

**Solution Biases and
Pheromone Representation Selection
in Ant Colony Optimisation**

Erin James Montgomery
BInfTech (Hons)

A dissertation submitted in fulfilment of the requirements of the degree of
Doctor of Philosophy for the School of Information Technology,
Bond University

14 June 2005

Statement of Originality

The material in this thesis has not been previously submitted for a degree or diploma in any university. To the best of my knowledge this thesis contains no material previously published or written by another person except where due acknowledgement is made in the thesis itself.

E. James Montgomery

Date:

Summary

Combinatorial optimisation problems (COPs) pervade human society: scheduling, design, layout, distribution, timetabling, resource allocation and project management all feature problems where the solution is some combination of elements, the overall value of which needs to be either maximised or minimised (i.e., optimised), typically subject to a number of constraints. Thus, techniques to efficiently solve such problems are an important area of research. A popular group of optimisation algorithms are the metaheuristics, approaches that specify how to search the space of solutions in a problem independent way so that high quality solutions are likely to result in a reasonable amount of computational time. Although metaheuristic algorithms are specified in a problem independent manner, they must be tailored to suit each particular problem to which they are applied. This thesis investigates a number of aspects of the application of the relatively new Ant Colony Optimisation (ACO) metaheuristic to different COPs.

The standard ACO metaheuristic is a constructive algorithm loosely based on the foraging behaviour of ant colonies, which are able to find the shortest path to a food source by indirect communication through pheromones. ACO's artificial pheromone represents a model of the solution components that its artificial ants use to construct solutions. Developing an appropriate *pheromone representation* is a key aspect of the application of ACO to a problem. Since its inception in the early 1990s, ACO has been applied to an increasing range of problems, leading to the development of a range of pheromone representations (Dorigo and Stützle, 2002). However, pheromone representations have typically been chosen in an *ad hoc* fashion, either by selecting a representation used previously for a similar problem or by developing a new representation that appears intuitively to suit the problem in question. The absence of a systematic approach to adapting ACO to different COPs has resulted in inconsistent performance across different problems.

An examination of existing ACO applications and the constructive approach more generally reveals how the metaheuristic can be applied more systematically across a range of COPs. The two main issues addressed in this thesis are biases inherent in the constructive process and the systematic selection of pheromone representations.

All constructive metaheuristics explore a tree of constructive decisions (a construction tree) the shape of which is determined by problem constraints and the chosen method of building solutions. The shape of this tree, combined with the mapping from paths in the tree to solutions, can bias a constructive search. This is particularly a problem in COPs where infeasible solutions cannot be avoided—such infeasible solutions have an elevated probability of being discovered by a constructive algorithm. This situation is common in COPs in which a number of entities (e.g., tasks, projects, or individuals) compete for limited resources. Alternative techniques for determining the order in which demands for

resources are considered can improve the probability of reaching feasible solutions in these types of COP. A number of these techniques are investigated, revealing that commonly used static and dynamic heuristics for this task generally work well, although cannot guarantee a good probability of finding feasible solutions across all problem instances.

The effectiveness of a given pheromone representation is related to how it maps onto the construction tree for a problem. Considering this finding, a systematic approach is developed for the selection of an appropriate pheromone representation based on characteristics of the COP being solved. The comparative performance of ACO using alternative pheromone representations—those used in the literature, suggested by the new system, or potentially applicable—is investigated for six problem types, including the travelling salesman, multiple knapsack, quadratic assignment, generalised assignment, shop scheduling and car sequencing problems. Results confirm that the algorithm’s suggested pheromone representations generally perform well, with two main exceptions. First, an intuitively inappropriate pheromone representation previously unused with the knapsack problem was found to perform best on that problem, a finding that will be the subject of further investigation. Second, results for the car sequencing problem suggest that using the simplest pheromone representation that models solution identity (rather than structure) may be better than attempting to model the full complexity of solution characteristics that contribute to solution cost.

The systematisation of ACO should lead to more consistently high performance of the algorithm across different problems. Additionally, it supports the creation of a generalised ACO system, capable of adapting itself to suit many different combinatorial problems without the need for manual intervention.

Acknowledgements

No work this large could be produced without the help, hindrance or influence of a large number of people and organisations. A sample of them get a mention here.

Professionally, I would like to extend my thanks to

Marcus Randall, for his guidance, ideas, encouragement, prodding, persuading, understanding, assistance during tough times and even for his confident, if unsettling, assertion in recent times, “Now, you are the expert.” I’m glad to call you my friend as well as my supervisor.

Tim Hendtlass, whose experience in supervision and many suggestions regarding the process of producing a thesis have helped immensely. Thanks also to his cohorts at the Centre for Intelligent Systems and Complex Processes at Swinburne University.

School of Information Technology, Bond University, for providing me with a scholarship to undertake my doctorate, for offering me the chance to teach some of my favourite undergraduate subjects, for the many friendly and helpful support staff, and for providing the most cohesive (yes, it is true) group of colleagues in the university.

