



Using Parallel Computing to Speed-up Beam Forming Patterns for Real Time Applications

A Thesis Submitted to the Department of Computer Science, Faculty
of Computer and Information Sciences, Ain Shams University, in
Partial Fulfilment of the Requirements for the Master Degree of
Computer and Information Sciences

By

Eman Ahmed Fahmy Mostafa

B.Sc. Degree in Computer and Information Sciences (2008)
Department of Computer Science,
Faculty of Computer and Information Sciences,
Ain Shams University.

Supervised by

Prof. Dr. Zaki Taha Fayed

Computer Science Department
Faculty of Computer and Information Sciences
Ain Shams University

Dr. Safwat Helmy Hamad

Scientific Computing Department
Faculty of Computer and Information Sciences
Ain Shams University

Dr. Korany Ragab Mahmoud

Electronics, Communications and Computers Department
Faculty of Engineering
Helwan University

October, 2013



Using Parallel Computing to Speed-up Beam Forming Patterns for Real Time Applications

A Thesis Submitted to the Department of Computer Science, Faculty
of Computer and Information Sciences, Ain Shams University, in
Partial Fulfilment of the Requirements for the Master Degree of
Computer and Information Sciences

By

Eman Ahmed Fahmy Mostafa

B.Sc. Degree in Computer and Information Sciences (2008)
Department of Computer Science,
Faculty of Computer and Information Sciences,
Ain Shams University.

Approved by the discussion committee:

Prof. Dr. Zaki Taha Fayed Member and Supervisor
Professor in the Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.
.....

Prof. Dr. Mahmoud Elsaid Gadalla Member
Professor of Computer Engineering at Military Technical College.
.....

Prof. Dr. Aboul Ella Hassanien Member
Professor at Faculty of Computers and Information,
Cairo University.
.....

October, 2013

ACKNOWLEDGMENTS

First, I want to thank Prof. Dr. Zaki Taha for his continuous caring and his support along the research period.

Also I want to thank Dr. Safwat as he is the first one encourage me to do this research and select the research topic with me and support me in it and open to me the way that I went in and he helps me to understand CUDA and Open-MP and help me in writing and publishing the thesis's papers.

Also he let me know Dr. Korany that helps me a lot to understand the Antenna and how it works and helps me in result validation and improving the accuracy of the results and also helps me in writing and publishing the thesis's papers.

Finally want to thank my parents and my brother for supporting me also want to thank my friends and my family who caring about me and always wish to me the best.

ABSTRACT

The scientific community is still interested in heuristic techniques and optimization algorithms that could be applied in complex problems such as adaptive antenna beam-forming problem. This thesis presents an empirical study of solving the problem of adaptive antenna beam-forming using Particle Swarm Optimization (PSO), Central Force Optimization (CFO) and Gravitational Search Algorithm (GSA) using Open Multi-Processing (Open-MP) or using Compute Unified Device Architecture (CUDA) which then applied on a Graphics Processing Unit (GPU). The PSO, CFO and GSA are well known alternatives for global optimization based on a nature-inspired Heuristic. Extensive experimentations were applied to compare their performance through a number of case studies. PSO showed to have good performance, low computational complexity, few parameters and gives good results. On the other hand, CFO has a higher computational complexity but it gives better results. Furthermore, although GSA has the same idea of CFO, it performs different computations and hence outperforms the other optimization techniques for the adaptive beam-forming problem. It is found that the resulting beam-patterns optimized by the PSO, CFO or GSA required a large processing time which is not acceptable for on line applications. Hence, the demand for a parallel solution that accelerates these computations is considered. Therefore, a parallel version of PSO, CFO and GSA is proposed and implemented using Open-MP or using CUDA which then applied on a GPU. The comparison is presented to show how the parallel version of the PSO, CFO and GSA outperforms the sequential one, thus an online

procedure is available for time-critical applications such as in adaptive beam-forming applications.

