TACTIC-BASED LEARNING FOR COLLECTIVE LEARNING SYSTEMS

by Alice Armstrong

Master of Science, August 2002, The George Washington University

Bachelor of Science, May 2000, The George Washington University

A Dissertation submitted to

The Faculty of
School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Science

May 18, 2008

Dissertation directed by

Peter Bock

Professor of Engineering

The School of Engineering and Applied Science of The George Washington University certifies that Alice Armstrong has passed the Final Examination for the degree of Doctor of Science as of April 11, 2008. This is the final and approved form of the dissertation.

TACTIC-BASED LEARNING FOR COLLECTIVE LEARNING SYSTEMS

Alice Armstrong

Dissertation Research Committee:

Peter Bock, Professor of Engineering, Dissertation Director

Abdou Youssef, Professor of Engineering and Applied Science, Committee Member

C. Dianne Martin, Professor of Engineering and Applied Science, Committee Member

DEDICATION

This work is dedicated to Edna Wright, who knew that an education was worth the work.

ACKNOWLEDGMENTS

I wish to thank all of those who have helped me through this process, especially William Lawrenson, for making many great leaps of faith with me; Don and Carolyn Armstrong, for never asking when I was going to get a real job; Peter Bock, for all the support and guidance; Alison Alvarez, for asking me to sign up for “this AI course”; Ken Byrer and Eugenia Martin, for their general awesomeness and open door; Amira Djebbari, for reading drafts and calling when I needed it; all members of the penis list, past and present, for being such excellent friends; and Tad Morgan, for Ciao! Bella!! in Ben Lomond, CA, a reminder, in restaurant form, that it is possible to live the dream.

ABSTRACT

Tactic-Based Learning for Collective Learning Systems

Tactic-Based Learning is a new selection policy for statistical learning systems that has been tested with a Collective Learning Automaton which solves a simple, but representative problem. Current selection policies respond to immature stimuli that do not yet have high-confidence responses associated with them by selecting responses randomly. Albeit unbiased, this policy ignores any confident information already acquired for *other* well-trained stimuli. To exploit this confident information, Tactic-Based Learning hypothesizes that *in the absence of a sufficiently confident response to a given stimulus, selecting a confident response to a different, but nonetheless well-trained stimulus is a better strategy than selecting a random response*. Tactic-Based Learning does not require any feature comparison in search of an appropriate response. Preliminary results show that Tactic-Based Learning significantly accelerates learning and reduces error, especially when several stimuli share the same response, *i.e.*, when broad domain generalization is possible. Tactic-Based Learning reduces the use of pseudo-random number generators in the response selection process. Additionally, Tactic-Based Learning assists the recovery of learning performance when the problem evolves over time.

