

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

MODULAR LOCAL SEARCH:  
A FRAMEWORK FOR SELF-ADAPTIVE METAHEURISTICS

A THESIS PRESENTED IN PARTIAL FULFILMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY  
IN DECISION SCIENCE AT  
MASSEY UNIVERSITY

David Colin Woods

2010



---

# Abstract

This research develops Modular Local Search (MLS), a framework such that trajectory-based metaheuristics can be expressed as subsets of “modules” from a common library, with a common structure. The standardized modules and structure allow the easy formulation of common metaheuristic paradigms, as well as the easy creation of relatively complex hybrids by simply listing the modules that should be included. A new markup language called Modular Local Search Markup Language (MLSML) is developed so that new metaheuristics can be implemented *declaratively*, rather than *programmatically*.

Some advanced ideas are introduced and explored, whereby metaheuristics are able to modify themselves during their execution, by varying parameters and swapping modules into and out of activation. This ability introduces the potential for semi-intelligent algorithms that are capable of a type of learning. Several demonstration methods are developed and these show promise on a small test set of problem instances.

A new combinatorial optimization problem is developed to serve as the testing ground for the new heuristic ideas. The Arc Subset Routing Problem (ASRP) involves routing a vehicle on a graph, choosing a subset of the arcs, such that the reward collected by traversing these arcs is maximised subject to a constraint on the total distance travelled. This problem is first formulated and explored as a traditional Operations Research investigation; construction heuristics are developed, as well as some improvement routines for local search, and computational tournaments are performed to compare the methods.

Some attention is given to developing methods to predict which of two heuristics is most suited to a given problem instance, based on an analysis of the characteristics of that problem. Initial results demonstrate the potential of such an approach.

The MLS framework offers a powerful and flexible structure both for the easy and consistent implementation of existing metaheuristics, and also as a platform for the development of new, advanced metaheuristic ideas. Early results are encouraging, and a number of directions for future research are discussed, including some complex real-world problems for which the self-adaptive capabilities of MLS would be especially useful.



---

# Acknowledgements

I would like to thank all the many people who have supported and encouraged me over the years. It has been a long journey, including several complete changes in direction, and the effort to complete this work has been made easier by their understanding.

First and foremost I need to express my gratitude to my supervisors, Mark Bebbington and John Giffin. Mark, who took over as chief supervisor after John moved to Canterbury, has served as a constant reality check and I appreciate his resisting the urge, overwhelming at times I'm sure, to wash his hands of me as the pressures of developing a career and a family meant that my progress was, at times, intermittent. Extra special thanks are due to John, who served as my mentor and friend during my undergraduate years. Many hours were spent in his office discussing the world and Operations Research, and many ideas were discussed, including the germs of what later became this research, although via a circuitous route. Since then he has provided much timely advice, and encouragement to PhD thinking, which has served, barely, as a restraint on my own tendency to bite off more than I can chew. His critical eye has also prevented many potentially embarrassing typos and misspellings, although of course I take responsibility for any mistakes in the final thesis, mindful of the words of Randy Milholland, who said that *"typos are very important to all written form. It gives the reader something to look for so they aren't distracted by the total lack of content in your writing"*. I should also acknowledge a former PhD student of John's, Mark Johnston, the formatting and layout of whose thesis I shamelessly copied.

Continuing to work on this research over these many years would not have been possible without the support of my boss, Graeme Gee. As well as providing me the opportunity to develop my career in analytics consulting, which has allowed me to gain a hands-on appreciation for techniques that work in the real world, and the complexities of "real" optimization problems, he has been unfailingly supportive. This support has extended to dedicated periods of time where I could focus on my PhD research, financially sponsoring my study, including fees and any text books I decided I must have, and a steady stream of computing resources. In the last stages of writing up this thesis he even drove me to a meeting at Massey because I hadn't been getting enough sleep to drive safely. His unwavering support has made the completion of this work possible.

Last, but certainly not least, I need to thank my family and friends. My friends, who have offered many opportunities to escape for a time from thinking about anything related to Operations Research, in fact probably too many opportunities. My family, who have always had faith that I would finish; one might say blind faith, but that is what families are for. Especial thanks to my Nana, who has been patiently looking forward to my finishing for more years than I care to count, and to my father, who never fails to nag me about it. My biggest appreciation is reserved for my partner Vin; she has been heroically patient and supportive, especially over the last few months of the write-up while I have been absent in order to devote myself to it. She undertook, mostly without complaint, essentially to act as a single mother to our daughter Tui while I industriously finished this thesis. Thanks are also due to Tui; the burning desire to get back to see her motivated more late nights and early mornings than are really healthy – she finally provided my inspiration to finish, regardless of how “finished” I feel.

