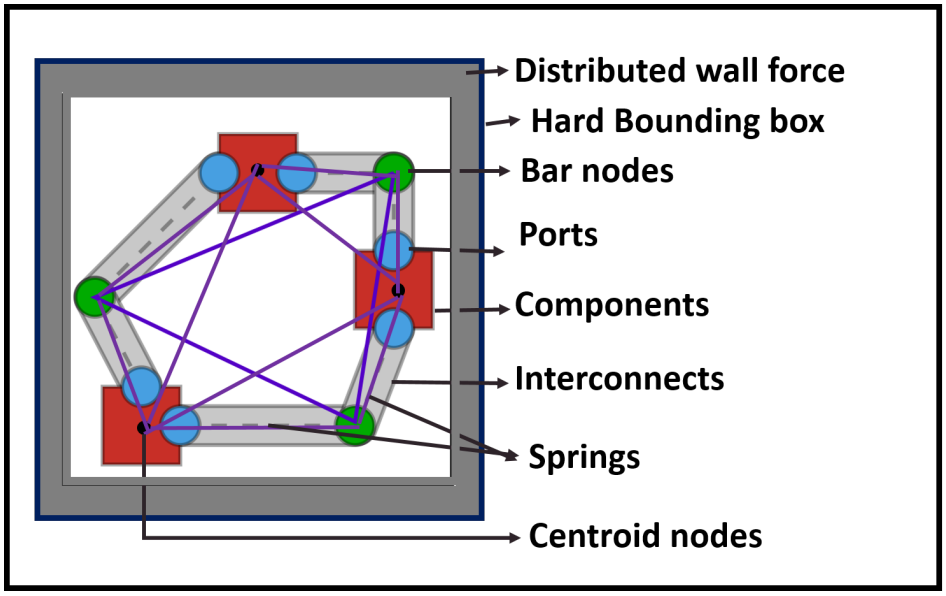


Figure 3.6: 2D Force-Directed Layout Representation.

the interconnects are represented as bars (with one or more links, providing piecewise-linear representation of curves). Both components and links are free to move within a hard bounding box (HBB). The graph nodes are component centroids, bar centers, and junction nodes for interconnects having multiple bar segments. Spring forces indicated as purple lines are present between all these nodes. To prevent overlapping between the components, the natural (equilibrium state) spring length (l) for the algorithm is initially set to a value greater than the maximum of the sum of radii of bounding circles for all component pairs i.e., $l > \max(r_i + r_j)$. The bounding circle radius is the distance from the component centroid to its farthest vertex. In addition, distributed constant boundary forces (gray shaded regions) are acting inward pushing the components and interconnect network to ensure they do not escape the HBB.

3.6.2 FDLM for multi-component system without interconnects

Components without the interconnects are considered first as a simplified illustration of the FDLM. Components of various shapes are placed within a HBB at random initial locations. Figure 3.7 shows both the initial and final layouts for the 10-component and 15-component systems, respectively. At the initial random states, many components typically overlap. The FDLM then attains a final component layout without any overlaps. Springs between the centers of the components are indicated in red. The computational time required to attain the equilibrium state are 112.3 seconds for 10-component system, and 186.2 seconds for 15-component system respectively. FDLM requires more time to attain equilibrium as the number of components is increased. All reported computational costs were obtained using a workstation with an Intel Xeon E5-2660 CPU @ 2.00 GHz, 64 GB DDR4-2400 RAM, Windows 10 64-bit, and Matlab 2019b.

To increase FDLM efficiency, a new spring connectivity network is utilized for a multi-component system. Note that the spring connectivity network is distinct from component connectivity. A fully-connected graph (FCG) network is used in FDLM as seen in Fig. 3.7 For an FCG network, the spring

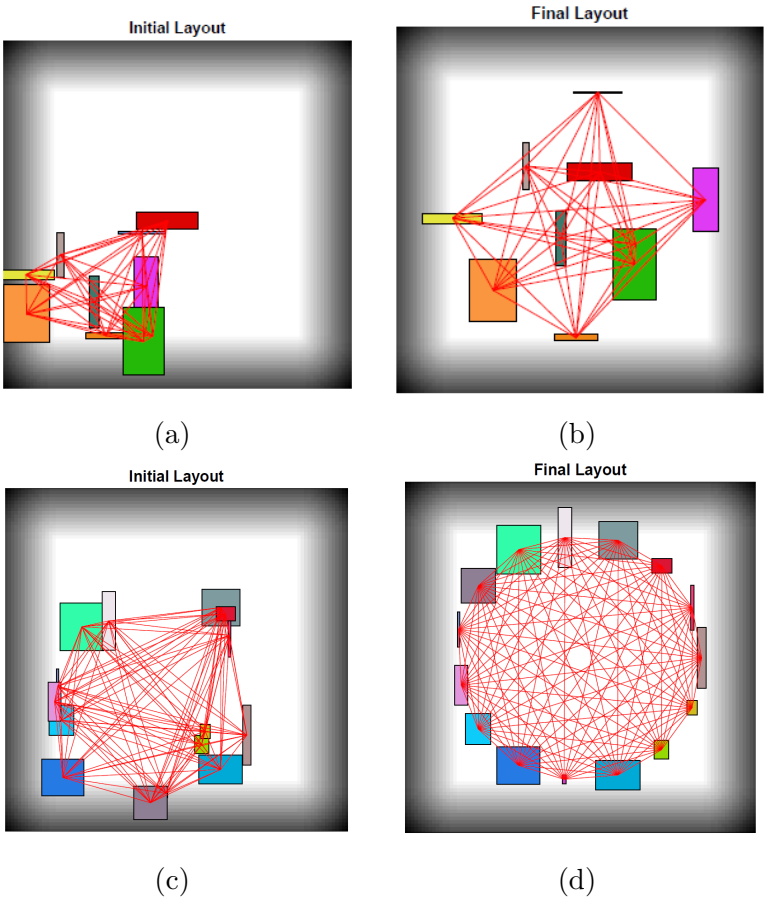
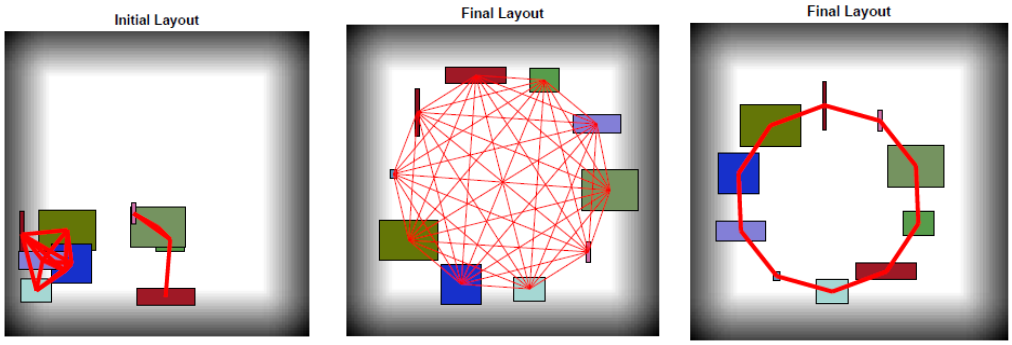


Figure 3.7: Initial and final layouts for 10-component (**(a)** and **(b)**), and 15-component (**(c)** and **(d)**) systems respectively with FDLM described under Sec.3.6.2.

forces are calculated for all node pairs at every iteration. Alternatively, the nearest neighbor graph (NNG) is used for determining spring connectivity. In NNGs, spring forces act between two nodes when they are within a certain radial distance from each other. In other words, each individual node has a circle of influence, and spring forces act only when other nodes are within its region. As the force-directed simulation progresses in NNGs, for every individual node, its neighboring nodes slowly move out of its circle of influence. This significantly reduces the number of node-pair spring force computations per iteration but can introduce non-smoothness. Figure 3.8 shows initial and final layouts for the two types of graph connectivity networks for a 10-component system. For the NNG layout example, in the final layout displayed, only the closest neighbor force interactions are illustrated. The total run time for NNG to attain equilibrium was 45.3 s, and for FCG



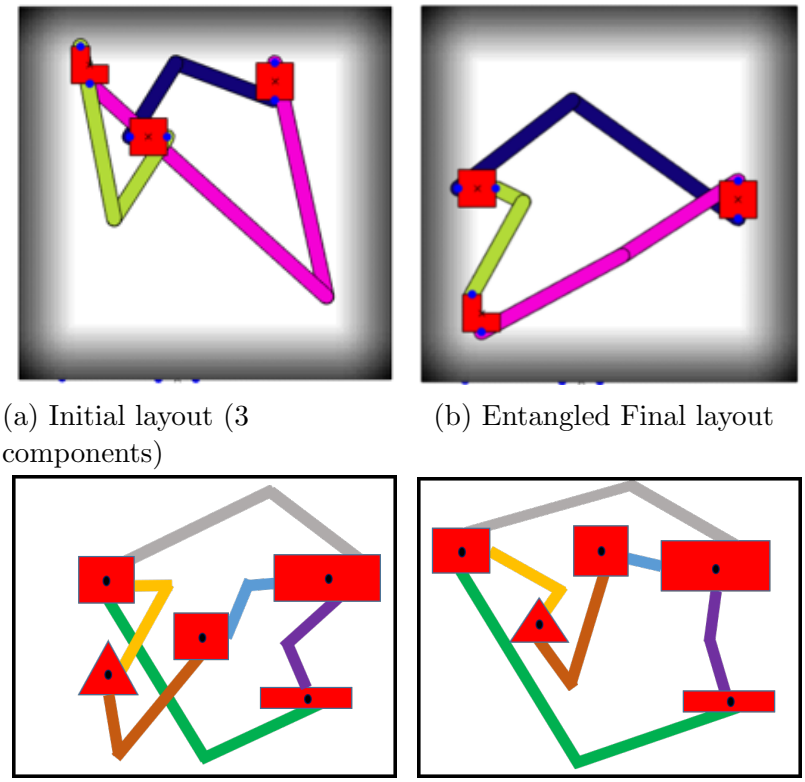
(a) Component Initial layout (b) FCG Final layout (c) NNG Final layout

Figure 3.8: Initial and final layouts attained with FDLM for a 10-component system using fully connected and nearest neighbours graph networks.

was 106.23 s. In the NNG layout, the components are more closely packed because they come to an equilibrium state when they do not experience forces from the closest neighbors, unlike FCG where all forces need to be balanced. Thus, NNGs are very efficient for solving large-scale force-directed layout problems.

3.6.3 FDLM for multi-component system with interconnects

Here feasible layouts are generated by simultaneously removing interference between components and untangling interconnect intersections. Once again a multi-component interconnect network is randomly placed in an HBB. Figure 3.9 shows initial and final layouts for a 3-component system (Figs. 3.9a, 3.9b) with interconnects that do not get fully untangled. Boundary forces are removed for a 5-component (Figs. 3.9c, 3.9d), multi-port system with interconnects in the next study; this aids untangling, resulting in complete untangling. The NNG network for both the 3- and 5-component systems required 132.3 s and 242.4 s, respectively, while FCG required 243.7 s and 411.9 s, respectively. While often effective, FDLM does not always support feasible layout generation for systems with complex routing networks. The interconnect bars tend to intersect with each other as they have nodes only at their centers and endpoints. Developments can be made where instead of having forces between discrete points (nodes), there could be a continuous



(a) Initial layout (3 components)

(b) Entangled Final layout

(c) Initial layout (5 components)

(d) Untangled Final layout

Figure 3.9: Initial and final layouts attained with FDLM in Sec.3.6.3 for a 3-component and 5-component interconnected systems.

force field throughout the entire boundary of these interconnect bars just as components here have a circle of influence around them. This strategy is not included in this initial study and is an important topic for future work. As an alternative remedy to the inability of FDLM to always untangle interconnect networks, the properties of efficient Shortest Path Algorithms (SPAs) are exploited in Sec. 3.7 to perform complex routing for a system where component overlaps are first removed using FDLM.

3.7 2D Shortest-path based Layout Algorithms (SPA)

Shortest path algorithms are widely used in applications such as robot path planning [130–133], obstacle avoidance [134], transportation networks [121], the traveling salesman problem [135], and web-based geographical maps [136]. Robot path-planning algorithms use grid-based techniques where obstacles

occupy a group of cells within the grid known as *closed cells*, and the aim is to reach a target point from a source point avoiding the obstacles. To achieve the task the robot moves towards the target through the remaining *empty cells* on the grid.

Several variants of SPAs exist, such as A* and Dynamic Programming (DP) methods. Here it is observed that interconnect routing for a fixed system architecture can be performed via sequential application of SPA methods, notably producing different spatial spatial topologies depending on port connection sequence. Consider a multi-component interconnected system where its components are randomly placed without any overlaps on a 2D grid. Let two components, $C1$ and $C2$, have ports P_{C1} and P_{C2}, respectively, that need to be connected by a route (path). If P_{C1} and P_{C2} are taken as the source and target points, respectively, a SPA can be applied to find a feasible path between the two ports (if one exists). All components are treated as obstacles. Once the SPA finds a feasible path on the grid, this process can be repeated for other interconnections. At each iteration, previously-generated routes are also treated as obstacles. Thus, the number of available empty cells decreases for paths connected later in a given sequence. If there are n interconnects in a system, then $\binom{n}{2}$ routing sequences are possible. Therefore, by using this method, a maximum of $\binom{n}{2}$ feasible final topologies can be obtained. Several routing sequences may fail to obtain feasible layouts in between if there is no path of empty cells available for connecting 2 ports due to an increased number of obstacles. Distinct sequences may produce identical geometric topologies.

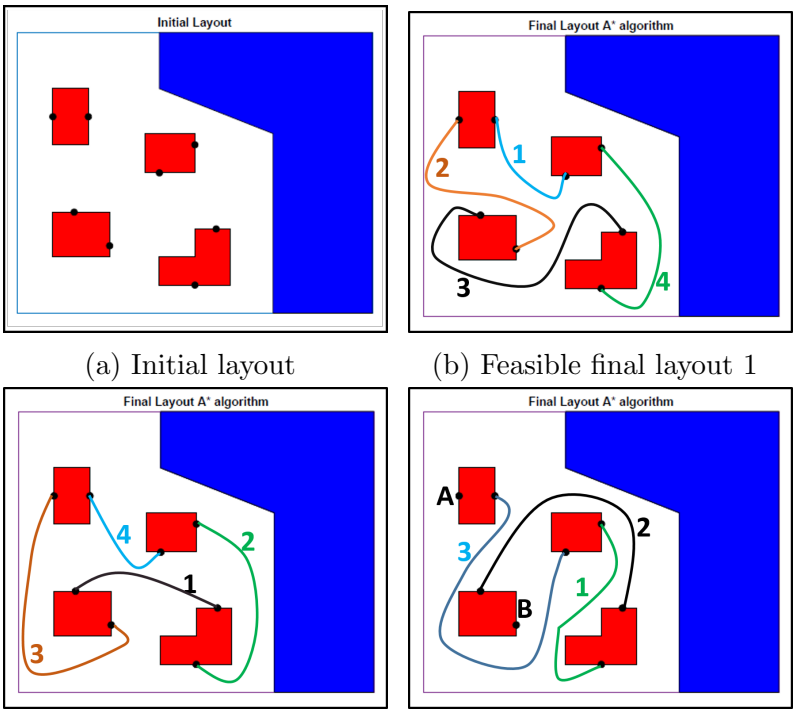
3.7.1 The A-Star (A*) algorithm

A* is the most popular choice for path-finding [102, 137] because it is very flexible and can be used in a wide range of applications. The most basic SPAs are the Dijkstra's and Dynamic Programming (DP) algorithms [131], upon which the A* star algorithm was developed. Dijkstra's method works by visiting the empty cells in a grid, beginning with the source cell. It repeatedly examines the closest unvisited neighbor cell, adding it to the cells to be visited. In this way, the algorithm reaches the target from the source via

the shortest path. The DP algorithm works in a way similar to Dijkstra's, except that it uses an estimate (called a greedy heuristic) of how far from the target any empty cell is. Instead of selecting the empty cell closest to the source point, it selects the empty cell closest to the target. Thus DP is not guaranteed to find the shortest path. But the advantage is that it runs much more quickly than Dijkstra's Algorithm because it uses the heuristic function to guide its way towards the target.

The A* algorithm combines the pieces of information that Dijkstra's Algorithm uses (favoring empty cells that are close to the source) and information that DP uses (greedy heuristic favoring empty cells that are close to the target). For every iteration, the A* algorithm examines the empty cell n that has the lowest path cost $f(n) = g(n) + h(n)$, where $g(n)$ represents the exact cost of the path from the source to any empty cell n, and $h(n)$ represents the heuristic estimated cost from empty cell n to the target. A* balances the two as it moves from the source to target cells. Figure 3.10 shows the results of the A* star method for a 4-component system on 50 $\times$ 50 grid. Each component in red has two ports (black circles), each as shown in the initial layout. The total number of possible routing sequences is $\binom{4}{2} = 6$, and hence a maximum of 6 feasible layouts are possible. The restricted domain in blue color is also treated as an obstacle where no interconnect routing can happen. Two final feasible layouts, shown in Figs. 3.10b and 3.10c, are obtained with their routing sequence numbers labeled from 1 through 4. An infeasible layout is shown in Fig. 3.10d, where ports A and B do not have a feasible connecting path as the grid does not have empty cells available. The computation time required for the A* method to generate feasible layouts 1 and 2 are 84.6 s and 93.5 s, respectively. Although the SPAs are highly efficient, one major limitation is they are grid-based methods. For instance, if there are 100s of interconnects in a highly-dense domain packed with many components (obstacles), these algorithms might fail if enough empty cells are not available as the interconnect sequence progresses. To overcome this limitation, much finer grids can be created, hence improving the likelihood of attaining feasible layouts, with the expense of greater computational time. Figure 3.12 shows an example of the A* algorithm being used to generate a feasible layout (see Fig. 3.12b) on a denser mesh. The components shown in red are at fixed locations; restricted regions are indicated as black boxes.

The interconnect routing sequence numbers are shown in Figs. 3.12(b, c). Figure 3.12c illustrates an infeasible final layout because the ports $P1$ and $P2$ shown here do not have a feasible connection path due to the coarse grid (60×40 elements) that limits the availability of empty cells.



(a) Initial layout (b) Feasible final layout 1

(c) Feasible final layout 2 (d) Infeasible final layout

Figure 3.10: Initial and final layouts attained with the shortest path A* method for a 4-component 2-port system. Sequence number of paths for each layout are shown.

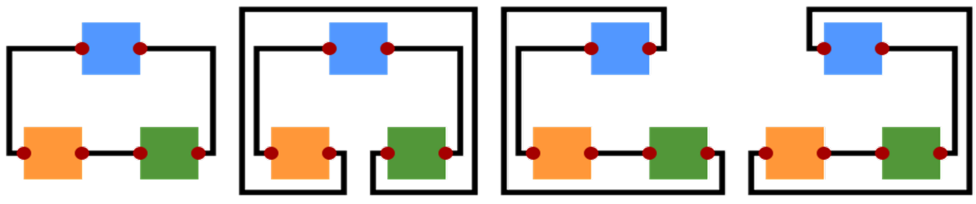
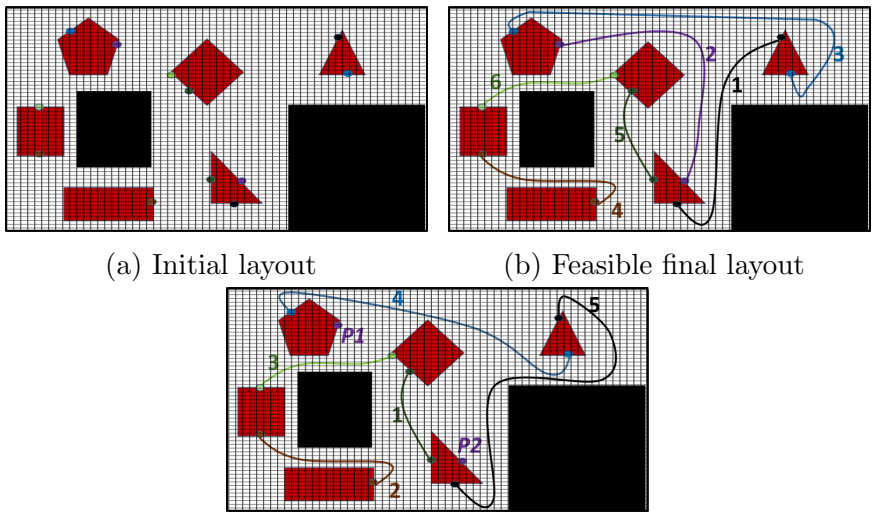


Figure 3.11: 4 Unique Spatial Topologies for a 3-component system. ST-A is a directly connected configuration. ST-B, -C, and -D are indirectly connected configurations.



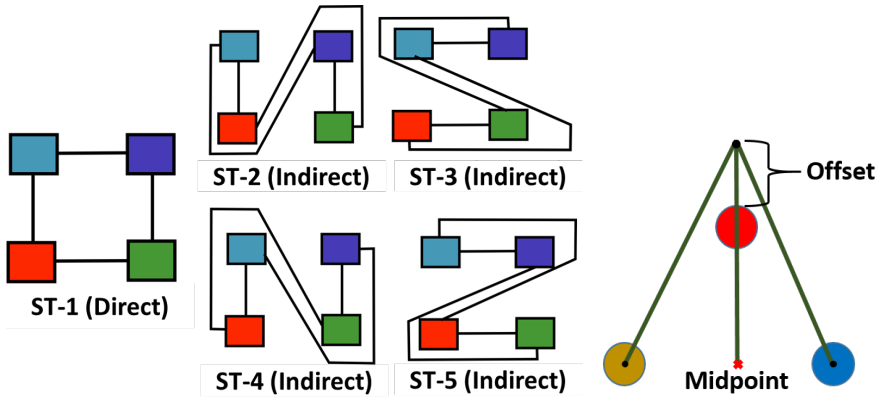
(a) Initial layout (b) Feasible final layout

(c) Infeasible final layout

Figure 3.12: Illustration of the shortest path A* algorithm on a denser mesh.

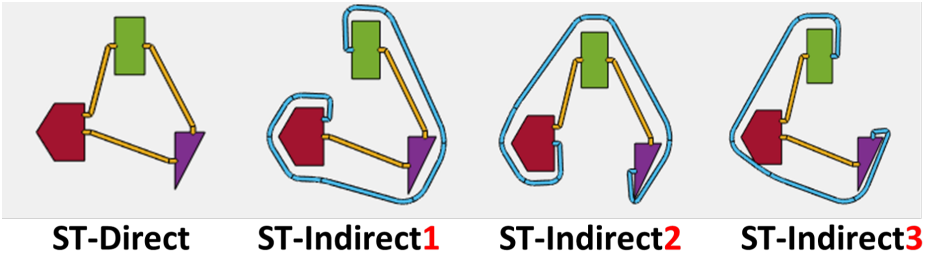
3.8 2D Unique Spatial Topology (UST) Enumeration Algorithm

Here a new layout enumeration algorithm is presented that can systematically generate feasible unique spatial topologies (STs) for given system architecture (SA). This exhaustive enumeration serves as a benchmark for assessing spatial topology design space coverage of other strategies. Since a continuous mapping exists between any two layouts that share a ST, the complete set of STs is an important set of starting design points for continuous layout optimization. Consider a 3-component system where each component has two ports, as shown in Fig. 3.11. Each of the 3 components is connected directly to the other two components in ST-A. This configuration is termed the *directly connected ST*. Figure 3.11 also shows three other unique STs (namely B, C, and D) for the same system architecture represented by ST-A. Note that in these STs one interconnect circulates over a component to reach the other component. Such a configuration is termed an *indirectly connected ST*. Thus, a 3-component and 2-port interconnected system has a total of 4 unique STs (one direct and 3 indirect). Similarly, for a 4-component system with each component having 2 ports, a single unique direct ST and four unique indirect STs are generated, as shown in Fig. 3.13a. By mathematical induction, for an n-component system with each having two ports each, 2D

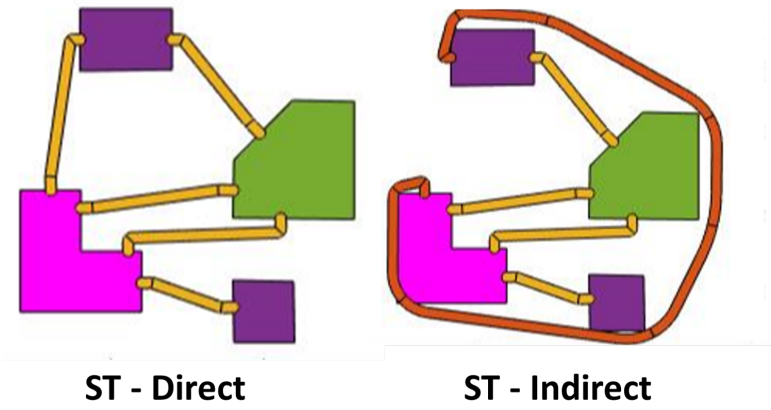


(a) (b)

Figure 3.13: UST method illustration: (a) Five unique spatial topologies of a 4-component 2-port system; and (b) offset representation.



(a) Unique STs for a 3-component system with different shapes.



(b) Different component shapes with varying number of ports.

Figure 3.14: UST method demonstration to enumerate feasible layouts with distinct spatial topologies.

layouts involve $n+1$ unique STs. This method can also be applied to a multi-component system with each component having different shapes and a varying number of ports. For example, in Fig. 3.14, different cases are demonstrated.

Algorithm 1: Unique Spatial Topology (UST) Generation Algorithm

Input: Component dimensions and locations, port connectivity, and restricted regions (if any).
Initialize - 1. Extract the vertex coordinates of all components. 2. Specify offset distance between component center and circulation node.;
for *i = 1: total number of interconnections (say n)* **do**
  find all possible paths for ith interconnect, say p;
  $k = 1 : p$ (total number of possible paths for ith interconnect);
  while $k \leq p$ **do**
    if *direct path* **then**
      form a direct connection;
      store coordinates of feasible connection;
    else
      check feasible indirect paths ;
      if *no self-interference or interference with components or existing paths* **then**
        form an indirect connection;
        store coordinates of feasible connecLtion;
      else
        discard path;
      end
    end
    $k = k + 1$;
  end
  $i = i + 1$;
end
Result: All feasible interconnection combinations' paths
Output: Plot all unique spatial topologies.

3.9 Stage 1: Case Studies

Here the FDLM, SPA, and UST methods are compared with each other using three case studies. First, their average computational efficiency in ob-

taining a feasible final layout and how it scales for an increasing number of components, say n. Second, their success rate at generating all feasible STs (design space coverage) as system dimension increases. For the case studies, the class of topologies are restricted to only n-identical component systems with each component having the same geometric shape and size with two ports each (say, nC-2P). The third case study illustrates how these methods perform in an arbitrarily-shaped bounding area both with and without restricted regions. The FDLM begins with both components and routing overlapped. The SPA and UST methods have non-overlapping components initially. **FDL-FETS (Free-Expansion-Then-Shrinking) method:** As a strategy for more effective assessment of FDLM, a two-stage approach is used where free expansion in a larger bounding area is allowed to promote untangling an improve the chances of attaining a feasible layout. After free expansion, boundary forces are applied to the expanded layout to shrink it within the prescribed bounding-box area. A final layout is considered a failure if it contains any interference between components or interconnects after it shrinks back to the original bounding box area. Figure 3.15 shows an illustrative example of an initial, a freely-expanded, and a feasible final layout attained using the FDL-FETS method. The FETS approach uses the nearest neighbors graph (NNG) to enforce spring forces between its components.

3.9.1 Case Study 1: Average computational times

In this case study the average computational time taken by each method to attain one feasible layout from a random initial layout of an n-component 2-port system is investigated. The computational time for first 200 simulations that attained a final feasible layout was recorded for each n (number of components) varying from three to ten. As shown in Fig. 3.17(a), the average times were computed in this case study with six layout generation methods: three force-directed layout (FDL) methods, the nearest neighbor graph (FDL-NNG), fully connected graph (FDL-FCG), and the modified FDL-FETS, two shortest path (SPA) methods (A* and DP), and the UST algorithm. From Fig. 3.17(a), it can be observed that the shortest path algorithm (SPA) and the unique geometry topology generation (UST) methods are comparable to each other and are very efficient. The force-directed

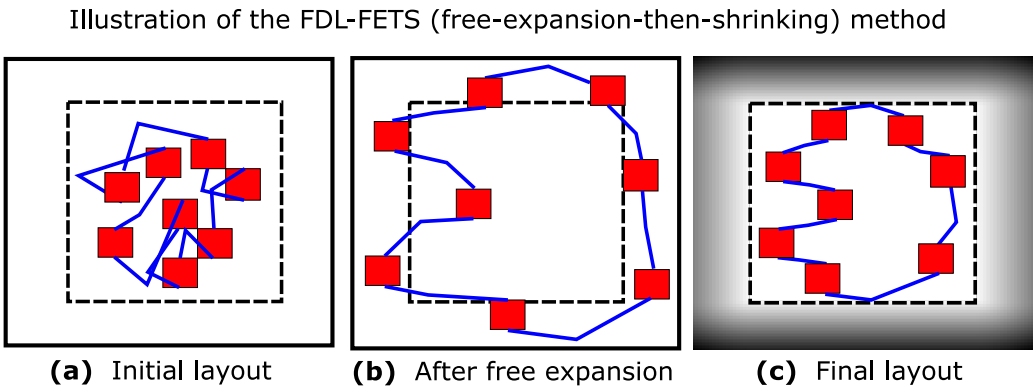
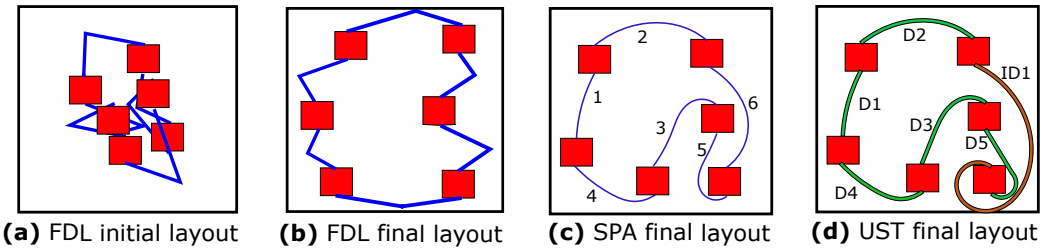


Figure 3.15: FDL-FETS method illustration with a random 8-component 2-port system.

layout methods, however, turn out to be computationally expensive even though they are implicit constraint methods. Two major reasons are that the final solutions require several iterations to attain an equilibrium state. The FDL method also contains components which are overlapped initially, unlike the SPA and UST methods which start from non-interfering component layouts. The FDL-NNG network implementation, however, exhibited significant performance improvements over the standard FDL-FCG network implementation. The FDL-FETS is slightly more efficient than the FDL-NNG method. As the bounding box constraint is not present for FDL-FETS initially, the minimum energy state can be attained easily, but it requires significant computational time to shrink back to the prescribed bounding box area as it needs to balance the internal spring forces with the external wall forces. Figure 3.17(b) compares the computational time distribution via a box plot for the six methods in finding a single feasible layout for a 10-component 2-port system. Figure 3.16 shows one representative final feasible layout obtained using FDL-NNG, SPA and UST methods, respectively. The initial layout for FDLM alone is shown in Fig. 3.16(a). The SPA and UST methods initially have components at fixed locations, and routing is performed to attain final layouts.

3.9.2 Case Study 2: Average success rates

In this case study the average success rates of the above six methods in generating a feasible layout over 500 simulations for nC-2P systems with n varying



(a) FDL initial layout **(b)** FDL final layout **(c)** SPA final layout **(d)** UST final layout

Figure 3.16: A result from Case Study 1 with a 6-component 2-port system showing an **(a)** initial layout; and **(b)** final layout attained using FDL-NNG method; **(c)** shows SPA final layout; and **(d)** an indirect UST final layout.

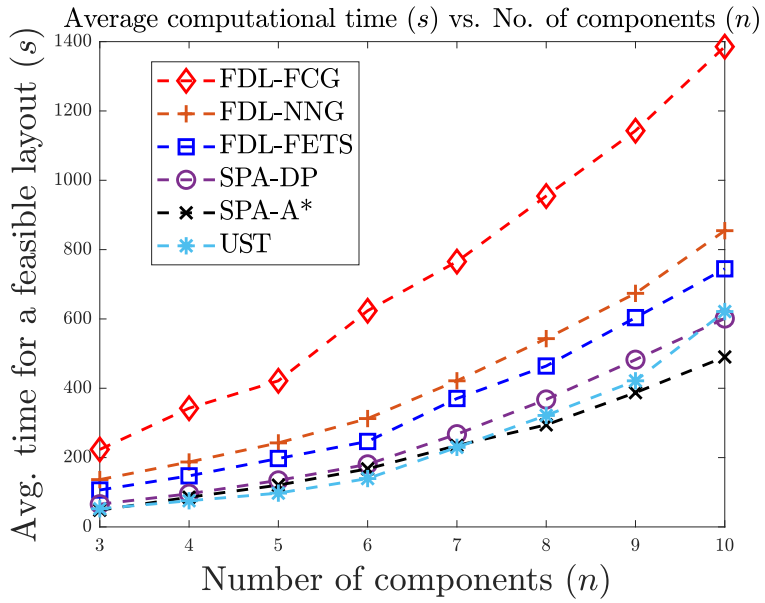
from three to ten are compared to one another. As shown in Fig. 3.21, for these nC-2P systems, the UST has almost a 100% average success rate, followed by SPAs success rate that gradually declines with increasing n. In principle, the UST should have exactly a 100% success rate, but at times the random initial layout may have component spacing small enough such that UST fails. One resolution may be to increase the bounding box size to prevent this failure, generate STs, and then follow-up with FDLM to attempt to contain all components into the original bounding box. FDLMs have the lowest average success rates, but it should be noted that they perform component and interconnect network untangling simultaneously within a fixed bounding box area. The modified FDL-FETS approach (as shown in Fig. 3.21) significantly improved the success rate of the FDL method with increasing scale as it provides more space for initial untangling. Section 3.10 below lists these points in more detail.