TABLE OF CONTENTS

DEDICATION iii

ACKNOWLEDGMENTS iv

ABSTRACT v

TABLE OF CONTENTS vi

LIST OF FIGURES xii

LIST OF TABLES xv

GLOSSARY OF TERMS xvii

EXECUTIVE SUMMARY xxvii

Original and Significant Contributions xxxiii

CHAPTER 1: INTRODUCTION 1

1.1 Motivation 1

1.2 Problem Statement 6

CHAPTER 2: RELATED WORK 7

2.1 The Psychological Basis of Infant Learning 7

2.1.1 The Theory of Jean Piaget 12

2.2 Applications of Piaget’s ideas to Artificial Intelligence 22

2.2.1 Drescher (1991) 22

2.2.2 Birk (1996); Birk & Paul (2000) 23

2.2.3 de Jong (1999) 24

2.3 The Pigeonhole Principle 25

2.4 Reinforcement Learning 26

2.4.1 Environment...27

2.4.2 Agent’s Perception of the Environment..27

2.4.3 Reinforcement Signal...27

2.4.4 Agent Actions...28

2.4.5 Policy...28

2.4.6 Unsupervised Learning 29

2.4.7 Model of the Environment 29

2.4.8 Practical Uses of Reinforcement Learning..29

2.5 Collective Learning Systems Theory 30

2.5.1 Introduction...30

2.5.2 Environment...33

2.5.3 Stimulus..34

2.5.4 Evaluation 34

2.5.5 Collective Learning Automaton..35

2.5.6 Compensation Policy 36

2.5.7 Update 37

2.5.8 State Transition Matrix (STM)..37

2.5.9 Response Selection..39

2.6 Applications of Collective Learning Systems 41

2.7 Maximum Likelihood...41

2.8 Primary Research Objective 44

CHAPTER 3: SOLUTION METHOD..45

3.1 Introduction to Solution Method 45

3.2 Tactic-Based Learning..46

3.2.1 Life Cycle of a Stimulant...46

3.2.2 Life cycle of Tactics...51

3.3 The TBL Algorithm and Data Structures..67

3.3.1 Flow of Control...67

3.3.2 Pseudo-code...68

3.3.3 *Environment*..74

3.3.4 *CLA*...75

3.3.5 *STM*...78

3.3.6 Discussion of the selection policy algorithms.......................................79

3.3.7 *Discussion of* `UpdateSTM`...82

3.3.8 *Tactic*...83

3.3.9 *Stimulant*...85

3.3.10 *Response*...88

3.3.11 *TruthTable*...90

3.3.12 Time Complexity of the TBL Algorithm..91

3.4 The TruthTable Game...92

3.4.1 Overview of the game...92

3.4.2 Rules of the game..96

3.4.3 Basic game play..96

3.4.4 Justification of phased game play...97

3.4.5 Game states used...99

3.5 Goals...103

3.6 Performance Metrics..108

3.6.1 Payoff 111

3.6.2 *n*-tile advantage 112

3.6.3 Expense 112

3.6.4 Footprints 113

3.6.5 Learning curves..116

3.6.6 TBL roles 117

3.6.7 Random responses..119

CHAPTER 4: EXPERIMENTS..121

4.1 Experiment Design 121

4.1.1 OP Pilots for fixed conditions and parameters..................................121

4.1.2 Organization of Experiments 124

4.1.3 Stationary Game, One Target Cell per Input 125

4.1.4 Stationary Game, Two Target Cells per Input 130

4.1.5 Task-switching Game, One Target Cell per Input 135

4.2 Experimentation & Reduced Results..142

CHAPTER 5: RESULTS AND CONCLUSIONS 143

5.1 Introduction 143

5.2 Experiment 1 143

5.2.1 Results..144

5.2.2 Formal Conclusions from Experiment 1 154

5.3 Experiment 2 155

5.3.1 Results..155

5.3.2 Best Performance: 4-tuple (κ_s=70, κ_w=70, κ_i=70, κ_d=70) 156

5.3.3 Neutral Performance: 4-tuple (κ_s=99.99, κ_w=95, κ_i=50, κ_d=50) 165

5.3.4 Poor Performance: 4-tuple (κ_s=95, κ_w=95, κ_i=95, κ_d=50).................. 168

5.3.5 Formal Conclusions from Experiment 2 172

5.4 Experiment 3 173

5.4.1 Results 174

5.4.2 Formal Conclusions for Experiment 3 180

5.5 Experiment 4 180

5.5.1 Results 181

5.5.2 Best Performance: 4-tuple (κ_s=70, κ_w=70, κ_i=70, κ_d=50) 181

5.5.3 Average Performance: 4-tuple (κ_s= 99.99, κ_w= 99.99, κ_i= 99.99, κ_d= 99.99) 184

5.5.4 Poor Performance: 4-tuple (κ_s= 99.99, κ_w= 50, κ_i= 50, κ_d= 50)......... 187

5.5.5 Formal Conclusions for Experiment 4 189

5.6 Experiment 5 189

5.6.1 Results 190

5.6.2 Formal Conclusions for Experiment 5 199

5.7 Experiment 6 200

5.7.1 Results 200

5.7.2 Best Performance: 4-tuple (κ_s=90, κ_w=90, κ_i=90, κ_d=70) 200

5.7.3 Average Performance: 4-tuple 208

5.7.3 (κ_s=99.99, κ_w=99.99, κ_i=70, κ_d=70) 208

5.7.4 Poor Performance: 4-tuple (κ_s=99.99, κ_w=50, κ_i=50, κ_d=50)............. 210

5.7.5 Formal Conclusion from Experiment 6 ... 215

5.8 Summary of Formal Conclusions ... 216

5.8 Informal Observations ... 220

5.9 Future Directions ... 221

REFERENCES ... 225

APPENDIX A: EXPERIMENT 1, PAYOFF RESULTS ... 229

APPENDIX B: EXPERIMENT 1, EXPENSE RESULTS ... 236

APPENDIX C: EXPERIMENT 1 *N*-TILE ADVANTAGE RESULTS ... 242

APPENDIX D: EXPERIMENT 3, PAYOFF RESULTS ... 251

APPENDIX E: EXPERIMENT 3, EXPENSE RESULTS ... 254

APPENDIX F: EXPERIMENT 3 *N*-TILE RESULTS ... 257

APPENDIX G: EXPERIMENT 5, PAYOFF RESULTS ... 266

APPENDIX H: EXPERIMENT 5, EXPENSE RESULTS ... 270

APPENDIX I: EXPERIMENT 5 *N*-TILE RESULTS ... 274

DIGITAL APPENDICES ... 287

LIST OF FIGURES

Figure 1: A complete Collective Learning System (CLS) …………………………….. 32

Figure 2: State Transition Matrix (STM) ……………………………………...…. 38

Figure 3: The Life Cycle of a Stimulant ………………………………………………. 47

Figure 4: Sample Game States ………………………………………………………… 92

Figure 5: Non-equivalent Game States ………………………………………………... 93

Figure 6: TBL$_\alpha$ for sample states ……………………………………………….... 95

Figure 7: Sample Initial and Secondary Phases ……………………………………….. 97

Figure 8: All Possible Initial States …………………………………………….…. 101

Figure 9: Initial States Used as Factors ……………………………………………… 102

Figure 10: Learning curves for a single treatment …………………………………… 110

Figure 11: Sample Footprint …………………………………………………….…. 115

Figure 12: Sample Learning Curve …………………………………………...…. 116

Figure 13: Simplified Stacked Bar Graph of TBL Roles …………………………..… 118

Figure 14: Sample TBL Role Graph …………………………………………….…. 119

Figure 15: Sample Graph of Random Responses …………………………………..… 120

Figure 16: Experiment 2 Results ……………………………………….………. 156

Figure 17: Experiment 2 Results ……………………………………….………. 158

Figure 18: Experiment 2 Results ……………………………………….………. 158

Figure 19: Experiment 2 Results ……………………………………….………. 160

Figure 20: Experiment 2 Results ……………………………………….………. 160

Figure 21: Experiment 2 Results ……………………………………….………. 161

Figure 22: Experiment 2 Results …………………………………………..……… 162

Figure 23: Experiment 2 Results …………………………………………..……… 162

Figure 24: Experiment 2 Results …………………………………………..……… 163

Figure 25: Experiment 2 Results …………………………………………..……… 164

Figure 26: Experiment 2 Results …………………………………………..……… 164

Figure 27: Experiment 2 Results …………………………………………..……… 166

Figure 28: Experiment 2 Results …………………………………………..……… 167

Figure 29: Experiment 2 Results …………………………………………..……… 167

Figure 30: Experiment 2 Results …………………………………………..……… 168

Figure 31: Experiment 2 Results …………………………………………..……… 169

Figure 32: Experiment 2 Results …………………………………………..……… 170

Figure 33: Experiment 2 Results …………………………………………..……… 171

Figure 34: Experiment 2 Results …………………………………………..……… 171

Figure 35: Experiment 2 Results …………………………………………..……… 172

Figure 36: Experiment 4 Results ………………………………….……..……...… 183

Figure 37: Experiment 4 Results ………………………………….……..……...… 183

Figure 38: Experiment 4 Results ………………………………….……..……...… 184

Figure 39: Experiment 4 Results ………………………………….……..……...… 185

Figure 40: Experiment 4 Results ………………………………….……..……...… 186

Figure 41: Experiment 4 Results ………………………………….……..……...… 186

Figure 42: Experiment 4 Results ………………………………….……..……...… 187

Figure 43: Experiment 4 Results ………………………………….……..……...… 188

Figure 44: Experiment 4 Results ……………………………….……..……...… 188

Figure 45: Experiment 6 Results ……………………………….……..……...… 202

Figure 46: Experiment 6 Results ……………………………….……..……...… 202

Figure 47: Experiment 6 Results ……………………………….……..……...… 203

Figure 48: Experiment 6 Results ……………………………….……..……...… 204

Figure 49: Experiment 6 Results ……………………………….……..……...… 205

Figure 50: Experiment 6 Results ……………………………….……..……...… 205

Figure 51: Experiment 6 Results ……………………………….……..……...… 206

Figure 52: Experiment 6 Results ……………………………….……..……...… 207

Figure 53: Experiment 6 Results ……………………………….……..……...… 207

Figure 54: Experiment 6 Results ……………………………….……..……...… 209

Figure 55: Experiment 6 Results ……………………………….……..……...… 209

Figure 56: Experiment 6 Results ……………………………….……..……...… 210

Figure 57: Experiment 6 Results ……………………………….……..……...… 211

Figure 58: Experiment 6 Results ……………………………….……..……...… 212

Figure 59: Experiment 6 Results ……………………………….……..……...… 213

Figure 60: Experiment 6 Results ……………………………….……..……...… 214

Figure 61: Experiment 6 Results ……………………………….……..……...… 214

Figure 62: Experiment 6 Results ……………………………….……..……...… 215

LIST OF TABLES

Table 1: Non-disjoint Classes …………………………………………………….… 42

Table 2: Parameters and Condition for the TruthTable OP Pilots …………………… 122

Table 3: Data collected during test points ……………………………………….… 124

Table 4: OP Pilot Experiment Design for TBL Thresholds ………………………..… 126

Table 5: OP Pilot Experiment Design for the Compensation Thresholds …………… 127

Table 6: Experiment 1 Block Design (canonical) …….……………………………… 128

Table 7: Design for Experiment 2 (close inspection) ……………………………...… 130

Table 8: OP Pilot Experiment Design for TBL Thresholds ………………………..… 131

Table 9: OP Pilot Experiment Design for Compensation Threshold ………………… 132

Table 10: Experiment 3 Block Design (canonical) ………………………………...… 133

Table 11: Design for Experiment 4 (close inspection)…………..…………………..… 135

Table 12: Tertiary Phases of the TruthTable game …………………………………...… 137

Table 13: OP Pilot Experiment Design for TBL Thresholds ………………………… 138

Table 14: OP Pilot Experiment Design for Compensation Threshold ……………..… 139

Table 15: Experiment 5 Block Design (canonical) ………………………………...… 140

Table 16: Design for Experiment 6 (close inspection) ……………………………..… 142

Table 17: Experiment 1 Results ……………………………………………………… 147

Table 18: Experiment 1 Results ……………………………………………………… 148

Table 19: Experiment 1 Results ……………………………………………………… 149

Table 20: Experiment 1 Results ……………………………………………………… 152

Table 21: Experiment 1 Results ……………………………………………………… 153

Table 22: Experiment 3 Results ……………………………….……….……… 177

Table 23: Experiment 3 Results ……………………………….……….……… 179

Table 24: Experiment 5 Results ……………………………….……….……..…. 192

Table 25: Experiment 5 Results ……………………………….……….……..…. 193

Table 26: Experiment 5 Results ……………………………….……….……..…. 194

Table 27: Experiment 5 Results ……………………………….……….……..…. 196

Table 28: Experiment 5 Results ……………………………….……….……..…. 197

Table 29: Experiment 5 Results ……………………………….……….……..…. 198

GLOSSARY OF TERMS

Defintion 05: The **environment** of a game includes the necessary space, matter, and energy to conduct the game. pp 33

Defintion 07: A **contest** is an instantation of a game that is one complete instance of a game played from beginning to end. pp 35

Defintion 08: A **stage** is the phase of a contest that leads to an evaluation. pp 35

Defintion 09: The **collection length**, $\boldsymbol{c}$, is the number of responses that the learning agent makes before being evaluated by the environment. pp 35

Defintion 10: The **history,** $\boldsymbol{\eta_s}$, of a stage is the collection of interactions that have occurred since te last evaluation. pp 35

Defintion 11: A **Collective Learning Automata** (**CLA**) is a learning agent inside a Collective Learning System. pp 36

Defintion 12: The **compensation,** $\boldsymbol{\gamma}$**,** is an interpretation of the environment’s evaluation by the CLA. pp 36

Defintion 13: The **State Transition Matrix** (**STM**) is the knowledge matix that maps the input domain into the output range. pp 38

Defintion 15: a **stimulant** is a permanent element in the stimulus domain of the STM. pp 38

Defintion 16: a **stimulus** is an input from the environment to the CLA. pp 39

Defintion 17: a **respondent** is a permanent element in the response range of an STM. pp 39

Defintion 18: a **response** is an output element passed from the CLA to the environment. pp 39

Defintion 19: A **selection policy** dictates how respondents are selected for a given stimulant. pp 39

Defintion 20: The **reject confidence** is the confidence that, for a given stimulant, it is possible to reject the hypothesis that there is no difference between the largest posterior probability and the *apriori* probability. pp 40

Defintion 21: The **tie confidence** is the confidence that, for a given stimulant, the hypothesis that there is no difference between the largest posterior probability and the second largest posterior probability *can be* rejected. pp 40

Defintion 23: **Maximum Likelihood Estimation (MLE)** is "a statistical decision rule that examines the probability function of a [stimulus] for each of the classes, and assigns the [stimulus] to the class with the highest probability." pp 43

Postulate 01: The **selection confidence** of a stimulant is the minimum of its tie and reject confidences (worst-case assumption). pp 48

Defintion 24: A **seeker stimulant** has no effective tactic and uses the Standard selection policy. pp 48

Defintion 25: A **tactic** is a respondent that follower stimulants may select. pp 48

Defintion 26: The **primary respondent** for a stimulant is a respondent with the largest weight. pp 48

Defintion 27: The **secondary respondent** for a stimulant is a respondent with the second largest weight. pp 48

Defintion 28: A **supporter stimulant** is a stimulant whose primary respondent is a tactic. pp 48

Defintion 29: The **support threshold** of a CLA specifies the minimum selection confidence required of a stimulant to support a tactic. pp 48

Defintion 30: The **withdrawal threshold** of a CLA specifies the minimum selection confidence of a supporter stimulant required to maintain support for a tactic. (By implication, the withdrawal threshold cannot be higher than the support threshold.) pp 49

Defintion 31: A **follower stimulant** is a stimulant that uses a tactic to guide its response selection. pp 49

Postulate 02: A tactic's **potency**, φ, is the average compensation that a follower stimulant has received while being a follower of the tactic. pp 50

Defintion 32: An **independent stimulant** is a stimulant that has an effective tactic, but follows the Standard selection policy. pp 50

Defintion 33: The **independence threshold** of a CLA is the minimum selection confidence required for a follower stimulant become independent. (By implication, the independence threshold cannot be higher than the support threshold.) pp 50

Defintion 34: The **dependence threshold** of a CLA is the minimum selection confidence an independent stimulant must maintain to remain independent. (By implication, the dependence threshold cannot be higher than the independence threshold.) pp 51

Postulate 03: The **local potency,** φ_l, of a tactic is a measure of effectiveness of a tactic for an individual follower stimulant and is calculated as follows:

$$\varphi_l = \sum_{\mathrm{i}=1}^{n} \frac{\gamma_i}{n_l}$$

where n_l is the number of times that the tactic has been used and γ_i is compensation value received. pp 52

Postulate 04: The **global potency,** φ_g, of a tactic is a measure of effectiveness of a tactic for stimulants that have followed the tactic and is calculated as follows:

$$\varphi_g = \sum_{\mathrm{i}=1}^{n} \frac{\gamma_i}{n_g}$$

where n_g is the number of times that the tactic has been used and γ_i is compensation value received. pp 52

Postulate 05: The **minimum local potency** of a tactic is 1, the minimum non-penalty. pp 53

Postulate 06: A tactic **resigns** when it no longer has any supporters. After a tactic resigns, it is no longer available globally or locally for any stimulants to follow. pp 53

Postulate 07: The evaluation policy for the game is the average evaluation for the collection length,

$$\frac{r_{correct}}{n}$$

where $r_{correct}$ is the number of correct *Responses* and n is the total number of *Responses*. pp 75

Defintion 35: A stimulant whose tie and reject confidences are greater than or equal to the tie and reject thresholds, respectively, generates a **confident response**. pp 76

Defintion 36: The **compensation threshold**, κ_γ**,** is a parameter in the compensation policy. When a CLA's average selection confidence crosses this threshold, the compensation policy becomes more stringent. pp 77

Postulate 08: The compensation policy for the unordered game generates the compensation value, γ, is as follows: pp 78

```
IF $\xi$ = 1 OR ($\xi > 0$ AND $\xi > \xi_{anticipated}$) THEN
    $\gamma_{confident} = 1 + 0.1(\xi)$
    $\gamma_{other} = \gamma_{confident}$
ELSE IF $\xi > 0$ AND $\xi = \xi_{anticipated}$ THEN
    IF avgerageSelectionConfidence < $\kappa_\gamma$ THEN
        $\gamma_{confident} = 1 + 0.1(\xi)$
        $\gamma_{other} = \gamma_{confident}$
    ELSE
        $\gamma_{confident} = 0.999$
        $\gamma_{other} = 0.975$
    END IF
ELSE IF $\xi > 0$ AND $\xi < \xi_{anticipated}$ THEN
    IF avgerageSelectionConfidence < $\kappa_\gamma$ THEN
        $\gamma_{confident} = 1 + 0.1(\xi)$
        $\gamma_{other} = \gamma_{confident}$
    ELSE
        $\gamma_{confident} = 0.96$
        $\gamma_{other} = 0.98$
    END IF
ELSE
    $\gamma_{confident} = 0.85$
    $\gamma_{other} = 0.9$
END IF
```

