

## TABLE OF CONTENTS

LIST OF FIGURES . . . . .	xi
ABSTRACT . . . . .	xv
Chapter	
1 INTRODUCTION . . . . .	1
1.1 Motivation . . . . .	1
1.1.1 The story of Mr. X . . . . .	1
1.2 Fully Dynamic Algorithms . . . . .	5
1.2.1 Interactive Design Processes . . . . .	5
1.2.2 Fundamental Concepts of Fully Dynamic Computation . . . . .	5
1.2.2.1 Off-line, On-line, and Fully Dynamic Algorithms . . . . .	8
1.2.3 Operations Supported by Fully Dynamic Algorithms . . . . .	11
1.2.4 Efficiency of Fully Dynamic Algorithms . . . . .	12
1.2.5 Interaction With the Outside World: Queries . . . . .	13
1.2.6 Efficiency of Queries: Being (Dis)honest . . . . .	15
1.3 Approximation Algorithms . . . . .	17
1.4 Putting the Concepts Together: Fully Dynamic Approximation Algorithms . . . . .	20
1.5 Existing Work on Fully Dynamic Algorithms . . . . .	21
1.5.1 Fully Dynamic Algorithms for Computational Geometry Problems . . . . .	21
1.5.2 Fully Dynamic Algorithms for Graph Problems . . . . .	23

1.5.3	Fully Dynamic Algorithms for Other Problems . . . . .	27
1.5.4	Important Tools . . . . .	28
1.6	Outline of the Dissertation . . . . .	29
<b>2</b>	<b>FULLY DYNAMIC VERTEX COVER . . . . .</b>	<b>30</b>
2.1	Introduction . . . . .	30
2.1.1	Vertex Cover – Existing Results . . . . .	31
2.1.2	Fully Dynamic Algorithms for Vertex Cover . . . . .	32
2.1.2.1	Fully Dynamic Vertex Cover . . . . .	32
2.1.2.2	Competitive Fully Dynamic Approximation Algorithms for Vertex Cover . . . . .	33
2.2	A Simple Algorithm for Fully Dynamic Vertex Cover . . . . .	34
2.3	An Improved, Amortized Algorithm $A_1$ . . . . .	39
2.3.1	What is Clean and What is Dirty? . . . . .	39
2.3.1.1	The Dirty Strategy . . . . .	40
2.3.1.2	The Clean Strategy . . . . .	41
2.3.2	Incorporating Both Strategies – Algorithm $B_1$ . . . . .	42
2.3.2.1	The Tests . . . . .	42
2.3.2.2	Executing Operations in an On-line Fashion . . . . .	43
2.3.2.3	The “Clean-up” . . . . .	51
2.3.2.4	Implementation Details . . . . .	53
2.3.2.5	The Running Time of $B_1$ . . . . .	56
2.3.3	A Slight Improvement – Algorithm $A_1$ . . . . .	56
2.4	Further Improvements – Algorithms $A_k$ . . . . .	59
2.5	Conclusion . . . . .	63

3	FULLY DYNAMIC BIN PACKING . . . . .	65
3.1	Introduction . . . . .	65
3.1.1	Bin packing – Existing Results . . . . .	66
3.1.2	Operations and Queries . . . . .	67
3.1.3	What’s to Come . . . . .	68
3.2	Towards Full Dynamization of Bin Packing . . . . .	69
3.2.1	Some Definitions . . . . .	70
3.2.2	A Grouping Rule Algorithm . . . . .	72
3.2.3	A Fully Dynamic Algorithm <i>MP</i> (Myopic Packing) . . . . .	73
3.2.3.1	The Thoroughness Property . . . . .	74
3.2.3.2	Thoroughness Suffices for the Competitive Ratio of $\frac{4}{3}$ . . . . .	77
3.2.3.3	The Algorithm <i>MP</i> . . . . .	80
3.3	Mostly Myopic Packing ( <i>MMP</i> ) . . . . .	87
3.3.1	More Definitions . . . . .	89
3.3.2	LLS–maximality, and M–thoroughness . . . . .	92
3.3.3	A Sketch of <i>Insert</i> and <i>Delete</i> . . . . .	97
3.3.4	The Data Structure of <i>MMP</i> . . . . .	99
3.3.4.1	The Regular Packing . . . . .	101
3.3.4.2	Inferior Trees . . . . .	103
3.3.4.3	The Auxiliary Storage ( <i>Aux</i> ) . . . . .	104
3.3.4.4	Details of the Data Structure of M–items/bundles . . . . .	105
3.3.4.5	Maintaining the List of Items <i>L</i> . . . . .	110
3.3.5	Details of <i>MMP</i> . . . . .	112
3.3.5.1	Top Level Description of <i>MMP</i> . . . . .	113
3.3.5.2	Details of <i>clear_Aux</i> . . . . .	117