3.9.3 Case Study 3: Arbitrarily-shaped bounding boxes

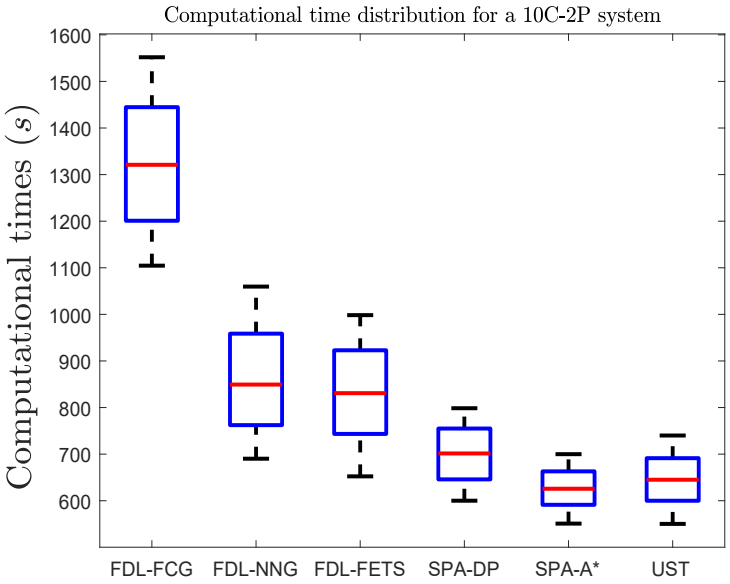
Many real-world systems may not have bounding boxes with simple geometric shapes such as rectangles and circles. Here the individual performance of the proposed layout generation methods are studied for arbitrarily-shaped bounding areas, both without and with restricted regions.

Without restricted regions:

This study is performed with systems that do not have restricted regions inside the bounding area. The FDL-NNG method, as shown in Fig. 3.18(a,



(a)



(b)

Figure 3.17: Results for **(a)** Average computational times required to attain a single feasible layout for 200 simulations for nC-2P systems; and **(b)** Box plot of computational times taken by different algorithms to attain a feasible layout for a 10C-2P system.

d), distributes the components and the interconnect routing network in the available arbitrarily-shaped space using spring force theory. FDL-FETS can be used in place of FDL-NNG here. The initial and final layouts attained with SPA and UST methods are shown in Fig. 3.19. The results obtained here clearly indicate that the performance of the three methods (FDLM, SPA and UST) is independent of the shape of the packing area.

With restricted regions:

The presence of restricted spaces in between the components does not affect the performance of the SPA and UST methods, as shown in Fig. 3.20. Such restricted regions are treated as obstacles along with other system components. FDL-NNG performance, however, depends on the way these restricted spaces are treated, either as movable obstacles (see Figs. 3.18(b, e)) or fixed obstacles (see Figs. 3.18(c, f)). If the obstacles are movable, then they can be treated as nodes in the spring network and a feasible layout is attained as usual. But for fixed obstacles, the final solution depends on the initial component and interconnect locations, and might result in an infeasible layout in most cases, as shown in Fig. 3.18(f), due to limited space availability between fixed obstacles.

3.10 Discussion

3.10.1 Observations from Case Studies:

The primary lessons learned from the above three case studies are discussed below:

1. The FDL methods are expensive compared to SPA and UST but powerful because they have the ability to simultaneously remove overlapping amongst components and untangle interconnects. The FDL-NNG and FDL-FETS methods can be utilized to handle large-scale problems with relaxed bounding area constraints. In addition, by varying the natural spring length and boundary forces, different sized layouts can be obtained.

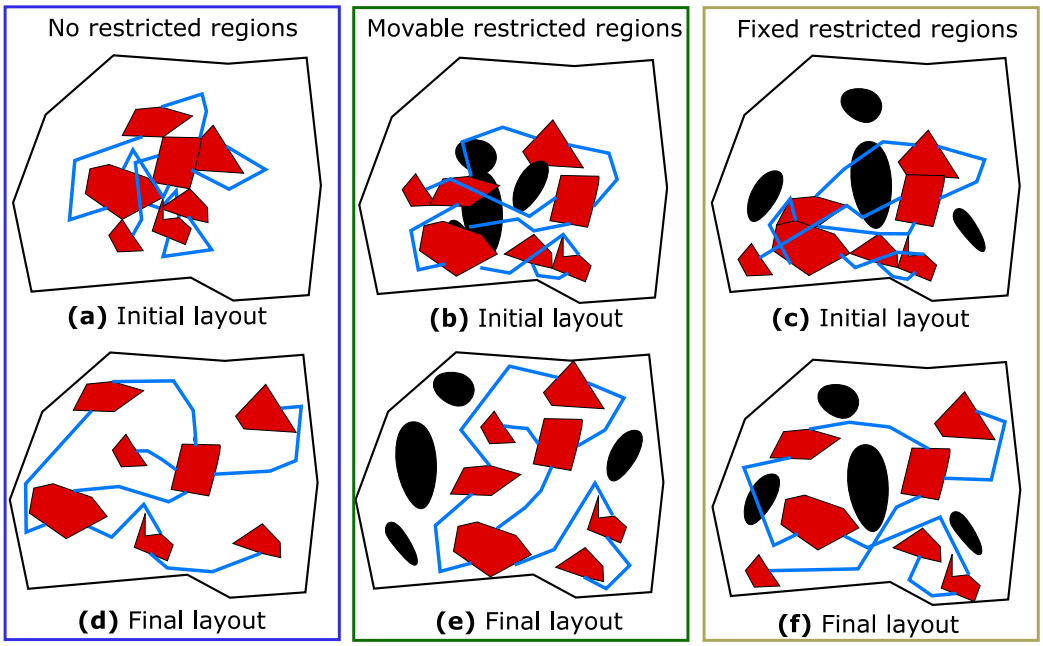


Figure 3.18: Illustration of final layouts obtained using FDLM with arbitrarily-shaped bounding boxes without and with restricted regions respectively.

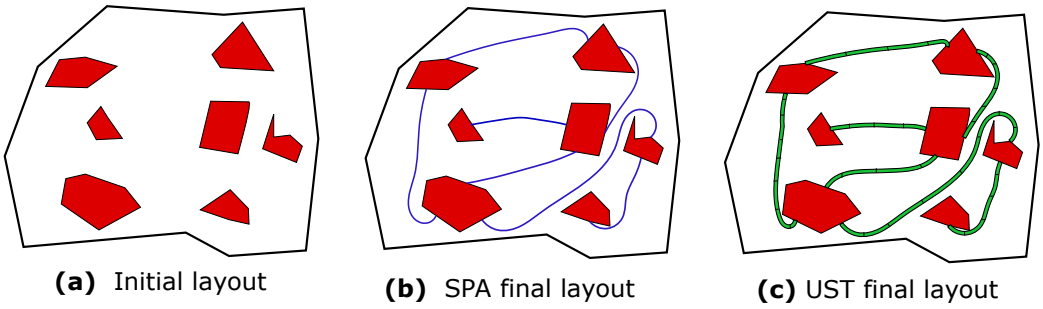
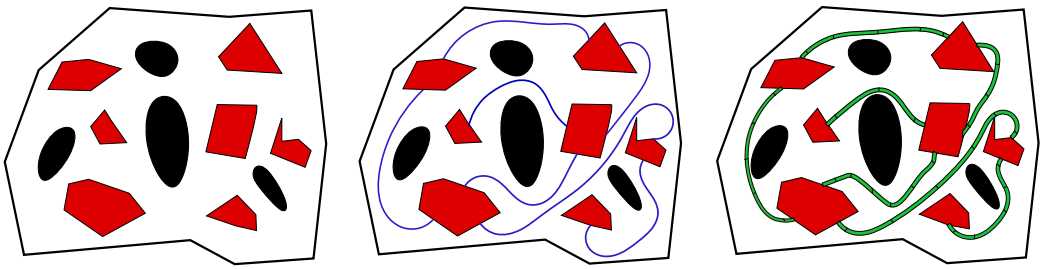


Figure 3.19: Illustration of final layouts obtained using SPA and UST methods respectively with arbitrarily-shaped bounding boxes *not containing restricted regions*.



(a) Initial layout **(b)** SPA final layout **(c)** UST final layout

Figure 3.20: Illustration of final layouts obtained using SPA and UST methods respectively with arbitrarily-shaped bounding boxes *containing restricted regions*.

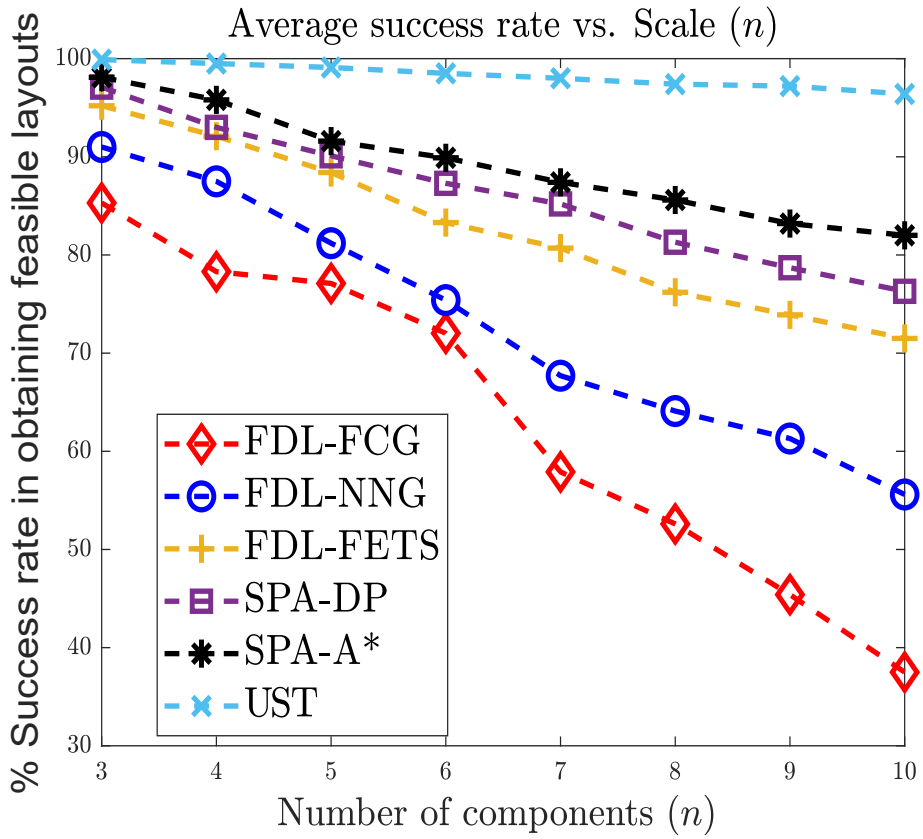


Figure 3.21: Average success rates to a attain feasible layout over 500 simulations for nC-2P systems.

2. The SPA method (A* algorithm) is very efficient in performing complex routing and scales up well. Core concerns include that it may become less efficient for finer grids required for large-scale systems and those with complex geometries, that it may not access all STs, and that the number of routing sequences that must be explored increases quickly with the number of interconnects.
3. The FDLM multi-component overlap removal and SPA routing can potentially form a *powerful tool* to handle large-scale problems.
4. The UST is efficient and comparable to the SPA method but is better in terms of providing unique geometric layouts at a higher success rate compared to the SPAs. The SPAs might fail for complex topologies, but UST intrinsically has the potential to find all STs. However, the SPA success rate can be improved by using a finer grid at the expense of its computational efficiency. Alternative UST strategies may be possible that avoid ST generation failure due to geometric limitations.
5. It must be noted that the UST was very efficient not only for the nC-2P type of systems, but performed really well with systems having varying ports, complex geometries, and restricted regions.
6. The FDL-FCG method, on average, has a low success rate with increasing n compared to the other methods. Relaxing bounding area constraints via the FDL-FETS method improves the success rate significantly.

3.10.2 Practical design guidelines:

Based on our observations, some practical guidelines for application of the proposed layout generation methods are outlined below:

1. The FDLM is a powerful method to untangle all the components and interconnects, provided that the 2D system has no fixed obstacles inside it or bounding box area constraints that limit complete untangling of the routing network. The FDLM is useful when the component and interconnect network is overlapped initially and no fixed locations are specified.
2. SPA and UST are valuable algorithms for generating complex layouts where the 2D system contains many components, interconnects, re-

stricted regions inside the domain, and arbitrarily-shaped bounding areas with tight spatial constraints.

3. SPA and UST are more effective than FDLM when component locations are fixed initially, and when multiple layouts with different interconnect routing networks need to be generated.
4. In cases where the interconnects may need to pass through way-points between two connected ports, SPA and UST can be utilized. For example, if a cooling pipe should be in contact with a particular heat element on its path, then SPA or UST methods can be used.

3.11 Summary

In this chapter, novel and computationally efficient methods for generating spatially-feasible 2D geometric layouts for spatial packing of interconnected components have been presented. Final 2D design layouts attained from these algorithms in stage 1 can be used as initial designs for performing simultaneous physics-based packing and routing optimization in stage 2 presented in chapter 4. The proposed force-directed method presented in this chapter showed how it effectively generates feasible initial graphs with simultaneous packing and routing. An important advantage of this method is explicit interference detection constraints are not required between different geometric elements in a multi-component interconnected system. Furthermore, very efficient extensions of the shortest path planning A* algorithm were demonstrated that can perform complex interconnect routing for a fixed multi-component spatial topology. Depending on the choice of the sequence of the component to component connections, different feasible layouts and STs can be attained. Finally, a systematic way of generating unique STs was presented. Based on direct and indirect connections between components, all interconnect topological configurations can be obtained. The layout algorithms have been compared for their efficiency and success in generating feasible layouts, and a separate example study was illustrated to showcase their performance in complex geometric environments. Practical design guidelines are also outlined for engineers to utilize these methods for different problem types.

CHAPTER 4

THE SPI2 DESIGN FRAMEWORK: STAGE 2

4.1 Introduction

In this chapter [1], stage 2 of the SPI2 design framework is presented using physics-based topology optimization method for optimizing the layout of component-routing systems. Gradient-based topology optimization techniques are used to simultaneously optimize both component locations and routing paths of component interconnects. In addition to geometric considerations, this method supports optimization based on system behavior by including physics-based objectives and constraints. Multiple physics domains are modeled using lumped parameter and finite element models. A geometric projection for components of arbitrary polygonal shape is developed along with sensitivity analysis. Several thermal-fluid systems are optimized to demonstrate the use of this method in this chapter.

4.1.1 Chapter Objectives and Contributions

The primary objective of this chapter is to demonstrate the use of gradient-based topology optimization on fluid-thermal system layout problems. The proposed method combines both the component placement and interconnect routing problems, in addition to using physics-based models for design comparison. The core contributions of this chapter are as follows:

1. A novel technique is presented that supports simultaneous optimization of component placement and interconnect paths, whereas existing methods treat component layout and interconnect routing separately (e.g., optimal routing with fixed layout).

[1] The geometric projection method presented in this chapter in based on work completed in Ref. [87] in collaboration with Alex Jessee (2019 MS graduate from Aerospace Engineering, University of Illinois).

2. Physics-based objectives and constraints were incorporated into the optimization problem, in addition to geometric constraints that prevent interference between components and interconnects. Both 1D lumped parameter and 2D finite element physics models are used within a single optimization problem to support physics-based evaluation.
3. The geometric projection method (GPM) of Norato et. *al* [83], which is an alternative to the well-established SIMP (Solid Isotropic Material with Penalization) method design parameterization [138] is utilized for solving the optimization problem.
4. The effectiveness of the proposed GPM is demonstrated via the solution of several test cases that utilize physics-based simulations including industry relevant problem such as a hybrid unmanned aerial vehicle (UAV) and the underhood of an automotive fuel-cell (AFC) packaging system .

4.2 Geometric Projection Method

The new simultaneous approach makes significant system volume reduction possible. The projection method of Norato et *al.* [83] is extended to allow components of arbitrary polygonal shape to be projected. Sensitivity analysis for this projection is provided to allow the efficient use of gradient-based optimization methods. Examples presented later in this chapter consider thermal conduction on the continuum level using the finite element method, and a lumped parameter pipe flow model. The methods presented here, however, could be extended to model other combinations of physics; for example, thermal-electric or structural-fluid systems. Section 4.3 presents the models used to simulate the physics response of the system. Section 4.4 states the optimization problem and presents the derived function sensitivities. Finally, the design method is demonstrated in Sections 4.5-4.6 through the optimization of several 2D SPI2 interconnected multi-component systems.

4.3 Physics models

4.3.1 Steady state thermal conduction

Temperature distribution was modeled on the continuum level using the finite element method. Two main assumptions were made for thermal modeling of SPI2 system layouts in this dissertation:

1. Only the steady-sate temperature distribution across the design domain and the components and fluid network is modeled, neglecting the transient thermal effects of the SPI2 system. This is because steady-state thermal conduction and convection capture the worst-case scenario of this class of systems in operation where maximum duration of system activity occurs. For instance, in an aircraft flight mission, the take-off and landing duration are negligible when compared to mid-air flight duration where the system operating conditions are almost constant and extreme due to lower ambient temperatures. Thus, transient thermal modeling was considered out of scope of this dissertation and is an aspect of future physics-based modeling work.
2. The effect of thermal expansion due to change in temperature on the geometry of the domain, components, and the pipe network is assumed to be negligible for modeling purposes.

The strong form of the boundary value problem for heat conduction is given by:

$$\nabla \cdot (\boldsymbol{\kappa} \nabla T(\boldsymbol{x})) + Q = 0, \quad \boldsymbol{x} \text{ in } \Omega \tag{4.1}$$

$$T(\boldsymbol{x}) = T^*, \quad \boldsymbol{x} \text{ on } \Gamma_T \tag{4.2}$$

$$\boldsymbol{n} \cdot (\boldsymbol{\kappa} \nabla T(\boldsymbol{x})) = q^*, \quad \boldsymbol{x} \text{ on } \Gamma_q, \tag{4.3}$$

where $\boldsymbol{\kappa}$ is the matrix of thermal conduction coefficients, $T(\boldsymbol{x})$ is the temperature solution field, Q is heat flux per unit volume in the domain, and $\boldsymbol{n}$ is the unit normal to the domain boundary. Temperature, T^*, and heat flux, q^*, boundary conditions are applied on the Γ_T and Γ_q portions of the domain boundary, respectively. Detailed derivation of the finite element equations and implementation can be found in Ref. [139]. Here we will skip to the final equation that solves for temperatures at the nodes of the finite element mesh, which is obtained by discretizing the boundary value problem in

Eqns. (4.1)-(4.3) using the finite element method.

$$\boldsymbol{K}\boldsymbol{T} = \boldsymbol{P} \tag{4.4}$$

Equation (4.4) is solved for the temperature field vector $\boldsymbol{T}$, where $\boldsymbol{K}$ is the global thermal stiffness matrix assembled from element stiffness matrices, $\boldsymbol{k}_{\text{el}}$, defined in Eqn. (4.5), and $\boldsymbol{P}$ is the global load vector assembled from element load vectors, $\boldsymbol{p}_{\text{el}}$, defined in Eqn. (4.6).

$$\boldsymbol{k}_{\text{el}} = \int_{\Omega_e} \boldsymbol{B}^T \boldsymbol{\kappa} \boldsymbol{B} d\Omega - \int_{\partial\Omega_h} h \boldsymbol{N} \boldsymbol{N}^T d\partial\Omega_h \tag{4.5}$$

$$\boldsymbol{p}_{\text{el}} = \int_{\Omega_e} Q \boldsymbol{N} d\Omega + \int_{\partial\Omega_h} h T_{\text{env}} \boldsymbol{N} d\partial\Omega_h. \tag{4.6}$$

Here, $\boldsymbol{N}$ and $\boldsymbol{B}$ are element shape function and shape function gradients, respectively. These equations also include convection boundary conditions on the $\partial\Omega_h$ portion of the boundary. The temperature of the convecting fluid is T_{env}, and the convection coefficient is h (assumed constant here).

Here the geometric projection method of Norato et *al* [83] is used. In the original projection method work, the new parameterization approach was used to optimize structures while ensuring that the resulting design could be made from stock materials, such as structural beams with standard shapes and sizes. It is discovered that this method can be extended beneficially to the combined layout and routing optimization problem. The geometric projection method is used to create routing designs that can be manufactured out of standard circular cross-section pipes. The geometric parameterization involves design variables that facilitate convenient derivation of lumped parameter model (Sec. 4.3.2) sensitivities. The remainder of this section will give a brief overview of the geometric projection method, and detail changes made for use with the routing and packaging problem.

In the projection method, each element in the mesh is assigned a density parameter ρ_i with a value between zero and one. Solid material corresponds to $\rho_i = 1$, and void material corresponds to $\rho_i = 0$. The material properties for each element stiffness matrix $\boldsymbol{k}_i$ are scaled by $0 \leq \rho_i \leq 1$. Leaving out

the convection boundary condition term, the element stiffness matrix is:

$$\boldsymbol{k}_i = (\rho_{\min} + (1-\rho_{\min})\rho_i^p) \int_{\Omega_e} \boldsymbol{B}^T \boldsymbol{\kappa} \boldsymbol{B} d\Omega = (\rho_{\min} + (1-\rho_{\min})\rho_i^p)\boldsymbol{k_0}, \quad (4.7)$$

where p is a penalization parameter used to penalize intermediate densities between 1 and 0, leading to a projection with less gray area between solid and void. The convection boundary condition term is independent of ρ_i, so can be omitted without loss of generality. If a regular mesh with all elements being the same shape and size is used, then the second form of Eqn. (4.7) can be applied to reduce computational expense, since the integral term is the same for all elements. A minimum density, $\rho_{\min}$, is enforced to prevent singularity in the global stiffness matrix. For structural problems, the smallest $\rho_{\min}$ that prevents ill-conditioning is used. In thermal problems, however, a physically-meaningful minimum density can be chosen to simulate the thermal conductivity of the surrounding medium, for example, air.

The density of each element is found by projecting geometric shapes onto the mesh. Norato et *al.* proposed bars with rounded ends as a shape which could be projected easily, and the same will be used here. Each bar involves three parameters: segment start and end points $\boldsymbol{x}_0$ and $\boldsymbol{x}_f$, and bar width w (Fig. 4.1). The parameter for out-of-plane thickness (that was presented in the original formulation) is left out here because the new method presented here requires that bars are not removed. This is important because the bars form a flow network, and bar removal could break flow paths. The signed distance between a bar q and an element with center at $\boldsymbol{p}$ is:

$$\phi_q(d_q(\boldsymbol{x}_{q_0}, \boldsymbol{x}_{q_f}, \boldsymbol{p}), w) = d_q(\boldsymbol{x}_{q_0}, \boldsymbol{x}_{q_f}, \boldsymbol{p}) - \frac{w}{2}, \quad (4.8)$$

where d_q is the distance between the segment q and point $\boldsymbol{p}$. See Ref. [83] for the distance calculation. A circle of radius r is placed at the element center. The density assigned to each element is the area of the circle covered by the bar divided by total area of the circle—see the shaded area of Fig. 4.1. The

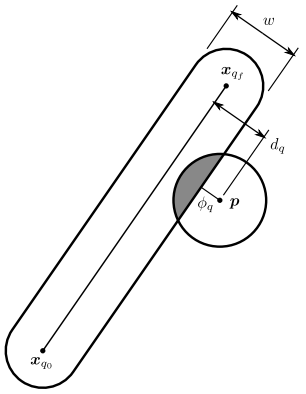


Figure 4.1: Bar projection

density as a function of signed distance is given by:

$$\rho_q(d_q(\boldsymbol{x}_{q_0}, \boldsymbol{x}_{q_f}, \boldsymbol{p}), r) = \begin{cases} 0 & \phi_q > r \\ \frac{1}{\pi r^2}\left[r^2 \cos^{-1}\left(\frac{\phi_q(d_q)}{r}\right) - \phi_q(d_q)\sqrt{r^2 - \phi_q(d_q)^2}\right] & -r \leq \phi_q \leq r \\ 1 & \phi_q < -r \end{cases} \tag{4.9}$$

The radius r determines the width of the grey area projected on to the mesh by the bar. A smaller radius will more accurately represent the bar geometry as a projection of mostly ones and zeros. To ensure that any element which touches a bar has a nonzero density, a radius that circumscribes the square elements is used in this chapter. The radius must be less than half of the bar width in order for Eqn.(9) to correctly calculate the area intersected by the circle and bar.

In the combined layout and routing optimization problem, components must also be included in the finite element analysis model. components are approximated as polygonal shapes with straight edges. Each component will be defined by a reference point, $\boldsymbol{c}_d$, and a set of vectors, $\boldsymbol{b}_i$, pointing from the reference point to polygon vertices. The component densities are calculated by first projecting each edge of the polygon as a rounded bar, and then filling in elements inside the polygon with density of 1 (Fig. 4.2). Densities for each edge $\tilde{\rho}_e$ are calculated using Eqn. (4.9). End points of edge segments, $\boldsymbol{x}_{e_0}$ and $\boldsymbol{x}_{e_f}$, are found by:

$$\boldsymbol{x}_{e_0} = \boldsymbol{x}_i = \boldsymbol{c_d} + \boldsymbol{b}_i \tag{4.10}$$

$$\boldsymbol{x}_{e_f} = \boldsymbol{x}_{i+1} = \boldsymbol{c_d} + \boldsymbol{b}_{i+1}. \tag{4.11}$$

Figure 4.2: component projection

The densities of all the edges in the component are then merged using a p-norm approximation of the maximum density, as quantified in Eqn. (4.12):

$$\rho_d(\boldsymbol{c_d}, \boldsymbol{p}) = \left[\sum_{e=1}^{N_e} (\tilde{\rho}_e(d_e(\boldsymbol{x}_{e_0}, \boldsymbol{x}_{e_f}, \boldsymbol{p})))^p \right]^{\frac{1}{p}}. \tag{4.12}$$

After merging edge densities, all elements with centers inside the polygon are assigned $\rho_d = 1$. Elements with centers inside the polygon can be found using the MATLAB function `inpoly()`, or by using the algorithm described in Ref. [140].

Finally, the density used in Eqn. (4.7) is calculated by merging densities of all bars and components in Eqn. (4.13). In the temperature field solution, heat is being conducted between the components and interconnects because the merged density field is used to calculate the stiffness matrix.

$$\rho_i = \left[\sum_{q=1}^{N_q} (\rho_q(d_q(\boldsymbol{x}_{q_0}, \boldsymbol{x}_{q_f}, \boldsymbol{p_i})))^p + \sum_{d=1}^{N_d} (\rho_d(\boldsymbol{c}_d, \boldsymbol{p}_i))^p \right]^{\frac{1}{p}}. \tag{4.13}$$

Section 4.3.2 introduces pipe elbows which form a smooth radius at the intersection of two straight segments. These curved pipe segments are used in the pipe flow model but are not modeled in the projection. The projection at the intersection of two segments is therefore an approximation based on the assumption of straight pipes with an elbow radius of zero. In addition,components may also add or remove heat from the domain. The projection in Eqn. (4.12) will be used to model this effect. Rather than assuming a constant internal heat generation Q across all components, each component will have its own Q_d value. Element load vectors are then modified using

this Q_d and the component density.

$$\boldsymbol{p}_e = \sum_{d=1}^{N_d} \rho_{\mathrm{de}}^p Q_d \int_{\Omega_e} \boldsymbol{N} d\Omega_e = \sum_{d=1}^{N_d} \rho_{\mathrm{de}}^p Q_d \boldsymbol{p}_0 \tag{4.14}$$

The convection boundary condition term of Eqn. (4.6) is omitted again as it will not be scaled with density.

4.3.2 Lumped parameter pipe flow model

This section presents a lumped parameter pipe flow model for the pressure throughout the flow loop. Pressure is a factor that influences pump power consumption, which is important to reduce. The lumped parameter model uses empirical relations to approximate flow loop sections using only a small number of parameters [141]. The lumped parameter model is computationally inexpensive compared to computational fluid dynamics (CFD) models, and provides suitable accuracy (important properties for design optimization). The following assumptions have been made in the pipe flow model presented here:

1. Flow is incompressible
2. All components are connected in series with no branches
3. Everything is in the same plane relative to ground (no height change)
4. Flow rate at the inlet is known
5. Flow is turbulent everywhere

Most of these assumptions could be relaxed if more accuracy is desired, with the penalty of increased computational expense and more complex sensitivity analysis.

We begin with Eqn. (4.15), which is derived in detail from an energy balance in Ref. [141]. Each term in this equation is formulated to have units of length. This equation relates head loss H_L to pressure, P, and velocity, V, at points 1 and 2 in the flow loop.

$$H_L = \frac{P_1 - P_2}{\rho_w} + \frac{V_1^2 - V_2^2}{2g} \tag{4.15}$$

Here, ρ_w is the weight density, and g is gravitational acceleration. Solving Eqn. (4.15) for the pressure difference (in terms of head) between two points

produces:

$$\frac{P_1 - P_2}{\rho_w} = H_L - \frac{V_1^2 - V_2^2}{2g} \tag{4.16}$$

The flow rate in the system is known, so fluid velocity at any point can be calculated easily using:

$$V_i = \frac{Q}{A_i}. \tag{4.17}$$

Here, Q is the volumetric flow rate (uniform for a series flow loop), and A_i is the cross sectional area of the pipe at location i. Head loss is determined next, which is a proxy metric for energy loss between points 1 and 2 for reasons other than velocity change. Head loss, in units of length, is a standard metric used in describing flow system properties, including pump efficiency and system characterization curves [142]. Models for estimating head loss for many different pipe flow system components can be found in Ref. [141], but here only two will be of interest: 1) losses due to friction between the fluid and pipe wall (sometimes called major loss), and 2) losses due to elbows connecting straight segments of pipe. Each straight segment of pipe and each elbow has a loss coefficient, K, assigned based on geometry. Total head loss for n_{pe} pipe elements in series can be calculated by combining the loss coefficients as follows:

$$H_L = \frac{\dot{w}^2}{2g\rho_w^2} \sum_{i=1}^{n_{\text{pe}}} \frac{K_i}{A_i^2}, \tag{4.18}$$

where $\dot{w}$ is the weight flow rate.

The loss coefficient for a straight segment of pipe with length l_i and diameter d_i is:

$$K_i^s = f_i \frac{l_i}{d_i}, \tag{4.19}$$

and for an elbow with bend angle α_i and bend radius r_i the loss coefficient is:

$$K_i^e = f_i \alpha_i \frac{r_i}{d_i} + (0.1 + 2.4 f_i) \sin\left(\frac{\alpha_i}{2}\right) + \frac{6.6 f_i \left(\sin\left(\frac{\alpha_i}{2}\right) + \sqrt{\sin\left(\frac{\alpha_i}{2}\right) + \varepsilon_s}\right)}{\left(\frac{r_i}{d_i}\right)^{4\alpha_i/\pi}} - f_i \frac{2l_c}{d_i}. \tag{4.20}$$

See Fig. 4.3 for a description of elbow geometry. In Eqn. (4.20), a small perturbation ε_s has been added to make the expression differentiable at $\alpha = 0$. The first term in Eqn. (4.20) accounts for frictional losses across the elbow arc length. The final term reduces the loss coefficient to account for the length

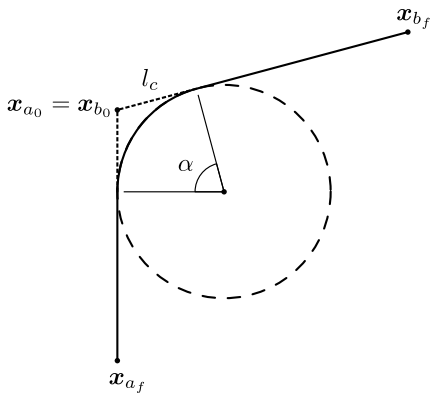


Figure 4.3: Pipe elbow geometry

of straight pipe that is overlapped by the elbow, l_c. Implementing loss coefficient calculations in this way allows each pipe section to be modular. If the length of straight pipe were reduced directly at the straight loss coefficient calculation, information about the connecting pipe and elbow would be needed. The bend angle is found by defining two vectors—$\boldsymbol{a}$ and $\boldsymbol{b}$—based on the endpoints of two connected segments:

$$\boldsymbol{a} = \boldsymbol{x}_{a_f} - \boldsymbol{x}_{a_0} \tag{4.21}$$

$$\boldsymbol{b} = \boldsymbol{x}_{b_f} - \boldsymbol{x}_{b_0}. \tag{4.22}$$

From the definition of the dot product, we obtain:

$$\theta = \cos^{-1}(v), \tag{4.23}$$

where:

$$v = (1 - \varepsilon_c)\frac{\boldsymbol{a} \cdot \boldsymbol{b}}{||\boldsymbol{a}||||\boldsymbol{b}||}. \tag{4.24}$$

A perturbation ε_c is incorporated into Eqn. (4.24) to restrict the range such that $v \in [0, (1 - \varepsilon_c)]$. This is done to prevent the derivative of θ from being undefined when $v = 1$. The angle α in Eqn. (4.20) is defined as the supplementary angle of θ:

$$\alpha = \pi - \theta. \tag{4.25}$$

The clipped length is calculated using:

$$l_c = r_i\sqrt{\frac{1+v}{1-v}}. \tag{4.26}$$

The *friction factor* (f_i) appears in both loss coefficient equations. The fric-

tion factor is a function of Reynolds number, and can also account for pipe wall roughness. A variety of approximate models for friction factor have been developed based on experimental results. Here, the equation for turbulent flow in smooth pipes proposed by Blasius [143] is used to estimate f_i:

$$f_i = 0.3164 Re_i^{-0.25}, \tag{4.27}$$

with Reynolds number:

$$Re_i = \frac{V_i d_i \rho_m}{\mu}, \tag{4.28}$$

where ρ_m is mass density and μ is fluid viscosity. The use of the thermal conductivity and pipe flow models in optimization will be discussed in the following section.