TABLE OF CONTENTS

<i>Abstract.....</i>	<i>4</i>
<i>Table of Contents.....</i>	<i>6</i>
<i>List of Figures.....</i>	<i>9</i>
<i>List of Tables.....</i>	<i>11</i>
<i>List of abbreviation.....</i>	<i>12</i>
<u>Chapter1 : Introduction.....</u>	<u>15</u>
1.1. Motivation and problem Statement.....	15
1.2. Research Objectives	16
1.3. Thesis Overview	17
<u>Chapter2 : Literature Survey</u>	<u>20</u>
2.1 Introduction	20
2.2 Antenna	21
2.3 Optimization and Evolutionary Algorithms	36
2.4 Parallel Computing	45
<u>Chapter3 : Proposed Optimization Algorithms</u>	<u>55</u>
3.1 Introduction	55
3.2 Evolutionary (Global) Optimization Algorithms.....	56
3.2.1 Paricle Swarm Optimization (PSO) Algorithm.....	58
3.2.2 Central Force Optimization (CFO) Algorithm	66
3.2.3 Gravitational Search Algorithm (GSA).....	76
3.3 Evolutionary Algorithms Comparison.....	80
3.4 Local Optimization Algorithms.....	82
3.4.1 Nelder Mead Algorithm	83

3.5	Hybrid Optimization Algorithms	84
<u>Chapter4 :</u>	<u>Parallel Optimization Algorithm</u>	<u>86</u>
4.1	Introduction	86
4.2	Parallel Implementation of Evolutionary (Global) Optimization Algorithms	87
4.2.1	Parallel Implementation of Particle Swarm Optimization (PSO) Algorithm	90
4.2.2	Parallel Implementation of Central Force Optimization (CFO) Algorithm	97
4.2.3	Parallel Implementation of Gravitational Search Algorithm (GSA)	105
4.3	Parallel Implementation of Local Optimization Algorithms	108
4.4	Parallel Implementation of Hybrid Optimization Algorithms	109
<u>Chapter5 :</u>	<u>Results and Discussion</u>	<u>112</u>
5.1	Sequential Results	112
5.2	Parallel Results	117
5.2.1	CUDA Results.....	117
5.2.2	Open-MP Results.....	124
5.3	Comparative Results.....	133
5.4	Discussion and Summary	141
<u>Conclusions</u>		<u>143</u>
<u>Future Work.....</u>		<u>144</u>
<u>Appendix A.....</u>		<u>146</u>
<u>SOURCE CODE OF PSO USING CUDA</u>		<u>146</u>
<u>Appendix B</u>		<u>158</u>

SOURCE CODE OF CFO USING OPEN-MP.....	158
<i>Appendix C</i>	165
SOURCE CODE OF GSA USING OPEN-MP	165
<i>Appendix D</i>	173
<i>Works Cited</i>	175

