Constraint-oriented refinement-efficacious memetic algorithms for efficient optimization of computationally-expensive problems

Handoko, Stephanus Daniel

2013

Handoko, S. D. (2013). Constraint-oriented refinement-efficacious memetic algorithms for efficient optimization of computationally-expensive problems. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/54901>

<https://doi.org/10.32657/10356/54901>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**CONSTRAINT-ORIENTED REFINEMENT-EFFICACIOUS
MEMETIC ALGORITHMS FOR EFFICIENT OPTIMIZATION
OF COMPUTATIONALLY-EXPENSIVE PROBLEMS**

**STEPHANUS DANIEL HANDOKO
SCHOOL OF COMPUTER ENGINEERING
2013**

**CONSTRAINT-ORIENTED REFINEMENT-EFFICACIOUS
MEMETIC ALGORITHMS FOR EFFICIENT OPTIMIZATION
OF COMPUTATIONALLY-EXPENSIVE PROBLEMS**

STEPHANUS DANIEL HANDOKO

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for
the degree of Doctor of Philosophy

2013

Abstract

Successes of the Memetic Algorithms (MAs) have since alleviated the problems of local optimum trap of the deterministic optimization techniques as well as of slow, imprecise convergence of the stochastic optimization approaches. Through the hybridization of local search and global search by interleaving deterministic and stochastic optimizers one after the other, the MAs have become capable of searching more efficiently than their conventional counterparts. Gradually, this has brought paradigm shift to the research works in the particular area—which currently are geared towards solving complex science and engineering problems that are computationally-expensive in nature. Where minutes to hours of CPU time is consumed per evaluation, local search for each individual as adopted by the simple MA is clearly not an efficient strategy. Surrogate models in place of the actual problem landscape have been proposed as a response. Alternatively, selective local search can be employed to refine only some potential individuals to ease the computational burdens. The choice of individuals in the population to undergo local refinement is indeed critical to not only the efficiency but also the effectiveness of the MA. Some simple heuristics can be pretty intuitive, *e.g.* refining only the top few individuals in the population. This, nevertheless, may compromise the effectiveness of the MA.

Presented in this thesis is a framework of MA named Constrained-Oriented Refinement-Efficacious Memetic Algorithm (CORE-MA) that enables intelligent selection of the individuals to undergo local refinement. CORE-MA is the first *optinformatics* (learning-from-archive paradigm) for the constrained real-valued optimization. Oriented by constraint values that describe the problem topology, CORE-MA allows restraints on some unnecessary refinements thereby focusing the search efforts on the important regions of the search space and accelerating the identification of the global optimum. Feasibility Structure Modeling (FSM), as the first CORE-MA instantiation, formulates a classification problem within the optimization framework for the first time and uses Support Vector Machine (SVM) to solve it. The SVM prediction is subsequently used to determine if an individual is worth refining. Constrained Landscape Stratification (CLS), being the second CORE-MA instantiation, employs the simple nearest-neighbor-based dichotomizer directly after stratifying the problem landscape using the gradient information. This, consequently, eliminates the need for SVM. Less overhead is incurred as the result. Multi-fold accelerations were seen following experiments using the CORE-MA on the benchmark problems. Greater rate of descent with respect to the potential energy trace and up to 25 times speed-ups with respect to the docking time were witnessed upon the applications of the CORE-MA to the molecular mechanics and molecular docking problems, respectively.

Contributions

The research works presented in this thesis have made significant contributions in the following ways:

1. CORE-MA provides the alternative to the surrogate-assisted MAs to ease the computational burdens in optimizing some computationally-expensive problems by executing local refinements selectively on the actual problem landscape.
2. CORE-MA is the first *optinformatics* (learning-from-archive paradigm) in the context of constrained real-valued optimization where the structure of the problem search space is learned as it evolves so as to enable intelligent and effective selection of the individuals to undergo local refinement.
3. In realizing the adaptive selection of the local refinements, a classification problem is for the first time formulated within an optimization framework in FSM. As the first CORE-MA instantiation, the FSM solves the arising classification problem using the SVM and subsequently uses its prediction to decide if an individual is worth refining. This presents a novel concept of employing the machine learning techniques for the better evolutions of the evolutionary computing procedures.

4. The use of SVM in FSM is substituted with the simple nearest-neighbor-based dichotomizer in CLS. As the second CORE-MA instantiation, CLS not only allows adaptive selection of local refinements but further reduces the computational overhead incurred by FSM-driven MAs over the simple MA by first stratifying the problem landscape using the objective and/or constraint gradients.

Publications

Journals

1. **S.D. Handoko**, C.K. Kwoh, and Y.S. Ong, "Feasibility Structure Modeling: An Effective Chaperone for Constrained Memetic Algorithms," *IEEE Transactions on Evolutionary Computation*, 14, pp.740–758, 2010. (IF: **3.341**)
2. X. Ouyang, **S.D. Handoko**, and C.K. Kwoh, "CScore: A Simple yet Effective Scoring Function for ProteinLigand Binding Affinity Prediction Using Modified CMAC Learning Architecture," *Journal of Bioinformatics and Computational Biology*, 9, pp.1–14, 2011.
3. M.M.H. Ellabaan, **S.D. Handoko**, Y.S. Ong, C.K. Kwoh, et al., "A Tree-structured Covalent-bond-driven Molecular Memetic Algorithm for Optimization of Ring-deficient Molecules," *Computers and Mathematics with Applications*, 64, pp.3792–3804, 2012. (IF: **1.747**)
4. **S.D. Handoko**, X. Ouyang, C.T.T. Su, C.K. Kwoh, and Y.S. Ong, "QuickVina: Accelerating AutoDock Vina Using Gradient-based Heuristics for Global Optimization," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9, pp.1266–1272, 2012. (IF: **2.246**)

5. M.M.H. Ellabaan, Y.S. Ong, **S.D. Handoko**, C.K. Kwoh, and H.Y. Man, "Discovering Unique, Low-energy Transition States Using Evolutionary Molecular Memetic Computing," IEEE Computational Intelligence Magazine, in Press. (IF: **3.368**)

Conferences

1. **S.D. Handoko**, C.K. Kwoh, Y.S. Ong, G.L. Zhang, and B. Vladimir, "Extreme Learning Machine for Predicting HLA Peptide Binding," in *Proceedings of the 3rd International Symposium on Neural Networks, Lecture Notes in Computer Science*, vol.3973, pp.716-721, May 2006.
2. **S.D. Handoko**, C.K. Kwoh, Y.S. Ong, and M.H. Lim, "A Study on Constrained MA Using GA and SQP: Analytical vs. Finite-difference Gradients," in *Proceedings of 2008 IEEE World Congress on Computational Intelligence*, pp.4031-4038, Jun 2008.
3. **S.D. Handoko**, C.K. Kwoh, and Y.S. Ong, "Using Classification for Constrained Memetic Algorithm: A New Paradigm," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pp.547-552, Oct 2008.
4. C.N. Otto, **S.D. Handoko**, C.K. Kwoh, "Identifying Patterns of Conformational Changes in HLA-A*0201-related Immunological Activities," in *Proceedings of 2009 IEEE International Conference on Biomedical and Pharmaceutical Engineering*, pp.1-6, Dec 2009.

5. **S.D. Handoko**, C.K. Kwoh, and Y.S. Ong, "Classification-assisted Memetic Algorithms for Equality-constrained Optimization Problems," in *Proceedings of the 22nd Australasian Joint Conference on Artificial Intelligence, Lecture Notes in Artificial Intelligence*, vol.5866, pp.391-400, Dec 2009.
6. T.T.C. Su, **S.D. Handoko**, C.K. Kwoh, C. Schnbach, and X. Li, "A Possible Mutation that Enables H1N1 Influenza-A Virus to Escape Antibody Recognition," in *Proceedings of 2010 IEEE International Conference on Bioinformatics and Biomedicine*, pp.81–84, Dec 2010.
7. **S.D. Handoko**, T.T.C. Su, C.K. Kwoh, and Y.S. Ong, "Structural Analysis of (TCR)HLA/peptide Complexes: An Initial Study," in *Proceedings of 2010 IEEE International Conference on Bioinformatics and Biomedicine Workshops*, pp.63–66, Dec 2010.
8. **S.D. Handoko**, C.K. Kwoh, Y.S. Ong, and J. Chan, "Classification-assisted Memetic Algorithms for Solving Optimization Problems with Restricted Equality Constraint Function Mapping," in *Proceedings of 2011 IEEE Congress on Evolutionary Computation*, pp.1209–1216, Jun 2011.

Acknowledgement

I would like to express my greatest gratitude to A/P Dr. Kwoh Chee Keong as my Ph.D. main supervisor in Nanyang Technological University. His invaluable comments and sincere encouragements during indecisive moments have allowed the fruitful achievements of this project.

I would also like to thank my Ph.D. co-supervisor, A/P Dr. Ong Yew Soon, for the beneficial discussions and comments on my works.

My gratitude too goes to Ms. Su Tran To Chinh and Mr. Ouyang Xuchang for the team works as well as to the anonymous examiners who have assisted in enhancing this thesis through their invaluable comments. My appreciation also goes to Mr. Anthony Kho and Mr. Charlie Nathanael Otto for helping me with their expertise in coding and to Mr. Wahyu Surya with his expertise in biology and Linux operating system.

Last but not least, I am also grateful for all the people who have supported me in one way or another to the completion of this thesis—parents, colleagues, and friends.

Contents

Abstract	i
Contributions	ii
Publications	iv
Acknowledgement	vii
List of Figures	xiv
List of Tables	xvii
List of Algorithms	xix
1 Introduction	1
1.1 Background and Motivation	1
1.2 Objectives	4
1.3 Contributions	4
1.4 Organization	5
2 Constrained Optimization Problem Solvers	9
2.1 Optimal Solution	10

2.2	Steepest Descent Direction	11
2.2.1	Analytical Differentiation	12
2.2.2	Numerical Differentiation	13
2.3	Mathematical Programming	14
2.3.1	Linear Programming	15
2.3.2	Quadratic Programming	16
2.3.3	Nonlinear Programming	17
2.4	Genetic Algorithm (GA)	20
2.4.1	Ranking	21
2.4.2	Selection	22
2.4.3	Crossover	23
2.4.4	Mutation	24
2.5	Memetic Algorithm (MA)	25
3	Common Practices in Computer Science	29
3.1	Lookup Table	30
3.1.1	Implementation	34
3.1.2	Manipulation	35
3.1.2.1	Searching	35
3.1.2.2	Appending	37
3.2	Nearest Neighbors	37
3.2.1	Definition	39
3.2.2	Identification	40
3.2.3	Characterization	41

3.2.3.1	Uniform Neighbors	43
3.2.3.2	Composite Neighbors	44
3.3	Support Vector Machine (SVM)	45
4	Constraint-Oriented Refinement-Efficacious MA	49
4.1	The Roadmap	51
4.2	Preliminaries	53
4.2.1	In-feasibility Class	53
4.2.2	Optimum Immediacy Corollary	55
4.2.2.1	Optimum in Feasible Space Interior	57
4.2.2.2	Optimum at Feasible Space Boundary	60
4.2.2.3	Optimum at Either Location	64
4.3	Feasibility Structure Modeling (FSM)	66
4.3.1	Feasibility Structure	67
4.3.2	Feasibility Boundary	71
4.3.3	FSM-driven MA	78
4.3.3.1	Original FSM	79
4.3.3.2	Generalized FSM	83
4.3.3.3	Ecumenical FSM	89
4.4	Constrained Landscapes Stratification (CLS)	95
4.4.1	Search Space Stratification	96
4.4.1.1	Stratification of Refinement Targets	97
4.4.1.2	Stratification of Optimizing Directions	99
4.4.2	Stratum Assessment	101

4.4.2.1	Feasible Space Interior	103
4.4.2.2	Feasible Space Boundary	104
4.4.2.3	Feasible Space Exterior	106
4.4.3	CLS-powered MA	108
5	Empirical Results and Discussions	113
5.1	Experimental Setups	115
5.1.1	Global Search	116
5.1.2	Local Search	116
5.1.3	<i>Optinformatics</i>	116
5.2	Results Overview	117
5.3	Efficacy Tests	119
5.3.1	Avaricious Static Strategies	119
5.3.1.1	Few Best Individuals	120
5.3.1.2	Feasible Individuals	128
5.3.2	Rational Dynamic Strategies	130
5.3.2.1	FSM-driven MA	138
5.3.2.2	CLS-powered MA	141
5.4	Robustness Tests	146
5.5	Time Complexity	157
6	Practical Applications	159
6.1	Background	160
6.2	Molecular Mechanics	162

6.2.1	Steric Energy	162
6.2.1.1	Bond Stretching	163
6.2.1.2	Bond Bending	164
6.2.1.3	Bond Rotation	165
6.2.1.4	Electrostatic Interactions	166
6.2.1.5	van der Waals Interactions	166
6.2.1.6	Parameterizations	167
6.2.2	Energy Minimization	168
6.2.2.1	Design Variables	168
6.2.2.2	Objective Function	172
6.2.2.3	Constraint Functions	172
6.2.2.4	Empirical Results	173
6.3	Molecular Docking	176
6.3.1	Vina	177
6.3.2	QuickVina	178
6.3.3	PDBbind Screening	180
7	Conclusion and Future Works	185
7.1	Conclusion	185
7.2	Future Works	187
7.2.1	Methods	187
7.2.1.1	Efficient Lookup Table	187
7.2.1.2	Optimal Neighborhood Size	188
7.2.1.3	Intra-individual Significance Assessment	188

7.2.1.4	Combinatorics CORE-MA	189
7.2.1.5	Multi-objective CORE-MA	189
7.2.2	Applications	190
7.2.2.1	Composite Formula Calculation	190
7.2.2.2	Intricate System Simulation	191
A	Supplementary Results	193
A.1	Threshold Variation	193
A.2	Running Time	224
	Bibliography	231

List of Figures

1.1	Roadmap to CORE-MA	7
1.2	Protocol for Efficacy and Robustness Tests of CORE-MA	8
3.1	Lookup Table for Optimization Archives	32
3.2	Possible Neighborhood Structures	43
4.1	CORE-MA Instantiation Scheme	50
4.2	An illustration of the ε -neighborhood of solution $\mathbf{x}$, $\mathcal{N}_\varepsilon(\mathbf{x})$, and the δ -neighborhood of optimum $\mathbf{x}^*$, $\mathcal{N}_\delta(\mathbf{x}^*)$, with $\delta = 2\varepsilon$ in the optimum immediacy.	57
4.3	An illustration of some optimum $\mathbf{x}^*$ at the trough of a feasible basin with its δ -neighborhood $\mathcal{N}_\delta(\mathbf{x}^*)$	58
4.4	An illustration of some optimum $\mathbf{x}^*$ at the feasibility boundary with its δ -neighborhood $\mathcal{N}_\delta(\mathbf{x}^*)$	61
4.5	Equality Constraints in Two- and Three-dimensional Space . . .	63
4.6	Feasibility Structure of Constrained Optimization Problem . . .	68
4.7	Linear Approximation of Local Feasibility Boundary	75
4.8	Misclassification from Linear Approximation	77
4.9	Nonlinear Local Feasibility Boundary	80

4.10 Feasibility Boundary Approximation of Equality Constraint . . .	85
4.11 Potential Regions under Generalized FSM	88
4.12 Aggregate vs. Individual Inequality Constraints	91
4.13 Potential Regions under Ecumenical FSM	93
4.14 Stratification of Search Space	97
4.15 Stratification Based on Optimizing Directions	100
4.16 Inaccurate Stratum Approximation During Early Generations of Memetic Search	107
6.1 Bond Stretching	163
6.2 Bond Bending	164
6.3 Bond Rotation	165
6.4 Cartesian Coordinates	169
6.5 Internal Coordinates	170
6.6 Amino Acid	171
6.7 Peptide Bond	171
6.8 Minimum Energy Trace of Polyalanine	175
6.9 Function of A Single Design Variable with Derivative at Different Points	179
6.10 Speed-up against Number of Rotatable Bonds	181
6.11 Scores of Docking Results Using Vina and QuickVina	181
6.12 RMSE of Docking Results with Respect to Native Conformations in PDB	182

6.13 Comparison of Vina (Blue) and QVina (Green) Docking Results	
against Native Conformations in PDB (Red)	184

List of Tables

5.1	Characteristics of Benchmark Problems	114
5.2	FFE Count Statistics of MAs Using Avaricious Static Strategies over 30 Independent Runs	121
5.3	FFE Count Statistics of MAs with Rational Dynamic Strategies over 30 Independent Runs	131
5.4	FFE Count Statistics of MAs with Rational Dynamic Strategies over 150 Independent Runs	147
A.1	FFE Count Statistics of CLS of Various Significance Thresholds on Uniform Neighborhood of Feasible Individual	194
A.2	FFE Count Statistics of CLS of Various Significance Thresholds on Composite Neighborhood of Feasible Individual	200
A.3	FFE Count Statistics of CLS of Various Significance Thresholds on Composite Neighborhood of Infeasible Individual	206
A.4	FFE Count Statistics of CLS of Various Significance Thresholds on Uniform Neighborhood of Infeasible Individual	212
A.5	FFE Count Statistics of CLS of Various Significance Thresholds under Insufficient Diversity	218

A.6 CPU Time Statistics of MAs Using Rational Dynamic Strategies	
over 150 Independent Runs	225

List of Algorithms

2.1	Simple Genetic Algorithm	21
2.2	Simple Memetic Algorithm	27
3.1	Lookup Table Search Procedure	35
3.2	Nearest Neighbors Search Procedure	41
4.1	Adaptive Memetic Algorithm	51
4.2	Original FSM-driven Memetic Algorithm	81
4.3	Generalized FSM-driven Memetic Algorithm	87
4.4	Ecumenical FSM-driven Memetic Algorithm	94
4.5	CLS-powered Memetic Algorithm	110

Chapter 1

Introduction

1.1 Background and Motivation

Successes of the Memetic Algorithms (MAs) have since alleviated the problems of local optimum trap of the deterministic optimization techniques as well as of slow, imprecise convergence of the stochastic optimization approaches. Through the hybridization of local search and global search by interleaving deterministic and stochastic optimizers one after the other, the MAs have become capable of searching more efficiently than their conventional counterparts. Gradually, this has brought paradigm shift to the research works in that particular area. Lower cost of the computing power has propelled research in the field towards solving complex science and engineering problems which are computationally-expensive in nature. Evaluating the fitness of a trial solution to such problems may easily require minutes to hours of CPU time. Completing the entire optimization run would therefore take months to years even with only a few tens of thousands of evaluations performed. Locating the global optimum with the smallest number of fitness evaluations possible is thus highly desirable.

In the simplest form, MA interleaves global search by means of a stochastic optimization procedure with local search using some deterministic optimization method one after another. Exhaustive search nature of the simple MA is surely not the most efficient strategy to solve computationally-expensive problems. In the attempt to address the huge number of fitness function evaluations (FFEes) normally required in the case of the simple MA, an approximated landscape to supersede the actual problem landscape was used for the purpose of performing evaluation of the trial solutions to unconstrained problems [1, 2, 3]. Addressing the constrained problems, a similar concept can be easily extended and applied to the constraint landscape. A considerable effort for executing expensive exact fitness evaluations during the local refinements can therefore be bypassed. It is, however, well-understood that errors may be resulted from such approximation thereby misleading local refinements to converge to false optima. When solving the unconstrained problems, repairing a false optimum on the actual landscape can indeed be more efficient since the false optimum may in fact be closer than its corresponding initial individual to the actual optimum. The repair can thus be more cost-efficient than complete refinement on the actual landscape. When solving the constrained problems, unfortunately, the compounded errors due to approximation of the multiple landscapes may complicate the repair, making it more susceptible to failing to be more cost-efficient than a complete refinement on the actual landscape. Choice of the individuals to undergo local refinements is therefore crucial to stay cost-efficient. Selective local refinements should not, however, hamper the effectiveness of the MA, thus the need for good strategies.

A simple heuristics, e.g. refining the few best individuals in the population, can be quite intuitive but may clearly compromise the effectiveness of the MA, particularly if the problem topology is a multi-modal landscape. More complex fitness-based strategies [4] had also been employed for adapting the probability of applying local refinements. In establishing if a local refinement will be worth executing, most—if not all—studies in the literature make no direct assessment of the problem topology. So far, there have not been assessment strategies that determine if an individual in the population will be worth refining by observing the problem topology in the surrounding of the individual. The search space of constrained optimization problems is clearly divided into feasible and infeasible spaces. An equality constraint requires the output of the constraint function to be equal to some constant, producing minute feasible space(s). With respect to the equality constraint, the infeasible space(s) in general can thus be separated into the spaces of two types of constraint values, *i.e.* below and above constant. Without loss of generality, the equality constraint may be transformed into one which requires the constraint value to be equal to zero, therefore distinguishing the infeasible space(s) into positive and negative infeasible spaces. Solutions to constrained optimization problems are known to rest on the boundary between a feasible and an infeasible spaces or the through of a feasible basin. Moreover, an archive in the form of a lookup table is often maintained in the optimization of computationally-expensive problems to avoid the expensive re-evaluations of those trial solutions that have been evaluated earlier. Following these cues, this thesis then focuses on the study of the topology-based assessment strategies.

1.2 Objectives

From the foregoing discussions about the underlying and motivating principles, it can be summarized that the purposes of the research works presented in this thesis is to address the challenges posed by the following questions:

1. With a lookup table that comprises the optimization archive maintained, can the problem topology be learned as the MA evolves?
2. Using the learned topology, can a good assessment strategy be devised to determine if an individual is worth refining?
3. Under some special circumstance, if any, can another assessment strategy be devised such that only minimum overhead is incurred?
4. Will the MA employing any of the devised strategies be not only efficient but also effective in solving computationally-expensive problems?

1.3 Contributions

Addressing the challenges posed by the foregoing questions, the research works presented in this thesis have thus made a number of significant contributions in ways that can be summarized as follows.

1. CORE-MA provides the alternative to the surrogate-assisted MAs to ease the computational burdens in optimizing some computationally-expensive problems by executing local refinements selectively on the actual problem landscape.

2. CORE-MA is the first *optinformatics* [5] (learning-from-archive model) in the context of constrained real-valued optimization where the structure of the problem search space is learned as it evolves so as to enable intelligent and effective selection of the individuals to undergo local refinement.
3. In realizing the adaptive selection of the local refinements, a classification problem is for the first time formulated within an optimization framework in FSM. As the first CORE-MA instantiation, the FSM solves the arising classification problem using the SVM and subsequently uses its prediction to decide if an individual is worth refining. This presents a novel concept of employing the machine learning techniques for the better evolutions of the evolutionary computing procedures.
4. The use of SVM in FSM is substituted with the simple nearest-neighbor-based dichotomizer in CLS. As the second CORE-MA instantiation, CLS not only allows adaptive selection of local refinements but further reduces the computational overhead incurred by FSM-driven MAs over the simple MA by first stratifying the problem landscape using the objective and/or the constraint gradients. Direct comparison between some individual and its respective relevant nearest neighbors is thus made possible.

1.4 Organization

Having introduced the problem to handle and the research objective to achieve in this chapter, the rest of this thesis is then organized as follows.

Chapter 2 reviews existing theories in operation research, methods for both local and global search, and strategies in addressing computationally-expensive optimization problems. Particularly, the review encompasses the mathematical programming routines as some representatives of the deterministic optimization algorithms, a genetic algorithm as representative of the stochastic optimization procedures, and memetic algorithms as representatives of the hybrid techniques that leverage the strengths of the deterministic and the stochastic optimization methods.

Chapter 3 then reviews the common practices in computer science that lend their specialties to the research works presented herein. These cover the lookup table and the nearest neighbors concept as well as the Support Vector Machine (SVM) approach to handle classification problems.

Chapter 4 provides step-by-step developments of some conceptual thoughts that eventually lead to the instantiation of CORE-MA by following a roadmap shown in Figure 1.1. As the first preliminaries for instantiating the CORE-MA are the In-feasibility Class in Section 4.2.1 and Optimum Immediacy Corollary in Section 4.2.2. The two concepts then give rise to the proposals of Feasibility Structure Modeling (FSM) and Constrained Landscape Stratification (CLS) in, respectively, Section 4.3 and Section 4.4. Subsequently, the FSM in turn brings about the instantiations of CORE-MA in the form of original (Section 4.3.3.1), generalized (Section 4.3.3.2), and ecumenical (Section 4.3.3.3) FSM-driven MA instances while the CLS, on the other hand, brings the instantiation of CORE-MA in the form of CLS-powered MA (Section 4.4.3).

Figure 1.1: Roadmap to CORE-MA

Chapter 5 presents and discusses some empirical results with the purpose of showcasing the efficacy and the robustness of different CORE-MA instances on a number of benchmark problems. First, 30 independent runs of adaptive MAs employing various assessment strategies were performed to show the efficacy of the CORE-MA instances in dealing with the limitations of the other strategies. Apart from the canonical simple MA that forms the baseline of comparisons by refining every individual in the population as well as the CORE-MA instances, the MAs consist of one that refines the feasible individuals only (Feasible MA), ones that allow refinement of the few best individuals (FBI-MAs), and one that refines individuals with feasible or composite neighborhood (FCN-MA). As can be seen from the experiment protocol in Figure 1.2, additional 120 independent runs were then carried out to show the robustness of CORE-MA.

Figure 1.2: Protocol for Efficacy and Robustness Tests of CORE-MA

Chapter 6 hence presents and discusses the efficacy of CORE-MA instances in solving practical applications of molecular mechanics and molecular docking problems.

Chapter 7 ultimately concludes this thesis with a measure of how well have the research works presented herein handled the questions posed in Section 1.2 and proposes plausible future works from both the theoretical aspect as well as the application perspective.

Chapter 2

Constrained Optimization Problem Solvers

A constrained optimization problem can generally be defined as the following.

Definition 2.1 (Constrained Optimization Problem) *Given an objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and some inequality and equality constraint functions $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^{n_g}$ and $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^{n_h}$, a constrained optimization problem seeks to identify a vector $\mathbf{x}$ of n real-valued independent variables that minimizes*

$$f(\mathbf{x})$$

subject to

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0} \tag{2.1}$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \tag{2.2}$$

The vector $\mathbf{x} \in \mathbb{R}^n$ is often referred to as the *candidate solution*. In addition, there may also exist some bound constraints in the form of $\mathbf{x}_\ell \leq \mathbf{x} \leq \mathbf{x}_u$ where $\mathbf{x}_\ell$ is the *lower bound* and $\mathbf{x}_u$ is the *upper bound*. These, however, can trivially be converted into the inequality constraints $\mathbf{x}_\ell - \mathbf{x} \leq \mathbf{0}$ and $\mathbf{x} - \mathbf{x}_u \leq \mathbf{0}$.

Although Definition 2.1 deals only with minimization problems, it is worth noting that maximizing $f(\mathbf{x})$ is equivalent to minimizing $-f(\mathbf{x})$. In this chapter, characteristics of the solutions to constrained optimization problems and how they can be identified will be discussed. The concept of steepest descent will first be reviewed. Its use in the deterministic mathematical programming will then follow. Stochastic methods will then conclude this chapter.

2.1 Optimal Solution

For any constrained optimization problems, the global or any local optimum $\mathbf{x}^*$ is characterized by satisfying the following optimality conditions well-known as the Karush-Kuhn-Tucker (KKT) conditions [6, 7].

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^{n_g} \mu_i \nabla g_i(\mathbf{x}^*) + \sum_{i=1}^{n_h} \nu_i \nabla h_i(\mathbf{x}^*) = \mathbf{0} \quad (2.3)$$

subject to

$$g_i(\mathbf{x}^*) \leq 0 \quad i = 1, \dots, n_g \quad (2.4)$$

$$h_i(\mathbf{x}^*) = 0 \quad i = 1, \dots, n_h \quad (2.5)$$

$$\mu_i \geq 0 \quad i = 1, \dots, n_g \quad (2.6)$$

$$\mu_i g_i(\mathbf{x}^*) = 0 \quad i = 1, \dots, n_g \quad (2.7)$$

From the conditions above, it can easily be appreciated that there would be multiple optima should the gradient of the objective or any constraint function was nonlinear. Furthermore, the optimum may occupy one of the following two possible locations in the absence of equality constraints.

1. The trough of a feasible basin.

Indicated by $\forall i \ g_i(\mathbf{x}^*) < 0$, the search for this particular type of optimum resembles the search for an unconstrained optimum.

2. A particular location along the feasibility boundary.

Indicated by $\exists i \ g_i(\mathbf{x}^*) = 0$, the search for this type of optimum should be focussed on the region surrounding the feasibility boundary [8].

2.2 Steepest Descent Direction

In the context of minimization, it is intuitive to look for some descent direction so as to proceed from one candidate solution to another in an iterative manner. Of the infinitely many identifiable descent directions is clearly the steepest one. Being heuristic, the steepest descent direction has been attested useful in many deterministic algorithms in dealing with optimization problems. The attraction remains on the fact that it can completely be described using mathematics, *i.e.* the negative of the gradient vector.

Gradient of a function $F(\mathbf{x})$ is the direction along which the greatest rate of increase of the function's value occurs. It is mathematically denoted as

$$\nabla F(\mathbf{x}) = \begin{bmatrix} \frac{\partial F(\mathbf{x})}{\partial x_1} \\ \frac{\partial F(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial F(\mathbf{x})}{\partial x_n} \end{bmatrix} \quad (2.8)$$

where

$$\frac{\partial F(\mathbf{x})}{\partial x_i} = \lim_{\lambda \rightarrow 0} \frac{F(x_1, \dots, x_i + \lambda, \dots, x_n) - F(x_1, \dots, x_i, \dots, x_n)}{\lambda} \quad (2.9)$$

is the partial derivative of $F(\mathbf{x})$ with respect to the variable x_i , which simply is the rate of change of the value of $F(\mathbf{x})$ as the result of the infinitesimally small change λ in the variable x_i .

Although there exists a stochastic method for approximating the derivatives [9, 10], most of the deterministic optimization algorithms employ either one of the two deterministic procedures of obtaining the derivatives since they produce reasonably small errors, if any. Discussed in brief in separate subsections below, the analytical differentiation—whenever available at reasonable cost—is always preferred to the numerical differentiation since finite-differencing causes greater needs of function evaluations [11].

2.2.1 Analytical Differentiation

The mathematical expression of some function is manipulated so as to arrive at the functional forms of its derivatives, hence the term analytical differentiation. Using the rules for derivatives, it is performed either by hands or by computers via the *automatic* or the *symbolic* differentiation [12]. Automatic differentiation may employ either *forward* or *reverse* mode [13] or combination of the two [14]. To perform analytical differentiation in the context of constrained optimization, mathematical expressions of the objective and the constraint functions must be available at hands. As the functions become more complicated, chain rules may simplify the analysis, which normally will result in the derivatives sharing some common sub-expressions, such as the case for atomic-level potential energy [15].

Once derived, computations of all the partial derivatives with respect to each of the n independent variables x_i for $i = 1, 2, \dots, n$ can be performed very quickly within only some constant multiple amount of time for evaluating the functions themselves [16].

2.2.2 Numerical Differentiation

In numerical differentiation, the infinitesimally small change λ is substituted by an arbitrarily small step δ , called the finite difference. Using different formulae, the derivatives are approximated to within certain error tolerances as expressed in the following two equations where $\mathbf{e}_i$ denotes a unit vector in the direction of the variable x_i .

$$\frac{\partial F(\mathbf{x})}{\partial x_i} = \frac{F(\mathbf{x} + \delta \mathbf{e}_i) - F(\mathbf{x})}{\delta} + \mathcal{O}(\delta) \quad (2.10)$$

$$\frac{\partial F(\mathbf{x})}{\partial x_i} = \frac{-F(\mathbf{x} + 2\delta \mathbf{e}_i) + 4F(\mathbf{x} + \delta \mathbf{e}_i) - 3F(\mathbf{x})}{2\delta} + \mathcal{O}(\delta^2) \quad (2.11)$$

As the matter of fact, only evaluations of the function F at $\mathbf{x}$ and the localities are necessary in order to perform numerical differentiation as may immediately be concluded from the equations above. In order to obtain the gradient vector, however, it is necessary to compute all the partial derivatives with respect to each of the n independent variables x_i for $i = 1, 2, \dots, n$. It is therefore obvious that the FFE requirement in the context of constrained optimization is a linear function of the dimension of the problem should numerical differentiation be of preference. Depending on the accuracy needed, the FFE requirement fluctuates among different finite-differencing formulae. The better the accuracy a formula

may offer, the larger the FFE requirement it will claim. Between the above two formulae, the latter requires more FFEs but provides more accurate derivatives than the earlier. For further comparison, two additional FFEs per computation may be exchanged for higher accuracy with error that is of the fourth power of the step δ as given by

$$\frac{\partial F(\mathbf{x})}{\partial x_i} = \frac{-3F_4 + 16F_3 - 36F_2 + 48F_1 - 25F_0}{12\delta} + \mathcal{O}(\delta^4) \quad (2.12)$$

where

$$F_j \equiv F(\mathbf{x} + j\delta\mathbf{e}_i) \quad (2.13)$$

Equations (2.10), (2.11), and (2.12) impose FFE requirements of $n + 1$, $2n + 1$, and $4n + 1$, requiring exactly those multiples amount of time of the single FFE in order to obtain the full gradient vector of the function F at one particular $\mathbf{x}$. Unless the mathematical expressions of the functions are not readily accessible, numerical differentiation may not be the right choice to use with problems that have large dimensionality.

2.3 Mathematical Programming

Depending on the type of the objective and the constraint functions, there exist deterministic algorithms for solving constrained optimization problems utilizing the first-order and possibly the second-order derivatives of the functions. These are often referred to as the *mathematical programming*.

2.3.1 Linear Programming

Characterized by having linear objective and constraint functions, as hinted by the name, a linear program can therefore be formulated as finding a vector $\mathbf{x}$ of n real-valued independent variables that minimizes

$$\mathbf{f}^T \mathbf{x} \tag{2.14}$$

subject to

$$\mathbf{G}\mathbf{x} \leq \mathbf{g} \tag{2.15}$$

$$\mathbf{H}\mathbf{x} = \mathbf{h} \tag{2.16}$$

where $\mathbf{f}$, $\mathbf{g}$, and $\mathbf{h}$ are real vectors and $\mathbf{G}$ and $\mathbf{H}$ are real matrices. The simplex method [17] invented in 1947 manages to efficiently solve many linear programs in as many iterations as twice or thrice the number of equality constraints [18]. Sequentially testing adjacent vertices of the feasible space, which is a polytope, so that the objective value improves or remains unchanged at every new vertex, the algorithm may suffer from an exponential time complexity in the worst case scenario as demonstrated using carefully constructed examples [19]. A different strategy of solving linear programs is through the use of interior-point methods with a polynomial time complexity in the average and the worst case scenarios. Motivated by the research paper in 1984 [20], interior-point methods construct a sequence of strictly feasible points that lie on the interior of the polytope but never on its boundary.

2.3.2 Quadratic Programming

Specified by quadratic objective and linear constraint functions is the quadratic program. Because of its characteristics, a quadratic program can be formulated as finding a vector $\mathbf{x}$ of n real-valued independent variables that minimizes

$$\mathbf{x}^T \mathbf{F} \mathbf{x} - \mathbf{f}^T \mathbf{x} \quad (2.17)$$

subject to

$$\mathbf{G} \mathbf{x} \leq \mathbf{g} \quad (2.18)$$

$$\mathbf{H} \mathbf{x} = \mathbf{h} \quad (2.19)$$

where $\mathbf{f}$, $\mathbf{g}$, and $\mathbf{h}$ are real vectors while $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{H}$ are real matrices. In fact, it is mandatory that the matrix $\mathbf{F}$ comprises non-zero real elements. Otherwise, the constrained optimization problem resembles a linear program. Should $\mathbf{F}$ be a positive-semi-definite matrix, the objective is a convex function in which case the quadratic program has a global minimizer if there exists at least one vector $\mathbf{x}$ that satisfies all the constraints whereas the objective is bounded from below on the feasible space. Furthermore, the global minimizer is unique should $\mathbf{F}$ be a positive-definite matrix. In the absence of inequality constraints, the problem can be solved by first solving the linear system of equations. Otherwise, several methods to solve the problem include interior-point and active-set methods [12] as well as conjugate-gradient method [21]. The maturity of the field has led the quadratic programming to be an indispensable part of one efficient solver of the nonlinear programming discussed next and of a machine-learning technique called the support vector machine discussed in the next chapter.

2.3.3 Nonlinear Programming

Where the objective is neither linear nor quadratic or not all of the constraints are linear, the problem is a nonlinear program. As early as the 1960s, a class of deterministic algorithms termed the *methods of feasible directions* [22] has been developed to handle nonlinear programs. Employing either first or second order functional approximations, methods in the class proceed in an iterative manner from one candidate solution to another. For every iteration, a direction-finding sub-problem is solved; the solution of which is to be used to direct a line search along which the search for the global optimum will have to progress.

One of the earliest techniques for identifying the series of feasible directions is the so-called Zoutendijk's algorithm [23]. After solving a linear programming sub-problem for a feasible direction, the algorithm conducts a line search along that particular direction. The original algorithm, however, may not converge to a KKT point. Convergent variants of the algorithm are one that avoids sudden change in directions as one or more inactive constraints become active [24] and one that applies the concept of near-binding constraints [25].

Conceptually similar to the Zoutendijk's algorithm are techniques known as the *sequential linear programming* (SLP) approaches [26, 27, 28, 29], developed under separate organizations for various types of applications at different time. Apart from securing popularity because of the availability of efficient solvers of the linear programs, these approaches has essentially no guarantee to converge. As the remedies, the penalty SLP [30] and the simplified penalty SLP [31] were later proposed as two convergent variants of the approaches.

Employing first-order functional approximations, the Zoutendijk's algorithm and the SLP approaches are predictably susceptible to slow convergence due to zigzagging behaviors. The SLP approaches, nonetheless, enjoy quadratic rate of convergence should the global optimum of the optimization problem happen to be situated at one of the vertices of the feasible space. Employing second-order functional approximations so as to achieve quadratic rate of convergence under all circumstances is the *sequential quadratic programming* (SQP) [32].

The SQP, also known as the *successive* or *recursive* quadratic programming, represents the state-of-the-art amid the nonlinear programming solvers [33, 34]. Utilizing the Newton's method, the SQP directly solves the KKT conditions as an attempt of acquiring a local minimizer, which may be the global optimum of the nonlinear program. For each iteration of the Newton's method, there exists an accompanying sub-problem that happens to be the minimization problem of a quadratic approximation to the Lagrangian function of the nonlinear program over a set of linear approximations to the constraint functions. Given an iterate $(\mathbf{x}^{(k)}, \mu^{(k)}, \nu^{(k)})$ where $\mu^{(k)} \geq \mathbf{0}$ and $\nu^{(k)}$ are the Lagrange multiplier estimates of the inequality and the equality constraints, respectively, the quadratic program below is solved at each major iteration of the SQP.

$$f(\mathbf{x}^{(k)}) + \nabla f(\mathbf{x}^{(k)})^T \mathbf{d}^{(k)} + \frac{1}{2}[\mathbf{d}^{(k)}]^T \nabla^2 L(\mathbf{x}^{(k)}) \mathbf{d}^{(k)} \quad (2.20)$$

subject to

$$g_i(\mathbf{x}^{(k)}) + \nabla g_i(\mathbf{x}^{(k)})^T \mathbf{d}^{(k)} \leq 0 \quad i = 1, \dots, n_g \quad (2.21)$$

$$h_i(\mathbf{x}^{(k)}) + \nabla h_i(\mathbf{x}^{(k)})^T \mathbf{d}^{(k)} = 0 \quad i = 1, \dots, n_h \quad (2.22)$$

where

$$\nabla^2 L(\mathbf{x}^{(k)}) = \nabla^2 f(\mathbf{x}^{(k)}) + \sum_{i=1}^{n_g} \mu_i^{(k)} \nabla^2 g_i(\mathbf{x}^{(k)}) + \sum_{i=1}^{n_h} \nu_i^{(k)} \nabla^2 h_i(\mathbf{x}^{(k)}) \quad (2.23)$$

The vector of direction $\mathbf{d}^{(k)}$ and the set of Lagrange multipliers associated with both inequality and equality constraints of the nonlinear program are produced after successfully solving the quadratic program above. The line search formula below and mere assignments are employed in order to construct the new iterate $(\mathbf{x}^{(k+1)}, \mu^{(k+1)}, \nu^{(k+1)})$.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)} \quad (2.24)$$

The appropriate step length α can be determined by using ℓ_1 penalty function, which penalizes the objective value of an infeasible candidate solutions [35, 36]. Due to the second-order approximation of the objective function in the form of a quadratic program, the SQP enjoys quadratic rate of convergence.