The Australian Government, for providing me with an Australian Postgraduate Award during my PhD candidature.

On a personal level, I would like to thank

My officemates, past and present: *Dr Sun Zhaohao*, for his boundless enthusiasm and efforts to teach me Chinese; *Kieth Hales*, for putting up with the antics of us young folk; *Henry Larkin*, for having a sense of humour so strange it could be my own; and the younger *James Larkin*, for encouraging my juggling skills at every opportunity.

Bond Ultimate Frisbee, for providing many energetic and fun times and some of the best friends I could ever hope to meet, in particular business PhD candidate *Lisa Watson* and my IT colleague *Dr Phil Stocks*.

Friends too numerable to mention, for all the good times that helped balance my life and even for some of the bad (because life’s like that).

Drs Bob Montgomery and Laurel Morris (Dad and Mum), for always having my best interests at heart and for frequently reminding me that there’s more to life than working on one’s thesis. The travel we’ve shared has been equally rewarding as my research. In particular, it was with their encouragement that early in my candidature I began my amateur underwater photography career, something which has brought me more joy than a whole bag of doctorates could ever do. And to all the unwilling aquatic victims of my camera: *So long, and thanks for all the snaps*.

Contents

1	Introduction	1
1.1	Combinatorial Optimisation	1
1.2	Evolutionary Computing and Collective Intelligence	3
1.3	Optimisation Inspired by Ant Colony Behaviour	5
1.4	Scope and Aim	8
1.5	Thesis Outline	8
2	ACO Algorithms and Applications	10
2.1	Constructive Heuristics and Metaheuristics	10
2.1.1	Sequence Space and Solution Space	12
2.1.2	Randomness and Greed in Constructive Metaheuristics	15
2.2	The ACO Metaheuristic	16
2.3	ACO Algorithms	18
2.3.1	Ant System	18
2.3.2	Ant-Q and Ant Colony System	20
2.3.3	<i>MAX</i> – <i>MIN</i> Ant System	21
2.3.4	Rank-based Ant System	22
2.3.5	Hyper-cube Framework for ACO	22
2.3.6	Other Variants	23
2.4	ACO Applications	24
2.4.1	Subset Problems	24
2.4.2	Permutation/Routing Problems	27
2.4.3	Scheduling Problems	29
2.4.4	Assignment Type Problems	33
2.4.5	Constraint Satisfaction Problems	36
2.4.6	Other Problems	37
2.5	Formalisations of ACO	41
2.5.1	Graph-based Ant System	41
2.5.2	Ant Programming	42
2.6	Divergence of Construction Graph and Pheromone Representation	43
2.6.1	Constructive Heuristics and Virtual Construction Graphs	45
2.7	Local Search in ACO	46
2.8	Summary	47

3	Bias in Constructive Heuristics	49
3.1	Bias Inherent in the Constructive Process	50
3.2	Variable Length and Infeasible Solutions	52
3.3	Assignment Problems and Assignment Order	55
3.3.1	Using Assignment Order to Reduce Infeasible Space	58
3.4	Examples of Bias in Constructive Algorithms	67
3.4.1	Solving the JSP, GSP or OSP as Assignment Problems	69
3.4.2	Solving Assignment Problems with No Assignment Order	70
3.5	Bias Effects and Problem Size	71
3.6	Deliberately Introduced Sources of Bias	72
3.7	Summary	74
4	Pheromone Representations	76
4.1	Formalisation of Pheromone Representations	76
4.1.1	Representation-Oriented and Identity-Oriented Pheromones	80
4.1.2	Atypical Pheromone Representations	81
4.2	Using Higher Order Pheromone Representations	84
4.2.1	Examples	86
4.2.2	Notes on Higher Order Pheromone Representation Usage	87
4.3	Summary	90
5	Interaction Between Constructed Solution Biases and Pheromone	92
5.1	Mechanisms of Interaction	92
5.1.1	Low Level Interactions	93
5.1.2	High Level Interactions and Competition-Balanced Systems	95
5.2	Case Study: Pheromone Representations for the JSP, GSP and OSP	100
5.3	Pheromone and Bias Interactions in Other Problems	108
5.3.1	MKP	109
5.3.2	GAP	110
5.3.3	TSP and QAP	110
5.4	Summary	113
6	Selecting Pheromone Representations Considering Bias	115
6.1	Controlling Bias Using the Pheromone Update	116
6.2	Unique Representation in Pheromone	117
6.2.1	Unique Representation and Bias	119
6.2.2	Unique Representation in Higher Order Pheromones	120
6.2.3	Examples of Unique and Multiple Representation	120
6.2.4	Shared Representations in Pheromone	123
6.3	Systematically Determining Appropriate Pheromone Representations	125
6.3.1	Case Study: Car Sequencing Problem	128
6.3.2	A Decision Algorithm for Pheromone Selection	129
6.4	Potential to Counteract Bias	133
6.5	Summary	134