---

# Table of Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Table of Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xiii</b>
<b>List of Algorithms</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Local search and metaheuristics.....	1
1.2 Trajectory-based metaheuristics.....	3
1.3 Introducing Modular Local Search.....	3
1.4 Research overview .....	5
1.5 Research questions and goals.....	7
1.6 Other frameworks.....	7
1.7 Thesis structure .....	10
<b>Part I The Arc Subset Routing Problem</b>	<b>13</b>
<b>2 Arc Routing Literature Review</b>	<b>17</b>
2.1 Vehicle routing problems in general .....	17
2.2 Arc routing problems .....	18
2.3 Subset routing problems.....	24



<b>3 Preliminary Investigation of the ASRP</b>	<b>33</b>
3.1 Formulation of the ASRP .....	33
3.2 Construction heuristics .....	42
3.3 Improvement procedures .....	48
3.4 Problem generation principles .....	52
3.5 Specific problem instances .....	58
3.6 Preliminary experimentation .....	61
3.7 Phase 1 experimentation.....	64
3.8 Phase 2 experimentation.....	68
 <b>Part II MLS Foundations</b>	 <b>83</b>
<b>4 Modular Local Search</b>	<b>87</b>
4.1 Introduction .....	87
4.2 Structure of MLS .....	88
4.3 The search scheme.....	89
4.4 The control system .....	97
4.5 The memory structures .....	105
4.6 Summary of MLS components.....	108
4.7 Examples of metaheuristics as MLS .....	109
4.8 Discussion.....	117
 <b>5 Metaheuristic Concepts</b>	 <b>123</b>
5.1 Ascent Search .....	123
5.2 Iterated Search .....	124
5.3 Thresholding.....	137
5.4 Adaptive Memory and Tabu Search.....	145
5.5 Other trajectory methods .....	153
 <b>Part III Experimentation and Analysis</b>	 <b>157</b>
<b>6 Applying MLS to the ASRP</b>	<b>161</b>
6.1 Introduction .....	161
6.2 Problem instance design .....	162
6.3 MLS metaheuristics.....	167
6.4 Experimentation and analysis.....	181
6.5 Hybrids .....	203
6.6 Discussion.....	212
 <b>7 Heuristic Problem Design</b>	 <b>215</b>
7.1 Introduction .....	215
7.2 New problem features.....	216
7.3 The Maximally Diverse Subset Selection Problem.....	218

7.4	A tiny illustrative problem .....	219
7.5	Measures of distance .....	219
7.6	Heuristics.....	223
7.7	Solving the tiny problem .....	224
7.8	Additional measures of diversity.....	225
7.9	A giant selection problem .....	228
7.10	Solving the giant problem .....	229
7.11	Using MLS to design problem instances.....	237
<b>8</b>	<b>Advanced MLS Applications</b>	<b>247</b>
8.1	Introduction .....	247
8.2	Using MLS to design MLS heuristics .....	250
8.3	Adaptive Diversification Local Search .....	267
<b>9</b>	<b>Conclusions and Recommendations for Future Research</b>	<b>277</b>
9.1	Overview of research .....	277
9.2	Experimental design.....	281
9.3	Contributions and implications .....	282
9.4	Further research directions .....	286
	<b>Appendices</b>	<b>301</b>
	<b>A Glossary of MLS Terms</b>	<b>303</b>
	<b>B Programmatic Structure</b>	<b>309</b>
B.1	Introduction .....	309
B.2	Object-oriented programming structure .....	310
B.3	Extension to new problem domains .....	318
	<b>C Modular Local Search Markup Language (MLSML)</b>	<b>321</b>
C.1	Structure of MLSML.....	321
C.2	Examples of MLSML specifications.....	324
	<b>D Discussion of Possible Extensions to the ASRP</b>	<b>339</b>
D.1	Introduction .....	339
D.2	Variations on the Basic ASRP .....	340
D.3	Reward structures.....	342
D.4	Service variation.....	348
D.5	Competition.....	350
	<b>E Bibliography</b>	<b>355</b>