The requirement of the second-order derivatives of the Lagrangian, $\nabla^2 L(\mathbf{x})$, at each major iteration of the SQP, nevertheless, poses significant challenges to the algorithm. Not only massive computational budget will have to be provided but also robust quadratic programming solver will need to be employed so as to deal with cases in which the exact calculations of the Hessian of the Lagrangian do not produce a positive-definite matrix. It is therefore recommended to keep the Hessian matrix positive-definite even though it may be positive-indefinite at the optimum point [35, 36]. This can be achieved should the Hessian matrix $\mathbf{H}$ be updated using the following quasi-Newton approximation, which also relaxes the computational budget.

$$\mathbf{H}^{(k+1)} = \mathbf{H}^{(k)} + \frac{\mathbf{q}^{(k)}[\mathbf{q}^{(k)}]^T}{[\mathbf{q}^{(k)}]^T \mathbf{p}^{(k)}} - \frac{\mathbf{H}^{(k)} \mathbf{p}^{(k)} [\mathbf{p}^{(k)}]^T \mathbf{H}^{(k)}}{[\mathbf{p}^{(k)}]^T \mathbf{H}^{(k)} \mathbf{p}^{(k)}} \quad (2.25)$$

where

$$\mathbf{p}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \quad (2.26)$$

$$\mathbf{q}^{(k)} = \nabla L(\mathbf{x}^{(k+1)}) - \nabla L(\mathbf{x}^{(k)}) \quad (2.27)$$

and

$$\nabla L(\mathbf{x}) = \nabla f(\mathbf{x}) + \sum_{i=1}^{n_g} \mu_i^{(k+1)} \nabla g_i(\mathbf{x}) + \sum_{i=1}^{n_h} \nu_i^{(k+1)} \nabla h_i(\mathbf{x}) \quad (2.28)$$

A super-linear rate of convergence can still be achieved even after the above modification to the basic SQP algorithm, yet convergence to any local optimum is an inherent characteristic of all deterministic optimization algorithms. Thus, it is practically impossible to devise a deterministic algorithm that outperforms an exhaustive search in assuring global convergence because nonlinear programs in the context of global optimization are generally intractable [37].

2.4 Genetic Algorithm

A class of stochastic algorithms having an ability to surmount the limitation of deterministic optimization approaches is the *evolutionary computation* [38, 39]. The most widely used technique in the class is the *genetic algorithm* (GA) [40], a population-based meta-heuristic optimization procedure initially proposed for solving combinatorial optimization problems. Inspired by natural inheritance of the genetic materials [41] and natural selections in the biological evolution [42], crossover, mutation, and survival-of-the-fittest are three main principles of GA.

Some successful applications of GA have been demonstrated in [43, 44, 45, 46].

Being the simplest variant, the simple GA assumes only one population evolved from one generation to the next using the procedure detailed in Algorithm 2.1.

For real-coded GA [47], four main schemes used in the procedures are discussed in the subsections that follow.

Algorithm 2.1 Simple Genetic Algorithm

```
Initialize a population of candidate solutions
Evaluate the fitness of each candidate solution
while no stopping criteria have been fulfilled do
    Rank the candidate solutions in accordance to their fitness values
    Select some candidate solutions based on their rankings for
        crossover and mutation
    Evolve the population through crossover and mutation
    Evaluate the fitness of each candidate solution
    Perform elitism
end while
```

2.4.1 Ranking

The fitness of the candidate solutions in the current population can directly be ranked by sorting their objective values in an ascending order in the context of unconstrained optimization. The existence of equality or inequality constraints in the context of constrained optimization, however, implies the requirement of some constraint-handling techniques [48]. Previously proposed methods include penalization schemes [49, 50, 51], repair method [52], diversity mechanism [53], stochastic [54] and the other rankings [55, 56, 57]. A straightforward scheme of

ranking [55] as described below has been used throughout this research work.

- Feasible candidate solution is preferred to the infeasible one
- Between two feasible candidate solutions, the one having better objective value is preferred
- Between two infeasible candidate solutions, the one having less amount of constraint violations is preferred

2.4.2 Selection

Selection is a genetic operator that selects candidate solutions from the current population to become parents of the offsprings generated by using other genetic operators: crossover and mutation. Candidate solutions with better fitness have better chance of being selected in general. Linear or other types of scaling need to be employed in order to transform their fitness values into the probability of being selected, based on which one of the following selection schemes will select the candidate solutions that may undergo crossovers and mutations. The first scheme is actually the most basic scheme, based on which the second and the third schemes were later developed.

- Roulette Wheel Selection Scheme [58]

Enforcing the concept of survival-of-the-fittest, roulette wheel selection is one basic selection scheme used throughout this research work that allots larger share of the roulette wheel to fitter candidate solutions and clearly smaller share of the roulette wheel to less fit candidate solutions.

- Stochastic Universal Selection Scheme [59]

While roulette wheel selection may repeatedly select the dominantly fitter candidate solutions in the population, stochastic universal selection shuns this phenomenon by uniformly assigning markers on the roulette wheel to enable simultaneous selection of the candidate solutions. It simply means a multi-armed instead of the normal single-armed hand equips the wheel.

- Tournament Selection Scheme [58]

Tournament selection carries out roulette wheel selection several times to produce a tournament pool of the candidate solutions. From such a pool, the best candidate solution is selected. The selection scheme thus applies additional selective pressure over original roulette wheel selection scheme.

2.4.3 Crossover

Crossover is a genetic operator that achieves a recombination of parental genes in order to produce the genes that will construct an offspring. The idea behind crossover is that the offspring may turn out to be better than both the parents should it inherit the best traits from each of its parents. Occurring throughout the course of evolution with certain probability, which commonly is fairly high, crossover allows GA to converge to some optimum as fitter and fitter offsprings are generated. For real-coded GA, the two crossover operators described below are commonly used and were indeed specifically designed where blend crossover is a generalization of arithmetic crossover.

- Arithmetic Crossover [60, 61, 62]

As suggested by the name, arithmetic crossover generates a new offspring by calculating the arithmetic mean of both parents. Should $\mathbf{x}_1$ and $\mathbf{x}_2$ be the two parents, the new offspring $\mathbf{x}_c$ is given by

$$\mathbf{x}_c = \frac{\mathbf{x}_1 + \mathbf{x}_2}{2} \quad (2.29)$$

- Blend Crossover [63, 64]

Similar to arithmetic crossover, blend crossover uses linear combination of the two parents to engender a new offspring. Unlike arithmetic crossover, however, blend crossover emphasizes no restriction that the new offspring must be the arithmetic mean of both parents. Should $\mathbf{x}_1$ and $\mathbf{x}_2$ represent the two parents, the new offspring $\mathbf{x}_c$ is given by

$$\mathbf{x}_c = \gamma \mathbf{x}_1 + (1 - \gamma) \mathbf{x}_2 \quad (2.30)$$

where $\gamma \in [-0.5, 1.5]$, giving more flexibility in generating new offsprings.

2.4.4 Mutation

Mutation is a genetic operator that modifies one or more genes of the offspring in order to occasionally introduce new genes to the gene pool. Happening after the crossover during the course of evolution with low probability, mutation has enabled GA to arrive at fitter candidate solutions than were ever found before. As the matter of fact, mutation allows GA to escape from a local optimum and converge to the true global optimum.

The most commonly used mutation scheme in the context of real-coded GA is the gaussian mutation [65, 66]. Obviously, gaussian mutation randomly adds Gaussian-distributed values to some specific genes of an offspring. These values have a mean of zero and a standard deviation σ as much as some proportion of the range of the variables. The standard deviation may also decrease with time such that stronger mutations in the early generations provide for exploration of the search space and weaker mutations in the late generations give supports to enhancement of the estimates of the global optimum. Let x_c represent the gene to mutate and x_m the mutation result. Thus, a mutation is given by

$$x_m = x_c + N(0, \sigma) \quad (2.31)$$

Although mutation has allowed GA to converge to the true global optimum, GA may still suffer from an excessively slow convergence attempting to identify the global optimum with sufficient precision. The incapability of the algorithm to exploit the local information has been identified as the major cause [67].

2.5 Memetic Algorithm

The trend of SQP as a deterministic algorithm to locate only a local instead of the global optimum and the adversity of GA as a stochastic algorithm to locate the global optimum with sufficient precision despite capability of converging to the true global optimum are effective indications that each of them lacks one of the two central yet competing goals in the design of an optimization procedure: *exploration* and *exploitation* [68]. While the exploration produces some reliable

estimates of the global optimum, the exploitation refines each estimate so as to identify the global optimum with sufficiently high precision. A hybridization of GA and SQP has thus become one feasible alternative to achieving the balance between these two goals.

Inspired by the notion of meme [69], a cultural entity capable of refinement, such hybridization is hence referred to as the *memetic algorithm* (MA) [70, 71]. A number of successful applications of the memetic algorithms have been frequently demonstrated and reported. In [72], MA was used to solve supply chain problems. In [73] and [74], the memetic algorithms were used to tackle scheduling problems. Still in another application [75], MA was used to address chemical process synthesis problems. To solve constrained optimization problems, agent-based MAs have been proposed in [76, 77]. Recently, handling undefined vectors was discussed in [78] whereas managing prediction uncertainty was addressed in [79] in the use of MA for the optimization of expensive problems.

The simplest incorporation of the exploitation into the exploration would intuitively be the interleaving of GA and SQP one after another as portrayed in Algorithm 2.2. The simple MA performs local refinement for each individual in the current population prior to evolving them to establish the next population. As the result, not only it is able to accurately identify the true global optimum but also the simple MA has searched more efficiently than its counterparts [80]: the multiple execution of the SQP with different initial estimates of the global optimum and the purely stochastic GA that relies only on the crossover and the mutation operators to generate better estimates of the optimum.

Algorithm 2.2 Simple Memetic Algorithm

```
Initialize a population of candidate solutions
Evaluate the fitness of each candidate solution
for each candidate solution in the population do
    Refine the candidate solution locally
end for
while no stopping criteria have been fulfilled do
    Rank the candidate solutions in accordance to their fitness values
    Select some candidate solutions based on their rankings for
        crossover and mutation
    Evolve the population through crossover and mutation
    Evaluate the fitness of each candidate solution
    Perform elitism
    for each candidate solution in the population do
        Refine the candidate solution locally
    end for
end while
```

In a single iteration, the FFE requirement of the simple MA is obviously far greater than that of the simple GA. Should each candidate solution be allotted a maximum of ℓ FFEs per local refinement and the population be composed of m candidate solutions, up to $m \times \ell$ FFEs will be required by the simple MA in a single generation while only m FFEs will be required by the simple GA under the same assumptions. Due to the necessarily high computational budget for each of the local refinements, approximation of the objective landscape of an optimization problem [1, 2, 3] has become one alternative to reduce the demands of FFEs.

For the constrained problems, similar concept may be extended and applied to the constraint landscape. A considerable effort for executing expensive exact fitness evaluations during the local refinements can therefore be bypassed. It is, however, well-understood that errors may be resulted from such approximation thereby misleading local refinements to converge to false optima. When solving the unconstrained problems, repairing a false optimum on the actual landscape can indeed be more efficient since the false optimum may in fact be closer than its corresponding initial individual to the actual optimum. The repair can thus be more cost-efficient than complete refinement on the actual landscape. When solving the constrained problems, unfortunately, the compounded errors due to approximation of the multiple landscapes may complicate the repair, making it more susceptible to failing to be more cost-efficient than a complete refinement on the actual landscape. Choice of the individuals to undergo local refinements is therefore crucial to stay cost-efficient. Selective local refinements should not, however, hamper the effectiveness of the MA, thus the need for good strategies. The fitness-based strategies [4] had been employed for adapting the probability of applying local refinements.

Chapter 3

Common Practices in Computer Science

Some common practices of the computer science—which in one way or another have contributed to the proposal and subsequently development of the research works presented in this thesis—will be discussed in this chapter. These include the concept of a *lookup table* and some *nearest neighbors* as well as the method of *Support Vector Machine* for binary classification. In addition to providing a literature review, this chapter also provides some discussions on the usability of these data structures and algorithms in the context of numerical constrained optimization involving computationally expensive problems. Note that discussions in the remaining of this chapter always assume the use of the population-based memetic algorithm while searching for the global optimum of any optimization problem. Individuals in the population represent trial solutions to the problem that undergo some refinements followed by an evolution to the next generation in an attempt to generate better trial solutions. The best individual represents the best solution found so far. The term *individual*, *trial solution*, and *solution* can thus be used interchangeably hereafter.

3.1 Lookup Table

Dated back to long before the advent of computers, the usage of lookup tables has been witnessed in the calculations of complex functions by hand. Covering logarithmic, trigonometric, as well as statistical density functions, using lookup tables speeds up the calculations [81]. In computer science today, lookup table is the data structure that facilitates replacing expensive runtime computations with the simpler table indexing operations. The unnecessary redundancies due to repeated computations of values that have been calculated before can hence be avoided. The savings in terms of processing time can be significant as value retrievals from the memory or even some storage devices are often much faster than repeating the expensive computations. The table might be pre-calculated and stored in the static program storage or some application-specific hardware or could also be calculated as part of the program—a method often referred to as *tabling* or *memoization* [82].

With enormous data generated experimentally daily, the field of computing has recently entered the era of big data [83]. Measuring the quality of the data would be very expensive, let alone performing any optimizations with the data. Consider, for example, the stable structure of biomolecules determined through some wet-lab experiments and deposited, shared in the Protein Data Bank [84]. Parameterized physical model that adopts—among others—the Lennard-Jones potential [85] may be used to describe the data [86]. The parameters, however, need to be optimized to best describe the biomolecules in the collection. There may be only a few parameters to optimize for atoms can be physically grouped

into a few atom types. With tens of thousands of structures deposited to-date, however few parameters are there, evaluating the quality of a set of parameters would unquestionably be extremely expensive. The quality evaluation needs to screen all biomolecules in the collection by calculating all the energy terms due to every possible interatomic interaction in each single biomolecule. Even with only the pairwise atomic interactions taken into consideration, a total of $\mathcal{O}(n^2)$ interactions is implied in the presence of n atoms. In order to find the best set of parameters, the quality measurement needs to be carried out multiple times. Re-evaluating sets of parameters encountered earlier is clearly undesirable.

With the parameters optimized, the resulting model may in turn be used to identify the most stable structure given a biomolecule as it thermodynamically minimizes the potential energy. Proteins, the building blocks of life, often have their sequences known but not their structures. Using some techniques such as *threading* [87], the unknown structure of a protein could be estimated based on the already known structures so far. Intuitively, the estimated structure would unlikely have minimized the potential energy. Extra optimization of the angles that describe some rotatable bonds then needs to be performed. Depending on the number of atoms of the protein as well as the fidelity of the model used, an evaluation of the potential energy of some given structure may take minutes to hours to complete. Multiple evaluations are required in order to find the set of angles that minimizes the energy. With a single evaluation that expensive, it is obvious that re-evaluation of any previously encountered structure is definitely undesirable.

In the context of optimization, maintaining a lookup table throughout the course of searching for the global optimum is often referred to as archiving. For expensive problems, such as those two discussed above, the archive could alleviate the problem of re-evaluation of the trial solutions encountered before. In the same spirit with the calculation of the i -th Fibonacci number with the aid of a lookup table, matching entry in the lookup table is searched for prior to performing the real evaluation of any given query. In such fashion, a great deal of computational costs can easily be eliminated as the expensive real evaluation is only executed in the absence of matching entry in the lookup table. As shown in Figure 3.1, a lookup table plays a crucial role within an optimization framework. Upon requirement of evaluating a trial solution, the optimizer will pass the solution vector to the evaluation manager, which would check for its existence in the lookup table. If the trial solution turns out to be newly encountered, the evaluation manager would perform the real evaluation and subsequently update the lookup table.

Figure 3.1: Lookup Table for Optimization Archives

In the calculation of the i -th Fibonacci number, the lookup table maintains all the smaller $i - 1$ Fibonacci numbers—an exhaustive list of all the necessary numbers in order to calculate the i -th number. Nonetheless, this is impractical in the context of numerical optimizations. For combinatorial problems, the size of the search space grows exponentially with the problem size, *i.e.* the number of decision or design variables to optimize. For continuous problems, moreover, the size of the search space also depends on the resolution of each of the design or decision variable to optimize. Maintenance of an exhaustive list of the entire possible combinations of the variables would simply be intractable. As a result, simple indexing operation in the case of calculating the i -th Fibonacci number would have to be replaced by more sophisticated searching algorithm.

A lookup table in an optimization framework is formalized in Definition 3.1. In the context of continuous constrained optimization problems studied herein, an entry in the lookup table represents a trial solution which consists of design or decision variables to optimize, an objective value, as well as some constraint values. Upon query, a trial solution generated by some optimization algorithm is compared against entries of the lookup table for a perfect match of all of its decision or design variables. If no matching entry is found, the trial solution in query is then evaluated and a new entry fills the lookup table with information about the newly encountered trial solution. In contrast, the stored information about the previously encountered trial solution is retrieved when there exists a matching entry. Afterwards, the corresponding objective and constraint values are returned to the optimizer.

Definition 3.1 (Lookup Table) *Within the framework of some optimization procedure, a lookup table $\mathcal{LT}$ is the set of trial solutions $\mathbf{x}$ encountered thus far and their pertinent evaluation results, which in the case of constrained problems comprise the objective values $f(\mathbf{x})$, the inequality and equality constraint values $\mathbf{g}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$, as well as the derivatives associated with them. Thus,*

$$\mathcal{LT} = \{(\mathbf{x}, f(\mathbf{x}), \mathbf{g}(\mathbf{x}), \mathbf{h}(\mathbf{x}), \nabla f(\mathbf{x}), \nabla \mathbf{g}(\mathbf{x}), \nabla \mathbf{h}(\mathbf{x})) : \mathbf{x} \in \mathbb{R}^n, f : \mathbb{R}^n \rightarrow \mathbb{R}, \mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^{n_g}, \mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^{n_h}\} \quad (3.1)$$

3.1.1 Implementation

In computer, a lookup table can be implemented using an *array* or a *linked-list* as the basic data structure. An array, as we know it, is good at random access. A linked-list, on the other hand, is good at sequential access. Clearly, an entry in the lookup table should be accessible at random. Unlike linked-list, however, array needs a pre-allocated space in the memory to do its work.

Maintaining a lookup table for the optimization of some expensive problem, it would be reasonable to assume some maximum number of entries the lookup table would ever have. With a single evaluation potentially consuming minutes to hours of CPU time, there should arguably be only few thousand evaluations performed within some reasonable time frame. As such, pre-allocation of space in the memory for the array implementation of the lookup table would pose no concern whatsoever. Hence, array is preferred to linked-list when implementing the lookup table within the optimization framework.

3.1.2 Manipulation

For the purpose of maintaining a lookup table in the optimization of expensive problems, two basic operations are necessary in order to be able to manipulate the table successfully. These include *searching* and *appending*. In two separate sections below, each of them will be discussed briefly.

3.1.2.1 Searching

To search for a particular trial solution in the lookup table, each corresponding design or decision variable of the trial solution in query and of those previously encountered solutions in the lookup table is matched against one another using procedure in Algorithm 3.1. Assessing the variables individually, the algorithm starts by assuming all entries match and continues with iteratively eliminating those entries that have one or more differing variables. Due to the sheer size of the search space in the context of continuous optimizations, it is intuitive that the number of matching entries would very likely be reduced significantly after comparison of the first variable. Upon termination, there should only be either one or no matching entries.

Algorithm 3.1 Lookup Table Search Procedure

```
assume all entries are matching entries
while there are multiple matching entries do
    update matching entries by inspecting an unchecked variable
end while
return the matching entry if there is any
```

Assessment of the first variable carries out exactly as many comparisons as the number of entries in the lookup table. Defining $m = |\mathcal{LT}|$, such assessment is thus of $\mathcal{O}(m)$ complexity. In the best case scenario, the number of matching entries following the assessment is reduced to zero after which no comparison would be further required. Next to the best case scenario is when the number of matching entries following the assessment of the first variable is reduced to one after which a number of $\mathcal{O}(n)$ comparisons would be necessary, where n is the number of variables. As high dimensionality is not the focus of the study presented in this thesis, it can therefore be assumed that generally $m \gg n$ nearing the successful identification of the global optimum or upon the exhaustion of the optimization budget. Searching for a particular trial solution in the lookup table is thus of $\mathcal{O}(m)$ complexity in the best case. In the worst case scenario, although extremely unlikely as each of the first $n - 1$ variables must have shared the same value in such a case, the number of matching entries is not reduced before the assessment of the last variable. The search, hence, has a complexity of $\mathcal{O}(mn)$ in the worst case. Note that it is also possible to have the number of matching entries reduced to zero or one later than the first but earlier than the last assessment of variable. In general, the number of matching entries can be reduced to zero or one after $mn - i$ comparisons where $i \in [0, m(n - 1)]$ with the smallest i representing the worst case while the largest i the best case scenario. However, since $m \gg n$ because high dimensionality is not the focus of the study presented herein, searching for a match in the lookup table is generally of $\mathcal{O}(m)$ complexity.

3.1.2.2 Appending

In the absence of matched entry in the lookup table, the trial solution in query would then have to undergo an expensive actual evaluation in order to have its fitness measured in terms of the objective and the constraint values. Following that evaluation, a new entry in the lookup table would subsequently be created and filled in with information concerning the freshly encountered trial solution. As it is implemented in the form of an array, the lookup table simply advances the index of the last entry and stores all relevant values in the respective fields when appending the most recently evaluated trial solution. This solution would consequently become the current final entry of the lookup table. It is therefore clear that appending a new trial solution is of $\mathcal{O}(1)$ complexity.

3.2 Nearest Neighbors

In the preceding sections, it has been shown that a lookup table can be useful for numerical optimization of computationally-expensive problems. With only a few thousand evaluations within some reasonable time frame, maintaining a lookup table is not too resource-demanding compared to the advantage of being able to avoid expensive re-evaluation of the trial solutions encountered earlier. In this section, another advantage gained from maintaining the lookup table will be discussed. It is the definite possibility of extracting the nearest neighbors of a trial solution of interest and using them to approximate the problem's topology in its surroundings.

Nearest neighbors—also known as the k nearest neighbors—are commonly encountered in the predictive analytics for estimation of the class or some property value of an instance based on the consensus of its neighbors. In the area of machine learning, particularly the field of pattern recognition, the nearest neighbors of a previously unseen instance are used to classify the instance into a category or regress its property value. The classification is based on the majority vote of its neighbors with the unseen instance being assigned to the class most common among its k nearest neighbors where k is a typically small, positive integer. The regression, in contrast, is based on some statistics of the property values of its neighbors with the unseen instance most frequently being assigned the average of the property values of its k nearest neighbors. In both cases, the neighbors are taken from a set of instances for which the correct classes or property values are already known.

In the context of numerical constrained optimization, the k nearest neighbors given a trial solution of interest reveal in an approximate sense the topology of the constrained optimization problem that surrounds it. This includes among others the feasibility information as to whether or not a nearby constraint boundary is present as well as the quality information as to whether the trial solution of interest is the best, the worst, or neither in the neighborhood. Knowing the problem's topology consequently allows focussing the search efforts around the possible locations of the local optima; one of which shall be the global optimum. This is particularly important in the context of computationally expensive problems for efficiency reasons.

In the following, the notion of nearest neighbors will first be formalized in mathematical terms in order to understand how they approximate the neighborhood of a trial solution of interest. Next, identification of the nearest neighbors from the lookup table will be elaborated. Finally, the possible characteristics of the nearest neighbors and the neighborhood they approximate will be discussed.

3.2.1 Definition

Definition 3.2 (Nearest Neighbors) *Given any trial solution $\mathbf{x}$ and a lookup table $\mathcal{LT}$ as defined in Equation (3.1), the k nearest neighbors of $\mathbf{x}$ in $\mathcal{LT}$ for typically small, positive integer k , some distance measure d , and an ascending ranking scheme $\mathcal{R}$, is given by*

$$\mathcal{N}_k(\mathbf{x}) = \{\mathbf{z} : \mathbf{z} \in \mathcal{LT}, \mathbf{z} \neq \mathbf{x}, \mathcal{R}(d(\mathbf{z}, \mathbf{x})) \leq k\} \quad (3.2)$$

In Definition 3.2, the notion of nearest neighbors is formalized. The resulting neighborhood is constituted by k already evaluated trial solutions sustaining the first k distances in increasing order to the trial solution $\mathbf{x}$ of interest. Based on Definition 3.2, it is obvious that $d(\mathbf{z}, \mathbf{x}) \leq d(\tilde{\mathbf{z}}, \mathbf{x})$ for all $\mathbf{z} \in \mathcal{N}_k(\mathbf{x})$ and $\tilde{\mathbf{z}} \in \mathcal{LT} \setminus \{\mathcal{N}_k(\mathbf{x}) \cup \{\mathbf{x}\}\}$. Used throughout this research work, the Euclidean distance measures the sparseness between any two candidate solutions. In n -dimensional space, with $\mathbf{p}$ and $\mathbf{q}$ being any two n -dimensional vectors having p_i and q_i as their respective i -th elements, the Euclidean distance is defined as

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (3.3)$$

Over the course of solving a constrained optimization problem using the population-based memetic algorithm, multiple trial solutions are generated and evaluated in the attempt to reveal the topology of the problem—that is, the landscape of its objective and constraint functions—by means of intelligent or educated sampling. Therefore, the k nearest neighbors of a particular trial solution $\mathbf{x}$ in fact samples the neighborhood of $\mathbf{x}$. As described in Definition 3.3, this neighborhood is often referred to as the ε -neighborhood of $\mathbf{x}$ with ε being the distance between the trial solution $\mathbf{x}$ and its k -th nearest neighbor. In the next chapter, this neighborhood will be used to derive the telltale signs of being in the immediacy of some optimum. The nearest neighbors information will then be used to approximate this neighborhood in the attempt to perform efficient optimization with selective local refinements.

Definition 3.3 (Neighborhood) *Given any trial solution $\mathbf{x}$ and a positive ε , the neighborhood—or often referred to as the ε -neighborhood—of $\mathbf{x}$ is defined as*

$$\mathcal{N}_\varepsilon(\mathbf{x}) = \{\mathbf{z} : \mathcal{R}(d(\mathbf{x}^i, \mathbf{x})) \leq k\} \quad (3.4)$$

3.2.2 Identification

To identify the k nearest neighbors $\mathbf{z}_i$ for $i = 1, \dots, k$ of any trial solution $\mathbf{x}$ of interest from the lookup table $\mathcal{LT}$, the procedure in Algorithm 3.2 is followed. This involves the following three steps: (1) calculating distances, (2) sorting based on distances in ascending order, (3) selecting the top k distances.

Algorithm 3.2 k Nearest Neighbors Search Procedure

```
for each trial solution  $\mathbf{x}$  in the lookup table do  
    calculate the distance between  $\mathbf{x}$  and the trial solution  $\hat{\mathbf{x}}$  of interest  
end for  
sort trial solutions in the lookup table based on the distances  
return trial solutions with the top  $k$  distances
```

In the distance calculation, the difference between every independent variable x_i of each of the trial solution $\mathbf{x}$ in the lookup table $\mathcal{LT}$ and that of the trial solution $\hat{\mathbf{x}}$ of interest need to be computed. A number of $\mathcal{O}(mn)$ operations are therefore needed recalling that $m = |\mathcal{LT}|$ and n is the number of design or decision variables. Using quicksort [82], a complexity of $\mathcal{O}(m \log m)$ is obtained in the best and average cases and $\mathcal{O}(m^2)$ in the worst case. Selecting the top k distances is easily of $\mathcal{O}(1)$ complexity. With $m \gg n$ as high dimensionality is not the focus herein, the complexity of the sorting algorithm prevails and dictates the complexity of nearest neighbors identification.

3.2.3 Characterization

Information stored in the lookup table that corresponds to the k nearest neighbors of a particular solution $\mathbf{x}$ characterizes the neighborhood they sample. This includes the objective and constraint values as well as their derivatives. For the purpose of characterization in this section, we will focus on the constraint values alone and leaves the other quantities for further discussions as they become necessary in the upcoming sections.

In general, a constraint function maps a solution vector of n real values to a single real constraint value. A constraint value can therefore be negative, zero, or positive. The set of solutions with zero constraint values defines the constraint boundary with respect to that constraint as depicted in Figure 3.2. For equality constraint, this boundary separates two types of infeasible solutions: one with positive values, another with negative values. For inequality constraint, it separates feasible from infeasible solutions. The ε -neighborhood of any solution $\mathbf{x}$ may therefore be composed of all zero, all negative, all positive, all non-positive, all non-negative, or all possible real constraint values as illustrated by neighborhood A, B, C, D, E, and F, respectively, in Figure 3.2. For equality constraint, uniform neighborhoods are observed in neighborhoods A, B, and C. Otherwise, it is composite. For inequality constraint, as zero and negative constraint values represent feasible solution, uniform neighborhoods are observed in neighborhoods A, B, C, and D. Otherwise, it is composite. The k nearest neighbors sample this epsilon neighborhood and may therefore be similarly uniform or composite. In the following, we shall discuss the implications of certain types of uniform or composite nearest neighbors. As with any other sampling, some information may be missed, in particular when the neighbors are very sparsely distributed. However, as the search goes by, the approximation becomes more and more accurate as the search space is getting more and more densely populated. In the following discussions, it will be witnessed how the different compositions of the k nearest neighbors may become the necessary or sufficient conditions for different types of ε -neighborhood of $\mathbf{x}$.

Figure 3.2: Possible Neighborhood Structures

3.2.3.1 Uniform Neighbors

Uniform k nearest neighbors of the same quality or category as that of the trial solution $\mathbf{x}$ with respect to which the neighbors were defined is the necessary condition for the uniform ε -neighborhood of $\mathbf{x}$ with that particular quality or category. The presence of just one member belonging to different class would deem the neighborhood non-uniform. Conversely, however, uniform k nearest neighbors of the same quality or category as that of the trial solution $\mathbf{x}$ with respect to which the neighbors were defined is not sufficient to guarantee a uniform ε -neighborhood of $\mathbf{x}$ with that particular quality or category. The k nearest neighbors sample the ε -neighborhood of $\mathbf{x}$ finitely and may therefore fail to capture the area in the neighborhood where trial solutions—if evaluated—would turn out to be of different class.

In the context of numerical optimization, uniform nearest neighbors may be encountered in three different scenarios. They may be all feasible, all positively infeasible, or all negatively infeasible. Upon encountering either one of these categories of the nearest neighbors, it is impossible to tell precisely if the individual $\hat{\mathbf{x}}$ with respect to which the neighbors were extracted is actually close to the constraint boundary. What is known for sure is that the individual seems to be in the interior of either space, depending on the constraint value of the individual and the type of constraint against which the neighborhood is characterized. Whether or not the individual is actually in the interior of the feasible space, undergoing local refinement could be beneficial as there could be an optimum therein. In contrast, should the individual be actually in the interior of either infeasible space, performing local refinement would lead to a waste of search effort. Although there is no guarantee that these neighbors actually signify a truly uniform ε -neighborhood, it would be reasonable to assume so until it is proven otherwise.

3.2.3.2 Composite Neighbors

Composite k nearest neighbors of at least two different qualities or categories are both the necessary and the sufficient conditions of a composite ε -neighborhood of a trial solution $\mathbf{x}$ with respect to which the neighbors were defined. This is regardless of the quality or category of $\mathbf{x}$ itself. In fact, uniform k nearest neighbors of different quality or category from that of the trial solution $\mathbf{x}$ is also the necessary and the sufficient conditions of a composite ε -neighborhood of $\mathbf{x}$.

No matter what qualities or categories compose the neighbors, presence of the boundary of the constraint against which the neighbors are characterized is implied. When either the individual $\mathbf{x}$ or any of its neighbors $\mathbf{x}_i$ for $i = 1, \dots, k$ is feasible, there must exist one or more infeasible members in the composite neighborhood. This signifies the presence of the boundary that separates the feasible from the infeasible solution spaces. When members of the neighborhood comprise of some positively infeasible solutions and some negatively infeasible solutions with respect to an equality constraint, the boundary of the constraint is also signified as the continuity of the constraint function implies there has to be trial solutions with zero constraint value between the positive infeasible space and the negative infeasible in accordance with the intermediate value theorem [88].

3.3 Support Vector Machine

From the lookup table discussed earlier in this chapter, the nearest neighbors of any trial solution of interest can be extracted as discussed in the preceding section. These neighbors may signify a composite ε -neighborhood of the trial solution of interest; within which lies a constraint boundary that separates the feasible from the infeasible spaces or the positive from the negative infeasible spaces in the context of equality constraint. Separation of two and only two classes within the neighborhood suggests that a two-class classification problem can be formulated with the nearest neighbors serving as the set of the training data.

Machine-learning technique originally proposed as a linearly separable binary classification problem solver [89] is the Support Vector Machine (SVM) [90, 91]. Characterized by its ability not only to minimize the classification error but also to maximize the geometric margin of separation that sets the two classes apart, the SVM can be used for solving the two-class classification problem that may easily arise from a composite nearest neighbors. On the provision of a set of k training data instances, $\{(\mathbf{x}^i, y^i) : i = 1, 2, \dots, k\}$, the SVM will aim at identifying an appropriate set of coefficients, $\{\mathbf{w}, w_0\}$, that defines a linear decision function

$$D(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + w_0 \quad (3.5)$$

that satisfies

$$\forall i \ y^i D(\mathbf{x}^i) \geq 1 \quad (3.6)$$

where $y^i \in \{+1, -1\}$.

The region in which none of the training data instances may reside is called the margin-of-separation. Given by all $\mathbf{x}$ satisfying $|D(\mathbf{x})| < 1$, it can easily be shown that the margin-of-separation has a width of $\frac{2}{\|\mathbf{w}\|}$. Hence, maximizing the margin-of-separation is equivalent to minimizing

$$\frac{1}{2} \|\mathbf{w}\|^2 \quad (3.7)$$

subject to

$$\forall i \ y^i [(\mathbf{w} \cdot \mathbf{x}^i) + w_0] \geq 1 \quad (3.8)$$

in which the factor of $\frac{1}{2}$ is introduced only for mathematical convenience.

The corresponding dual to the above primal problem is the maximization of a quadratic program

$$\sum_{i=1}^k \alpha^i - \frac{1}{2} \sum_{i=1}^k \sum_{j=1}^k \alpha^i \alpha^j y^i y^j (\mathbf{x}^i \cdot \mathbf{x}^j) \quad (3.9)$$

subject to

$$\sum_{i=1}^k \alpha^i y^i = 0 \quad (3.10)$$

$$\forall i \alpha^i \geq 0 \quad (3.11)$$

Should a training data instance $\mathbf{x}^i$ has $\alpha^i > 0$, it is then the support vector. Each support vector occupies either decision surface, $D(\mathbf{x}) = +1$ or $D(\mathbf{x}) = -1$, depending on which class it belongs to. With all the support vectors identified, the weight vector $\mathbf{w}$ and the bias w_0 can then be computed using

$$\begin{aligned} \mathbf{w} &= \sum_{i=1}^k \alpha^i y^i \mathbf{x}^i \\ &= \sum_{i \in SV} \alpha^i y^i \mathbf{x}^i \end{aligned} \quad (3.12)$$

and

$$w_0 = \frac{1}{|SV|} \sum_{i \in SV} \left[y^i - \sum_{j=1}^k \alpha^j y^j (\mathbf{x}^i \cdot \mathbf{x}^j) \right] \quad (3.13)$$

in which SV denotes the set of indices of the support vectors.

While the SVM uses linear decision function, the actual boundary that separates one class from another may be highly nonlinear. Kernel tricks [92] can be used to accommodate the necessity of nonlinear decision functions [93]. This, however, requires correct selection of the kernel function which is quite difficult in the absence of any background knowledge on the classification problem.

In the context of constrained optimization studied herein, linear SVM should suffice as within sufficiently localized neighborhood, the constraint boundary would be close to linear. Nevertheless, the SVM may indeed fail to construct a model using linear decision function. This may be an indication of too sparse or too dense distribution of nearest neighbors. Too sparse nearest neighbors normally happens during early generations of the search. Too dense distribution of nearest neighbors, on the other hand, signifies extreme closeness to the constraint boundary. In both cases, exploitation is significant. In the event of successful construction of a model by the SVM, nonlinearity of the constraint boundary would generally result in misclassification. Even in the case of linear constraint boundary, distribution of the training data could easily result in misclassification as this affect the produced decision function. Fortunately, however, misclassification generally signifies closeness to the actual constraint boundary. Additionally, the margin-of-separation produced by the SVM likewise signifies closeness to the constraint boundary. Exploitation is therefore significant. More detail about these issues will be elaborated in the next chapter where the efficient constraint-oriented refinement-efficacious memetic algorithm paradigm will be proposed.

Chapter 4

Constraint-Oriented Refinement-Efficacious Memetic Algorithm

Representing new paradigm in the adaptive memetic algorithm, the constraint-oriented refinement-efficacious memetic algorithm (CORE-MA) to be presented in this chapter rudimentarily aims at eliminating unnecessary local refinements which would have been performed under the simple MA framework yet only to re-exploit regions on the search space that have been exploited earlier. Prior to undergoing a local refinement, an individual in the population shall go through significance assessment to determine whether or not its refinement would worth the search effort. Under the CORE-MA framework, two classes of significance assessment strategies would be elaborated in detail later in this chapter. These comprise the *Feasibility Structure Modeling* (FSM) and the *Constraint Landscape Stratification* (CLS) as portrayed in Figure 4.1. The earlier gives rise to three different instances of MA, namely the original FSM-driven MA, the generalized FSM-driven MA, and the ecumenical FSM-driven MA. The latter, on the other hand, only gives rise to one specific instance of MA, the CLS-powered MA.

Figure 4.1: CORE-MA Instantiation Scheme

In the FSM, as will be elaborated later, classification approach is used not to model the approximate objective and constraint functions in the vicinity of an individual but to predict the relative location of that individual in the search space. The original FSM as reported in [94, 95] works only for inequality-constrained problems whereas the generalized FSM as reported in [96, 97] caters the tiny ratio of the feasible to the entire search space of the equality-constrained problems. The ecumenical FSM, which will be reported in the immediate future, addresses the robustness issues of the earlier reported FSM-driven MA instances. In the CLS, gradient information of the objective as well as the constraint functions is utilized to stratify the search space of a constrained optimization problem. Preliminarily applied in [98] for molecular docking problem, the CLS will first be conceptualized for the general constrained optimization problems and then instantiated as the CLS-powered MA herein.

Presented in Algorithm 4.1 is a framework of the adaptive MA that enables selective local refinements. Using appropriate assessment strategies proposed in this chapter, many unnecessary local refinements shall be eliminated—reducing the number of evaluations performed in the search for the global optimum. For computationally-expensive problems, reduction of tens to hundreds evaluations easily translates into finding the optimal solutions days ahead.

Algorithm 4.1 Adaptive Memetic Algorithm

```

while no stopping criterion is met
    if empty population
        initialize population
    else
        evolve population through crossover and mutation
    end if
    evaluate individuals in population observing elitism
    for each individual in population
        assess significance of individual for local refinement
        if significant individual
            refine individual
        end if
    end for
end while

```

4.1 The Roadmap

To guide readers through the considerable length of the chapter, a brief history of the CORE-MA development will first be presented in this section. The links between the various stages of the development and their corresponding sections will be established.