Postulate 09: The **update policy** for both games is: pp 83

```
IF ($\phi$ is confident) THEN
    $w^1 \leftarrow w$
ELSIF ($\phi$'s tie confidence >= tie threshold AND
    $\phi$'s reject confidence >= reject threshold)
    $w^1 \leftarrow w\gamma_{confident}$
ELSE
    $w^1 \leftarrow w\gamma_{normal}$
END IF
```

Postulate 10: The time complexity of the removal of a global tactic has

$$O(N) = N \log N$$ pp 91

Postulate 11: The average time complexity increase for using TBL is

$$O(N)_{TBL} = (0, N]O(N)_{standard}$$ pp 92

Defintion 37: A **state** of the TruthTable game is one of its arrangements of input-output pairs. pp 92

Postulate 12: The **Tactic-Based Learning advantage** (**TBL$_{\alpha}$**) of a given game state is computed as follows:

$$TBL_{\alpha} = \sum_{i=1}^{n} c_i(c_i - 1)$$

where n = the size of the range (the number of classes) and c_i = the number of inputs assigned to a given output. pp 94

Postulate 13: The **confident accuracy** of a CLA is the average selection confidence for a given test point multiplied by the average score for the test point expressed as a percentage. pp 96

Postulate 14: The **scoring function** of the TruthTable game is the fraction of correct outputs in the current history. pp 96

Defintion 38: A **target cell** is a cell in a TruthTable, which signifies a correct input-output pair. pp 99

Defintion 39: A **target response** is an output in a TruthTable that is associated with at least one target cell. pp 99

Defintion 40: A **substate** is a small section of a state; it usually has all of the target responses in it. pp 99

Defintion 41: The **first termination**, t_1, is the contest at which the first CLA satisfies the stopping criterion. pp 109

Defintion 42: The **second termination**, t_2, is the contest at which the second CLA reaches the stopping criterion. pp 109

Postulate 15: the **benefit**, $\boldsymbol{b_i}$, at a given test point is computed as follows:

$$\boldsymbol{b_i} = \mathscr{R}_{H0}\,(s_{TBL} - s_{Standard})$$

where s_{TBL} and $s_{Standard}$ are the scores of the TBL and Standard-CLAs, respectively, and $\mathscr{R}_{H0}$ is the confidence with which the null hypothesis H_0 that there is no difference between the two scores can be rejected. pp 111

Postulate 16: the **Payoff**, ***P*,** of a treatment is: pp 111

$$P = \frac{\sum_{i=1}^{n} b_i}{n_{t_1}}$$

Postulate 17: The ***n*-tile advantage** of a treatment at a given contest is the two-tailed rejection confidence of H_0. pp 112

Postulate 18: The **Expense** is the difference between the number of contests the TBL-CLA needs to terminate and the number of contests the Standard-CLA needs to terminate. pp 113

LIST OF ASSUMPTIONS AND RESTRICTIONS

- The fundamental ideas of Jean Piaget provide a valid psychological basis for this research. pp 12
- Learning agents are endowed with a fixed number of responses (classes), *i.e.*, the cardinality of the response range (intent) is fixed. pp 20
- Only single-layer, non-hierarchical learning systems are considered in this research. pp 20
- This research does not consider multi-agent systems. pp 24
- When training is complete, the final cardinality (extent) of the stimulus domain is greater than or equal to the cardinality of the response range. pp 26
- CLAs will be implemented without forward context. pp 31
- This research will not consider multi-agent systems. pp 34
- The compensation policy is fixed. pp 37
- the posterior probabilities of stimulants are normally distributed. pp 41
- This research is does not consider presupervised learning scenarios. pp 41

EXECUTIVE SUMMARY

In Collective Learning Systems (CLS), a Collective Learning Automaton (CLA) learns the appropriate response for each stimulus by selecting responses until one of them emerges as statistically optimal, guided by feedback from an evaluating Environment (Bock 1976). Generally, CLS theory ignores what has already been learned by other stimuli when making decisions about a new stimulus. Many psychologists agree that applying successful solutions for old problems to new and often unrelated problems is a useful learning strategy (Piaget 1936, Pulaski 1980, Berk 2003). Although this research does not attempt to replicate human behavior at any level, biologically and psychologically inspired mechanisms and methods can often provide useful insights and hints for machine learning methods (Heckman 2004).

This research deals with a selection policy for CLAs, called Tactic-Based Learning (TBL), which accelerates learning by applying knowledge about one well-learned situation to another. Although many machine learning algorithms can achieve excellent results by identifying similar feature vectors (explicit domain generalization), they all require postulating a sensible and computable distance metric. For example, the k-Nearest Neighbor algorithm (Mitchell 1997, Moore & Lee 1994) computes similarity using the Euclidean distance between vectors in an ordered n-dimensional space. On the

other hand, although case-based reasoning (Sycara *et al.* 1992) allows feature vectors to be categorical, a distance metric of some kind must be postulated to identify similar cases.

For many problem domains, it is not possible to postulate a meaningful distance metric. For example, in Natural Language Processing there is no direct way to compute the distance between the meanings of words, so other methods must be devised (Portnoy & Bock 2005). TBL, however, does not compare feature vectors at all, and is thus applicable to a wide problem domain.

A CLA learns how to respond to stimuli appropriately using the algedonic cycle (Beer 1966), as illustrated in Figure 1. The CLA is embedded in an Environment that sends a stream of stimuli to the CLA and periodically issues evaluations of the CLA's responses to these stimuli. A stimulus is a vector of several features that describes some state of the Environment. The CLA uses a State Transition Matrix (STM) to store each unique stimulus that has been received, along with its occurrence count (sample size) and an estimate of the probability that each possible response is valid for this stimulus. For each stimulus that is received, the CLA uses these probabilities to select a response, which is then sent to the Environment. These selection probabilities are updated based on periodic evaluations issued to the CLA by the Environment at the end of a stage, which is a sequence of responses by the CLA.

For a given stimulus, the Standard CLA (a CLA that does not use TBL) selects the response with the highest statistical confidence if the confidence is sufficiently high; otherwise, a response is selected at random. All responses are sent to the Environment, and at the end of each stage, the Environment evaluates their collective performance. This evaluation is issued to the CLA, where the compensation function converts the evaluation into an update. The update is applied to all the elements of the probability vectors in the STM that were used to generate the CLA’s responses since the last evaluation (the history of the stage) (Bock 1993).

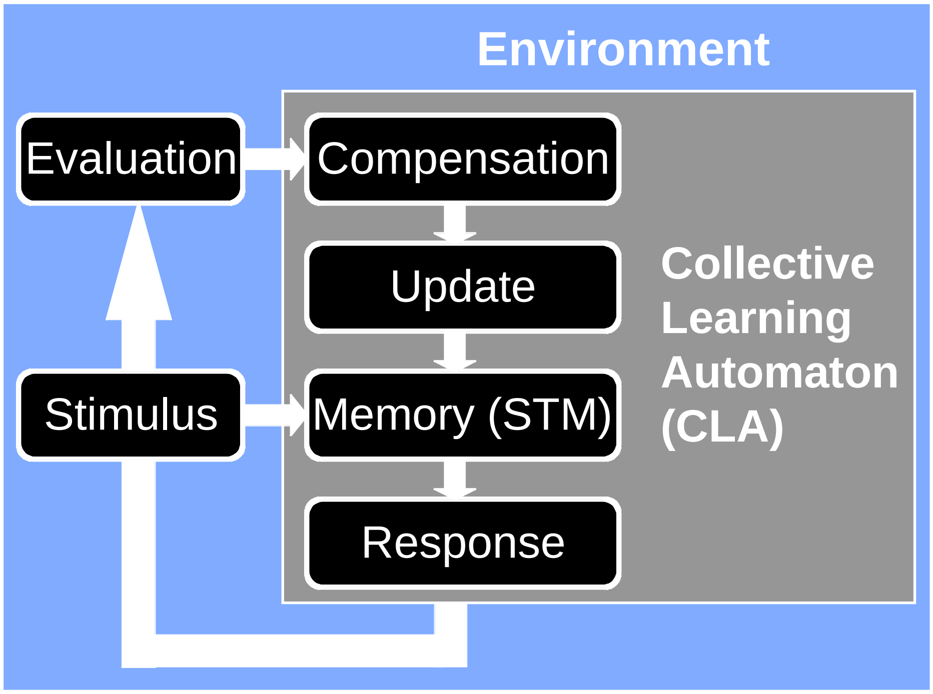


Figure 1: A complete Collective Learning System (CLS). The learning agent, the CLA, engages in the algedonic cycle to acquire knowledge about the environment, eventually eliciting correct responses to its stimuli.

The standard difference of two proportions is used to compute the statistical confidence of each response for every stimulus, which is called the selection confidence of a response.

Tactic-Based Learning is an algorithm that overrides the Standard selection policy used by a Standard-CLA. A TBL-CLA follows the Standard selection policy until one stimulus is sufficiently well trained to elect its primary response as a tactic. A stimulus supports a tactic when its selection confidence is very high. Stimuli that are using a tactic (follower stimuli) simply use this response, assuming it is better than a random response. However, each follower stimulus tracks the effectiveness of the tactic and uses it only as long as it remains effective (an average compensation ≥ 1). When a new tactic becomes available, all stimuli that do not yet have an effective tactic will try it.

The lifecycle of a hypothetical stimulus in a Tactic-Based CLA is described in Figure 2. When there are no tactics in a CLA, all stimuli follow the Standard selection policy and are called seekers. As soon as the first tactic appears, all seekers will investigate it. When a stimulus selects a tactic, it becomes a follower of that tactic. As long as a tactic remains effective for a follower, the follower will continue to use the tactic's response. If a tactic proves ineffective (a parameter of the algorithm), the follower drops this tactic and looks for another. If no other effective tactics are available, the stimulus reverts to the Standard selection policy and becomes a seeker. After a follower has attained a specified selection confidence, it becomes an independent stimulus and reverts to the Standard selection policy. Dropping the tactic allows the independent stimulus to explore its response range. Exploration is useful because it helps avoid settling into a local maximum of the reward function for the CLA. An independent stimulus will either lose confidence in its response and revert to being a follower, or will become confident enough to become a supporter of a tactic itself. An independent stimulus is allowed some latitude, and it will only revert to being a follower if its selection confidence falls below the dependence threshold.

In the event that a supporter loses confidence in its response, the supporter withdraws its support from the tactic it was supporting and reverts to being independent. If the tactic no longer has any supporters, it will no longer be available for use, and any follower stimulants of it will become seekers.