3.3.5.3	Implementation of Queries . . . . .	119
3.3.6	<i>MMP</i> is $\frac{5}{4}$ -competitive . . . . .	122
3.3.6.1	Overview of the Proof of the Upper Bound on the Competitive Ratio of <i>MMP</i> . . . . .	124
3.3.6.2	<i>MMP</i> Maintains LLS-maximality . . . . .	125
3.3.6.3	<i>MMP</i> Maintains <i>M</i> -thoroughness . . . . .	126
3.3.6.4	Consideration of Lists with no <i>M</i> -items . . . . .	127
3.3.6.5	Consideration of Arbitrary Lists . . . . .	161
3.3.6.6	Is Bundling Necessary? . . . . .	162
3.3.7	The Running Time of <i>MMP</i> is $\Theta(\log n)$ . . . . .	166
3.4	Inserts Only . . . . .	169
3.4.1	Uniform Algorithms for <i>Inserts</i> . . . . .	169
3.4.2	Amortized Algorithms for <i>Inserts</i> . . . . .	169
3.5	Conclusion . . . . .	177
4	<b>LINEAR TIME BIN PACKING</b> . . . . .	<b>179</b>
4.1	Introduction . . . . .	179
4.1.1	Linear Time Bin Packing – Some Existing Results . . . . .	180
4.1.2	What's to Come . . . . .	182
4.2	Review of <i>GXFG</i> . . . . .	182
4.2.1	Group- <i>X</i> Fit Rule . . . . .	184
4.3	The Main Result – <i>LINBP</i> . . . . .	185
4.3.1	The Central Idea . . . . .	186
4.3.2	The Algorithm <i>LINBP</i> . . . . .	186
4.3.2.1	Preliminaries . . . . .	186
4.3.2.2	The Description of <i>LINBP</i> . . . . .	188
4.3.2.3	The Running Time and the Competitive Ratio of <i>LINBP</i> . . . . .	192

4.3.2.4	Practicality of <i>LINBP</i> . . . . .	205
4.4	Bounded Bin Packing . . . . .	206
4.4.1	Review of the Relevant Features of <i>LINBP</i> . . . . .	207
4.4.2	<i>LINBP</i> Can Do Better Than $\frac{4}{3}$ for Bounded Bin Packing . . . . .	208
4.4.3	The Main Idea Behind $A_\epsilon$ . . . . .	209
4.5	Parallel <i>LINBP</i> . . . . .	213
4.5.1	The Algorithm . . . . .	214
4.6	Conclusion . . . . .	217
5	CONCLUSION . . . . .	219
5.1	Defining Fully Dynamic Approximation Algorithms . . . . .	219
5.2	Fully Dynamic Vertex Cover . . . . .	220
5.2.1	Open Problems . . . . .	221
5.3	Fully Dynamic Bin Packing . . . . .	221
5.3.1	Open Problems . . . . .	222
5.4	Linear Time Bin Packing . . . . .	222
5.4.1	Open Problems . . . . .	223
5.5	Summary of Contributions . . . . .	223
5.6	Future Work . . . . .	224
	BIBLIOGRAPHY . . . . .	225