LIST OF FIGURES

FIGURE 2.1: ANTENNA AS A TRANSITIONAL STRUCTURE BETWEEN FREE-SPACE AND A GUIDING DEVICE.....	19
FIGURE 2.2 : TRANSMISSION-LINE THEVENIN EQUIVALENT OF ANTENNA IN TRANSMITTING MODE.	20
FIGURE 2.3: RADIATION PATTERN OF AN ANTENNA	25
FIGURE 2.4: LINEAR ANTENNA ARRAY	28
FIGURE 2.5 : CIRCULAR ANTENNA ARRAY.....	30
FIGURE 2.6 : SWITCHED-BEAM SYSTEM [7].....	32
FIGURE 2.7 : COMPARISON OF (A) SWITCHED-BEAM SCHEME, AND (B) ADAPTIVE ARRAY SCHEME [7].	34
FIGURE 2.8 : FUNCTIONAL BLOCK DIAGRAM OF AN ADAPTIVE ARRAY SYSTEM [7].....	35
FIGURE 2.9: SMART ANTENNA	35
FIGURE 2.10 : SDMA SYSTEM BLOCK DIAGRAM [7].....	36
FIGURE 2.11 : GLOBAL AND LOCAL OPTIMA OF A TWO-DIMENSIONAL FUNCTION.	37
FIGURE 2.12: DIFFERENT POSSIBLE PROPERTIES OF FITNESS LANDSCAPES (MINIMIZATION).	41
FIGURE 2.13 : TRADITIONAL SYSTEM	48
FIGURE 2.14 : SHARED MEMORY SYSTEM	48
FIGURE 2.15 : DISTRIBUTED MEMORY SYSTEM.....	48
FIGURE 2.16 : DISTRIBUTED SHARED MEMORY SYSTEM	48
FIGURE 2.17 : TYPICAL PARALLEL COMPUTING EXECUTION	50
FIGURE 3.1 : THE BASIC CYCLE OF EVOLUTIONARY ALGORITHMS.	57
FIGURE 3.2 : PARTICLE SWARM OPTIMIZATION AS MODELLED BY A SWARM OF BEES SEARCHING FOR FLOWERS [9]	58
FIGURE 3.3 : FLOWCHART OF THE PROPOSED PSO ALGORITHM	65
FIGURE 3.4 : (A) ERRANT PROBE REPOSITION FACTOR RETRIEVAL. (B) ILLUSTRATION OF PROBE REPOSITIONING IN 2-D DS.	69
FIGURE 3.5 : FLOW CHART OF THE MAIN STEPS OF THE PROPOSED CFO ALGORITHM.....	74
FIGURE 3.6 : FLOW CHART OF GSA [23]	79
FIGURE 4.1: PARALLELIZATION POTENTIAL IN EVOLUTIONARY ALGORITHM.....	88
FIGURE 4.2 : A SEQUENTIALLY PROCEEDING EVOLUTIONARY ALGORITHM (SINGLE THREAD / LOCAL MACHINE).	89
FIGURE 4.3 : A PARALLEL EVOLUTIONARY ALGORITHM WITH TWO WORKER THREADS.	89
FIGURE 4.4 : PSO TIME DISTRIBUTION	90
FIGURE 4.5 : FLOW CHART OF THE PARALLEL PSO ALGORITHM.....	94
FIGURE 4.6 : CFO TIME DISTRIBUTION	98

FIGURE 4.7 : FLOW CHART OF THE PARALLEL CFO ALGORITHM	102
FIGURE 4.8 : GSA TIME DISTRIBUTION	106
FIGURE 4.9 : FLOW CHART OF THE PARALLEL GSA ALGORITHM	110
FIGURE 5.1 : SAMPLE RUNS AND AVERAGE SPEED UP =16.5.....	121
FIGURE 5.2 : SAMPLE RUNS AND AVERAGE SPEED UP =5.5.....	122
FIGURE 5.3 : FITNESS OF PSO, CFO, AND GSA ALGORITHMS WITH DIFFERENT PROP NUMBER.....	134
FIGURE 5.4 : TIME OF PSO, CFO, AND GSA ALGORITHMS WITH DIFFERENT PROP NUMBER.....	134