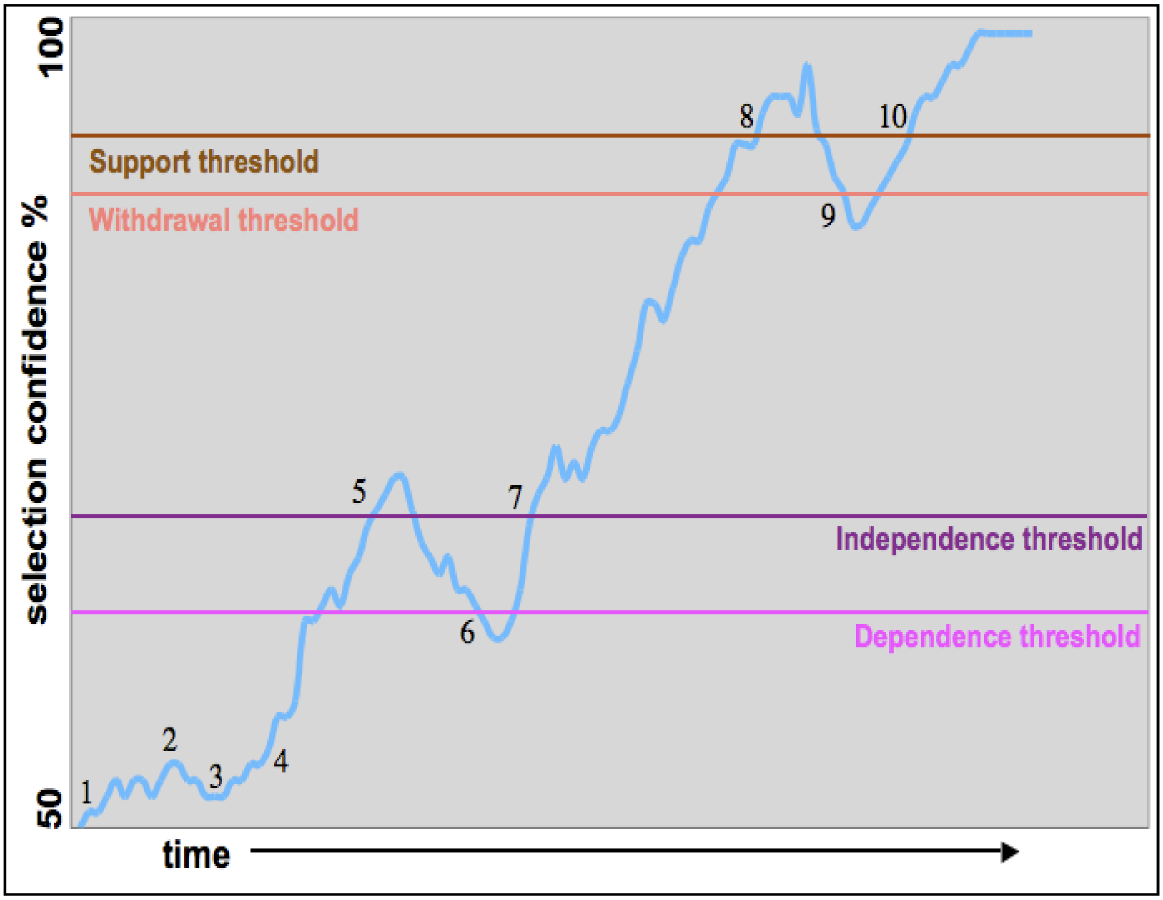


Figure 2: The Life Cycle of a Stimulus, ϕ. (1) The stimulus, ϕ, is encountered for the first time. ϕ is a seeker. (2) The first tactic appears and ϕ becomes a follower of it. (3) The first tactic is not effective, so ϕ abandons it and returns to being a seeker. (4) A second tactic appears and ϕ becomes a follower of this new tactic. (5) the stimulus' selection confidence reaches the independence threshold. ϕ becomes independent. (6) ϕ does not become confident in a respondent and ϕ's selection confidence drops below the dependence threshold. ϕ to becomes follower again of its most effective tactic. (7) ϕ recovers its selection confidence and becomes an independent again. (8) ϕ has become confident enough to become a supporter of its own confident response. If this response is not already a tactic, a new tactic is available to other stimuli. (9) ϕ loses confidence and withdraws its support from its confident response. If ϕ was the only supporter of that tactic, the tactic is no longer available to other stimuli. (10) Once again, ϕ supports its confident response.

There were three factors in these experiments: (1) the TBL thresholds that govern when a stimulus can use or support a tactic (2) the environmental conditions and (3) the collection length (the number of responses that a CLA makes between evaluations).

The TBL thresholds (support, withdrawal, independence, and dependence) were varied incrementally from 50% to 99.99% under the following restrictions:

- Independence threshold ≤ Support threshold
- Dependence threshold ≤ Independence threshold
- Withdrawal threshold ≤ Support threshold

The environmental conditions were varied to consider classification tasks with single and multiple correct classifications of stimuli. Experiments were also conducted in stationary environments and a special case of a non-stationary environment, called a task-switching environment. In a task-switching environment, at some point in the match after the initial contest, the environment suddenly changes the correct solution to the game (partially or completely), and the CLA must abandon its obsolete solution and learn a new solution.

The collection length was varied incrementally between 1 and 12. A collection length of 1 is trivial because the problem becomes a simple process of elimination. A collection length of 12 is considerably more difficult. As the collection length gets longer, it becomes harder to tell which stimuli chose correct responses and which did not.

Original and Significant Contributions

This research makes the following original and significant contributions:

- Tactic-Based Learning is a method for improving learning performance on categorical reinforcement learning tasks by leveraging existing knowledge that *does not require any feature analysis of the domain or the range.*

- Tactic-Based Learning significantly reduces the learning agent's reliance on a pseudo-random number generator for breaking ties when making selections.

- At optimal settings, a CLA which has implemented Tactic-Based Learning *always performs better* on learning tasks than a CLA which has implemented the Standard selection policy *without any significant increase in training time* under the environmental conditions examined in this research.

CHAPTER 1: INTRODUCTION

1.1 Motivation

“When all you have is a hammer, everything looks like a nail”. Although this popular saying sounds a bit pessimistic, it expresses a general quality of human cognition: we take what we already know and try to apply that knowledge or skill to new situations in life. We know full well that the solution may succeed or fail, but we regard it as a better idea to use what we know than simply to act in a random fashion.

In adults, this behavior is often quite subtle. Consider the person who was invited to go boating with some friends and was offered the chance to pilot the boat for a few minutes, being assured that it was “just like driving a car”. A little skeptical, the person got behind the wheel, started the motor, engaged the throttle, and found that almost everything about the experience was different than driving a car. The boat simply did not respond to the controls in the same way as a car. The person quickly adapted and managed to pilot across the lake without putting anyone in excessive danger (although, admittedly, everyone was very glad when an experienced pilot took the helm again).

This story deals with adult behavior, but the ability to learn about new situations by applying known solutions to other problems is one of the cornerstones of *infant* learning.

Consider a twelve-month old who has just discovered the joys of voluntarily dropping things. Toys are grasped and released in her crib. Pacifiers are dropped from her stroller. Food is dropped from her high chair. Some objects she can reach again and drop again, others fall out of her reach. Parents may return some items, but other times items, like food or dirty pacifiers, are not returned.

Through experimentation, the young child learns about dropping objects and about the objects that are dropped. Some things bounce when dropped, some things are too heavy to be lifted, dropping certain things makes mom angry, and other things are fun to drop but are missed once they have disappeared or are out of reach. Dropping soon progresses on to throwing and, once again, the lessons learned about dropping (and perhaps other actions which may seem unrelated) clearly help to sharpen the new skill (Berk 2003).

This research presents a new method for machine learning that allows a learning agent to use information it has acquired about one situation to other, possibly unrelated, situations. The method is based on an understanding of human learning and development. Before going any further, it should be said that the mechanisms for machine learning developed in this research do not replicate human learning mechanisms, nor do the conclusions of the research claim to make formal any statements about human learning based on the machine learning mechanisms.

It seems to make sense to look at what is known about human learning when considering new mechanisms for machine learning. The field of psychology can often help light the way in the search for new approaches to machine learning. Jean Piaget’s study of newborns has influenced psychological research for the past fifty years. Although some of the finer points of his theory have been challenged, his description of the way a newborn acquires knowledge is still considered a reasonable framework.

One of Piaget's fundamental ideas was that of **schemes**, a generalized term for organized knowledge about the world. Schemes are used and modified as the newborn learns about her environment. A newborn's first scheme arises out of her reflexes, but once the first scheme is acquired she will apply it to everything she comes in contact with. This first scheme becomes her first tool for exploring her world.

Piaget's idea of schemes is used in this research as a starting point for the development of a new strategy for reinforcement learning. For this reason, the relevant background and theories that Piaget used to explain the psychological basis of learning in human infants are presented in Section 2.1.1.

The idea of using one solution for two problems may also be considered from a mathematical perspective. If the goal of an adaptive learning agent is to learn the appropriate transform that maps x inputs into y outputs and $x>y$, then by the pigeonhole principle, there must be some inputs which share the same output (Ross and Wright 1988, p207). In other words, the outputs are not mutually exclusive.

Classifiers that use Bayesian posterior probabilities to select the output must accept the lack of mutual exclusivity as "a cost of doing business". A Bayes Optimal classifier gives the minimum possible error, but only for supervised learning agents (Mitchell 1997). Online learning agents do not have a labeled set of training data from which to learn; therefore, the mistakes the learning agent makes in the learning process are included in its knowledge base and they are not estimating the class conditional probabilities. Maximum Likelihood classifiers are not interested in class conditional probabilities and are more appropriate for an online learning agent, but a Maximum Likelihood classifier does not acknowledge or take advantage of any dependence between classes. Perhaps there is a classification strategy that actually exploits any overlap among the classes to improve learning. If a reinforcement learning agent were to apply a solution it had learned in one context to other contexts about which it was still unsure, perhaps this could improve learning by biasing the system towards proven solutions, especially if the solutions from one context to the next shared something in common, albeit some not-at-all-obvious quality.

In current state-of-the-art adaptive learning systems, responses are selected randomly before the learning agent has identified a clear relationship between a given stimulus and a given response. Random selection insures that the response range is fully explored and minimizes bias in the system. Although this is an effective and reasonable strategy for a learning agent that knows nothing, once a learning agent has discovered some relationships in its environment, it seems reasonable that these relationships should be used to tackle new situations.

In their thorough overview of reinforcement learning, Kaelbling *et al.* (1996) close with the following remark:

> There are a variety of reinforcement learning techniques that work effectively on a variety of small problems. But very few of these techniques scale well to larger problems. This is not because researchers have done a bad job of inventing learning techniques, but because it is very difficult to solve arbitrary problems in the general case. In order to solve highly complex problems, we must give up *tabula rasa* learning techniques and begin to incorporate bias that will give leverage to the learning process. [p32]

Kaelbling *et al.* acknowledge that learning from scratch is not sufficient for reinforcement learning to be effective for larger problems. They suggest that sensible biasing and leveraging of learned information be used to address reinforcement learning’s failure to perform on large-scale problems. That suggestion is taken very much to heart in this research.