Development of the novel paradigm of CORE-MA started with the concept of Optimum Immediacy Corollary discussed in Section 4.2.2, advising that it is possible to learn the local topology of the problem to hence determine whether an individual is worth refining based on the information revealed by its nearest neighbors. In addition, the development of the CORE-MA partially began with the notion of the In-feasibility Class discussed in Section 4.2.1. All these finally gave rise to the Feasibility Structure Modeling (FSM) presented in Section 4.3. The original FSM-driven MA proposed in Section 4.3.3.1 is the first instance of the CORE-MA used to efficiently solve the inequality-constrained optimization problems. The generalized FSM-driven MA proposed in Section 4.3.3.2 later on extended the original FSM-driven MA to additionally be capable of addressing equality constraints thereby efficiently solving general constrained optimization problems. Finally, the ecumenical FSM-driven MA presented in Section 4.3.3.3 addressed the robustness issue that may arise from the generalized FSM-driven MA. In modeling the local topology of the problem, the FSM for the first time formulates a classification problem within an optimization framework and then solves the arising problem using the Support Vector Machine (SVM) which can obviously incur huge overhead, especially in the generalized and the ecumenical FSM-driven MAs. The Constrained Landscape Stratification (CLS) elaborated in Section 4.4 then gave rise to the second CORE-MA instantiation, addressing the overhead issue. The CLS-powered MA proposed in Section 4.4.3 eliminates the requirement of employing the SVM in exchange with some readily available gradients information.

4.2 Preliminaries

Proposals of the *Feasibility Structure Modeling* and the *Constrained Landscape Stratification* later in this chapter as two instances of the assessment strategies that allow the efficient CORE-MA paradigm are rudimentarily inspired by two elemental concepts: *in-feasibility class* and *optimum immediacy corollary*.

4.2.1 In-feasibility Class

Over the course of searching for the global optimum given a constrained optimization problem, one would generally encounter feasible and infeasible solutions. The feasible solutions are feasible with respect to all constraints. They compose the feasible solution space(s) in the search space of the problem. The infeasible solutions, on the other hand, are infeasible with respect to one or more constraints. They populate the remaining space(s) in the problem's search space. A trial solution $\mathbf{x}$ to the constrained optimization problem therefore belongs to either the *feasible* or the *infeasible* class with respect to a constraint. This applies to both the equality $\mathbf{h}(\mathbf{x}) = \mathbf{0}$ and the inequality $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ constraints. With respect to the inequality constraints they violate, the infeasible solutions must assume positive constraint values. With respect to the equality constraints they violate, however, the infeasible solutions may be of positive or negative constraint values. A trial solution $\mathbf{x}$ to the constrained optimization problem thus additionally belongs to either the *positive* or the *negative* infeasible class with respect to an equality constraint. Based on the foregoing discussions, Definition 4.1 presents the mathematical formulation of the in-feasibility class.

Definition 4.1 (In-feasibility Class) *The in-feasibility class $\mathcal{IFC}_c(\mathbf{x})$ given a trial solution $\mathbf{x}$ to the constrained optimization problem $\mathcal{COP}$ with respect to some constraint c is defined as*

$$\mathcal{IFC}_c(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \text{ is feasible} \\ \text{sign}(c(\mathbf{x})) & \text{if } \mathbf{x} \text{ is infeasible} \end{cases} \quad (4.1)$$

An equality constraint divides the entire solution space into the feasible, the positive infeasible, and the negative infeasible spaces. With respect to an equality constraint, a trial solution may therefore be of feasible, positive infeasible, or negative infeasible class. Composed of solutions with zero constraint value, the feasible space clearly occupies only a tiny fraction of the entire search space. In the early generations of the search for the global optimum using some population-based stochastic algorithm, sampling a feasible trial solution with respect to an equality constraint would intuitively be difficult. Depending on the ratio between the spaces of the two infeasible classes, it could also be difficult to identify a trial solution belonging to either class. When there is only one of the two infeasible classes out of the three possible in-feasibility classes with respect to an equality constraint in the lookup table, it is said that the trial solutions are of *insufficient diversity* with respect to that particular constraint. After evolution over a number of generations and some refinements, one or more feasible solutions could have been identified. Should the other infeasible class be still unidentified, it is then said that the constraint is possibly of *restricted function mapping*.

In the context of an inequality constraint, the entire solution space is divided into the feasible and the positive infeasible spaces. With respect to an equality constraint, therefore, a trial solution may be of feasible or positive infeasible class. Unlike in the case of an equality constraint, the feasible space in the context of an inequality constraint generally occupies a reasonably large fraction of the entire search space. If indeed the feasible space occupies only a tiny fraction of the entire search space, it would be difficult to sample a feasible trial solution with respect to the inequality constraint. With only trial solutions in the lookup table belonging to the infeasible class, it is also said that there is *insufficient diversity* with respect to that particular constraint until a feasible trial solution is successfully sampled. Should all the feasible solutions sampled are of zero constraint value, it is then said that the constraint is possibly of *restricted function mapping*. The above discussion also applies to the aggregate which assumes the maximum constraint value in the presence of multiple inequality constraints.

4.2.2 Optimum Immediacy Corollary

In this subsection, the ensuing consequences of being in the close proximity of or immediate to an optimal solution will be derived. The corollary will be used in the subsequent sections for orientating the search efforts towards regions of the search space where local optima might potentially reside. Identification of these local optima is crucial as the global optimum is none other than one of the possibly many local optima. Before proceeding any further, the notion of optimum immediacy will first be formalized in Definition 4.2.

Definition 4.2 (Optimum Immediacy) *A solution $\mathbf{x}$ is defined as being in the optimum immediacy if and only if there exists some local or possibly global optimum $\mathbf{x}^*$ and arbitrarily small ε such that*

$$\|\mathbf{x} - \mathbf{x}^*\| < \varepsilon \quad (4.2)$$

The concept of optimum immediacy is of importance as rapid convergence of the deterministic optimization techniques is based on this assumption [22], i.e. they are initiated with estimates that are sufficiently close to an optimum. It is indeed impossible to know a priori with definite certainty if an estimate is actually close to an optimum as such knowledge would require identification of the exact location of the optimum, and hence, render the local search by any deterministic optimization method unnecessary. It is, nonetheless, possible to suggest if an estimate could actually be close to an optimum through observation of the neighborhood of the estimate. As an alternative to Definition 4.2, the solution $\mathbf{x}$ is in the optimum immediacy if the optimum $\mathbf{x}^*$ is in the ε -neighborhood of $\mathbf{x}$. This, in turn, implies that the ε -neighborhood of $\mathbf{x}$ is in the δ -neighborhood of $\mathbf{x}^*$ where $\delta = 2\varepsilon$ as illustrated in Figure 4.2. Characteristics of the δ -neighborhood of $\mathbf{x}^*$ thus provides what will be the telltale signs should an estimate be in the optimum immediacy. Conversely, the characteristics would then serve as the necessary condition for the optimum immediacy.

Figure 4.2: An illustration of the ε -neighborhood of solution $\mathbf{x}$, $\mathcal{N}_\varepsilon(\mathbf{x})$, and the δ -neighborhood of optimum $\mathbf{x}^*$, $\mathcal{N}_\delta(\mathbf{x}^*)$, with $\delta = 2\varepsilon$ in the optimum immediacy.

For the purpose mentioned above, the telltale signs of being in the optimum immediacy will first be derived separately in the following for optimum at the trough of a feasible basin and at the feasibility boundary. The formal corollary will then be presented under the lack of knowledge of the location of the optimum, summarizing the observations made thus far.

4.2.2.1 Optimum in Feasible Space Interior

Resembling the unconstrained ones, certain constrained optimization problems have their optimum situated at the trough of some feasible basin, minimizing the objective value locally. In general, only constrained optimization problems with inequality constraints $\mathbf{g}$ may have an optimum $\mathbf{x}^*$ at the trough of some feasible basin. It is when the optimum satisfies the strict inequality $\mathbf{g}(\mathbf{x}^*) < \mathbf{0}$ in the absence of the equality constraints as illustrated in Figure 4.3.

Figure 4.3: An illustration of some optimum $\mathbf{x}^*$ at the trough of a feasible basin with its δ -neighborhood $\mathcal{N}_\delta(\mathbf{x}^*)$.

Based on the illustration in Figure 4.3, δ -neighborhood of the optimum $\mathbf{x}^*$ will then be investigated. As $\mathbf{x}^*$ minimizes the objective value in the respective feasible basin, $\forall \mathbf{x} \in \mathcal{N}_\delta(\mathbf{x}^*)$ $f(\mathbf{x}^*) \leq f(\mathbf{x})$, $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ when δ is sufficiently small. $\mathcal{N}_\delta(\mathbf{x}^*)$ is therefore a uniform neighborhood of feasible solutions with minimum objective value of $f(\mathbf{x}^*)$ when δ is sufficiently small. When δ is sufficiently large, however, $\mathcal{N}_\delta(\mathbf{x}^*)$ would also engage the neighboring feasible basin—particularly in the case of multimodal objective landscape—in which another solution with lower objective value could possibly be identified. This suggests that objective values may not be of direct expediency to the optimum immediacy. For an even larger δ , $\mathcal{N}_\delta(\mathbf{x}^*)$ may also engage infeasible solutions making the neighborhood composite. With respect to each individual inequality constraint, $\mathcal{N}_\delta(\mathbf{x}^*)$ may understandably be a composite or a uniform neighborhood of feasible solutions depending on the value of δ .

Following the observation of the δ -neighborhood of the optimum $\mathbf{x}^*$ will be the characterization of the ε -neighborhood of some estimate $\mathbf{x}$ in the optimum immediacy. Recall that $\mathcal{N}_\varepsilon(\mathbf{x}) \subset \mathcal{N}_\delta(\mathbf{x}^*)$ when $\mathbf{x}$ is in the optimum immediacy. When $\mathcal{N}_\delta(\mathbf{x}^*)$ is a uniform neighborhood of feasible solutions, so will $\mathcal{N}_\varepsilon(\mathbf{x})$ be. When $\mathcal{N}_\delta(\mathbf{x}^*)$ is a composite neighborhood, $\mathcal{N}_\varepsilon(\mathbf{x})$ may similarly be a composite neighborhood. $\mathcal{N}_\varepsilon(\mathbf{x})$, however, can also be a uniform neighborhood of feasible solutions when no constraint boundary passes through it. As such, $\mathcal{N}_\varepsilon(\mathbf{x})$ may in conclusion be a composite or a uniform neighborhood of feasible solutions in the immediacy to some optimum at the trough of some feasible basin. In other words, one should necessarily be able to find solutions belonging to the *feasible* class in the ε -neighborhood of an estimate $\mathbf{x}$ if $\mathbf{x}$ is in the optimum immediacy. Occasionally, but not necessarily, one may also encounter solutions that belong to the *infeasible* class in the neighborhood. By contrapositive, finding solutions that are in the *infeasible* class alone throughout the entire neighborhood, *i.e.* $\forall \mathbf{z} \in \mathcal{N}_\varepsilon(\mathbf{x}) \mathbf{g}(\mathbf{z}) > \mathbf{0}$, implies $\mathbf{x}$ is not in the optimum immediacy.

As discussed earlier in Section 3.2, feasibility of all the k nearest neighbors of some estimate $\mathbf{x}$ is the necessary condition for the uniform neighborhood of feasible solutions though it may actually refer to some composite neighborhood in reality. Clearly, feasibility of only some of the k nearest neighbors serves as the sufficient condition for the composite neighborhood. Having all or some of the k nearest neighbors feasible could therefore be a sign of the presence of an optimum at the trough of a feasible basin within $\mathcal{N}_\varepsilon(\mathbf{x})$; the radius ε of which is dictated by the distance between the k -th nearest neighbor and the estimate.

Having all the k nearest neighbors infeasible, on the contrary, is the necessary condition for a uniform neighborhood of infeasible solutions which contains no optimum. Although it may actually refer to some composite neighborhood in reality, the lack of feasible neighbors provides no guarantee at all for a really composite neighborhood. Further exploitation by means of some deterministic optimization techniques on this neighborhood should intuitively be deferred or applied restrictively.

4.2.2.2 Optimum at Feasible Space Boundary

Whenever one or more constraint boundaries cause any lower objective value inaccessible despite its existence or coincide with the trough of some partially feasible basin, any constrained optimization problems must consequently have their optimum located at the feasibility boundary. The feasibility boundary is generally determined by the intersection of the boundaries of probably some of the inequality constraints $\mathbf{g}$ and all of the equality constraints $\mathbf{h}$. The boundary is mathematically defined as the following set:

$$\{\mathbf{x} : \exists i \forall j \ j \neq i, \ g_i(\mathbf{x}) = 0, \ g_j(\mathbf{x}) < 0 \ \vee \ \mathbf{g}(\mathbf{x}) < \mathbf{0} \ \wedge \ \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$$

The feasibility boundary hence distinguishes the space of the feasible solutions from that of the infeasible solutions. When the equality constraints are present, the boundary might intuitively be a mere single point in the vast search space, rendering inspection by the feasibility class—like in the case of the optimum at the trough of some feasible basin—impractical.

Figure 4.4: An illustration of some optimum $\mathbf{x}^*$ at the feasibility boundary with its δ -neighborhood $\mathcal{N}_\delta(\mathbf{x}^*)$.

As hinted by the illustrative examples in Figure 4.4, the δ -neighborhood of the optimum $\mathbf{x}^*$ at the feasibility boundary must be a composite neighborhood in terms of feasibility. With respect to each individual constraint, intuitively, a composite $\mathcal{N}_\delta(\mathbf{x}^*)$ is mandatory for each equality and every active inequality constraint while either a composite or a uniform $\mathcal{N}_\delta(\mathbf{x}^*)$ of feasible solutions is consequential from every inactive inequality constraint depending on the value of δ . In addition to being composite in terms of feasibility, $\mathcal{N}_\delta(\mathbf{x}^*)$ should also be composite in terms of infeasibility with respect to every equality constraint in general.

Any optimum $\mathbf{x}^*$ at the feasibility boundary in the presence of one or more equality constraints $\mathbf{h}$ is almost always attributable to the equality constraints alone, that is $\mathbf{x}^*$ must solve $\mathbf{h}$ in addition to optimizing some objective value in the interior of the feasible solution space with respect to inequality constraints $\mathbf{g}$ as depicted by Figure 4.4b. With $\mathbf{g}(\mathbf{x}^*) < \mathbf{0}$, as with the case of optimum at the trough of some feasible basin, $\mathcal{N}_\delta(\mathbf{x}^*)$ and $\mathcal{N}_\varepsilon(\mathbf{x})$ when $\mathbf{x}$ is in the optimum immediacy may be a *composite* or a *uniform* neighborhood of *feasible* solutions with respect to each individual as well as the aggregated inequality constraints. In the absence of equality constraints, the optimum $\mathbf{x}^*$ must be attributable to one or more inequality constraints. Boundaries of these constraints are actively obstructing access to the solution space beyond them. Lower objective value in that particular solution space—if any—is thus unattainable due to infeasibility. With respect to each of the active inequality constraints, $\mathcal{N}_\delta(\mathbf{x}^*)$ unmistakably is a *composite* neighborhood as the respective constraint boundary passes right through it. When an estimate $\mathbf{x}$ is in the optimum immediacy, $\mathcal{N}_\varepsilon(\mathbf{x})$ similarly is a *composite* neighborhood as it covers both the optimum $\mathbf{x}^*$ and segments of the constraint boundaries in its interior by definition.

Thus far, the composite neighborhoods have been observed only in terms of feasibility. Equality constraints, however, are highly restrictive with regards to the feasibility. As opposed to the inequality constraints in n -dimensional space whose feasible solutions are also of n -dimensional space, an equality constraint in the n -dimensional space has feasible solutions of a lower $(n - 1)$ -dimensional space, separating in general the solutions in *negative infeasible* class from those

in the *positive infeasible* class. In a two-dimensional plane, solutions to a single equality constraint form a one-dimensional curve. In a three-dimensional cube, solutions to an equality constraint form a two-dimensional surface. As depicted by Figure 4.5, the feasible solution space of an equality constraint—which also is the constraint boundary—separates the space of *negative infeasible* solutions from that of the *positive infeasible* ones. Hence, with respect to each individual equality constraint, $\mathcal{N}_\delta(\mathbf{x}^*)$ is a *composite* neighborhood both in feasibility and in infeasibility. Given an estimate $\mathbf{x}$ in the optimum immediacy, $\mathcal{N}_\epsilon(\mathbf{x})$ likewise is a *composite* neighborhood—both in terms of feasibility and infeasibility—for the optimum $\mathbf{x}^*$ as well as segment of the constraint boundary are by definition in its interior. Being composite in terms of feasibility and infeasibility suggests that whenever one finds one or more neighbors that belong to the *feasible* class and few in the *infeasible* class or some neighbors in the *positive infeasible* class and some belonging to the *negative infeasible* class, the constraint boundary of an equality constraint is implied.

Figure 4.5: Equality Constraints in Two- and Three-dimensional Space

4.2.2.3 Optimum at Either Location

In two preceding sections above, the interior as well as the boundary optimum immediacy corollaries have been elaborated in great detail. From the foregoing discussions, Corollary 4.1 then summarizes the ensuing consequences of being in the optimum immediacy for the case of optimum at the trough of some feasible basin and that of the optimum at the feasibility boundary. Hence, it formalizes a general optimum immediacy corollary in the absence of any knowledge about the location of the optimum—a situation most commonly encountered.

Corollary 4.1 (Optimum Immediacy Corollary) *Given any estimate $\mathbf{x}$ in the immediacy of some optimum $\mathbf{x}^*$, i.e. $\|\mathbf{x} - \mathbf{x}^*\| < \varepsilon$, then*

1. *with respect to any individual equality constraint h , $\mathcal{N}_\varepsilon(\mathbf{x})$ is a composite neighborhood in terms of feasibility and infeasibility in general, i.e.*

$$\exists \mathbf{p}, \mathbf{q} \in \mathcal{N}_\varepsilon(\mathbf{x}) [h(\mathbf{p}) = 0, h(\mathbf{q}) \neq 0 \vee h(\mathbf{p}) < 0, h(\mathbf{q}) > 0] \quad (4.3)$$

2. *with respect to every individual inequality constraint g , $\mathcal{N}_\varepsilon(\mathbf{x})$ is either a composite neighborhood in terms of feasibility or a uniform neighborhood of feasible solutions, i.e.*

$$\exists \mathbf{p}, \mathbf{q} \in \mathcal{N}_\varepsilon(\mathbf{x}) g(\mathbf{p}) \leq 0, g(\mathbf{q}) > 0 \vee \forall \mathbf{p} \in \mathcal{N}_\varepsilon(\mathbf{x}) g(\mathbf{p}) \leq 0 \quad (4.4)$$

In essence, Corollary 4.1 states that one should be able to encounter either a sample of feasible solutions or samples of both the positive and the negative infeasible solutions in the neighborhood $\mathcal{N}_\varepsilon(\mathbf{x})$ of an estimate $\mathbf{x}$ if the estimate is in the immediacy to an optimum. Should one be able to find only samples of either the entirely positive or the entirely negative infeasible solutions in the neighborhood, the estimate could not be in the immediacy to an optimum. There could not be an optimum in such neighborhood as the matter of fact. Harnessing the information revealed through the evaluation of some previously encountered trial solutions in the vicinity of the estimate, which are its k nearest neighbors, the following should be noted. For an optimum in the feasible space interior, it would generally be easy to encounter some feasible solutions unless the trough of the feasible basin on which the optimum lies is close to the feasible space boundary. In such a case, the ratio between the feasible solutions to the infeasible ones with respect to the strictest inequality constraint within the neighborhood $\mathcal{N}_\varepsilon(\mathbf{x})$ could easily be very low. This makes the encounter with a sample of the feasible solutions in the neighborhood rather difficult. For the optimum at the feasible space boundary, the above scenario may be more easily encountered noting that the imbalanced ratio also applies to the positive and the negative infeasible solutions with respect to the equality constraints. These are indeed the inherent characteristics of approximation using k nearest neighbors. As nearest neighbors with uniform infeasibility are the necessary condition for ε -neighborhood of uniform infeasibility in which no optimum can be found, local refinements should be deferred or applied restrictively.

In the remaining of this chapter, the consequences of being in the immediacy of some optimum as given in Corollary 4.1 will be used as heuristics to instantiate two assessment strategies that would enable the efficient CORE-MA paradigm: *Feasibility Structure Modeling* and *Constrained Landscape Stratification*.

4.3 Feasibility Structure Modeling

Feasibility Structure Modeling (FSM) is a conceptual framework that, based on some or all of the trial solutions encountered so far, accommodates the deduction of the feasibility structure of an optimization problem. Equipped with a dynamic model of the feasibility structure that evolves over the course of the search for the global optimum solution, focusing the search effort on regions of the search space of the constrained optimization problem at hand that require more exploitations therefore becomes viable. This enables achieving the purpose of the efficient CORE-MA paradigm. In what follows, what exactly is the feasibility structure and how it is modeled will first be elaborated. The feasibility boundary that becomes an essential part of the feasibility structure and how a classification technique can be used to approximate it will then be discussed. Indeed, this is a new paradigm in continuous optimization, hence the main novelty of the FSM, in that classification (instead of regression) technique is used in such context. Up to this point, the discussion will be based on a single equality or inequality constraint. Finally, three instantiations of the FSM will be presented in the form of FSM-driven MAs taking into account multiple equality and/or inequality constraints.

4.3.1 Feasibility Structure

The concept of FSM proposed herein deduces the feasibility structure of a constrained optimization problem based on the trial solutions encountered so far. To start with, Definition 4.3 describes what exactly the feasibility structure is.

Definition 4.3 (Feasibility Structure) *Feasibility structure of a constrained optimization problem is the arrangement of feasible and infeasible solutions in its search space.*

The search space of a constrained optimization problem can be sub-divided into one or more spaces containing the feasible solutions and one or more spaces containing the infeasible solutions. Depending on the constraints involved, there may be a single feasible solution space and a single infeasible solution space throughout the entire search space or there may also be multiple disjointed feasible and infeasible solution spaces as depicted in Figure 4.6. Prior to the generation and evaluation of any trial solution, the feasibility structure of an optimization problem is clearly unknown unless the problem is analytically analyzable—in which case there is no need of numerical solution. As the search for the global optimum progresses, more and more trial solutions are generated and evaluated, revealing more and more about the feasibility structure of the optimization problem at hand. By modeling this feasibility structure, the search effort can therefore be focused on the promising regions where any optimum may be located, hence avoiding seemingly unnecessary refinement until it is proven otherwise.

(a) Single feasible and infeasible space (b) Multiple feasible and infeasible spaces

Figure 4.6: Feasibility Structure of Constrained Optimization Problem
 Lighter regions denote contour of feasible spaces.
 Darker regions represent infeasible spaces.

For the purpose of modeling the feasibility structure of an optimization problem, the trial solutions that have been generated and evaluated previously need to be stored in a lookup table for future reference. As discussed earlier in Chapter 3, maintaining a lookup table is advantageous in the optimization of computationally expensive problems in that it prevents the expensive re-evaluation of previously encountered trial solutions. From the lookup table, it is also possible to extract certain number of nearest neighbors to a trial solution. This can in turn be used to model the local feasibility structure of the optimization problem around the trial solution of interest in order to decide if that particular trial solution is located in a promising region of the search space. If it is, only then the trial solution will undergo a local refinement to improve its quality. Note that local refinement perturbs the vector of a trial solution to within some locality in an attempt to improve its quality. As such, local model is arguably more relevant than the global one.

Encountering uniform neighbors belonging to the same in-feasibility class as that of the individual under assessment is the necessary condition of having a uniform ε -neighborhood of that particular in-feasibility class. Entirely feasible neighbors when the individual is also feasible are precursor to having a uniformly feasible ε -neighborhood. As shown by the optimum immediacy corollary, an optimum may be located in such a neighborhood making local refinements significant. Entirely feasible neighbors of a feasible individual may also signify a composite ε -neighborhood in reality. The corollary also shows that an optimum may be located within such a neighborhood making local refinements significant. Thus, regardless of whether it signifies a uniformly feasible or a composite ε -neighborhood, encountering all feasible neighbors of a feasible individual would deem the individual worthy to undergo local refinement.

In contrast, entirely infeasible neighbors of the same in-feasibility class as that of the individual under assessment are precursor to having a uniformly infeasible ε -neighborhood of that particular in-feasibility class in which there cannot be any optimum. Being only the necessary condition for a uniformly infeasible ε -neighborhood of either infeasible class, entirely infeasible neighbors of the same in-feasibility class as that of the individual may actually signify a composite ε -neighborhood in reality. Although an optimum may be located in such a neighborhood, the fact that all infeasible neighbors of the same class are implied by the uniformly infeasible ε -neighborhood makes deferment of local refinement on an individual with all infeasible neighbors belonging to the same infeasibility class a reasonable strategy.

In FSM, should otherwise the local refinement be executed upon encounter with infeasible neighbors of the same in-feasibility class as that of the individual with respect to which the neighbors were extracted, there would hardly be any savings in terms of the number of evaluations required to identify the global optimum. This particularly applies to problems solvable within reasonably low generation count. Unless the local refinement can be applied restrictively, it would be reasonable to defer the execution of the local refinement. If indeed a composite neighborhood is signified, the stochastic optimizer would later generate a feasible trial solution in that neighborhood and local refinement then stands a chance of being executed.

Upon encounter with some or all of the neighbors being in the different in-feasibility class from that of the individual under assessment, it is certain that a composite ε -neighborhood is implied. This includes encountering neighbors of feasible and either infeasible classes, neighbors of both infeasible classes, and neighbors of all possible in-feasibility classes. In any case of compositeness, existence of a boundary that separates the feasible from the infeasible solution spaces is guaranteed. For neighbors of both infeasible classes, the boundary additionally separates the positive from the negative infeasible solution spaces. This boundary is dubbed the feasibility boundary. More details on this boundary and how to approximate it using a classification technique—a novel paradigm in the context of numerical optimization—will be described next.

4.3.2 Feasibility Boundary

In FSM, the deduction of the feasibility structure given an optimization problem is based on the trial solutions encountered so far. As discussed previously, this involves the extraction of a number of nearest neighbors given a trial solution of interest followed by the inspection of the in-feasibility class of each of the neighbors. Given an optimization problem with a single equality or inequality constraint, encountering uniform neighbors with the same in-feasibility class as that of the trial solution of interest with respect to the constraint suggests a uniform ε -neighborhood of that class with respect to the constraint. Encountering at least one neighbor of different class from the trial solution of interest implies a composite ε -neighborhood with respect to the constraint within which lies a separating boundary—dubbed the feasibility boundary—of the problem that distinguishes the feasible from the infeasible solution spaces. It should be noted that with respect to an equality constraint, the feasibility boundary is in fact the feasible space, therefore additionally separating the positive from the negative infeasible spaces. Before proceeding further, Definition 4.4 first formalizes the notion of feasibility boundary taking into account presence of multiple constraints instead of only a single constraint.

Definition 4.4 (Feasibility Boundary) *Given any constrained optimization problem with a set of constraint functions $\mathbf{c}$, its feasibility boundary is defined by the set of solutions having zero constraint values w.r.t. each and every constraint present therein, i.e. $\{\mathbf{x} : \mathbf{c}(\mathbf{x}) = 0\}$.*

Note that the following discussion will still be based on single equality or inequality constraint even though the formalization of feasibility boundary above mentions multiple constraints. On the boundary itself, any solution is feasible by definition. The boundary is simply the collection of the outermost points of a feasible space. In other words, the boundary is the surface of the feasible space. The boundary is implied when an individual and its nearest neighbors populate the feasible class and either one of the infeasible classes or are of both infeasible classes. Of many different possible combinations of the in-feasibility class of the trial solutions, there are situations where the feasibility boundary need to be approximated to assess if the trial solution is close to the actual feasibility boundary and therefore is worthy of local refinement.

Clearly, the feasibility boundary need to be approximated only when the trial solution is infeasible in the first place. A feasible trial solution is always worthy of local refinement as an optimum may be located both in the interior and on the surface of the feasible space. When the trial solution is infeasible, it can have all of its nearest neighbors belonging to the feasible class or of the opposite infeasible class. In such cases—that is, when the individual under assessment is the only solution of one particular in-feasibility class—there must have been no previous exploitation between the individual and any of the feasible neighbors or the neighbors of the opposite infeasible class in the absence of any feasible neighbors. The trial solution is therefore worthy to undergo local refinement for further exploitation of its surrounding.

When there are one or more neighbors that are of the same infeasibility class as that of the trial solution being assessed, these neighbors may clearly be resulted from earlier exploitation. Approximation of the feasibility boundary then becomes necessary to determine whether the individual under assessment or its neighbors of the same infeasibility class that is as the matter of fact closer to the boundary. If any of the neighbors are actually closer, no further exploitation starting with the individual under assessment would be needed. Otherwise, the individual must be located in some unexploited region so far, hence worthy of local refinement.

Approximating the feasibility boundary is for the first time viewed as a two-class classification problem separating the feasible class from the infeasible class in FSM. Never before such approach was attempted in the context of numerical optimization. In the presence of all in-feasibility classes within the neighborhood, however, any feasible neighbors should be discarded from the training data so as to allow formulation of a two-class classification problem separating the positive infeasible class from the negative infeasible one. As mentioned above, the feasibility boundary also separates the negative from the positive infeasible solution spaces in addition to separating the feasible from the infeasible solution spaces in the context of equality-constrained optimization. Removal of the feasible neighbors is therefore reasonable. Furthermore, through the approximation of the feasibility boundary, we wish to be able to deduce if an infeasible trial solution in assessment is located in the unexploited region. Therefore, removal of the feasible neighbors is in line with our goal.

As with any classification technique, algorithm that solves a two-class classification problem generally produces a decision function $D(\mathbf{x})$ where $\mathbf{x}$ in this context is the n -dimensional solution vector and employs a threshold of zero between the two classes. The set $\{\mathbf{x} : D(\mathbf{x}) = 0\}$ therefore serves as the decision boundary. On the provision of training data consisting of neighbors from exactly two different in-feasibility classes, as discussed above, the decision boundary approximates the local segment of the feasibility boundary within that particular neighborhood. The actual feasibility boundary may actually be highly nonlinear. Within sufficiently small neighborhood, however, it is approximately linear as depicted in Figure 4.7. This suggests that the nonlinearity will be captured only during the early generations of the search when distribution of the trial solutions in the lookup table are still quite sparse. As the search progresses and more trial solutions are generated and evaluated, the denser distribution of encountered trial solutions will result in smaller neighborhood making linear approximation within some locality sufficient. Therefore, the use of linear classifier such as the linear SVM presented in Section 3.3 with a decision function as described by Equation 4.5 in a local neighborhood would be reasonable.

$$D(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + w_0 \quad (4.5)$$

This simplifies the task of choosing the right nonlinear model for the decision function. Furthermore, multiple piecewise linear models is arguably easier and cheaper to construct than a single nonlinear model. This further supports the choice of local model over the global one.

Figure 4.7: Linear Approximation of Local Feasibility Boundary
The shaded area is the margin-of-separation of a linear SVM, signifying region significantly close to the feasibility boundary.

Classifying feasible class against either infeasible class, the feasible class substitutes the missing infeasible class making all the training data belong to +1 or -1 class. A classification algorithm can then produce a decision function that separates the two classes with a threshold of zero decision value. It is desirable that $y_i D(\mathbf{x}_i) \geq 1$ for all neighbors $\mathbf{x}_i$. Between the two classes of the neighbors lies the margin of separation where no information is known as no neighbors has been identified. On this gap, by convention, the decision value of any solution should therefore be between -1 and 1 and therein lies the decision boundary $\{\mathbf{x} : D(\mathbf{x}) = 0\}$. By solving a quadratic program, the SVM unlike other classifiers, as discussed in Section 3.3, maximizes this margin of separation.

Viewing the decision boundary as an approximation to the feasibility boundary within the context of optimization, the value of the decision function $D(\mathbf{x})$ serves as an estimate of the distance measure of any solution $\mathbf{x}$ from the actual feasibility boundary. With decision value between -1 and 1, any solution on the margin of separation can be deemed close to the feasibility boundary as $|D(\mathbf{x}_i)| \geq 1$ for any neighbor $\mathbf{x}_i$. Most importantly, however, the margin of separation indicates region of the search space in which no exploitation has been performed so far. Maximizing the margin of separation means covering most of the allegedly unexploited area within a particular neighborhood that is close to the feasibility boundary. Nevertheless, even with the margin of separation maximized, it can often be difficult to estimate the unexploited region very accurately. Unless the actual feasibility boundary is really linear and all of the neighbors that become the support vectors are of the same distance to the actual feasibility boundary, misclassification will most likely occur. Fortunately, in the context of linear classifier, misclassification signifies relative closeness to the feasibility boundary as depicted by Figure 4.8. The misclassification may also occur because the actual feasibility boundary is highly nonlinear. The fact that a linear classification model can be constructed, that indicates the area where misclassification occurs also represents an allegedly unexploited region so far. Should that particular region have undergone exploitation previously, there must have been at least one solution with known in-feasibility class which would prevent either the successful construction of the SVM model or the misclassification from occurring.

Figure 4.8: Misclassification from Linear Approximation
Any candidate solution lying on the shaded area should be misclassified,
indicating region sufficiently close to the feasibility boundary.

Thus far, it has been assumed that the neighbors are linearly separable. Frequently, this means the feasibility boundary is approximately linear on the relevant scale. Occasionally, the feasibility boundary is in fact highly nonlinear yet a linear decision function is deducible. That means misclassification would definitely occur indicating some previously unexploited region as discussed above. In the event of nonlinearly separable neighbors, the linear decision function cannot be deduced. This suggests that the neighborhood may not be sufficiently dense such that it captures the nonlinearity. It is normally the case with early generations of the search during which the goal is to reveal more about the topology of the problem. Hence, local refinement is significant in the failure to deduce a classification model.

4.3.3 FSM-driven Memetic Algorithm

Based on the foregoing discussions, it can be summarized that a trial solution is significant for local refinement if with respect to a constraint,

- the individual is feasible, or
- the nearest neighbors are all feasible or of opposite infeasible class from that of the individual, or
- the nearest neighbors are composite and either the SVM is not deducible or the individual is misclassified or on the margin of separation.

Additionally, so long as the trial solutions in the lookup table are of insufficient diversity with respect to any constraints, any individual will be deemed significant for local refinement for quicker identification of feasible solutions or solutions of the opposite infeasible class.

Using the conceptual framework FSM described above, three instances of the FSM-driven MA will be presented in the following. The FSM-driven MA was initially inspired by the inequality-constrained optimization problems, the details of which is given in the original FSM for inequality constraints below. It was then generalized to accommodate assessment of the equality-constrained optimization problems, which at the same time also gives rise to the assessment of optimization problems with both equality and inequality constraints. To handle minute ratio of feasible to the entire solution space that are often resulted from the aggregate of the inequality constraints, ecumenical FSM is then proposed. Each of these will be discussed in great detail below.

4.3.3.1 Original FSM

An inequality constraint determines if a trial solution to an optimization problem is feasible or infeasible. In the presence of multiple inequality constraints, violation of one or more constraints causes a solution to be infeasible. On the contrary, a solution is feasible when it satisfies all the constraints. The division of the entire search space of an inequality-constrained optimization problem into the feasible and the infeasible solution spaces constitutes the feasibility structure of that particular problem. It describes where the optimum solution to the problem may be located, which would allow focused local refinements only on the promising regions should the feasibility structure can be modeled.

An inequality constraint function $g : \mathfrak{R}^n \rightarrow \mathfrak{R}$ generally maps n -dimensional solution vector $\mathbf{x}$ to a negative, zero, or positive real number $g(\mathbf{x})$. By definition, *i.e.* $g(\mathbf{x}) \leq 0$, a trial solution $\mathbf{x}$ is feasible if it is of negative or zero constraint values. It is infeasible otherwise. In the presence of multiple equality constraints, the maximum among the inequality constraint values given a trial solution describes the most stringent constraint value and can be thought of as an aggregate $\hat{g}$ of all the inequality constraints, that is $\max\{\mathbf{g}(\mathbf{x})\}$. Given a trial solution $\mathbf{x}$, it is feasible with respect to all inequality constraints when $\max\{\mathbf{g}(\mathbf{x})\}$ is negative or zero and it is infeasible otherwise. The aggregate inequality constraint $\max\{\mathbf{g}(\mathbf{x})\}$ therefore divides the entire solution space into feasible and infeasible spaces, between which lies a separating boundary dubbed the feasibility boundary. This boundary is precisely defined by the set of solutions with zero aggregate value, *i.e.* $\{\mathbf{x} : \max\{\mathbf{g}(\mathbf{x})\} = 0\}$.

In the search for the global optimum, the feasibility boundary can be approximated upon encountering an individual with composite nearest neighbors. Exact replica of the feasibility boundary is, however, not of interest. Approximated feasibility boundary should suffice to deduce the closeness of an infeasible trial solution to the actual feasibility boundary. Using SVM to separate feasible from infeasible neighbors, the margin of separation in which no training data are located signifies unexploited region close to the actual feasibility boundary. Furthermore, as described earlier, any misclassification resulted would also signify closeness to the actual feasibility boundary. Local refinements are necessary in both cases. Additionally, failure to construct the approximation model would easily signify the importance of local refinement as the nonlinearly-separable neighbors could have been resulted from sparse distribution capturing somewhat global neighborhood or local neighborhood around the intersection of two constraint boundaries as illustrated in Figure 4.9.

(a) Intersection of Constraint Boundaries (b) Highly Nonlinear Constraint Boundary

Figure 4.9: Nonlinear Local Feasibility Boundary

*: Individual of interest with respect to which neighbors are defined.

o: Feasible neighbors. x: Infeasible neighbors.

Algorithm 4.2 presents the procedures for performing the assessment of individuals in the original FSM. Handling only the inequality-constrained problems, it treats multiple inequality constraints as a single aggregate $\hat{g}(\mathbf{x}) = \max\{\mathbf{g}(\mathbf{x})\}$. Under the original FSM assessment, an individual in the population-based MA to solve inequality-constrained problems is significant if

- the individual is feasible, or
- all nearest neighbors of the individual are feasible, or
- the nearest neighbors are composite and either the SVM is not deducible or the individual is misclassified or on the margin of separation.

Algorithm 4.2 Original FSM-driven Memetic Algorithm

```

while no stopping criterion is met do
    ...
    for each individual  $\mathbf{x}$  in the population do
        if  $\hat{g}(\mathbf{x}) > 0$  AND sufficient diversity w.r.t.  $\hat{g}$  then
             $\mathcal{N}_k(\mathbf{x}) = \{\mathbf{z} : \mathbf{z} \neq \mathbf{x}, \mathbf{z} \in \mathcal{LT}, \mathcal{R}(d(\mathbf{z}, \mathbf{x})) \leq k\}$ 
            if  $\forall i \hat{g}(\mathbf{z}_i) > 0, \mathbf{z}_i \in \mathcal{N}_k(\mathbf{x})$  then
                continue
            elseif  $\exists i, j \hat{g}(\mathbf{z}_i) > 0, \hat{g}(\mathbf{z}_j) \leq 0, \mathbf{z}_i, \mathbf{z}_j \in \mathcal{N}_k(\mathbf{x})$  then
                 $\mathcal{D}_{\hat{g}}(\mathbf{x}) = \text{SVM}((\mathbf{z}, \mathcal{IFC}_{\hat{g}}(\mathbf{z})) : \mathbf{z} \in \mathcal{N}_k(\mathbf{x}))$ 
                if  $\mathcal{D}_{\hat{g}}(\mathbf{x})$  is deduced AND  $\mathbf{x}$  is neither misclassified nor on
                    margin-of-separation then continue
            end if
        end if
        refine  $\mathbf{x}$ 
    end for
end while

```

Among the most expensive operations in the assessment are the nearest neighbors identification and the construction of SVM model in the event of encounter with composite nearest neighbors. The assessment would therefore incur an overhead of up to $\mathcal{O}(m^2 + n^3)$ noting that m is the size of the lookup table and n is the dimensionality of the problem. Depending on the ratio of the feasible to the entire solution space, an overhead of only $\mathcal{O}(1)$ in the case of feasible individual or $\mathcal{O}(m^2)$ in the case of infeasible individual may dominate the search for the global optimum. With relatively small m , the overhead incurred would be insignificant compared to the time-consuming fitness evaluation of a trial solution to the computationally-expensive problem.