LIST OF TABLES

TABLE 3.1: RESULTS OF CFO DEVELOPMENT FOR FIVE TEST FUNCTION.....	72
TABLE 3.2: COMPARISON BETWEEN (PSO, CFO, AND GSA).....	81
TABLE 5.1: COMPARITIVE RESULTS OF BEAM-FORMING PROBLEM FOR THE THREE ALGORITHMS.....	113
TABLE 5.2: RESULTS OF BEAM-FORMING PROBLEM FOR THE THREE ALGORITHMS AFTER APPLYING NELDER MEAD ALGORITHM.	116
TABLE 5.3: SAMPLE RUNS AND THE AVERAGE SPEED UP FOR PSO USING GPU/CUDA.....	119
TABLE 5.4: EXPERIMENTAL RESULTS OF PSO USING CUDA.....	119
TABLE 5.5: EXPERIMENTAL RESULTS OF CFO USING CUDA	123
TABLE 5.6: SAMPLE RUNS OF SEQUENTIAL AND PARALLEL PSO USING OPEN-MP	125
TABLE 5.7: EXPERIMENTAL RESULTS OF PSO USING OPEN-MP.....	126
TABLE 5.8: SAMPLE RUNS OF SEQUENTIAL AND PARALLEL CFO USING OPEN-MP	128
TABLE 5.9: EXPERIMENTAL RESULTS OF CFO USING OPEN-MP.....	128
TABLE 5.10: SAMPLE RUNS OF SEQUENTIAL AND PARALLEL GSA USING OPEN-MP.....	131
TABLE 5.11: GSA OPEN-MP RESULTS	131
TABLE 5.12: TEST FUNCTIONS	135
TABLE 5.13: RESULTS OF THE THREE ALGORITHMS ALONG THE TEST FUNCTIONS.....	137
TABLE D.1 : NUMERICAL RESULTS OF THE THREE ALGORITHMS (PSO, CFO AND GSA) ALONG THE TEST FUNCTIONS.....	173

LIST OF ABBREVIATION

A

API (application programming interface)	52
array factor AF	27

B

bit error rate (BER)	31
----------------------------	----

C

Central Force Optimization (CFO)	4
central processing unit (CPU)	45
Compute Unified Device Architecture (CUDA)	4

D

decibel (dB)	118
decision space (DS)	67
Differential Evolution (DE)	20
direction-of-arrival (DOA)	33

E

Evolutionary algorithms (EAs)	56
-------------------------------------	----

G

Genetic Algorithms (GA)	20
global best (g_{best})	60
Graphics Processing Unit (GPU)	4
Gravitational Search Algorithm (GSA)	4

I

iteration best ($iter_{best}$)	62
--	----

L

Local Area Network (LAN)	23
--------------------------------	----

M

Message Passing Interface (MPI)	49
multi-objective evolutionary algorithm (MOEA)	56
multiple instruction stream – multiple data stream (MIMD)	46

N

Nelder-Mead (NM)	17
network of workstations (NOW)	48
Nondeterministic Polynomial time (NP)	39

O

Open Multi-Processing (Open-MP)	4
Open-MP Architecture Review Board (or Open-MP ARB)	53

P

Particle Swarm Optimization (PSO)	4
personal best (<i>pbest</i>)	59

R

Radio Frequency (RF)	32
----------------------------	----

S

side lobe level (SLL)	26
Signal not of Interest SNOIs	32
Signal of Interest (SOI)	32
single instruction stream – multiple data stream (SIMD)	46
single instruction stream–single data stream (SISD)	45
Spatial Division Multiple Access (SDMA)	34
symmetric multiprocessor (SMP)	47

INTRODUCTION

- 1.1 Motivation and Problem Statement**
 - 1.2 Research Objectives**
 - 1.3 Thesis Overview**
 - 1.4 Published Work**
-

CHAPTER 1

Introduction

1.1. Motivation and problem Statement

Smart antenna systems have been widely considered to provide interference reduction and improve the capacity, data rates, and performance of wireless mobile communication. Smart antenna arrays with adaptive beam-forming capability are very effective in the suppression of interference and multipath signals [1] [2]. The techniques of placing nulls in the antenna patterns to suppress interference and maximizing their gain in the directions of desired signals have received considerable attention in the past and are still of great interest using evolutionary algorithms such as genetic algorithm (GA) and particle swarm optimization (PSO) algorithm [3] [4].

For adaptive arrays using space division multiple access (SDMA), the optimal radiation pattern is synthesized using recent different optimization techniques such as Central Force Optimization (CFO), and Gravitational Search Algorithm (GSA). The algorithms are applied to a 24-element uniform circular array (UCA) to synthesize the array beam-pattern by optimizing the complex excitations values, amplitudes and phases of the adaptive array elements. The antenna elements consist of vertical (z-directed) half-wave dipole elements equally spaced in the x-y plane along a circular ring, where