4.4 Optimization problem and sensitivity analysis

This section presents the optimization problem formulation, as well as derivations for the sensitivities needed to use gradient-based optimization methods. The models presented in Section 4.3 lead to a specific choice of design variables which will be discussed in this section. Furthermore, this set of design variables can be used to define geometric constraints which are needed to prevent interference between different components of the system. The system being optimized here consists of a number of components and the interconnects routed between components.

The goal is to find the optimal system layout. System layout is defined as the placement of the components, the routing of component interconnects, as well as limited sizing parameters (such as interconnect diameter, assuming tubular connections between components). Each component and interconnect are parameterized in a general manner to help simplify sensitivity calculations and support object-oriented code implementation. More complex components and routing can be modeled without significant additional work due to the use of object-oriented programming.

In addition to $\boldsymbol{c}_d$ and $\boldsymbol{b}_i$ introduced in Sec. 4.3, components may have ports with location $\boldsymbol{p}_i$ relative to the reference point. Ports are the required locations for interconnect attachment to each component. As the reference point moves, the polygon and ports will move with it. component shape, size,

and port location are held fixed during the optimization, so the only design variable for each component is $\boldsymbol{c}_d$. In some optimization studies, it may be useful to omit the reference point corresponding to a component from the set of optimization variables, holding the component fixed in a particular location. This can also be used to specify fixed inputs and outputs of the flow loop.

Pairs of ports are connected via physical interconnects, and component connection topology is assumed to be given (and unchanging) here. Each interconnect is represented here using one or more straight geometric segments. Increasing the number of segments in a connection supports consideration of approximately curved (and more complex) interconnect geometries, but increases computational expense. Interconnect segment i is associated with parameters for its start and end points, $\boldsymbol{x}_{i_0}$ and $\boldsymbol{x}_{i_f}$, respectively, as well as width, w_i. All of these quantities are optimization variables.

The models presented in Sec. 4.3 are reformulated in terms of the above parameterization to simplify sensitivity calculation. A tradeoff, however, exists between ease of sensitivity calculation and problem conditioning. Specifically, the connection between two interconnect segments, or a segment and a port, present a challenge. As defined above, each system element has its own independent parameters, so connections are free to be broken. There are two ways to solve this issue. The first method attempts to enforce constraints between connected points. Such constraints could be implemented as either linear equality or nonlinear inequality constraints, shown in Eqns. (4.29) and (4.30), respectively:

$$x_i - x_j = 0 \tag{4.29}$$

$$(x_i - x_j)^2 - \varepsilon \leq 0, \tag{4.30}$$

where x_i and x_j are copies of the same parameter across two different elements. Equation (4.30) constrains a norm of the parameter error to be within a tolerance ε, approximating the equality constraint in Eqn. (4.29).

While this may be useful for use with optimization algorithms that do not support equality constraints, it can also help in cases where satisfaction of equality constraints is difficult and causes optimization algorithm convergence problems. Equation (4.30) is a relaxation of the original equality constraint. A small value of ε provides an accurate approximation, but can

degrade problem conditioning. Using a large value of ε improves problem conditioning, but reduces solution accuracy. Replacing linear with nonlinear constraints can also impact computational expense. A related alternative remedy is to increase constraint satisfaction tolerances for these consistency constraints within optimization algorithm settings, but this may not be a practical approach in general. In numerical experiments performed for this study, the value of ε was observed to be a critical parameter to tune the balance between problem difficulty in satisfying constraints (problem stiffness or conditioning), while maintaining geometric consistency.

A second approach to ensure geometric consistency is to define an implicit parameterization, making use of two design variable vectors: the *expanded* and *reduced* design variable vectors. Consider a system with n_d components and n_s routing segments. The expanded design variable vector:

$$\boldsymbol{z}' := [\boldsymbol{c}_1, ..., \boldsymbol{c}_{n_d}, \boldsymbol{x}_{1_0}, \boldsymbol{x}_{1_f}, w_1, ..., \boldsymbol{x}_{n_{s0}}, \boldsymbol{x}_{n_{sf}}, w_{n_s}]^T$$

contains all the parameters discussed above for each element in the layout. The reduced design variable vector contains only the reference points of free components, $\boldsymbol{c}_d^f$, locations where routing segments meet, $\boldsymbol{x}_i^f$, and the width of each connection, w_i. It is assumed that the width of all segments that connect two ports are the same (e.g., representing interconnects with uniform properties, such as electrical wiring or piping). The reduced design vector is then:

$$\boldsymbol{z} := [\boldsymbol{c}_1^f, ..., \boldsymbol{c}_{n_{\text{df}}}^f, \boldsymbol{x}_1^f, ..., \boldsymbol{x}_{n_{sf}}^f, w_1, ..., w_{n_c}]^T,$$

where n_{df} is the number of free components, n_{sf} is the number of points where routing segments meet, and n_c is the number of connections between pairs of ports. With these two vectors introduced, a mapping can be defined from the reduced to the expanded design variable vectors, defined in matrix form as:

$$\boldsymbol{z}' = \boldsymbol{M}\boldsymbol{z} + \boldsymbol{P} \tag{4.31}$$

where $\boldsymbol{M}$ is a binary mapping (or selection) matrix. It is derived by identifying which elements of $\boldsymbol{z}$ and $\boldsymbol{z}'$ correspond to each other. The vector $\boldsymbol{P}$ represents fixed components that are not in the reduced design variable vector, but have a fixed value throughout the optimization. Also, segment endpoints that are connected to component ports are mapped to the component reference point in the reduced design variable vector by including an

offset corresponding to the port location $\boldsymbol{p}_i$. It should be noted that, in general, the mapping matrix is not invertible, so the inverse mapping based on Eqn. (4.31) may not be possible.

This mapping preserves the simplified sensitivity calculations described above, while eliminating the need for consistency constraints. All calculations of objective functions and constraints, and their sensitivities, are performed in terms of the expanded design variable vector. Next, the sensitivities are computed in terms of the reduced design variables by using the chain rule:

$$\frac{df}{d\boldsymbol{z}} = \frac{\partial f}{\partial \boldsymbol{z}'}\frac{d\boldsymbol{z}'}{d\boldsymbol{z}}, \tag{4.32}$$

$$\frac{d\boldsymbol{z}'}{d\boldsymbol{z}} = \boldsymbol{M}. \tag{4.33}$$

The reduced design variable vector is used as the optimization vector by the solution algorithm. To maintain correspondence between reduced and expanded vectors, the mapping defined in Eqn. (4.31) is applied each time the reduced design variable vector is updated.

Now that the parameterization of the design space has been determined, the complete optimization problem formulation can be presented:

$$\begin{aligned}
\min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}, \boldsymbol{T}) && \text{(4.34a)}\\
\text{s.t.:} \quad & \boldsymbol{g}_{\text{phys}}(\boldsymbol{x}, \boldsymbol{T}) \leq 0 && \text{(4.34b)}\\
& \boldsymbol{g}_{\text{dd}}(\boldsymbol{x}) \leq 0 && \text{(4.34c)}\\
& \boldsymbol{g}_{\text{sd}}(\boldsymbol{x}) \leq 0 && \text{(4.34d)}\\
& \boldsymbol{g}_{\text{ss}}(\boldsymbol{x}) \leq 0 && \text{(4.34e)}\\
\text{where:} \quad & \boldsymbol{K}(\boldsymbol{x})\boldsymbol{T} = \boldsymbol{P}(\boldsymbol{x}) && \text{(4.34f)}
\end{aligned}$$

Here $f(\boldsymbol{x}, \boldsymbol{T})$ is the objective function and $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{T})$ are constraint functions. In general, these functions may depend both on design ($\boldsymbol{x}$) and state ($\boldsymbol{T}$) variables. The function $f(\cdot)$
can be any one of the candidate objectives discussed later in Sec. 4.4.1. The constraints $\boldsymbol{g}_{\text{phys}}(\boldsymbol{x}, \boldsymbol{T})$ are constraints that depend both on design and on solutions to the physics models (i.e. the value of the state vector $\boldsymbol{T}$). The interference constraints $\boldsymbol{g}_{\text{dd}}(\boldsymbol{x})$, $\boldsymbol{g}_{\text{sd}}(\boldsymbol{x})$, and $\boldsymbol{g}_{\text{ss}}(\boldsymbol{x})$ prevent interference

between two components, one routing segment and one component, and two routing segments, respectively. These constraints are independent of any physics models, so they are all explicit functions of the design variables.

4.4.1 Objective function and physics-based constraints

This section presents objective function options and their derivatives. Objective functions and physics-based constraints are discussed together because they both depend on design and state variable values. In addition, these functions are interchangeable as either objective or constraint functions.
The first candidate function, $f_1(\cdot)$, relates to the solution of the lumped-parameter flow model:

$$f_1(\boldsymbol{x}, \boldsymbol{T}) = H_L \tag{4.35}$$

The objective is to minimize the head loss (H_L) in the flow loop as calculated in Eqn. (4.18). When head loss is used as an objective or constraint, the radius of each pipe elbow is also included a design variable. The elbow radii r_i are appended to the end of expanded and reduced design vectors (with a one-to-one mapping). The total derivative using the chain rule is:

$$\frac{dH_L}{d\boldsymbol{x'}} = \frac{\partial H_L}{\partial \boldsymbol{x'}} + \sum_{i=1}^{n_{pe}} \frac{\partial H_L}{\partial K_i} \frac{dK_i}{d\boldsymbol{x'}}. \tag{4.36}$$

The design variable vector contains component reference point coordinates, bar end coordinates, bar widths, and elbow radii. For each routing segment, the pipe diameter will be equal to the bar width. The only nonzero elements of the explicit derivative $\partial H_L / \partial \boldsymbol{x'}$ are those corresponding to bar width:

$$\frac{\partial H_L}{\partial d_i} = -\frac{\pi d_i \dot{w}^2}{2g\rho_w^2 A_i^2} \sum_{i=1}^{n_{pe}} K_i \tag{4.37}$$

For each lumped parameter element, whether a straight section or elbow, the following equation applies:

$$\frac{\partial H_L}{\partial K_i} = \frac{1}{A_i^2} \frac{\dot{w}^2}{2g\rho_w^2}. \tag{4.38}$$

The final derivative in Eqn. (4.36), $dK_i / d\boldsymbol{x'}$, depends whether the element is a straight or elbow section. For a straight section, design variables are

segment end points, $\boldsymbol{x}_{i_0}$ and $\boldsymbol{x}_{i_f}$, and segment diameters d_i. The sensitives are given below:

$$\frac{dK_i^s}{d\boldsymbol{x}_{i_0}} = \frac{f_i}{d_i l_i}(\boldsymbol{x}_{i_0} - \boldsymbol{x}_{i_f}) \tag{4.39}$$

$$\frac{dK_i^s}{d\boldsymbol{x}_{i_f}} = \frac{f_i}{d_i l_i}(\boldsymbol{x}_{i_f} - \boldsymbol{x}_{i_0}) \tag{4.40}$$

$$\frac{dK_i^s}{dd_i} = -\frac{f_i l_i}{d_i^2} + \frac{l_i}{d_i}\frac{df_i}{dd_i} \tag{4.41}$$

$$\frac{df_i}{dd_i} = 0.25(0.3164)Re^{-1.25}\frac{4\dot{m}}{\pi\mu d_i^2}. \tag{4.42}$$

If the lumped parameter element is an elbow, the design variables are the four end points of connected segments, $\boldsymbol{x}_{a_0}$, $\boldsymbol{x}_{a_f}$, $\boldsymbol{x}_{b_0}$, and $\boldsymbol{x}_{b_f}$, diameter, d_i, and radius of the elbow, r_i. It is assumed that the diameters of connected segments are the same so there is only one diameter variable. The sensitivity of the elbow loss coefficient with respect to pipe diameter is:

$$\frac{dK_i^e}{dd_i} = \frac{\partial K_i^e}{\partial d_i} + \frac{\partial K_i^e}{\partial f_i}\frac{df_i}{dd_i}. \tag{4.43}$$

Equation (4.42) can be used again here. The partial derivatives are:

$$\frac{\partial K_i^e}{\partial d_i} = -f_i\alpha_i\frac{r_i}{d_i^2} + 6.6\left(\frac{4\alpha_i}{\pi}\right)f_i\left(\sin\left(\frac{\alpha_i}{2}\right) + \sqrt{\sin\left(\frac{\alpha_i}{2}\right) + \varepsilon_s}\right)r^{(-4\alpha_i/\pi)}d^{(4\alpha_i/\pi)-1} + f_i\frac{2l_c}{d_i^2} \tag{4.44}$$

$$\frac{\partial K_i^e}{\partial f_i} = \alpha_i\frac{r_i}{d_i} + 2.4\sin\left(\frac{\alpha_i}{2}\right) + \frac{6.6\left(\sin\left(\frac{\alpha_i}{2}\right) + \sqrt{\sin\left(\frac{\alpha_i}{2}\right) + \varepsilon_s}\right)}{\left(\frac{r}{d}\right)^{4\alpha_i/\pi}} - \frac{2l_c}{d_i} \tag{4.45}$$

With respect to elbow radius, the sensitivity is:

$$\frac{dK_i^e}{dr_i} = f_i\alpha_i\frac{1}{d_i} - 6.6\frac{4\alpha_i}{\pi}f_e\left(\sin\left(\frac{\alpha_i}{2}\right) + \sqrt{\sin\left(\frac{\alpha_i}{2}\right) + \varepsilon_s}\right)r^{(-4\alpha_i/\pi)-1}d^{(4\alpha_i/\pi)} - \frac{2f_i}{d_i}\frac{dl_c}{dr_i}. \tag{4.46}$$

The chain rule can be used to calculate sensitivities with respect to the four segment end points:

$$\frac{dK_i^e}{d\boldsymbol{x}_{a_0}} = \frac{\partial K_i^e}{\partial \alpha_i}\frac{d\alpha_i}{d\boldsymbol{x}_{a_0}} - \frac{2f_i}{d_i}\frac{dl_c}{d\boldsymbol{x}_{a_0}}, \tag{4.47}$$

with:

$$\frac{\partial K_i^e}{\partial \alpha_i} = f_i \frac{r_i}{d_i} + \frac{1}{2}(0.1 + 2.4 f_e) \cos\left(\frac{\alpha_i}{2}\right) + \frac{26.4}{\pi} f_i \left(\sin\left(\frac{\alpha_i}{2}\right) + \sqrt{\sin\left(\frac{\alpha_i}{2}\right) + \varepsilon_s}\right) \left(\frac{d}{r}\right)^{(4\alpha_i/\pi)} \ln\left(\frac{d}{r}\right) + 6.6 f_i \left(\frac{d}{r}\right)^{(4\alpha_i/\pi)} \left(\frac{1}{4} \frac{1}{\sqrt{\sin\left(\frac{\alpha_i}{2}\right) + \varepsilon_s}} \cos\left(\frac{\alpha_i}{2}\right) + \frac{1}{2} \cos\left(\frac{\alpha_i}{2}\right)\right). \quad (4.48)$$

The derivatives of angle α are:

$$\frac{d\alpha_i}{d\boldsymbol{x}_{a_0}} = \frac{\partial \alpha_i}{\partial v} \frac{dv}{d\boldsymbol{x}_{\boldsymbol{a_0}}}, \quad (4.49)$$

with:

$$\frac{\partial \alpha_i}{\partial v} = \frac{1}{\sqrt{1 - v^2}}. \quad (4.50)$$

The required derivatives of the clipped length are as follows:

$$\frac{dl_c}{dr_i} = \sqrt{\frac{1+v}{1-v}} \quad (4.51)$$

$$\frac{dl_c}{d\boldsymbol{x}_{a_0}} = \frac{\partial l_c}{\partial v} \frac{dv}{d\boldsymbol{x}_{a_0}} \quad (4.52)$$

where:

$$\frac{\partial l_c}{\partial v} = \frac{r_i}{(1-v)^2} \sqrt{\frac{1-v}{1+v}}. \quad (4.53)$$

The calculations in Eqns. (4.49) and (4.52) require derivatives of v with respect to the four bar end points:

$$\frac{dv}{d\boldsymbol{x}_{a_f}} = -\frac{dv}{d\boldsymbol{x}_{a_0}} = (1 - \varepsilon_c) \left[\frac{\boldsymbol{b}}{||\boldsymbol{a}||||\boldsymbol{b}||} - \frac{1}{||\boldsymbol{a}||^3 ||\boldsymbol{b}||} (\boldsymbol{a} \cdot \boldsymbol{b}) \boldsymbol{a}\right], \quad (4.54)$$

$$\frac{dv}{d\boldsymbol{x}_{b_f}} = -\frac{dv}{d\boldsymbol{x}_{b_0}} = (1 - \varepsilon_c) \left[\frac{\boldsymbol{a}}{||\boldsymbol{a}||||\boldsymbol{b}||} - \frac{1}{||\boldsymbol{a}||||\boldsymbol{b}||^3} (\boldsymbol{a} \cdot \boldsymbol{b}) \boldsymbol{b}\right]. \quad (4.55)$$

A second objective function option is to minimize one of the component temperatures. Temperature at the component center is used here, so the objective function is:

$$f_2(\boldsymbol{x}, \boldsymbol{T}) = T(\boldsymbol{c}_d). \quad (4.56)$$

This depends on the solution of the thermal conduction physics problem.

Sensitivities of functions depending on solution of a finite element problem can be calculated using the adjoint method [144]:

$$\frac{df_2}{dx'_i} = \frac{\partial f_2}{\partial x'_i} + \boldsymbol{\Psi}^T \frac{d\boldsymbol{R}}{dx'_i}, \tag{4.57}$$

where $\boldsymbol{\Psi}$ is the *adjoint vector*, which can be calculated with the following equation:

$$\boldsymbol{\Psi} = \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{y}}\right]^{-T} \left[\frac{\partial f_2}{\partial \boldsymbol{y}}\right]^T. \tag{4.58}$$

The residual $\boldsymbol{R}$ comes from manipulating Eqn. (4.4) so that one side is zero:

$$\boldsymbol{R} = \boldsymbol{K}\boldsymbol{T} - \boldsymbol{P} = \boldsymbol{0} \tag{4.59}$$

The vector $\boldsymbol{y}$ in Eqn. (4.58) is the unknown vector. In the finite element problem the unknowns are $\boldsymbol{y} = [\boldsymbol{P}^p, \boldsymbol{T}^f]^T$. The vector $\boldsymbol{P}^p$ is the flux at nodes where prescribed temperature boundary conditions are applied, and $\boldsymbol{T}^f$ are temperatures at all remaining nodes. Equation (4.59) is partitioned into blocks, p and f, corresponding to the *prescribed* and *free* (unknown) degrees of freedom, respectively. The derivative of the residual with respect to the unknown vector results in a partitioned matrix:

$$\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{y}}\right] = \begin{bmatrix} -\boldsymbol{I} & \boldsymbol{K}^{pf} \\ \boldsymbol{0} & \boldsymbol{K}^{ff} \end{bmatrix}. \tag{4.60}$$

The temperature at an arbitrary location in the domain needs to be calculated by interpolation from nodal temperatures:

$$T(\boldsymbol{c}_d) = \boldsymbol{N}^T(\xi, \eta)\boldsymbol{T}_{\text{el}}, \tag{4.61}$$

where ξ and η are the location of $\boldsymbol{c}_d$ in the local element coordinate system, and $\boldsymbol{T}_{\text{el}}$ is the vector of element nodal temperatures. The only nonzero derivative with respect to design variables $\boldsymbol{x}'$ is for the component reference point. Using the definition of the matrix $\boldsymbol{B} = d\boldsymbol{N}/d\boldsymbol{x}$, explicit derivatives of $f_2(\cdot)$ can then be calculated:

$$\frac{\partial f_2}{\partial \boldsymbol{x}'} = \boldsymbol{B}^T \boldsymbol{T}_{\text{el}} \tag{4.62}$$

$$\frac{\partial f_2}{\partial \boldsymbol{y}} = \boldsymbol{N}(\xi, \eta) \tag{4.63}$$

The derivative of the residual with respect to design variables is calculated using the chain rule.

$$\frac{d\boldsymbol{R}}{dx_i'} = \frac{\partial \boldsymbol{R}}{\partial \rho_e}\frac{d\rho_e}{dx_i'} \tag{4.64}$$

Finally, taking into account Eqns. (4.4) and (4.5), the total derivative is:

$$\frac{df_2}{dx_i'} = \frac{\partial f_2}{\partial x_i'} + \sum_{e=1}^{N_e} p(1-\rho_{\min})\rho_e^{(p-1)}\boldsymbol{\Psi}_e^T \boldsymbol{k}_0 \boldsymbol{T}_e \frac{d\rho_e}{dx_i'} + \sum_{d=1}^{N_d} Q_d \sum_{e=1}^{N_e} p\rho_{\text{de}}^{(p-1)}\boldsymbol{\Psi}_e^T \boldsymbol{p}_0 \frac{d\rho_{\text{de}}}{dx_i'}, \tag{4.65}$$

where N_e is the number of finite elements in the mesh, ρ_e is the element density from Eqn. (4.13), ρ_{de} is the component element density from Eqn. (4.12), and $\boldsymbol{\Psi_e}$ and $\boldsymbol{T}_e$ are the adjoint and temperature vectors corresponding to element degrees of freedom.

The derivative of the geometric projection is the final part of Eqn. (4.65). The derivative of density resulting from the merge in Eqn. (4.13) is:

$$\frac{d\rho}{dx_i'} = \sum_{q=1}^{N_q}\left(\frac{\rho_q}{\rho}\right)^{p-1}\frac{d\rho_q}{dx_i'} + \sum_{d=1}^{N_d}\left(\frac{\rho_d}{\rho}\right)^{p-1}\frac{d\rho_d}{dx_i'}. \tag{4.66}$$

See Ref. [83] for derivatives with respect to bar ends. For components, the derivative with respect to component centers must be calculated using:

$$\frac{d\rho_d}{d\boldsymbol{c}_d} = \rho_d^{1-p}\sum_{e=1}^{N_e}(\tilde{\rho}_e)^{p-1}\frac{\partial\tilde{\rho}_e}{\partial \boldsymbol{c}_d}, \tag{4.67}$$

where:

$$\frac{\partial\tilde{\rho}_e}{\partial \boldsymbol{c}_d} = \frac{\partial\tilde{\rho}_e}{\partial \boldsymbol{x}_{e_0}}\frac{\partial \boldsymbol{x}_{e_0}}{\partial \boldsymbol{c}_d} + \frac{\partial\tilde{\rho}_e}{\partial \boldsymbol{x}_{e_f}}\frac{\partial \boldsymbol{x}_{e_0}}{\partial \boldsymbol{c}_d}. \tag{4.68}$$

Derivatives with respect to edge ends can again be found in Norato et *al.* [83], and appear here as $\partial\tilde{\rho}_e/\partial\boldsymbol{x}_{e_0}$ and $\partial\tilde{\rho}_e/\partial\boldsymbol{x}_{f_0}$. From the definition of edge endpoints in Eqn. (4.10) and (4.11), derivatives $\partial\boldsymbol{x}_{e_0}/\partial\boldsymbol{c}$ and $\partial\boldsymbol{x}_{e_0}/\partial\boldsymbol{c}$ are the identity matrix. The density derivative of any element inside the component polygon is set to $\partial\rho_d/\partial\boldsymbol{c} = \boldsymbol{0}$. To justify making the sensitivity of interior elements zero, imagine perturbing the center of the component by a small amount. Most elements inside the polygon are still inside the polygon, so there is no change in the density. Some elements near the edges may have switched from being inside the polygon to in the edge bar, or vice versa. These elements will be near the bar center and should still have full density.

4.4.2 Geometric constraints

The interference constraint functions will be presented below along with sensitivity analysis. Any constraint involving a component is not enforced on the component boundary, but on a bounding circle with radius r^b centered at the component reference point, see Figs. 4.4a or 4.4b. The radius can be found by calculating the maximum vertex distance from the component reference point. The constraint between two components i and j, and its sensitivity, are shown below (see also Fig. 4.4a for an illustration of the constraint).

$$g_{\text{dd}}(\boldsymbol{x}) = (r_i^b + r_j^b)^2 - ||\boldsymbol{c}_j - \boldsymbol{c}_i||^2 \leq 0 \tag{4.69}$$

$$\frac{dg_{\text{dd}}}{d\boldsymbol{c}_i} = 2(\boldsymbol{c}_j - \boldsymbol{c}_i) \tag{4.70}$$

$$\frac{dg_{\text{dd}}}{d\boldsymbol{c}_j} = -2(\boldsymbol{c}_j - \boldsymbol{c}_i) \tag{4.71}$$

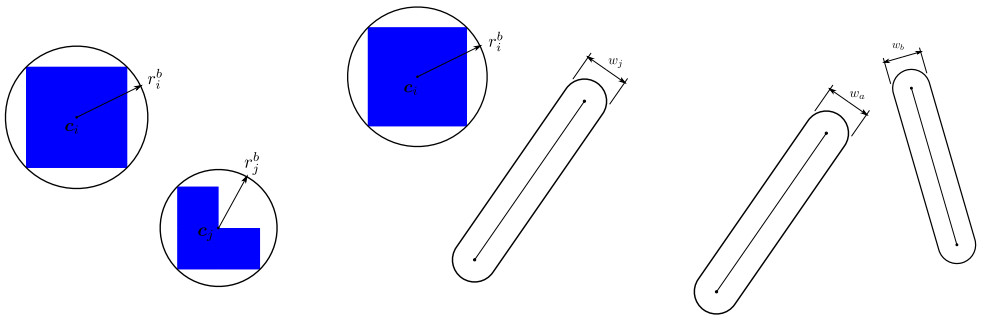
For the constraint between a component i and a segment of routing j, previous results from projecting a bar onto the mesh will be used. As an intermediate step in the projection, the distance between a line segment and a point was found. Here, the distance will be found between the line segment and a component reference point rather than a mesh element center. The constraint function is:

$$g_{\text{sd}} = \frac{w_j}{2} + r_i^b - d_{ij} \leq 0. \tag{4.72}$$

Sensitivities $dd_{ij}/d\boldsymbol{x}_{j_0}$ and $dd_{ij}/d\boldsymbol{x}_{j_f}$ are already known from previous results, as well as $dg_{\text{sd}}/dw_j = \frac{1}{2}$. The component reference point is a design variable, whereas the element centers were not design variables in previous results, so an additional sensitivity needs to be calculated:

$$\frac{dd_{ij}}{d\boldsymbol{c}_i} = \begin{cases} \frac{\boldsymbol{b}}{||\boldsymbol{b}||} & \boldsymbol{a}\cdot\boldsymbol{b} \leq 0 \\ \frac{1}{||\boldsymbol{g}||}\left(\boldsymbol{I} - \frac{1}{||\boldsymbol{a}||^2}(\boldsymbol{a}\otimes\boldsymbol{a})\right)\boldsymbol{g} & 0 < \boldsymbol{a}\cdot\boldsymbol{b} < \boldsymbol{a}\cdot\boldsymbol{a} \\ \frac{\boldsymbol{e}}{||\boldsymbol{e}||} & \boldsymbol{a}\cdot\boldsymbol{b} \geq \boldsymbol{a}\cdot\boldsymbol{a}. \end{cases} \tag{4.73}$$

A constraint to prevent interference between two routing segments requires finding the distance between two line segments. Reference [145] describes an algorithm for calculating the distance between two line segments. First,



(a) Constraint between two components

(b) Constraint between a component and routing segment

(c) Constraint between two routing segments

Figure 4.4: Geometric constraints

the two segments are extended into infinite lines, and the minimum distance is found. If the minimum occurs at a point on the line outside of the segment endpoints, then a series of cases must be tested to find the distance between endpoints and the other segment. MATLAB code to compute the minimum distance between two segments, and the derivative with respect to the segment endpoints, is included in the Appendix. The constraint to avoid interference between routing segments a and b with minimum distance d_{ab} between them is:

$$g_{\text{ss}} = \left(\frac{w_a}{2} + \frac{w_b}{2}\right)^2 - d_{\text{ab}}^2 \leq 0. \tag{4.74}$$

The squared distance is used to avoid undefined derivatives when the distance is zero. The sensitivity with respect to segment end points can be found in the MATLAB code in the Appendix. Sensitivity with respect to the bar widths are:

$$\frac{dg_{\text{ss}}}{dw_a} = \frac{dg_{\text{ss}}}{dw_b} = \left(\frac{w_a}{2} + \frac{w_b}{2}\right). \tag{4.75}$$

4.5 Stage 2 Illustration: UAV test platform

To illustrate stage 2, a notional power electronics cooling system for an unmanned aerial vehicle (UAV) as represented in Fig. 4.5 was optimized. This test platform has rich potential in multi-physics optimization for thermal, fluid, and mechanical packing. Both the system architecture and the geometric topology of the system are fixed during the optimization. The initial system layout is depicted in Fig. 4.6a, and corresponding component prop-

erties are given in Table 4.1. The system consists of two battery packs, an AC/DC converter, and a heat exchanger. The battery packs and AC/DC converter add heat to the system, and the heat exchanger removes heat. A fixed-location inlet and outlet for the fluid loop are placed on the left edge. Boundary conditions for the thermal problem are shown in Fig. 4.7. The top and bottom edges of the domain have convection boundary conditions with a convection coefficient of h = 35.4 W/(m^{2}K) and environment temperature of 0 ℃. The convection boundaries on the top and bottom edges can be related to the surfaces of the UAV exposed to external cooling by the environment during flight. The right edge has a fixed temperature of 100 ℃. For instance, this could be considered as the heat coming from the engine during flight, and the left edge is insulated. The maximum pipe diameter is allowed to be 0.03 m. There are two free points in each connection to allow for more complex interconnect routing paths. The optimization was solved using both head loss and bounding box objective functions.

The optimization formulation for minimizing the *system bounding box area* is given by:

$$
\begin{aligned}
\min_{\boldsymbol{x}} \quad & A(\cdot) = (\max(\bar{\mathrm{x}}) - \min(\bar{\mathrm{x}}))(\max(\bar{\mathrm{y}}) - \min(\bar{\mathrm{y}})) && \text{(4.76a)} \\
\text{subject to:} \quad & T_{\boldsymbol{d}_{i=1\ldots4}} \leq T_{\boldsymbol{d}_{max}} && \text{(Max. component temp. constraint)} \\
& T_{\boldsymbol{f}} \leq T_{\boldsymbol{f}_{max}} && \text{(Max. fluid temp. constraint)} \\
& H_l \leq H_{l_c} && \text{(Max. head loss constraint)} \\
& \boldsymbol{g}_{\text{geo}} \leq \boldsymbol{0} && \text{(Geometric constraints)} \\
\text{where:} \quad & \boldsymbol{K}(\boldsymbol{x})\boldsymbol{T} = \boldsymbol{P}(\boldsymbol{x}), && \text{(Physics-based model eqns.)}
\end{aligned}
$$

where $\bar{\mathrm{x}}$ and $\bar{\mathrm{y}}$ are the set of x and y coordinates of component reference points and bar segment end points. The critical component temperatures, T_c, for all components are shown in Table 4.1. Head loss constraints, H_{l_c}, will be used for the bounding box objective function. Here all geometric constraints from the generic optimization formulation have been lumped together into $\boldsymbol{g}_{\text{geo}}(\cdot)$. The function in Eqn. (4.76a) represents a rectangular bounding box containing all free components and interconnects that is aligned with the x and y axes. Similarly, the objective function for minimizing the *pressure head*

loss objective function is given by:

$$\begin{aligned}
\min_{\boldsymbol{x}} \quad & f(\cdot) = H_l && \text{(4.77a)}\\
\text{subject to:} \quad & T_{\boldsymbol{d}_{i=1\ldots4}} \leq T_{\boldsymbol{d}_{max}} && \text{(Max. component temp. constraint)}\\
& T_{\boldsymbol{f}} \leq T_{\boldsymbol{f}_{max}} && \text{(Max. fluid temp. constraint)}\\
& \boldsymbol{g}_{\text{geo}}(\cdot) \leq \boldsymbol{0} && \text{(Geometric constraints)}\\
\text{where:} \quad & \boldsymbol{K}(\boldsymbol{x})\boldsymbol{T} = \boldsymbol{P}(\boldsymbol{x}). && \text{(Physics-based model eqns.)}
\end{aligned}$$

Both optimization problems enforce component temperature constraints as described in the component properties table. In addition, the optimization for minimizing the bounding box had a head loss constraint of 1.5 m. Resulting layouts are shown in Figs. 4.6 (b,c), and some corresponding values from the final layouts are listed in Table 4.2. The pressure head loss objective improved by 37.96% from an initial head loss value of 1.452 m to 0.876 m and the bounding box area was reduced by 49.95% from an initial value of 0.6219m ^{2} to 0.3112 m^2. Objective function value convergence history plots and their corresponding first-order optimalities for each iteration are shown for head loss and bounding box functions in Figs.4.8 (a) and (b), respectively. It should be noted here that these values will change depending on different operating boundary conditions, fluid and component temperature constraints, component heat flux rates, and other relevant factors. In the head loss optimization, component 1 and 2 temperature constraints were active. In the bounding box optimization, the head loss constraint and the component 1 temperature constraint were active. As expected, using sharp angles at the elbows enables designs with smaller bounding boxes. The head loss objective layout has a higher total piping length, but lower head loss. This suggests that elbow geometry is the dominant contributor to head loss. To satisfy component temperature constraints, one of the routing interconnects touches the convection boundary. This conducts heat from the components through the routing to the boundary where it can be dissipated. The optimization finds a balance between smooth bends and reducing pipe length to reduce head loss in a way that is best for system performance.

component number	Description	Q_d (W/m^2)	T_{max} (°C)
1	Battery	5000	30
2	Battery	5000	30
3	AC/DC converter	1000	70
4	Heat exchanger	-2000	-

Table 4.1: Unmanned aerial vehicle's component properties.