Anticipated Limitations Thus far, it has been discussed how the original FSM could help alleviate the problem of executing unnecessary local refinement, hence savings precious computing resources which in the context of computationally-expensive problem could translate into solving the problems days earlier. When the ratio of the feasible to the entire search space is tiny, however, it would be difficult for the stochastic optimizer of the MA to sample a feasible trial solutions. Most trial solutions generated by the stochastic optimization methods would be infeasible, deferring the execution of the deterministic optimization techniques. This eventually results in slow convergence as the adaptive MA would resemble a simple GA. Hence, the original FSM would only be suitable for problems with moderate ratio of feasible to entire search space.

4.3.3.2 Generalized FSM

In the presence of one or more equality constraints, the feasibility structure of an optimization problem is in line with those having very restrictive inequality constraints. As the matter of fact, an equality constraint $h(\mathbf{x}) = 0$ can be decomposed into two inequality constraints $h(\mathbf{x}) \leq 0$ and $-h(\mathbf{x}) \leq 0$. These two are understandably very restrictive inequality constraints. As such, modeling the feasibility structure using the original technique that distinguishes the feasible from the infeasible trial solutions will not work. Most trial solutions generated by the stochastic optimization methods would be infeasible, deferring the execution of the deterministic optimization techniques. This eventually results in slow convergence as the search would rely much on the stochastic method.

In general, a constraint function $c : \Re^n \rightarrow \Re$ maps the n -dimensional solution vector $\mathbf{x}$ to a negative, zero, or positive real number $c(\mathbf{x})$. Given an inequality constraint $g(\mathbf{x}) \leq 0$, the constraint function g understandably divides the solution space into the feasible space with negative and zero constraint values and the infeasible space with positive constraint values. Given an equality constraint $h(x) = 0$, the constraint function h similarly divides the solution space into feasible and infeasible space. However, the feasible space only contains zero constraint values whereas the infeasible space encompasses negative and positive values. The set of solution with zero constraint values, i.e. $\{\mathbf{x} : h(\mathbf{x}) = 0\}$, then serves as the constraint boundary between the feasible and the infeasible solutions and additionally between two types of infeasible solutions.

In the original FSM for inequality constraints, the feasibility boundary is approximated by deducing a classification model that takes as its training data some previously encountered feasible solutions whose constraint values are all zero or negative and the evaluated infeasible solutions with at least one positive constraint value. Following this paradigm, generalizing FSM for equality constraints simply imposes additional requirement in that the infeasible solutions used to deduce the classification model must be of the same infeasible class, that is all negative infeasible or all positive infeasible, as illustrated in Figure 4.10. In the presence of infeasible solutions of different infeasibility classes within the composite neighborhood of a trial solution, the classification model is deduced based on the infeasibility classes of the neighbors after eliminating from the training data the feasible solutions within the neighborhood, if any.

Modeling the feasibility structure of an optimization problem in the presence of multiple equality constraints requires multiple classification models. Intuitively, the intersection of all the models would be the region closest to the feasible space of the problem. Encountering in this region a trial solution generated by the stochastic optimization method, however, must be difficult. Many of the trial solutions in the early evolution could, however, have populated regions that are close to the boundary of one of the equality constraints. Performing local refinement starting with this kind of trial solutions would be finding a direction that follow the boundary as well as minimizes the other constraint violations. Therefore, a trial solution significant w.r.t. any of the equality constraints would be deemed significant for the local refinement.

(a) Feasible vs. Positive Infeasible Neighbors

(b) Feasible vs. Negative Infeasible Neighbors

(c) Positive vs. Negative Infeasible Neighbors

Figure 4.10: Feasibility Boundary Approximation of Equality Constraint

Having generalized the FSM for handling equality constraints, it can therefore be used in the optimization problems involving both equality and inequality constraints. Intuitively, an individual in the MA population used to solve an optimization problem involving both the equality and the inequality constraints would be deemed significant for local refinement if it is significant with respect to not only the equality but also the inequality constraints. First, the individual would be assessed against the aggregate of the inequality constraints as with the case of the original FSM. Should it be insignificant, no further assessment would be conducted. The individual would not undergo local refinement. Otherwise, the individual will then be assessed against the equality constraints. As such, the assessment is of $\mathcal{O}(n^3)$ in the best case and of $\mathcal{O}(n_h \cdot n^3)$ in the worst case, giving a complexity of $\mathcal{O}(n_h \cdot n^3)$ on average.

Algorithm 4.3 presents the assessment procedure used in the generalized FSM. Multiple inequality constraints are still treated as a single aggregate $\hat{g}(\mathbf{x}) = \max\{\mathbf{g}(\mathbf{x})\}$. Under the generalized FSM, as illustrated in Figure 4.11, an individual would be deemed significant if

- the individual is feasible, or
- all nearest neighbors of the individual are feasible with respect to the aggregate of the inequality constraints or of opposite infeasible class with respect to any of the equality constraint, or
- the nearest neighbors are composite and either the SVM is not deducible or the individual is misclassified or on the margin of separation.

Unlike in the case of the original FSM whose complexity is invariant against the number of inequality constraints involved, multiple classification models increases the complexity of the assessment.

Algorithm 4.3 Generalized FSM-driven Memetic Algorithm

```

while no stopping criterion is met do
    ...
    for each individual  $\mathbf{x}$  in the population do
        if  $\hat{g}(\mathbf{x}) > 0$  AND sufficient diversity w.r.t.  $\hat{g}$  then
             $\mathcal{N}_k(\mathbf{x}) = \{\mathbf{z} : \mathbf{z} \neq \mathbf{x}, \mathbf{z} \in \mathcal{LT}, \mathcal{R}(d(\mathbf{z}, \mathbf{x})) \leq k\}$ 
            if  $\exists i, j \hat{g}(\mathbf{z}_i) > 0, \hat{g}(\mathbf{z}_j) \leq 0, \mathbf{z}_i, \mathbf{z}_j \in \mathcal{N}_k(\mathbf{x})$  then
                 $\mathcal{D}_{\hat{g}}(\mathbf{x}) = \text{SVM}((\mathbf{z}, \mathcal{IFC}_{\hat{g}}(\mathbf{z})) : \mathbf{z} \in \mathcal{N}_k(\mathbf{x}))$ 
                if  $\mathcal{D}_{\hat{g}}(\mathbf{x})$  is deduced AND  $\mathbf{x}$  is not on margin-of-separation
                    AND  $\mathbf{x}$  is not misclassified then continue
            elseif  $\forall i \hat{g}(\mathbf{z}_i) > 0, \mathbf{z}_i \in \mathcal{N}_k(\mathbf{x})$  then
                continue
            end if
        end if
        for each equality constraint  $h$  do
            if  $h(\mathbf{x}) \neq 0$  AND sufficient diversity w.r.t.  $h$  then
                 $\mathcal{N}_k(\mathbf{x}) = \{\mathbf{z} : \mathbf{z} \neq \mathbf{x}, \mathbf{z} \in \mathcal{LT}, \mathcal{R}(d(\mathbf{z}, \mathbf{x})) \leq k\}$ 
                if  $\exists i, j h(\mathbf{x})h(\mathbf{z}_i) > 0, h(\mathbf{x})h(\mathbf{z}_j) \leq 0, \mathbf{z}_i, \mathbf{z}_j \in \mathcal{N}_k(\mathbf{x})$  then
                     $\mathcal{D}_h(\mathbf{x}) = \text{SVM}((\mathbf{z}, \mathcal{IFC}_h(\mathbf{z})) : \mathbf{z} \in \mathcal{N}_k(\mathbf{x}))$ 
                    if  $\mathcal{D}_h(\mathbf{x})$  is deduced AND  $\mathbf{x}$  is neither misclassified nor
                        on margin-of-separation then continue
                elseif  $\forall i h(\mathbf{x})h(\mathbf{z}_i) > 0, \mathbf{z}_i \in \mathcal{N}_k(\mathbf{x})$  then
                    continue
                end if
            end if
            refine  $\mathbf{x}$ 
            break
        end for
    end for
end while

```

(a) Inequality-Constrained Problem

(b) Equality-Constrained Problem

(c) Equality-and-Inequality-Constrained Problem

Figure 4.11: Potential Regions under Generalized FSM
Darker regions represent infeasible spaces.

Anticipated Limitations The FSM has been generalized with additional overhead of deducing at most n more SVM models (n is the dimensionality of the problem) to handle equality constraints which cause minute ratio of the feasible to the entire solution space given some optimization problem. Treating the inequality constraints as an aggregate, tiny ratio of the feasible to the entire solution space with respect to the aggregate could easily cause difficulty to the generalized FSM in that almost all local refinements would be skipped, hence slowing down convergence to the optimum.

4.3.3.3 Ecumenical FSM

In the original as well as the generalized FSM, multiple inequality constraints are treated as an aggregate $\hat{g}(\mathbf{x}) = \max\{\mathbf{g}(\mathbf{x})\}$. Multiple inequality constraints can sometimes be quite restrictive, leaving tiny ratio of the feasible to the entire solution space. Modeling the feasibility structure based on the aggregate needs some feasible and infeasible trial solutions with respect to the aggregate. Tiny ratio of the feasible to the entire solution space would easily cause the infeasible trial solutions to dominate the search space. With the adversary to encounter the feasible trial solutions, modeling feasibility structure of the optimization problem at hand to a reasonable resemblance would likewise be difficult. With the most commonly encountered trial solutions being infeasible, most—if not all—individuals would be deemed insignificant for local refinements. Without much reliance on the deterministic optimizer, convergence to an optimum could therefore be rather slow as with the case of the simple GA.

To alleviate the above-mentioned problem, the ecumenical FSM presented in this section model the feasibility structure of the problem with respect to each of the inequality constraints individually. As illustrated in Figure 4.12, an infeasible individual may actually be close to the feasibility boundary yet easily deemed as deep in the infeasible space due to failure to encounter a sample of the feasible solutions in its neighborhood. All its nearest neighbors are infeasible with respect to the aggregate of the inequality constraints. Examining the inequality constraints individually, however, it is clear that there are feasible neighbors with respect to every inequality constraint. What seems to be a uniformly infeasible ε -neighborhood when examining the inequality constraints as an aggregate turns out to be a composite neighborhood by examining the inequality constraints individually.

As with the case of the generalized FSM for the equality constraints, closeness to the feasibility boundary with respect to any of the inequality constraints implies the significance of an individual to undergo local refinement. Given an infeasible individual, enhancing its quality through deterministic local refinement would turn it into a feasible solution faster than trying to improve it using the stochastic methods by means of some intelligent sampling of the search space. Refining the infeasible individual, furthermore, could have intuitively been easier should there be some nearby boundary to follow compared to when the individual is deep within the infeasible space with no guiding boundaries. Closeness to the feasibility boundary with respect to any of the inequality constraints would therefore deem the individual significant for local refinement.

(a) Aggregate Inequality Constraint

(b) Individual Inequality Constraint 1

(c) Individual Inequality Constraint 2

(d) Individual Inequality Constraints Combined

Figure 4.12: Aggregate vs. Individual Inequality Constraints
Darker regions represent infeasible spaces.

As the inequality constraints are now examined individually, it is no longer that an individual has to be deemed significant with respect to any of the equality and the aggregate inequality constraints to finally undergo local refinement. Assessed as sufficiently close to the boundary of any inequality constraint would deem the individual worthy to undergo local refinement as depicted by Figure 4.13. Algorithm 4.4 details the assessment procedure under the ecumenical FSM instance. Intuitively, assessing the inequality constraints individually could incur additional overhead of up to $\mathcal{O}(n_g \cdot n^3)$ in the event of insignificance with respect to the aggregate inequality. Therefore, the total worst-case complexity of the ecumenical FSM-driven MA is $\mathcal{O}(m^2 + n^3\{1 + \max(n_g, n_h)\})$.

Anticipated Limitations Under the generalized FSM-driven MA framework, insignificance of an individual with respect to the aggregate inequality would halt the assessment, incurring no further overhead. Under the ecumenical FSM-driven MA discussed herein, however, such insignificance would be followed by significance assessment of the individual with respect to each of the inequality constraints. This intuitively poses constant modeling cost to the assessment of almost every individual. Should it be possible to eliminate the use of the SVM or replace it with something cheaper, the assessment cost could consequently be brought down as shall be seen in the next section. Furthermore, all FSM-driven MAs require positive and negative constraint values to be effective. The strategy presented next, however, can also work under restricted constraint function mapping scenario.

(a) Inequality-Constrained Problem

(b) Equality-Constrained Problem

(c) Equality-and-Inequality-Constrained Problem

Figure 4.13: Potential Regions under Ecumenical FSM
Darker regions represent infeasible spaces.


```
while no stopping criterion is met do
  ...
  for each individual  $\mathbf{x}$  in the population do
    if  $\mathbf{x}$  is not significant w.r.t.  $\hat{g}$  then
       $\mathcal{N}_k(\mathbf{x}) = \{\mathbf{z} : \mathbf{z} \neq \mathbf{x}, \mathbf{z} \in \mathcal{LT}, \mathcal{R}(d(\mathbf{z}, \mathbf{x})) \leq k\}$ 
      for each inequality constraint  $g$  do
        if  $\exists i, j \ g(\mathbf{z}_i) > 0, g(\mathbf{z}_j) \leq 0, \mathbf{z}_i, \mathbf{z}_j \in \mathcal{N}_k(\mathbf{x})$  then
           $\mathcal{D}_g(\mathbf{x}) = \text{SVM}((\mathbf{z}, \mathcal{IFC}_g(\mathbf{z})) : \mathbf{z} \in \mathcal{N}_k(\mathbf{x}))$ 
          if  $\mathcal{D}_g(\mathbf{x})$  is deduced AND  $\mathbf{x}$  is neither misclassified nor
            on margin-of-separation then continue
        else if  $\forall i \ g(\mathbf{x})g(\mathbf{z}_i) > 0, \mathbf{z}_i \in \mathcal{N}_k(\mathbf{x})$  then
          continue
        end if
      refine  $\mathbf{x}$ 
      break
    end for
    continue
  end if
  for each equality constraint  $h$  do
    if  $h(\mathbf{x}) \neq 0$  AND sufficient diversity w.r.t.  $h$  then
       $\mathcal{N}_k(\mathbf{x}) = \{\mathbf{z} : \mathbf{z} \neq \mathbf{x}, \mathbf{z} \in \mathcal{LT}, \mathcal{R}(d(\mathbf{z}, \mathbf{x})) \leq k\}$ 
      if  $\exists i, j \ h(\mathbf{x})h(\mathbf{z}_i) > 0, h(\mathbf{x})h(\mathbf{z}_j) \leq 0, \mathbf{z}_i, \mathbf{z}_j \in \mathcal{N}_k(\mathbf{x})$  then
         $\mathcal{D}_h(\mathbf{x}) = \text{SVM}((\mathbf{z}, \mathcal{IFC}_h(\mathbf{z})) : \mathbf{z} \in \mathcal{N}_k(\mathbf{x}))$ 
        if  $\mathcal{D}_h(\mathbf{x})$  is deduced AND  $\mathbf{x}$  is neither misclassified nor
          on margin-of-separation then continue
        else if  $\forall i \ h(\mathbf{x})h(\mathbf{z}_i) > 0, \mathbf{z}_i \in \mathcal{N}_k(\mathbf{x})$  then
          continue
        end if
      end if
      refine  $\mathbf{x}$ 
      break
    end for
  end for
end while
```

4.4 Constrained Landscapes Stratification

Constrained Landscapes Stratification (CLS) is another conceptual framework that—similar to the FSM—enables attainment of the purpose of the efficient CORE-MA paradigm. Employing the CLS, the assessment of an individual in the population prior to undergoing local refinement is similarly based on the trial solutions encountered so far in the neighborhood of the individual. Unlike the FSM, however, the CLS does not make use of all the neighbors but selects a number of relevant neighbors that share the same goals and resources as the individual’s. As the result, the objective value or the amount of constraint violation can be directly compared to assess the individual’s significance to undergo local refinement depending on its in-feasibility class. This eliminates the need to model the feasibility structure which in the case of composite neighborhood requires the approximation of the feasibility boundary. Lower overhead is therefore anticipated as there is no need to use the SVM. Nevertheless, note that the CLS was designed for computationally expensive problem with readily available derivatives, such as the molecular structure determination using Gaussian, AMBER, etc. These simulation packages compute the energy of the molecule and its derivatives—which are actually the forces. Problems with known functional forms of their objectives and constraints also suit this as computing their derivatives will not increase their original computing time significantly as shown in [11]. Additionally, note also that the CLS was designed with the Sequential Quadratic Programming as the deterministic refinement technique and therefore is best suited for gradient-based local refinement methods.

Stratification of the search space given a constrained optimization problem and how the previously evaluated trial solutions that share the same stratum as the individual of interest can be approximated will first be elaborated in what follows. The stratum assessment will then be discussed to determine whether or not local refinement of the individual would worth the search effort. Finally, an efficient adaptive memetic algorithm will be instantiated using the assessment strategy based on the proposed constrained landscape stratification technique, namely the CLS-powered MA.

4.4.1 Search Space Stratification

Primary novelty of the CLS is the search space stratification, dividing it into strata or patches of the constrained landscape. Each of these patches or strata shares similar qualities of

1. Refinement Target

This defines what needs to be achieved on an individual by means of local refinement.

2. Relative Optimizing Direction

This defines what are available to achieve the target, and hence, how local refinement could achieve it.

For this purpose, CLS stratifies the solution space in two stages: stratification of targets by the neighborhood types and stratification of resources by the relative direction of optimizing steps as will be discussed separately in the following.

4.4.1.1 Stratification of Refinement Targets

In the search space of a constrained optimization problem, there are regions of feasible and infeasible solutions. Depending on their locations, individuals' refinement may have different goals. For feasible individuals, the refinements aim at minimizing their objective values. For infeasible individuals, the refinements is clearly to minimize their constraint violations. Near the boundary between the feasible and infeasible space, minimizing objective values must not result in any constraint violations while minimizing the constraint violations aims to produce directions that simultaneously minimize the objective values. The neighborhood type can be used to hint whether or not an individual in the population is near the boundary between the feasible and the infeasible spaces as illustrated in Figure 4.14.

Figure 4.14: Stratification of Search Space

Stratified by the composition of the neighbors and the feasibility of the individual with respect to which the nearest neighbors were identified, the search space of a constrained optimization problem can be categorized into the feasible space interior, boundary, and exterior. In the interior, the individual must definitely be feasible. Its k nearest neighbors are uniformly feasible with respect to all constraints. Presence of a composite neighborhood with respect to any constraint deems closeness to the feasibility boundary when with respect to all other constraints, the neighborhood is uniformly feasible. Individual on the boundary is therefore signified by at least one of its neighbors having different in-feasibility class with respect to one or more constraints whereas with respect to the remaining constraints, the individual is feasible and all its k nearest neighbors are uniformly feasible, as well. Finally, the exterior of the feasible space is signified by the presence of the uniformly infeasible neighborhood with respect to any constraint. As the neighbors are retrieved from previously evaluated trial solutions, they then serve as the predecessors of the individuals that has previously explored the same stratum.

Having identified the goal of the individual by assessing its relative location, any comparisons for the purpose of assessing the significance of the individual for local refinement thus will be done with respect to some or all of these neighbors. In the next section, further stratification of resources by the relative direction of optimizing steps will be discussed. This second-stage stratification is meant to enable comparison of the individual under assessment with its neighbor in the same stratum to determine if the individual is worthy of local refinement.

4.4.1.2 Stratification of Optimizing Directions

Following the stratification by the neighborhood type above, the individual and its neighbors share the same goals. However, what are available to the individual may not necessarily be the same as what are available to its neighbors. As the result, their optimizing directions may not be the same. As illustrated in Figure 4.15, the constraint landscape can simply be divided into three strata. Infeasible individuals in the same stratum should move mainly downwards to minimize the constraint violation, eventually ending up with the same feasible space. The objective landscape are composed of three strata. In the same stratum, feasible individuals share the same relative direction that minimizes the objective value. From the same combination of the objective and constraint strata, refinement would converge to the same optimum. This means even though the neighbors are the predecessors of the individual which has explored that region previously, different optimum may possibly be identified should the individual undergo local refinement. This suggests that the stratum of same goal can still be further divided into even smaller strata; each of which share the same relative optimizing direction. In the context of gradient-based method like the SQP used herein, the optimizing direction is dictated by the derivatives of the objective and/or the constraint functions. For the purpose of this second-stage stratification, only the relative direction is actually important. Therefore, only signs of the derivatives would in fact be of interest. Neighbors with the same signs of derivatives as those of the individual would be selected for later comparisons in the interest of assessing the significance of the individual to undergo local refinement.

Figure 4.15: Stratification Based on Optimizing Directions

For an individual in the interior of the feasible space, only derivatives of the objective function would intuitively matter to the minimization of the objective value. Near the boundary of the feasible space, the derivatives of all the constraint functions would also contribute in determining the optimizing direction. This is as illustrated by neighborhood A in Figure 4.15. A feasible individual may have neighbors sharing the same objective but different constraint derivatives. Should any of the neighbors be of better fitness, local refinement of the individual could easily be skipped until better individuals are generated by the stochastic global search. At the exterior of the feasible space, however, derivatives of the objective function will not be taken into account. As the refinement target is just to minimize the amount of constraint violation as much as possible, only the constraint derivatives would intuitively matter in the minimization of the violation. Note that any vanishing derivatives are not used for the purpose of the stratification.

Given an individual in the population whose significance for local refinement is to be determined, therefore, first its k nearest neighbors would be extracted from the lookup table to achieve stratification of the refinement targets in the search space by the neighborhood type. Then, neighbors that share the same signs of objective and/or constraint derivatives are selected among the k nearest neighbors to achieve stratification of the optimizing direction in the localized search space within the neighborhood. At this point, the individual under assessment and its selected neighbors would share the same refinement target and relative optimizing direction. They can therefore be compared sufficiently fairly in the assessment of the significance of the individual for undergoing local refinement as will be discussed next.

4.4.2 Stratum Assessment

Given a constrained optimization problem, it has been discussed earlier that its search space can be divided into landscape strata; each of which shares the same refinement target. Each stratum can then be further sub-divided into more detailed strata; each of which shares the same optimizing direction that local refinement of any solution in that particular stratum would proceed in similar manner. With the same target and relative direction, refinement of any solution in the same stratum would have the same set of increased and decreased design variables. As such, it is likely that two refinements from two different trial solutions on the same stratum would attain the same optimum, wasting the search efforts.

Should these strata be known a priori, preventive measures can be taken so as not to waste search effort due to the unnecessary refinement. Knowing the boundaries of the stratum an individual is located at makes possible the identification of the trial solutions therein which has been evaluated previously. If there is none, no predecessor of the individual has been detected making the individual significant to undergo local refinement for further exploitation. Otherwise, further assessment is necessary to determine the significance of the individual to undergo local refinement. In the preceding sections, a procedure to identify the previously evaluated trial solutions sharing the same stratum as the individual under assessment has been presented. Herein, the detailed comparison in the interest of assessing the significance of an individual in the population to undergo local refinement will be elaborated. Note that, beforehand, that the landscape of a constrained optimization problem is in general completely unknown prior to evaluating any trial solution. As the optimization process progresses, more and more details about the landscape starts to unveil, enabling the deduction of the constrained landscape stratification. In the early generation of the search, evaluated trial solutions are generally sparsely distributed, hence any disjointed strata with the same target and relative direction could have appeared as if they were connected, missing all others in-between the disjointed strata. As population of trial solution evolves from one generation to the next, however, deduction of the constrained landscape stratification will become more and more accurate due to denser distribution of the evaluated trial solutions, particularly in the regions where optimum may be located.

With the above noted, the detailed comparison for assessing the significance of an individual to undergo a local refinement will be elaborated separately for individual in the feasible space interior, boundary, and exterior in the following three sections.

4.4.2.1 Feasible Space Interior

An individual in the feasible space interior must itself be feasible with respect to all constraints. Its k nearest neighbors must also be feasible with respect to all constraints. The optimum immediacy corollary states that in the immediacy of an optimum at the trough of a feasible basin, there exists a neighborhood in which all neighbors are feasible. In other words, a feasible individual with all feasible neighbors is a precursor to some immediate optimum. Intuitively, not all feasible individuals would in fact be significant for local refinement. In the basin where the feasible individual is located, any previous exploration or exploitation attempt would leave telltale signs in the form of some evaluated trial solutions. Indeed, it could have been resulted from random sampling of the global search component. Therefore, feasible individual in the feasible space interior would be worthy of local refinement if there exists no neighbor among the k nearest neighbors that share the same signs of the objective derivatives. When there are some predecessors sharing the same stratum with the individual, the objective value of the individual needs to be compared with those of its selected neighbors. The individual is worthy of local refinement only if all its selected neighbors are of higher objective values.

4.4.2.2 Feasible Space Boundary

An individual near the feasible space boundary must have neighbors of different in-feasibility class with respect to at least one constraint. With respect to the remaining constraint, the individual must have neighbors that are all feasible. The optimum immediacy corollary states that in the immediacy to an optimum at one or more constraint boundaries, composite neighborhood with respect to the active constraints must be observed. Meanwhile, a composite or a uniformly feasible neighborhood may be observed for each of the remaining constraints with the former being observed when the neighborhood radius is sufficiently large. Furthermore, it is also stated in the corollary that there must exist a composite neighborhood with respect to one or more constraints when the radius of the neighborhood is sufficiently large in the immediacy of an optimum at the trough of a feasible basin. This signifies that an individual near the feasible space boundary could be worthy of local refinement.

Near the feasible space boundary, an individual may be feasible or infeasible. A feasible individual satisfies all the constraints. However, near the feasible space boundary, a feasible individual is just next to violating one or more constraints. Refinement of such individual, therefore, attempts to find a direction that minimizes its objective value and simultaneously maintains its feasibility. In the identification of its relevant neighbors, the derivatives of both the objective and the constraint functions play a crucial role. With its relevant neighbors identified, assessing the individual can simply be done by comparing the objective values of trial solutions in that stratum like in the previous section.

An infeasible individual, on the other hand, violates one or more constraints. It, however, need not necessarily violate all the constraints. It may still satisfy some of the constraints. The refinement of an infeasible individual thus attempts to find a direction that minimizes the amount of violations with respect to the violated constraints while maintaining the feasibility with respect to the satisfied constraints. Near the feasible space boundary, additionally, the refinement should also aims at finding a direction that simultaneously minimizes the objective value. The derivatives of both the objective and the constraint functions, therefore, are used to identify the relevant neighbors of the individual. With its relevant neighbors sharing the same stratum, assessment of the individual as to whether or not it is significant for local refinement can easily be done by comparing the amount of constraint violations with respect to the violated constraints. In this regards, the CLS assesses the significance of an individual like the ecumenical FSM does. In CLS, closeness to the boundary with respect to any of the violated constraints implies the significance of an individual to undergo local refinement. Refining infeasible individuals could have intuitively been easier should there be some nearby boundary to follow compared to when the individual is deep within the infeasible space with no guiding boundaries. This is because the refinement would simultaneously maintain the feasibility with respect to the constraints already satisfied while attempting to minimize the violation with respect to the remaining constraints. Closeness to the feasibility boundary with respect to any of the inequality constraints would therefore deems the individual significant for local refinement.

4.4.2.3 Feasible Space Exterior

The feasible space exterior is as the matter of fact the interior of the infeasible space. Clearly, a neighborhood in the feasible space exterior must entirely be composed of infeasible solutions with respect to one or more constraints. This means an infeasible trial solution in the feasible space exterior may not after all be infeasible with respect to all constraints. Deduction of the neighborhood type by means of k nearest neighbors given an individual provides no guarantee that an infeasible individual with entirely infeasible nearest neighbors with respect to any constraint would definitely imply a truly uniformly infeasible neighborhood. Therefore, even though the optimum immediacy corollary hints that there cannot be any optimum within a truly uniformly infeasible neighborhood, it would be best if local refinement upon encounter with the infeasible individual with uniformly infeasible nearest neighbors with respect to a constraint can be applied quite restrictively. Stratum assessment for infeasible individual near the feasible space boundary would therefore be observed to allow this.

In the early generations of the search, as discussed earlier, the stratum approximation given an individual may not be so accurate. Disjointed strata may appear to be connected. Early generations are often signified by insufficient diversity of the trial solutions in the lookup table with respect to one or more constraints. The trial solutions belong to only either infeasible class with respect to these constraints. To accommodate the possibility of inaccurate stratum approximation, more lenient assessment is required for reasons illustrated by Figure 4.16.

Figure 4.16: Inaccurate Stratum Approximation During Early Generations of Memetic Search

In Figure 4.16, the individual is in fact located at a stratum where only a single worse predecessor can be identified. The individual would have been deemed as significant for local refinement if the stratum can be approximated very accurately. However, due to sparse distribution of the trial solutions in the early generation of the search, disjointed strata are captured. In the supposedly different strata which are yet modeled as a single stratum due to the same sign of the constraint derivatives, there lies a neighbor with better objective value. Very stringent criterion would result in deferment of the local refinement while actually it is significant to refine the individual. Once refined, it could quickly converge to the global optimum, hence solving the problem. Therefore, more lenient criterion is used. Rather than refining only the individual with minimum constraint violation, refinement of an individual with violation below the average violation of its relevant neighbors would be allowed upon insufficient diversity.

4.4.3 CLS-powered Memetic Algorithm

Based on the foregoing discussions, assessing whether refining some individuals would worth the search effort involves identification of some relevant neighbors of each of the individuals by observing:

1. feasibility of the individuals and composition of their nearest neighbors
2. combination of the signs of the derivatives of
 - the objective function for individuals in the feasible space interior
 - the constraint functions for individuals in the feasible space exterior
 - both the objective and the constraint functions for those individuals near the feasible space boundary

Each of those individuals above would in summary be deemed significant to undergo local refinement if any of the following is satisfied.

1. the individual is feasible with an objective value smaller than those of its relevant neighbors
2. the individual is infeasible with amount of violation smaller than those of its relevant neighbors with respect to any constraint unless trial solutions in the lookup table are of insufficient diversity in which case the violation below the average violation of the neighbors would be acceptable

Bearing similarities to the ecumenical FSM-driven MA, it should be noted that the CLS-powered MA applies restrictive instead of exhaustive local refinements upon insufficient diversity.

Employing the assessment strategy as summarized in the above, procedural details of the CLS-powered MA is presented as a pseudo-code in Algorithm 4.5 with $\mathcal{LT}$ being the lookup table that contains those trial solutions encountered and evaluated earlier, $\mathcal{R}$ representing the ascending ranking scheme, whereas d denoting the distance measure between a pair of trial solutions. The Euclidean distance in Equation (3.3) is used in CLS for the identification of the k nearest neighbors $\mathcal{N}_k(\mathbf{x})$ with respect to any individual $\mathbf{x}$. The sgn function—mapping a vector or matrix of real values into other vector or matrix of its element-wise signs with a '+1' for positive entry, a '-1' for negative entry, and a '0' for zero entry—is then used in the identification of relevant neighbors of the individual. In conjunction with the operator $\overset{=}{=}$, which turns true when comparing zeros or the same quantity of two corresponding entries in l.h.s. and r.h.s., the function enables selection of the relevant neighbors $\tilde{\mathcal{N}}_k(\mathbf{x})$ for a fair assessment.

Observing Algorithm 4.5 closely, it should be intuitive that identification of the k nearest neighbors of the assessed individual would be the most expensive component of the assessment strategy in the CLS-powered MA with the worst-case complexity of $\mathcal{O}(m^2)$ and similar average-case and best-case complexity of $\mathcal{O}(m \log m)$ where $m = |\mathcal{LT}|$. The rest of the components can be accomplished in $\mathcal{O}(n + n_g + n_h)$ operations at best to $\mathcal{O}(n^2(n_g + n_h))$ operations at worst, in which n_h and n_g are respectively number of equality and inequality constraints and k is assumed to be just a few multiple of the problem size n . With $m \gg n$ sooner or later, it is clear that the k nearest neighbors identification dominates the complexity of the assessment strategy.


```
while no stopping criterion is met do
  ...
  for each individual  $\mathbf{x}$  in the population do
     $\mathcal{N}_k(\mathbf{x}) = \{\mathbf{z} : \mathbf{z} \in \mathcal{LT}, \mathbf{z} \neq \mathbf{x}, \mathcal{R}(d(\mathbf{z}, \mathbf{x})) \leq k\}$ 
    if  $\mathcal{N}_k(\mathbf{x}) \cup \{\mathbf{x}\}$  is uniformly feasible w.r.t. all constraints then
       $\tilde{\mathcal{N}}_k(\mathbf{x}) = \{\mathbf{z} : \mathbf{z} \in \mathcal{N}_k(\mathbf{x}), \text{sgn}(\nabla f(\mathbf{z})) \doteq \text{sgn}(\nabla f(\mathbf{x}))\}$ 
    elseif  $\mathcal{N}_k(\mathbf{x}) \cup \{\mathbf{x}\}$  is uniformly feasible w.r.t. some constraints but
      composite w.r.t. the remaining constraints then
       $\tilde{\mathcal{N}}_k(\mathbf{x}) = \{\mathbf{z} : \mathbf{z} \in \mathcal{N}_k(\mathbf{x}), \text{sgn}(\nabla f(\mathbf{z})) \doteq \text{sgn}(\nabla f(\mathbf{x})),$ 
         $\text{sgn}(\nabla \mathbf{g}(\mathbf{z})) \doteq \text{sgn}(\nabla \mathbf{g}(\mathbf{x})), \text{sgn}(\nabla \mathbf{h}(\mathbf{z})) \doteq \text{sgn}(\nabla \mathbf{h}(\mathbf{x}))\}$ 
    else
       $\tilde{\mathcal{N}}_k(\mathbf{x}) = \{\mathbf{z} : \mathbf{z} \in \mathcal{N}_k(\mathbf{x}), \text{sgn}(\nabla \mathbf{g}(\mathbf{z})) \doteq \text{sgn}(\nabla \mathbf{g}(\mathbf{x})),$ 
         $\text{sgn}(\nabla \mathbf{h}(\mathbf{z})) \doteq \text{sgn}(\nabla \mathbf{h}(\mathbf{x}))\}$ 
    end if
    if  $\mathbf{x}$  is feasible then
      if  $\forall \mathbf{z} \in \tilde{\mathcal{N}}_k(\mathbf{x}) f(\mathbf{z}) > f(\mathbf{x})$  then refine  $\mathbf{x}$ 
    else
      if insufficient diversity w.r.t. any constraint
        if  $\exists i [g_i(\mathbf{x}) > 0, \forall \mathbf{z} \in \tilde{\mathcal{N}}_k(\mathbf{x}) \overline{g_i(\mathbf{z})} > g_i(\mathbf{x})]$ 
           $\vee \exists j [h_j(\mathbf{x}) \neq 0, \forall \mathbf{z} \in \tilde{\mathcal{N}}_k(\mathbf{x}) \overline{h_j(\mathbf{z})} > h_j(\mathbf{x})]$ 
          then refine  $\mathbf{x}$ 
        else
          if  $\exists i [g_i(\mathbf{x}) > 0, \forall \mathbf{z} \in \tilde{\mathcal{N}}_k(\mathbf{x}) g_i(\mathbf{z}) > g_i(\mathbf{x})]$ 
             $\vee \exists j [h_j(\mathbf{x}) \neq 0, \forall \mathbf{z} \in \tilde{\mathcal{N}}_k(\mathbf{x}) h_j(\mathbf{z}) > h_j(\mathbf{x})]$ 
            then refine  $\mathbf{x}$ 
          end if
        end if
      end if
    end for
  end while
```

Instantiation of the CORE-MA in the form of CLS-powered MA has finally concluded this chapter. Earlier in the chapter, instantiations of the CORE-MA had also taken place in the forms of original, generalized, as well as ecumenical FSM-driven MA. A full quick recapitulation is provided below.

The introduction of CORE-MA in this chapter begins with discussing some preliminaries, covering the In-feasibility Class (Section 4.2.1) and the Optimum Immediacy Corollary (Section 4.2.2). The In-feasibility Class notion is intended to serve as the preliminary to general CORE-MA capable of handling problems having both the equality and inequality constraints. The Optimum Immediacy Corollary, on the other hand, is derived to outline the consequences of being in the close proximity of some optimum—the necessary conditions to mine before refining an individual.

The FSM is subsequently proposed for modeling a local feasibility structure around an individual through the formulation of a classification problem, which is then solved using the SVM. Next, the model is used to estimate the location of the individual on the problem topology to decide if it truly is worth refining. Given an inequality-constrained problem, the original FSM-driven MA aims at classifying the search space into spaces of feasible and infeasible solutions, with respect to the aggregate of all the (inequality) constraints. To handle the small ratio of the feasible to the entire search space which characterizes the equality-constrained problem, the generalized FSM-driven MA additionally attempts to divide the search space into spaces of positive and negative infeasible solutions, with respect to each equality constraint. Finally, to tackle the fact that a set of inequality constraints may also produce a tiny ratio of the feasible to the entire search space, the ecumenical FSM-driven MA aims at distinguishing the search space into the spaces of feasible and infeasible solutions with respect to each—rather than the aggregate—of the inequality constraints.

The use of SVM in the FSM obviously incurs a considerable overhead. This then leads to the proposal of CLS that completely eliminates the need for SVM by making use of some gradient information to stratify the problem landscape. Direct quality comparison between one individual and the relevant neighbors is hence allowed. Instantiated as the CLS-powered MA, simple nearest-neighbors-based dichotomizer replaces the expensive SVM to determine the worthiness of the individual to undergo local refinement.

Hereinafter, both the original and the generalized FSM-driven MAs shall be also referred to as "FSM". The original FSM-based MA is as the matter of fact a special case of the generalized FSM-based MA, in the absence of any equality constraints. Furthermore, the ecumenical FSM-based MA shall also be denoted as "e-FSM". Finally, the CLS-powered MA shall be interchangeably referred to as "CLS" for brevity.

Chapter 5

Empirical Results and Discussions

In this chapter, the efficacy of the proposed CORE-MA is tested against some benchmark problems compiled in [99]. They are as tabulated in Table 5.1. The first four problems are with both equality and inequality constraints. The next six are with equality constraints only whereas the remaining thirteen are with inequality constraints only. All the experiments were aimed to emulate the use of the proposed CORE-MA under the conditions normally encountered in the optimization of computationally-expensive problems: small population, relatively low budget. Furthermore, a zero in numerical implementation is frequently treated as a range of very small value. In the experiments presented in this chapter, a tolerance of in the order of 10^{-10} has been used. With this, feasible solutions $\mathbf{x}$ with respect to an equality constraint h have constraint values of $|h(\mathbf{x})| \leq 10^{-10}$ and with respect to inequality constraint g have constraint values of $g(\mathbf{x}) \leq 10^{-10}$. Similarly, a solution is said to have converged to the global optimum if the difference between the objective values of the solution and that of the global optimum is at most 10^{-10} .