One common and sensible biasing solution is to perform a comparison of features that describe a stimulus or a response. Most supervised learning techniques rely on feature comparison to generalize from the training set to the test set (see Support Vector Machines as an example). In many cases it is quite rational to assume stimuli that "look similar" should have similar responses or even share the same response; however, there are two major drawbacks to this approach. The first drawback is that for each new problem domain, a new and appropriate comparison operator must be defined; there is no general comparison operator that can be used. The second drawback is that not all problems allow the application of tractable similarity metrics, especially those with categorical stimuli or responses.

For example, in the field of natural language processing, it is very difficult to measure the distance between the words "cat" and "fuel". As a fanciful example, an inappropriate distance metric could result in buying gasoline when one needed cat food; or worse, feeding the cat gasoline because the cat needed "fuel". To avoid both of these drawbacks, the solution cannot depend on the meaning of the stimuli or the responses.

In the general case, the only other possible source of potentially useful information that can be shared is in the stimulus-response space (histogram) of an adaptive learning engine. Currently, there are no machine learning techniques that use this source information without a high risk of becoming stuck in suboptimal solutions (*e.g.,* hill climbing, local beam search, and, in some cases, Genetic Algorithms [Russell & Norvig 2003]). Thus, any method applied to reinforcement learning that makes use of previously learned information must minimize the risk of settling on suboptimal solutions. In addition, any solution should allow the learning agent to apply previously learned responses to new stimuli only as long as the responses prove consistently effective.

1.2 Problem Statement

Although comparing feature-vectors in the domain of a classifier can be a powerful mechanism for generalization, it requires a meaningful similarity metric, which is often very difficult or impossible to devise (*e.g.*, with categorical features). As an alternative, allowing random selection of responses may provide the advantage of promoting exploration, but it can be very slow and can easily result in suboptimal solutions. Nonetheless, there is a great deal of information in the stimulus-response space of a learning agent that perhaps can be exploited to assist in decision-making, even in the early stages of the maturation of the learning agent.

Research Problem: Is there a general and effective method to exploit the knowledge accumulating in the decision space of an adaptive learning agent to improve classification performance without comparing feature-vectors or incurring the cost of excessive random exploration?

CHAPTER 2: RELATED WORK

2.1 The Psychological Basis of Infant Learning

Understanding how newborn children learn is pertinent to this research. This research is primarily concerned with learning in its earliest stages, when little or nothing has been learned about the self or the environment. The reason for this focus lies in the basic assumption that as humans mature and develop more cognitive capacity, they begin to employ various kinds of feature-vector comparison and analysis on many different levels of cognition. Even very young children can categorize people and objects and make decisions about what to do with those things.

If a young child lives in a household with a pet German Shepherd, he will soon learn to recognize other dogs as potential pets and will often take great delight in watching or petting other dogs. This kind of behavior suggests that the young child is generalizing something about dogs, perhaps the fact that dogs are hairy, or that they have four legs and a tail. The fact that the child will extend his knowledge about his household pet to all other dogs but not to, say, dishes or cars, suggests that the young child must be generalizing about what makes a dog a dog.

Studies have shown that by 12 months, children's memory and problem solving skills become significantly less context-dependent (Hartshorn *et al.* 1998; Hayne, Boniface, &

Barr 2000), meaning that children do not need to be in the exact same situation in order to remember how to perform a task or recognize an object. These studies argue in favor of children's ability to generalize learned responses to relevant, new situations (Berk 2003).

Once children exhibit what computer scientists would call *feature-vector analysis* and what psychologists would call *generalization*, the psychology of learning and development is no longer useful for this research. Clearly, machine learning research must eventually tackle the problem of automatic feature selection because it is a vital part of learning and intelligence. However, that is well out of the scope of this research.

Many psychologists have developed theories of child development and have studied young children, but a much smaller number of psychologists have based their work and theories on **neonates**, newborns less than 6 months old (see [Berk 2003] for an excellent textbook on child development). Perhaps most famously, Freud's theories largely hinge on a person's childhood experiences, but Freud did not study children himself (Cohen 2002).

The reasons that fewer people choose to study neonates are usually practical. Neonates are not thought to understand language and therefore cannot have instructions explained to them. Neonates are undergoing a period of extremely rapid development. Assembling a representative sample of infants of the same age can prove to be logistically difficult because even a difference in age of a week can represent a significant difference in cognitive ability. Additionally, describing and quantifying neonate behavior can be very difficult because of neonates' short attention spans and strong reflexive behaviors (Berk 2003).

This is not to say that no one has done significant work on neonates. Work has been done in the areas of understanding neonatal reflexes, neonatal classical conditioning learning capacities, neonatal operant conditioning learning capacities, and habituation and recovery studies, which demonstrate the extent to which infants can categorize or generalize. An overview of these areas of research is presented here.

Neonatal reflexes: A newborn's reflexes are the first signs of organized behavior. Some of these reflexes persist into adulthood (*i.e.* eye blinking in response to bright light or a puff of air in the face, sucking on an object placed in the mouth) while others disappear a few months after birth (*i.e.* rooting, the reflex which causes a neonate to turn its head in the direction of stimulation on the side of the face). Most reflexes are believed to serve an adaptive purpose. For example, the grasping reflex is thought to help babies hang on to their caretakers when carried and the swimming reflex is thought to help babies stay afloat long enough to allow for rescue should they fall into water (Berk 2003).

Brazelton's Neonatal Behavioral Assessment Scale (NBAS): Because neonatal reflexes are so strong and appear to be universal, assessing them at birth can be a very useful diagnostic tool for doctors. If a reflex response is abnormally weak, it may signal a problem with the development of the nervous system. T. Berry Brazelton has studied neonatal reflexes for most of his career and developed a scale that can be used to assess the health of neonates (Brazelton & Nugent 1995). In addition, this assessment has proved to be a useful tool for educating mothers about how to respond to their neonate's needs (Eiden & Reifman 1996).

Classical Conditioning in neonates: Neonatal reflexes force reactions to the environment, but neonates also use them to learn about stimulus-response relationships. Classical conditioning provides an important way for neonates to begin to make sense of their environment and helps them anticipate events in their world. Neonates can quickly learn relationships important to their survival through classical conditioning. In fact, the ability to learn through classical conditioning is present from birth.

In classical conditioning, an unconditioned stimulus is a stimulus that produces a reflexive, or unconditioned, response. An example of this would be the neonatal reflex of sucking on a nipple when it is presented. The nipple produces the sucking reflex and the baby takes in milk. If a neutral stimulus, like gentle stroking on the head, is introduced consistently during feeding, neonates will quickly learn that head-stroking and feeding go together. After a short period of conditioning, a matter of hours, a neonate will demonstrate the conditioned response of sucking if presented with the newly conditioned stimulus of head-stroking (Blass, Ganchrow, & Steiner 1984).

Operant conditioning in neonates: Operant conditioning is a kind of learning that takes advantage of spontaneous behavior. When a spontaneous action is followed by a reinforcing stimulus, it changes the probability that the spontaneous behavior will occur again. For example, if a young infant "smiles" inadvertently when a parent is watching, the child will most likely receive a great deal of reinforcement in the form of attention from the parent. Enjoying the attention, the infant is more likely to repeat this behavior in the future.

As newborns, there is very little that babies can control, but they are still capable of operant conditioning. In the first weeks of life, head turning and sucking are about all that babies can voluntarily control (Berk 2003). Sucking has been used to study infants' ability to categorize sounds. In one experiment, a tape of the mother's voice was played when babies sucked on a non-nutritive bottle at a given speed. The tape stopped when the babies stopped sucking on the bottle or when their sucking pattern deviated from the desired speed. Neonates demonstrated the capacity for learning through operant conditioning by learning to suck at the speed required to hear the mother's voice.

Additionally, neonates demonstrated that they preferred to hear their mothers' voice over other adult voices by learning a second sucking pattern (Floccia, Christophe & Bertoncini 1997).

By three months of age, infants can also control leg kicking. Studies by Shields and Rovee-Collier (1992) have used leg kicking to investigate infant memory. In their experiment, infants were placed in a crib and had one leg attached to a mobile. The infants learned to kick their leg in order to see the mobile move. This behavior is an example of learning through operant conditioning.

The studies by Shields and Rovee-Collier were focused on infant memory and generalization ability. The study found that the youngest infants (2-3 months old) could repeat the kicking motion the next day only if they were placed in the same crib. If anything were altered in the environment (*i.e.* the color of the crib mattress), the youngest infants would take as long to discover the kicking motion as they did in the first learning phase. However, if they were placed into the exact same setting, the kicking motion was repeated almost immediately. This study showed that neonates were relying heavily on all aspects of the environment for learning and memory.

The studies of neonatal reflexes, classical conditioning, and operant conditioning learning provide a wealth of information about *what* neonates are capable of learning, but they do not provide a unified theory of *how* neonates are learning. Jean Piaget provides the strongest and most widely accepted general theory of how and why certain abilities emerge in neonates. He also produced an extensive theory of development which he applied to older children, adolescents, and adults.

His theories are not without critics. Many people have questioned the specific ages at which Piaget claimed certain capacities emerged (Miller 1993; Siegler & Ellis 1996;

Bjorklund 2000). However, his theory offers a general road map that still proves very useful. His findings have served as the starting point for almost every contemporary perspective on child development in the past 50 years (Berk 2003). It is to Piaget's work that one must look, then, to try to understand what is happening in the neonatal mind and how the neonate approaches the world.

2.1.1 The Theory of Jean Piaget

The Swiss psychologist Jean Piaget developed his theory of child development in the late 1920's. Over a long career, he studied many different aspects of child development and behavior. His early writings were based on careful observation of his own children. He later established a center for study in Geneva and became one of the most respected authorities in the field of psychology.

His work has had a tremendous impact on the way educators and psychologists view children and their development. One of his major contributions was to portray children as active, curious knowledge seekers. His major ideas about child development are outlined in this section. Although many of the particulars about when certain behaviors emerge and if development occurs in rigid, fixed stages have been disputed, his work still provides a very useful guide to child development (Berk 2003). In this research it is assumed that Piaget's fundamental ideas are sound and that his work provides a reasonable foundation for the solution used in this research.

Assumption 01: The fundamental ideas of Jean Piaget provide a valid psychological basis for this research.

In his work, Piaget dealt with the *stages* of development that all children must pass through as they mature and the *schemes* children use to codify their abilities and understanding of the world around them. In this section, the stages of development are discussed first and then schemes are explained in more detail.

Stages of Development. Piaget believed that children went through four major stages of development: sensorimotor, preoperational, concrete operational, and formal operational. These four stages explain how an infant moved from reflex-based behaviors into the mature, abstract, and logical thought of adolescence. Piaget's theory has three major properties. First, it is a general theory, meaning that it assumes that all aspects of cognition develop in an integrated fashion. Second, the four stages are invariant, meaning that children must pass through each stage in order and no stage can be skipped, although it is possible that later stages may never be attained. Third, the stages are said to be universal, meaning that they should apply to all children everywhere (Piaget, Inhelder & Szeminska 1948/1960).