---

## List of Figures

2.1	Example of a graph that cannot be made Eulerian .....	20
2.2	Example illustrating the difference between <i>service</i> and <i>traversal</i> .....	22
3.1	Examples of cycles .....	45
3.2	Example of nested cycles .....	45
3.3	Basic ASRP move-types .....	49
3.4	Example of a 6x6 grid graph .....	52
3.5	Example of a grid graph.....	54
3.6	Graph partition .....	56
3.7	Grids generated using GRID GROW .....	57
3.8	Grids generated using GRID SELECT .....	57
3.9	Grids which were misclassified.....	57
3.10	Designed problem instances .....	59
3.11	Examples of random grid graphs.....	59
3.12	Ratio of reward to computation time (efficiency) .....	63
3.13	Results of constructive heuristics on complete grids with increasing budget .....	66
3.14	Approximate divisions of heuristic performance for unimproved heuristics .....	68
4.1	The search iteration process .....	90
4.2	Relationships of solutions and moves in the MLS search iteration process.....	91
4.3	The MLS control system .....	97
4.4	Partial solution hierarchy .....	117
5.1	Performance of initial methods on benchmark instances .....	127
5.2	Performance of initial methods on random instances.....	127
5.3	Pictorial representation of the perturbation step for iterated local search. ....	128

5.4	Performance of perturbation strengths on benchmark instances .....	129
5.5	Performance of perturbation strengths on random instances .....	129
6.1	Sum of reward collected by heuristic for 246 instance overlap set .....	183
6.2	Score means for Steepest Ascent heuristics with 95% confidence bars .....	184
6.3	Total reward for Steepest Ascent.....	185
6.4	Mean score for Steepest Ascent.....	185
6.5	Total reward for Simulated Annealing .....	190
6.6	Mean score for Simulated Annealing .....	190
6.7	Mean score by “temp” .....	191
6.8	Mean score for each problem instance by “temp” .....	192
6.9	Mean score by “rate” .....	193
6.10	Mean score by “iterations” .....	193
6.11	Mean score by “threshold” .....	193
6.12	Proportion of problem instances correct for each heuristic pair .....	197
6.13	Total reward for Tabu Search.....	198
6.14	Mean score for Tabu Search.....	198
6.15	Total reward for VNS.....	201
6.16	Mean score for VNS.....	201
6.17	Sum of reward collected by heuristic for 246 instance overlap set, including hybrids .....	209
6.18	Scatter plot of scores: TS-VNS1 vs Tabu10.....	211
6.19	Scatter plot of scores: TS-VNS3 vs Tabu10.....	211
7.1	Distribution of normalised Euclidean distances for the tiny problem .....	220
7.2	Distribution of generalized interpoint distances for the tiny problem .....	221
7.3	Scatter plot of GID vs NED for the tiny problem.....	222
7.4	Examples of problem instances with differing diversity .....	227
7.5	Overlap of instances for the three methods on the giant problem .....	230
7.6	Distribution of Z for instances where budget = 0.75 * arcs.....	239
7.7	Trajectory of Z when adding arcs (StpAscBasic – StpAscExt12).....	240
7.8	Trajectory of Z* when adding arcs (StpAscExt12 – StpAscBasic) .....	241
7.9	Distribution of Z by number of arcs .....	242
7.10	Distribution of Z (105 arcs) .....	242
7.11	Distribution of Z (210 arcs) .....	242
7.12	Trajectory of Z with 10 candidates .....	243
7.13	Trajectory of Z* with 10 candidates .....	243
7.14	Trajectory of Z with 20 candidates .....	243

7.15	Trajectory of $Z^*$ with 20 candidates.....	243
8.1	Graph for problem instance P1.....	248
8.2	Graph for problem instance P2.....	248
8.3	Graph for problem instance P3.....	249
8.4	Graph for problem instance P4.....	249
8.5	Graph for problem instance P5.....	249
8.6	Objective function trajectories for MDP1 and MDP2 on P1 .....	263
8.7	Objective function trajectories for MDP1 and MDP2 on P2 .....	263
8.8	Objective function trajectories for MDP1 and MDP2 on P3 .....	264
8.9	Objective function trajectories for MDP1 and MDP2 on P4 .....	264
8.10	Objective function trajectories for MDP1 and MDP2 on P5 .....	264
9.1	An example of a route displayed with the route visualizer .....	318