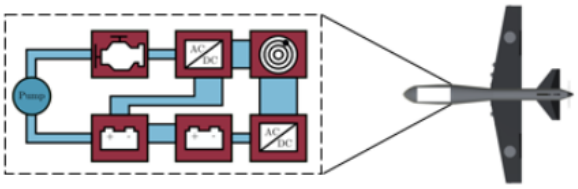


Figure 4.5: 2D schematic of an unmanned aerial vehicle (UAV).

Table 4.2: Power electronics cooling system optimization results. Optimal objective function values are highlighted in yellow.

objective	H_l (m)	bounding box (m^2)	T_1 (°C)	T_2 (°C)	T_3 (°C)
pressure head loss	0.876	0.722	29.8	30.0	27.3
bounding box	1.50	0.311	30.0	22.9	17.2

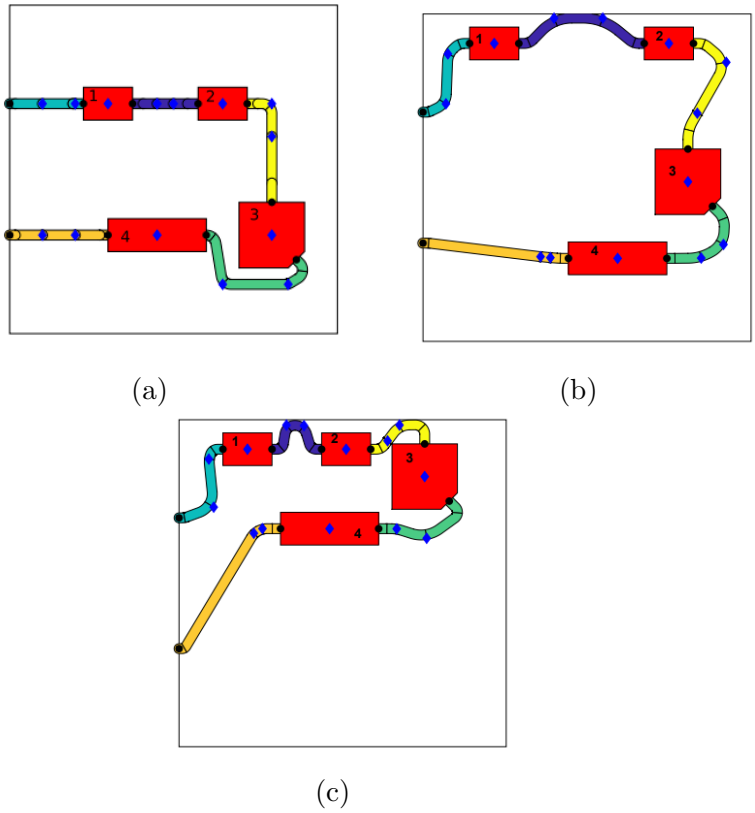


Figure 4.6: Results of hybrid UAV power electronics cooling system illustrative study: (a) Initial layout; (b) final optimal layout for head loss objective function; (c) final optimal layout for bounding box objective function.

4.5.1 Case study: Underhood packaging of an automotive fuel cell (AFC) system

Stage 2 of the two-stage design framework is also demonstrated on an industry-relevant automotive fuel cell (AFC) application. Underhood spacing of the fuel cell system, front end electric drive system and other thermal management components presents a complex problem for minimizing cost and weight while delivering the required vehicle capability and performance. Of interest is optimal packing and routing (PR) for thermal management with the goals of minimizing hose lengths, coolant volumes and the total number of connections. The schematic of the 2D AFC layout is shown in Fig. 4.10. However, some of the components of the AFC system have been grouped together based on functionality and proximity into subsystems for design simplification.

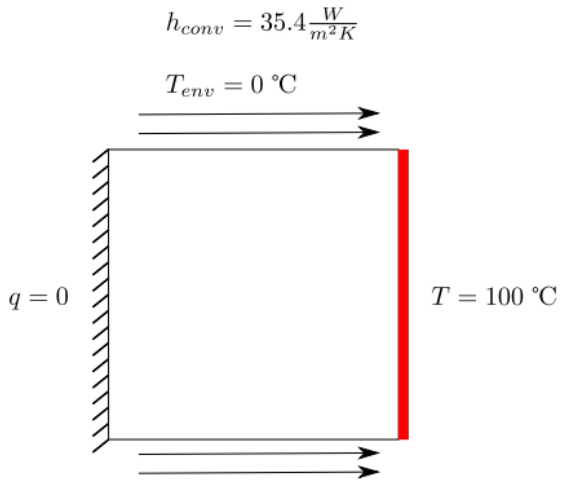


Figure 4.7: Boundary conditions for the thermal finite element analysis.

Two problem cases are solved: 1) with four subsystems, and 2) with six subsystems, respectively. The goal is to minimize the bounding box area and the total AFC system interconnection length of the AFC layout subject to 2D interference constraints. The results for both the four and the six subsystem cases are shown in Figs. 4.11, and 4.12 respectively. The decrease in bounding box areas for the four and six subsystem layouts are 83.83% and 79.26% respectively. Similarly, the decrease in the total interconnection lengths of the four and six subsystem AFC layouts are 75.8%, and 47.9% respectively. The main observations from the results are as follows: 1) an increase in the number of components (subsystems) from four to six leads to a less significant percentage decrease in total bounding box area and total interconnection length, and 2) with an increase in layout complexity (number of components and interconnects), there is an increase in final packing density. It must be noted that this case study only shows a simple illustration of stage-2 of the framework using an AFC system. The original AFC layout was significantly simplified for illustration purposes and the comprehensive layout solution will be published in future work.

4.6 Case Studies

4.6.1 Case Study 1: Comparison of layout generation methods

In this case study the FDLM, SPA, and UST methods are compared with each other in terms of how their resulting layouts perform as start points in stage 2. A multi-start approach is used where all feasible layouts from stage 1 are used as initial designs for stage 2 optimization problems with bounding box area and pressure head loss objective functions separately to see how it scales over an increasing number of components n. For the case studies, the class of topologies are restricted to only n-component systems with each component having two ports each (say, nC-2P). The FDLM methods begin with both components and routing overlapped. The SPA and UST methods have non-overlapping components initially.

The problem specifications are as follows: The multi-component interconnected system has n equally-sized square components, with each component dissipating at least 3,000 W/m^2 of heat, and with each additional component individually dissipating 50 W/m^2 more than its predecessor. For example, a system with 10 components will have the 10th component generating $3,000 + (10 - 1)50 = 3,450$ W/m^2 of heat. The components are connected in a single loop. Each connection has one free intersection point. A short fixed pipe segment is attached to each port. This allows constraints between components and all free pipe segments to be enforced. The same boundary conditions shown in the UAV illustration study are enforced here. The maximum temperature constraint on all components is set to 30 ℃. For the bounding box objective function, the head loss constraint is set to 5 meters for all cases of $n \in \{3, 4, \ldots, 9, 10\}$. The optimization problem terminates when one of the following convergence criteria is satisfied: 1) objective function step is within a prescribed tolerance, i.e., $|f_{n+1}(\boldsymbol{x}) - f_n(\boldsymbol{x})| \leq \delta_f$, or 2) design variable vector change magnitude is within a prescribed tolerance i.e. $\|\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\| \leq \delta_{\boldsymbol{x}}$. Here, δ_f and $\delta_{\boldsymbol{x}}$ are set to be 10^{-7} and 10^{-5}, respectively. All computations in this chapter were performed using a workstation with an Intel Xeon E5-2660 CPU @ 2.00 GHz, 64 GB DDR4-2400 RAM, Windows 10 64-bit, and Matlab 2019b.

Figure 4.13 shows the final optimal bounding box objective function values for all the initial feasible layouts. Each layout is represented on the plot using different colored circles. The direct UST is the best performer and the average values of the indirect USTs are worse because of the longer interconnects that reduce the potential to shrink. The objective function values for the indirectly-connected USTs and the SPA layouts for a given nC-2P system are averaged to keep it consistent with the other solutions. Figure 4.14 shows the initial feasible and final optimal layouts of a 6-component two-port system (6C-2P), corresponding to design points (a)-(d) in Fig. 4.13. Similarly, Fig. 4.15 shows the final pressure head loss objective function values for all the initial feasible layouts. Figure 4.16 shows the initial feasible and final design layouts of corresponding to points (a)-(d) in Fig. 4.15. UST-Indirect is the best performer for the pressure head loss objective function because the longer interconnects it produces can help achieve smoother pipe bends. UST-Direct does not significantly improve the head loss objective function as it contains shorter interconnects where a greater possibility of having sharper pipe bends exists. It should be noted that, for illustration purposes only, one representative indirect-UST layout and one representative SPA layout are shown in Figs. 4.14 and 4.16, although the plots in Figs. 4.13 and 4.15 show averaged times for all indirect-USTs and SPAs, respectively. In addition, the initial SPA layouts used in this case study are generated with some obstacles to perform routing so that the effect of interconnect complexity can be assessed during optimization. Those obstacles or restricted regions are removed during optimization to make a fair comparison with other layout generation methods. Section 4.7 describes the important inferences from this case study in more detail. To achieve convergence given the prescribed tolerances, the average computation times for solving the bounding box minimization and the pressure head loss minimization problems were 2,345.5 s and 2,743.2 s, respectively.

Here it is also observed that the physics-based component temperature constraints for all feasible layouts were either active (equal to the allowable maximum temperature which is 30° C), or well within the feasible range. For example, Table 4.3 shows the final component temperatures of the six components from each of the four feasible layouts shown in Fig. 4.14. In the final optimized layouts, components near the boundaries had temperatures

Table 4.3: Final component temperatures of optimized layouts shown in Fig. 4.14 are satisfying the physics-based temperature constraints.

Final layouts	T_1 (℃)	T_2 (℃)	T_3 (℃)	T_4 (℃)	T_5 (℃)	T_6 (℃)
(a) Direct-UST	22.4	18.6	29.8	28.9	30.0	29.9
(b) FDLM	23.2	20.7	27.7	22.6	23.4.0	29.3
(c) SPA	27	19.7	28.5	21.6	23.7	30.0
(d) Indirect-UST	29.7	28.9	26.4	23.0	24.2	29.0

well below 30° C, but those far from the boundaries were nearly active (or at least hotter) due to less convective cooling from ambient conditions.

4.6.2 Case Study 2: Bi-objective optimization problem

Multi-objective optimization studies are performed to analyze the trade-offs that exist between two or more conflicting performance metrics. The epsilon-constraint method tackles multi-objective problems by solving a series of single objective subproblems, where all but one objective are transformed into constraints. A Pareto front can be obtained efficiently for bi-objective problems. In this case study, the two competing objectives are bounding box area (f_1) and pressure head loss (f_2) functions. The head loss objective function f_2 is treated here as a constraint, varying within the range $2.2 \leq \varepsilon_i \leq 6.2$ m over 40 equally-spaced intervals, and solve the corresponding bounding box minimization problems sequentially. The optimization is performed on a specific 5C-2P system, and its spatially feasible initial layout is shown in Fig. 4.17. The same physics-based boundary conditions are incorporated as earlier, component sizes, heat dissipation rates, and temperature constraints as specified for case study in Sec. 4.6.1. The ε-constraint optimization for-

mulation is as follows:

$$
\begin{aligned}
\min_{\boldsymbol{x}} \quad & f(\cdot) = f_1(x) && \text{(4.78a)}\\
\text{subject to:} \quad & T_{\boldsymbol{d}_{i=1\ldots4}} \leq T_{\boldsymbol{d}_{max}} && \text{(Max. component temp. constraint)}\\
& f_2(x) \leq \varepsilon_j, j = 1, ..., 40 && \textbf{(Head loss obj. as } \varepsilon\textbf{-constraint)}\\
& T_{\boldsymbol{f}} \leq T_{\boldsymbol{f}_{max}} && \text{(Max. fluid temp. constraint)}\\
& \boldsymbol{g}_{\text{geo}}(\cdot) \leq \mathbf{0} && \text{(Geometric constraints)}\\
\text{where:} \quad & \boldsymbol{K}(\boldsymbol{x})\boldsymbol{T} = \boldsymbol{P}(\boldsymbol{x}), && \text{(Physics-based model eqns.)}\\
& 2.2 \leq \varepsilon_j \leq 6.2. && \text{(range of } \varepsilon\text{-constraint values)}
\end{aligned}
$$

The Pareto optimal solutions plotted in the objective function space are shown in Fig. 4.18. As the overall bi-objective design task is to minimize both the bounding box area and the pressure drop across interconnects, the points in the space close to the bottom-left corner (toward the origin) are desired.

Optimal solutions have a range of pressure drop values from 2.2 to 6.2 m, and it results in a range of bounding box values from 0.015 to 0.072 m^2. The labels (a) through (d) that identify specific marked points in Fig. 4.18 correspond to the designs given in Fig. 4.17(a)-(d). These representative solutions (a) through (d) were chosen subjectively such that they are evenly distributed in the objective function space. Points (a) and (d) are the anchor points of the Pareto front. This case study indicates how sharp pipe bends cause more head loss across the system.

4.6.3 Case Study 3: Multi-loop optimization example

In this study, how the proposed framework also supports multi-loop optimization problems and can be used for more complex interconnected component layouts is demonstrated. A two-loop system optimization result is shown in Fig. 4.19, illustrating the efficacy of this framework. The same boundary conditions are maintained as earlier, component sizes, heat dissipation rates, and temperature constraints as defined for the case study in Sec. 4.6.1. Optimal layouts are obtained for the bounding box objective function (as shown in Fig 4.19(a), improving by 66.03% from 0.1443 m^2 to 0.04901 m^2), and for

the pressure head loss objective function (as shown in Fig 4.19(b), improving by 40.01% from 5.98 m to 3.578 m). The two optimization studies required 1,855.4 s and 2,234.7 s of computation time, respectively, using the same hardware defined above.

4.7 Discussion

The primary findings from the above case studies (especially from Secs. 4.6.1 and 4.6.2) are as follows:

1. The directly-connected topologies obtained from the USTs have the best performance for the bounding box objective compared to all other kinds of topologies. This is because they can shrink very easily since there is no complex routing involved in such layouts. The indirectly-connected GTs on average did not perform well for the bounding box objective function because their interconnects have longer lengths compared to the direct GTs. This restricts volume reduction significantly because there is a greater chance of the interconnects intercepting the components they are circulating.
2. The feasible layouts from the force-directed layout method (FDLM) performed well for the bounding box objective function after the direct GTs. The disadvantage of the FDLM is that it fails for complex interconnects. This turns out to be a hidden advantage in that the feasible layouts attained using FDLM are free from interconnect untangling and component overlapping. They are produced by a spring layout method that spread the system components uniformly until all the nodes reach minimal energy. This property may not be advantageous in other problem contexts.
3. The results for the SPAs with simpler topologies were good but worsened for more complex networks. This might be because larger-scale systems have interconnects that are very long, not allowing the components to pack closely.
4. The initial feasible layouts that had longer interconnects tend to have lower pressure (head) losses when pressure head was used as the objective. These layouts could better avoid sharp bends (the dominant factor in head loss), expanding within the bounding box as some of the

interconnects may not be constrained between the components.

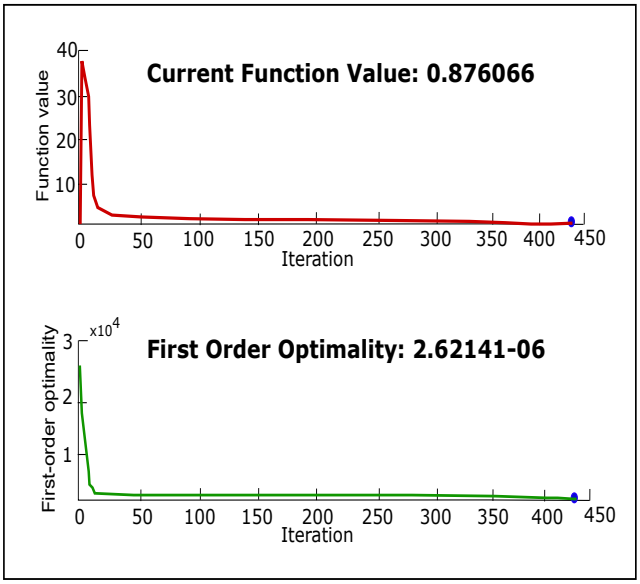
5. The direct GTs have higher head loss results. This is because the interconnects are constrained within the region between the components, and sharp bends might occur while avoiding intersections with the components.
6. The FDLM performs reasonably well for both objective functions. The interconnect untangling is done systematically, uniformly balancing all nodes in the FDLM. The feasible layouts that are produced are favorable designs and tend to be flexible in supporting both smooth bends and volume reduction.

Finally, it is important to note that the above observations only present a general notion of how the 2D layout generation methods perform in terms of the start points they provide. These initial results indicate that the two-stage design framework helps obtain a variety of solutions, each having the potential to represent appropriate applications. In the future, work from this chapter can be extended to investigate a more comprehensive set of system types that can provide deeper insights.

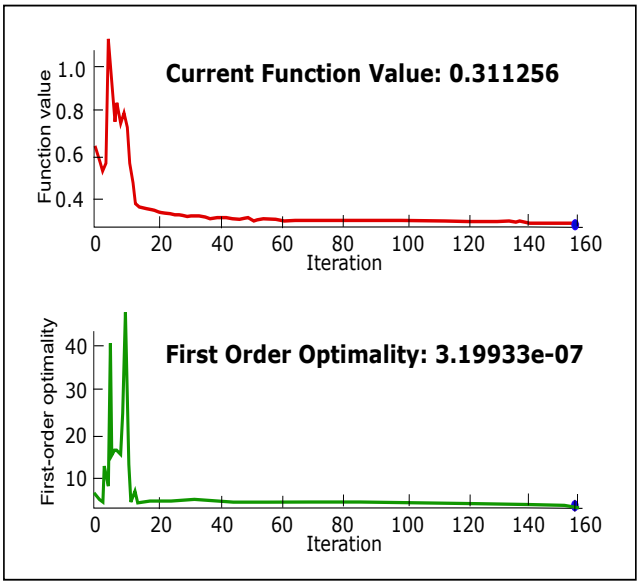
4.8 Summary

In this chapter, stage 2 of the two-stage design automation approach was demonstrated by utilizing the the set of feasible topological layouts of a fluid-thermal system attained from stage 1, and then subsequently optimizing each layout via a gradient-based design optimization procedure that accounts for the physics-based performance of the system. Three distinct layout generation methods for generating feasible geometric layouts in stage 1 were described and demonstrated. A stage 2 physics-based optimization method was described. A notional power electronic cooling system for an unmanned aerial vehicle (UAV) was optimized in an illustrative study. The example demonstrates our technique for simultaneous optimization of component placement and interconnect routing, given a GT and initial layout from stage 1. Starting from the initial condition, both the components and interconnects move through the domain in search of a locally optimal solution. The three layout generation methods have been compared over different system performance metrics. The UST-Direct method was the best

for the minimum bounding box objective because the direct interconnects between the components have relatively short lengths and reduced complexity, thus supporting enhanced volume reduction compared to other methods. The SPAs and UST-Indirect methods generate layouts that, in general, have complex routing paths. Longer interconnect routing helps in minimizing the head loss in the pipes because lengthy paths can help reduce the occurrence of very sharp pipe bends in the final optimized layout. A bi-objective optimization problem with bounding box and head loss objective functions was solved in the second case study using the ε-constraint method to attain Pareto-optimal design solutions. The final case study demonstrated the solution of a multi-loop optimization problem using the proposed framework. Future work items for Chapter4 are mentioned in Chapter 6.



(a)



(b)

Figure 4.8: Objective function convergence history and first order optimality plots of (a) a pressure head loss function; and (b) bounding box objective function.

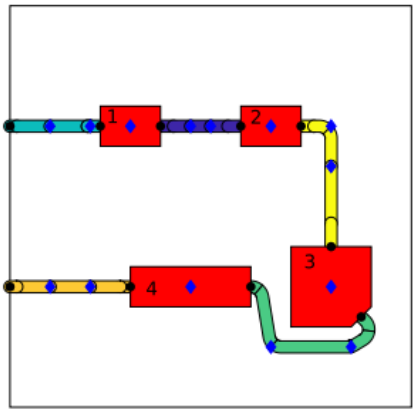


Figure 4.9: Initial layout of a UAV power electronics cooling system.

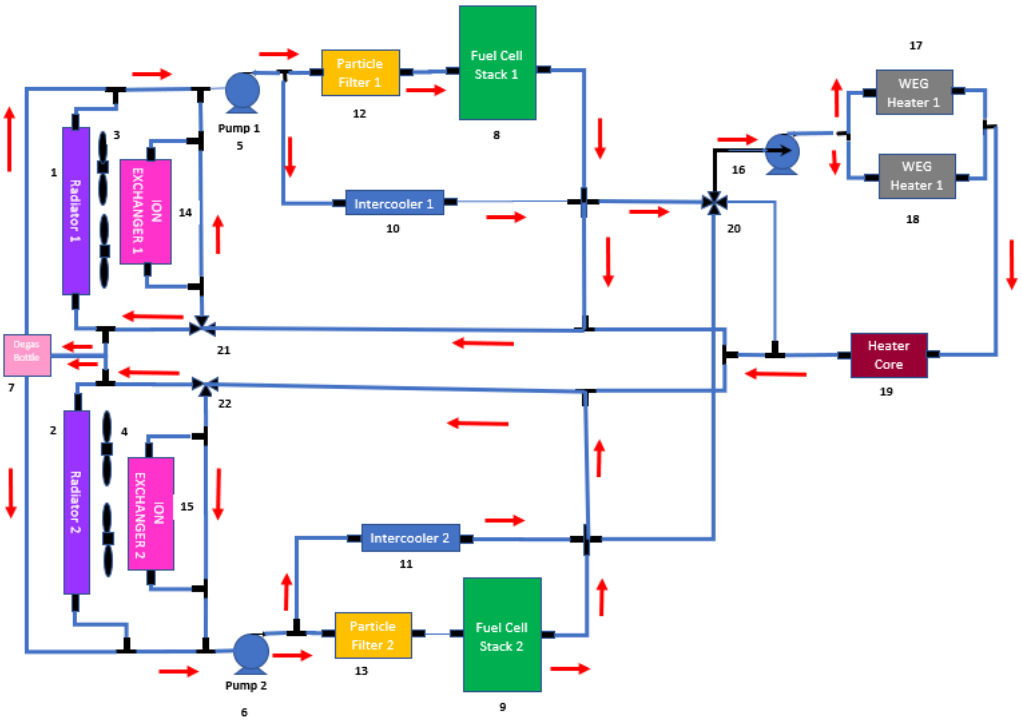


Figure 4.10: The 2D schematic diagram of the automotive fuel cell (AFC) system with all its components and their interconnections.

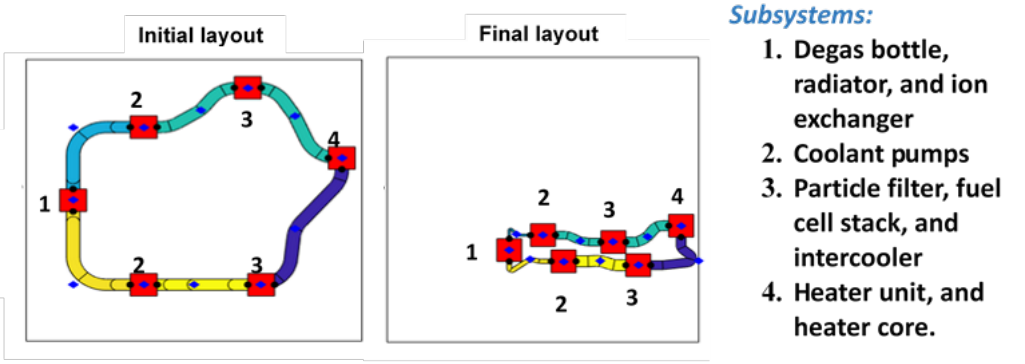


Figure 4.11: Initial and final optimized AFC layout with a four subsystems' layout. The corresponding subsystems have been labelled.

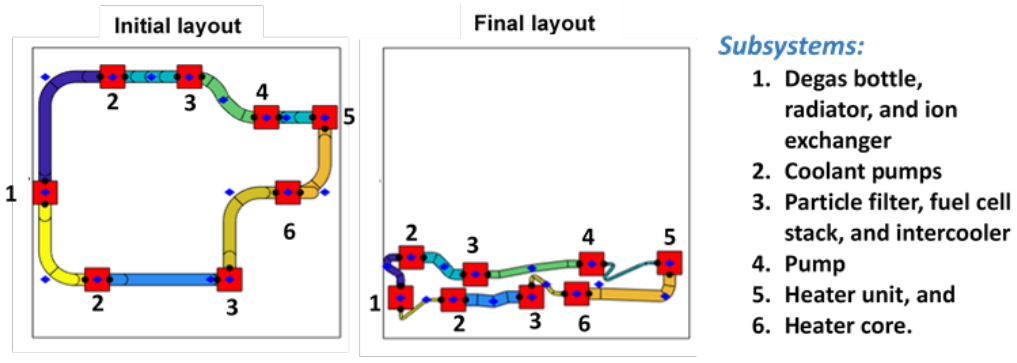


Figure 4.12: Initial and final optimized AFC layout with a six subsystems' layout. The corresponding subsystems have been labelled.

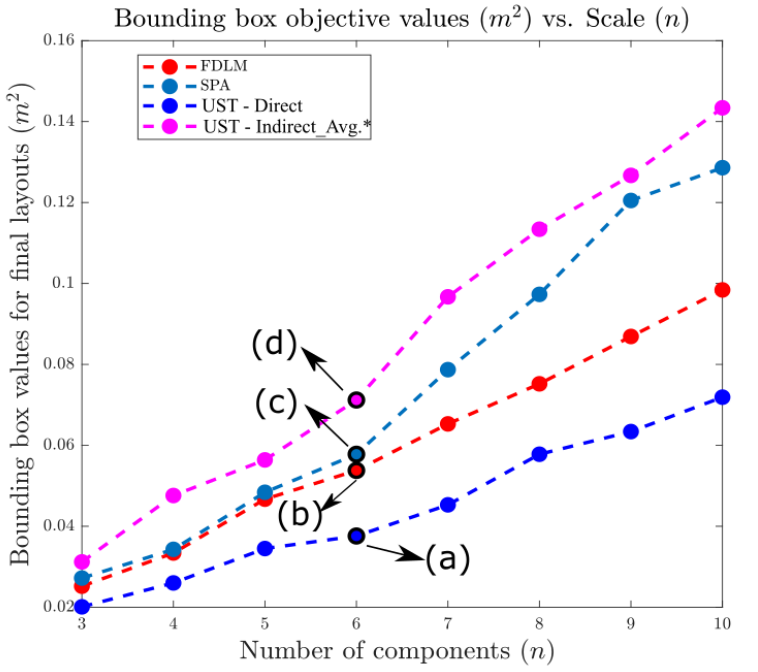


Figure 4.13: Bounding box objective function values for nC-2P systems.

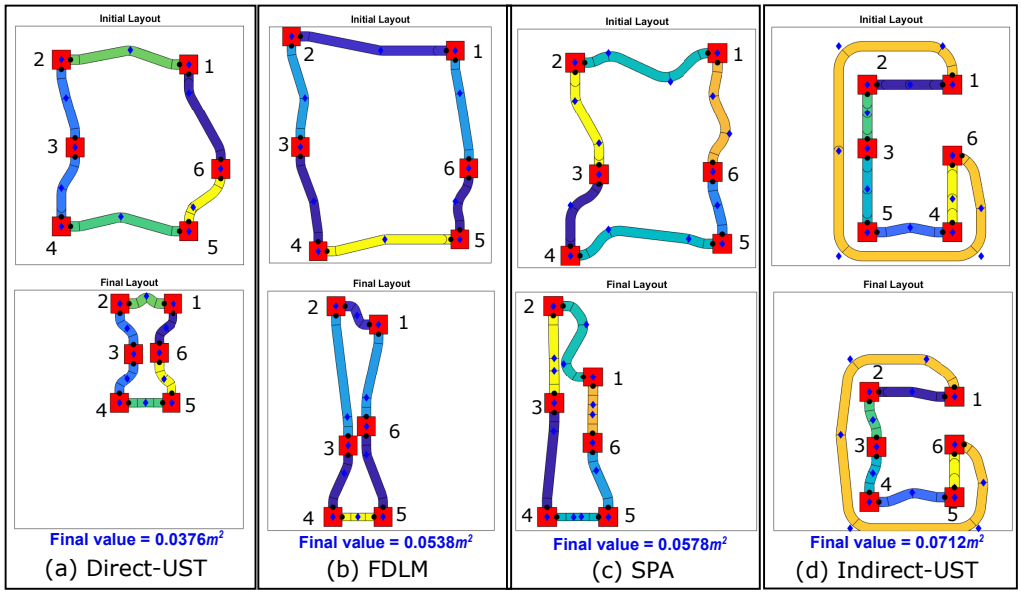


Figure 4.14: Corresponding initial and final optimal 6C-2P layouts of designs (a)-(d) indicated in Fig. 4.13 for bounding box objective function.

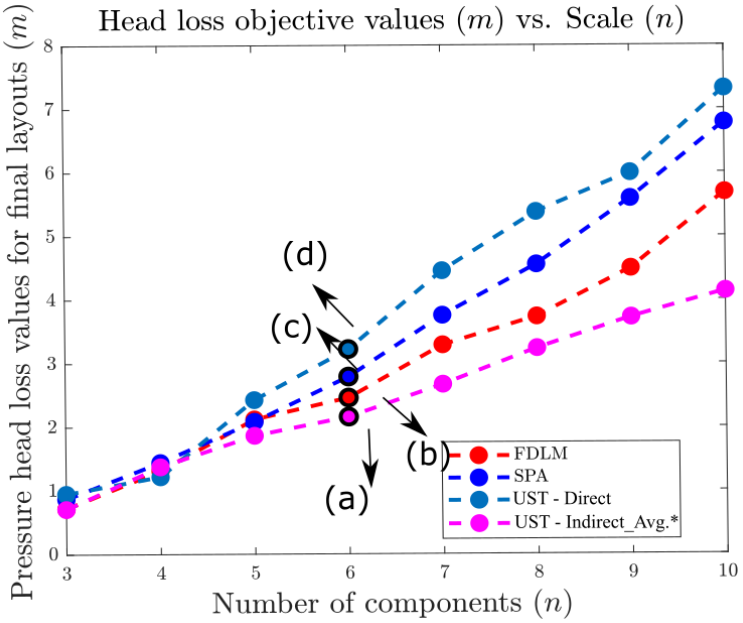


Figure 4.15: Pressure (head loss) objective function values for nC-2P systems.

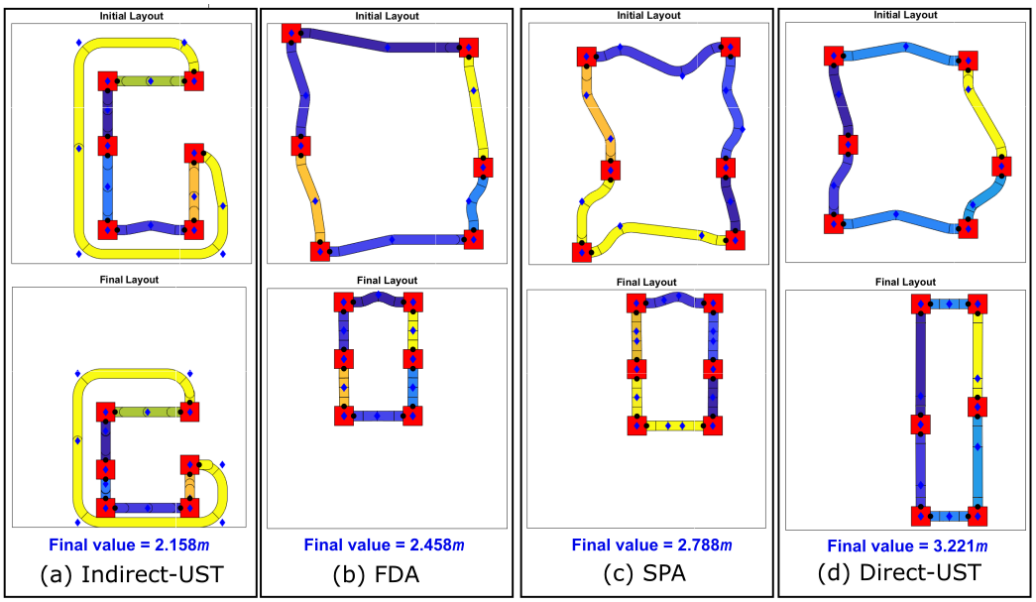


Figure 4.16: Corresponding initial and final optimal 6C-2P layouts of designs (a)-(d) indicated in Fig. 4.15 for pressure head loss objective function.

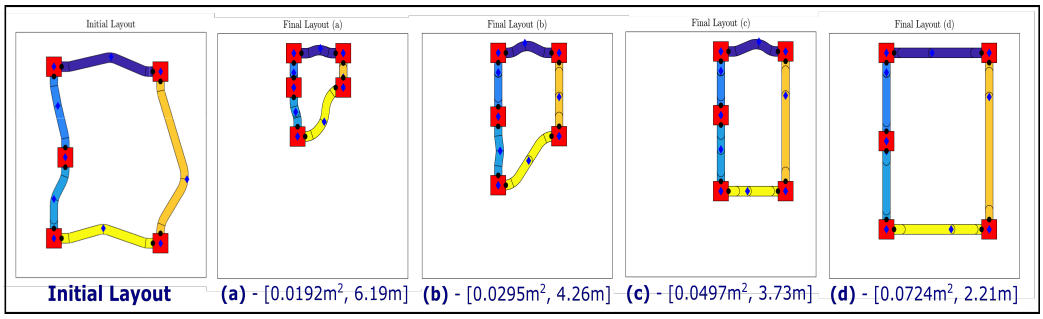


Figure 4.17: Initial and final optimal layouts for a 5C-2P interconnected multi-component system for four selected designs from the Pareto set.