Piaget thought that development must be rooted in biology and was the result of the brain developing and becoming increasingly adept and analyzing and interpreting experiences common to all children. Piaget accounted for individual differences in rates of development by pointing out that genetic and environmental factors held significant sway over the speed with which children would move through each phase (Piaget 1929/1928).

Piaget's stages deal largely with the way in which a child organizes structures for making sense of the world and how this organization changes over time. These organized structures are called **schemes**. In the early stages, schemes are physical routines that the infant has mastered, such as voluntarily reaching for objects of interest or making voluntary vocalizations. As the child matures, the schemes begin to include mental representations of the world. The understanding of object permanence is one of the earliest demonstrations of a mental representation. The shift from purely physical schemes to mental representation schemes is a major one in Piaget's theory and marks the transition from a physical exploration of the world to a cognitive and mental exploration of the world (Berk 2003).

The four stages are broadly outlined here. For the purposes of this research, the first stage is most important and is discussed in more detail later. The first stage, the **sensorimotor** stage, generally extends from birth to 2 years. During this stage, infants are focused first on their own bodies and then later on objects in their immediate environment. In the sensorimotor stage, intentional goal-directed behaviors emerge and, later, children begin to explore the properties of objects by manipulating them in interesting ways. By the end of the sensorimotor stage, children begin to show evidence of mental representations by coming up with sudden solutions to problems and by engaging in make-believe play.

With the emergence of mental representations, the child enters into the **preoperational** stage that generally lasts from two to seven years of age. In the preoperational stage, mental representations grow rapidly and become increasingly more complex. Make-believe play becomes more detached from the immediate environment. For example, children in the preoperational stage will happily pretend that a cup is a hat, but children still in the sensorimotor phase will refuse to pretend that a cup is anything un-cup-like (O'Reilly 1995).

Another major milestone accomplishment of the preoperational stage is the ability to understand the relationship between a symbol and the real world. In a study by DeLoache (1987), children were shown a physical model of a room and watched as an adult hid a miniature toy (a small Snoopy) somewhere in the model room. The children were then asked to find the toy (a bigger Snoopy) in a real room. Children younger than three years could not accomplish this task, but older children were quite capable. The younger children seemed to be struggling with the idea that the model room was not only a toy but also a symbol of another room (Berk 2003).

The understanding of symbol-to-real-world relationships grows more sophisticated during the preoperational phase and, by seven years of age, most children are quite adept at following simple maps and understanding that one abstract object can stand for another object in the real world. During the preoperational stage, children begin to form hierarchical categories and can break categories into basic-level categories (*e.g.* chairs, tables, sofas), subcategories (*e.g.* kitchen chairs and desk chairs), and superordinate categories (*e.g.* furniture).

Even though the preoperational stage is typified by a dramatic increase in a child's mental representations, Piaget claimed that a child's thoughts were not very logical or organized. Organized thought emerges during the **concrete operational** stage, which generally lasts from 7 to 11 years of age. During this time, children's mental representations become more abstract and start to take on logical forms. One of the classic examples of this in Piaget's theory is the idea of conservation of volume. Children in the preoperational stage will watch water poured from a wide shallow container into a tall skinny container and claim that there is now more water in the tall container. Piaget explains that children in the preoperational stage can only mentally represent one dimension of an object at a time. During the concrete operational stage, children expand their mental representations to include multiple dimensions of a conservation problem. During this phase, children's hierarchical categories become more flexible and they become more aware of classification relations (Ni 1998).

The last phase of development is the **formal operational** stage. Most children enter this stage around 11 years of age. Whereas a concrete operational child can only operate on objects in reality, formal operational children are able to operate on other operations. Concrete things and actual events are no longer required as objects of thought (Inhelder & Piaget 1955/1958). Adolescents become able to reason in a scientific manner, moving from a general theory to a hypothesis and then are able to test the hypothesis in an orderly fashion. When children are in the concrete operational stage, they can only come up with the most obvious predictions about how to solve a problem. When these fail, they are unable to think of other alternatives and fail to solve the problem (Berk 2003). Another major characteristic of the formal operational phase is propositional thought. Adolescents can evaluate the validity of verbal or written statements without having to check them against evidence in reality.

Development of Schemes. Throughout all four stages of Piaget's theory, children are using schemes to codify their knowledge of their own abilities and their knowledge of the world around them. These schemes start out as physical routines and become mental representations. Schemes are incorporated into the thought process by the use of one of four mechanisms: assimilation, accommodation, equilibrium, and organization. A cornerstone idea about Piaget's schemes is that they are all acquired and built through direct interaction with the environment. Only through interaction with the environment is there any impetus to learn, acquire new behaviors, or change existing ones. In recent years, Piaget's ideas of schemes and equilibration have been criticized for being too vague. In fact, the most specific definition of a scheme is "a very broad way to denote organized behavior" (Evans 1973). It is not clear exactly *what* is being assimilated and accommodated, but because Piaget was focused on broad changes in behavior his ideas remain useful tools for discussing developmental changes (Miller 1993, Siegler & Ellis 1996).

Assimilation. During assimilation, existing schemes are used to interpret the external world. An 18-month-old girl with a small, but functional, vocabulary may refer to every animal she encounters as "doggie". Likewise, an even younger child may have mastered a scheme for holding a small object in his hand and dropping it. It is a simple release of the fingers. The child may enjoy dropping any small object he can find, but they are all manipulated in the same way.

Accommodation. In accommodation, new schemes are created or old schemes are adapted into new and different ones when it becomes apparent that the current schemes do not adequately capture or describe the environment. Recall the girl mentioned in the previous example and imagine that she is now two and a half years old. She has now expanded her vocabulary considerably and has many more names for different animals (doggie, kitty, horse, cow, duck, bird, *etc.*). Imagine her first trip to the zoo. When she first encounters a platypus, she may be seen to stare for a few moments before pronouncing the strange new animal to be a "duck-doggie". In this case she has noticed that none of her current schemes are quite appropriate for the new animal and so she accommodates an existing scheme and creates a new one for the platypus. As the young boy with the dropping scheme grows up, he will learn to accommodate his dropping scheme for different objects based on their shape and texture. He will soon further accommodate his dropping scheme by changing the force with which certain objects are released. To the great joy of his parents, he has now adapted a new scheme: throwing.

Equilibrium. Equilibrium describes periods of time when a child is assimilating more often than accommodating. These are periods of relative cognitive ease for the child. Most things fit easily into existing schemes. This is contrasted with periods of disequilibrium where a child's schemes are largely not effective for the environment and the child experiences "cognitive discomfort" (Berk 2003). When in disequilibrium, a child must shift the balance away from assimilation and towards accommodation. Piaget also used the term equilibrium to refer to the constant ebb and flow of assimilation and accommodation.

Organization. Although assimilation, accommodation, and equilibrium are all processes that require interaction with the environment, organization is a strictly internal process. In Piaget's view, organization was an important part of equilibrium. After acquiring new schemes, children relate schemes to each other and create a broad network of relationships (Piaget 1936, 1952).

Sensorimotor Stage. In Piaget's theory, the sensorimotor stage represents the period of greatest cognitive development in children. Due to the rapid change during this stage, the sensorimotor stage is broken down into six substages. For the purposes of this research, the first three substages are of the greatest interest and will be discussed in more detail. The 6 substages are summarized below:

1. *Reflexive schemes* (birth to 1 month): behaviors limited to newborn reflexes
2. *Primary circular reactions* (1-4 months): simple "motor habits", centered on the infant's own body
3. *Secondary circular reactions* (4-8 months): actions intended to repeat interesting experiences and effects in the surrounding environment
4. *Coordination of secondary circular reactions* (8-12 months): goal-oriented behaviors, ability to find some hidden items
5. *Tertiary circular reactions* (12-18 months): exploration of the properties of objects

by operating on them in novel ways

6. *Mental representations* (18-24 months): ability to internally depict events and ideas, appearance of sudden solutions to problems

For the purposes of this research, substages 1 through 3 are of the most interest. During the first 6 to 8 months, children are heavily context dependent. This is to say, the neonates are not capable of much generalization over different contexts (Shields & Rovee-Collier 1992). Although they are active investigators of their own bodies and surroundings, they are not yet capable of complex problem solving. If this idea is extended a bit farther, it is possible to say that neonates are performing only a minimal amount of feature analysis and therefore not generalizing from one situation to the next. Pulaski (1980) describes the behavior of a neonate assimilating his first scheme:

> "But sometime during the first month the baby's fist accidentally finds its way into his mouth. By reflex action he begins to suck on it and apparently finds this activity satisfying. At any rate, the baby repeats this action over and over again until he learns to bring his fist to his mouth at will. After that, not only his fist, but also everything else he grasps will find its way into his mouth. 'For him', says Piaget, 'the world is essentially a thing to be sucked'." (p. 20)

In substages 1 through 3, Piaget outlines how newborn transitions from having nothing but reflexes that dictate all responses to the environment to having a repertoire of simple schemes with which she can begin to explore her world. This period is also one of rapid physical changes and brain development. For the purposes of this research, it is not possible to mimic the changes, which expand the number of responses an infant may make to her environment. In this research, the learning agent need only learn the correct transform of stimulus to response.

Defintion 01: Learning agents are endowed with a fixed number of responses (classes), *i.e.*, the cardinality of the response range (intent) is fixed.

Piaget's theory also discusses the idea of organization whereby schemes are rearranged and related to one another. Modern adaptive learning methods (*e.g.*, Collective Learning Systems) might be able to achieve this through the use of hierarchical systems. However, this research focuses only on a single learning agent, albeit an agent with arbitrarily complex domains and ranges.

Defintion 02: Only single-layer, non-hierarchical learning systems are considered in this research.

This research focuses only on a single-layer, non-hierarchical learning system, and so the earliest stages of development are the most interesting because it is assumed that a newborn's brain and behaviors are the least organized and developed. Although statistical learning and machine learning methods cannot begin to replicate the complexity of the brain, studies of and theories about older children and adults often deal with many layers of judgment, categorization, analysis, social interaction, and language that would interfere with the main question of this research: is it possible to apply existing solutions to new problems without doing any feature analysis? By examining the early stages of development, it is assumed that evidence might be found for such a mechanism in the brain.

Substage 1 - Reflexive schemes (birth to 1 month). During the first substage, neonates are acting entirely based on reflexes. Piaget viewed these reflexes as the building blocks for sensorimotor development. During the first month, a baby will always produce the same responses to the same stimuli. This repeated action gives consistent, ordered feedback to the baby. For example, an infant will suck on anything placed in his mouth and always turn his head in the direction in which his cheek is rubbed.

Substage 2 – Primary circular reaction (1 to 4 months). During the second substage, infants begin to gain some voluntary control over their actions. Many reflexes are still powerful, but infants discover that they can repeat some action that first occurred randomly. These actions are almost entirely centered on their own bodies. Piaget referred to these as *primary circular reactions*. The word "circular" is used because infants will often repeat the same action many times in succession as they assimilate a new scheme. For example, an infant may accidentally move her fist to her mouth. By reflex, she starts to suck on it. Finding this sensation pleasing and soothing, she will attempt to bring her fist to her mouth again the next time it passes into her field of vision (Berk 2003, Pulaski 1980).