---

## List of Tables

3.1	Definition of incidence and adjacency sets .....	35
3.2	Specifications for problem set A of random graphs .....	60
3.3	Specifications for problem set B of random graphs .....	61
3.4	Rewards and computation times for RICHEST NEIGHBOUR sensitivity analysis .....	62
3.5	Results from Tabu Search sensitivity analysis .....	64
3.6	Heuristics used in phase 1 experiments.....	65
3.7	Results for unimproved heuristics on complete grids .....	65
3.8	Results for improved heuristics on complete grids .....	67
3.9	Results for set A random graphs .....	67
3.10	Heuristics used in phase 2 experiments.....	69
3.11	Results from experiments on designed graphs with $C = 36$ .....	70
3.12	Results from experiments on designed graphs with $C = 72$ .....	71
3.13	Results from experiments on designed graphs with $C = 108$ .....	72
3.14	Results from experiments on designed graphs with $C = 144$ .....	73
3.15	Results from experiments on designed graphs with $C = 180$ .....	74
3.16	Results from experiments on random graphs with $C = 36$ .....	75
3.17	Results from experiments on random graphs with $C = 72$ .....	76
3.18	Results from experiments on random graphs with $C = 108$ .....	77
3.19	Results from experiments on random graphs with $C = 144$ .....	78
3.20	Results from experiments on random graphs with $C = 180$ .....	79
6.1	Configuration settings for the Steepest Ascent MLS instances.....	174
6.2	Configuration settings for the Simulated Annealing MLS instances .....	177
6.3	Configuration settings for the Tabu Search MLS instances .....	179



6.4	Configuration settings for the Variable Neighbourhood Search MLS instances.....	181
6.5	Proportion of problem instances at each rank for Steepest Ascent.....	185
6.6	Combinations of ranks for Steepest Ascent.....	187
6.7	Relative importance of input characteristics to neural net.....	188
6.8	Classification results on the test set.....	189
6.9	Distribution of ranks for Simulated Annealing heuristics ordered by total reward.....	190
6.10	Parameters for top 4 Simulated Annealing heuristics.....	191
6.11	Proportion of problem characteristic variation explained by each principle component .....	194
6.12	Coefficients of the first six principle components under varimax rotation .....	195
6.13	Distribution of ranks for Tabu Search heuristics ordered by total reward.....	199
6.14	Results of pair-wise Tabu Search prediction .....	200
6.15	Partial summary of rank distribution of VNS heuristics .....	202
6.16	Frequency of ranks between SA6 and Tabu10.....	202
6.17	Classification results on the test set for TS and SA.....	203
6.18	Combinations of ranks for TS-VNS hybrids and Tabu10 .....	210
6.19	Comparison of Tabu10 and TS-VNS3 .....	211
6.20	Comparison of Tabu10 and SA-TS .....	211
7.1	Results for the tiny problem .....	225
7.2	Diversity measures for the example sets .....	228
7.3	Total distance for the results of giant problem.....	231
7.4	Total absolute distance by characteristic for the giant problem .....	232
7.5	Average consecutive distance by characteristic for the giant problem.....	233
7.6	Standard deviation of consecutive differences by characteristic for the giant problem .....	234
7.7	Coefficient of variation of consecutive differences by characteristic for the giant problem.....	235
7.8	Maximum consecutive difference by characteristic for the giant problem .....	236
8.1	Density and budget characteristics for the test set of problem instances.....	248
8.2	Reward collected by each heuristic on the test problem instances.....	250
8.3	Reward collected by the MDP heuristics and benchmark heuristics on the test problems.....	262
8.4	MDP move frequencies for MDP2.....	265
8.5	Reward collected by the ADLS heuristic, compared with other heuristics .....	274

---

# List of Algorithms

2.1	<b>algorithm</b> CONSTRUCT EULER TOUR FROM EULERIAN GRAPH.....	19
2.2	<b>heuristic</b> TSTSP .....	30
3.1	<b>heuristic</b> PRUNE AND ROUTE.....	44
3.2	<b>heuristic</b> ROUTE AND PRUNE.....	46
3.3	<b>heuristic</b> RICHEST NEIGHBOUR ( $n$ ).....	47
3.4	<b>procedure</b> DELETE REDUNDANCY .....	49
3.5	<b>procedure</b> GRID GROW.....	55
3.6	<b>procedure</b> GRID SELECT.....	56
4.1	<b>procedure</b> MLS SEARCH ITERATION PROCESS .....	89
5.1	<b>metaheuristic</b> ITERATED LOCAL SEARCH .....	125
5.2	<b>metaheuristic</b> MULTI-START.....	131
5.3	<b>procedure</b> GREEDY RANDOMIZED ADAPTIVE CONSTRUCTION .....	133
5.4	<b>metaheuristic</b> SIMULATED ANNEALING.....	138
5.5	<b>MLS admissibility condition</b> METROPOLIS CONDITION .....	144
5.6	<b>MLS admissibility condition</b> BASIC THRESHOLD ACCEPTING.....	144
5.7	<b>MLS admissibility condition</b> GREAT DELUGE.....	145
5.8	<b>MLS admissibility condition</b> RECORD-TO-RECORD TRAVEL .....	145
5.9	<b>procedure</b> HYPERHEURISTIC.....	154
6.1	<b>procedure</b> GRIDDESELECT .....	163
6.2	<b>procedure</b> GRIDGROW- $k$ -SEEDS.....	164
6.3	<b>MLS admissibility condition</b> FEASIBLE (ASRP).....	169
6.4	<b>MLS admissibility condition</b> IMPROVING .....	169
6.5	<b>MLS fitness function</b> OBJECTIVE.....	170