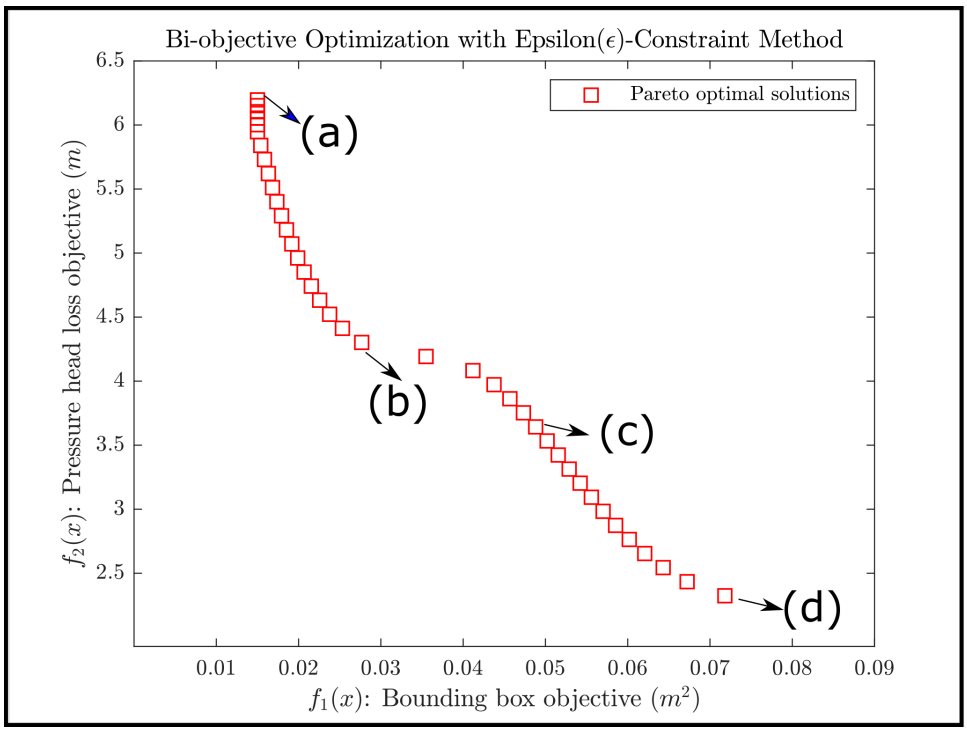


Figure 4.18: Bi-objective optimization Pareto plot for a 5C-2P interconnected multi-component system.

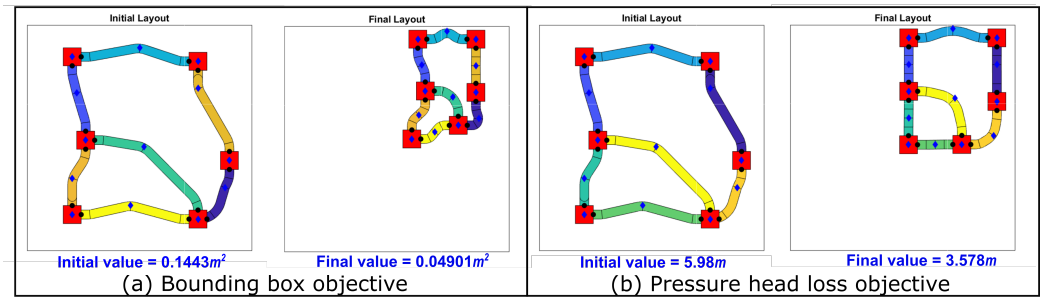


Figure 4.19: Initial and final optimal layouts for a two-loop interconnected multi-component system.

CHAPTER 5

DESIGN REPRESENTATION AND SYSTEMATIC ENUMERATION OF 3D SYSTEM SPATIAL TOPOLOGIES

5.1 Introduction

In Chapters 3-4, the two stages of the SPI2 design framework were discussed in detail, relevant case studies were demonstrated and practical implementation guidelines were outlined, especially with the focus on the 2D SPI2 problem. This chapter concentrates on stage 1 of the framework for the 3D SPI2 problem. Systematic enumeration and identification of unique 3D spatial topologies of complex engineering systems such as automotive cooling layouts, hybrid-electric power trains, and aero-engines are essential to search their exhaustive design spaces to identify spatial topologies that can satisfy challenging system requirements. However, efficient navigation through discrete 3D spatial topology options is a very challenging problem due to its combinatorial nature and can quickly exceed human cognitive abilities at even moderate complexity levels. This chapter[1] presents a new, efficient, and generic design framework that utilizes mathematical spatial graph theory to represent, enumerate, and identify distinctive 3D topological classes for an engineering system, given its system architecture (SA) – its components and interconnections. Spatial graph diagrams (SGDs) are generated for a given SA from zero to a specified maximum crossing number. Corresponding Yamada polynomials for all the planar SGDs are then generated. SGDs are categorized into topological classes, each of which shares a unique Yamada polynomial. Finally, for each topological class, one 3D geometric model is generated for an SGD with the fewest interconnect crossings. Several case studies are shown to illustrate the different features of our proposed ST enumeration framework. At the end of this chapter, design guidelines are also provided for practicing engineers to aid the utilization of this framework for

[1]Some elements of this chapter are published in Ref. [146]

application to different types of real-world problems.
Systems engineering [147, 148] often involves choosing the most suitable candidate among many alternative design solutions to meet specific performance criteria using techniques such as comparative design analysis and optimization [149, 150]. In most cases, the component technologies and the component-to-component connectivity (system architecture) remains fixed between different feasible designs to preserve the functionality of the system. Figure 5.1 shows 3D systems A and B1 having two different system architectures because interconnect IC1 is connected between components $\{1, 2\}$ in A but between $\{1, 3\}$ in B1. In other words, component-to-component connectivity is different in A and B1 respectively. Systems B1 and B2 have the same system architecture as all the component-to-component interconnections are the same. However, B1 and B2 have different spatial topologies because the interconnect IC2 between the component 1 and 2 is topologically different (please see the crossing patterns in Fig. 5.1). As both the ends of the interconnect IC2 are fixed, it cannot be continuously morphed between B1 and B2. Hence, B1 and B2 are topologically different systems. The scope of this chapter deals with enumerating such unique spatial topologies for each 3D system architecture within a design problem. The example shown in Fig. 5.1 is kept simple for illustration purposes, but the framework can be used to generate STs for more complex architectures and larger interconnected 3D systems with multiple crossings.

Any systematic design process [151, 152] involves four key tasks: *representation*, *generation*, *evaluation*, and *design guidance*. Representation refers to the task of describing a system using a generic model that captures the functionality of the various system elements. Depending on the design analysis tools and application requirements, design representations can be mathematical, graphical, physics-based, or conceptual [153, 154]. The generation phase involves creating feasible design alternatives using the representation based on design rules. Evaluation is the process of measuring the design quality in terms of the performance criteria. Finally, design guidance is providing feedback for the generation task based on the evaluation output to find better alternatives in the design space. The generation, evaluation, and design guidance tasks are usually performed in an automated loop that converges finally to a design solution. However, what actually enables high design accuracy,

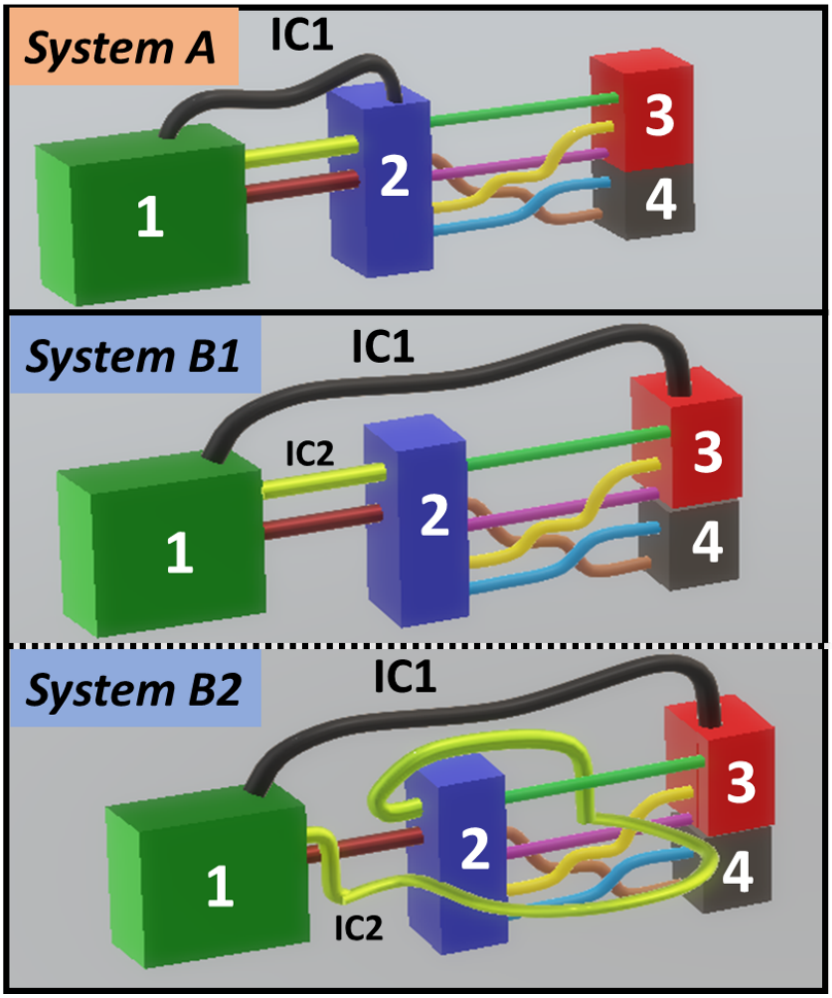


Figure 5.1: A simple illustration of 3D system architecture and spatial topology concepts. 3D systems A and B1 have different system architectures. Systems B1 and B2 have the same system architecture but are two distinct spatial topologies.

comprehensive search space navigation, and computational efficiency of the output from the last three tasks is the design representation selected in the first task. Hence, a design representation for 3D interconnected systems that captures the relevant problem attributes and aligns well with a generation computation is critical to navigate through the discrete 3D spatial topology options efficiently.

The design elements of a three-dimensional interconnected system are its components, their 3D spatial locations and orientations, their port valencies, interconnections, and the crossings of their interconnections. Unlike 2D system layout enumeration performed in the development of very large scale integrated circuits (VLSI), the 3D spatial layout enumeration problem is fundamentally different and even more challenging. More specifically, even for a given system architecture, options exist for how the interconnects are routed relative to one another and to the components (e.g., say, should duct A go over cable B or under, or should a pipe be routed through the hole in the casing, or around the edge of the casing, etc.). These are topologically discrete design differences. To cater to that need, in this chapter, a mathematical design representation has been identified to describe 3D interconnected spatial layouts as *spatial graph diagrams (SGDs)*.

5.1.1 Chapter Objectives and Contributions

Complex engineering systems such as autonomous aerial vehicles [155], electric power trains [107, 156], aero-jet engines [157], or vehicle thermal management and cooling systems [103, 158] have different kinds of components connected together either through wires, ducts, or pipes entangles with one another in a tightly packed three-dimensional volume. Design alternatives with distinct 3D spatial topologies may exist, with different values of metrics such as efficiency, spatial packaging density, maintenance costs, and design complexity. Current practice for exploring different 3D system spatial topologies relies largely upon human expertise, design rules, modification of existing designs, and manual adjustments. This approach precludes use of spatial topologies for practical application to typical complex systems, and suggests that automated methods are needed to apply spatial topologies for

typical systems design. In addition, the realization of optimal functionality or performance is not guaranteed with current practice due to incomplete design space coverage. Therefore, the main objective of this chapter is to develop an automated and systematic enumeration framework to both represent 3D engineering systems using spatial graph diagrams (SGDs) and efficiently generate distinct spatial topologies (STs) for a given system architecture using a rigorous mathematical approach. Advantages of using a spatial graph representation are: 1) simplicity, while capturing necessary system elements and features, 2) easy to visualize, 3) flexibility to add additional geometric features, 4) distinct topologies can be detected using polynomial invariants, 5) scalable or even decomposable into a set of smaller graphs, 6) supports automated 3D model generation, and 7) features such as node locations, edge diameters, edge trajectory shape functions, port locations, crossing information, etc., can be parameterized for performing numerical optimization. For example, items 6 and 7 can be very useful for using different 3D models as initial start points for multi-physics packing and routing optimization.

The major contributions of the proposed design framework include:

1. A new way to represent 3D engineering systems using spatial graph theory. This representation supports description of components as nodes, interconnects as edges, multiple crossings, and variable component valency.
2. Combinatorial enumeration of all SGDs for a given system architecture up to a specified level of complexity.
3. Efficient and systematic identification of unique SGDs from the exhaustively enumerated SGD set using Yamada polynomial invariants. This serves as a foundation to remove redundant topologies and explore the 3D spatial topology design space more thoroughly.
4. Topological equivalence between spatial graphs is tested here using Yamada polynomials rather than comparing diagrams directly.
5. Several case studies that illustrate the efficacy of this design automation framework.
6. Practical guidelines to help system design engineers apply this proposed framework to different kinds of problems.

The remainder of this chapter is organized as follows. The terminology and notation of the proposed spatial graph representation are discussed in detail

in Sec. 5.2. Section 5.3 describes the characteristics of Yamada polynomials and how they are evaluated for an individual SGD. Section 5.4 demonstrates using the SGD enumeration design framework to produce unique spatial graph layouts and categorize them into topological equivalence classes. Section 5.5 presents several practical case studies based on the proposed framework. The results are discussed in Sec. 5.6. Finally, the chapter summary is presented in Sec. 5.7.

5.2 Spatial Graphs

The study of graphs in 3-space has been mathematically formalized using *spatial graphs* [159–161], which is now described here. Suppose G is a graph, that is, a set of vertices and a set of edges, where an edge is just a pair of vertices. (Edges are undirected and multiple edges between the same pair of vertices are allowed.) A *spatial embedding* of a graph G is a set of points (nodes) in $\mathbf{R}^3$ corresponding to the vertices of G, and a set of smooth arcs (links) corresponding to the edges of G that join appropriate pairs of vertices; here, each arc meets the vertices only at its two endpoints, and it intersects other arcs only at these vertices. Collectively, these points and arcs form a *spatial graph* with underlying (abstract) graph G. More formally, the spatial embedding is a function $f : G \to \mathbf{R}^3$, whose image $\tilde{G} := f(G)$ is the spatial graph. See Figure 5.2(a) for a sample spatial graph. The natural topological notion of equivalence for spatial graphs is *isotopy*, when two spatial graphs $\tilde{G}_1$ and $\tilde{G}_2$ can be continuously deformed from one to the other without any arc passing through another arc or itself.

Spatial graphs are a natural extension of knot theory, which is the study of circles embedded in $\mathbf{R}^3$, since one can put vertices on a knot to make it into a spatial graph. While the study of knot theory has its origin in the physics of the late 19th century [88], spatial graph theory has its roots in chemistry [89, 90] and is different from graph theory because graph theory studies abstract graphs while spatial graph theory studies embeddings of graphs in $\mathbf{R}^3$ or even in other 3-manifolds [91–93]. This theory was used in polymer stereochemistry [89, 94] and molecular biology (e.g., protein folding) to distinguish different topological isomers. A folded protein can be thought of as

a spatial graph where residues are the nodes and edges connect the residues in close proximity.

If a spatial graph is projected onto a plane, then some arcs (edges) may appear to cross in the projection. If information about which arc is on top at the apparent crossings is omitted, the projection is called a *shadow* of the spatial graph, as shown in Fig. 5.2(b). If the information about which arc is on top at each apparent crossing is tracked, the projection or planar representation is called a *diagram* of the spatial graph, as shown in Fig. 5.2(c). In other words, diagrams are the images of embedded graphs under a projection $\mathbf{R}^3 \to \mathbf{R}^2$ whose singularities are a finite number of crossings of edges equipped with over-under crossing information. Hence, many different spatial diagrams of a spatial graph G may have the same shadow. Thus, a family of spatial graph diagrams (SGDs) can be produced by assigning all possible permutations of overstrand or understrand information for a given shadow.

5.2.1 Reidemeister moves

Given an abstract graph G, one can use the spatial graph diagrams above to begin enumerating spatial embeddings of G. The challenge is then to determine which of these SGDs actually describe isotopic spatial embeddings (i.e., are topologically equivalent), so that later steps in the design process consider each topological possibility only once. Fortunately, it has been shown that two diagrams represent isotopic embeddings if and only if they are related by a finite sequence of fundamental *Reidemeister moves (*$R0$ *to* $R6$*)* [162–164] as shown in Fig. 5.3. Figure 5.4 shows a simple illustration of three diagrams where SGDs A and B are *topologically equivalent* under the first Reidemeister move $R1$ whereas C is not equivalent to either A or B as its edges cannot be continuously deformed using the Reidemeister moves to attain A or B, so they represent *topologically distinct* spatial graphs.

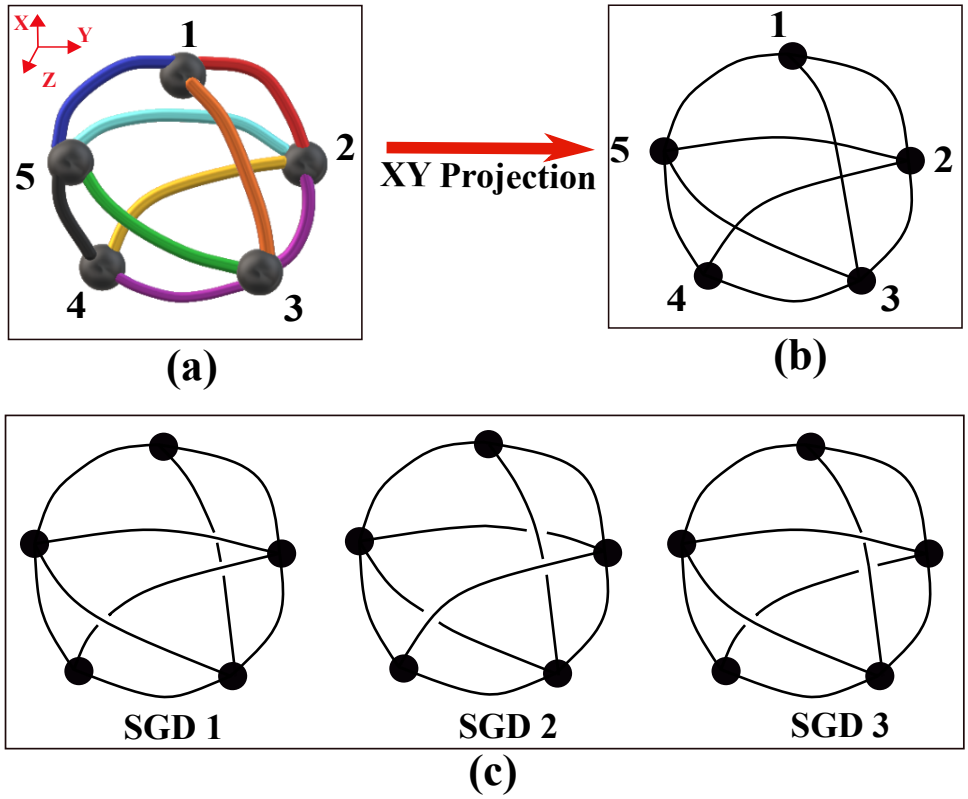


Figure 5.2: Figure **(a)** represents a five component interconnected 3D system; **(b)** is the shadow of the 3D diagram; and **(c)** shows three distinct spatial graph diagrams (SGD1, SGD2, and SGD3)

5.2.2 Flat vertex graphs and ribbon graphs

The topological formulation of spatial graphs is quite idealized in that each vertex has no local structure and the edges are infinitely thin. An additional, but still purely topological, structure can be imposed by considering *flat vertex graphs* and *ribbon graphs*, which may be more suitable for certain design applications. A flat vertex graph is a spatial graph where the vertices correspond to flat disks in $\mathbf{R}^3$ as shown in Fig. 5.5. In particular, this gives the edges coming in to each disk a cyclic order. A flat vertex graph can also be encoded by a SGD, with the convention that each disk is rotated parallel to the projection plane before projecting. Two SGDs represent isotopic flat vertex graphs if and only if they differ by a series of Reidemeister moves $R0$ to $R5$; here $R6$ is no longer allowed since it would change the order of the edges coming into the vertex disk.

A ribbon graph is a spatial graph whose vertices have become flat disks and whose edges have become thin bands, depicted in Fig. 5.6. These too can be encoded as SGDs by using the blackboard framing convention (a way to view a knot diagram on a plane) illustrated in Fig. 5.6. This framing is obtained by converting each component to a ribbon lying flat on the plane. Two SGDs represent isotopic ribbon graphs if and only if they differ by a sequence of Reidemeister moves $R0$ and $R2$ to $R5$. The basic notation of a spatial graph introduced in Sec. 5.2 is sometimes referred to as a *pliable* spatial graph to contrast the notion with flat vertex and ribbon graphs. This chapter only focuses on pliable and flat vertex graphs, but note that ribbon graphs would be useful for measuring twisting along interconnects in the final 3D system.

5.3 Yamada Polynomial Invariants

Reidemeister moves are valuable for identifying when two embeddings are isotopic (that is, topologically equivalent); however, finding the specific sequence of moves between two equivalent spatial diagrams can be extremely challenging, especially when the spatial graphs have many nodes and edges. Even for knots, which are the simplest class of spatial graphs, it is unknown whether there exists a polynomial-time algorithm for determining when two knots are isotopic. (It is not impossible that such an algorithm exists: the question of

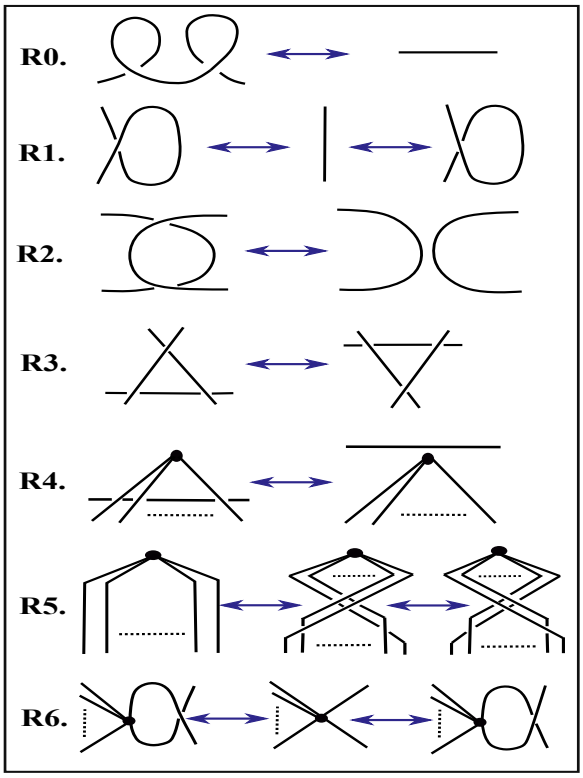


Figure 5.3: Fundamental Reidemeister moves for spatial graphs.

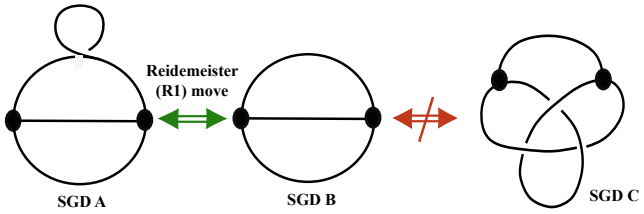


Figure 5.4: SGDs A and B are topologically equivalent θ_1 graphs under Reidemeister-I (RI) move. C is a θ_2 graph and is topologically distinct from A and B under any fundamental R moves.

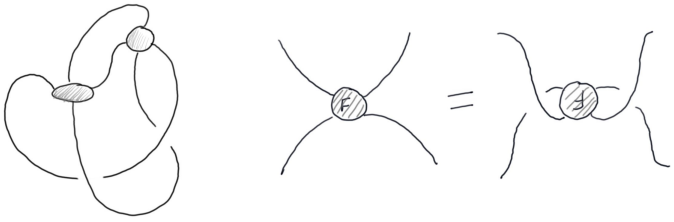


Figure 5.5: At left is a typical flat vertex graph. The Reidemeister $R5$ move on this kind of spatial graph is shown at right.

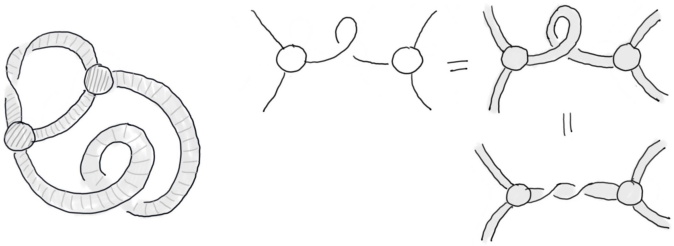


Figure 5.6: At left is a typical ribbon graph; notice the twists in the topmost band. The upper-right shows how a SGD encodes a ribbon graph via the blackboard framing, with the vertical isotopy showing how loops in the SGD can describe twists in the ribbons.

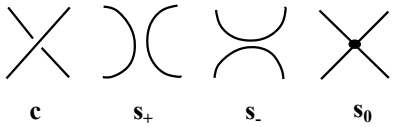


Figure 5.7: Class of spins for a crossing.

whether a knot is equivalent to a round circle is in **NP**∩**coNP** [165,166].) To show that two embeddings are not *isotopic* requires an *invariant*: a function of the embeddings whose output is not changed by isotopies, and which takes different values on the two embeddings [167–170]. Mathematicians often use such invariants that are computable and yet powerful enough to detect some delicate differences of embeddings of the same graph. Over the last century, many *polynomial invariants* [168, 171–173] were discovered by knot theorists, such as the Alexander-Conway [174], Jones [175], Kauffmann [176], and Yoshinaga [177] polynomials. Some of these have been extended to spatial graph theory [178–180] using similar constructions. These invariants satisfy nice *skein relations* which are mathematical tools that give linear relationships between the polynomials of closely related diagrams. Relevant skein relations are sufficient to calculate the polynomials recursively and are relatively convenient to use for this purpose. The proof of invariance then relies on using the skein relation to show the value of the invariant is unchanged by Reidemeister moves.

5.3.1 Yamada polynomial properties

The specific polynomial invariant used here is the Yamada polynomial, which associates to each SGD a polynomial in an indeterminate A, which is an ar-

bitrary independent variable. For example, the Yamada polynomial for the SGD C in Fig. 5.4 is $-A^{-6}-A^{-5}-A^{-4}-A^{-3}-A^{-2}-1-A^2+A^6$. The Yamada polynomial is defined in terms of a polynomial invariant H of ordinary (non-spatial) graphs. Continuing the example, the abstract theta graph that underlies all the SGDs in Fig. 5.4 has $H(A) = -A^{-2}-A^{-1}-2-A-A^2$. First, the polynomial invariant H and its properties P1–5 are discussed, which can be used to recursively compute H for any graph. Then the Yamada polynomial and its fundamental properties S1–4 are mentioned which similarly allow for recursive computation of it for any SGD. After giving some helpful additional properties Y1–8 of the Yamada polynomial, it is computed for four different example SGDs in Sec. 5.3.2.

Let $G = (V, E)$ be an abstract graph, where V is the vertex set and E is the edge set of G. For two graphs G_1 and G_2, $G_1 \sqcup G_2$ denotes the disjoint union of G_1 and G_2, and $G_1 \vee G_2$ denotes a wedge at a vertex of G_1 and G_2, that is $G_1 \vee G_2 = G_1 \cup G_2$ and $G_1 \cap G_2 = \{\text{vertex}\}$. Disjoint union of graphs is analogous to disjoint union of sets, is an operation that combines two or more graphs to form a larger graph. A disjoint union of graphs is constructed by making the vertex set be the result of the disjoint union of the vertex sets of the given graphs, and by making the edge set of the result be the disjoint union of the edge sets of the given graphs. More generally, every graph is the disjoint union of connected graphs, or its connected components. Intersection of two graphs is constructed by the common vertices and edges between the given graphs. For example, if $G1$ s a graph with labeled vertex set $\{1, 2, 3\}$ and labeled edge set $\{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$, and $G2$ is a graph with vertex set $\{2, 3, 5, 6\}$ and edge set $\{\{2, 3\}, \{3, 5\}, \{5, 6\}, \{2, 6\}\}$. So $G_1 \cup G_2$ is a graph with vertex set $\{1, 2, 3, 5, 6\}$ and edge set $\{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{3, 5\}, \{5, 6\}, \{2, 6\}\}$. As you can see, I have simply taken the union of the vertex and the edge sets. Similarly $G_1 \cap G_2$ is a graph with vertex set 2,3 and edge set 2,3. Here I have taken the intersections of the vertex and edge sets. In addition, a graph G has a cut-edge e (also known as bridge or isthmus) if $G - e$ has more connected components than G. First, following Ref. [181], a polynomial invariant $H(G)(A)$ of an abstract graph G is described, where A is an indeterminate (arbitrary independent variable); precisely, our $H(G)(A)$ is Yamada's $h(x, y)$ with $x = -1$ and $y = -A - 2 - A^{-1}$. The polynomial

$h(G)(A)$ is characterized by the following properties:

P1. $H(\text{empty graph}) = 1$ and $H(\text{simple loop}) = A + 1 + A^{-1}$.

P2. $H(G_1 \sqcup G_2) = H(G_1)H(G_2)$

P3. $H(G_1 \vee G_2) = -H(G_1)H(G_2)$

P4. If G has a cut edge, then $H(G) = 0$.

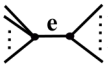
P5. Let e be a non-loop edge of a graph G. Then $H(G) = H(G/e) + H(G - e)$, where G/e is the graph obtained from G by contracting e to a point and $G - e$ is G with e deleted.

Based on the above rules, a powerful and much-studied invariant of spatial graphs is defined, which is the *Yamada polynomial* [180–184]. Let g be the a spatial graph diagram. For a crossing c of g, three *graph reductions* are defined as: s_+, s_-, and s_0 (denotes a vertex), with the class of spin $+1$, -1, and 0, respectively, as shown in Fig. 5.7. These graph reductions are used to replace crossings in a spatial graph for Yamada polynomial calculation. Let S be the planar graph obtained from g by replacing each crossing with a spin. S is called a *state* on g and $U(g)$ denotes the set of states on g obtained by applying all possible reductions in its crossings. Set $\{g|S\} = A^{n_1 - n_2}$, where n_1 and n_2 are the numbers of crossings with spin of $+1$ and spin of -1, respectively, and A is an indeterminate. The Yamada polynomial $R\{g\}(A) \in \mathbf{Z}[A, A^{-1}]$ is defined as:

$$R(g) = R(g)(A) = \sum_{S \in U(g)} \{g|S\} H(S),$$

In particular, if the diagram of g does not have crossings, then $R(g) = H(g)$. This Yamada polynomial for a spatial graph can be computed recursively using the following skein relations and the properties of H:

S1. $R($  $) = AR($  $) + A^{-1}R($  $) + R($  $)$,

S2. $R($  $) = R($  $) + R($  $)$, where e is a non-loop edge.

S3. $R(g_1 \sqcup g_2) = R(g_1)R(g_2)$,

S4. $R(g_1 \vee g_2) = -R(g_1)R(g_2)$.

So far, the Yamada polynomial is a function of the given diagram g and one needs an invariant of the spatial graph $\tilde{G}$ it describes. Yamada showed:

I1. Any two diagrams g and g' whose flat vertex graphs $\tilde{G}$ and $\tilde{G}'$ are isotopic have $R(g') = (-A)^n R(g)$ for some integer n.

I2. If every vertex has valence at most three, then two diagrams g and g'

whose spatial graphs $\tilde{G}$ and $\tilde{G}'$ are isotopic have $R(g') = (-A)^n R(g)$ for some integer n.

I3. Any two diagrams g and g' whose associated ribbon graphs $\tilde{G}$ and $\tilde{G}'$ are isotopic have $R(g') = R(g)$.

The next set of relations for the Yamada polynomial can be derived from the previous ones and are very useful aides for its calculation. Detailed proofs for these relations (Y1-Y8) are provided in Ref. [181]. They are as follows:

Y1. $R($  $) = B, \quad$ where $B = A + 1 + A^{-1}$,

Y2. $R($  $) = -BR($$)$,

Y3. $R($$) = -AR($$) - (A^2 + A)R($$)$,

Y4. $R($  $) = -A^{-1}R($$) - (A^{-2} + A^{-1})R($$)$,

Y5. $R($  $) = -AR($$), \; R($$) = -A^{-1}R($$)$,

Y6. $R($  $) = A^2 R($$), \; R($$) = A^{-2}R($$)$,

Y7. Edge subdivision does not change the polynomial:

$R($  $) = R($$)$

Y8. Petals to concentric self-loops:

$R($  $) = R($$) = -B^2 = -(A + 1 + A^{-1})^2.$

5.3.2 Illustrative Examples

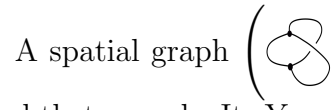
Yamada polynomials for a few spatial graphs are calculated by reducing the spatial graph diagram into a linear combination of smaller elements based on the skein relations stated above.

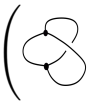
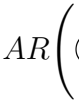
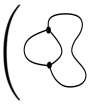
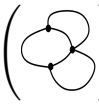
Example 1 - Theta (θ_1) graph: The Yamada polynomial for a standard theta graph is calculated as follows:

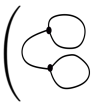
$$R\Big($$  $$\Big) = \underbrace{H\Big($$$$\Big)}_{\left(\begin{smallmatrix}\text{Apply S2 on}\\ \text{the center}\\ \text{edge}\end{smallmatrix}\right)\rightarrow} = \underbrace{H\Big($$$$\Big)}_{\downarrow\text{(Apply Y7)}} + \underbrace{H\Big($$$$\Big)}_{\downarrow\text{(Apply Y8)}},$$

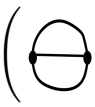
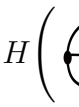
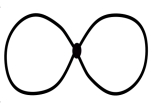
$$\implies R\Big($$$$\Big) = H\Big($$$$\Big) + H\Big($$$$\Big),$$

$$\implies R(\theta_1) = B - B^2, \qquad (\text{where } B = A + 1 + A^{-1})$$
$$\implies R(\theta_1) = -(2 + A + A^{-1} + A^2 + A^{-2})$$

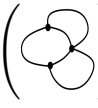
Example 2: A spatial graph  which is isotopic (by the R6 move) to the standard theta graph. Its Yamada polynomial is calculated as follows, though one could instead use property Y5 as a shortcut.