7	Pheromone Representation and Assignment Order Comparative Performance	136
7.1	Methodology	136
7.1.1	Problems and Pheromone Representations	137
7.1.2	Implementation of ACO Algorithms for Studied COPs	138
7.1.3	Variables and Analysis Techniques Used	140
7.2	Results	142
7.2.1	TSP	142
7.2.2	MKP	143
7.2.3	GSP	146
7.2.4	QAP	147
7.2.5	GAP	150
7.2.6	CSeqP	155
7.3	Summary	159
8	Conclusions and Further Work	163
8.1	Summary	163
8.2	Contributions	166
8.3	Conclusions and Future Work	168
	Bibliography	170
A	Glossary of Problem Names and Acronyms	184
B	Objective Functions Used in Pheromone Representation Selection	190
C	Details of Non-benchmark Problem Instances Used	198
D	Tables of Results	200
E	Publications Arising from this Study	229
F	Notation Used in Thesis	230

List of Tables

1.1	A continuum of collective intelligence approaches	5
2.1	Sample of ACO applications	25
3.1	Feasible probability under different assignment orders for gap1 instances .	66
3.2	Feasible probability under different assignment orders for gap2 instances .	66
3.3	Pairwise comparison of assignment orders for gap1 instances	68
3.4	Pairwise comparison of assignment orders for gap2 instances	68
3.5	Examples of bias in constructive algorithms	70
3.6	Bias effects and instance size	72
4.1	Common pheromone representations	80
4.2	Sample of ways to customise generic higher order pheromone equation . . .	88
4.3	Pheromone representations used in the literature and potentially applicable	91
6.1	Suggested pheromone representations for common COPs	135
7.1	Problems and pheromone representations studied	137
7.2	Parameter values used in ACO algorithm implementations	139
7.3	Values of β used with each problem	139
7.4	Heuristic information used by ACO and ACO _{undir} algorithms	139
7.5	Example comparison table	141
7.6	TSP instances	142
7.7	TSP pheromone representation comparisons	143
7.8	MKP instances	144
7.9	MKP pheromone representation comparisons	145
7.10	GSP instances	146
7.11	JSP, GSP & OSP pheromone representation comparisons	147
7.12	OSP pheromone representation comparisons	147
7.13	QAP instances	148
7.14	QAP pheromone representation comparisons	149
7.15	GAP instances	150
7.16	GAP pheromone representation comparisons	152
7.17	GAP assignment order comparisons	154
7.18	CSeqP instances	156
7.19	CSeqP pheromone representation comparisons	158
D.1	TSP results, no local search	201
D.2	TSP results, with local search	202

D.3	MKP results	203
D.4	GSP results	206
D.5	QAP results	208
D.6	GAP results	212
D.7	CSeqP results	222
F.1	Notation used in thesis	230

List of Figures

2.1	Constructive hypercube	13
2.2	Construction tree for a small subset problem	14
2.3	Solution space for a small subset problem	15
2.4	Construction graph for a small subset problem	17
2.5	A small JSP instance	31
2.6	Alternative assignment type problem construction graphs	34
2.7	Constraint satisfaction problem alternative construction graph	37
3.1	A small JSP instance: <code>jsp2-2</code>	51
3.2	Construction tree for small example JSP instance	51
3.3	Solution probability for 15 and 50 item MKP instances	54
3.4	Construction trees for small GAP instance with different assignment orders	56
3.5	Distribution of feasible solution probabilities for <code>gap2</code> instances	59
3.6	Number of infeasible sequences against $P(\text{feasible})$ in <code>gap2-1</code>	60
3.7	Distribution of path lengths for first <code>gap2</code> instance under different assign- ment orders	63
3.8	Construction tree for <code>jsp2-2</code> using alternative construction approach . . .	71
5.1	Full details of <code>jsp2-2</code> instance	94
5.2	Construction tree for <code>jsp2-2</code>	94
5.3	Decisions affected by sequence $\langle 1, 2, 3, 4 \rangle$ in <code>jsp2-2</code>	96
5.4	Decisions affected by sequence $\langle 1, 3, 4, 2 \rangle$ in <code>jsp2-2</code>	97
5.5	Details of <code>jsp3-3</code> and <code>osp3-3</code> instances	101
5.6	Line scheduling factor against various measures for <code>jsp3-3</code> and <code>osp3-3</code> . .	103
5.7	Line scheduling factor against solution cost using ACO	105
5.8	Line scheduling factor against sequence probability using ACO	105
5.9	Solution characteristic usage frequency against corresponding solution cost	107
5.10	Potential bias interaction in the <code>mknab1-15item</code> MKP instance	111
5.11	Potential bias interaction in the <code>gap2-1</code> GAP instance	112
6.1	Use of TSP-like pheromone with a small subset problem	118
6.2	Small example OSP instance	123
6.3	Decision tree for selection of pheromone representations	131