Table 5.1: Characteristics of Benchmark Problems

Problem	n	Objective	n_g	n_h	Constraints	$\rho^\dagger$	$f(\mathbf{x}^*)$
g05	4	cubic	2	3	nonlinear	0.0000%	5,126.4981095953
g21	7	linear	1	5	nonlinear	0.0000%	193.7869252615
g22	22	linear	1	19	nonlinear	0.0000%	236.7869252615
g23	9	linear	2	4	quadratic	0.0000%	-400.0000000000
g03	10	polynomial	0	1	quadratic	0.0000%	-1.0000000000
g11	2	quadratic	0	1	quadratic	0.0000%	0.7500000000
g13	5	nonlinear	0	3	cubic	0.0000%	0.0539498478
g14	10	nonlinear	0	3	linear	0.0000%	-47.7610908594
g15	3	quadratic	0	2	quadratic	0.0000%	961.7151721301
g17	6	nonlinear	0	4	nonlinear	0.0000%	8,927.5977354978
g01	13	quadratic	9	0	linear	0.0005%	-15.0000000000
g02	20	nonlinear	2	0	nonlinear	99.9967%	-0.8036191041
g04	5	quadratic	6	0	quadratic	26.9127%	-30,665.5386717833
g06	2	cubic	2	0	quadratic	0.0059%	-6,961.8138755802
g07	10	quadratic	8	0	quadratic	0.0000%	24.3062090682
g08	2	nonlinear	2	0	quadratic	0.8750%	-0.0958250414
g09	7	polynomial	4	0	polynomial	0.5249%	680.6300573744
g10	8	linear	6	0	quadratic	0.0007%	7,049.2480205287
g12	3	quadratic	1	0	quadratic	4.8065%	-1.0000000000
g16	5	nonlinear	38	0	nonlinear	0.0000%	-1.9051552585
g18	9	quadratic	13	0	quadratic	0.0000%	-0.8660254038
g19	15	nonlinear	5	0	quadratic	33.4711%	32.6555929502
g24	2	linear	2	0	polynomial	44.1929%	5.5080132716

$\dagger$: ratio of the feasible to the entire search space

In the rest of this chapter, results from various experiments observing the settings described in Section 5.1 will be presented. The experiments encompass testing the efficacy of not only the more complicated rational dynamic strategies as elaborated in the previous chapter, but also some simple avaricious static strategies to showcase their limitations. This is to show that it is indeed necessary to introduce the slightly more expensive yet more intelligent assessment strategies for the efficient optimization of computationally-expensive problems. It is then followed by discussing the robustness tests to conclude this chapter.

Note that the following guidelines are generally adhered to when comparing the performance of various instances of the memetic algorithms for the assessment of their efficacy and robustness.

- Higher success rate (S.R.) is better.
- Lower number of evaluations is better.

5.1 Experimental Setups

Due to the stochastic nature of the memetic algorithms, multiple independent runs were conducted for each of the assessment strategies tested. Statistics were then used to appraise the testing results. As many as 30 independent runs were performed for the purpose of comparing the efficacy of various assessment strategies. A total of 150 independent runs were eventually carried out to examine the robustness of selected assessment strategies. Each run of the memetic algorithm employing a particular assessment strategy is equipped with a maximum budget of 40,000 evaluations of the objective and constraint functions of the optimization problem at hand. As the objective and constraint values determine the fitness of an individual, each of these evaluations will be referred to as fitness function evaluation (FFE) hereafter. When the budget has been exhausted or when the global optimum has been successfully identified, whichever is earlier, execution of the memetic algorithm will then be terminated. Throughout, the following settings were observed during the course of the experimentation.

5.1.1 Global Search

For the purpose of the identification of the global optimum given a constrained optimization problem, the genetic algorithm approach described in Section 2.4 is used to perform the global search. A relatively small population size of 10 individuals initialized using the Latin Hypercube Sampling method [100] is assumed to emulate solving computationally-expensive problems for which large population size would not be affordable.

5.1.2 Local Search

In order to refine an individual in the population so as to allow it to converge quickly to an optimum, the method of Sequential Quadratic Programming is employed with a local search budget of $10n$ where n is the dimensionality of the problem. This is to allow most individual to achieve convergence rather than to exhaust the local search budget and leave convergence to local searches in the upcoming generations since in the context of computationally-expensive problems, long evolution might not always be possible.

5.1.3 *Optinformatics*

To allow selective local refinement of individuals in the population, local topology of the optimization problem at hand is learned based on the information revealed by the k nearest neighbors. To ensure locality and at the same time allow reasonable time for the construction of the SVM model, $k = 2n$ is chosen to be the size of the neighborhood where n is the problem size.

5.2 Results Overview

The benchmark problems considered herein possess different levels of difficulty. Some problems are pretty easy to solve, requiring the simple MA to evolve just as long as a couple of generations to successfully identify their global optimum. For some of these problems, the simple MA can even sometimes find the global optimum after refining just a few individuals in the initial population. For such problems, the attempt to intelligently execute local refinements selectively may in some unfortunate cases end up with skipping important refinements through which the global optimum could have been successfully identified. In the really fortunate cases, however, the intelligence of the proposed CORE-MA is able to correctly identify and eliminate the unnecessary refinements, therefore reducing the number of FFEs required to solve the problems. Although CORE-MA may sometimes reduce the FFE requirements, the simple MA should be the method of choice when it comes to solving such easy problems. Within short evolution, the sampling of the problem search space may not have been representative yet due to the stochastic nature of the global search employed in the MA. As such, accuracy of the CORE-MA learning with respect to the true problem topology may be compromised. This in turn may lead to the wrong decision on whether an individual is worth refining. With little room for improvements, dependency of the CORE-MA on the optimization archives may cause delayed convergence, especially in the case of FSM-driven MAs. In the case of CLS-powered MA, on the other hand, the simple MA is resembled when solving some easy problems, yielding similar performance but with higher overhead.

Tackling harder problems, the proposed CORE-MA instances outperformed the simple MA by some considerable margins. Solving them regularly demands sufficiently long evolution, hence large number of generations. Using the simple MA, considerable amount of FFEs may not have been wisely requested through the execution of unnecessary refinements. In contrast, more and more accurate decisions to skip unnecessary refinements over the course of evolution cut down the FFE requirements of the CORE-MA, thanks to the long evolution that has enabled better sampling of the problem search space—hence higher accuracy of the CORE-MA learning with respect to the actual problem topology. Selective local refinement execution in the CORE-MA, furthermore, implies smaller FFE requirements compared to those of the simple MA for the same evolution span. Equivalently, the CORE-MA enables longer evolution than does the simple MA given the same FFE budget, allowing for even better estimation of the problem topology. In consequence, the FSM-driven MAs and the CLS-powered MA had performed more efficiently than the simple MA on hard problems. On the very hard problems that involve tiny ratios of the feasible to the entire search space, the original and generalized FSM-driven MAs may find it somewhat difficult to perform efficiently due to the way they handle the inequality constraints. Then the ecumenical FSM-driven MA alleviates the problem with a higher overhead. Of a notable achievement is the performance of the CLS-powered MA. Making a good use of the information held by the objective and constraint gradients—whenever these are available—the CLS-powered MA is by far the most efficient CORE-MA instantiation.

5.3 Efficacy Tests

In this section, various assessment strategies to determine the significance of an individual in the population for local refinement will be subjected to the efficacy tests. The success rate and some other statistics from the 30 runs of memetic algorithms for each of the assessment strategies under test will be reported and discussed. These encompass the avaricious static strategies as well as some rational dynamic strategies introduced in the previous chapter.

5.3.1 Avaricious Static Strategies

Belonging to this category are memetic algorithms that refine only the few best individuals or the feasible individuals in the current population. The avaricious static strategies essentially assess the significance of each individual in the population using the information directly related to the individual itself, such as its ranking among other individuals in the population and its feasibility. As these strategies depend only on the information revealed by the current population and do not take into account the information revealed by the preceding ones, they are deemed static. From generation to generation, MAs that employ these strategies would always refine the same number of best individuals of the population or individuals with the same quality. In other words, the decision as to which individuals in the population should undergo local refinement remains the same throughout the evolution. Furthermore, as these strategies select only individuals that are deemed as the best in the population to undergo local refinement, they form greedy algorithms and are deemed avaricious.

Table 5.2 presents from left to right the results of performing 30 independent runs of the memetic algorithms that refine all individuals in the population as well as only the top 10%, the top 20%, the top 30%, and the feasible individuals in the population. Using the last strategy, local refinements are also performed until at least one feasible trial solution is found. The results are presented in terms of identification rate and some descriptive statistical measures. From the table, it is observable that the simple MA may be outperformed when solving a number of benchmark problems but would easily win in solving the other problems. This lack of robustness in employing the avaricious static strategies indicates the necessity of a more intelligent assessment strategy.

5.3.1.1 Few Best Individuals

Refining only the few best individuals in the population would be the direct intuition to limit the number of evaluations performed in a single generation. With less number of evaluations in each generation, it is expected that the total number of evaluations performed over the course of searching for the global optimum would be reduced compared to that of the simple MA assuming the same length of evolution. This is, however, not observed in general. Refining only few best individuals in the population easily causes early convergence to an optimum identified by the first few refinements. In multimodal landscape, this can easily promote convergence to the local optimum and prolong the convergence to the global optimum since there exists high probability that the result of the refinement would be the best individuals over many generations.

Table 5.2: FFE Count Statistics of MAs Using Avaricious Static Strategies over 30 Independent Runs

Problem	Measure	All	Top 10%	Top 20%	Top 30%	Feasible
g05	S.R.	100.00%	100.00%	100.00%	100.00%	93.33%
	MIN	190	20	35	46	20
	Q1	229	24	45	62	24
	Q2	245	37	55	76	37
	Q3	261	44	71	98	47
	MAX	285	50	87	118	>40,000
	AVG	245	35	57	80	>35
g21	S.R.	100.00%	56.67%	93.33%	93.33%	83.33%
	MIN	205	24	29	39	24
	Q1	300	80	86	94	80
	Q2	365	116	112	157	80
	Q3	404	>40,000	161	220	4,440
	MAX	508	>40,000	>40,000	>40,000	>40,000
	AVG	356	>89	>133	>177	>1,687
g23	S.R.	100.00%	63.33%	96.67%	100.00%	63.33%
	MIN	579	66	74	114	66
	Q1	633	100	111	168	100
	Q2	714	100	158	196	100
	Q3	757	>40,000	190	249	>40,000
	MAX	888	>40,000	>40,000	280	>40,000
	AVG	705	>94	>166	208	>256
g22	S.R.	100.00%	10.00%	53.33%	70.00%	93.33%
	MIN	92	215	121	174	83
	Q1	184	>40,000	584	409	182
	Q2	349	>40,000	7,120	764	350
	Q3	581	>40,000	>40,000	>40,000	545
	MAX	1,170	>40,000	>40,000	>40,000	>40,000
	AVG	448	>392	>1,503	>584	>662

continued on the next page ...

Table 5.2 – continued from the previous page

Problem	Measure	All	Top 10%	Top 20%	Top 30%	Feasible
g11	S.R.	100.00%	100.00%	100.00%	100.00%	93.33%
	MIN	111	18	25	32	18
	Q1	132	20	32	42	20
	Q2	145	24	39	53	26
	Q3	156	30	47	61	30
	MAX	176	50	50	70	>40,000
	AVG	144	25	39	53	>25
g15	S.R.	100.00%	96.67%	100.00%	100.00%	93.33%
	MIN	35	24	24	31	24
	Q1	64	40	40	40	40
	Q2	82	40	56	66	40
	Q3	109	55	70	70	40
	MAX	230	>40,000	110	170	>40,000
	AVG	89	>51	57	65	>42
g13	S.R.	100.00%	46.67%	90.00%	96.67%	66.67%
	MIN	220	41	55	55	41
	Q1	345	60	90	114	60
	Q2	371	>40,000	113	160	160
	Q3	413	>40,000	247	253	>40,000
	MAX	487	>40,000	>40,000	>40,000	>40,000
	AVG	373	>305	>1,411	>333	>100
g17	S.R.	100.00%	100.00%	100.00%	100.00%	93.33%
	MIN	322	26	55	78	26
	Q1	347	35	75	110	35
	Q2	377	49	86	121	50
	Q3	411	63	95	136	65
	MAX	444	70	115	164	>40,000
	AVG	380	48	85	121	>48

continued on the next page ...

Table 5.2 – continued from the previous page

Problem	Measure	All	Top 10%	Top 20%	Top 30%	Feasible
g03	S.R.	100.00%	63.33%	100.00%	100.00%	63.33%
	MIN	605	102	117	151	102
	Q1	769	110	210	249	110
	Q2	831	143	210	310	165
	Q3	904	>40,000	386	310	>40,000
	MAX	1,010	>40,000	1,868	2,647	>40,000
	AVG	832	>136	370	442	>1,804
g14	S.R.	100.00%	60.00%	100.00%	100.00%	80.00%
	MIN	1,010	220	419	310	220
	Q1	2,020	6,787	1,039	928	6,454
	Q2	3,030	15,665	2,695	1,613	17,380
	Q3	4,971	>40,000	3,764	3,712	30,021
	MAX	8,075	>40,000	12,385	7,327	>40,000
	AVG	3,764	>9,733	3,075	2,612	>13,552
g06	S.R.	100.00%	100.00%	100.00%	100.00%	93.33%
	MIN	197	24	44	57	24
	Q1	210	30	50	70	30
	Q2	210	30	50	70	30
	Q3	210	30	50	70	30
	MAX	210	58	50	70	>40,000
	AVG	209	31	50	69	>30
g08	S.R.	100.00%	83.33%	96.67%	93.33%	93.33%
	MIN	179	30	50	64	30
	Q1	404	60	100	138	96
	Q2	417	204	197	140	227
	Q3	419	9,969	434	323	510
	MAX	796	>40,000	>40,000	>40,000	>40,000
	AVG	418	>2,317	>2,244	>2,189	>634

continued on the next page ...

Table 5.2 – continued from the previous page

Problem	Measure	All	Top 10%	Top 20%	Top 30%	Feasible
g24	S.R.	100.00%	100.00%	100.00%	100.00%	93.33%
	MIN	94	17	24	30	34
	Q1	108	21	28	37	48
	Q2	119	23	32	43	57
	Q3	133	25	40	48	71
	MAX	139	597	1,223	3,335	>40,000
	AVG	119	50	101	153	>59
g12	S.R.	100.00%	100.00%	100.00%	100.00%	93.33%
	MIN	64	12	14	17	12
	Q1	89	14	18	44	15
	Q2	119	16	43	49	38
	Q3	148	40	46	74	40
	MAX	201	207	70	98	>40,000
	AVG	121	37	39	54	>39
g04	S.R.	100.00%	100.00%	100.00%	100.00%	93.33%
	MIN	245	18	30	42	18
	Q1	338	22	68	97	60
	Q2	366	60	75	121	98
	Q3	420	60	110	137	148
	MAX	510	77	110	160	>40,000
	AVG	373	44	79	118	>96
g16	S.R.	100.00%	100.00%	100.00%	100.00%	93.33%
	MIN	67	176	176	176	67
	Q1	151	395	387	387	170
	Q2	384	494	470	470	437
	Q3	677	739	739	739	739
	MAX	980	1,142	980	980	>40,000
	AVG	423	543	528	528	>436

continued on the next page ...

Table 5.2 – continued from the previous page

Problem	Measure	All	Top 10%	Top 20%	Top 30%	Feasible
g09	S.R.	100.00%	100.00%	100.00%	100.00%	93.33%
	MIN	704	80	144	214	80
	Q1	710	80	150	220	80
	Q2	710	159	150	220	150
	Q3	710	236	295	220	229
	MAX	710	475	449	627	>40,000
g10	AVG	710	167	203	270	>150
	S.R.	100.00%	96.67%	100.00%	100.00%	93.33%
	MIN	423	41	85	119	41
	Q1	544	68	124	167	69
	Q2	595	134	130	197	129
	Q3	673	251	271	355	251
g18	MAX	1,228	>40,000	672	803	>40,000
	AVG	619	>890	214	260	>340
	S.R.	100.00%	93.33%	100.00%	100.00%	90.00%
	MIN	119	35	37	37	35
	Q1	170	44	65	70	42
	Q2	223	64	90	106	52
g07	Q3	264	104	131	133	103
	MAX	371	>40,000	460	380	>40,000
	AVG	223	>98	111	111	>68
	S.R.	100.00%	100.00%	100.00%	100.00%	93.33%
	MIN	624	56	113	179	56
	Q1	671	71	136	199	71
g07	Q2	696	79	147	218	79
	Q3	724	88	161	235	89
	MAX	776	97	173	262	>40,000
	AVG	698	78	147	217	>77

continued on the next page ...

Table 5.2 – continued from the previous page

Problem	Measure	All	Top 10%	Top 20%	Top 30%	Feasible
g01	S.R.	100.00%	86.67%	100.00%	100.00%	93.33%
	MIN	92	12	16	18	12
	Q1	141	612	39	51	148
	Q2	197	2,204	135	96	472
	Q3	262	4,652	551	320	1,822
	MAX	1,193	>40,000	2,699	3,730	>40,000
	AVG	241	>2,075	359	374	>1,066
g19	S.R.	100.00%	100.00%	100.00%	100.00%	93.33%
	MIN	934	78	156	233	100
	Q1	1,027	88	182	313	315
	Q2	1,102	97	238	322	366
	Q3	1,159	160	247	373	480
	MAX	1,336	160	310	434	>40,000
	AVG	1,108	113	226	333	>367
g02	S.R.	30.00%	0.00%	33.33%	40.00%	46.67%
	MIN	11,503	-	23,593	10,065	17,293
	Q1	32,675	-	37,828	24,557	29,311
	Q2	>40,000	-	>40,000	>40,000	>40,000
	Q3	>40,000	-	>40,000	>40,000	>40,000
	MAX	>40,000	-	>40,000	>40,000	>40,000
	AVG	>26,218	-	>30,951	>22,921	>28,607

Complete success has been witnessed in solving problems g05, g11, g17, g04, g06, g07, g09, and g19 in that refining only the top few individuals were always more efficient than the simple MA. Complete failure, on the other hand, has been observed when solving problems g22 and g16. Either the top-10% or top-20% strategy is also observed to have achieved complete success when solving problems g23, g14, g15, g02, g10, and g12 whereas the top-10% strategy has completely failed to assist MA to perform more efficiently. For the rest, refining the top few individuals has suffered from partial failure or in other words, gained only partial success. At a glimpse, the ability of the three strategies to help MA identify correctly the unnecessary local refinements looks very much dependant on the problem to solve and the number of best individuals refined.

Among the failures, many are due to the top-10% strategy. With a population size of 10 individuals, this strategy is equivalent to refining only the single best individual in the population. When the best individual was in fact a result from past local refinement in the preceding generation which is then passed on to the current generation by means of elitism, refining this best individual will clearly fail to improve its quality. Unless the refinement result is the global optimum, the search will continue. With no other refinement performed, the MA search would then resemble a GA search. This easily causes prolonged or even failed convergence to the global optimum. In Table 5.2, they are signified by large FFE requirement and exhaustion of the entire search budget, respectively. Employing the top-20% or top-30% strategy, this phenomenon may still be observed when the best individuals converge to the same local optimum.

From the table, it is also observable that the top-10% strategy would be sufficient to solve some problems. This is understandably due to the relatively short evolution length needed to identify the global optimum. For some problems, this may be achievable only when the initial population has a sufficiently good distribution. With the first few local refinements already converging to the global optimum, resemblance to pure GA would not take effect. For problems that cannot be solved efficiently using the top-10% strategy, the top-20% strategy could be sufficient whereas the top-30% strategy might be necessary. For the remaining problems that still cannot be solved efficiently, more individuals in the population might need to be refined. This indicates the number of individuals in the population that should undergo local refinement varies from one problem to another and may also vary from one population to the others depending on their distributions.

5.3.1.2 Feasible Individuals

Unlike in the previous section where a fixed number of best individuals given a fixed population size would undergo local refinement, refining only the feasible individuals in the population would vary the number of individuals that would undergo the local refinement. In general, it is obvious that this strategy would not be plausible on problems with equality constraints due to the tiny ratio of the feasible to the entire solution space. As such, local refinements were always assumed significant until at least a feasible solution is encountered. Only then the local refinement were performed on the feasible individuals only.

From Table 5.2, this strategy is observed to have achieved partial success on all problems except g16 on which this strategy had completely failed to assist MA to perform more efficiently in all cases. For problems with sufficiently large ratio of the feasible to the entire solution space, the initial population would undoubtedly be populated by some feasible individuals. For problems with tiny ratio, it is very likely that the initial population are entirely populated by infeasible individuals. With local refinement executed until at least one feasible individual is encountered, the order by which the infeasible individuals undergo refinement is crucial. Depending on the individuals being refined and the difficulty of the problems, refining an infeasible individual may fail to produce a feasible individual. In such event, the next infeasible individual in the population would be refined until a feasible individual is produced. In the very fortunate cases, the first encountered feasible solution is either the global optimum itself or an estimate that is sufficiently close to the global optimum. In the former, the MA execution would be terminated right away. In the latter, the feasible estimate will be further refined, leading to the discovery of the true global optimum. In the unfortunate cases, however, the first encountered feasible solution could be a local optimum. With feasible solutions regarded as better than the infeasible ones, convergence would then be geared towards this local optimum. This easily results in delay or failure in the identification of the global optimum. In Table 5.2, this is signified by the larger FFE requirement and the smaller identification rate, respectively. This suggests that refining some infeasible individuals after encountering a feasible one could be beneficial if chosen properly.

5.3.2 Rational Dynamic Strategies

In contrast to the avaricious static strategies described above, the rational dynamic strategies essentially assess the significance of each individual in the population for local refinement by mining information from its neighborhood. This information is dynamically revealed as more and more trial solutions to the optimization problem at hand are generated and evaluated. That is because the neighborhood would change over the course of the evolution. Based on this, local models of the problem landscape are constructed and intelligent decisions based on some rationales are drawn with the help of machine learning techniques that range from the simple nearest neighbors estimation to the complex SVM. The rational dynamic strategies encompass a simple strategy that refines individuals with feasibly uniform or composite neighborhood upon sufficient diversity in the lookup table and the various CORE-MA instances presented in the previous chapter. Employing the notion of the composite neighborhood in the ecumenical FSM, encountering compositeness in the neighborhood of an individual with respect to some constraints as well as both uniformity and feasibility with respect to the remaining constraints would deem the individual significant for local refinement. During the early generations of the search, however, there may not have been sufficient diversity in the values of some constraints. As with other CORE-MA strategies, the local refinement would constantly be executed in such case until there are sufficient diversity on the values of all constraints. In Table 5.3, this simple neighborhood-based strategy will be referred to as FCN whereas the other CORE-MA strategies are denoted as FSM, e-FSM, and CLS.

Table 5.3: FFE Count Statistics of MAs with Rational Dynamic Strategies over 30 Independent Runs

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g05	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	190	46	159	159	40
	Q1	229	86	206	207	104
	Q2	245	119	229	229	127
	Q3	261	152	257	257	150
	MAX	285	205	280	280	189
	AVG	245	121	229	231	125
g21	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	205	205	198	198	174
	Q1	300	300	256	256	212
	Q2	365	326	309	309	255
	Q3	404	378	363	363	322
	MAX	508	508	508	508	465
	AVG	356	336	316	316	274
g23	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	579	147	67	322	579
	Q1	633	578	129	432	625
	Q2	714	626	219	521	710
	Q3	757	720	308	618	757
	MAX	888	888	630	725	888
	AVG	705	626	234	510	702
g22	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	92	92	92	92	87
	Q1	184	184	182	182	179
	Q2	349	309	340	340	337
	Q3	581	581	580	580	578
	MAX	1,170	1,866	1,170	1,170	1,083
	AVG	448	492	436	436	391

continued on the next page ...

Table 5.3 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g11	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	111	52	86	86	27
	Q1	132	70	111	111	38
	Q2	145	84	123	123	50
	Q3	156	98	142	142	60
	MAX	176	124	159	159	78
g15	AVG	144	85	125	125	50
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	35	34	35	35	24
	Q1	64	40	64	64	40
	Q2	82	61	70	70	40
	Q3	109	70	109	109	40
g13	MAX	230	140	230	230	149
	AVG	89	58	87	87	48
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	220	165	219	219	220
	Q1	345	235	331	331	293
	Q2	371	287	360	360	351
g17	Q3	413	322	410	410	380
	MAX	487	7,919	487	487	437
	AVG	373	664	360	360	338
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	322	79	285	285	35
	Q1	347	126	345	345	105
g17	Q2	377	160	369	369	119
	Q3	411	211	399	399	149
	MAX	444	308	435	435	284
	AVG	380	170	370	370	133

continued on the next page ...

Table 5.3 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g03	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	605	102	164	164	102
	Q1	769	210	573	573	110
	Q2	831	217	755	755	124
	Q3	904	318	830	830	236
	MAX	1,010	1,156	1,010	1,010	615
	AVG	832	311	700	700	196
g14	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	1,010	1,010	610	610	220
	Q1	2,020	2,020	1,829	1,829	975
	Q2	3,030	3,030	2,482	2,482	1,730
	Q3	4,971	4,971	3,847	3,847	3,927
	MAX	8,075	8,075	7,382	7,382	7,646
	AVG	3,764	3,764	2,885	2,885	2,424
g06	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	197	24	24	24	197
	Q1	210	30	30	30	210
	Q2	210	30	30	30	210
	Q3	210	30	30	30	210
	MAX	210	50	50	50	210
	AVG	209	33	31	31	209
g08	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	179	70	50	129	179
	Q1	404	100	99	219	300
	Q2	417	173	174	279	345
	Q3	419	288	306	335	394
	MAX	796	987	636	666	561
	AVG	418	226	213	289	350

continued on the next page ...

Table 5.3 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g24	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	94	73	58	70	75
	Q1	108	94	83	91	93
	Q2	119	106	97	102	104
	Q3	133	114	103	113	118
	MAX	139	137	124	133	132
g12	AVG	119	105	94	102	103
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	64	48	13	13	55
	Q1	89	80	40	40	84
	Q2	119	87	47	47	108
	Q3	148	116	76	76	140
g04	MAX	201	162	110	110	170
	AVG	121	98	56	56	107
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	245	245	39	194	130
	Q1	338	326	163	297	277
	Q2	366	366	218	332	324
g16	Q3	420	397	274	393	353
	MAX	510	510	359	471	510
	AVG	373	368	213	341	318
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	67	67	67	67	67
	Q1	151	160	151	151	151
g04	Q2	384	570	384	393	384
	Q3	677	764	677	677	677
	MAX	980	1,257	980	980	980
	AVG	423	506	417	429	423
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	67	67	67	67	67

continued on the next page ...

Table 5.3 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g09	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	704	150	80	430	704
	Q1	710	150	80	570	710
	Q2	710	220	150	570	710
	Q3	710	360	229	640	710
	MAX	710	510	347	710	710
g10	AVG	710	252	163	600	710
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	423	120	41	422	423
	Q1	544	192	69	523	544
	Q2	595	368	166	588	595
	Q3	673	581	284	654	673
g18	MAX	1,228	918	575	1,228	1,191
	AVG	619	411	203	610	618
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	119	107	44	98	119
	Q1	170	159	63	170	170
	Q2	223	184	100	217	223
g07	Q3	264	233	147	255	264
	MAX	371	28,608	228	313	371
	AVG	223	1,434	105	215	223
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	624	113	56	552	624
	Q1	671	146	72	671	671
g07	Q2	696	157	79	696	696
	Q3	724	215	89	724	724
	MAX	776	448	119	776	776
	AVG	698	189	80	695	698

continued on the next page ...

Table 5.3 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g01	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	92	92	14	22	92
	Q1	141	141	50	89	141
	Q2	197	197	110	157	193
	Q3	262	262	156	201	229
	MAX	1,193	1,193	295	324	424
	AVG	241	241	114	156	201
g19	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	934	934	588	934	632
	Q1	1,027	1,027	829	1,027	807
	Q2	1,102	1,102	952	1,102	864
	Q3	1,159	1,159	1,055	1,159	926
	MAX	1,336	1,336	1,336	1,336	1,134
	AVG	1,108	1,108	954	1,108	871
g02	S.R.	30.00%	50.00%	46.67%	46.67%	63.33%
	MIN	11,503	14,893	21,669	21,669	11,140
	Q1	32,675	23,593	32,160	32,160	17,968
	Q2	>40,000	>40,000	>40,000	>40,000	26,945
	Q3	>40,000	>40,000	>40,000	>40,000	>40,000
	MAX	>40,000	>40,000	>40,000	>40,000	>40,000
	AVG	>26,218	>24,629	>30,981	>30,981	>21,991

Testing the efficacy of the simple neighborhood-based assessment strategy in this section is meant to showcase that it is indeed possible to improve the efficiency of the simple MA by focussing the search effort on the regions where local optima may be located as one of them would be the global optimum. As implied by the optimum immediacy corollary, local optima may be located only in the feasibly uniform or the composite neighborhood. Intuitively, focussing the search effort in these regions could eliminate the unnecessary refinement in the uniform neighborhood that contains only the infeasible solutions. It is expected, therefore, that the FFE requirement of the simple MA would consequently be reduced. More reductions should be achievable when further restrictions are imposed on the feasibly uniform and the composite neighborhood as with the case of either the FSM-driven MAs or the CLS-powered MA.

As can be seen from Table 5.3, the simple neighborhood-based assessment strategy successfully assists MA to solve most benchmark problems more efficiently than or at least as efficiently as the simple MA. An exception, however, is observed for problem g16. With the amount of constraints that outnumbers the dimensionality of the problem by some multiples, it is conceivably quite difficult to find an individual with composite neighborhood with respect to each of the constraints unless the individual is in the feasibly uniform neighborhood with respect to that constraint. Another exception is also observed for problem g22, g13, g03, g08, and g18 in which the simple MA had performed more efficiently in the worst case. This is very likely due to the minute ratio of their feasible to the entire solution space or the multimodality in the case of problem g08.

Comparing the MA with the simple neighborhood-based assessment strategy against the CORE-MA, it is observed that there would always be an instance of the CORE-MA that had performed either more efficiently or just as efficiently. This shows that further restriction on the feasibly uniform or the composite neighborhood indeed reduces the FFE requirement as intuitively suggested. In the following, the CORE-MA instances will be discussed in greater detail. Note that the column FSM in Table 5.3 depicts the original FSM designed for problems with inequality constraints (treated as an aggregate) and generalized to handle the equality constraints. The column e-FSM then refers to the ecumenical FSM in which inequality constraints are assessed individually. Finally, the column CLS denotes the strategy of the constrained landscape stratification.

5.3.2.1 FSM-driven MA

The original and the generalized FSM-based MA—referred to as FSM in Table 5.3—shows better or comparable performance to the simple MA in solving all the benchmark problems considered herein. Although similar results are observed in some cases, this FSM-based MA had always reduced the FFE requirements of the simple MA on average. In solving problem g16 in particular, the original FSM-based MA seems to mimic exactly the performance of the simple MA except for some runs where it had performed more efficiently, resulting in the lower average FFE requirement. When solving problem g02, it has been witnessed that this FSM-based MA had achieved higher rate of the global optimum identification with somewhat worse best-case performance.

When compared to the MA that employs the simple neighborhood-based assessment strategy, this FSM-based MA turns out to have performed more efficiently or at least comparably in solving the inequality-constrained problems for which it was primarily designed. This is as expected since the original FSM-based MA imposes further restriction on the execution of the local refinement upon encountering the composite neighborhood with respect to the aggregate inequality constraint. Exception, however, is observed in solving problem g02. Lower successes in identifying the global optimum were resulted although it is still much better than what was achieved by the simple MA. This is easily caused by the large ratio of the feasible to the entire solution space and the fact that the global optimum is located at the feasibility boundary. With highly multimodal objective landscape, there is no doubt that sufficiently long evolution would be needed to solve problem g02. Over time, the estimates would densely populate the surrounding of the feasibility boundary segment that hosts the global optimum. Refining every individual with composite neighborhood would definitely reach the global optimum faster as there is no risk of skipping the local refinement of any significant individuals. Refining these individuals restrictively, on the other hand, it is possible that local refinement of a significant individual is skipped, delaying the successful identification of the global optimum. This is an indication that the efficacy of the FSM-based MA to achieve better efficiency by refining individuals with composite neighborhood restrictively rather than refining all such individuals is very much dependant on the distribution of the data generated from the previous generations.

In solving problems with equality constraints, the simple neighborhood-based strategy requires a significant individual to be of composite neighborhood with respect to each of these constraints and be of either composite or feasibly uniform neighborhood with respect to the inequality constraints. The generalized FSM-based MA, on the other hand, refines every individual that is significant with respect to any equality constraint as long as it is located in the significance area of the aggregate inequality constraint. This is because compositeness with respect to every equality constraint could be a very difficult encounter especially with the imposition of further restriction on the composite neighborhood. Under normal condition whereby no such further restriction was imposed, finding the intersection of the boundaries of all the equality constraints could be difficult as evidently shown by the better performance of the FSM-based MA in solving problems g21, g23, g22, g13, and g14. In solving problem g13, however, it is observed that the success of the generalized FSM-based MA to perform more efficiently on average is due to delayed convergence in the worst case of using the simple neighborhood-based assessment strategy. In general, however, the MA with the simple neighborhood-based strategy can identify the global optimum in a single generation following some local refinements due to either the composite neighborhood with respect to all constraints or the insufficient diversity. With the more restricted local refinements of the FSM-based MA, longer evolution is necessary. This explains the less efficient performance of the generalized FSM-based MA as compared to the MA with the simple neighborhood-based strategy in solving problems g05, g11, g15, g17, and g03.

In the fourth column of Table 5.3, the FFE requirements of the ecumenical FSM-based MA in solving the benchmark problems are tabulated. It is observed that it had not performed more efficiently than the original or the generalized FSM-based MA even though it had always performed more efficiently or at least comparably to the simple MA. Recall that the ecumenical FSM-based MA was proposed to overcome the possible event of failure of the original or the generalized FSM-based MA to efficiently identify the global optimum when the ratio of the feasible to the entire solution space of the problem is extremely minute. In the first 30 runs of the original or the generalized FSM-based MA reported in Table 5.3, however, there has been no such adversary observed. The discussion of the ecumenical FSM-based MA would therefore be deferred until the next section where robustness test results are presented.

5.3.2.2 CLS-powered MA

Similar to the FSM-driven MAs, the CLS-powered MA had always performed more efficiently or at least comparably to the simple MA as seen in Table 5.3. Quite notable is that it had performed extremely efficiently in solving problems g05, g21, g11, g15, g17, g03, g14, g19, and g02 with at least 20% and up to 76% reduction in the FFE requirements on average as compared to the simple MA. At best, more than 80% reduction was achievable when solving problem g17 and g03. When used to solve some of the remaining benchmark problems, particularly those that involve inequality constraints and is solvable relatively easily, the CLS-powered MA may however perform just as efficiently.

In general, the CLS-powered MA had performed more efficiently than the simple MA in solving difficult problems that require long evolution or large number of generations to solve. Constrained optimization problems are difficult when local refinements are easily trapped in any of the local optima. This implies the presence of numerous strata in the landscape of the problems, which further signifies the high complexity of the problems. Over sufficiently long evolution, however, the population would represent estimates that are closer and closer to the true global optimum. The individuals would therefore be much more densely populated near the global optimum. With extensive exploitation of the surrounding regions, the CLS-powered MA would actively preventing local refinements from being executed until significant individuals are encountered. Refining these individuals would undoubtedly have higher chance of locating the global optimum successfully. It is then understandable that with the cost-cutting resulted from restricting many of the local refinements, the CLS-powered MA is consequently able to give better chance to the global search procedure to produce better global optimum estimates than the simple MA. When on the contrary the problems are complex but solvable within a small number of generations, the CLS-powered MA would most likely perform just as efficiently as the simple MA. Early in the evolution, individuals in the population are sparsely distributed and there have not been much exploitations. Therefore, the CLS-powered MA would easily decide to refine these individuals as they likely belong to different strata due to the complexity of the problems. This in turn easily resembles the simple MA, hence similar performance.

For simpler problems like g11 and g15, however, there are relatively small number of strata in the landscape of the problems. Even though these problems are solvable in a single generation, the CLS-powered MA could manage to eliminate some of the local refinements. With the small number of strata, it is more probable for an individual in the population to have neighbors that belong to the same stratum. Consequently, there is a chance of finding a trial solution resulted from the previous exploitation that has better fitness than the individual. In such a case, local refinement of the individual would be skipped. This results in the significant reduction in the FFE requirements as compared to the simple MA.

Compared to the MA that employs the simple neighborhood-based strategy, the CLS-powered MA had also performed more efficiently in most cases except in the case of resemblance to the simple MA. This indicates that in many cases, particularly when the problem at hand requires long evolution to solve or has a minute ratio of the feasible to the entire solution space and a complex landscape, the CLS-powered MA allows better efficiency by restricting local refinement upon encountering an individual with composite or feasibly uniform neighborhood. When either a composite or a feasibly uniform neighborhood with respect to each constraint is easy to find or the first few local refinement of any individual easily converge to the global optimum, the MA with the simple neighborhood-based strategy would definitely excel. Nevertheless, such problems have only small FFE requirements. Even the simple MA should pose no problem in solving these problems.

To ascertain that the success of the CLS-powered MA in solving the benchmark problems more efficiently is due to the right choice of thresholds, Table A.1–A.5 presents the results of performing the CLS-powered MA with different thresholds under different conditions. In the following, the effect of refining the feasible and the infeasible individuals restrictively will be discussed separately.

Refining Feasible Individuals In the CLS-powered MA, feasible individuals may be deep inside the interior of the feasible space or at the feasible space boundary. In either location, a feasible individual would undergo local refinement if it has the best objective value compared to its k nearest neighbors that share the same stratum. Should no neighbor belongs to the same stratum, the individual would also be refined as this indicates no previous exploitation has been performed to that stratum. When used to solve problems with minute ratio of the feasible to the entire solution space, as can be seen from Table A.1 and A.2, the CLS-powered MA performance seems mostly unaffected by the various thresholds applied both in the case of feasible individual in the interior and the boundary of the feasible space. Also observed from these tables, the CLS-powered MA yielded better results on some problems but produced worse results on other problems when employing custom thresholds. For instance, refining the feasible individuals with no neighbor sharing the same stratum only yielded better results on problems g19 but worse on g01. This shows that the originally proposed thresholds that refines a feasible individual as long as it has the best objective value in its stratum would be the best choice.

Refining Infeasible Individuals An infeasible individual must be in the feasible space exterior whether or not it is near the feasible space boundary. In Table A.3–A.5, the results of varying the thresholds employed by the CLS-powered MA for refining an infeasible individual are presented. These encompass the cases of infeasible individual near the feasible space boundary, in the feasible space exterior, and in the case of insufficient diversity. In the last case, the individual is also in the feasible space exterior. Recall that in the CLS-powered MA as proposed in Section 4.4, the infeasible individuals would be refined if it has the lowest violation with respect to any constraint in the case of close proximity to the feasible space boundary. It would also be refined if it has a violation below the average violation of the neighbors in the same stratum in the remaining cases.

In contrast to the refinement of the feasible individuals, the performance of the CLS-powered MA is generally invariant against the thresholds used when solving the inequality-constrained problems as can be seen from Table A.3 and A.4. In solving the remaining problems, however, it is observed that the CLS-powered MA could perform better on some problems should different thresholds be used. Using these thresholds, however, could impair the performance of the CLS-powered MA in solving some other problems. This shows that the choice of refining the infeasible individuals near the feasible space boundary or in the exterior of the feasible space when it has the lowest violation with respect to any violated constraint under sufficient diversity of the encountered values in the lookup table of this constraint would be the best choice.

Under insufficient diversity, the results also show that refining an infeasible individual with a violation that is lower than the average violation of its relevant neighbors with respect to any violated constraint would similarly be the best choice. This further strengthens the postulate that upon insufficient diversity, distribution of the individuals in the population are quite sparse hence needing less stringent criterion to allow slightly worse individual to stand a chance of undergoing local refinement. Noting that the slightly worse individual could have been an estimate to the true global optimum, this allows exploitation of the actually different strata that appear to be the same stratum in the early generation of the search such that the global optimum can be identified quickly and robustly.

5.4 Robustness Tests

In this section, robustness of the memetic algorithms employing selected assessment strategies will be subjected to test. For this purpose, additional 120 runs of memetic algorithms were conducted employing the selected assessment strategies for determining if local refinement of an individual would worth the search effort. The results from a total of 150 runs of different memetic algorithms—five times the number of runs conducted for test in the previous section—will be reported and discussed herein for the simple MA, the MA that employs the simple neighborhood-based strategy, the original and the generalized FSM-driven MA, the ecumenical FSM-driven MA, and the CLS-powered MA. These are denoted as SMA, FCN, FSM, e-FSM, and CLS, respectively, in Table 5.4.