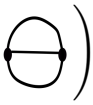
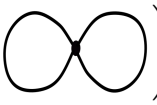
$R($  $) = AR($  $) + A^{-1}R($  $) + R($  $),$

$R($  $) = 0, \qquad (\text{because of isthmus based on property P3})$

$R($  $) = H($  $) = H($  $) + H($  $),$

$= B - B^2, \qquad (\text{where } B = A + 1 + A^{-1})$

$R($  $) = H($  $) = H($  $) + H($  $),$

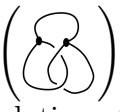
$= H($  $) + H($  $) + H($  $),$

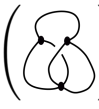
$= B^3 + (-B^2 + B) + -B^2 = B^3 - 2B^2 + B,$

$\implies R($  $) = A(0) + A^{-1}(B - B^2) + (B^3 - 2B^2 + B),$

$\implies R($  $) = A^3 + A^2 + 2A + 1 + A^{-1},$

$= -A(R(\theta_1)).$ Note the $-A$ factor permitted in I2.

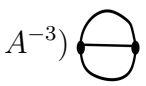
Example 3: The spatial graph is  . This example involves extensive use of Yamada polynomial skein relations (Y1-Y8). Its Yamada polynomial is calculated as follows:

$= \underbrace{AR($  $)}_{\downarrow(\text{Apply Y5})} + \underbrace{A^{-1}R($  $)}_{\downarrow(\text{Apply Y6})} + R($  $),$

Drop the Rs for simplicity,

$= A(-A$  $+ A^{-1}(A^{-2}$  $) +$  ,

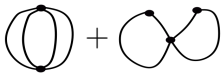
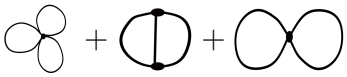
↓(Apply Y4)

$= (-A^2 + A^{-3})$  $+ -A^{-1}$  $- (A^{-2} + A^{-1})$ 

$= (-A^2 - A^{-2} - A^{-1} + A^{-3})$  $- A^{-1}$  ,

$= A^4 + A^3 + A^2 + A - A^{-2} - A^{-3} - A^{-4} - A^{-5}$

Since taking $B = A + 1 + A^{-1}$ we have:

 $=$  $=$  ,

$= +B^3 + (-B^2 + B) + -B^2 = B^3 - 2B^2 + B.$

Example 4: The spatial graph is  . This example involves extensive use of skein relations (Y1-Y8). Its Yamada polynomial is calculated as follows:

 $= AR$  $+ A^{-1}R$  $+ R$  ,

Drop the Rs for simplicity,

$= A^3$  $+A^{-1}$  $-A$  $- (A^2 + A)$  ,

Example 2 Example 3

$= -A^4$  $+ -A^{-1}$  $- A(-A^{-1})$ 

$- (A^{-2} + A^{-1})$  $- (A^2 + A)(-A)$ 

$= (-A^4 + A^{-1} + 1 + A^3 + A^2)$  $+ A^{-1}$  $+$ 

$= A^6 - A^2 - 1 - A^{-2} - A^{-3} - A^{-4} - A^{-5} - A^{-6}.$

5.4 3D Spatial Topology Enumeration Framework

Figure 5.9 shows the steps of the proposed design framework to represent, enumerate and categorize unique spatial topologies of a 3D system (assuming inter-component connectivity is fixed). The detailed steps of the enumeration design framework are:

- **1. Define system architecture:** Provide the specific 3D system architecture (SA) for which spatial topologies must be enumerated. From the SA, extract the number of nodes (components), their valencies, system interconnectivity, and the corresponding edges (interconnects in the system).
- **2. Enumerate spatial graph diagrams:** Combinatorially enumerate all possible spatial graph diagrams (SGDs) for the SA from zero crossings to the maximum crossing number ($k = 0, 1, ..., k_m$) using their corresponding shadows.
- **3. Check graph planarity:** Planar diagrams (PDs) of spatial graphs are used for the calculation of the Yamada polynomials. However, before calculation, each enumerated graph must be checked to determine whether it is planar, for which there are linear-time algorithms [185]. The graphs start with a circular order of the edges at each vertex, making the planarity check even easier. The algorithm shown in Fig. 5.8 recursively contracts the edges of a graph until the diagram is a bouquet of circles, and then use the fact that if the diagram is planar, there must exist at least one loop edge whose endpoints come consecutively in the cyclic ordering around the vertex, i.e., a self-loop. This self-loop can be removed without altering planarity. The recursive steps for the planarity check (PC) algorithm are enumerated below:

  PC1. Convert all vertices to crossings, as it does not affect planarity.

  PC2. Contract all non-loop edges (edges shared between two vertices) to a vertex, as it does not affect planarity.

  PC3. Remove all planar self-loops at a vertex.

  PC4. Empty vertex does not affect planarity, so remove it. By doing all these steps recursively, if the result is an empty diagram, then the original diagram is planar. If not, the diagram is non-planar.

- **4. Evaluate Yamada polynomials:** The Yamada polynomials for all the valid planar SGDs are evaluated using the Yamada polynomial

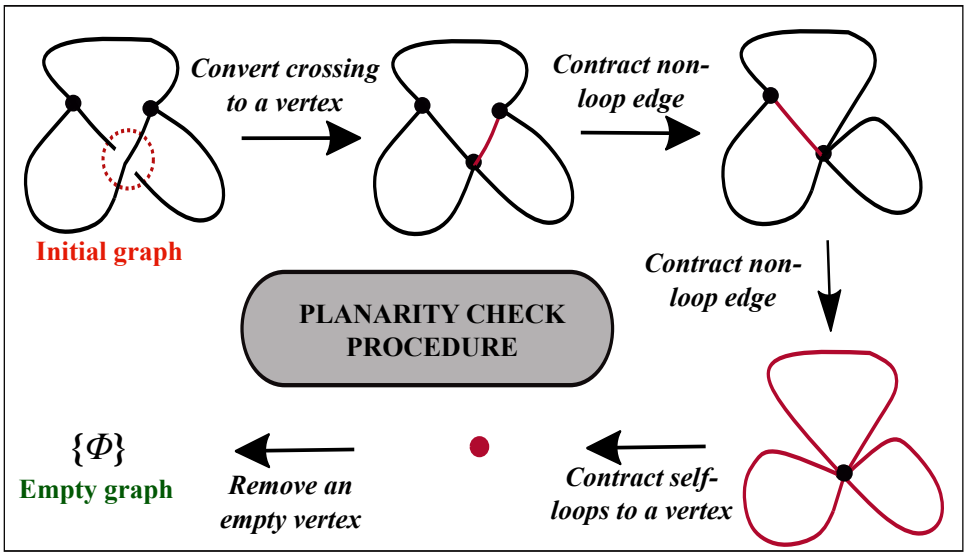


Figure 5.8: Identifying connected planar spatial diagrams from the combinatorially enumerated set is performed using this procedure.

properties detailed in Sec. 5.3.

- **5. Categorize different spatial topologies**: Cluster SGDs into classes so that the SGDs in each class have the same Yamada polynomial and no two classes have the same polynomial. Any pair of SGDs that represent isotopic spatial graphs will be in the same Yamada class, so then only one SGD per Yamada class is considered. In most engineering applications, multiple crossings between a pair of interconnects is undesirable. For example, there is no advantage to two wires intertwining several times unless they intentionally function as a twisted pair. Similarly, a pair of pipes that intertwine are more costly to fabricate, and more complex to install and remove. Hence, from each class an SGD is selected having the fewest crossings, and is used to generate a 3D geometric model.
- **6. 3D model generation:** Simpler SGDs from step 5 are utilized as underlying skeleton structures for generating various 3D system geometric models.

5.4.1 Comparison to Other Methods

Steps 2–5 above follow the standard strategy in low-dimensional topology for enumerating knots, which are a special class of spatial graphs, see [88]

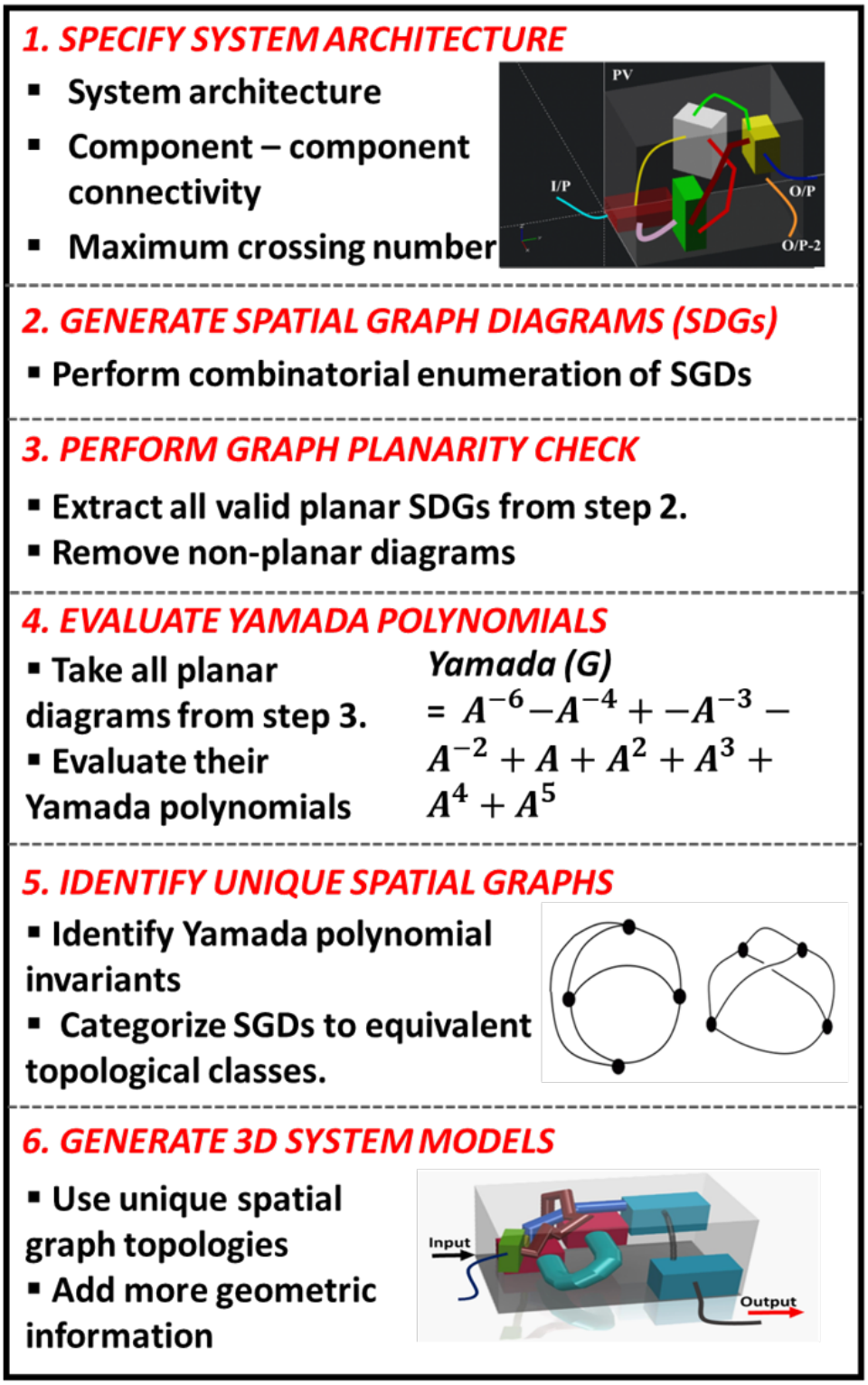


Figure 5.9: The six sequential steps of the proposed spatial graph-based topology enumeration framework.

for an overview. Using additional techniques from hyperbolic geometry, it is possible to exactly enumerate all knot topologies with less that 20 crossings, of which there are more than 350 million [186].

Compared to such massive computations, prior work on spatial graphs where the underlying graph is more complicated than just a loop is limited: mostly tabulations of less than 100 topologies [187–192]. For example, the authors in Ref. [190] generated two vertex bouquet spatial graphs with a maximum of seven crossings. As our cases studies demonstrate, the strategy above allows for much larger-scale enumerations, with arbitrary specified system architectures. This is very suitable for representing large-scaled complex engineering systems easily, and for enumerating their spatial topologies efficiently.

5.5 Case Studies

In this section, a number of case studies are provided to demonstrate the proposed enumeration framework discussed in Sec. 5.4. All computations (Yamada polynomial calculation, planarity check, etc.) in the case studies were performed using Wolfram Mathematica 11.3 software with an Intel Xeon E5-2660 CPU @ 2.00 GHz, 64 GB DDR4-2400 RAM, Windows 10 64-bit workstation.

5.5.1 Case Study 1: Components with equal valencies

In this case study, a 3D system with architecture as shown in Fig. 5.10 is considered for ST enumeration. This system contains four identical trivalent components (nodes), and six interconnects (edges). It is found that the unique 3D spatial topologies (STs) of the system for crossing numbers varying from zero to three. The notation used to indicate each SGD is given by SGD_k where k is the crossing number of that diagram, and the letter refers to the specific SGD. SGD_0 is the original system architecture without any crossings. Using the proposed framework, all the SGDs are combinatorially enumerated and passed through a planarity check procedure. Yamada polynomials were then calculated for 3, 31, 118, and 231 valid planar SGDs having 0, 1, 2 and 3 crossing numbers respectively. The SGDs having the

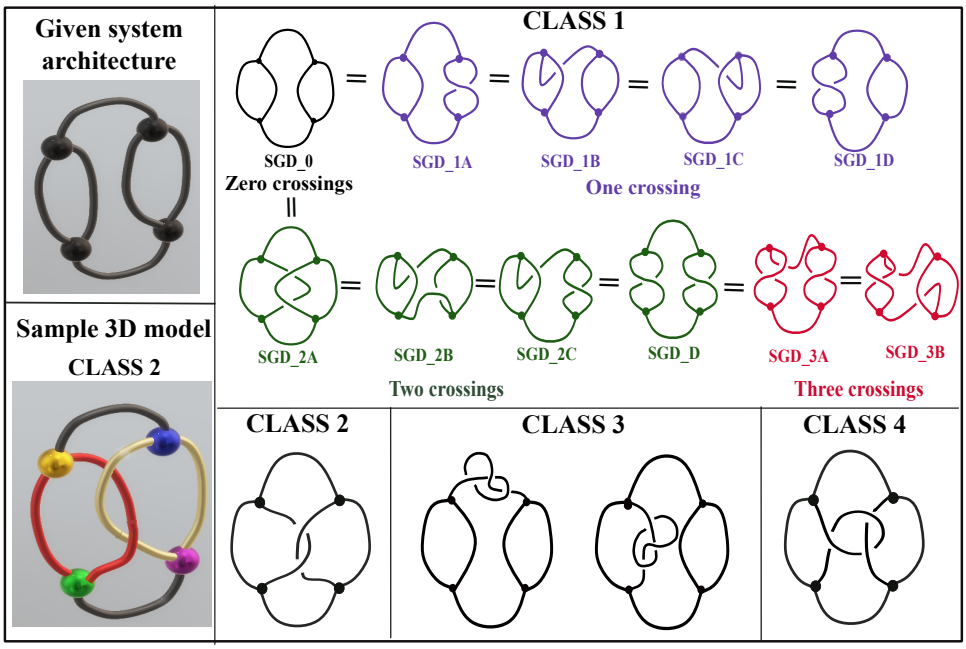


Figure 5.10: Results from case study 5.5.1 for a given system architecture for maximum crossing numbers from 0 through 3.

same Yamada polynomial or differing by a factor $(-A)^n$ under the same topological class are grouped together based on property I2. Through this a total of four unique Yamada classes are attained as shown in Fig. 5.10. The Yamada polynomials for these different classes of topologies are shown in Table 5.1. For class 1, it is observed that three distinct Yamada polynomials exist for different crossing numbers, but they all differ by a factor $(-A)^n$. This strongly suggests that the SGDs shown under class 1 are isotopic to SGD_0, and indeed this can be verified using the R6 move. In contrast, the two SGDs in class 3 can be shown to be nonisotopic using other tools. A sample 3D model of class 2 spatial topology candidate design solution is also shown in Fig. 5.10. The computational time for this entire case study is 78.3 seconds (s).

5.5.2 Case Study 2: Scaling the system

The goal of this case study is to provide insight about how the number of unique spatial topologies varies with increasing number of components and the maximum number of crossings in a 3D system. For consistency, here all the components are assumed to be trivalent. Table 5.2 shows the number

Table 5.1: Yamada polynomials of diagrams shown in Fig. 5.10 (case study 5.5.1).

Classes	**Yamada polynomials**
Class 1 SGD_0:	$A^{-3}+A^{-2}+3A^{-1}+2+3A+A^2+A^3$
Class 1 SGD_1:	$(-A)^1(A^{-3}+A^{-2}+3A^{-1}+2+3A+A^2+A^3)$
Class 1 SGD_2:	$(-A)^2(A^{-3}+A^{-2}+3A^{-1}+2+3A+A^2+A^3)$
Class 1 SGD_3:	$(-A)^3(A^{-3}+A^{-2}+3A^{-1}+2+3A+A^2+A^3)$
Class 2	$A^{-4}+A^{-3}+A^{-2}+2A^{-2}+2A+2A^3+A^4+A^5+A^6$
Class 3	$A^{-7}+A^{-6}+3A^{-5}+3A^{-4}+4A^{-3}+3A^{-2}+3A^{-1}$ $-3A^2-2A^3-2A^4-A^5+A^6+A^8$
Class 4	$A^{-7}+A^{-6}+A^{-5}+2A^{-4}+2A^{-3}+A^{-2}+3A^{-1}+2A-A^2-A^5+A^6$

of unique Yamada classes obtained for each of the component and crossing number combinations, where here SGDs having the same Yamada polynomial or differing by a factor $(-A)^n$ are categorized together. The computational time increased drastically as one moves from a small-scaled (2 components and 2 crossings) to a large-scaled (10 components and 10 crossings) system case study from 12.3 sec to 24,256.6 sec (6.73 hrs), respectively.

Table 5.2: Case study 2: Unique spatial topology classes with increasing number of components and crossings.

Components \ **Crossings**	2	4	6	8	10
2	1	2	3	4	6
4	1	4	6	8	9
6	2	4	8	15	21
8	4	6	9	20	28
10	5	12	16	27	36

5.5.3 Case Study 3: Components with different valencies

Unlike case study 5.5.1, a system with four components but each with different valencies (number of ports) is considered. In addition, flat vertex graphs (FVGs) as described in Sec. 5.2.2 are used. As FVGs have local structures at nodes, the edge connectivity order around the nodes is preserved, and thus FVG representations are highly suitable for design applications where

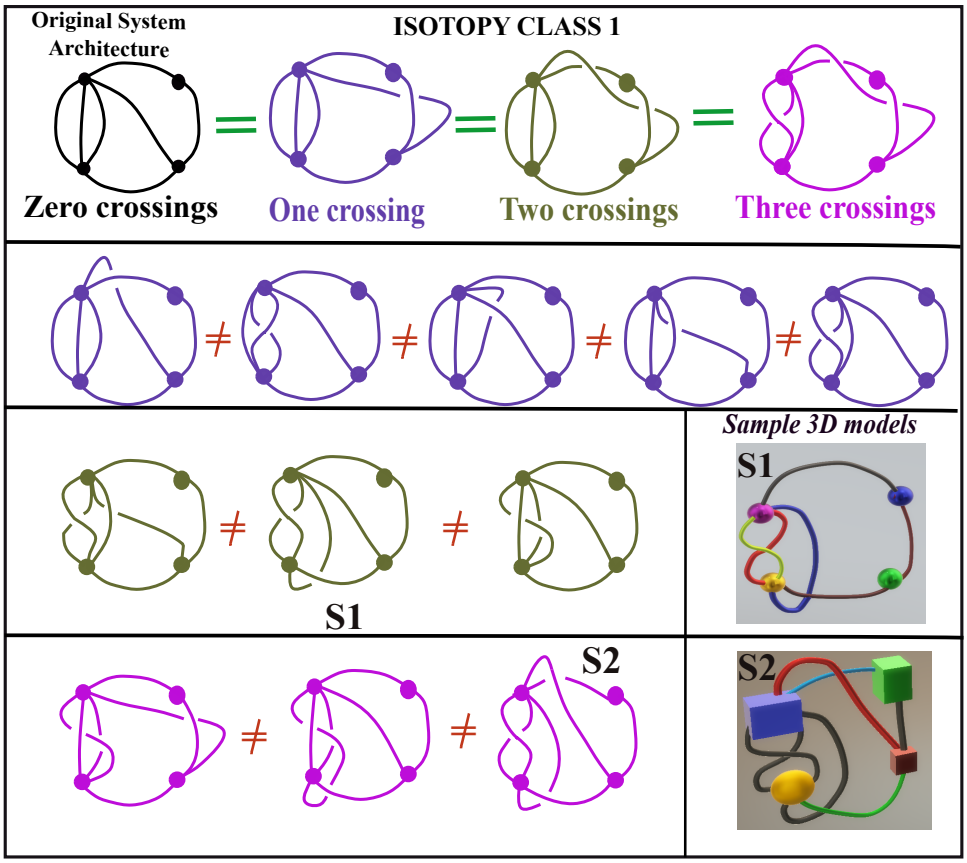


Figure 5.11: Results from case study 5.5.3 for a given system architecture for maximum crossing numbers from 0 through 3 with components having different valencies.

nodes have a specific cyclic ordering of ports. Here R0 to R5 moves are valid but not R6. Figure 5.11 shows some of the results obtained in this study. After computing the Yamada polynomials of hundreds of planar SGDs, a total of 27 unique Yamada classes are obtained. For illustration purposes, some isotopes of SGD_0 (original system architecture) are shown as class 1 isotopes. Furthermore, unique SGDs belonging to some unique Yamada classes are shown for crossing numbers one, two, and three respectively. Two final 3D system geometric models (referred as S1 and S2) are also shown in Fig. 5.11. The total computational time taken for study B is 211.4 sec. It can be observed from this study that with components with different valencies, more unique Yamada classes are attained than those with identical components. Thus, manually generating such designs is very challenging and the automated enumeration framework proposed here is very valuable.

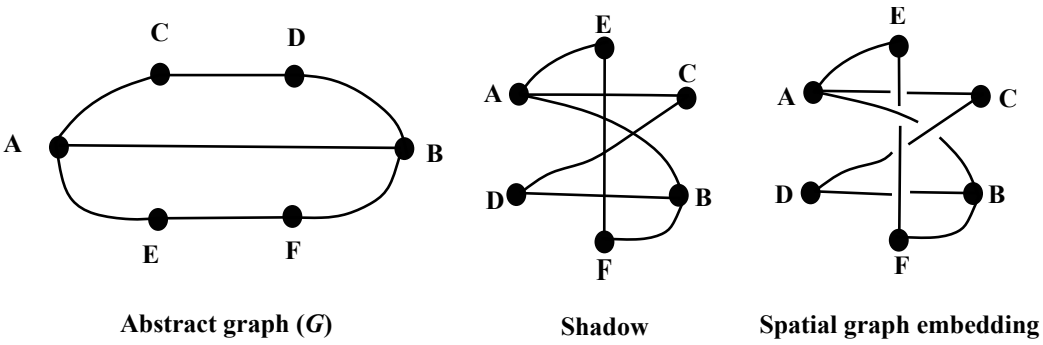


Figure 5.12: Implementation of the circular spatial graph representation technique in Sec. 5.5.4 to avoid unnecessary or extra twists between any two edges of a diagram.

5.5.4 Case Study 4: Circular graph representation

While filtering out isotopic spatial graph diagrams in the previous case studies, it is observed that in a few occasions where system topologies have many crossings, two edges in that diagram twist around each other multiple times. Although a higher crossing number is satisfied, intertwining between edges can often be reduced by Reidemeister moves to a smaller crossing number, so essentially, no unique spatial topology is attained. Such intertwining is practically not observed or desirable between pipes or ducts in most complex systems (e.g., aero-engine externals, hydraulic systems). Some syntactic constraints need to be imposed to prevent more than a simple crossing between any two edges. This requires a representation that implicitly forbids twisting of two edges multiple times around each other. One way to get different spatial embeddings of an input abstract graph G as shown in Fig. 5.12 is to: 1) Pick an ordering of the nodes and use that to arrange them along a circle on the plane, 2) Connect the nodes by straight lines corresponding to the edges of G. This gives the "shadow", and 3) Resolve the intersections lines of the shadow into over or under crossings.

Figure 5.12 shows the shadow of graph G based on a particular cyclic order of nodes and one spatial graph embedding. As there are five crossings, a total of $2^5 = 32$ spatial embeddings are possible. The unique ones can be identified using the proposed design framework.

5.5.5 Case Study 5: Large-scaled system - spatial graph decomposition approach

From the observations made in the previous case studies, it is evident that enumerating spatial topologies for most real-world systems containing many components and approximately hundreds of crossings is intractable with manual processes and can become computationally expensive with automated methods such as the one presented in this chapter. In contrast, enumerating spatial topologies of each subsystem of components can be a simple and efficient process. A complex spatial graph can thus be converted to a set of sub-graphs, and the unique spatial topologies of these sub-graphs can then be enumerated separately. The sub-graphs can be decoupled and can be considered as super-nodes. This decouples the task into two subtasks: 1) Enumerate STs of the system graph with only the subsystems as super-nodes, and 2) enumerate unique STs within each sub-graph. This presents fewer design candidates about which to make decisions, which greatly reduces the overall computational expense. Figure 5.13 shows a random complex spatial graph with 14 nodes, 20 edges and allowing at most 10 edge crossings. Approximately 1.134×10^4 SGDs are attained for this entire system that fall under 434 unique Yamada polynomial categories. As this is a very large set, decomposition of the graph into sub-graphs (as super-nodes) is appropriate. First, a unique spatial topology of the super-nodes graph is found. Case study 1 in Sec. 5.5.1 is utilized as a sub-graph for demonstration purposes. Note that while enumerating STs for the spatial sub-graph, the rest of the system is condensed as an extra node in the sub-graph to preserve spatial connectivity information. Finally, using the proposed design framework, unique STs of the sub-graph can be plugged into the original system to attain system configurations. The scope of this chapter only deals with enumerating unique STs, hence research related to how each of these unique topologies affect overall system performance in considered part of future work.

To explain this decomposition concept using a concrete engineering design example, suppose that the complex network represents the spatial topology of a hybrid-electric vehicle powertrain; one possible subsystem could be a fluid-thermal cooling circuit. Each distinct circuit topology can be geometrically optimized for fair comparison, revealing how the topological features contribute to the overall system efficiency, fuel economy, thermal loss man-

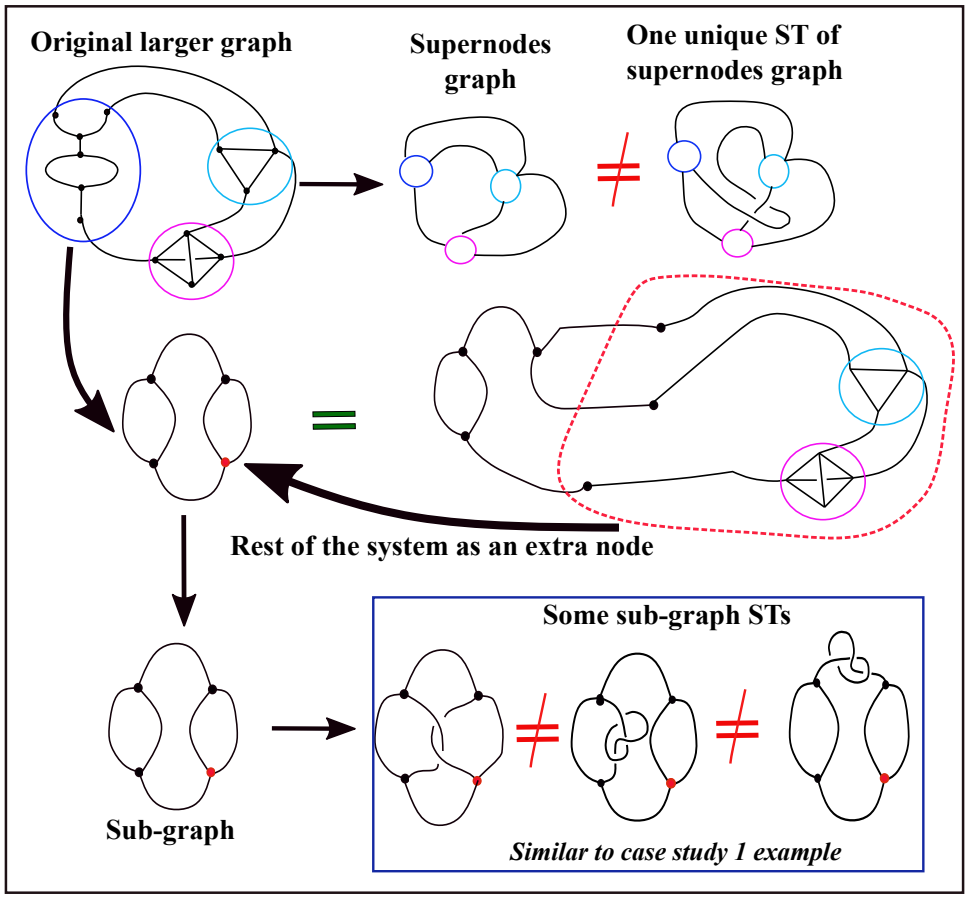


Figure 5.13: Demonstration of spatial graph diagram decomposition approach discussed in case study 5.5.5.

agement, and other figures of merit due to physical interactions between components, interconnections, and the environment. The best candidate ST can then be chosen according to the desired performance requirements as in Refs [103, 105] where same procedure was followed but for ranking different system architectures (SAs).

5.5.6 Braid-based representations

Braids refer to a collection of strings crossing each other between two planes. Each braid can be represented using a *braid word* that provides information about the crossings (either under or over) and their locations within the braid. Initial progress was made in enumeration of braid-closure based spatial graphs for 3D system design representation. Braids are very good mathematical forms for representing crossing between strands. In the 3D

PR problem, these can be considered as interconnects passing under or over each other. Developments include:

- An efficient enumeration algorithm that can generate all braid representations for N strands and K crossings using braid word combinations.
- Nodes, branches and merges are created outside the braid boxes to convert general braid representations to spatial graphs. Nodes can be considered as components in a 3D system.
- Simple geometric shapes were used as nodes with varying valencies (number of ports) to mimic a PR problem.
- Spatial graph invariants are successfully extracted from an exhaustive enumeration set by checking for braid equivalency using lexicographic braid word comparisons.
- Unnecessary spatial topologies with different system architectures are eliminated by checking the port connectivity matrices of all the enumerated braid-based graphs.
- Inspiration derived from asteroid field theory are used to cover with more topological aspects of the design representation.

Some of the results are shown in figures below. Figure 5.14 shows braid words and corresponding braid representations for a braid enumeration method with $N = 4$ strands and maximum $K = 3$ crossings. Corresponding braid words are shown on top of each braid. Figure 5.15a shows 20 randomly generated braid-closure based spatial graphs for a $N = 5$ strands, $K = 6$ crossings with two tri-valent and two bi-valent nodes. The red cherries represent the nodes. The next Fig. 5.15b shows nodes with simple geometric shapes for the same example shown in Figure 5.15a. These are just simple demonstrations of how braids can be very useful representing interconnect interaction between components of a 3D system. Similar to spatial graphs, polynomial invariants can be used to identify unique braid topologies and braid-based enumeration can be implemented to study large-scale systems. But these topics are out of scope of this dissertation and will be investigated in future work.

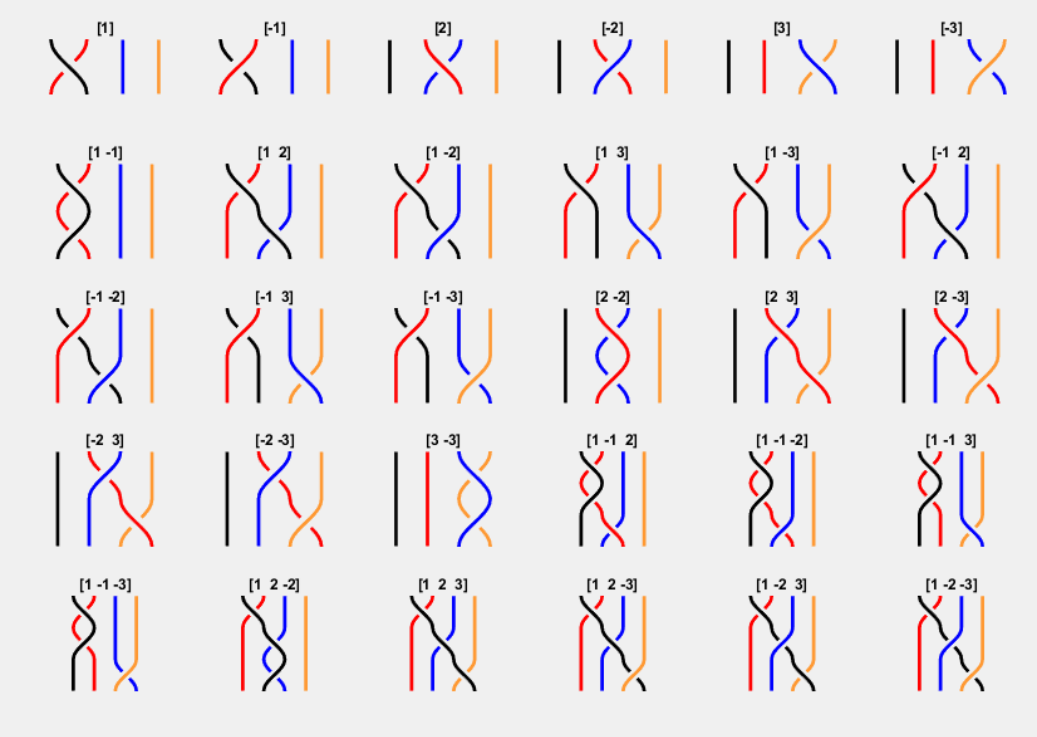


Figure 5.14: First 30 braid words for a set of 4-braid with maximum 3-crossing representations.