Substage 3 – Secondary circular reactions (4 to 8 months). In the third substage, children begin to exhibit *secondary circular reactions*. These differ from primary circular reactions because they are moving away from the child's body and outwards towards her environment. By the fourth month, children have acquired a few schemes from their primary circular reactions, for example, voluntary sucking, grasping, and leg kicking. Once the infant has some level of control over her body, she can begin to explore her environment.

She can swing her arms towards a mobile suspended over her crib. When she makes contact with the mobile, she can watch the mobile move with great interest. Piaget describes this sort of outward directed behavior as "the first outline of what will become classes or concepts… perceiving an object as something ‘to shake’, ‘to rub’, *etc.*” (Piaget 1952).

In the third substage, infants accommodate their body-focused schemes into object-focused schemes. When an infant encounters a new object in her environment in the third substage, she will attempt to apply one of her schemes to it in an attempt to discover if this object is a good object to rub, or to suck, or to kick. With secondary circular reactions, infants are primarily assimilating new experiences and objects into existing schemes. This is not to say that no new schemes are accommodated during the third substage, but in the third substage, infants are not spending much time experimenting with new schemes, but rather *they are applying known schemes to new objects and situations.*

This idea of applying known solutions to new problems, without regard for particular features of an object is exactly the kind of solution this research seeks to implement.

2.2 Applications of Piaget’s ideas to Artificial Intelligence

Piaget’s ideas have attracted the attention of the machine learning and artificial intelligence communities. This section presents a brief overview of the application of Piaget’s theories to machine learning and artificial intelligence as they relate to this research.

2.2.1 Drescher (1991)

Gary Drescher used Piaget’s ideas of schemes and constructivist learning in a massively parallel neural network-like implementation. Starting with a few basic schemes, the learning agent expanded its knowledge by exploring its environment. His work showed some very interesting results. The learning agent was able to successfully construct new schemes and succeed at increasingly complicated tasks.

One of the important aspects of Dresher's work was the ability to create new schemes, or *classes*, as learning increased. Although finding a reasonable and generalized way to do just that is an open topic in Collective Learning Systems, the learning paradigm used in Drescher's research, is outside the scope of this research. As previously defined in Definition 01, the possible schemes or actions remain fixed for all of the treatments in an experiment. See Section 2.5 for more information about Collective Learning Systems.

In Dresher's work, schemes were expressed as propositional logic. This research focuses on Collection Learning Systems, which do not deal with propositional logic. In addition, Drescher developed a specialized parallel processing architecture for his learning agent, similar to an artificial neural network (ANN). Because this research only considers Collective Learning Systems (CLS) theory, it is not possible to extend any conclusions to ANNs or any other machine learning paradigm.

2.2.2 Birk (1996); Birk & Paul (2000)

Birk (1996, 2000) was interested in combining Dresher's approach to schemes with genetic algorithms. Birk (1996) applied his approach to a computer model of a robot arm and in (Birk & Paul 2000) applied similar ideas to a physical robotic arm. As has been noted earlier, this research is restricted to CLS theory and does not attempt to extend to a robotics or evolutionary programming application.

A common theme between this research and Birk's is the desire to design learning algorithms that would be independent of any particular problem. Birk was interested in using evolutionary programming to generate the preconditions for using schemes but this addition was not dependent on the robot model nor on the task set for the learning agent. Although Birk succeeded in developing a new system which was problem independent, the *stimulus-response rules*, Birk's name for what Drescher called *schema*, are entirely context dependent, and no attempt is made to translate the lessons learned in one context to another context. This research seeks to leverage learned knowledge to improve performance on new problems.

2.2.3 de Jong (1999)

de Jong was intrigued by Dresher's ideas, but was most concerned with developing a machine learning agent that could observe and learn about an environment that changed not just in response to an agent's action. de Jong's work focused multi-agent systems and inter-agent communication. The major goal for de Jong's learning agents was to decide when to take advice from the agents' personal percepts or information coming from other agents. That is, in a multi-agent environment, each individual agent had a limited percept region or "field of view". All agents broadcast their perceptions to all other agents at all times. Individual agents then had to learn when information coming from another agent might be useful.

Defintion 03: This research does not consider multi-agent systems.

Although this research does not consider multi-agent problems, it looks as if de Jong's approach could be recast as a single agent problem. If it were, de Jong's work might provide the solution to the problem this research is investigating: a way to apply knowledge from another situation to a current, possibly unrelated situation; however, this analogy falls apart upon closer inspection.

In the multi-agent situation presented by de Jong, considering information from another agent adds more information to the feature vector. Therefore, either the signals from other agents and an individual's percepts must be combined into a new stimulus or an analysis of the feature vectors is necessary for this kind of learning. If the first option is chosen, then de Jong's method does not provide a solution because adding more stimuli does not provide a way to communicate knowledge from one stimulus to another. If feature analysis is necessary to make use of the increased knowledge, then this is not an appropriate solution because this research mandates an improvement to learning that does not rely on the analysis of feature vectors in any way.

2.3 The Pigeonhole Principle

The previous section discussed the application of Piaget's theories of child development to machine learning. Psychology is not the only field that can serve to inform machine learning. Indeed, from discrete mathematics, the concept of the Pigeonhole Principle provides some useful insight for the problem considered in this research.

The Pigeonhole Principle states that if there are n pigeons and, at most, $n-1$ holes and all the pigeons are in holes, then there must be at least one hole with more than one pigeon in it. To put it more formally, consider the function f where $f: S \to T$ and where S and T are finite sets satisfying $|S| > k|T|$. Then at least one of the sets $f^{-1}(t)$ has more than k elements. (Ross and Wright 1988, p207).

Although this may seem obvious, the Pigeonhole Principle is a very useful tool for problem solving. To put it in the context of adaptive learning, an agent's stimulus domain is almost always significantly larger than its response range. A minor exception to this may occur in the early stages of learning if an agent is dynamically allocating its domain. Although the agent starts out with a domain size of zero, stimuli will quickly be added during training and in a short time the size of the domain is larger than the size of the range.

Assumption 02: When training is complete, the final cardinality (extent) of the stimulus domain is greater than or equal to the cardinality of the response range.

Given that there is a greater number of stimuli than responses, the Pigeonhole Principle can be used to prove that at least one response must have more than one corresponding stimuli. This proof gives further credence to the idea of trying a solution that works for one problem to solve a new problem in a new context.

Assume that there is only one correct response for each stimulus. If an agent picks random choice from *n* responses, then there is a $\frac{1}{n}$ probability of picking the correct response. By selecting a response, ω, that is known to work in another context, the agent has a slightly higher probability of picking the correct response. Because the number of stimuli is greater than n, the probability of ω being the correct response is $p(n) \geq \frac{1}{n}$. On the other hand, if ω is not an effective response for the new stimulant, very little has been risked because a random choice was also very likely to be wrong.

2.4 Reinforcement Learning

Reinforcement learning is a machine learning paradigm that encompasses an entire class of approaches. Generally, the learning agent is in contact with an environment through its perception of the environment and the actions it can perform in the environment. The learning agent receives information about the current state of the environment and then attempts to pick the most appropriate action. When the agent performs an action, the state of the environment changes and the agent receives feedback about the value of the action and the new state of the environment. The agent's main goal is to maximize the return of its value function in the long term (Barto & Dietterich 2004).

2.4.1 Environment

There are few requirements made on the environment in reinforcement learning. Most importantly, the environment must respond to *each* action of the learning agent. The environment does not need to be fixed. In fact, one of the strengths of reinforcement learning is that learning agents can adapt over time to a changing environment (Kaelbling, Littman, & Moore 1996).

2.4.2 Agent's Perception of the Environment

The learning agent must be able to perceive its environment in order to make an intelligent decision. The environment, at any given time, is represented by some input function *I*. It is generally assumed that *I* gives a full representation of the environment, but that is not a requirement. In most real-world problems, it is not possible to have complete knowledge of the environment; however, reinforcement learning problems are often constructed as Markov Decision Problems and therefore a complete view of the environment is optimal (Sutton & Barto 1998).

2.4.3 Reinforcement Signal

The goal of the learning agent is to optimize performance based on an unknown (from the learning agent's perspective) reinforcement signal, usually represented as some scalar valued function. This signal may be stochastic or deterministic and it may change over time. Learning agents are also interested in optimizing the long-term value of the reinforcement signal; it must learn to balance large immediate rewards against larger projected rewards in the future (Kaelbling, Littman, & Moore 1996, Bock 1993).

A learning agent's goal may appear to be simple function optimization, but the *learning* occurs in the sense of long-term memory. Barto & Dietterich (2004) give the example of trying to find good cell phone reception in an outdoor setting. A person trying to find good cell phone reception may wander around looking at the signal bars or asking the person on the other end of conversation “Is this better? How about now?”

Once a good spot is found, the person with the cell phone is likely to make a note of it, especially if it is on his regular travel route, and return there directly the next time he needs to make a call. Likewise, a learning agent in a reinforcement learning system may struggle to find a state that optimizes the reinforcement signal for the problem or some subset of the problem; however, once that state has been found, the learning agent can return there directly.

2.4.4 Agent Actions

When the learning agent chooses an action, its action affects the state of the environment. Because the goal of the learning agent is to maximize its long-term reward and because the reward function is based on the current state of the environment, it is possible that any given action will affect more than the immediate reward. If the agent’s action moves the environment into a more favorable position, this could affect the rewards the agent receives for the rest of its learning cycle. Likewise, a particularly bad move could negate the possibility of the learning agent ever optimizing its reward function (Heckman 2004).

2.4.5 Policy

The learning agent is attempting to find a policy, *p*. that maximizes the reinforcement signal. This policy can change over time to accommodate the feedback from the reinforcement signal. There is no standard form that this policy takes in reinforcement learning. It can be a simple look-up table or a complicated algorithm for computing the next appropriate response (Barto & Dietterich 2004; Sutton & Barto 1998).

2.4.6 Unsupervised Learning

It should be noted that reinforcement learning is not the same as supervised learning (Barto & Dietterich 2004; Kaelbling, Littman, & Moore 1996). In supervised learning, agents are often presented with a series of correct input-output pairs during training. They organize that information in some way, and the *organization* is tested. In reinforcement learning, feedback from the reinforcement signal is presented *online*, during learning instead of beforehand. Additionally, an agent in a reinforcement learning system does not receive corrective information about which response it *should have* chosen in a previous state. If it did, this could be considered a form of supervised learning.

2.4.7 Model of the Environment

In some cases, agents in reinforcement learning systems may be provided with or may develop a model of the environment. This model mimics the behavior of the real environment. The model may be provided to the learning agent, or the agent may develop and improve the model over time. This model can be useful to the learning agent because the agent can "imagine" what the outcome of a given response is likely to be. This model is not always a part of a learning agent, as the agent's policy eventually comes to dictate the appropriate response to the input signal (Heckman 2004).