6.6	<b>MLS update-memory</b> UPDATE BEST-SO-FAR (OBJECTIVE) .....	170
6.7	<b>MLS trigger</b> LOCAL OPTIMUM.....	170
6.8	<b>MLS trigger</b> ITERATIONS SINCE LAST TRIGGER ( <i>trig</i> ).....	171
6.9	<b>MLS trigger</b> TOTAL ITERATIONS .....	171
6.10	<b>MLS trigger</b> TRIGGER TRIP COUNT ( <i>trig</i> ) .....	172
6.11	<b>MLS response</b> TERMINATE.....	172
6.12	<b>MLS response</b> DEACTIVATE TRIGGER ( <i>trig</i> ).....	172
6.13	<b>MLS response</b> ACTIVATE TRIGGER ( <i>trig</i> ).....	172
6.14	<b>MLS response</b> DEACTIVATE ADMISSIBILITY CONDITION ( <i>c</i> ).....	172
6.15	<b>MLS response</b> ACTIVATE ADMISSIBILITY CONDITION ( <i>c</i> ).....	173
6.16	<b>MLS configuration</b> RICHEST NEIGHBOUR.....	173
6.17	<b>MLS configuration</b> STEEPEST ASCENT.....	173
6.18	<b>MLS configuration</b> SIMULATED ANNEALING.....	174
6.19	<b>MLS admissibility condition</b> ANNEALING PROBABILITY .....	175
6.20	<b>MLS response</b> REDUCE ANNEALING TEMPERATURE .....	175
6.21	<b>MLS trigger</b> TEMPERATURE THRESHOLD.....	176
6.22	<b>MLS configuration</b> TABU SEARCH.....	178
6.23	<b>MLS admissibility condition</b> TABU ARCS WITH ASPIRATION .....	178
6.24	<b>MLS update-memory</b> UPDATE TABU ARCS .....	179
6.25	<b>MLS configuration</b> VARIABLE NEIGHBOURHOOD SEARCH .....	180
6.26	<b>MLS response</b> SWITCH TO BASIC MOVE-TYPES.....	180
6.27	<b>MLS response</b> SWITCH TO EXTENDED MOVE-TYPES .....	181
6.28	<b>MLS configuration</b> HYBRID – SA & TS .....	204
6.29	<b>MLS configuration</b> HYBRID – SA & VNS .....	206
6.30	<b>MLS response</b> SET CANDIDATE LIST SIZE ( <i>size</i> ).....	207
6.31	<b>MLS response</b> SET ANNEALING TEMPERATURE ( <i>temp</i> ).....	207
6.32	<b>MLS configuration</b> HYBRID – TS & VNS .....	208
8.1	<b>MLS configuration</b> ITERATIVE SAMPLING LOCAL SEARCH.....	250
8.2	<b>MLS configuration</b> ASRP TEMPLATE FOR MDP.....	257
8.3	<b>MLS admissibility condition</b> ALL ADMISSIBLE .....	258
8.4	<b>MLS admissibility condition</b> IMPROVING FITNESS AND FEASIBLE OR INFEASIBLE BUT DECREASING COST .....	258
8.5	<b>MLS admissibility condition</b> ANNEALING PROBABILITY AND FEASIBLE .....	259
8.6	<b>MLS admissibility condition</b> TABU ARCS WITH ASPIRATION AND FEASIBLE .....	260
8.7	<b>MLS configuration</b> MDP CONTROL HEURISTIC 1 .....	261
8.8	<b>MLS configuration</b> MDP CONTROL HEURISTIC 2.....	262

8.9	<b>MLS response</b> START DIVERSIFICATION PHASE.....	270
8.10	<b>MLS response</b> END DIVERSIFICATION PHASE.....	271
8.11	<b>MLS update-memory</b> UPDATE DIVERSIFICATION WEIGHTS.....	272
8.12	<b>MLS configuration</b> ADAPTIVE DIVERSIFICATION LOCAL SEARCH.....	273
8.13	<b>MLS fitness function</b> REWARD TO COST RATIO.....	274