5.6 Discussion

This section summarizes a list of important observations from the five case studies as follows:

1. From the above case studies, especially 5.5.1, and 5.5.2, it can be observed that the number of unique spatial topologies finally attained for a given interconnect crossing complexity are much smaller than the combinatorially enumerated set of spatial graph diagrams as most of them are isotopic to each other under the Reidemeister moves.
2. Case study 5.5.3 presents another contribution of this work as the proposed method enumerates and identifies newer spatial topologies for components with different valencies in contrast to existing work that is mostly limited to two or three equivalent vertices. [187–191].
3. The circular graph representation method, presented in case study 5.5.4, is a simple way to enumerate and realize SGDs and avoid edge intertwining, although Yamada polynomials should still be used for identifying unique STs. Furthermore, specific syntactic constraints can be

added to significantly reduce the initial set of SGDs obtained for planarity checking and Yamada polynomial evaluation. For example, by adding constraints on total crossings allowed between two edges of a system, there is greater control on the type of spatial topologies finally obtained. This will be studied more in future work.

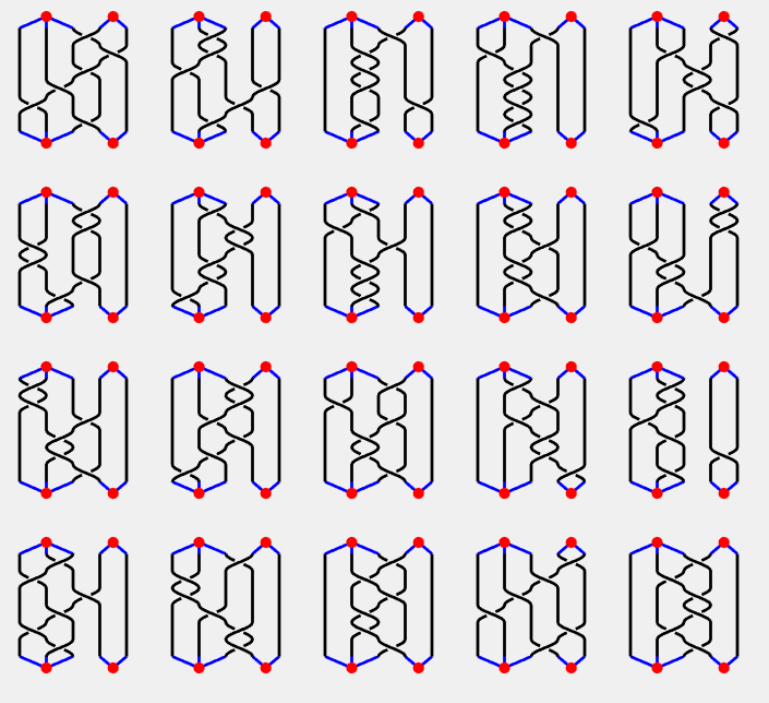
4. As seen in case study 5.5.5, for large-scaled systems, the best way to achieve different STs and search effectively is by graph decomposition. The spatial graph of the subsystem, which plays a critical role in performance impact, can be extracted to find its unique topologies. This avoids the need to enumerate thousands of diagrams of a complex network, compute their polynomials and compare them. Moreover, subgraph designs can be optimized for performance independently and then combined with the remaining system.
5. Another very impactful aspect of this framework is that for one system architecture, there can be a range of spatial topologies from zero to many crossings. The spatial embeddings with fewer interconnect crossings are generally more useful for practical engineering purposes. Therefore, for existing, complex real-world system designs with 10s of crossings, using the proposed design framework, a simpler spatial topology can be found for that network with a much lower crossing number, but still keeping the same system connectivity.

In this chapter, it is more important to illustrate different topological designs than to analyze the difference in the corresponding Yamada polynomials, hence, the polynomials for all the diagrams shown in these case studies are not listed.

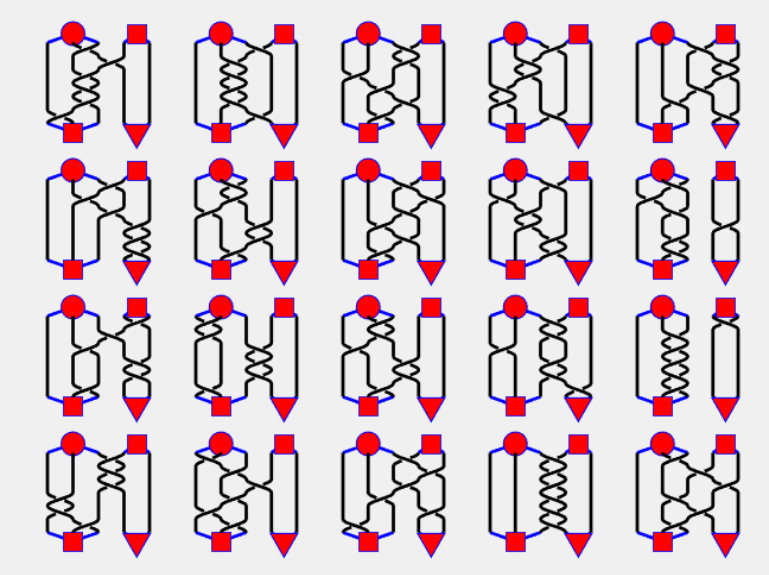
5.7 Summary

The design representation presented in this chapter greatly enhances the study of 3D engineering system spatial topologies in a systematic manner and is supported by rigorous mathematical foundations in spatial graph theory. Topologies of complex engineering systems, designed for particular applications, are conventionally created manually. But for more effective performance and efficiency, systematic identification, enumeration, and classification of possible system topologies is necessary. A framework for repre-

senting three-dimensional interconnected engineering systems using spatial graph embeddings is presented. Initially, all the combinatorial spatial graph descriptions up to some fixed topological complexity are enumerated for an input system architecture. A polynomial invariant, the Yamada polynomial is then calculated for the set of all the spatial graphs attained from the combinatorial permutations. The Yamada polynomial helps identify the duplicate spatial graph topologies from the exhaustive set and a smaller set of unique spatial embeddings (equivalent topological classes) is obtained. This smaller set of spatial graphs can be used for generating three dimensional geometric system models. Five case studies have been demonstrated using the proposed enumeration strategy. The results show that this method is efficient, scalable, applicable to all general 3D interconnected system networks, allows comprehensive exploration of the design space, and greatly aids in the design and development of unprecedented system topologies. Details on future work are presented in Chapter 6.



(a) First 20 random braid-closure based spatial graphs.



(b) Simple geometric shapes embedded into the spatial graphs

Figure 5.15: Braid-closure based spatial graph representations for $N = 5$ strands and $K = 6$ crossings system with two bivalent and trivalent components.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Dissertation summary

In this chapter the dissertation is summarized in addition to highlighting the main contributions that have been made. A number of interesting future work items that could be performed are also outlined. The optimal spatial packaging of complex engineering systems is a challenging multi-faceted problem. To tackle the different research elements of SPI2 together requires an innovative design automation framework. The two-stage sequential design framework proposed in this dissertation has been effectively used to solve the SPI2 problem by combining component packing, interconnect routing, and physics performance evaluation sub-problems together. Chapters 2-5 contain the core design method contributions of this dissertation.

Chapter 2 focuses on reviewing four key SPI2 design research areas comprehensively. The interfaces that exist between component packing, interconnect routing, design representations, and physics-based topology optimization have been addressed in that chapter. A list of critical gaps are outlined and some of them have been presented in this dissertation. For example, the continuous and discrete elements of the SPI2 design problem were separately addressed in the two stages of the design framework. Similarly, a holistic systems approach was adopted to combine the pieces of the SPI2 problem together in stage 2. Furthermore, several associated SPI2 design challenges were pointed out in Chapter 2. This chapter sets a technical foundation that helps in building practical design methods that are presented in the remaining chapters.

In the next chapter, the major similarities and differences between the 2D

and 3D SPI2 design problems are outlined. The bulk of Chapter 3 focuses on the description and implementation of stage 1 of the two-stage sequential design framework to 2D SPI2 problems. Three novel and computationally efficient methods: force-directed layout generation, shortest path algorithms, and unique spatial topology enumeration algorithms for generating spatially-feasible 2D geometric layouts have been presented. Final 2D design layouts attained from these algorithms in stage 1 can be used as initial designs for performing simultaneous physics-based packing and routing optimization in stage 2 presented in Chapter 4. The proposed force-directed method presented in this chapter showed how it effectively generates feasible initial graphs with simultaneous packing and routing, especially when utilizing the FDLM-FETS approach. An important advantage of this method is that explicit interference detection constraints are not required between different geometric elements in a multi-component interconnected system. Furthermore, very efficient extensions of the shortest path planning A* algorithm were demonstrated that can perform complex interconnect routing for a fixed multi-component spatial topology. Depending on the choice of the sequence of the component to component connections, different feasible layouts and STs can be attained. In addition, greater success in attaining feasible layouts using SPAs can be possible by using finer grids. Finally, a systematic way of generating unique STs was presented. Based on direct and indirect connections between components, all 2D interconnect topological configurations can be obtained. The layout algorithms have been compared for their efficiency and success in generating feasible layouts, and a separate example study was illustrated to showcase their performance in complex geometric environments. Practical design guidelines are also outlined for engineers to utilize these methods for different problem types.

Chapter 4 presents stage 2 of the sequential design framework. Initially, the physics-based topology optimization method is discussed in detail. Continuous optimization with a gradient-based method combines both the component placement and interconnect routing problem together, in addition to using the physics-based models. The optimization problem formulation shown in this chapter supports both physics-based (overall system temperature, pressure head loss, etc.) and spatial (packing density, overall interconnection length, etc.) performance objectives. Geometric constraints of different

types such as device shapes, voids, 2D interference, pipe diameter, minimum distance between two components, maximum possible pipe length, pipe bend radius range, etc. can be employed as part of the optimization formulation. Physics-based constraints include maximum component temperature, maximum fluid flow rate, head loss, etc. Different kinds of boundary conditions were also illustrated that support both fixed temperature at boundaries or convection currents for cooling purposes. The 2-stage method has been illustrated on an unmanned aerial vehicle (UAV) test platform and an underhood automotive fuel cell (AFC) packaging system. Furthermore, a number of case studies were performed with different spatial layouts obtained from stage 1. Multi-loop optimization problems were also illustrated that can be extended to more complex system architectures in future work.

Chapter 5 concentrates on the 3D SPI2 design problem. As described earlier in Chapter 2, the 3D SPI2 topological space is very challenging to navigate in a systematic manner. In Chapter 5, a design framework to explore the 3D design space thoroughly has been identified. First, a given 3D system architecture was represented using spatial graphs. SGDs for a given system architecture are systematically enumerated upto a maximum crossing number. The SGDs are then categorized into equivalent Yamada classes if they share the same Yamada polynomial or a multiple of it by a factor A^n. The enumeration framework was implemented through different case studies and was very efficient in enumerating unique SGDs for small-scaled systems up to 10 components with 5 to 8 crossings. In addition, this framework was also very useful with large-scale systems through the implementation of a graph decomposition approach. Currently, 3D spatial topologies of systems with 10-15 components and approximately 20 interconnects can be easily enumerated using the framework proposed in this dissertation. However, most real-world SPI2 applications have a much larger number of components and interconnects. In such cases, it might not be a possible practice to exhaustively enumerate all of the 3D spatial topologies. For simplification, a group of components can be clustered together either based on functionality or their relative spatial proximity to reduce the number of nodes in the graph representation, enabling enumeration for large-scale problems that is not comprehensive, but still useful. This can help reduce explore the complexity of the 3D spatial design space of the system more efficiently. In addition, within the

small-sized clusters, enumeration can be performed. Several practical design strategies (for example, using design heuristics based on the SPI2 application) can be used to handle such large-scale systems. Data-driven strategies may also be effective in scaling to large SPI2 systems, enabling significant improvement even if global optimality is not assured. These aspects will be thoroughly investigated in future work.

6.2 Contributions

The dissertation has provided the following significant and novel intellectual contributions that address the research objectives (Sec. 1.7) outlined in Chapter 1:

1. Introduction of a newly-applied 2D force-directed layout method (FDLM) in Chapter 2 utilizing spring-force theory inspired by planar graph drawing algorithms [123,124]. This method implicitly satisfies interference constraints between the components, interconnects and other system elements. An extension to FDLM-FETS (Free-Expansion-Then-Shrinking) method improved the success rate of the original FDLM method by allowing more untangling of interconnects in the 2D layout.
2. Extension of the well-established A* (A-star) 2D shortest path planning algorithm (SPA) to perform sequential interconnect routing while ensuring non-interference. This can be utilized to perform complex interconnect routing both in 2D and 3D spatial layouts generation.
3. A systematic enumeration strategy for generating all unique 2D spatial topologies for a multi-component system, for given system architecture was presented in Chapter 2. This UST method serves as a baseline for design space coverage benchmarking for the 2D PR (packing and routing) problem.
4. The 2D geometric projection method (GPM) described in Chapter 2 was inspired by Norato et. *al* [83], which is an alternative to the well-established SIMP (Solid Isotropic Material with Penalization) method design parameterization [138], and is used for solving the combined physics-based PR optimization problem.
5. Physics-based objectives (bounding box volume, total head loss, and maximum component temperature, etc.) and constraints are incor-

porated into the optimization problem, in addition to geometric constraints that prevent interference between devices and interconnects. Both 1D lumped parameter and 2D finite element physics models are used within a single optimization problem to support physics-based evaluation.

6. The effectiveness and the computational efficiency of the SPI2 two-stage sequential design framework was demonstrated with the help of rigorous numerical experiments, and design method comparison studies over different system performance metrics. Numerical simulation experiments were also performed on challenge problems and case studies created with generic design representations that cover several engineering design and industry-relevant practical applications.
7. Novel 3D spatial topology enumeration algorithms were developed that greatly support the representation of complex 3D engineering systems inspired by mathematical braid and spatial graph theories. Classes of spatial topologies attained can be utilized for different engineering design applications.
8. Most importantly, for each design method presented in this dissertation, a set of practical design guidelines were listed. These discussions can help design engineers to implement the SPI2 design framework in expanding design capabilities across many relevant application domains.

It should be noted that this work is applicable only for systems that satisfy the assumptions made for developing these methods. However, the goal is to lay a foundation to aid design practitioners and engineers to gain deeper insights and to further develop the theory and methods required to tackle the most challenging real-world 3D packing and routing problems.

6.3 Future Work

The following future work items have been identified where significant research contributions can be made:

- **Future work in Force-directed layout method (FDLM):** The existing method only supports straight interconnect segments. Curved interconnected segments can be incorporated to provide more flexibility

for the layout to untangle itself. Furthermore, an interesting aspect as mentioned earlier would be to use a continuous force field along the boundaries of the components and the interconnect routing network instead of springs between components and interconnects. This will enforce no contact between any two geometric elements in the system. An extension of this concept to 3D SPI2 problem is commonly known as the *artificial potential field (APF) algorithm*. In an APF approach, the components and interconnects would be 3D objects and their surfaces would have a potential that induces repulsive forces, allowing systems elements to untangle until they attain a natural equilibrium state by balancing the 3D wall forces. Another aspect of application of FDLM is for large-scaled systems, where clusters of components can be grouped together as subsystems within a given 3D space and the FDLM can be applied to the set of clusters. Clusters can be categorized based on number of components, functionality of sub-systems (e.g., electrical subsystem, thermal subsystem, etc.).

- **2- and 3-D Shortest-path algorithms (SPA):** The A* and Dijkstra's SPAs are utilized here as an initial step to demonstrate the capability of SPAs in this dissertation. Existing literature contains more efficient and precise interconnect routing methods that can be leveraged in the future with very fine meshes. In addition, this dissertation only demonstrates a sequential method to use SPAs for creating interconnect paths between components. However, simultaneous interconnect routing methods, such as *2D multi-agent path finding* methods, exist which could help produce further improvements of 2D interconnected spatial layouts. Similarly, 3D MAPF methods are promising and are very efficient, and already well developed in areas such as 3D robotics path planning, aerial vehicle guidance, navigation, and control. Both grid-based and tree-based 3D routing algorithms could be utilized, each having their unique potential benefits.
- **Unique 2D Spatial Topology (UST) Enumeration:** The UST enumeration method for generating unique 2D spatial topologies can be further improved and scaled for very large-scale systems using machine learning techniques. The data obtained from enumeration can be utilized as described in Ref. [193] for the synthesis of larger-scale design problems where exhaustive enumeration may become practically

intractable. Another future work item could also include identification of patterns or number series of how many different USTs can be obtained with increasing number of components having identical port valencies.

- **Component rotation:** 2D packing and routing while allowing for rotation of geometric elements is another aspect of future work. This will provide more design freedom for optimization to attain better solutions.
- **Identifying more SPI2 related design research areas:** As already noted at the end of Chapter 2, there are still several SPI2 research areas that need to be investigated thoroughly, such as unified geometric parameterization methods, 3D physics-based topology optimization, machine learning techniques, methods to estimate product life-cycle costs, reliability estimation methods, and other topics.
- **Further investigation into 2D physics-based geometric projection:** GPM in 2D SPI2 problem can include more physics aspects such as electrical effects, material stress, and thermoelastic models. Multi-domain modeling and analysis is important to attain more realistic optimal solutions. Other future work items would be to incorporate the concept of irregularly-shaped bounding boxes and complex component shapes, multiple flow loops, pumps, fluid storage tanks, heat sinks, etc., to accommodate more complex system architectures.
- **Transition to 3D geometric projection:** A core challenge with extending this SPI2 effort to 3D PR problems involves a transition to 3D GPMs, which use plates as geometric primitives, whereas 2D GPMs use bars. Optimization should be performed with respect to plate location, shape, and orientation parameters as opposed to discretized design representations (e.g., element-wise densities or node-wise level set values of conventional topology optimization approaches).
- **Chapter 5 future work:** Future work based on Chapter 5 includes adding more geometric features to these spatial graph embeddings, such as representing nodes with geometric shapes and ports. A deeper investigation of braid-based representations of interconnect networks is also anticipated. As the system becomes larger, evaluating Yamada polynomials for many SGDs is very time-consuming. This can be overcome by implementing a mix of Reidemeister moves to eliminate isotopic diagrams quickly to produce a smaller set of diagrams that require

Yamada calculations. Other application aspects include utilizing the unique spatial topologies obtained here as starting points for physics-based packing and routing optimization of 3D systems. Furthermore, research areas that can benefit from SGD representations are 3D pipe routing, topological 3D path planning for robotic operations, aerial drone navigation, generation of new automotive cooling system configurations, 3D integrated circuit interconnect technology, and many others. This initial work serves as a foundation to bridging the gap between engineering design and mathematical low-dimensional topology. There are many interesting aspects which are yet to be explored and can have a great impact when applied to classical engineering design problems.

- **Extension of 3D spatial topology enumeration framework to industry applications:** Although case studies were presented in Chapter 5 to demonstrate the effectiveness of the spatial graph enumeration framework, it was not demonstrated on real-world industry applications. Future work on this topic will include studying 3D spatial topologies of examples such as automotive cooling systems, wind-turbines, aero-engines, etc.
- **Computational design optimization software tools:** The SPI2 design automation methods could be developed into commercial packing and routing optimization tools in collaboration with engineers/practical personnel to enable industry adoption of SPI2 design automation, helping to realize significant societal impact of enhanced SPI2 system design capabilities.
- The SPI2 design framework in this dissertation handles the discrete and continuous elements of the problem in two separate stages. Stage 1 explores the unique, spatially-feasible discrete design options of the system via enumeration, while stage 2 optimizes the continuous design parameters of the layouts to meet specific performance objectives within geometric and physics-based constraints. However, the coverage of the design space using this framework might be inefficient when considering very large-scaled systems. To develop a simultaneous optimization method to combine these two stages when using gradient-based methods, we need a continuous relaxation technique that can generate smooth gradients and is compatible with physics-based finite

element analysis. This way the components and interconnects might pass through each other without disrupting the physics analysis. Implementing this approach, however, will require a significant effort, and should be addressed as a topic for future work to compare it against the proposed sequential design framework. Furthermore, in stage 1, efficient exploration techniques, such as Latin hypercube sampling (LHS), coupled with strategic enumeration can be utilized to search the discrete design space in a more robust manner. This, however, requires deeper investigation into estimating the design space and relationships between the discrete and continuous design parameters.

- Lastly, a subset of the best 2D or 3D SPI2 designs identified as superior based on comparative performance analysis can be validated using simple bench-top experiments. This can both provide feedback to fine-tune the design framework and improve the *technology readiness level* (TRL) of the SPI2 design automation design methods.
- One long term goal is to provide engineers practical design guidelines based on this work to support the development of unprecedented SPI2 systems that enable future applications.

The future work items listed are only a sampling of the vast research scope both 2D and 3D SPI2 domains encompass. It is hoped that the fundamental breakthroughs presented here, such as new mathematical SPI2 design representations and their incorporation into practical design methods, catalyzes a surge of research activity in this domain.

REFERENCES

[1] R. Kasper and N. Borchardt, "Boosting power density of electric machines by combining two different winding types," *IFAC-PapersOnLine*, vol. 49, no. 21, pp. 322 – 329, 2016, 7th IFAC Symposium on Mechatronic Systems MECHATRONICS 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896316321668

[2] P. Ramesh and N. C. Lenin, "High power density electrical machines for electric vehicles—comprehensive review based on material technology," *IEEE Transactions on Magnetics*, vol. 55, no. 11, pp. 1–21, Nov 2019, doi:10.1109/TMAG.2019.2929145.

[3] M. Rajasekhar, J. Perumal, S. Rawte, and N. Nepal, "Integration and packaging for vehicle electrification," in *Symposium on International Automotive Technology 2015*. SAE International, jan 2015. [Online]. Available: https://doi.org/10.4271/2015-26-0115

[4] G. Su, L. Tang, C. Ayers, and R. Wiles, "An inverter packaging scheme for an integrated segmented traction drive system," in *2013 IEEE Energy Conversion Congress and Exposition*, Sep. 2013, doi:10.1109/ECCE.2013.6647064. pp. 2799–2804.

[5] C. Van der Velden, C. Bil, X. Yu, and A. Smith, "An intelligent system for automatic layout routing in aerospace design," *Innovations in Systems and Software Engineering*, vol. 3, no. 2, pp. 117–128, June 2007, doi:10.1007/s11334-007-0021-4. [Online]. Available: https://doi.org/10.1007/s11334-007-0021-4

[6] U. of Oxford, "Best way to reduce emissions is to make cars smaller," Jan. 2010, https://www.sciencedaily.com/releases/2010/01/100116102818.htm.

[7] P. Patel, "The battery design smarts behind rolls royce's ultrafast electric airplane," Jan. 2020, https://spectrum.ieee.org/energywise/energy/batteries-storage/the-battery-innovations-behind-rolls-royces-ultrafast-electric-airplane.

[8] R. L. Nason and M. J. Heldmann, "Performance characteristics of the space station avionics air cooling package," in *International Conference On Environmental Systems*. SAE International, jul 1996. [Online]. Available: https://doi.org/10.4271/961352

[9] D. Beosing, "Packaging innovations for medical wearables," Aug. 2018, https://blog.samtec.com/post/packaging-innovations-for-medical-wearables.

[10] T. Hollingshead, "Compact mechanisms show promise for medical devices," Feb. 2019.

[11] D. Heussner, "Wearable technologies present packaging challenges," Mar. 2014, https://www.electronicdesign.com/technologies/digital-ics/article/21799376/wearable-technologies-present-packaging-challenges.

[12] R. Mehta and M. Hadley, "Vehicle spaciousness and packaging efficiency," apr 2014. [Online]. Available: https://doi.org/10.4271/2014-01-0348

[13] C. Howard, "Avionics and military electronics thermal management challenges are sparking innovative solutions to keep these systems cool," Nov. 2010, https://www.intelligent-aerospace.com/avionics/article/16543961/avionics-and-military-electronics-thermal-management-challenges-are-sparking-innovative-solutions-to-keep-these-systems-cool.

[14] C. Howard, "Power and thermal management considerations move to the forefront of aerospace and defense electronic systems," Oct. 2011, https://www.militaryaerospace.com/trusted-computing/article/16716997/power-and-thermal-management-considerations-move-to-the-forefront-of-aerospace-and-defense-electronic-systems.

[15] J. Bauer, *Leadless carrier applications for avionics packaging*, 1977. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.1977-1487

[16] F. Poradish, "High density modular avionics packaging," in *Digital Avionics Systems Conference*, 1984. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.1984-2749

[17] J. Kanz, "New directions in aerospace packaging," in *5th Computers in Aerospace Conference*, 1985. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.1985-6033

[18] J. Seals, "Putting ten pounds of avionics in a one pound package (can we do it again?)," in *8th Computing in Aerospace Conference*, 1991. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.1991-3766

[19] R. Mayer, "Vehicle/manipulator/packaging interaction - a synergistic approach to large erectable space system design," in *18th Structural Dynamics and Materials Conference*, 1977. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.1977-394

[20] J. Huang and L. Gong, "A knowledge based engineering framework for rapid prototyping in vehicle packaging system," in *Seoul 2000 FISITA World Automotive Congress*. Society of Automotive Engineers of Korea, jun 2000.

[21] I. P. Abramov, R. K. Sharipov, A. I. Skoog, and N. Herber, "Space suit life support system packaging factors," in *International Conference On Environmental Systems*. SAE International, jun 1994. [Online]. Available: https://doi.org/10.4271/941380

[22] R. Howe, C. Diep, B. Barnett, M. Rouen, G. Thomas, and J. Kobus, "Advanced space suit portable life support subsystem packaging design," in *International Conference On Environmental Systems*. SAE International, jul 2006. [Online]. Available: https://doi.org/10.4271/2006-01-2202

[23] N. Sharma and M. Kaur, "A survey of vlsi techniques for power optimization and estimation of optimization," 2014.

[24] S. Devadas and S. Malik, "A survey of optimization techniques targeting low power vlsi circuits," *32nd Design Automation Conference*, pp. 242–247, 1995.

[25] X. Tang, R. Tian, and M. D. F. Wong, "Optimal redistribution of white space for wire length minimization," in *ASP-DAC*, 2005.

[26] S. R. T. Peddada, L. E. Zeidner, K. A. James, and J. T. Allison, "An introduction to 3d spi2 (spatial packaging of interconnected systems with physics interactions) design problems: A review of related work, existing gaps, challenges, and opportunities," in *ASME 2021 International Design Engineering Technical Conferences*, Virtual, Online, Aug. 2021.

[27] J. Cagan, D. Degentesh, and S. Yin, "A simulated annealing-based algorithm using hierarchical models for general three-dimensional component layout," *Computer-Aided Design*, vol. 30, no. 10, pp. 781–790, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010448598000360

[28] H. Dong, P. Guarneri, and G. Fadel, "Bi-level Approach to Vehicle Component Layout With Shape Morphing," *Journal of Mechanical Design*, vol. 133, no. 4, 05 2011, 041008. [Online]. Available: https://doi.org/10.1115/1.4003916

[29] M. Schafer and T. Lengauer, "Automated Layout Generation and Wiring Area Estimation for 3D Electronic Modules ," *Journal of Mechanical Design*, vol. 123, no. 3, pp. 330–336, 05 1999, doi:10.1115/1.1371478. [Online]. Available: https://doi.org/10.1115/1.1371478

[30] Natsuko Yano, Takashi Morinaga, and Tsutomu Saito, "Packing optimization for cargo containers," in *2008 SICE Annual Conference*, Aug 2008, doi:10.1109/SICE.2008.4655264. pp. 3479–3482.

[31] N. Bansal, A. Lodi, and M. Sviridenko, "A tale of two dimensional bin packing," in *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, Oct 2005, doi:10.1109/SFCS.2005.10. pp. 657–666.

[32] K. A. Abdel-Malek, H. J. Yeh, and N. Maropis, "Determining interference between pairs of solids defined constructively in computer animation," *Engineering with Computers*, vol. 14, no. 1, pp. 48–58, Mar. 1998, doi:doi.org/10.1007/BF01198974. [Online]. Available: https://doi.org/10.1007/BF01198974

[33] A. Panesar, D. Brackett, I. Ashcroft, R. Wildman, and R. Hague, "Design Framework for Multifunctional Additive Manufacturing: Placement and Routing of Three-Dimensional Printed Circuit Volumes," *Journal of Mechanical Design*, vol. 137, no. 11, 10 2015, 111414. [Online]. Available: https://doi.org/10.1115/1.4030996

[34] S. Yin, J. Cagan, and P. Hodges, "Layout Optimization of Shapeable Components With Extended Pattern Search Applied to Transmission Design ," *Journal of Mechanical Design*, vol. 126, no. 1, pp. 188–191, 03 2004, doi:10.1115/1.1637663. [Online]. Available: https://doi.org/10.1115/1.1637663

[35] S. Jain and H. C. Gea, "Two-dimensional packing problems using genetic algorithms," *Engineering with Computers*, vol. 14, no. 3, pp. 206–213, 1998. [Online]. Available: https://doi.org/10.1007/BF01215974

[36] E. López-Camacho, G. Ochoa, H. Terashima-Marín, and E. K. Burke, "An effective heuristic for the two-dimensional irregular bin packing problem," *Annals of Operations Research*, vol. 206, no. 1, pp. 241–264, 2013. [Online]. Available: https://doi.org/10.1007/s10479-013-1341-4

[37] R. Sridhar, D. Chandrasekaran, C. Sriramya, and T. Page, "Optimization of heterogeneous bin packing using adaptive genetic algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 183, p. 012026, 03 2017.

[38] R. L. Rao and S. S. Iyengar, "Bin-packing by simulated annealing," vol. 27, no. 5, pp. 71–82, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0898122194900779

[39] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959. [Online]. Available: https://doi.org/10.1007/BF01386390

[40] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[41] C. Y. Lee, "An algorithm for path connections and its applications," *IRE Transactions on Electronic Computers*, vol. EC-10, no. 3, pp. 346–365, 1961.

[42] T. Mitsuta, Y. Kobayashi, Y. Wada, T. Kiguchi, and T. Yoshinaga, "A knowledge-based approach to routing problems in industrial plant design," in *6th Internation Workshop Vol. 1 on Expert Systems & Their Applications*. FRA: Agence de l'Informatique, 1987, p. 237–256.

[43] N. Wang, J. Wan, G. Gomez-Levi, V. Kiridena, S. Sieczka, and D. Pulliam, "An integrated design and appraisal system for vehicle interior packaging," 2007.

[44] I. T. O. TERUAKI, "A genetic algorithm approach to piping route path planning," *Journal of Intelligent Manufacturing*, vol. 10, no. 1, pp. 103–114, 1999. [Online]. Available: https://doi.org/10.1023/A:1008924832167

[45] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[46] W.-Y. Jiang, Y. Lin, M. Chen, and Y.-Y. Yu, "A co-evolutionary improved multi-ant colony optimization for ship multiple and branch pipe route design," *Ocean Engineering*, vol. 102, pp. 63–70, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0029801815001031

[47] W.-y. Jiang, Y. Lin, M. Chen, and Y.-y. Yu, "An ant colony optimization-genetic algorithm approach for ship pipe route design," *International Shipbuilding Progress*, vol. 61, no. 3-4, pp. 163–183, 2014.

[48] L. Qiang and W. Chengen, "A discrete particle swarm optimization algorithm for rectilinear branch pipe routing," *Assembly Automation*, vol. 31, no. 4, pp. 363–368, Jan. 2011. [Online]. Available: https://doi.org/10.1108/01445151111172952

[49] X.-Y. Shao, X.-Z. Chu, H.-B. Qiu, L. Gao, and J. Yan, "An expert system using rough sets theory for aided conceptual design of ship's engine room automation," *Expert Systems with Applications*, vol. 36, no. 2, Part 2, pp. 3223–3233, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417408000948

[50] E. E. S. Calixto, P. G. Bordeira, H. T. Calazans, C. A. C. Tavares, M. T. D. Rodriguez, R. M. de Brito Alves, C. A. O. do Nascimento, and E. C. Biscaia, "Plant design project automation using an automatic pipe routing routine," in *Computer Aided Chemical Engineering*. Elsevier, 2009, vol. 27, pp. 807–812. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570794609703554

[51] J.-H. Park and R. L. Storch, "Pipe-routing algorithm development: case study of a ship engine room design," *Expert Systems with Applications*, vol. 23, no. 3, pp. 299–309, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417402000490

[52] C.-K. Koh and P. H. Madden, "Manhattan or non-manhattan? a study of alternative vlsi routing architectures," in *Proceedings of the 10th Great Lakes Symposium on VLSI*, ser. GLSVLSI '00. New York, NY, USA: Association for Computing Machinery, 2000. [Online]. Available: https://doi.org/10.1145/330855.330961 p. 47–52.