Table 5.4: FFE Count Statistics of MAs with Rational Dynamic Strategies over 150 Independent Runs

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g05	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	163	46	145	145	24
	Q1	228	88	220	221	95
	Q2	251	114	244	244	122
	Q3	266	144	261	261	147
	MAX	331	212	311	311	213
	AVG	249	117	239	240	121
g21	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	205	157	190	190	89
	Q1	331	306	294	294	199
	Q2	391	337	356	356	255
	Q3	424	396	401	401	315
	MAX	585	532	540	540	465
	AVG	382	352	355	355	266
g23	S.R.	100.00%	100.00%	98.67%	100.00%	100.00%
	MIN	422	147	14	113	415
	Q1	625	504	100	419	623
	Q2	678	623	175	497	677
	Q3	752	692	277	602	744
	MAX	888	888	>40,000	847	888
	AVG	682	591	>203	499	678
g22	S.R.	100.00%	99.33%	100.00%	100.00%	100.00%
	MIN	43	43	43	43	43
	Q1	196	199	185	185	171
	Q2	373	372	380	380	317
	Q3	584	601	580	580	477
	MAX	1,199	>40,000	1,184	1,184	1,019
	AVG	428	>429	418	418	343

continued on the next page ...

Table 5.4 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g11	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	99	34	59	59	27
	Q1	128	73	108	108	38
	Q2	142	84	121	121	44
	Q3	153	96	137	137	53
	MAX	195	129	182	182	97
g15	AVG	141	85	120	120	46
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	23	23	23	23	20
	Q1	70	40	68	68	36
	Q2	96	46	95	95	40
	Q3	114	70	112	112	40
g13	MAX	230	140	230	230	140
	AVG	91	54	89	89	43
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	220	131	204	204	196
	Q1	355	217	342	342	310
	Q2	410	260	399	399	360
g17	Q3	454	326	440	440	416
	MAX	2,203	39,640	2,174	2,174	1,782
	AVG	437	785	424	424	384
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	303	70	257	257	28
	Q1	366	142	355	355	101
g17	Q2	395	184	380	380	140
	Q3	433	224	421	421	181
	MAX	512	321	512	512	289
	AVG	398	187	383	383	146

continued on the next page ...

Table 5.4 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g03	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	602	102	164	164	102
	Q1	796	210	628	628	110
	Q2	857	310	762	762	210
	Q3	920	449	875	875	365
	MAX	2,603	3,015	3,656	3,656	3,851
	AVG	889	408	791	791	364
g14	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	1,010	1,010	610	610	209
	Q1	3,023	3,023	2,339	2,339	1,044
	Q2	4,035	4,035	3,641	3,641	1,820
	Q3	5,049	5,049	4,742	4,742	3,101
	MAX	13,111	13,111	12,943	12,943	12,043
	AVG	4,343	4,343	3,938	3,938	2,500
g06	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	197	22	17	17	197
	Q1	210	30	30	30	210
	Q2	210	30	30	30	210
	Q3	210	30	30	30	210
	MAX	210	50	50	50	210
	AVG	209	33	31	31	209
g08	S.R.	100.00%	99.33%	99.33%	100.00%	100.00%
	MIN	179	30	30	79	179
	Q1	389	109	100	219	259
	Q2	410	177	189	279	328
	Q3	418	317	336	346	368
	MAX	1,603	>40,000	>40,000	1,004	655
	AVG	392	>277	>535	297	329

continued on the next page ...

Table 5.4 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g24	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	92	63	55	69	51
	Q1	111	94	84	92	92
	Q2	120	104	95	101	103
	Q3	130	113	104	110	112
	MAX	150	238	7,587	138	144
	AVG	121	105	187	102	102
g12	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	54	46	12	12	52
	Q1	91	76	41	41	83
	Q2	120	88	49	49	112
	Q3	148	114	75	75	137
	MAX	205	352	195	195	204
	AVG	126	99	60	60	108
g04	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	205	205	35	131	130
	Q1	324	321	155	280	264
	Q2	364	356	212	332	299
	Q3	400	396	264	372	351
	MAX	510	510	370	464	510
	AVG	360	357	209	324	305
g16	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	67	67	67	67	67
	Q1	176	293	176	190	176
	Q2	479	565	479	483	479
	Q3	725	780	694	728	725
	MAX	1,318	1,318	1,318	1,318	1,318
	AVG	487	553	480	490	487

continued on the next page ...

Table 5.4 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g09	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	704	80	80	430	704
	Q1	710	150	80	570	710
	Q2	710	194	150	640	710
	Q3	710	290	220	640	710
	MAX	710	923	573	710	710
	AVG	710	237	171	611	710
g10	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	357	64	35	357	357
	Q1	551	198	70	539	551
	Q2	604	414	156	602	604
	Q3	673	572	297	673	673
	MAX	1,482	1,351	861	1,482	1,482
	AVG	648	420	197	643	645
g18	S.R.	100.00%	100.00%	93.33%	100.00%	100.00%
	MIN	88	82	36	88	88
	Q1	182	156	80	179	182
	Q2	230	192	102	228	230
	Q3	274	242	160	273	274
	MAX	697	38,814	>40,000	697	697
	AVG	239	729	>2,192	235	239
g07	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	617	93	47	518	617
	Q1	680	140	69	648	680
	Q2	699	158	80	691	699
	Q3	734	193	89	724	734
	MAX	777	505	239	777	777
	AVG	703	178	84	684	703

continued on the next page ...

Table 5.4 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g01	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	92	92	14	22	92
	Q1	141	141	50	83	139
	Q2	178	178	107	143	175
	Q3	229	229	190	218	213
	MAX	1,193	1,193	3,684	1,065	444
g19	AVG	204	204	196	175	187
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	803	803	436	803	491
	Q1	1,035	1,035	861	1,035	786
	Q2	1,114	1,114	979	1,114	878
	Q3	1,191	1,191	1,082	1,191	1,002
g02	MAX	1,426	1,426	1,365	1,426	1,276
	AVG	1,116	1,116	971	1,116	887
	S.R.	32.67%	50.00%	64.67%	64.67%	66.67%
	MIN	9,283	9,283	7,103	7,103	8,253
	Q1	32,442	26,110	26,780	26,780	20,119
	Q2	>40,000	>40,000	35,043	35,043	29,097
	Q3	>40,000	>40,000	>40,000	>40,000	>40,000
	MAX	>40,000	>40,000	>40,000	>40,000	>40,000
	AVG	>26,516	>25,719	>28,478	>28,478	>23,694

The results tabulated in Table 5.4 are in line with the discussions about the rational dynamic strategies presented earlier with some exceptions. Even though the adaptive MAs presented herein had performed more efficiently or at least comparably to the simple MA in the majority of the 150 runs, it is observable that the MA with the simple neighborhood-based strategy had yielded lower identification rate of the global optimum when solving problems g22, g13, and g08. Additionally, it had also been impaired by the huge worst-case FFE requirements when solving problem g18. With 38,814 FFEs required to identify the global optimum successfully under a budget of 40,000 FFEs, it is close to exhausting the entire budget, hence lowering the identification rate. Refining only individuals with either composite or feasibly uniform neighborhood with respect to every constraint, it has been witnessed how this MA could find it difficult to encounter a composite neighborhood with respect to every constraint when solving problems with extremely minute ratio of the feasible to the entire solution space. It has also been witnessed how this MA could find it difficult to escape from the local optimum in the interior of the feasible space when solving a multimodal problem g08. With sufficiently large ratio of the feasible to the entire solution space, it should arguably be easy to find feasible solutions and the population would in fact be dominated by feasible individuals after sufficiently long evolution. With a combination of unfortunate situation and local refinement for every feasible individual, it is very likely that convergence to the same local optimum become unavoidable, decreasing the population diversity drastically, making it virtually impossible for the GA to escape the local optimum.

Similar observation also applies to the original FSM-driven MA when solving problem g08. However, the run only ended up with huge worst-case FFE requirements exhausting some 50% of the total search budget instead of complete failure to identify the global optimum. In solving problem g24, the original FSM-driven MA had also been trapped in a local optimum at the feasibility boundary. This is again the result of unfortunate circumstances and local refinement for virtually feasible individuals only. After sufficiently long evolution, this FSM-driven MA would decline to refine any infeasible solution in the surrounding of the local optimum as that infeasible space would have been heavily exploited. As the matter of fact, however, refining the right infeasible solution near the local optimum in this problem would eventually bring about convergence to the true global optimum. Since there was virtually no more refinement for the infeasible individuals, convergence to the global optimum could only be resulted from the right feasible solution which in this context is more difficult to produce stochastically. In solving problems g23, g01, and g18, the lower global optimum identification rate or the huge worst-case FFE requirement is simply caused by the minute feasible space of these problems with respect to the inequality constraints. Failure to identify the global optimum in the first few refinement attempts could easily trick the global search component to densely populate the region surrounding the local optimum. With minute feasible space and with previous refinements carried out in this region, it is likely that further refinement would be skipped until a feasible individual near the true global optimum is successfully generated, delaying the convergence until then.

Specifically designed to address the possible issue of failed or delayed convergence to the global optimum of the original/generalized FSM-driven MAs due to the tiny ratio of the feasible to the entire solution space with respect to the inequality constraints thereby skipping many of the actually necessary local refinements is the ecumenical FSM-driven MA. In the fortunate cases, the ecumenical FSM-driven MA had performed just as efficiently if not worse. In the unfortunate cases, however, the original/generalized FSM-driven MAs could end up with failed or delayed convergence to the global optimum as discussed above. Using the ecumenical FSM-driven MA, such phenomenon is not observable as can be seen from the results of solving problem g23, g08, g24, and g01. Even in the worst case, the ecumenical FSM-driven MA had never performed worse than the simple MA, making it more robust than the original and the generalized FSM-driven MAs. While the original and the generalized FSM-driven MAs are more suitable to solve simple problems, the ecumenical FSM-driven MA would be a more robust choice in solving more difficult problems, particularly those with tiny feasible space with respect to the inequality constraints. When performing just as efficiently as the simple MA, it is arguable that the ecumenical FSM-driven MA would consume more computing time compared to the simple MA. In the optimization of computationally expensive problems, however, majority of the computation time would be consumed by the evaluation of the trial solutions, making the overhead of modeling the feasibility structure rather insignificant. Considering the possibility of reducing the FFE requirements in most cases, the use of the ecumenical FSM-driven MA is therefore justifiable.

Employing a similar paradigm in that an infeasible individual would be deemed significant to undergo local refinement should it be significant with respect to any of the constraints is the CLS-powered MA. Coupled with its ability to determine the relevant neighbors out of its k nearest neighbors, fairer comparisons and assessments that lead to significantly lower FFE requirements in solving difficult problem can be seen from Table 5.4. As discussed previously, difficult problems require sufficiently long evolution to solve. The longer the evolution, the better the chance of the CLS-powered MA to prevent the re-exploitations of the previously exploited strata as there would be higher chance of finding multiple trial solutions belonging to the same stratum, making the search for the global optimum more efficient. When on the other hand the problems are solvable within short evolution length, it is unlikely that multiple trial solutions belonging to the same stratum would be found. This easily cause the CLS-powered MA to perform just as efficiently as the simple MA. Like the ecumenical FSM-driven MA, the CLS-powered MA had never performed worse than the simple MA, making it another robust choice of adaptive MA when the first-order derivatives of the problem at hand are readily available. As the matter of fact, the CLS-powered MA is a more economical alternative to the ecumenical FSM-driven MA as it eliminates the need to model the feasibility structure of the optimization problem using the SVM, hence reducing the overhead costs incurred by the CORE-MA paradigm. Optimization of the molecular structure would be the right problem to solve using the CLS-powered MA as will be seen in the next chapter.

5.5 Time Complexity

Hitherto, the efficiency of the proposed CORE-MA instances has been explored only from the empirical perspective by subjecting them to the identifications of the global optimum of several benchmark problems. Prior to further subjecting them to solving some computationally-expensive practical problems, it shall be worthwhile to additionally survey in this section the expected efficiency of both FSM-driven and CLS-powered MAs from the time complexity perspective.

At a glance, global search with time complexity of $\mathcal{O}(n_G)$, local search with time complexity of $\mathcal{O}(n_{LS})$, and fitness evaluation of the trial solutions bearing $\mathcal{O}(n_{FFE})$ time complexity contribute to the sheer complexity of the simple MA as well as the CORE-MA instances, with n_G , n_{LS} , and n_{FFE} respectively being the number of generations gone through, local refinements executed, and fitness evaluations performed. As the global search only evolves the population of trial solutions from one generation to the next, $\mathcal{O}(n_G)$ thus becomes negligible since $\mathcal{O}(n_G) \ll \mathcal{O}(n_{LS}) + \mathcal{O}(n_{FFE})$. The use of CORE-MA is anticipated to suppress n_{LS} and n_{FFE} in comparison to that of the simple MA, hence a smaller overall time complexity. The CORE-MA, however, incurs the following overheads.

1. k nearest neighbors identification with time complexity of $O_{NN} \equiv \mathcal{O}(m^2)$ where $m = |\mathcal{LT}|$ is the size of the lookup table $\mathcal{LT}$. For computationally-expensive problems, it is notable that the number of evaluations that can be afforded within some reasonable time frame is relatively small. Hence, m is also relatively small.

2. Significance assessment with time complexity of $\mathcal{O}_{SA}$ which depends on k as the size of the neighborhood. Due to the use of SVM in approximating the feasibility boundary, $\mathcal{O}_{SA} \equiv \mathcal{O}(k^3)$ in the case of FSM-driven MAs. In the case of CLS-powered MA, $\mathcal{O}_{SA} \equiv \mathcal{O}(k)$. To ensure locality, k must be just a small multiple of n , the problem dimension, thus $k \ll m$ if $n \ll m$.

Denoting the time complexity of a single local search execution and a single fitness function evaluation as $\mathcal{O}_{LS}$ and $\mathcal{O}_{FFE}$, hence $\mathcal{O}(n_{LS}) \equiv n_{LS}\mathcal{O}_{LS}$ whereas $\mathcal{O}(n_{FFE}) \equiv n_{FFE}\mathcal{O}_{FFE}$. Employing the SQP for refinements, $\mathcal{O}_{LS}$ and $\mathcal{O}_{SA}$ are comparable in the case of FSM-driven MAs. Depending on whether a sufficient reduction in term of FFEs can be earned to compensate $\mathcal{O}_{NN}$, there may not be savings in the CPU time taken to handle the inexpensive benchmark problems. In the case of CLS-powered MA, $\mathcal{O}_{LS}$ dominates $\mathcal{O}_{SA}$, making it a lot easier to compensate $\mathcal{O}_{NN}$ with just slight reduction of n_{FFE} , thereby saving some CPU time. It is hence not very attractive to observe the CPU time required to solve the inexpensive benchmark problems tabulated in Table A.6. Solving expensive problems, nevertheless, $\mathcal{O}_{FFE}$ shall dominate the CPU time utilization. Hence, n_{FFE} reduction will surely bring about CPU time savings. When the expensive problems are additionally hard to solve, the entire budget normally would have been exhausted before the actual global optimum can be identified successfully. In such cases, the CORE-MA is expected to be able to find better sub-optimal solutions. This can be observed from the results of applying the CORE-MA on some practical problems as reported in the next chapter.

Chapter 6

Practical Applications

Having showcased the efficacy of the proposed algorithms in the identification of the global optimum of the benchmark problems compiled in [99] in the previous chapter, application of the CORE-MA instances to the real-world problems will be presented in this chapter. In two separate sections below, the search for the global optimum of two practical problems using the CORE-MA procedures will be discussed individually. The first problem is the determination of molecular structure of the biomolecule by means of potential energy minimization subject to the absence of clashes between any pair of atoms. Using the polyalanine as the molecule of interest and the AMBER force field [101] as the molecular model, it will be discussed how the CORE-MA can outperform the simple MA in the minimization of the potential energy. The second problem is the determination of the most stable docked conformation given two biomolecules in the forms of a receptor and a ligand. Extending the AutoDock Vina [102] by means of some strategy that would eventually evolve to become the CLS-powered MA, it will be discussed how the proposed method can further accelerate the AutoDock Vina by a factor of up to 25 times.

6.1 Background

Predictions of antigenic peptides that bind Human Leukocyte Antigens (HLAs) are indispensable in rational vaccine design [103, 104]. The highest polymorphism of the HLA among all the human genes [105] with more than 2,000 known alleles [106] and the vast amount of combinations of the 20 naturally occurring amino acids that may possibly constitute the antigenic peptides pose serious impediment in carrying out exhaustive wet-lab experiments for each and every possible pair of the HLA-peptide complexes. Not only the amount of resources required will be tremendous, but also the amount of time taken to complete all the experiments will be prolonged. Computational predictive tools are developed so as to enable the screening of potentially antigenic peptides from some pathogen's proteome. Thus, wet-lab experiments can be restricted to the pre-selected complexes only, which have high probability of being binders. The existing computational tools, as reviewed and discussed in [107], can generally be classified into two categories: *sequence-based* and *structure-based*.

Sequence-based predictions are essentially attainable through the training of machine-learning techniques using sequence data from wet-lab experiments [108]. The techniques include a decision tree [109], Hidden Markov Model (HMM) [110], Artificial Neural Network (ANN) [111], Extreme Learning Machine (ELM) [112], and Support Vector Machine (SVM) [113]. Information Learning Approach [114], which is based on the principles of information theory, does feature selection to improve the performance of the machine-learning techniques. Being reasonably inexpensive in term of computational costs, these methods are

thus suitable for large-scale screening of the potential HLA binders. Being data-driven, however, the prediction accuracy depends largely on the available data used for training. As long as the experimental data are not yet complete, the prediction accuracy can never reach 100%.

Structure-based predictions involve molecular dynamics [115] and free energy scoring functions [116] that enable the predictions of binding affinities. All these may provide detailed insight into three-dimensional structures and interactions, which may in turn provide useful information for building data-driven methods, such as the HMMs [110]. The availability of several crystal structures can be used as templates for comparative modeling. The minimum potential energy [117, 118] may be adopted to determine the final conformation of HLA-peptide complexes with unknown crystal structures. The energy landscape with a large number of local optima makes the blend between stochastic and deterministic procedures, termed memetic algorithm (MA), an appropriate choice. As single evaluation of the objective and the constraint functions given a specific conformation of the HLA-peptide complex is computationally expensive, the structure-based predictions are not currently suitable for a large-scale screening of the potential HLA binders. It has thus become a major issue how to relax the computational budget requirements of the methods in this category. One alternative is through the effective design of efficient MA as proposed in Chapter 4 for the minimization of the steric energy of the complexes subject to some steric constraints.

6.2 Molecular Mechanics

For the purpose of applying the proposed CORE-MA to the molecular mechanics problem, the concept of steric energy will first be discussed in the following. This will be followed by formulating the problem in the form of constrained minimization. Finally, the result will be presented and discussed to showcase the efficacy of the proposed CORE-MA as compared to the simple MA.

6.2.1 Steric Energy

Steric energy is the potential energy of a molecule due to the way its atoms are arranged in space. From the point of view of classical physics, this energy gives rise to the dynamic—accelerated and decelerated—movements of the atoms. In fact, the derivatives of the energy is the force, which is related to the acceleration through the Newton’s second law of motion.

The total energy of a molecule consisting of N atoms can be expressed as the sum of all interactions from two-body to the N -body interaction [119]. These account for $2^N - N - 1$ possible interactions. Because of this exponential complexity, many molecular mechanics force fields, e.g. AMBER [101, 120, 121, 122], CHARMM [123], ENCAD [124], and OPLS [125], consider only the two-body interactions, which account for a maximum of $\frac{N^2-N}{2}$ pairwise bonded and non-bonded interactions. Generally, the two-body interactions in a molecule can be described using the following five components of the steric energy. The first three model the bonded interactions while the last two the non-bonded interactions.

6.2.1.1 Bond Stretching

The energy of a covalent bond is lowest at some particular natural length, also known as the reference length ℓ_0 . When the bond is compressed, the electron clouds of the two atoms that form the bond will gradually overlap, causing a rapid increase in the energy. When the bond is stretched beyond equilibrium, the energy will also increase until the bond eventually dissociates.

Figure 6.1: Bond Stretching

Approximating the energy of a covalent bond at some small deviations from the reference length ℓ_0 using the Taylor expansion up to the second-order term, the energy due to bond stretching is thus given by

$$E_{stretch} = \frac{1}{2}k_\ell(\ell - \ell_0)^2 \quad (6.1)$$

where

$$\ell = |\mathbf{r}_a - \mathbf{r}_b| \quad (6.2)$$

in which $\mathbf{r}_a$ and $\mathbf{r}_b$ are the Cartesian coordinates of atom a and b, respectively, in Figure 6.1 and k_ℓ is some stretching coefficient that depends on the type of atom a and b.

6.2.1.2 Bond Bending

Bond angle is formed between two consecutive covalent bonds, involving three atoms as depicted in Figure 6.2. The energy of the molecule is lowest at some particular natural angle, also referred to as the reference angle θ_0 . Intuitively, the distance between atom a and atom c becomes smaller as the bond angle θ decreases, and vice versa. Nevertheless, it should be noted that θ only assumes the value from 0 to 180°.

Figure 6.2: Bond Bending

Similar to the case of bond stretching, the energy at some small deviations from the reference angle θ_0 can be approximated using the Taylor expansion up to the second-order term, giving

$$E_{bend} = \frac{1}{2}k_{\theta}(\theta - \theta_0)^2 \quad (6.3)$$

where

$$\theta = \arccos \left(\frac{\mathbf{r}_{ba} \cdot \mathbf{r}_{bc}}{|\mathbf{r}_{ba}| |\mathbf{r}_{bc}|} \right) \quad (6.4)$$

in which $\mathbf{r}_{ba} = \mathbf{r}_a - \mathbf{r}_b$ and $\mathbf{r}_{bc} = \mathbf{r}_c - \mathbf{r}_b$, and k_{θ} is some bending coefficient that depends on the type of atom a, b, and c.

6.2.1.3 Bond Rotation

Bond rotations are arguably the most important intramolecular interaction in a force field [126]. Interactions due to bond rotations are different from those due to bond stretching and bond angle bending. Changes in torsion angles (the angles describing the bond rotations) can be large as rotational barriers are relatively low. Thus, it would be inappropriate to use the Taylor expansion to approximate the energy due to bond rotations.

Figure 6.3: Bond Rotation

It is most common to model the energy using the Fourier series as given by

$$E_{rotate} = \sum_i k_{\varphi}^i \cos(i\varphi) \quad (6.5)$$

where k_{φ}^i is the Fourier coefficients related to the torsional barriers while the calculation of the torsion angle φ can be found in [127, 128]. For simple illustration, the torsion angle φ is the angle between the plane formed by atom a, b, and c and the plane formed by atom b, c, and d in Figure 6.3.

6.2.1.4 Electrostatic Interactions

Electrostatic interactions are resulted from unequal distribution of charges in a molecule. This uneven distribution can be modeled by placing point charges at each of the atomic sites. The energy due to this type of interactions is generally modeled using Coulomb potential as given by

$$E_{es} = 332 \frac{q_i q_j}{|\mathbf{r}_{ij}|} \quad (6.6)$$

where q_i and q_j are atomic partial charge of atom i and atom j , respectively, and $|\mathbf{r}_{ij}|$ is the distance between the two atoms. It should be noted, however, that atom i and j commonly must not be connected by three or fewer covalent bonds.

6.2.1.5 van der Waals Interactions

van der Waals interactions occur between two non-bonded atoms and include dispersion, repulsion, and induction among other interactions [129]. Dispersion is due to correlations between electrons in different atoms such that the energy is lowered and attraction occurs. Repulsion happens when the electron clouds of two atoms overlap. Induction arises because of distortion of the distribution of the charge of an atom in the field of its neighbors.

For a reasonable model of van der Waals interactions, it is imperative to observe the behavior as the distance between two atoms changes. At small distances, atomic repulsion leads to large positive energy that tends toward infinity as the distance goes to zero. At large distances, the dispersion interaction gives small negative energy that tends toward zero as the distance goes to infinity.

One of the most common functional forms that encompasses this behavior is Lennard-Jones (LJ) potential [85], which has the form

$$\frac{A_{ij}}{|\mathbf{r}_{ij}|^{12}} - \frac{B_{ij}}{|\mathbf{r}_{ij}|^6} \quad (6.7)$$

While the $|\mathbf{r}_{ij}|^{-6}$ dependence of the dispersion term arises from averaging over all orientations of the multi-pole and the dipole-dipole interactions, there is no physical arguments for choosing the repulsive term to vary with $|\mathbf{r}_{ij}|^{-12}$ except computational simplicity. The repulsive potential is actually dominated by the overlap between electron clouds on different atoms. As the electron density falls off nearly exponentially with distance from the nuclei, this repulsive interaction should have an exponential dependence with distance [85]. Hence, the more rigorous choice for the van der Waals interactions is the Hill or Buckingham potential [130] as given by

$$E_{vdw} = -\frac{A_{ij}}{|\mathbf{r}_{ij}|^6} + B_{ij}e^{-C_{ij}|\mathbf{r}_{ij}|} \quad (6.8)$$

where $|\mathbf{r}_{ij}|$ is the distance between the two non-bonded atoms connected with more than three covalent bonds.

6.2.1.6 Parameterizations

For the simulation of biomolecules, the AMBER force fields [101] are commonly used to quantify the coefficients of each of the terms discussed above taking into account the various possible atom pairs. Developed for protein simulations, the AMBER force fields will be used herein to model the energy terms which serve as the objective function of the biomolecule of interest, the polyaniline.

6.2.2 Energy Minimization

Recognition of the conformations of molecules is important as lots of reactions especially in biochemistry [131] obey the lock-and-key mechanism. Knowing the molecular conformations of the constituents of some reaction, hence, enables us to predict whether or not the reaction will take place. This is beneficial in the field of drug designs where there are needs to predict the behaviors of the drugs prior to clinical trials. In the following, the formulation of the potential energy minimization will be presented. It will be followed by the results of applying the CORE-MA to minimize the potential energy of the polyalanine (polypeptide made of alanine).

6.2.2.1 Design Variables

In order to perform minimization of the energy of a molecule, it is imperative to describe the conformation of the molecule in a way that this description can be somehow varied, hence become the independent design variables. The most intuitive representation would be the Cartesian coordinates of the nucleus of each atom as depicted in Figure 6.4 where $\mathbf{r}_x = (x_x, y_x, z_x)$. However, it is clear that varying this representation using GA, for example, may not always result in a meaningful conformation of the molecule. While the mutation operator may deviate the atomic positions by only a small amount, the crossover operator may produce a conformation that makes no sense at all. Furthermore, it is easily appreciated that the number of independent variables will be three times the number of atoms when using this representation.

Figure 6.4: Cartesian Coordinates

Another representation would be the internal coordinates within the molecule itself. The same sample molecule as that in Figure 6.4 can be described using bond lengths, bond angles, and torsion angles as depicted in Figure 6.5. While fifteen real values are required to describe the molecule using Cartesian coordinates, the same molecule can be represented with only nine real values using the internal coordinates. Further reduction can also be easily achieved should the bond lengths or the bond angles be treated as constant. Furthermore, it is quite easy to limit the range of each independent variable. For example, a bond length can be restricted between a value near to zero and some amount larger than its reference length. For bond and torsion angles, modulus operation can be employed to restrict their range to $[0, 360^\circ)$. To measure the distance between any two non-bonded atoms, it is true that the relative Cartesian coordinates are required. The conversion, however, can easily be performed systematically using the technique described in [132, 133].

Figure 6.5: Internal Coordinates

For large molecules, e.g. proteins, the bond lengths and the bond angles are relatively fixed [134]. As such, the only degrees of freedom in large molecules are the torsion angles. As large molecules, proteins are constructed from linear chains of amino acids [131, 135], which share a common backbone as shown in Figure 6.6 where **R** signifies the residual side chain of the amino acid that differs among different amino acids. Two consecutive amino acids are connected to one another through a peptide bond as shown in Figure 6.7. With relatively fixed bond lengths and bond angles, the three-dimensional structure of a protein is thus primarily determined by the three backbone torsion angles, namely ϕ , ψ , and ω , as well as the side-chain torsion angles χ_i where meaningful index i varies from one amino acid to another. The backbone torsion angles, however, were observed to have assumed only certain real values in the range of $[0, 360^\circ)$ [136]. The torsion angle ϕ and ψ are free to rotate subject to steric constraints due to the side chains of the amino acids. On the contrary, the torsion angle ω is not actually free to rotate with a value very close to 180° , forming the amide planes (the shaded rectangles in Figure 6.7).

Figure 6.6: Amino Acid

Figure 6.7: Peptide Bond

For alanine amino acid with a methyl group ($-\text{CH}_3$) residue, there is only a single side-chain torsion angle χ_1 . It governs the relative position of the group with respect to the remaining atoms in the polypeptide. For the optimization of N -alanine in which N represents the number of alanine's that compose the polypeptide, therefore, the design variables are the $3N$ torsion angles φ , ψ , and χ_1 in each alanine.

6.2.2.2 Objective Function

With the simplification of the potential energy terms noting that the bond lengths and the bond angles of large proteins are relatively fixed [134], the total potential energy function is thus given by

$$E = \sum E_{rotate} + \sum E_{es} + \sum E_{vdw} \quad (6.9)$$

where the sum is carried out over all possible torsion angles in the case of potential energy due to bond rotations and over all non-bonded pairs of atoms connected by more than three covalent bonds in the case of the electrostatic and van der Waals energy. This energy function will become the objective function of the optimization problem. The coefficients related to each component of the energy function are taken from the AMBER force fields [101] as previously mentioned.

6.2.2.3 Constraint Functions

As described earlier, the minimization of the total potential energy is subjected to some steric constraints. These are simply the restriction that no two atoms may overlap each other. Treating each atom like a hard sphere, the constraints can generally be written mathematically as

$$|\mathbf{r}_{ij}| \geq r_i + r_j \quad (6.10)$$

or equivalently

$$r_i + r_j - |\mathbf{r}_{ij}| \leq 0 \quad (6.11)$$

where $|\mathbf{r}_{ij}|$ is the distance between the nuclei of atom i and atom j while r_i and r_j are the van der Waals radii [137] of atom i and atom j , respectively. As the bond lengths and bond angles has been fixed and the only degrees of freedom are the torsion angles, it should be intuitive that these constraints need to be checked only against any pairs of atoms connected by three or more covalent bonds.

With these constraints, the CORE-MA proposed earlier in Chapter 4 can then be applied to solve the molecular mechanics problem with higher efficiency. It is generally known that the distance between two atoms are always larger than the sum of the van der Waals radii of the two atoms. In other words, the strict inequality for every possible i and j is always observed in the constraint function above. This suggests that the global optimum of the energy minimization problem must be located on the trough of a feasible basin in the interior of the problem's feasible space.

6.2.2.4 Empirical Results

In testing the efficacy of the CORE-MA in solving the molecular mechanics problem with the objective and constraint functions as formulated above, the same algorithmic settings as described in Section 5.1 was adhered to. In the course of the experimentation, a population of 10 individuals was evolved with a neighborhood size of twice and a local search budget of ten times the problem dimensionality until a total of 40,000 FFEs had been exhausted to minimize the potential energy of a polyalanine with length 10 ($N = 10$).

From the experiments, it is found that the CLS-powered MA and the original FSM-driven MA had performed more efficiently than the simple MA whereas the ecumenical FSM-driven MA had just performed as efficiently as the simple MA. The comparable performance of the ecumenical FSM-driven MA with respect to the simple MA is mainly attributed to the large ratio of the feasible to the entire solution space of the problem and the fact that the minimum energy conformation of the polyalanine lies in the interior of the feasible space of its energy landscape. As can be seen from Figure 6.8 where the minimum energy trace over the course of the optimization is portrayed, the original FSM-driven MA managed to improve the minimum energy reached upon exhaustion of the search budget. This must be due to the eventual success in the elimination of the unnecessary local refinement when most of the global optimum estimates had converged towards the interior of the feasible space. The occasionally infeasible individuals generated by the global search method in the CORE-MA would easily be deemed insignificant for local refinement under the original FSM-driven MA. Under the ecumenical FSM-driven MA, however, these individuals would likely pass the significance assessment, thereby resembling the performance of the simple MA. Quite notably is the performance of the CLS-powered MA. Designed particularly for problems with readily available first-order derivatives, the CLS assessment strategy has proven its ability to correctly identify the significance of the local refinements. As the result, the necessary ones had been maintained and executed while the unnecessary ones eliminated. This successfully brings about a great improvement in the minimum energy conformation achieved.

Figure 6.8: Minimum Energy Trace of Polyalanine

From Figure 6.8, it is also observed that the original FSM-driven MA had been outperformed by the simple MA during the early generations of the search. When the FSM assessment strategy started to take effect, some local refinements may have been incorrectly skipped because of the data dependent nature of the model constructed in the FSM. This, however, helps maintain the diversity in the population which translates into better attainment in the later generations of the search. This further explains why the FSM may not be suitable for easy problems where the global optimum can be identified in the first few generations of the search. The simple MA, on the other hands, is clearly not very efficient in handling complex problems that require sufficiently long evolution to solve.

In conclusion, it has been shown in this section that the CORE-MA would be able to identify the minimum energy conformation of biomolecules faster than the simple MA. As discussed earlier, these conformations could be involved in the molecular docking as will be discussed in the next section.

6.3 Molecular Docking

Molecular docking is a computational process in the attempt to find the binding between a macromolecule (the receptor) and a small molecule (the ligand). It is of great value to drug design [138] as it can be used in predicting binding conformations and affinities between drug molecules and their target proteins, leading to the understanding of the biological mechanism behind those bindings. Generally, docking is an optimization problem of finding the binding conformation with global lowest energy, the landscape of which is approximated by a scoring function. Flexibility in the ligand—and further in the receptor—makes the problem more sophisticated by increasing the number of degrees of freedom in the molecular system model [138, 139]. Since the 1980s, various programs and software have been developed in order to perform molecular binding, such as DOCK [138], AutoDock [140], GOLD [141], FlexX [142], and ICM [143] with different scoring functions proposed. However, after decades of development, docking is still a time-consuming task even with the most powerful computing resources to-date. Recently, AutoDock Vina [102] (referred to as Vina afterwards) was released by the same group who invented the earlier versions of AutoDock, which is one of the most popular docking software. Vina uses an empirical scoring function to evaluate the binding affinity between the molecules, and the iterated local search global optimizer for global optimization. This combination is reported to be successful to achieve approximately two orders of magnitude improvement in speed, and simultaneously, a significantly better accuracy of the binding mode prediction compared to AutoDock 4 [102].

6.3.1 Vina

Vina, the nickname for AutoDock Vina, is fundamentally a form of Memetic Algorithm (MA) [71] that interleaves global and local searches by means of stochastic and deterministic optimization algorithms, respectively. It is indeed mentioned in [102] that Vina employs the iterated local search global optimizer. Vina uses a Monte Carlo method to perform the global search. To perform the local search, Vina employs the commonly used algorithm of Broyden-Fletcher-Goldfarb-Shanno (BFGS) for nonlinear unconstrained optimization [144]. In order to make progress towards better decision variables, BFGS-being essentially a Newton method-requires the first- and second-order derivatives of the objective function-which in this case is the scoring function-with respect to all design variables (*i.e.* the translation and rotation vectors between the receptor and the ligand as well as all the rotatable bonds). The second-order derivatives (also known as the Hessian), however, are approximated in BFGS as they are often very expensive to compute and not readily available. Hence, employing BFGS involves calculating the gradients of the target function, finding a direction along which the optimization progresses, and updating the approximation of the Hessian. Bearing similarities to the SQP in the absence of any constraint, accelerating Vina was in fact the first step towards the proposal of the CLS-powered MA. With no constraints, it would not be possible to use the FSM-driven MAs. The derivatives, which are calculated alongside the calculation of the scoring function in order to allow the BFGS algorithm to work, could however be used in the way that will be described next.

6.3.2 QuickVina

In Vina, as with any other MA instances, the most time-consuming component is intuitively the local search. With the first-order derivatives with respect to all design variables readily available in Vina, it is possible to devise a heuristics that makes use of this information to adaptively adjust the local search frequency of Vina. Considering the fact that local search is to find a local minimum, which must be a stationary point (in which the derivative is zero) in continuous functions according to Fermat's theorem, it is possible to prevent some of the point from performing local search. For illustrative purpose, an example of a continuous function in one-dimensional space is shown in Figure 6.9. Point A, B and D are with negative derivatives while C is with a positive derivative. As the derivatives of the scoring function in Vina are continuous, there must be a point with zero derivatives between the points of A and C according to the Intermediate Value Theorem. Based on this observation, an improvement in the global minimization in Vina can theoretically be achieved by heuristically preventing some of the point from performing local search using the information of the points which has been seen in the previous searching steps. The basic idea of the proposed heuristic is to test an intermediate point ready for local search against a database which stored the information of other points generated and evaluated earlier. Should it be possible to find any neighbor of the point with totally opposite signs of derivatives in the database, this point passes the test, local search is performed and the result of the local search is sent to the database to be stored for possible future usage.

Figure 6.9: Function of A Single Design Variable with Derivative at Different Points

To gauge the efficacy empirically, the proposed heuristics above have been implemented as QuickVina—or QVina for short. The amount of modifications have been kept to the minimum possible so as to ensure the fairness when comparing the performance of QVina to that of the original Vina. The implemented QVina is rudimentarily an embryonic form of the CLS-powered MA in that stratification of the search space landscape has powered the assessment of whether a trial solution is significant to undergo local refinement. Lacking of any constraints, however, only the objective landscape needs to be stratified. In handling unconstrained optimization problems, it is very intuitive that any optimum would have its derivatives vanished. Going in two opposite directions towards the regions of non-vanishing derivatives, one would find trial solutions of opposite derivative signs. As the first attempt to utilize the derivatives information, it is therefore reasonable to limit local refinement of a trial solution only when a neighbor with completely opposite derivative signs has been found.

6.3.3 PDBbind Screening

Following the implementation, the QVina was tested on the core set of PDBbind [145] version 2010. The PDBbind dataset is a well-established dataset of protein-ligand docking. The data of PDBbind are selected from the Protein Data Bank (PDB), and several hierarchical subsets are defined for the convenience of different applications. Firstly, 6772 complexes including protein-ligand, nucleic-acid-ligand, protein-nucleic-acid, and protein-protein complexes with experimental measured binding affinity are identified and selected from the entire PDB to form the general set of PDBbind. From this general set, 2061 entries of protein-ligand complexes only are selected based on several strict requirements on the data quality, among them are the minimum coordinate resolution of 2.5 Å and the absent of uncommon element types. These 2061 entries are compiled as the refined set. Finally, a core set is compiled by clustering the refined set based on the protein sequence, forming 77 clusters and then selecting the highest, the lowest and the medium binding affinity from each cluster to ensure the diversity in both the structure and binding affinity. The core set of PDBbind version 2010—comprising 231 protein-ligand complexes—was used for the test of QVina.

During the efficacy test, all the default built-in parameters of the original Vina were observed with 8 CPUs and exhaustiveness level of 128. In QVina, furthermore, the neighborhood was defined with a size of $2N$ where N being the number of independent variables, that is the number of rotatable bond in the ligand.

The results of this performance test which consist of the running time, the docking result score and the docking result conformation of both QVina and Vina were collected. First, the speed-up against the number of rotatable bonds is plotted in Figure 6.10. Then, the minimum score reached by the QVina and the original Vina are shown in Figure 6.11. Finally, the RMSE between the native structure in PDB and the docking result structure of both QVina and Vina is shown in Figure 6.12.

