

# Self-Reduction for Combinatorial Optimisation

Nicholas Paul Sheppard

## Abstract

This thesis presents and develops a theory of *self-reduction*. This process is used to map instances of combinatorial optimisation problems onto smaller, more easily solvable instances in such a way that a solution of the former can be readily re-constructed, without loss of information or quality, from a solution of the latter. Self-reduction rules are surveyed for the Graph Colouring Problem, the Maximum Clique Problem, the Steiner Problem in Graphs, the Bin Packing Problem and the Set Covering Problem.

This thesis introduces the problem of determining the maximum sequence of self-reductions on a given structure, and shows how the theory of *confluence* can be adapted from term re-writing to solve this problem by identifying rule sets for which all maximal reduction sequences are equivalent. Such confluence results are given for a number of reduction rules on problems on discrete systems. In contrast, NP-hardness results are also presented for some reduction rules.

A probabilistic analysis of self-reductions on graphs is performed, showing that the expected number of self-reductions on a graph tends to zero as the order of the graph tends to infinity.

An empirical study is performed comparing the performance of self-reduction, graph decomposition and direct methods of solving the Graph Colouring and Set Covering Problems. The results show that self-reduction is a potentially valuable, but sometimes erratic, method of finding exact solutions to combinatorial problems.

## Acknowledgements

The author would like to thank his thesis supervisor, Dr. Jeffrey Kingston, for his assistance and guidance throughout the course of this work. Several of the papers on which this thesis is based were developed in collaboration with Dr. Kingston [39, 40, 42].

The author would also like to acknowledge the support of an Australian Postgraduate Award, without which this work could not have been completed.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Combinatorial Optimisation . . . . .	1
1.2	Decomposition and Self-Reduction . . . . .	2
1.2.1	Decomposition . . . . .	2
1.2.2	Self-Reduction . . . . .	3
1.3	This Thesis . . . . .	4
<b>2</b>	<b>Self-Reduction</b>	<b>5</b>
2.1	Basic Definitions . . . . .	5
2.2	Confluence . . . . .	6
2.3	Self-Reduction and Combinatorial Optimisation . . . . .	8
2.4	Decomposition . . . . .	9
2.4.1	The Simplicial Decomposition . . . . .	9
2.4.2	The Substitution Decomposition . . . . .	10
<b>3</b>	<b>Problems on Vertex-Weighted Graphs</b>	<b>13</b>
3.1	Introduction . . . . .	13
3.1.1	The Graph Colouring Problem . . . . .	14
3.1.2	The Maximum Clique Problem . . . . .	15
3.1.3	The Minimum Dominating Set Problem . . . . .	16
3.2	Self-Reductions . . . . .	16
3.2.1	Autonomous Reduction (A) . . . . .	16

3.2.2	Subset Reduction (S)	17
3.2.3	Under-constrained Deletion (UC)	18
3.2.4	Even-pair Reduction (E)	19
3.2.5	Quasi-autonomous clique reduction ( $\hat{C}$ )	20
3.3	Confluence	21
3.4	Maximum Reduction Sequences	25
3.4.1	Two-Pair Reduction Sequences	28
3.4.2	Quasi-Autonomous Clique Sequences	30
3.4.3	Sequences of Other Reductions	36
<b>4</b>	<b>The Steiner Tree Problem</b>	<b>39</b>
4.1	Introduction	39
4.1.1	Definitions	40
4.2	Self-Reductions	41
4.2.1	Reachability Reduction (RT)	41
4.2.2	Cut Reachability Reduction (CRT)	42
4.2.3	Nearest Special or Chord Vertices Reduction (NSC)	43
4.2.4	Smaller Special Distance Reduction (SD)	43
4.2.5	Local Steiner Cost Reduction (LSC)	44
4.2.6	Local Bottleneck Distance Reduction (LBD)	45
4.2.7	Change Edge-Costs Reduction (CEC)	46
4.3	Maintaining the Upper Bound	46
4.4	Confluence	47
<b>5</b>	<b>Problems on Set Systems</b>	<b>55</b>
5.1	Introduction	55
5.1.1	The Bin Packing Problem	56
5.1.2	The Set Covering Problem	56
5.2	Self-Reductions	57
5.2.1	Dominant Feasible Set Reduction (DF)	57
5.2.2	Dominated Column Reduction (DC)	59

<i>CONTENTS</i>	vii
5.2.3 Dominated Row Reduction (DR) . . . . .	60
5.2.4 Disconnected System Reduction (DS) . . . . .	60
5.2.5 Row Difference Reduction (RD) . . . . .	61
5.2.6 Reduced Costs Reduction (RC) . . . . .	61
5.3 Confluence . . . . .	61
<b>6 Self-Reduction and Probability</b>	<b>65</b>
6.1 Introduction . . . . .	65
6.2 Vertex-Weighted Graphs . . . . .	66
6.3 Other Structures . . . . .	71
<b>7 Computational Experience</b>	<b>73</b>
7.1 Introduction . . . . .	73
7.2 Experimental Parameters . . . . .	74
7.2.1 Graph Colouring . . . . .	74
7.2.2 Set Covering . . . . .	75
7.3 Results . . . . .	78
7.3.1 Graph Colouring . . . . .	78
7.3.2 Set Covering . . . . .	88
7.4 Conclusion . . . . .	89
<b>8 Conclusion</b>	<b>91</b>
8.1 Suggestions for Future Work . . . . .	91
8.1.1 $\mathcal{A}$ -Extensions . . . . .	92
8.1.2 Effects of Self-Reduction on Direct Algorithms . . . . .	92
8.1.3 Miscellaneous Issues . . . . .	93
<b>A Notation</b>	<b>95</b>