[53] C. Van der Velden, C. Bil, X. Yu, and A. Smith, "An intelligent system for automatic layout routing in aerospace design," *Innovations in Systems and Software Engineering*, vol. 3, no. 2, pp. 117 – 128, 2007, doi:10.1007/s11334-007-0021-4. [Online]. Available: http://dx.doi.org/10.1007/s11334-007-0021-4

[54] J.-H. Park and R. Storch, "Pipe-routing algorithm development: case study of a ship engine room design," *Expert Syst. Appl. (UK)*, vol. 23, no. 3, pp. 299 – 309, 2002, doi:10.1016/S0957-4174(02)00049-0. [Online]. Available: http://dx.doi.org/10.1016/S0957-4174(02)00049-0

[55] R. Guirardello and R. E. Swaney, "Optimization of process plant layout with pipe routing," *Computers and Chemical Engineering*, vol. 30, no. 1, pp. 99–114, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098135405001985

[56] C. Liu, "Optimal design of high-rise building wiring based on ant colony optimization," *Cluster Computing*, pp. 1 – 8, 2018. [Online]. Available: http://dx.doi.org/10.1007/s10586-018-2195-y

[57] V. Betz and J. Rose, "Vpr: a new packing, placement and routing tool for fpga research," in *Field-Programmable Logic and Applications*, W. Luk, P. Y. K. Cheung, and M. Glesner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 213–222.

[58] J. Tisdale, Z. Kim, and J. K. Hedrick, "Autonomous uav path planning and estimation," *IEEE Robotics Automation Magazine*, vol. 16, no. 2, pp. 35–42, 2009.

[59] G. E. Jan, K. Yin Chang, and I. Parberry, "Optimal path planning for mobile robot navigation," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 4, pp. 451–460, 2008.

[60] M. D. Landon and R. J. Balling, "Optimal Packaging of Complex Parametric Solids According to Mass Property Criteria," *Journal of Mechanical Design*, vol. 116, no. 2, pp. 375–381, 06 1994, doi:10.1115/1.2919389. [Online]. Available: https://doi.org/10.1115/1.2919389

[61] S. Szykman and J. Cagan, "Constrained Three-Dimensional Component Layout Using Simulated Annealing," *Journal of Mechanical Design*, vol. 119, no. 1, pp. 28–35, 03 1997, doi:10.1115/1.2828785. [Online]. Available: https://doi.org/10.1115/1.2828785

[62] S. Szykman, J. Cagan, and P. Weisser, "An Integrated Approach to Optimal Three Dimensional Layout and Routing," *Journal of Mechanical Design*, vol. 120, no. 3, pp. 510–512, 09 1998. [Online]. Available: https://doi.org/10.1115/1.2829180

[63] C. Aladahalli, J. Cagan, and K. Shimada, "Objective Function Effect Based Pattern Search—Theoretical Framework Inspired by 3D Component Layout," *Journal of Mechanical Design*, vol. 129, no. 3, pp. 243–254, 03 2006, doi:10.1115/1.2406095. [Online]. Available: https://doi.org/10.1115/1.2406095

[64] S. Yin and J. Cagan, "An Extended Pattern Search Algorithm for Three-Dimensional Component Layout ," *Journal of Mechanical Design*, vol. 122, no. 1, pp. 102–108, 01 2000. [Online]. Available: https://doi.org/10.1115/1.533550

[65] T. Ren, Z.-L. Zhu, G. Dimirovski, Z.-H. Gao, X.-H. Sun, and H. Yu, "A new pipe routing method for aero-engines based on genetic algorithm," *Proceedings of the Institution of Mechanical Engineers, Part G (Journal of Aerospace Engineering)*, vol. 228, no. 3, pp. 424 – 34, 2014, doi:10.1177/0954410012474134. [Online]. Available: http://dx.doi.org/10.1177/0954410012474134

[66] Y. Qu, D. Jiang, G. Gao, and Y. Huo, "Pipe routing approach for aircraft engines based on ant colony optimization," *Journal of Aerospace Engineering*, vol. 29, no. 3, p. 04015057, 2016, doi:10.1061/(ASCE)AS.1943-5525.0000543. [Online]. Available: http://dx.doi.org/10.1061/(ASCE)AS.1943-5525.0000543

[67] M. Gulić and D. Jakobović, "Evolution of vehicle routing problem heuristics with genetic programming," in *2013 36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2013, pp. 988–992.

[68] O. Sigmund, "A 99 line topology optimization code written in matlab," *Structural and Multidisciplinary Optimization*, vol. 21, no. 2, pp. 120–127, 04 2001. [Online]. Available: https://doi.org/10.1007/s001580050176

[69] H. Kazemi, A. Vaziri, and J. A. Norato, "Topology Optimization of Structures Made of Discrete Geometric Components With Different Materials," *Journal of Mechanical Design*, vol. 140, no. 11, 09 2018, 111401. [Online]. Available: https://doi.org/10.1115/1.4040624

[70] A. Iga, S. Nishiwaki, K. Izui, and M. Yoshimura, "Topology optimization for thermal conductors considering design-dependent effects, including heat conduction and convection," *International Journal of Heat and Mass Transfer*, vol. 52, no. 11-12, pp. 2721 – 2732, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.ijheatmasstransfer.2008.12.013

[71] J. Dirker and J. P. Meyer, "Topology optimization for an internal heat-conduction cooling scheme in a square domain for high heat flux applications," *Journal of Heat Transfer*, vol. 135, no. 11, 2013. [Online]. Available: http://dx.doi.org/10.1115/1.4024615

[72] N. de Kruijf, S. Zhou, Q. Li, and Y.-W. Mai, "Topological design of structures and composite materials with multiobjectives," *International Journal of Solids and Structures*, vol. 44, no. 22-23, pp. 7092 – 109, 2007. [Online]. Available: http://dx.doi.org/10.1016/j.ijsolstr.2007.03.028

[73] A. Takezawa, G. H. Yoon, S. H. Jeong, M. Kobashi, and M. Kitamura, "Structural topology optimization with strength and heat conduction constraints," *Computer Methods in Applied Mechanics and Engineering*, vol. 276, pp. 341 – 61, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.cma.2014.04.003

[74] Z. Kang and K. A. James, "Multimaterial topology design for optimal elastic and thermal response with material-specific temperature constraints," *International Journal for Numerical Methods in Engineering*, vol. 117, no. 10, pp. 1019–1037, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.5989

[75] K. James, G. Kennedy, and J. Martins, "Concurrent aerostructural topology optimization of a wing box," *Computers &; Structures*, vol. 134, pp. 1 – 17, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.compstruc.2013.12.007

[76] P. Dunning, B. Stanford, and H. Kim, "Coupled aerostructural topology optimization using a level set method for 3d aircraft wings," *Structural and Multidisciplinary Optimization*, vol. 51, no. 5, pp. 1113 – 32, 2015. [Online]. Available: http://dx.doi.org/10.1007/s00158-014-1200-1

[77] E. Oktay, H. Akay, and O. Merttopcuoglu, "Parallelized structural topology optimization and cfd coupling for design of aircraft wing structures," *Comput. Fluids (UK)*, vol. 49, no. 1, pp. 141 – 5, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.compfluid.2011.05.005

[78] J. Zhu, W. Zhang, P. Beckers, Y. Chen, and Z. Guo, "Simultaneous design of components layout and supporting structures using coupled shape and topology optimization technique," *Structural and Multidisciplinary Optimization*, vol. 36, no. 1, pp. 29 – 41, 2008. [Online]. Available: http://dx.doi.org/10.1007/s00158-007-0155-x

[79] J.-H. Zhu, W.-J. Guo, W.-H. Zhang, and T. Liu, "Integrated layout and topology optimization design of multi-frame and multi-component fuselage structure systems," *Structural and Multidisciplinary Optimization*, vol. 56, no. 1, pp. 21 – 45, 2017. [Online]. Available: http://dx.doi.org/10.1007/s00158-016-1645-5

[80] T. Zegard and G. H. Paulino, "Bridging topology optimization and additive manufacturing," *Structural and Multidisciplinary Optimization*, vol. 53, no. 1, pp. 175 – 192, 2016. [Online]. Available: http://dx.doi.org/10.1007/s00158-015-1274-4

[81] G. G. Tejani, V. J. Savsani, V. K. Patel, and P. V. Savsani, "Size, shape, and topology optimization of planar and space trusses using mutation-based improved metaheuristics," *Journal of Computational Design and Engineering*, vol. 5, no. 2, pp. 198 – 214, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2288430017300763

[82] X. S. Zhang, G. H. Paulino, and A. S. Ramos, "Multi-material topology optimization with multiple volume constraints: a general approach applied to ground structures with material nonlinearity," *Structural and Multidisciplinary Optimization*, vol. 57, no. 1, pp. 161 – 182, 2018. [Online]. Available: http://dx.doi.org/10.1007/s00158-017-1768-3

[83] J. Norato, B. Bell, and D. Tortorelli, "A geometry projection method for continuum-based topology optimization with discrete elements," *Computer Methods in Applied Mechanics and Engineering*, vol. 293, pp. 306 – 27, 2015, doi:10.1016/j.cma.2015.05.005. [Online]. Available: http://dx.doi.org/10.1016/j.cma.2015.05.005

[84] S. Zhang, J. A. Norato, A. L. Gain, and N. Lyu, "A geometry projection method for the topology optimization of plate structures," *Structural and Multidisciplinary Optimization*, vol. 54, no. 5, pp. 1173 – 1190, 2016. [Online]. Available: http://dx.doi.org/10.1007/s00158-016-1466-6

[85] A. E. Patterson and J. T. Allison, "Generation and mapping of minimally-restrictive manufacturability constraints for mechanical design problems," in *ASME 2019 International Design Engineering Technical Conferences*, no. DETC2019-97386, Anaheim, CA, USA, Aug. 2019.

[86] S. R. T. Peddada, K. A. James, and J. T. Allison, "A novel two-stage design framework for 2d spatial packing of interconnected components," in *ASME 2020 International Design Engineering Technical Conferences.*, no. IDETC2020-22695, Virtual, Online, 17-19 August 2020, [Invited to ASME *Journal of Mechanical Design* 2020 Special Issue].

[87] A. Jessee, S. R. T. Peddada, D. J. Lohan, J. T. Allison, and K. A. James, "Simultaneous Packing and Routing Optimization Using Geometric Projection," *Journal of Mechanical Design*, vol. 142, no. 11, 05 2020. [Online]. Available: https://doi.org/10.1115/1.4046809

[88] J. Hoste, M. Thistlethwaite, and J. Weeks, "The first 1,701,936 knots," *Math. Intelligencer*, vol. 20, no. 4, pp. 33–48, 1998. [Online]. Available: https://doi.org/10.1007/BF03025227

[89] C. Liang and K. Mislow, "Classification of topologically chiral molecules," *Journal of Mathematical Chemistry*, vol. 15, no. 1, pp. 245–260, 1994. [Online]. Available: https://doi.org/10.1007/BF01277563

[90] E. Flapan and W. Fletcher, "Intrinsic chirality of multipartite graphs," *Journal of Mathematical Chemistry*, vol. 51, no. 7, pp. 1853–1863, 2013. [Online]. Available: https://doi.org/10.1007/s10910-013-0187-y

[91] E. R. I. C. A. FLAPAN, "Rigidity of graph symmetries in the 3-sphere," *J. Knot Theory Ramifications*, vol. 04, no. 03, pp. 373–388, Sep. 1995. [Online]. Available: https://doi.org/10.1142/S0218216595000181

[92] B. Mellor, "Invariants of spatial graphs," *arXiv: Geometric Topology*, 2018.

[93] T. Fleming and B. Mellor, "An introduction to virtual spatial graph theory," *arXiv: Geometric Topology*, 2006.

[94] G. Rapenne, J. Crassous, L. E. Echegoyen, L. Echegoyen, E. Flapan, and F. Diederich, "Regioselective one-step synthesis and topological chirality of trans-3, trans-3,trans-3 and e,e,e [60]fullerene-cyclotriveratrylene tris-adducts: Discussion on a topological meso-form," *HCA*, vol. 83, no. 6, pp. 1209–1223, June 2000. [Online]. Available: https://doi.org/10.1002/1522-2675(20000607)83:6⟨1209::AID-HLCA1209⟩3.0.CO;2-Y

[95] K. Murasugi and K. Murasugi, "The theory of braids," in *Knot Theory and Its Applications*. Boston, MA: Birkhäuser Boston, 1996, pp. 197–216. [Online]. Available: https://doi.org/10.1007/978-0-8176-4719-3_11

[96] E. Flapan, A. He, and H. Wong, "Topological descriptions of protein folding," *Proc Natl Acad Sci USA*, vol. 116, no. 19, p. 9360, May 2019. [Online]. Available: http://www.pnas.org/content/116/19/9360.abstract

[97] C. I. Mavrogiannis and R. A. Knepper, "Multi-agent path topology in support of socially competent navigation planning," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 338–356, June 2018. [Online]. Available: https://doi.org/10.1177/0278364918781016

[98] C. I. Mavrogiannis and R. A. Knepper, "Decentralized multi-agent navigation planning with braids," in *Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics*, K. Goldberg, P. Abbeel, K. Bekris, and L. Miller, Eds. Cham: Springer International Publishing, 2020, pp. 880–895. [Online]. Available: https://doi.org/10.1007/978-3-030-43089-4_56

[99] S. Bhattacharya, M. Likhachev, and V. Kumar, "Search-based path planning with homotopy class constraints in 3d," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, ser. AAAI'12. AAAI Press, 2012, p. 2097–2099.

[100] J. Hershberger and J. Snoeyink, "Computing minimum length paths of a given homotopy class," in *Algorithms and Data Structures*, F. Dehne, J.-R. Sack, and N. Santoro, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 331–342.

[101] E. Schmitzberger, J. L. Bouchet, M. Dufaut, D. Wolf, and R. Husson, "Capture of homotopy classes with probabilistic road map," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 2002, pp. 2317–2322 vol.3.

[102] D. Demyen and M. Buro, "Efficient triangulation-based pathfinding," in *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, ser. AAAI'06. AAAI Press, 2006, p. 942–947.

[103] S. R. T. Peddada, D. R. Herber, H. C. Pangborn, A. G. Alleyne, and J. T. Allison, "Optimal Flow Control and Single Split Architecture Exploration for Fluid-Based Thermal Management," *Journal of Mechanical Design*, vol. 141, no. 8, 04 2019, doi:10.1115/1.4043203. [Online]. Available: https://doi.org/10.1115/1.4043203

[104] D. R. Herber, T. Guo, and J. T. Allison, "Enumeration of architectures with perfect matchings," *J. Mech. Des*, vol. 139, no. 5, Apr. 2017. [Online]. Available: https://doi.org/10.1115/1.4036132

[105] D. R. Herber and J. T. Allison, "A problem class with combined architecture, plant, and control design applied to vehicle suspensions," *J. Mech. Des*, vol. 141, no. 10, May 2019. [Online]. Available: https://doi.org/10.1115/1.4043312

[106] S. R. T. Peddada, S. B. Rodriguez, K. A. James, and J. T. Allison, "Automated layout generation methods for 2d spatial packing," in *ASME 2020 International Design Engineering Technical Conferences.*, no. IDETC2020-22627, Virtual, Online, 17-19 August 2020.

[107] A. E. Bayrak, Y. Ren, and P. Y. Papalambros, "Topology generation for hybrid electric vehicle architecture design," *J. Mech. Des*, vol. 138, no. 8, June 2016. [Online]. Available: https://doi.org/10.1115/1.4033656

[108] X. Ai and S. Anderson, "An electro-mechanical infinitely variable transmission for hybrid electric vehicles," 2005.

[109] M. I. Ramdan and K. A. Stelson, "Optimal design of a power-split hybrid hydraulic bus," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 230, no. 12, pp. 1699–1718, Jan. 2016. [Online]. Available: https://doi.org/10.1177/0954407015621817

[110] R. Hongpipatsak and M. Wongsaisuwan, "Motion interference detection and identification in mobile robots using driving motor currents," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2017, doi:10.1109/IECON.2017.8216541. pp. 3203–3208.

[111] D. C. H. Yang, "Collision-free path planning by using nonperiodic b-spline curves," *J. Mech. Des*, vol. 115, no. 3, pp. 679–684, Sep. 1993, doi:10.1115/1.2919245. [Online]. Available: https://doi.org/10.1115/1.2919245

[112] M. Shinya and M.-C. Forgue, "Interference detection through rasterization," *The Journal of Visualization and Computer Animation*, vol. 2, no. 4, pp. 132–134, 1991, doi:10.1002/vis.4340020408. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/vis.4340020408

[113] M. C. Lin and D. Manocha, "Interference detection between curved objects for computer animation," in *Models and Techniques in Computer Animation*, N. M. Thalmann and D. Thalmann, Eds. Tokyo: Springer Japan, 1993, pp. 43–57.

[114] Y. Cai, Z. Fan, H. Wan, S. Gao, B. Lu, and K. T. Lim, "Hardware-accelerated collision detection for 3d virtual reality gaming," *Simulation & Gaming*, vol. 37, no. 4, pp. 476–490, 2006, doi:10.1177/1046878106293678. [Online]. Available: https://doi.org/10.1177/1046878106293678

[115] N. Gageik, P. Benz, and S. Montenegro, "Obstacle detection and collision avoidance for a uav with complementary low-cost sensors," *IEEE Access*, vol. 3, pp. 599–609, 2015, doi:10.1109/ACCESS.2015.2432455.

[116] P. Vrba, V. Mařík, L. Přeučil, M. Kulich, and D. Šišlák, "Collision avoidance algorithms: Multi-agent approach," in *Holonic and Multi-Agent Systems for Manufacturing*, V. Mařík, V. Vyatkin, and A. W. Colombo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 348–360.

[117] L. Epstein, "Two-dimensional online bin packing with rotation," *Theoretical Computer Science*, vol. 411, no. 31, pp. 2899 – 2911, 2010, doi:10.1016/j.tcs.2010.04.021. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304397510002161

[118] M. Mongeau and C. Bes, "Optimization of aircraft container loading," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 1, pp. 140–150, Jan 2003, doi:10.1109/TAES.2003.1188899.

[119] N. Beldiceanu, M. Carlsson, and E. Poder, "New filtering for the cumulative constraint in the context of non-overlapping rectangles," in *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, L. Perron and M. A. Trick, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 21–35.

[120] M. Han and Y. Wang, "A survey for vehicle routing problems and its derivatives," *IOP Conference Series: Materials Science and Engineering*, vol. 452, p. 042024, dec 2018, doi:10.1088/1757-899x/452/4/042024. [Online]. Available: https://doi.org/10.1088%2F1757-899x%2F452%2F4%2F042024

[121] Z. Zhang, W. Jigang, and X. Duan, "Practical algorithm for shortest path on transportation network," in *2010 International Conference on Computer and Information Application*, Dec 2010, doi:10.1109/ICCIA.2010.6141534. pp. 48–51.

[122] A. Sakti, L. Zeidner, T. Hadzic, B. S. Rock, and G. Quartarone, "Constraint programming approach for spatial packaging problem," in *Integration of AI and OR Techniques in Constraint Programming*, C.-G. Quimper, Ed. Cham: Springer International Publishing, 2016, pp. 319–328.

[123] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Information Processing Letters*, vol. 31, no. 1, pp. 7 – 15, 1989, doi:10.1016/0020-0190(89)90102-6. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0020019089901026

[124] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Software: Practice and Experience*, vol. 21, no. 11, pp. 1129–1164, 1991, doi:10.1002/spe.4380211102. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.4380211102

[125] H. Gibson, J. Faith, and P. Vickers, "A survey of two-dimensional graph layout techniques for information visualisation," *Information Visualization*, vol. 12, pp. 324 – 357, 2013.

[126] D. L. Hansen, B. Shneiderman, M. A. Smith, and I. Himelboim, "Chapter 4 - installation, orientation, and layout," in *Analyzing Social Media Networks with NodeXL (Second Edition)*, second edition ed., D. L. Hansen, B. Shneiderman, M. A. Smith, and I. Himelboim, Eds. Morgan Kaufmann, 2020, pp. 55 – 66, doi:10.1016/B978-0-12-817756-3.00004-2. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128177563000042

[127] O. Garcia, C. Saveanu, M. S. Cline, M. Fromont-Racine, A. Jacquier, B. Schwikowski, and T. Aittokallio, "Golorize: a cytoscape plug-in for network visualization with gene ontology-based layout and coloring," *Bioinformatics*, vol. 23 3, pp. 394–6, 2007.

[128] H. Heberle, M. F. Carazzolle, G. P. Telles, G. V. Meirelles, and R. Minghim, "Cellnetvis: a web tool for visualization of biological networks using force-directed layout constrained by cellular components," *BMC Bioinformatics*, vol. 18, no. 10, p. 395, Sep. 2017, doi:10.1186/s12859-017-1787-5. [Online]. Available: https://doi.org/10.1186/s12859-017-1787-5

[129] W. T. Tutte, "How to draw a graph," *Proceedings of the London Mathematical Society*, vol. s3-13, no. 1, pp. 743–767, 1963, doi:10.1112/plms/s3-13.1.743. [Online]. Available: https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/plms/s3-13.1.743

[130] I. Chaari, A. Koubaa, H. Bennaceur, A. Ammar, M. Alajlan, and H. Youssef, "Design and performance analysis of global path planning techniques for autonomous mobile robots in grid environments," *International Journal of Advanced Robotic Systems*, vol. 14, no. 2, p. 1729881416663663, 2017, doi:10.1177/1729881416663663. [Online]. Available: https://doi.org/10.1177/1729881416663663

[131] L.-S. Chou and S.-M. Song, "Geometric Work of Manipulators and Path Planning Based on Minimum Energy Consumption," *Journal of Mechanical Design*, vol. 114, no. 3, pp. 414–421, 09 1992, doi:10.1115/1.2926567. [Online]. Available: https://doi.org/10.1115/1.2926567

[132] S. Muthuswamy and S. Manoochehri, "Optimal Path Planning for Robot Manipulators," *Journal of Mechanical Design*, vol. 114, no. 4, pp. 586–595, 12 1992, doi:10.1115/1.2917048. [Online]. Available: https://doi.org/10.1115/1.2917048

[133] Z. Shiller and Y.-R. Gwo, "Collision-free path planning of articulated manipulators," *J. Mech. Des*, vol. 115, no. 4, pp. 901–908, Dec. 1993, doi:10.1115/1.2919285. [Online]. Available: https://doi.org/10.1115/1.2919285

[134] C. Alexopoulos and P. M. Griffin, "Path planning for a mobile robot," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 318–322, March 1992.

[135] K. L. Hoffman, M. Padberg, and G. Rinaldi, *Traveling Salesman Problem*. Boston, MA: Springer US, 2013, pp. 1573–1578, doi:10.1007/978-1-4419-1153-7-1068. [Online]. Available: https://doi.org/10.1007/978-1-4419-1153-7_1068

[136] A. C. Arrumdany, P. P. Sari, P. Rahmadani, and A. I. Lubis, "Web-based geographic information system (GIS) in determining shortest path of MSME medan city using bellman-ford algorithm," *Journal of Physics: Conference Series*, vol. 1255, p. 012075, aug 2019, doi:10.1088/1742-6596/1255/1/012075. [Online]. Available: https://doi.org/10.1088%2F1742-6596%2F1255%2F1%2F012075

[137] A. K. Guruji, H. Agarwal, and D. Parsediya, "Time-efficient a* algorithm for robot path planning," *Procedia Technology*, vol. 23, pp. 144 – 149, 2016, 3rd International Conference on Innovations in Automation and Mechatronics Engineering 2016, ICIAME 2016 05-06 February, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212017316300111

[138] M. Bendsøe, "Optimal shape design as a material distribution problem," *Structural optimization*, vol. 1, no. 4, p. 193–202, 1989.

[139] T. J. R. Hughes, *The Finite Element Method*. Dover Publications, 2000.

[140] K. Hormann and A. Agathos, "The point in polygon problem for arbitrary polygons," *Computational Geometry: Theory and Applications*, vol. 20, no. 3, pp. 131 – 44, 2001. [Online]. Available: http://dx.doi.org/10.1016/S0925-7721(01)00012-8

[141] D. C. Rennels and H. M. Hudson, *Pipe Flow: A Practical and Comprehensive Guide*. "", 2012. [Online]. Available: http://dx.doi.org/10.1002/9781118275276

[142] A. J. Tarquin and J. Dowdy, "Optimal pump operation in water distribution," *Journal of Hydraulic Engineering*, vol. 115, no. 2, pp. 158–168, 1989. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%290733-9429%281989%29115%3A2%28158%29

[143] Y. Nakayama, "Chapter 7 - flow in pipes," in *Introduction to Fluid Mechanics (Second Edition)*, second edition ed., Y. Nakayama, Ed. Butterworth-Heinemann, 2018, pp. 135 – 161. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780081024379000073

[144] J. Martins and J. Hwang, "Review and unification of methods for computing derivatives of multidisciplinary computational models," *AIAA Journal*, vol. 51, no. 11, pp. 2582 – 99, 2013. [Online]. Available: http://dx.doi.org/10.2514/1.J052184

[145] D. Sunday, "Distance between 3d lines & segments," 2012. [Online]. Available: http://geomalgorithms.com/a07-_distance.html

[146] S. R. T. Peddada, N. M. Dunfield, L. E. Zeidner, K. A. James, and J. T. Allison, "Systematic enumeration and identification of unique spatial topologies of 3d systems using spatial graph representations," in *ASME 2021 International Design Engineering Technical Conferences,*, Virtual, Online, Aug. 2021.

[147] H. H. Goode and R. E. Machol, *System engineering : an introduction to the design of large-scale systems / [by] Harry H. Goode [and] Robert E. Machol*. McGraw-Hill New York, 1957.

[148] P. H. Sydenham, *Systems approach to engineering design*. Artech House Boston, MA, 2003.

[149] B. W. B. W. Field and U. of Melbourne. Department of Mechanical Engineering, *Introduction to engineering design / Bruce W. Field*. Clayton, Vic: University of Melbourne, Dept. of Mechanical Engineering, 2006, no. Accessed from https://nla.gov.au/nla.cat-vn3585408.

[150] H. Ashrafiuon and N. K. Mani, "Analysis and optimal design of spatial mechanical systems," *J. Mech. Des*, vol. 112, no. 2, pp. 200–207, June 1990. [Online]. Available: https://doi.org/10.1115/1.2912593

[151] B. W. Field, *Introduction to design processes / Bruce W. Field*, 2nd ed. Department of Mechanical Engineering, Monash University Clayton, Vic, 1997.

[152] S. Challender, "Systems thinking, systems practice. by peter b. checkland. published by john wiley, chichester, uk, 1981, 330 pp., isbn 0 471 27911 0 (republished 1999 in paperback, with a 30-year retrospective). from a practitioner perspective," *Systems Research and Behavioral Science*, vol. 17, no. S1, pp. S78–S80, 2000. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/1099-1743%28200011%2917%3A1%2B%3C%3A%3AAID-SRES384%3E3.0.CO%3B2-N

[153] D. F. Wyatt, D. C. Wynn, and P. J. Clarkson, "A scheme for numerical representation of graph structures in engineering design," *J. Mech. Des*, vol. 136, no. 1, Nov. 2013. [Online]. Available: https://doi.org/10.1115/1.4025961

[154] L. C. Schmidt, H. Shetty, and S. C. Chase, "A graph grammar approach for structure synthesis of mechanisms," *J. Mech. Des*, vol. 122, no. 4, pp. 371–376, July 1999. [Online]. Available: https://doi.org/10.1115/1.1315299

[155] E. Moguel, J. M. Conejero, F. Sánchez-Figueroa, J. Hernández, J. C. Preciado, and R. Rodríguez-Echeverría, "Towards the use of unmanned aerial systems for providing sustainable services in smart cities," *Sensors (Basel, Switzerland)*, vol. 18, no. 29280984, p. 64, Dec. 2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5795605/

[156] J. Yu, Z. Cao, M. Cheng, and R. Pan, "Hydro-mechanical power split transmissions: Progress evolution and future trends," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 233, no. 3, pp. 727–739, Jan. 2018. [Online]. Available: https://doi.org/10.1177/0954407017749734

[157] K. Kyprianidis, "An approach to multi-disciplinary aero engine conceptual design," 2017.

[158] S. Park and D. Jung, "Design of vehicle cooling system architecture for a heavy duty series-hybrid electric vehicle using numerical system simulations," *J. Eng. Gas Turbines Power*, vol. 132, no. 9, June 2010. [Online]. Available: https://doi.org/10.1115/1.4000587

[159] E. Flapan, T. Mattman, B. Mellor, R. Naimi, and R. Nikkuni, "Recent developments in spatial graph theory," *arXiv: Geometric Topology*, 2016.

[160] S. Taylor, "Abstractly planar spatial graphs," *arXiv: Geometric Topology*, 2019.

[161] E. Flapan, B. Mellor, and R. Naimi, "Spatial graphs with local knots," *Revista Matemática Complutense*, vol. 25, pp. 493–510, 2012.

[162] B. Trace, "On the reidemeister moves of a classical knot," 1983.

[163] J. Hass and J. Lagarias, "The number of reidemeister moves needed for unknotting," *Journal of the American Mathematical Society*, vol. 14, pp. 399–428, 1998.

[164] C. Hayashi, "The number of reidemeister moves for splitting a link," *Mathematische Annalen*, vol. 332, no. 2, pp. 239–252, 2005. [Online]. Available: https://doi.org/10.1007/s00208-004-0599-x

[165] J. Hass, J. C. Lagarias, and N. Pippenger, "The computational complexity of knot and link problems," *J. ACM*, vol. 46, no. 2, pp. 185–211, 1999. [Online]. Available: https://doi.org/10.1145/301970.301971

[166] M. Lackenby, "The efficient certification of knottedness and thurston norm," p. 101 pages, 2019.

[167] A. Ishii, "On normalizations of a regular isotopy invariant for spatial graphs," *International Journal of Mathematics*, vol. 22, pp. 1545–1559, 2011.

[168] S. Negami, "Polynomial invariants of graphs," *Transactions of the American Mathematical Society*, vol. 299, pp. 601–622, 1987.

[169] S. Cho and Y. Koda, "Topological symmetry groups and mapping class groups for spatial graphs," *Michigan Mathematical Journal*, vol. 62, pp. 131–142, 2011.

[170] E. Flapan, A. Henrich, A. Kaestner, and S. Nelson, "Knots, links, spatial graphs, and algebraic invariants," 2017.

[171] D. Bar-Natan, "On the vassiliev knot invariants," *Topology*, vol. 34, pp. 423–472, 1995.

[172] L. Kauffman, "New invariants in the theory of knots," *American Mathematical Monthly*, vol. 95, pp. 195–242, 1988.

[173] A. Thompson, "A polynomial invariant of graphs in 3-manifolds," *Topology*, vol. 31, pp. 657–665, 1992.

[174] J. Alexander, "Topological invariants of knots and links," *Transactions of the American Mathematical Society*, vol. 30, pp. 275–306.

[175] K. Murasugi, "Jones polynomials and classical conjectures in knot theory," *Topology*, vol. 26, pp. 187–194, 1987.

[176] L. Kauffman, "Invariants of graphs in three-space," *Transactions of the American Mathematical Society*, vol. 311, pp. 697–710, 1989.

[177] A. Dobrynin and A. Vesnin, "On the yoshinaga polynomial of spatial graphs," *Kobe journal of mathematics*, vol. 20, pp. 31–37, 2003.

[178] Y. Yokota, "Topological invariants of graphs in 3-space," *Topology*, vol. 35, pp. 77–87, 1996.

[179] T. Kong, A. Lewald, B. Mellor, and V. Pigrish, "Colorings, determinants and alexander polynomials for spatial graphs," *arXiv: Geometric Topology*, 2015.

[180] J. Murakami, "The yamada polynomial of spacial graphs and knit algebras," *Communications in Mathematical Physics*, vol. 155, pp. 511–522, 1993.

[181] S. Yamada, "An invariant of spatial graphs," *Journal of Graph Theory*, vol. 13, no. 5, pp. 537–551, nov 1989.

[182] A. Vesnin and A. Dobrynin, "The yamada polynomial for graphs, embedded knotwise into three-dimensional space," *Vychislitel'nye Sistemy*, vol. 155, Jan. 1996.

[183] M. Li, F. Lei, F. Li, and A. Vesnin, "On yamada polynomial of spatial graphs obtained by edge replacements," *arXiv: Geometric Topology*, 2018.

[184] Q. Deng, X. Jin, and L. Kauffman, "The generalized yamada polynomials of virtual spatial graphs," *arXiv: Geometric Topology*, 2018.

[185] J. Hopcroft and R. Tarjan, "Efficient planarity testing," *J. Assoc. Comput. Mach.*, vol. 21, pp. 549–568, 1974. [Online]. Available: https://doi-org.proxy2.library.illinois.edu/10.1145/321850.321852

[186] B. A. Burton, "The Next 350 Million Knots," in *36th International Symposium on Computational Geometry (SoCG 2020)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), S. Cabello and D. Z. Chen, Eds., vol. 164. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2020/12183 pp. 25:1–25:17.

[187] N. Oyamaguchi, "Enumeration of spatial 2-bouquet graphs up to flat vertex isotopy," *Topology and its Applications*, vol. 196, pp. 805–814, 2015.

[188] T. Kanenobu and K. Sugita, "Finite type invariants of order 3 for a spatial handcuff graph," *Topology and its Applications*, vol. 159, pp. 966–979, 2012.

[189] H. Moriuchi, "Enumeration of algebraic tangles with applications to theta-curves and handcuff graphs," *Kyungpook Mathematical Journal*, vol. 48, pp. 337–357, 2008.

[190] H. Moriuchi, "A table of theta-curves and handcuff graphs with up to seven crossings," 2009.

[191] T. Soma, "Spatial-graph isotopy for trivalent graphs and minimally knotted embeddings," *Topology and its Applications*, vol. 73, pp. 23–41, 1996.

[192] E. Fominykh, S. Garoufalidis, M. Goerner, V. Tarkaev, and A. Vesnin, "A census of tetrahedral hyperbolic manifolds," *Exp. Math.*, vol. 25, no. 4, pp. 466–481, 2016. [Online]. Available: https://doi-org.proxy2.library.illinois.edu/10.1080/10586458.2015.1114436

[193] T. Guo, D. R. Herber, and J. T. Allison, "Reducing evaluation cost for circuit synthesis using active learning," in *ASME 2018 International Design Engineering Technical Conferences*, no. DETC2018-85654, Quebec City, Canada, Aug. 2018, doi:10.1115/DETC2018-85654. p. V02AT03A011.