Figure 6.10: Speed-up against Number of Rotatable Bonds

Figure 6.11: Scores of Docking Results Using Vina and QuickVina

Figure 6.12: RMSE of Docking Results with Respect to Native Conformations in PDB

From Figure 6.10, it can be clearly seen that the QVina has performed significantly faster than the original Vina. An average speed-up of 8.35 times over the original Vina has been achieved when tested on the test data set of PDB-bind core set. The QVina achieved the maximum speed-up of 25.24 times when dealing with the protein–ligand complex 1ZEA, reducing the running time of docking from 6640 seconds-roughly 1 hour and 50 minutes-to merely 263 seconds, which is 4 minutes 23 seconds, while maintaining the minimized score of docking the same. Such a tremendous speed-up will ultimately be useful for high-throughput screening of various ligands or small molecules, such as potential drug candidates, against a given receptor. This would eventually allow wet-lab experiments to be conducted only on the highly potential candidates, significantly reducing the cost and time needed for drug developments. Furthermore, it is also observed that the speed-up increases as the number of rotatable bonds increases.

With greater flexibility—that is, more rotatable bonds—it is intuitive that the optimization process would be much more complicated. This implies larger search space to be explored by the global search method. Additionally, there would be more derivatives to evaluate as there are more design variables. There would also be more steps required by the local search method to converge to an optimum point. All these add to the time complexity of the global optimization procedure, increasing significantly the running time of the original Vina. Using the QVina, however, the proposed heuristics easily eliminates the unnecessary local searches. Consequently, computational time and resources which would otherwise be used to perform the unnecessary local searches in the original Vina could be eradicated in the QVina. This explains the tendency of having even more speed-up for higher number of rotatable bonds. This also further supports the observation that the CORE-MA is more suitable for complex problems.

It should also be noted that the docking accuracy is not compromised while the running time is significantly reduced. Since the docking score serves as the objective function in the global optimization procedure of both QVina and the original Vina, we can use it as the estimation of the docking performance. From Figure 6.11, it can clearly be seen that the quality of both docking software are identical. In fact, these two sets of scores have a correlation as high as 0.997—almost statistically identical—suggesting that the docking performance of QVina is never compromised. In other words, even with some local searches omitted, our QVina is able to find comparable global optimum conformations as those of the original Vina.

To further understand the results, final docked conformations were also collected for structural analysis. In Figure 6.12, the RMSE between the native structures in PDB and the docking results are shown. As a stochastic method, it is possible that the docking programs give different structures although the score is stable. This is observed for example in the complexes 1BMA, 1BCU and 1PB8 in Figure 6.13 (A, B, and C, respectively). Their docking results are scored identically by Vina and QVina, hence both Vina and QVina regarded the obtained docked conformations equally optimal in the view of optimization. However, QVina and Vina gave very close docking results only on the complex 1BCU, which is also very close to the native structure shown in PDB. Meanwhile, QVina yielded a result closer to the native structure in the PDB on the complex 4ER2 than Vina did, while the opposite is true for the complex 1BMA. This may suggest either the existence of the docking isomers or that there is a need for a better scoring function that best describe the interaction between the receptor and the ligand. In conclusion, the QVina employing the CLS-like heuristics has performed much more efficiently without compromising the accuracy.

Figure 6.13: Comparison of Vina (Blue) and QVina (Green) Docking Results against Native Conformations in PDB (Red)

Chapter 7

Conclusion and Future Works

7.1 Conclusion

In its entirety, this thesis has been dedicated to addressing the challenges listed in Section 1.2. In the following, contributions made by the different works from the various sections of the thesis in answering the questions are summarized.

1. CORE-MA is the first *optinformatics* (learning-from-archive paradigm) in the context of constrained real-valued optimization where the structure of the problem search space is learned as it evolves so as to enable intelligent and effective selection of individuals to undergo local refinement.
2. In realizing the adaptive selection of the local refinements, a classification problem is for the first time formulated within an optimization framework in FSM. As the first CORE-MA instantiation, the FSM solves the arising classification problem using the SVM and subsequently uses its prediction to decide if an individual is worth refining. This presents a novel concept of employing the machine learning techniques for the better evolutions of the evolutionary computing procedures.

3. The use of SVM in FSM is substituted with the simple nearest-neighbor-based dichotomizer in CLS. As the second CORE-MA instantiation, CLS not only allows adaptive selection of local refinements but further reduces the computational overhead incurred by FSM-driven MAs over the simple MA by first stratifying the problem landscape using the objective and/or the constraint gradients. Direct comparison between some individual and its respective relevant nearest neighbors is then possible. The availability of gradient information can thus be utilized to incur less overhead.
4. Through Chapters 5 and 6, the CORE-MA has been subjected to various benchmark and practical problems. From the experiments, it was noticed that all instances of the CORE-MA have performed more efficiently than the simple MA as well as the other adaptive MAs considered. Exceptions were observed, however, when handling simple problems that are solvable with the simple MA over just a couple of generations. As there is smaller room for improvements, instances of the CORE-MA may have performed just as efficiently as the simple MA. Upon comparing various instances of the CORE-MA, CLS-powered MA is observed to have delivered the most efficient and effective performances on the hard problems whilst incurring the least overhead. The CLS-powered MA, however, requires the gradient information of all the objective and the constraint functions to be readily available. Even so, CLS-powered MA may find its application on the not-so-expensive problems, such as low-fidelity of the biological and chemical simulations for screening purpose, interesting and potential.

7.2 Future Works

The achievements as concluded in the previous section have opened up the way to the plethora of future research works both in the methodological aspect and from the application perspective. Herein, some of the possible future works will be discussed briefly.

7.2.1 Methods

In its present form, the CORE-MA implements the lookup table as linear array of trial solutions, learns the problem topology using as many nearest neighbors as twice the problem dimensionality, carries out the significance assessments at population level, and is intended to efficiently solve real-valued single-objective optimization problems. As such, there are plenty of rooms for improvements to be addressed in the near future as will be discussed briefly in the following.

7.2.1.1 Efficient Lookup Table

Fundamental to the learning capability of CORE-MA is the lookup table which holds the optimization archive. Complexity of extracting the nearest neighbors from the lookup table for the purpose of a significance assessment is dependant on the size of the lookup table. Lower cost of computing power in the future is making many more expensive evaluations than just few tens of thousands more affordable. Determining the relevant subset of the lookup table is hence crucial to keep an acceptable overhead due to extraction of the nearest neighbors—one possible direction for future works.

7.2.1.2 Optimal Neighborhood Size

All the while, the CORE-MA has inferred the problem topology using as many nearest neighbors as twice the problem dimensionality. Such neighborhood size has been used as a norm for it is intuitive to process the information uncovered by perturbing the values of every dimension slightly higher and slightly lower if one wishes to reveal a topology. In CORE-MA, choice of the neighborhood size borrows no basis from the field of information theory. The neighborhood size of twice the problem dimensionality hence provides no optimal warranty. Finding the most optimal neighborhood size for different circumstances with some basis derived from the information theory may in fact improve the learning accuracy of the CORE-MA and make its optimization process even more efficient—other possible direction for future works.

7.2.1.3 Intra-individual Significance Assessment

In the current CORE-MA, the significance assessment of a trial solution occurs at the population level, *i.e.* the significance is determined among individuals in the population. When an individual is deemed significant, it shall then undergo local refinement until convergence is reached or budget is exhausted, whichever is earlier. Local refinement may in fact lead the individual to some basin where previous exploitation has been performed. Incorporation of the intra-individual CORE assessment can potentially terminate such refinement prematurely. This may sound helpful but intra-individual significance assessment may easily incur huge overhead—making this potential direction for future works.

7.2.1.4 Combinatorics CORE-MA

The present forms of CORE-MA are intended for the real-valued optimization. As such, CORE-MA currently tackles neither the binary variables in particular nor the integer variables in general. It is well-known, nonetheless, that discrete optimization may be transformed into the continuous one through introduction of new equality constraints. Alternatively, some characteristics of the optimum of combinatorial optimization may be studied, derived, and employed for direct mining as with the case of the current CORE-MA. While a function of discrete variables is generally non-differentiable, its functional values may be treated in a similar fashion as the finite-difference derivatives to obtain some gradient-like quantities. Further research is clearly necessary, making this a potential area of research for future works.

7.2.1.5 Multi-objective CORE-MA

Beside being designed for the real-valued optimization, the current instances of CORE-MA are also intended for the single-objective optimization. In handling multiple objectives, all but one objective can be converted into some inequality constraints with evolving thresholds. The current CORE-MA can then be used to solve the arising optimization problem efficiently. By rotating the objectives which are treated as the constraints and updating their thresholds periodically, the current CORE-MA can therefore be employed to obtain the pareto front of the multi-objective problems. Although this may seem straightforward, further research is clearly needed, making it another potential area for future works.

7.2.2 Applications

The application of CORE-MA on the practical problems in Chapter 6 confirms the efficacy of the proposed strategies in solving the computationally-expensive problems. The expensive fitness evaluation of such problems may be attributed to either some composite formula calculation or an intricate system simulation. Intuitively, applying CORE-MA on these two categories of expensive problems constitutes potential future works as discussed briefly in the following.

7.2.2.1 Composite Formula Calculation

The expensive evaluation of trial solutions to molecular mechanics and docking problems in Chapter 6 may be attributed to the composite formula calculation. The complexity arises from the number of atoms involved. Since the evaluation generally takes into account all interactions between each and every atom pair, the complexity grows quadratically with the number of the atoms. The fidelity of the interaction model in use can further complicate the calculation. The use of CORE-MA is likewise promising for the following applications.

1. Structure-based computational vaccine design [146, 147, 148] is important due to the high polymorphism of the HLAs, the huge variety of the short antigenic peptides, and the specificity of the HLA-peptide complexes that can trigger the immune responses. The use of CORE-MA in such context is truly promising in that the design may be accomplished within shorter time frame, considering the enormous amount of computation involved in the pipeline.

2. Discovery of the transition states of small chemical compounds [149, 150] is significant for the synthesis of non-naturally occurring drug substances. With the slightest modifications to the CORE-MA in that a saddle-point identifier instead of a local minimizer is employed for the purpose of local refinement, promising efficiency can be expected with the use of the CLS-powered MA. Both saddle points and minima share one commonality, *i.e.* they are characterized by vanishing gradients.

7.2.2.2 Intricate System Simulation

In addition to some composite formula calculation, the expensive evaluation of trial solutions to the computationally-expensive problems may be attributed to an intricate system simulation. Events taking place in some chronological order are often observed in simulations, making them difficult—if not impossible—to parallelize. The promising applications of CORE-MA in this context include—but not limited—to the following problems.

1. Chemical processes [151] are always studied prior to construction of some chemical plants. Different processes are simulated and optimized through alteration of control parameters that may include concentration, pressure, and volume among many others. The study allows efficient outputs when the plants are eventually in operation, but may take considerable amount of time during planning due to the expensive simulations of the processes. The CORE-MA can therefore be used to accomplish the planning within shorter time frame, thereby allowing quicker profit scooping.

2. Electrical power flow [152] is commonly addressed in designing the smart power grids which aim at minimizing power loss, thereby reducing carbon footprints by responding to different electrical load conditions. Since it is impractical to know a priori all the possible load combinations, the use of CORE-MA in this context may help reduce the delay when responding to some changes in the load conditions, therefore obtaining the optimal flow at close to real-time most of the time.
3. Vehicle routing [153] is frequently—if not always—involved in a logistical planning. The use of CORE-MA within the context can lead to a greater efficiency such that almost real-time updates as the response to changing traffic conditions become possible.

Appendix A

Supplementary Results

A.1 Threshold Variation

In the following, Tables A.1–A.5 tabulate the results of varying the significance thresholds observed by the CLS-powered MA upon encounter with the different individual and neighborhood combinations: uniform neighborhood of a feasible individual, uniform neighborhood of an infeasible individual, and the composite neighborhood of an individual of either feasibility class. Uniform neighborhood of an infeasible individual may indicate the event of insufficient diversity, which employ yet another threshold. The experiment was performed to ascertain that the success of the CLS-powered MA in handling the benchmark problems more efficiently can be attributed to not only the effectiveness of the stratification of the problem landscape but also the proper choice of the significance thresholds. In summary, it is observed that the CLS-powered MA shall yield better results on some problems but produce worse results on the other problems when using custom significance thresholds. This indicates that the proposed thresholds are indeed the best choice.

Table A.1: FFE Count Statistics of CLS of Various Significance Thresholds on Uniform Neighborhood of Feasible Individual

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g05	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	40	40	40	40	40
	Q1	104	104	104	104	104
	Q2	127	127	127	127	127
	Q3	150	150	150	150	150
	MAX	189	189	189	189	189
	AVG	125	125	125	125	125
g21	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	174	174	174	174	174
	Q1	212	212	212	212	212
	Q2	255	255	255	255	255
	Q3	322	322	322	322	322
	MAX	465	448	465	465	465
	AVG	274	272	274	274	274
g23	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	579	579	579	579	579
	Q1	625	625	625	625	625
	Q2	710	710	710	710	710
	Q3	757	757	757	757	757
	MAX	888	888	888	888	888
	AVG	702	702	702	702	702
g22	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	87	87	87	87	87
	Q1	179	179	179	179	179
	Q2	337	337	337	337	337
	Q3	578	578	578	578	578
	MAX	1,083	1,083	1,083	1,083	1,083
	AVG	391	391	391	391	391

continued on the next page ...

Table A.1 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g11	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	27	27	27	27	27
	Q1	38	38	38	38	38
	Q2	50	50	50	50	50
	Q3	60	60	60	60	60
	MAX	78	78	78	78	78
g15	AVG	50	50	50	50	50
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	24	24	24	24	24
	Q1	40	40	40	40	40
	Q2	40	40	40	40	40
	Q3	40	40	40	40	40
g13	MAX	149	149	149	149	149
	AVG	48	48	48	48	48
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	220	220	220	220	220
	Q1	293	293	293	293	293
	Q2	351	351	351	351	351
g17	Q3	380	380	380	380	380
	MAX	437	437	437	437	437
	AVG	338	338	338	338	338
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	35	35	35	35	35
	Q1	105	105	105	105	105
g17	Q2	119	119	119	119	119
	Q3	149	149	149	149	149
	MAX	284	284	284	284	284
	AVG	133	133	133	133	133

continued on the next page ...

Table A.1 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g03	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	102	102	102	102	102
	Q1	110	110	110	110	110
	Q2	124	124	124	124	124
	Q3	236	223	236	236	249
	MAX	615	615	615	615	638
	AVG	196	194	198	198	202
g14	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	220	483	220	220	220
	Q1	975	1,466	975	975	2,048
	Q2	1,730	2,790	1,780	1,780	3,084
	Q3	3,927	5,153	4,127	3,915	4,067
	MAX	7,646	10,787	7,646	7,646	9,719
	AVG	2,424	3,698	2,447	2,348	3,277
g06	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	197	197	197	197	197
	Q1	210	210	210	210	210
	Q2	210	210	210	210	210
	Q3	210	210	210	210	210
	MAX	210	210	210	210	210
	AVG	209	209	209	209	209
g08	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	179	179	179	179	179
	Q1	300	278	318	338	369
	Q2	345	367	359	374	401
	Q3	394	427	400	409	419
	MAX	561	865	547	595	1,324
	AVG	350	374	354	374	421

continued on the next page ...

Table A.1 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g24	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	75	75	75	75	75
	Q1	93	93	93	93	93
	Q2	104	104	104	104	104
	Q3	118	118	118	118	119
	MAX	132	132	132	132	132
	AVG	103	103	103	103	104
g12	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	55	55	55	55	55
	Q1	84	84	84	84	84
	Q2	108	108	108	108	108
	Q3	140	140	140	140	140
	MAX	170	170	170	170	170
	AVG	107	107	107	107	107
g04	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	130	130	130	130	130
	Q1	277	277	277	277	277
	Q2	324	324	324	324	324
	Q3	353	353	353	353	353
	MAX	510	510	510	510	510
	AVG	318	318	318	320	320
g16	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	67	67	67	67	67
	Q1	151	151	151	151	151
	Q2	384	384	384	384	384
	Q3	677	677	677	677	677
	MAX	980	980	980	980	980
	AVG	423	423	423	423	423

continued on the next page ...

Table A.1 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g09	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	704	704	704	704	704
	Q1	710	710	710	710	710
	Q2	710	710	710	710	710
	Q3	710	710	710	710	710
	MAX	710	710	710	710	710
g10	AVG	710	710	710	710	710
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	423	423	423	423	423
	Q1	544	544	544	544	544
	Q2	595	595	595	595	595
	Q3	673	673	673	673	673
g18	MAX	1,191	1,191	1,191	1,191	1,228
	AVG	618	618	618	618	619
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	119	119	119	119	119
	Q1	170	170	170	170	170
	Q2	223	223	223	223	223
g07	Q3	264	264	264	264	264
	MAX	371	371	371	371	371
	AVG	223	223	223	223	223
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	624	624	624	624	624
	Q1	671	671	671	671	671
g07	Q2	696	696	696	696	696
	Q3	724	724	724	724	724
	MAX	776	776	776	776	776
	AVG	698	698	698	698	698

continued on the next page ...

Table A.1 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g01	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	92	92	92	92	92
	Q1	141	141	141	141	141
	Q2	193	194	193	194	194
	Q3	229	252	232	240	251
	MAX	424	891	569	690	787
	AVG	201	222	209	214	221
g19	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	632	632	632	632	632
	Q1	807	807	807	807	807
	Q2	864	864	864	864	864
	Q3	926	926	926	926	926
	MAX	1,134	1,134	1,134	1,134	1,134
	AVG	871	871	871	871	871
g02	S.R.	63.33%	66.67%	50.00%	56.67%	60.00%
	MIN	11,140	15,936	16,017	11,868	11,868
	Q1	17,968	24,223	24,285	24,070	23,650
	Q2	26,945	28,991	>40,000	39,070	33,175
	Q3	>40,000	>40,000	>40,000	>40,000	>40,000
	MAX	>40,000	>40,000	>40,000	>40,000	>40,000
	AVG	>21,991	>25,389	>25,223	>26,565	>24,670

Table A.2: FFE Count Statistics of CLS of Various Significance Thresholds on Composite Neighborhood of Feasible Individual

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g05	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	40	40	40	40	40
	Q1	104	104	104	104	104
	Q2	127	127	127	127	127
	Q3	150	150	150	150	150
	MAX	189	189	189	189	189
	AVG	125	125	125	125	125
g21	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	174	174	174	174	174
	Q1	212	212	212	212	212
	Q2	255	255	255	255	255
	Q3	322	322	322	322	322
	MAX	465	465	465	465	465
	AVG	274	274	274	274	274
g23	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	579	579	579	579	579
	Q1	625	625	625	625	625
	Q2	710	710	710	710	710
	Q3	757	757	757	757	757
	MAX	888	888	888	888	888
	AVG	702	702	702	702	702
g22	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	87	87	87	87	87
	Q1	179	179	179	179	179
	Q2	337	337	337	337	337
	Q3	578	578	578	578	578
	MAX	1,083	1,083	1,083	1,083	1,083
	AVG	391	391	391	391	391

continued on the next page ...

Table A.2 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g11	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	27	27	27	27	27
	Q1	38	38	38	38	38
	Q2	50	50	50	50	50
	Q3	60	60	60	60	60
	MAX	78	78	78	78	78
g15	AVG	50	50	50	50	50
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	24	24	24	24	24
	Q1	40	40	40	40	40
	Q2	40	40	40	40	40
	Q3	40	40	40	40	40
g13	MAX	149	149	149	149	149
	AVG	48	48	48	48	48
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	220	220	220	220	220
	Q1	293	293	293	293	293
	Q2	351	351	351	351	351
g17	Q3	380	380	380	380	380
	MAX	437	437	437	437	437
	AVG	338	338	338	338	338
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	35	35	35	35	35
	Q1	105	105	105	105	105
g17	Q2	119	119	119	119	119
	Q3	149	149	149	149	149
	MAX	284	284	284	284	284
	AVG	133	133	133	133	133

continued on the next page ...

Table A.2 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g03	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	102	102	102	102	102
	Q1	110	110	110	110	110
	Q2	124	124	124	124	124
	Q3	236	249	236	236	236
	MAX	615	1,078	615	615	615
g14	AVG	196	228	196	196	196
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	220	220	220	220	220
	Q1	975	975	975	975	975
	Q2	1,730	1,919	1,730	1,730	1,730
	Q3	3,927	5,128	3,927	3,927	3,927
g06	MAX	7,646	14,929	7,646	7,646	7,646
	AVG	2,424	3,096	2,424	2,424	2,424
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	197	197	197	197	197
	Q1	210	210	210	210	210
	Q2	210	210	210	210	210
g08	Q3	210	210	210	210	210
	MAX	210	210	210	210	210
	AVG	209	209	209	209	209
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	179	179	179	179	179
	Q1	300	284	304	304	337
g08	Q2	345	339	345	345	364
	Q3	394	427	394	456	465
	MAX	561	561	561	852	852
	AVG	350	346	351	365	375
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	179	179	179	179	179

continued on the next page ...

Table A.2 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g24	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	75	59	75	75	94
	Q1	93	80	93	96	107
	Q2	104	92	105	105	119
	Q3	118	113	118	119	128
	MAX	132	149	132	132	139
g12	AVG	103	94	104	106	119
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	55	55	55	55	55
	Q1	84	84	84	84	84
	Q2	108	108	108	108	108
	Q3	140	140	140	140	140
g04	MAX	170	170	170	170	170
	AVG	107	107	107	107	107
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	130	115	130	180	245
	Q1	277	235	277	297	338
	Q2	324	276	324	339	366
g16	Q3	353	342	353	376	414
	MAX	510	510	510	510	510
	AVG	318	291	321	337	371
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	67	67	67	67	67
	Q1	151	156	151	151	151
g04	Q2	384	384	384	384	384
	Q3	677	677	677	677	677
	MAX	980	980	980	980	980
	AVG	423	424	423	423	423
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	67	67	67	67	67
g16	Q1	151	156	151	151	151
	Q2	384	384	384	384	384
	Q3	677	677	677	677	677
	MAX	980	980	980	980	980
	AVG	423	424	423	423	423
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%

continued on the next page ...

Table A.2 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g09	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	704	704	704	704	704
	Q1	710	710	710	710	710
	Q2	710	710	710	710	710
	Q3	710	710	710	710	710
	MAX	710	710	710	710	710
g10	AVG	710	710	710	710	710
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	423	423	423	423	423
	Q1	544	544	544	544	544
	Q2	595	595	595	595	595
	Q3	673	673	673	673	673
g18	MAX	1,191	1,191	1,191	1,191	1,191
	AVG	618	618	618	618	618
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	119	119	119	119	119
	Q1	170	170	170	170	170
	Q2	223	223	223	223	223
g07	Q3	264	264	264	264	264
	MAX	371	371	371	371	371
	AVG	223	223	223	223	223
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	624	624	624	624	624
	Q1	671	671	671	671	671
g07	Q2	696	696	696	696	696
	Q3	724	724	724	724	724
	MAX	776	776	776	776	776
	AVG	698	698	698	698	698

continued on the next page ...

Table A.2 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g01	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	92	92	92	92	92
	Q1	141	141	141	141	141
	Q2	193	195	193	193	196
	Q3	229	235	245	245	247
	MAX	424	424	761	697	1,449
	AVG	201	202	217	216	243
g19	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	632	544	671	807	934
	Q1	807	696	810	922	1,027
	Q2	864	738	888	993	1,102
	Q3	926	849	964	1,083	1,159
	MAX	1,134	1,104	1,183	1,215	1,336
	AVG	871	776	897	1,003	1,108
g02	S.R.	63.33%	66.67%	60.00%	66.67%	66.67%
	MIN	11,140	15,062	11,148	12,992	16,172
	Q1	17,968	20,000	20,586	23,052	22,614
	Q2	26,945	35,170	33,638	32,127	32,522
	Q3	>40,000	>40,000	>40,000	>40,000	>40,000
	MAX	>40,000	>40,000	>40,000	>40,000	>40,000
	AVG	>21,991	>26,374	>23,930	>25,418	>27,037

Table A.3: FFE Count Statistics of CLS of Various Significance Thresholds on Composite Neighborhood of Infeasible Individual

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g05	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	40	40	84	95	115
	Q1	104	82	137	142	142
	Q2	127	117	164	170	174
	Q3	150	137	179	194	212
	MAX	189	166	231	237	257
	AVG	125	109	157	171	179
g21	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	174	96	186	205	205
	Q1	212	143	268	299	300
	Q2	255	167	322	333	346
	Q3	322	221	364	389	396
	MAX	465	687	507	507	508
	AVG	274	194	317	339	350
g23	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	579	576	579	579	579
	Q1	625	623	633	633	633
	Q2	710	710	714	714	714
	Q3	757	757	757	757	757
	MAX	888	888	888	888	888
	AVG	702	699	705	705	705
g22	S.R.	100.00%	96.67%	100.00%	100.00%	100.00%
	MIN	87	91	92	92	92
	Q1	179	382	184	196	184
	Q2	337	509	345	355	349
	Q3	578	866	725	664	689
	MAX	1,083	>40,000	1,573	1,221	1,170
	AVG	391	>885	459	445	447

continued on the next page ...

Table A.3 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g11	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	27	18	29	29	57
	Q1	38	28	41	45	82
	Q2	50	38	53	69	93
	Q3	60	47	66	79	105
	MAX	78	115	85	95	138
g15	AVG	50	42	54	64	94
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	24	24	24	24	24
	Q1	40	40	39	39	40
	Q2	40	40	40	40	40
	Q3	40	46	40	40	69
g13	MAX	149	140	170	170	200
	AVG	48	45	45	46	52
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	220	179	220	220	220
	Q1	293	310	310	310	310
	Q2	351	348	354	354	360
g17	Q3	380	374	381	387	410
	MAX	437	546	437	437	437
	AVG	338	339	342	345	352
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	35	25	84	84	105
	Q1	105	73	147	190	202
g17	Q2	119	108	200	234	241
	Q3	149	203	259	268	288
	MAX	284	304	358	395	435
	AVG	133	128	201	229	248

continued on the next page ...

Table A.3 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g03	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	102	102	102	102	102
	Q1	110	110	110	110	110
	Q2	124	156	124	124	154
	Q3	236	249	219	219	249
	MAX	615	787	498	449	449
	AVG	196	219	177	179	190
g14	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	220	220	220	220	220
	Q1	975	975	647	1,158	1,826
	Q2	1,730	1,499	1,714	2,066	2,790
	Q3	3,927	2,459	3,053	3,938	3,685
	MAX	7,646	11,517	9,781	9,598	8,074
	AVG	2,424	2,115	2,238	2,712	2,975
g06	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	197	197	197	197	197
	Q1	210	210	210	210	210
	Q2	210	210	210	210	210
	Q3	210	210	210	210	210
	MAX	210	210	210	210	210
	AVG	209	209	209	209	209
g08	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	179	179	179	179	179
	Q1	300	300	300	300	300
	Q2	345	345	345	345	345
	Q3	394	394	394	394	394
	MAX	561	561	561	561	561
	AVG	350	350	350	350	350

continued on the next page ...

Table A.3 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g24	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	75	75	75	75	75
	Q1	93	93	93	93	93
	Q2	104	104	104	104	104
	Q3	118	118	118	118	118
	MAX	132	132	132	132	132
g12	AVG	103	103	103	103	103
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	55	54	55	55	60
	Q1	84	84	84	84	86
	Q2	108	107	108	108	109
	Q3	140	137	140	140	142
g04	MAX	170	170	170	171	171
	AVG	107	106	107	107	112
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	130	130	130	130	130
	Q1	277	277	277	277	277
	Q2	324	324	324	324	324
g16	Q3	353	353	353	353	353
	MAX	510	510	510	510	510
	AVG	318	318	318	318	318
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	67	67	67	67	67
	Q1	151	151	151	151	151
g04	Q2	384	384	384	384	384
	Q3	677	677	677	677	677
	MAX	980	980	980	980	980
	AVG	423	423	423	423	423
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	67	67	67	67	67
g16	Q1	151	151	151	151	151
	Q2	384	384	384	384	384
	Q3	677	677	677	677	677
	MAX	980	980	980	980	980
	AVG	423	423	423	423	423
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%

continued on the next page ...

Table A.3 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g09	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	704	704	704	704	704
	Q1	710	710	710	710	710
	Q2	710	710	710	710	710
	Q3	710	710	710	710	710
	MAX	710	710	710	710	710
g10	AVG	710	710	710	710	710
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	423	423	423	423	423
	Q1	544	544	544	544	544
	Q2	595	595	595	595	595
	Q3	673	673	673	673	673
g18	MAX	1,191	1,191	1,191	1,191	1,191
	AVG	618	618	618	618	618
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	119	119	119	119	119
	Q1	170	170	170	170	170
	Q2	223	223	223	223	223
g07	Q3	264	264	264	264	264
	MAX	371	371	371	371	371
	AVG	223	223	223	223	223
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	624	624	624	624	624
	Q1	671	671	671	671	671
g07	Q2	696	696	696	696	696
	Q3	724	724	724	724	724
	MAX	776	776	776	776	776
	AVG	698	698	698	698	698

continued on the next page ...

Table A.3 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Average	Below Maximum	Unbounded
g01	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	92	92	92	92	92
	Q1	141	141	141	141	141
	Q2	193	193	193	193	193
	Q3	229	229	229	229	229
	MAX	424	424	424	424	424
	AVG	201	201	201	201	201
g19	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	632	632	632	632	632
	Q1	807	807	807	807	807
	Q2	864	864	864	864	864
	Q3	926	926	926	926	926
	MAX	1,134	1,134	1,134	1,134	1,134
	AVG	871	871	871	871	871
g02	S.R.	63.33%	63.33%	63.33%	63.33%	63.33%
	MIN	11,140	11,140	11,140	11,140	11,140
	Q1	17,968	17,968	17,968	17,968	17,968
	Q2	26,945	26,945	26,945	26,945	26,945
	Q3	>40,000	>40,000	>40,000	>40,000	>40,000
	MAX	>40,000	>40,000	>40,000	>40,000	>40,000
	AVG	>21,991	>21,991	>21,991	>21,991	>21,991

Table A.4: FFE Count Statistics of CLS of Various Significance Thresholds on Uniform Neighborhood of Infeasible Individual

Problem	Measure	CLS	Unexplored	Below Minimum	Below Maximum	Unbounded
g05	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	40	21	21	74	74
	Q1	104	73	73	141	141
	Q2	127	87	92	161	172
	Q3	150	111	114	204	219
	MAX	189	158	158	240	280
	AVG	125	91	92	168	176
g21	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	174	174	174	174	174
	Q1	212	212	212	212	212
	Q2	255	255	255	255	265
	Q3	322	315	315	322	353
	MAX	465	465	465	465	465
	AVG	274	269	269	275	282
g23	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	579	579	579	579	579
	Q1	625	625	625	625	625
	Q2	710	710	710	710	710
	Q3	757	757	757	757	757
	MAX	888	888	888	888	888
	AVG	702	702	702	702	702
g22	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	87	87	87	87	87
	Q1	179	183	182	179	179
	Q2	337	381	355	355	355
	Q3	578	624	537	622	622
	MAX	1,083	1,597	937	1,154	1,699
	AVG	391	461	377	417	431

continued on the next page ...

Table A.4 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Minimum	Below Maximum	Unbounded
g11	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	27	19	27	27	46
	Q1	38	32	36	52	79
	Q2	50	38	41	63	91
	Q3	60	49	50	70	102
	MAX	78	60	70	113	139
	AVG	50	40	43	63	91
g15	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	24	24	24	24	24
	Q1	40	40	40	40	40
	Q2	40	40	40	40	40
	Q3	40	40	40	40	55
	MAX	149	175	140	170	170
	AVG	48	50	45	47	50
g13	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	220	229	220	220	220
	Q1	293	277	277	308	324
	Q2	351	322	351	351	357
	Q3	380	364	374	380	405
	MAX	437	7,092	437	437	487
	AVG	338	549	333	342	358
g17	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	35	29	30	35	35
	Q1	105	66	70	144	144
	Q2	119	92	94	216	237
	Q3	149	103	109	287	310
	MAX	284	154	164	373	411
	AVG	133	87	91	211	231

continued on the next page ...

Table A.4 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Minimum	Below Maximum	Unbounded
g03	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	102	102	102	102	102
	Q1	110	110	110	110	110
	Q2	124	124	124	156	156
	Q3	236	219	219	255	255
	MAX	615	615	615	533	804
	AVG	196	191	191	212	232
g14	S.R.	100.00%	96.67%	100.00%	100.00%	100.00%
	MIN	220	220	220	220	220
	Q1	975	1,148	1,036	1,311	1,201
	Q2	1,730	1,958	2,119	1,938	1,933
	Q3	3,927	4,308	2,821	3,194	3,194
	MAX	7,646	>40,000	5,238	6,270	9,878
	AVG	2,424	>2,846	2,143	2,294	2,409
g06	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	197	197	197	197	197
	Q1	210	210	210	210	210
	Q2	210	210	210	210	210
	Q3	210	210	210	210	210
	MAX	210	210	210	210	210
	AVG	209	209	209	209	209
g08	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	179	179	179	179	179
	Q1	300	300	300	300	300
	Q2	345	345	345	345	345
	Q3	394	394	394	394	394
	MAX	561	561	561	561	561
	AVG	350	350	350	350	350

continued on the next page ...

Table A.4 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Minimum	Below Maximum	Unbounded
g24	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	75	75	75	75	75
	Q1	93	93	93	93	93
	Q2	104	104	104	104	104
	Q3	118	118	118	118	118
	MAX	132	132	132	132	132
g12	AVG	103	103	103	103	103
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	55	53	55	55	55
	Q1	84	82	83	84	84
	Q2	108	108	108	110	117
	Q3	140	140	140	140	147
g04	MAX	170	167	170	191	200
	AVG	107	107	107	109	116
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	130	130	130	130	130
	Q1	277	277	277	277	277
	Q2	324	324	324	324	324
g16	Q3	353	353	353	353	353
	MAX	510	510	510	510	510
	AVG	318	318	318	318	318
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	67	67	67	67	67
	Q1	151	151	151	151	151
g04	Q2	384	384	384	384	384
	Q3	677	677	677	677	677
	MAX	980	980	980	980	980
	AVG	423	423	423	423	423
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	67	67	67	67	67
g16	Q1	151	151	151	151	151
	Q2	384	384	384	384	384
	Q3	677	677	677	677	677
	MAX	980	980	980	980	980
	AVG	423	423	423	423	423
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%

continued on the next page ...

Table A.4 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Minimum	Below Maximum	Unbounded
g09	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	704	704	704	704	704
	Q1	710	710	710	710	710
	Q2	710	710	710	710	710
	Q3	710	710	710	710	710
	MAX	710	710	710	710	710
g10	AVG	710	710	710	710	710
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	423	423	423	423	423
	Q1	544	544	544	544	544
	Q2	595	595	595	595	595
	Q3	673	673	673	673	673
g18	MAX	1,191	1,191	1,191	1,191	1,191
	AVG	618	618	618	618	618
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	119	119	119	119	119
	Q1	170	170	170	170	170
	Q2	223	223	223	223	223
g07	Q3	264	264	264	264	264
	MAX	371	371	371	371	371
	AVG	223	223	223	223	223
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	624	624	624	624	624
	Q1	671	671	671	671	671
g07	Q2	696	696	696	696	696
	Q3	724	724	724	724	724
	MAX	776	776	776	776	776
	AVG	698	698	698	698	698

continued on the next page ...

Table A.4 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Minimum	Below Maximum	Unbounded
g01	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	92	92	92	92	92
	Q1	141	141	141	141	141
	Q2	193	193	193	193	193
	Q3	229	229	229	229	229
	MAX	424	424	424	424	424
	AVG	201	201	201	201	201
g19	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	632	632	632	632	632
	Q1	807	807	807	807	807
	Q2	864	864	864	864	864
	Q3	926	926	926	926	926
	MAX	1,134	1,134	1,134	1,134	1,134
	AVG	871	871	871	871	871
g02	S.R.	63.33%	63.33%	63.33%	63.33%	63.33%
	MIN	11,140	11,140	11,140	11,140	11,140
	Q1	17,968	17,968	17,968	17,968	17,968
	Q2	26,945	26,945	26,945	26,945	26,945
	Q3	>40,000	>40,000	>40,000	>40,000	>40,000
	MAX	>40,000	>40,000	>40,000	>40,000	>40,000
	AVG	>21,991	>21,991	>21,991	>21,991	>21,991

Table A.5: FFE Count Statistics of CLS of Various Significance Thresholds under Insufficient Diversity

Problem	Measure	CLS	Unexplored	Below Minimum	Below Maximum	Unbounded
g05	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	40	40	40	40	40
	Q1	104	104	104	104	104
	Q2	127	127	127	127	127
	Q3	150	150	150	150	150
	MAX	189	189	189	189	189
	AVG	125	125	125	125	125
g21	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	174	174	174	174	174
	Q1	212	212	212	212	212
	Q2	255	255	255	255	255
	Q3	322	322	322	322	322
	MAX	465	465	465	465	465
	AVG	274	274	274	274	274
g23	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	579	579	579	579	579
	Q1	625	625	625	625	625
	Q2	710	710	710	710	710
	Q3	757	757	757	757	757
	MAX	888	888	888	888	888
	AVG	702	702	702	702	702
g22	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	87	87	87	87	87
	Q1	179	179	179	179	179
	Q2	337	337	337	337	337
	Q3	578	578	578	578	578
	MAX	1,083	1,083	1,083	1,083	1,083
	AVG	391	391	391	391	391

continued on the next page ...

Table A.5 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Minimum	Below Maximum	Unbounded
g11	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	27	27	27	27	27
	Q1	38	38	38	38	38
	Q2	50	50	50	50	50
	Q3	60	60	60	60	60
	MAX	78	78	78	78	78
g15	AVG	50	50	50	50	50
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	24	39	24	24	24
	Q1	40	40	40	40	40
	Q2	40	77	40	40	40
	Q3	40	98	40	40	40
g13	MAX	149	154	149	149	149
	AVG	48	78	51	48	48
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	220	220	220	220	220
	Q1	293	293	293	293	293
	Q2	351	351	351	351	351
g17	Q3	380	380	380	380	380
	MAX	437	437	437	437	437
	AVG	338	338	338	338	338
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	35	35	35	35	35
	Q1	105	105	105	105	105
g17	Q2	119	119	119	119	119
	Q3	149	149	149	149	149
	MAX	284	284	284	284	284
	AVG	133	133	133	133	133

continued on the next page ...

Table A.5 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Minimum	Below Maximum	Unbounded
g03	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	102	110	102	102	102
	Q1	110	157	110	110	110
	Q2	124	329	124	124	124
	Q3	236	534	236	255	255
	MAX	615	1,342	615	615	615
	AVG	196	414	196	208	208
g14	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	220	944	110	220	220
	Q1	975	1,443	975	975	975
	Q2	1,730	2,515	1,908	1,730	1,730
	Q3	3,927	3,973	5,128	3,927	3,927
	MAX	7,646	6,757	13,186	7,646	7,646
	AVG	2,424	2,928	3,067	2,424	2,424
g06	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	197	197	197	197	197
	Q1	210	210	210	210	210
	Q2	210	210	210	210	210
	Q3	210	210	210	210	210
	MAX	210	210	210	210	210
	AVG	209	209	209	209	209
g08	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	179	179	179	179	179
	Q1	300	300	300	300	300
	Q2	345	345	345	345	345
	Q3	394	394	394	394	394
	MAX	561	561	561	561	561
	AVG	350	350	350	350	350

continued on the next page ...

Table A.5 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Minimum	Below Maximum	Unbounded
g24	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	75	75	75	75	75
	Q1	93	93	93	93	93
	Q2	104	104	104	104	104
	Q3	118	118	118	118	118
	MAX	132	132	132	132	132
g12	AVG	103	103	103	103	103
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	55	47	55	55	55
	Q1	84	60	84	84	84
	Q2	108	86	108	108	108
	Q3	140	127	140	140	140
g04	MAX	170	170	170	170	170
	AVG	107	96	107	107	107
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	130	130	130	130	130
	Q1	277	277	277	277	277
	Q2	324	324	324	324	324
g16	Q3	353	353	353	353	353
	MAX	510	510	510	510	510
	AVG	318	318	318	318	318
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	67	67	67	67	67
	Q1	151	151	151	151	151
g04	Q2	384	384	384	384	384
	Q3	677	677	677	677	677
	MAX	980	980	980	980	980
	AVG	423	423	423	423	423
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	67	67	67	67	67
g16	Q1	151	151	151	151	151
	Q2	384	384	384	384	384
	Q3	677	677	677	677	677
	MAX	980	980	980	980	980
	AVG	423	423	423	423	423
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%

continued on the next page ...

Table A.5 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Minimum	Below Maximum	Unbounded
g09	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	704	704	704	704	704
	Q1	710	710	710	710	710
	Q2	710	710	710	710	710
	Q3	710	710	710	710	710
	MAX	710	710	710	710	710
g10	AVG	710	710	710	710	710
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	423	423	423	423	423
	Q1	544	544	544	544	544
	Q2	595	595	595	595	595
	Q3	673	673	673	673	673
g18	MAX	1,191	1,191	1,191	1,191	1,191
	AVG	618	618	618	618	618
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	119	119	119	119	119
	Q1	170	170	170	170	170
	Q2	223	223	223	223	223
g07	Q3	264	264	264	264	264
	MAX	371	371	371	371	371
	AVG	223	223	223	223	223
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	624	624	624	624	624
	Q1	671	671	671	671	671
g07	Q2	696	696	696	696	696
	Q3	724	724	724	724	724
	MAX	776	776	776	776	776
	AVG	698	698	698	698	698

continued on the next page ...

Table A.5 – continued from the previous page

Problem	Measure	CLS	Unexplored	Below Minimum	Below Maximum	Unbounded
g01	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	92	92	92	92	92
	Q1	141	141	141	141	141
	Q2	193	193	193	193	193
	Q3	229	229	229	229	229
	MAX	424	424	424	424	424
	AVG	201	201	201	201	201
g19	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	632	632	632	632	632
	Q1	807	807	807	807	807
	Q2	864	864	864	864	864
	Q3	926	926	926	926	926
	MAX	1,134	1,134	1,134	1,134	1,134
	AVG	871	871	871	871	871
g02	S.R.	63.33%	63.33%	63.33%	63.33%	63.33%
	MIN	11,140	11,140	11,140	11,140	11,140
	Q1	17,968	17,968	17,968	17,968	17,968
	Q2	26,945	26,945	26,945	26,945	26,945
	Q3	>40,000	>40,000	>40,000	>40,000	>40,000
	MAX	>40,000	>40,000	>40,000	>40,000	>40,000
	AVG	>21,991	>21,991	>21,991	>21,991	>21,991

A.2 Running Time

For the sake of completeness, Table A.6 depicts the CPU time taken by various MA instances in solving every benchmark problem addressed in Chapter 5 over 150 independent runs. The MA instances of interest encompass the simple MA (SMA), the one that refines individual with feasible or composite neighborhood (FCN), and all the CORE-MA instantiations (FSM, e-FSM, and CLS). Due to the pretty inexpensive evaluations involved in the benchmark problems, it may actually be rather uninteresting to observe those figures presented in the table. It must therefore be cautioned that the running time of the CORE-MA should not be mistaken for being inefficient, particularly that of the FSM-driven MAs. Herein, only matters related to the overhead incurred by performing significant assessments can actually be concluded. It can be seen from the table that upon comparable FFE requirements, the CLS-powered MA utilizes the least whereas the ecumenical FSM-driven MA needs the most CPU time. This observation is not very surprising as the ecumenical FSM-driven MA constructs in the case of composite neighborhood one SVM model per constraint per individual assessed over the course of the evolution. In contrast, the CLS-powered MA only incurs the slightest overhead such that even with some comparable FFE requirements to those of the simple MA, the CLS-powered MA only consumes comparable or slightly longer CPU time. Furthermore, the MA that refines individuals having uniformly feasible or composite neighborhood also seems to be quite efficient in term of CPU time but may suffer from the robustness issue in term of FFEs as discussed in Section 5.4.

Table A.6: CPU Time[†] Statistics of MAs with Rational Dynamic Strategies over 150 Independent Runs

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g05	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	0.25	0.09	0.30	0.31	0.06
	Q1	0.29	0.12	0.37	0.38	0.14
	Q2	0.30	0.15	0.39	0.40	0.17
	Q3	0.32	0.18	0.41	0.41	0.19
	MAX	0.36	0.25	0.48	0.48	0.26
	AVG	0.30	0.15	0.39	0.40	0.17
g21	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	0.42	0.38	0.76	0.76	0.27
	Q1	0.57	0.53	0.94	0.93	0.39
	Q2	0.65	0.58	1.01	1.02	0.46
	Q3	0.69	0.65	1.08	1.08	0.54
	MAX	0.85	0.82	1.31	1.28	0.83
	AVG	0.64	0.58	1.01	1.01	0.47
g23	S.R.	100.00%	100.00%	98.67%	100.00%	100.00%
	MIN	0.36	0.14	0.23	0.42	0.39
	Q1	0.54	0.46	0.33	0.69	0.56
	Q2	0.59	0.54	0.41	0.79	0.62
	Q3	0.64	0.60	0.51	0.89	0.66
	MAX	0.80	0.78	>1.65	1.25	0.82
	AVG	0.60	0.52	>0.45	0.79	0.61
g22	S.R.	100.00%	99.33%	100.00%	100.00%	100.00%
	MIN	0.17	0.17	0.64	0.64	0.22
	Q1	0.93	0.93	15.57	15.31	1.05
	Q2	1.41	1.38	31.09	30.00	1.57
	Q3	2.13	2.14	51.98	51.62	2.52
	MAX	40.18	>5.82	2,872.21	2,874.91	64.19
	AVG	1.83	>1.63	55.26	55.47	2.30

continued on the next page ...

Table A.6 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g11	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	0.22	0.07	0.13	0.12	0.06
	Q1	0.24	0.13	0.21	0.21	0.07
	Q2	0.25	0.15	0.23	0.23	0.08
	Q3	0.26	0.16	0.24	0.24	0.10
	MAX	0.32	0.22	0.30	0.31	0.14
	AVG	0.25	0.14	0.22	0.23	0.09
g15	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	0.06	0.04	0.08	0.08	0.05
	Q1	0.10	0.06	0.12	0.12	0.06
	Q2	0.13	0.08	0.15	0.15	0.06
	Q3	0.15	0.09	0.17	0.17	0.07
	MAX	0.30	0.23	0.37	0.34	0.23
	AVG	0.13	0.08	0.15	0.15	0.07
g13	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	0.30	0.17	0.40	0.35	0.30
	Q1	0.47	0.30	0.54	0.54	0.42
	Q2	0.53	0.37	0.60	0.61	0.49
	Q3	0.60	0.46	0.69	0.68	0.56
	MAX	2.56	63.18	2.93	2.93	2.16
	AVG	0.58	1.56	0.67	0.67	0.53
g17	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	0.42	0.11	0.51	0.51	0.06
	Q1	0.46	0.19	0.63	0.63	0.15
	Q2	0.49	0.24	0.66	0.65	0.19
	Q3	0.52	0.29	0.70	0.70	0.24
	MAX	0.64	0.40	0.84	0.83	0.40
	AVG	0.50	0.24	0.66	0.66	0.20

continued on the next page ...

Table A.6 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g03	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	0.95	0.14	0.38	0.34	0.13
	Q1	1.17	0.32	1.01	1.01	0.19
	Q2	1.33	0.41	1.18	1.20	0.34
	Q3	1.44	0.65	1.39	1.38	0.57
	MAX	4.52	4.68	4.83	4.85	5.55
	AVG	1.39	0.58	1.27	1.27	0.56
g14	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	0.89	0.89	0.86	0.86	0.33
	Q1	2.54	2.55	2.86	2.90	1.79
	Q2	3.62	3.59	4.15	4.23	2.70
	Q3	4.74	4.79	5.42	5.51	4.62
	MAX	11.86	11.94	14.05	13.44	15.49
	AVG	4.05	4.03	4.48	4.50	3.52
g06	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	0.44	0.05	0.05	0.05	0.46
	Q1	0.49	0.07	0.07	0.07	0.49
	Q2	0.50	0.07	0.07	0.07	0.51
	Q3	0.53	0.08	0.08	0.08	0.53
	MAX	1.52	0.15	0.13	0.15	0.59
	AVG	0.52	0.08	0.07	0.08	0.51
g08	S.R.	100.00%	99.33%	99.33%	100.00%	100.00%
	MIN	0.35	0.05	0.05	0.17	0.36
	Q1	0.62	0.18	0.16	0.41	0.51
	Q2	0.68	0.27	0.30	0.50	0.60
	Q3	0.74	0.48	0.53	0.59	0.67
	MAX	1.34	>6.56	>22.72	1.54	1.15
	AVG	0.66	0.43	0.56	0.52	0.59

continued on the next page ...

Table A.6 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g24	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	0.19	0.14	0.13	0.15	0.13
	Q1	0.22	0.18	0.18	0.20	0.18
	Q2	0.23	0.20	0.20	0.22	0.20
	Q3	0.25	0.22	0.22	0.24	0.22
	MAX	0.30	0.56	8.41	0.30	0.29
	AVG	0.23	0.21	0.27	0.22	0.20
g12	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	0.09	0.08	0.04	0.04	0.09
	Q1	0.16	0.12	0.08	0.08	0.14
	Q2	0.20	0.15	0.09	0.10	0.17
	Q3	0.24	0.18	0.13	0.13	0.20
	MAX	0.42	0.39	0.32	0.33	0.32
	AVG	0.20	0.15	0.11	0.11	0.17
g04	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	0.32	0.32	0.12	0.30	0.23
	Q1	0.38	0.38	0.25	0.45	0.31
	Q2	0.41	0.40	0.30	0.48	0.35
	Q3	0.43	0.43	0.35	0.52	0.39
	MAX	0.50	0.58	0.44	0.62	0.55
	AVG	0.41	0.40	0.29	0.48	0.35
g16	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	2.28	2.28	2.32	3.04	2.39
	Q1	6.04	8.74	6.52	8.93	6.93
	Q2	18.76	21.20	18.51	28.83	21.45
	Q3	28.04	29.12	27.76	39.76	32.39
	MAX	55.93	53.98	55.46	75.72	65.42
	AVG	18.74	20.51	18.63	26.96	21.72

continued on the next page ...

Table A.6 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g09	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	1.43	0.16	0.15	1.08	1.44
	Q1	1.59	0.33	0.21	1.42	1.60
	Q2	1.65	0.42	0.32	1.58	1.67
	Q3	1.73	0.62	0.43	1.76	1.75
	MAX	2.01	1.51	1.26	2.19	2.02
	AVG	1.67	0.50	0.36	1.59	1.69
g10	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	0.77	0.16	0.14	1.07	0.78
	Q1	1.01	0.48	0.23	1.41	1.03
	Q2	1.11	0.79	0.43	1.54	1.11
	Q3	1.28	1.02	0.68	1.71	1.31
	MAX	3.23	3.03	1.66	4.37	3.14
	AVG	1.23	0.82	0.49	1.66	1.23
g18	S.R.	100.00%	100.00%	93.33%	100.00%	100.00%
	MIN	0.31	0.20	0.17	0.92	0.30
	Q1	0.59	0.48	0.25	1.25	0.60
	Q2	0.70	0.60	0.35	1.39	0.72
	Q3	0.83	0.72	0.52	1.56	0.85
	MAX	1.65	351.89	>337.04	3.25	1.90
	AVG	0.72	3.64	26.40	1.41	0.73
g07	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	1.80	0.21	0.18	2.31	1.81
	Q1	1.93	0.39	0.23	2.64	1.95
	Q2	2.01	0.43	0.25	2.72	2.01
	Q3	2.09	0.52	0.28	2.80	2.09
	MAX	2.44	1.30	0.65	3.34	2.43
	AVG	2.02	0.49	0.26	2.72	2.03

continued on the next page ...

Table A.6 – continued from the previous page

Problem	Measure	All	FCN	FSM	e-FSM	CLS
g01	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	0.35	0.36	0.12	0.51	0.38
	Q1	0.41	0.41	0.25	0.95	0.43
	Q2	0.46	0.46	0.50	1.23	0.48
	Q3	0.65	0.66	0.75	1.71	0.56
	MAX	2.89	2.89	8.24	3.75	1.51
g19	AVG	0.56	0.56	0.69	1.39	0.55
	S.R.	100.00%	100.00%	100.00%	100.00%	100.00%
	MIN	1.95	1.97	1.31	2.80	0.97
	Q1	2.17	2.18	2.00	3.07	1.60
	Q2	2.23	2.24	2.16	3.18	1.76
	Q3	2.30	2.31	2.31	3.28	1.97
g02	MAX	2.67	2.65	2.73	3.86	2.32
	AVG	2.24	2.25	2.16	3.19	1.77
	S.R.	32.67%	50.00%	64.67%	64.67%	66.67%
	MIN	30.67	30.30	33.32	33.30	22.66
	Q1	100.37	73.61	138.12	139.21	58.81
	Q2	>135.26	>147.87	189.23	186.06	91.94
	Q3	>135.26	>147.87	>231.00	>231.06	>155.97
	MAX	>135.26	>147.87	>231.00	>231.06	>155.97
	AVG	>115.57	>111.75	>180.32	>180.15	>99.57

†: in seconds

Bibliography

- [1] Ong, Y.S., P.B. Nair, and A.J. Keane, "Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling," *American Institute of Aeronautics and Astronautics*, 41(4), pp. 687–696, 2003.
- [2] Zhou, Z.Z., et al., "Memetic Algorithm using Multi-Surrogates for Computationally Expensive Optimization Problems," *Soft Computing: A Fusion of Foundations, Methodologies, and Applications*, 11(10), pp. 957–971, 2007.
- [3] Zhou, Z.Z., et al., "Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization," *IEEE Transactions On Systems, Man, and Cybernetics - Part C*, 37(1), pp. 66–76, 2007.
- [4] Land, M.W.S., "Evolutionary Algorithms with Local Search for Combinatorial Optimization," Ph.D. Thesis, University of California, 1998.
- [5] Le, M.N., Y.S. Ong, and Q.H. Nguyen, "Optinformatics for Schema Analysis of Binary Genetic Algorithms," in *Proceedings of the 2008 Genetic and Evolutionary Computation Conference*, pp. 1121–1122, 2008.
- [6] Karush, W., "Minima of Functions of Several Variables with Inequalities as Side Conditions," M.Sc. Dissertation, University of Chicago, 1939.
- [7] Kuhn, H.W. and A.W. Tucker, "Nonlinear Programming," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492, 1950.

- [8] Michalewicz, Z., "Genetic Algorithms, Numerical Optimization, and Constraints," in *Proceedings of the Sixth International Conference on Genetic Algorithms*, pp. 151–158, 1995.
- [9] Spall, J.C., "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation," *IEEE Transactions on Automatic Control*, 37(3), pp. 332–341, 1992.
- [10] Spall, J.C., "Implementation of the Simultaneous Perturbation Algorithm for Stochastic Optimization," *IEEE Transactions on Aerospace and Electronic Systems*, 34(3), pp. 817–823, 1998.
- [11] Handoko, S.D., et. al., "A Study on Constrained MA Using GA and SQP: Analytical vs. Finite-Difference Gradients," in *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*, 2008.
- [12] Nocedal, J. and S.J. Wright, *Numerical Optimization*, Springer, 2006.
- [13] Griewank, A., *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, SIAM, 2000.
- [14] Naumann, U., "Optimal Accumulation of Jacobian Matrices by Elimination Methods on the Dual Computational Graph," *Mathematical Programming*, 99(3), pp. 399–421, 2004.
- [15] Levitt, M., "Conformation Analysis of Proteins," Ph.D. Thesis, University of Cambridge, 1971.
- [16] Ong, Y.S., K.Y. Lum, and P.B. Nair, "Hybrid Evolutionary Algorithm with Hermite Radial Basis Function Interpolants for Computationally Expensive Adjoint Solvers," *Computational Optimization and Applications*, 39(1), pp. 97–119, 2007.
- [17] Dantzig, G.B., *Linear Programming and Extensions*, Princeton University Press, 1963.

- [18] Forsgren, A., P.E. Gill, and M.H. Wright, "Interior Methods for Nonlinear Optimization," *SIAM Review* 44(4), pp. 525–597, 2002.
- [19] Klee, V. and G.J. Minty, "How Good is the Simplex Algorithm?," in *Inequalities III*, pp. 159–175, 1972.
- [20] Karmarkar, N., "A New Polynomial-time Algorithm for Linear Programming," *Combinatorica*, 4(4), pp. 373–395, 1984.
- [21] Hestenes, M.R. and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," *Journal of Research of the National Bureau of Standards*, 49(6), pp. 409–436, 1952.
- [22] Bazaraa, M.S., H.D. Sherali, and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms*, Wiley-Interscience, 2006.
- [23] Zoutendijk, G., *Methods of Feasible Directions*, Elsevier, 1960.
- [24] Topkis, D.M. and A.F. Veinott, "On the Convergence of Some Feasible Direction Algorithms for Nonlinear Programming," *SIAM Journal on Control*, 5(2), pp. 268–279, 1967.
- [25] Zangwill, W.I., *Nonlinear Programming: A Unified Approach*, Prentice-Hall, 1969.
- [26] Griffith, R.E. and R.A. Stewart, "A Nonlinear Programming Technique for the Optimization of Continuous Processing Systems," *Management Science*, 7(4), pp. 379–392, 1961.
- [27] Buzby, B.R., "Techniques and Experience Solving Really Big Nonlinear Programs," in *Proceedings of Conference on Optimization Methods for Resource Allocation*, pp. 227–237, 1974.
- [28] Boddington, C.E. and W.C. Randall, "Nonlinear Programs for Product Blending," Joint National TIMS/ORSA Meeting, New Orleans, 1979.

- [29] Baker, T.E. and R. Ventker, "Successive Linear Programming in Refinery Logistic Models," ORSA/TIMS Joint National Meeting, Colorado Springs, 1980.
- [30] Zhang, J.Z., N.H. Kim, and L.S. Lasdon, "An Improved Successive Linear Programming Algorithm," *Management Science*, 31(10), pp. 1312–1331, 1985.
- [31] Baker, T.E. and L.S. Lasdon, "Successive Linear Programming at Exxon," *Management Science*, 31(3), pp. 264–274, 1985.
- [32] Wilson, R.B., "A Simplicial Algorithm for Convex Programming," Ph.D. Thesis, Harvard University, 1963.
- [33] Schittkowski, K., "NLPQL: A FORTRAN Subroutine Solving Constrained Nonlinear Programming Problems," *Annals of Operations Research*, 5(2), pp. 485–500, 1986.
- [34] Spedicato, E., *Algorithms for Continuous Optimization: The State of the Art*, Springer, 1994.
- [35] Han, S.P., "Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems," *Mathematical Programming*, 11(1), pp. 263–282, 1976.
- [36] Powell, M.J.D., "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations," in *Lecture Notes in Mathematics*, 630, pp. 144–157, 1978.
- [37] Wright, S.J., "Nonlinear and Semidefinite Programming," in *Proceedings of the 2004 Symposia in Applied Mathematics*, pp. 115–138, 2004.
- [38] Dumitrescu, D., et al., *Evolutionary Computation*, CRC Press, 2000.
- [39] De Jong, K.A., *Evolutionary Computation: A Unified Approach*, The MIT Press, 2006.

- [40] Holland, J.H., *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, 1975.
- [41] Strickberger, M.W., *Genetics*, Macmillan, 1976.
- [42] Darwin, C., *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*, John Murray, 1859.
- [43] Li, X., "Using Genetic Algorithms for An Optical Thin-Film Learning Model," in *Proceedings of Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, pp.126–130, 1997.
- [44] Donate, J.P., et al., "Time Series Forecasting by Evolving Artificial Neural Networks Using Genetic Algorithms and Differential Evolution," in *Proceedings of 2010 International Joint Conference on Neural Networks*, pp.3999–4006, 2010.
- [45] Donate, J.P., et al., "Time Series Forecasting by Evolving Artificial Neural Networks Using Genetic Algorithms, Differential Evolution and Estimation of Distribution Algorithm," *Journal of Neural Computing and Applications*, 2011.
- [46] Bonyadi, M.R. and X. Li, "A New Discrete Electromagnetism-based Meta-heuristic for Solving the Multidimensional Knapsack Problem Using Genetic Operators", *Operational Research*, 12(2), pp.229–252, 2012.
- [47] Goldberg, D.E., "Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking," *Complex Systems*, 5, pp. 139–167, 1991.
- [48] Coello, C.A.C., "Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art," *Computer Methods in Applied Mechanics and Engineering*, 191(11–12), pp. 1245–1287, 2002.

- [49] Hamida, S.B. and M. Schoenauer, "ASCHEA: New Results Using Adaptive Segregational Constraint Handling," in *Proceedings of the 2002 Congress on Evolutionary Computation*, pp. 884–889, 2002.
- [50] Barbosa, H.J.C. and A.C.C. Lemonge, "A New Adaptive Penalty Scheme for Genetic Algorithms," *Information Science*, 156(3–4), pp. 215–251, 2003.
- [51] Farmani, R. and J.A. Wright, "Self-Adaptive Fitness Formulation for Constrained Optimization," *IEEE Transactions on Evolutionary Computation*, 7(5), pp. 445–455, 2003.
- [52] Chootinan, P. and A. Chen, "Constraint Handling in Genetic Algorithms Using a Gradient-based Repair Method," *Computers and Operation Research*, 33(8), pp. 2263–2281, 2006.
- [53] Mezura-Montes, E. and C.A.C. Coello, "A Simple Multi-Membered Evolution Strategy to Solve Constrained Optimization Problems," *IEEE Transactions on Evolutionary Computation*, 9(1), pp. 1–17, 2005.
- [54] Runarsson, T.P. and Y. Xin, "Stochastic Ranking for Constrained Evolutionary Optimization," *IEEE Transactions on Evolutionary Computation*, 4(3), pp. 284–294, 2000.
- [55] Deb, K., "An Efficient Constraint Handling Method for Genetic Algorithms," *Computer Methods in Applied Mechanics and Engineering*, 186(2–4), pp. 311–338, 2000.
- [56] Ho, P.Y. and K. Shimizu, "Simple Addition of Ranking Method for Constrained Optimization in Evolutionary Algorithms," in *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, pp.889–896, 2005.
- [57] Elfeky, E.Z., R.A. Sarker, and D.L. Essam, "A Simple Ranking and Selection for Constrained Evolutionary Optimization," in *Lecture Notes in Computer Science*, 4247, Springer-Verlag, 2006.

- [58] Freitas, A.A., *Data Mining and Knowledge Discovery with Evolutionary Algorithms*, Springer, 2002.
- [59] Baker, J.E., "Reducing Bias and Inefficiency in the Selection Algorithm," in *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 14–21, 1987.
- [60] Davis, L., *Genetic Algorithms and Simulated Annealing*, Pitman, 1987.
- [61] Davis, L., "Adapting Operator Probabilities in Genetic Algorithms," in *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 61–69, 1989.
- [62] Michalewics, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 1992.
- [63] Eshelman, L.J. and J.D. Schaffer, "Real-coded Genetic Algorithms and Interval-Schemata," *Foundations of Genetic Algorithms*, 2, pp. 187–202, 1993.
- [64] Raghuwanshi, M. and O. Kakde, "Survey on Multiobjective Evolutionary and Real Coded Genetic Algorithms," in *Proceedings of the Eighth Asia Pacific Symposium on Intelligent and Evolutionary Systems*, pp. 150–161, 2004.
- [65] Fogel, L.J., A.J. Owens, and M.J. Walsh, *Artificial Intelligence through Simulated Evolution*, Wiley, 1966.
- [66] Fogel, D.B., *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, 1995.
- [67] Tang, J., M.H. Lim, and Y.S. Ong, "Diversity-Adaptive Parallel Memetic Algorithm for Solving Large Scale Combinatorial Optimization Problems," *Soft Computing: A Fusion of Foundations, Methodologies, and Applications*, 11(9), pp. 873–888, 2007.

- [68] Torn, A. and A. Zilinskas, *Global Optimization*, Springer-Verlag, 1989.
- [69] Dawkins, R., *The Selfish Gene*, Oxford University Press, 1976.
- [70] Moscato, P., "On Evolution, Search, Optimization, Genetic Algorithms, and Martial Arts: Towards Memetic Algorithms," Technical Report C3P 826, California Institute of Technology, 1989.
- [71] Ong, Y.S. and A.J. Keane, "Meta-Lamarckian Learning in Memetic Algorithms," *IEEE Transactions on Evolutionary Computation*, 8(2), pp. 99–110, 2004.
- [72] Yeh, W.C., "An Efficient Memetic Algorithm for the Multi-stage Supply Chain Network Problem," *International Journal of Advanced Manufacturing Technology*, 29(7–8), pp. 803–813, 2006.
- [73] Yeh, W.C., "A Memetic Algorithm for the $n/2$ /Flowshop/ $aF+bC_{max}$ Scheduling Problem," *International Journal of Advanced Manufacturing Technology*, 20(6), pp. 464–473, 2002.
- [74] Al-Betar, M.A., A.T. Khader, and M. Zaman, "University Course Timetabling Using a Hybrid Harmony Search Metaheuristic Algorithm," *IEEE Transactions on Systems, Man, and Cybernetics - Part C*, 42(5), pp. 664–681, 2012.
- [75] Urselmann, M., et al., "A Memetic Algorithm for Global Optimization in Chemical Process Synthesis Problems," *IEEE Transactions on Evolutionary Computation*, 15(5), pp. 659–683, 2011.
- [76] Ullah, A.S.S.M.B., et al., "An Agent-based Memetic Algorithm (AMA) for Solving Constrained Optimazation Problems," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, pp. 999–1006, 2007.
- [77] Ullah, A.S.S.M.B., R. Sarker, and C. Lokan, "Handling Equality Constraints with Agent-based Memetic Algorithms," *Memetic Computing*, 3(1), pp. 51–72, 2011.

- [78] Tenne, Y., K. Izui, and S. Nishiwaki, "Handling Undefined Vectors in Expensive Optimization Problems," in *Lecture Notes in Computer Science*, 6024, pp. 582–591, 2010.
- [79] Tenne, Y., K. Izui, and S. Nishiwaki, "A Hybrid Model-classifier Framework for Managing Prediction Uncertainty in Expensive Optimisation Problems," *International Journal of Systems Science*, 43(7), pp. 1305–1321, 2012.
- [80] Hart, W.E., "Adaptive Global Optimization with Local Search," Ph.D. Thesis, University of California, 1994.
- [81] Campbell-Kelly, M., et al. *The History of Mathematical Tables: From Sumer to Spreadsheets*, Oxford University Press, 2003.
- [82] Cormen, T.H., et al., *Introduction To Algorithms*, MIT Press, 2001.
- [83] Lynch, C., "Big Data: How Do Your Data Grow?," *Nature*, 455, pp.28–29, 2008.
- [84] Berman, H., K. Henrick, and H. Nakamura, "Announcing the Worldwide Protein Data Bank," *Nature Structural Biology*, 10(12), pp. 980, 2003.
- [85] Lennard-Jones, J.E., "Cohesion," *Proceedings of the Physical Society*, 43, pp. 461–482, 1931.
- [86] Bujnicki, J., *Prediction of Protein Structures, Functions, and Interactions*, John Wiley & Sons, 2009.
- [87] Jones, D.T., W.R. Taylor, and J.M. Thornton, "A New Approach to Protein Fold Recognition," *Nature*, 358(6381), pp. 86–89, 1992.
- [88] Clarke, D.A., *Foundations of Analysis*, Appleton-Century-Crofts, 1971.
- [89] Vapnik, V.N. and A. Lerner, "Pattern Recognition Using Generalized Portrait Method," *Automation and Remote Control*, 24, 1963.

- [90] Vapnik, V.N., *The Nature of Statistical Learning Theory*, Springer, 1995.
- [91] Schölkopf, B., C.J.C. Burges, and S. Mika, *Advances in Kernel Methods*, The MIT Press, 1998.
- [92] Aizerman, M., E. Braverman, and L. Rozonoer, "Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning," *Automation and Remote Control*, 25, pp. 821–837, 1964.
- [93] Boser, B.E., I.M. Guyon, and V.N. Vapnik. "A Training Algorithm for Optimal Margin Classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152, 1992.
- [94] S.D. Handoko, C.K. Kwoh, and Y.S. Ong, "Using Classification for Constrained Memetic Algorithm: A New Paradigm," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pp.547-552, 2008.
- [95] S.D. Handoko, C.K. Kwoh, and Y.S. Ong, "Feasibility Structure Modeling: An Effective Chaperone for Constrained Memetic Algorithms," *IEEE Transactions on Evolutionary Computation*, 14(5), pp.740–758, 2010.
- [96] S.D. Handoko, C.K. Kwoh, and Y.S. Ong, "Classification-assisted Memetic Algorithms for Equality-constrained Optimization Problems," *Lecture Notes in Artificial Intelligence*, 5866, pp.391-400, 2009.
- [97] S.D. Handoko, C.K. Kwoh, Y.S. Ong, and J. Chan, "Classification-assisted Memetic Algorithms for Solving Optimization Problems with Restricted Equality Constraint Function Mapping," in *Proceedings of 2011 IEEE Congress on Evolutionary Computation*, pp.1209–1216, 2011.
- [98] Handoko, S.D., et al., "QuickVina: Accelerating AutoDock Vina Using Gradient-based Heuristics for Global Optimization," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(5), pp.1266–1272, 2012.

- [99] Liang, J.J., et al., "Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization," Technical Report, Nanyang Technological University, 2006.
- [100] McKay, M.D., R.J. Beckman, and W.J. Conover, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, 21(2), pp. 239-245, 1979.
- [101] Cornell, W.D., et al., "A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules," *Journal of the American Chemical Society*, 117, pp. 5179–5197, 1995.
- [102] Trott, O. and A.J. Olson, "AutoDock Vina: Improving the Speed and Accuracy of Docking with A New Scoring Function, Efficient Optimization, and Multithreading," *Journal of Computational Chemistry*, 31(2), pp. 455–461, 2010.
- [103] Romagnoli, P., "Peptide-MHC Interaction: A Rational Approach to Vaccine Design," *International Reviews of Immunology*, 6(1), pp. 61–73, 1990.
- [104] Zhao, B., et al., "MHC-Peptide Binding Prediction for Epitope-based Vaccine Design," *International Journal of Integrative Biology*, 1(2), pp. 127–140, 2007.
- [105] Williams, T.M., "Human Leukocyte Antigen Gene Polymorphism and the Histocompatibility Laboratory," *Journal of Molecular Diagnostics*, 3(3), pp. 98–104, 2001.
- [106] Marsh, S.G.E., et al., "Nomenclature for factors of the HLA system," *Tissue Antigens*, 65(4), pp. 301–369, 2005.
- [107] Brusica, V., V.B. Bajic, and N. Petrovsky, "Computational Methods for Prediction of T-cell Epitopes—A Framework for Modelling, Testing, and Applications," *Methods*, 34(4), pp. 436–443, 2004.

- [108] Yu, K., et al., "Methods for Prediction of Peptide Binding to MHC Molecules: A Comparative Study," *Molecular Medicine*, 8(3), pp. 137–148., 2002.
- [109] Savoie, C.J., et al., "Use of BONSAI Decision Trees for the Identification of Potential MHC Class I Peptide Epitope Motifs," in *Proceedings of the 1999 Pacific Symposium on Biocomputing*, pp. 182–189, 1999.
- [110] Brusic, V., et al., "Prediction of Promiscuous Peptides that Bind HLA Class I Molecules," *Immunology & Cell Biology*, 80(3), pp. 280–285, 2002.
- [111] Zhang, G.L., et al., "Neural Models for Predicting Viral Vaccine Targets," *Journal of Bioinformatics and Computational Biology*, 3(5), pp. 1207–1225, 2005.
- [112] Handoko, S.D., et al., "Extreme Learning Machine for Predicting HLA-Peptide Binding," in *Lecture Notes in Computer Science*, 3973, pp. 716–721, 2006.
- [113] Zhang, G.L., et al., "Prediction of Supertype-specific HLA Class I Binding Peptides Using Support Vector Machines," *Journal of Immunological Methods*, 320(1–2), pp. 143–154, 2007.
- [114] Zhang, G.L., et al., "Computational Models for Identifying Promiscuous HLA-B7 Binders Based on Information Theory and Support Vector Machine," in *Proceedings of the 2006 IEEE International Conference on Biomedical and Pharmaceutical Engineering*, pp. 319–323, 2007.
- [115] Rognan, D., et al., "Molecular Dynamics Simulation of MHC-Peptide Complexes as a Tool for Predicting Potential T Cell Epitopes," *Biochemistry*, 33(38), pp. 11476–11485, 1994.
- [116] Rognan, D., et al., "Predicting Binding Affinities of Protein Ligands from Three-Dimensional Models: Application to Peptide Binding to Class I Major Histocompatibility Proteins," *Journal of Medicinal Chemistry*, 42(22), pp. 4650–4658, 1999.

- [117] Leach, A.R., *Molecular Modelling: Principles and Applications*, Prentice Hall, 2001.
- [118] Wales, D.J., *Energy Landscapes: With Applications to Clusters, Biomolecules, and Glasses*, Cambridge University Press, 2003.
- [119] Grozema, F.C., R.W.J. Zijlstra, and P.T. van Duijnen, "Many-body Interactions Calculated with the Direct Reaction Field Model," *Chemical Physics*, 246, pp. 217-227, 1999.
- [120] Weiner, S.J., et al., "A New Force Field for Molecular Mechanical Simulation of Nucleic Acids and Proteins," *Journal of the American Chemical Society*, 106, pp. 765-784, 1984.
- [121] Weiner, S.J., et al., "An All Atom Force Field for Simulations of Proteins and Nucleic Acids," *Journal of Computational Chemistry*, 7(2), pp. 230-252, 1986.
- [122] Duan, Y., et al., "A Point-charge Force Field for Molecular Mechanics Simulations of Proteins Based on Condensed-phase Quantum Mechanical Calculations," *Journal of Computational Chemistry*, 24(16), pp. 1999-2012, 2003.
- [123] Brooks, B.R., et al., "CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations," *Journal of Computational Chemistry*, 4(2), pp. 187-217, 1983.
- [124] Levitt, M., et al., "Potential Energy Function and Parameters for Simulations of the Molecular Dynamics of Proteins and Nucleic Acids in Solution," *Computer Physics Communications*, 91, pp. 215-231, 1995.
- [125] Jorgensen, W.L. and J. Tirado-Rives, "The OPLS [Optimized Potentials for Liquid Simulations] Potential Functions for Proteins, Energy Minimization for Crystals of Cyclic Peptides and Crambin," *Journal of the American Chemical Society*, 110(6), pp. 1657-1666, 1988.

- [126] van Holde, K.E., W.C. Johnson, and P.S. Ho, *Principles of Physical Biochemistry*, Prentice-Hall, 1998.
- [127] Blondel, A. and M. Karplus, "New Formulation for Derivatives of Torsion Angles and Improper Torsion Angles in Molecular Mechanics: Elimination of Singularities," *Journal of Computational Chemistry*, 17(9), pp. 1132–1141, 1996.
- [128] Tuzun, R.E., D.W. Noid, and B.G. Sumpter, "Computation of Internal Coordinates, Derivatives, and Gradient Expressions: Torsion and Improper Torsion," *Journal of Computational Chemistry*, 21(7), pp. 553–561, 2000.
- [129] Stone, A.J., *The Theory of Intermolecular Forces*, Oxford University Press, 1997.
- [130] Hill, T.L., "Steric Effects. I. van der Waals Potential Energy Curves," *Journal of Chemical Physics*, 16(4), pp. 399–404, 1948
- [131] Voet, D. and J.G. Voet, *Biochemistry*, Wiley, 2004.
- [132] Thompson, H.B., "Calculation of Cartesian Coordinates and Their Derivatives from Internal Molecular Coordinates," *Journal of Chemical Physics*, 47(9), pp. 3407–3410, 1967.
- [133] Thompson, H.B., "Calculation of Cartesian Coordinates and Their Derivatives from Internal Molecular Coordinates. II. Second and Higher Derivatives and Derivatives of Vectors," *Journal of Chemical Physics*, 53(8), pp. 3034–3036, 1970.
- [134] Momany, F.A., et al., "Energy Parameters in Polypeptides. VII. Geometric Parameters, Partial Atomic Charges, Nonbonded Interactions, Hydrogen Bond Interactions, and Intrinsic Torsional Potentials for the Naturally Occurring Amino Acids," *Journal of Physical Chemistry*, 79(22), pp. 2361–2381, 1975.
- [135] Withford, D., *Proteins: Structure and Function*, Wiley, 2005.

- [136] Ramachandran, G.N. and V. Sasisekharan, "Conformation of Polypeptides and Proteins," *Advances in Protein Chemistry*, pp. 283–437, 1968.
- [137] Pauling, L., *The Nature of the Chemical Bond*, Cornell University Press, 1945.
- [138] Sousa, S.F., P.A. Fernandes, and M.J. Ramos, "Protein–Ligand Docking: Current Status and Future Challenges," *Proteins: Structure, Function, and Bioinformatics*, 65(1), pp. 15–26, 2006.
- [139] Halperin, I., et al., "Principles of Docking: An Overview of Search Algorithms and A Guide to Scoring Functions," *Proteins: Structure, Function, and Bioinformatics*, 47(4), pp. 409–443, 2002.
- [140] Morris, G.M., et al., "Automated Docking Using A Lamarckian Genetic Algorithm and An Empirical Binding Free Energy Function," *Journal of Computational Chemistry*, 19(14), pp. 1639–1662, 1998.
- [141] Jones, G., et al., "Development and Validation of A Genetic Algorithm for Flexible Docking," *Journal of Molecular Biology*, 267(3), pp. 727–748, 1997.
- [142] Rarey, M., et al., "A Fast Flexible Docking Method Using An Incremental Construction Algorithm," *Journal of Molecular Biology*, 261(3), pp. 470–489, 1996.
- [143] Abagyan, R., M. Totrov, and D. Kuznetsov, "ICM—A New Method for Protein Modeling and Design: Applications to Docking and Structure Prediction from the Distorted Native Conformation," *Journal of Computational Chemistry*, 15(5), pp. 488–506, 1994.
- [144] Shanno, D., "Conditioning of Quasi-Newton Methods for Function Minimization," *Mathematics of Computation*, 24(111), pp. 647–656, 1970.

- [145] Wang, R., et al., "The PDBbind Database: Collection of Binding Affinities for Protein–Ligand Complexes with Known Three-dimensional Structures," *Journal of Medicinal Chemistry*, 47(12), pp. 2977–2980, 2004.
- [146] Toussaint, N.C., P. D’nnnes, and O. Kohlbacher, "A Mathematical Framework for the Selection of An Optimal Set of Peptides for Epitope-based Vaccines," *PLoS Computational Biology*, 4(12), 2008.
- [147] Yanover, C. and P. Bradley, "Large-scale Characterization of Peptide–MHC Binding Landscapes with Structural Simulations," *PNAS*, 108(17), pp.6981–6986, 2011.
- [148] Rueckert, C. and C.A. Guzmán, "Vaccines: From Empirical Development to Rational Design," *PLoS Pathogens*, 8(11), 2012.
- [149] Ellabaan, M.M.H., et al., "A Tree-structured Covalent-bond-driven Molecular Memetic Algorithm for Optimization of Ring-deficient Molecules," *Computers and Mathematics with Applications*, 64, pp.3792–3804, 2012.
- [150] Ellabaan, M.M.H., et al., "Discovering Unique, Low-energy Transition States Using Evolutionary Molecular Memetic Computing," *IEEE Computational Intelligence Magazine*, in Press.
- [151] Ruthven, D.M., *Principles of Adsorption & Adsorption Processes*, Wiley, 1984.
- [152] Grigsby, L.L., *Power System Stability and Control*, CRC Press, 2012.
- [153] Toth, P. and D. Vigo, *The Vehicle Routing Problem*, SIAM, 2002.