2.4.8 Practical Uses of Reinforcement Learning

Reinforcement learning has been applied to real-world problems because a flexible learning system is a very attractive solution when the environment and the requirements of the system are changing. Areas of practical research that have used reinforcement learning include robotics, computer games, industrial manufacturing, credit card risk assessment, and predicting user's internet browsing behavior (Barto & Dietterich 2004; Kaelbling, Littman, & Moore 1996).

2.5 Collective Learning Systems Theory

Collective Learning Systems Theory is a statistical learning paradigm that has much in common with reinforcement learning. It was first developed by Bock in 1976 and has been refined continuously over the years (Bock 1976; Bock 1985; Bock *et al.* 1990; Bock 1993; Bock & Riopka 2000, Bock & Heckman 2002). This section presents an overview of a Collective Learning System and its major components. All definitions and citations are from (Bock 1993) unless otherwise noted.

2.5.1 Introduction

In a Collective Learning System (**CLS**), one or more learning agents interact with their environment, encountering different stimuli and producing responses from the repertoire of possible actions. A learning agent receives periodic evaluations of its performance and uses these evaluations to compile statistics about the effectiveness of its actions. The goal of a learning agent is to converge to the correct probabilistic mapping of stimuli to responses.

In a CLS, the learning agent is called a Collective Learning Automaton (**CLA**). A CLA learns the appropriate response for each stimulus by selecting responses at random until one emerges as statistically optimal, guided by feedback from an evaluating environment. This process of learning through positive and negative feedback is known as the algedonic cycle (Beer 1966). Learning agents try to maximize positive outcomes and minimize negative ones.

> The process of learning, in all arenas, must take place over time and must involve some set of rules and a means to evaluate performance. No matter what the task, be it playing checkers or sustaining a successful relationship with a spouse, the previous statement holds true. The rules may change over time and expectations and evaluations may differ with experience, but they are still there. For this reason, it is possible to frame all learning tasks as

> games, where a **game** is defined as a set of rules governing a contest among several players in a specified environment (Bock 1993).

For the purposes of conducting canonical experiments with a CLS, the associated game is usually defined in this sense, as well as in the colloquial sense. There is further discussion of the game that is used for this research in Section 3.4, but this definition of a game quoted above will serve for this discussion.

Some games require a great deal of knowledge about how one came to be in a certain state in order to make a good decision about what to do next. The number of states in the past which a player is allowed to consider is called the **backward context**. Likewise, the number of potential moves into the future a player is allowed to search is called the **forward context**.

Being able to predict the future is a very powerful tool. Heckman (2004) implemented and explored the effect of forward context in CLS. Although forward context is useful, it is not necessary for all learning tasks. Thus, in order to limit the scope of the research, no games requiring forward context are used in the canonical experiments.

Defintion 04: CLAs will be implemented without forward context.

The CLA is embedded in an environment that sends a stream of stimuli to the CLA and periodically issues evaluations of the CLA's responses to these stimuli. The major feature of CLS that sets it apart from other reinforcement learning methods is that the CLA's performance is evaluated only *periodically*, based on the assumption that periodic evaluations from the environment are a more realistic model of the real world. For example, in school, followers receive feedback about their performance only a few times during the semester when they receive grades for their assignments. In a complex environment, it is usually impractical, and often counterproductive, for the environment

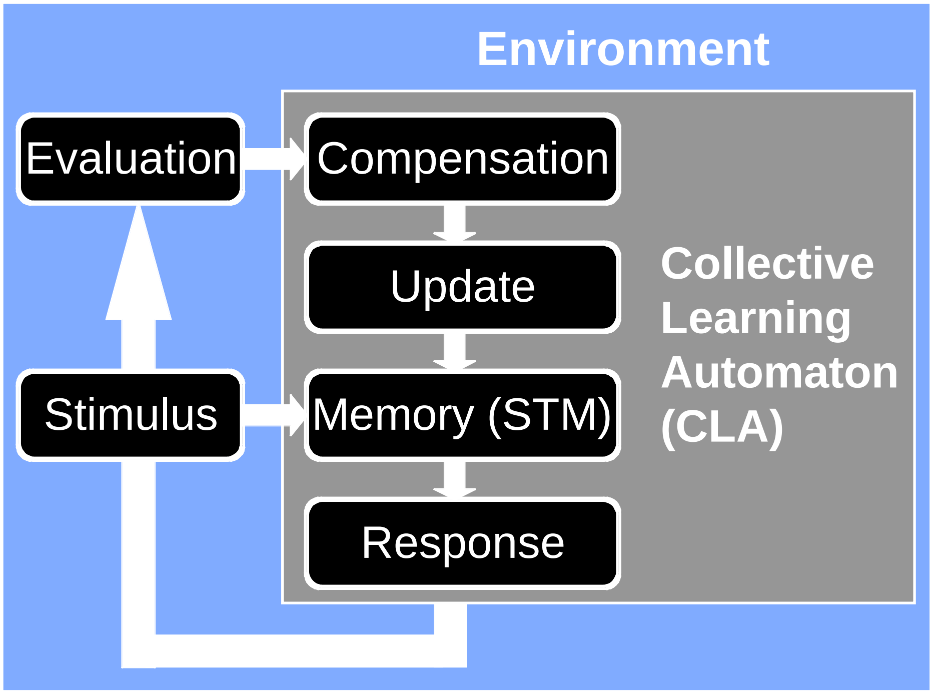


Figure 1: A complete Collective Learning System (CLS). The learning agent, the CLA, engages in the algedonic cycle to acquire knowledge about the environment, eventually eliciting correct responses to its stimuli.

to provide feedback for *each and every* action taken by the learning agent. Therefore, evaluations are usually **collective**. Figure 1 shows a CLS, including all the major component parts.

The environment encompasses everything outside the CLA. The environment provides a series of stimuli to the CLA. The CLA examines its memory, a State Transition Matrix (STM), and chooses a reponse to each stimulus according to its response-selection policy. The responses are returned to the environment. Depending on the game being played, the responses may or may not affect the next stimulus presented to the CLA. At some point, the environment provides an evaluation of the CLA's responses. The CLA receives the evaluation and interprets it according to a compensation policy. The compensation policy determines how the evaluation should be applied to the most recently chosen responses. Once an update has been calculated, it is applied to the statistics held in the STM. With the updated statistics in the STM, the CLA is ready to apply its new knowledge to the next series of stimuli presented to it by the environment.

2.5.2 Environment

The environment represents everything outside of the CLA. The environment provides stimuli to the CLA and evaluates the responses that come back from the CLA.

Defintion 05: The **environment** of a game includes the necessary space, matter, and energy to conduct the game.

The environment contains everything necessary for the CLA to learn. It generates the stimuli, provides the evaluations, and enforces the rules of the game. If the game is interactive, *i.e.*, if a response from the CLA affects the next stimulus or the state of the game, then the environment keeps track of the changes in the state of the game as well.

The environment is usually considered to be stationary, deterministic, and correct, but this is not a requirement. Many students of CLS theory have experimented with unreliable environments, because it is rare indeed that feedback in the real world is consistent and correct. Consider the confusing explanations flustered, unprepared parents may provide to children to such questions as “why is the sky blue?” or “where do babies come from?” Nonetheless, for the research experiments that are conducted under controlled conditions, the environment is deterministic and correct. Note that the environment is not required to be stationary. Learning to recover from changes in the environment (the game) is a very important skill and something that is considered in this research.

Depending on the game, the environment may or may not change its state based on the responses from a CLA. For example, if the CLA is learning to play solitaire, then clearly each response changes the state of the environment (or at least the representation of the solitaire game). On the other hand, if a CLA is classifying pixels in an image, then the responses do not change the state of the environment. The image remains the same, no matter what label the CLA assigns to a given pixel. In this case, the environment would keep track of the image and the pixel labels provided by the CLA.

There are no restrictions, in CLS, on the number of learning agents that may be contained in an environment. There is also no prohibition against different CLAs working in a competitive or cooperative manner; however, this research only focuses on environments which contain a single CLA. In other words, this research does not consider CLAs that interact with one another in any way.

Defintion 06: This research will not consider multi-agent systems.

2.5.3 Stimulus

The stimuli are provided by the environment. They are usually encoded as a vector of several features, but all that is truly necessary is that significantly different stimuli be distinguishable based on a unique identification code. In other words, the environment may present some quantized view of the problem domain. The more unique stimuli a CLA encounters, the longer learning will take; however, failure to provide enough detail may also lead to a failure to learn anything useful.

2.5.4 Evaluation

The environment is also responsible for periodically evaluating the responses from the learning agent. The learning agent is evaluated based on some evaluation function which generates an evaluation, ξ. The evaluation is usually a numeric value, but there is no reason to limit it to a number.

The evaluation function must be tuned for each game, but the purpose of the evaluation is to provide an assessment of the learning agent's performance that contains as little bias as possible. The environment evaluates the learning agent's performance after a series of interactions between the learning agent and the environment.

There are several terms that deal with the number of responses between evaluations. They are contest, collection length, and history. A contest is an instantiation of a game: one complete instance of a game played from beginning to end. The **collection length**, c, is the number of responses that the learning agent must make before it is evaluated by the environment. Lastly, the **history**, η_s, of a **stage** is the collection of interactions that have occurred since the last evaluation. The length of the history is the same as the collection length.

Defintion 07: A **contest** is an instantation of a game that is one complete instance of a game played from beginning to end.

Defintion 08: A **stage** is the phase of a contest that leads to an evaluation.

Defintion 09: The **collection length**, $\boldsymbol{c}$, is the number of responses that the learning agent makes before being evaluated by the environment.

Defintion 10: The **history,** $\boldsymbol{\eta_s}$, of a stage is the collection of interactions that have occurred since te last evaluation.

2.5.5 Collective Learning Automaton

The learning agent in a CLS is known as a Collection Learning Automaton (CLA). Its internal structures include a compensation policy, an update policy, and memory, which contains the state transition matrix and generates responses to the environment.

Defintion 11: A **Collective Learning Automata** (**CLA**) is a learning agent inside a Collective Learning System.

The CLA itself does not have any function aside from defining the boundaries of its internal structures. The internal structures are so important that they are discussed separately.

2.5.6 Compensation Policy

The compensation policy interprets the evaluations provided by the environment and generates a compensation, γ. The compensation policy determines how seriously the CLA takes its evaluations. This is *roughly* analogous to the limbic system, the system in the mammalian brain that controls emotional responses.

Defintion 12: The **compensation, $\boldsymbol{\gamma}$,** is an interpretation of the environment's evaluation by the CLA.

Most reinforcement learning systems just translate the evaluation directly into the reinforcement, but this is not the case in a CLS. Consider two people playing chess. If one is a young adult who has never played before, she is unlikely to be upset about losing her first game, nor is she likely to take her first win as a sign of mastery of the game. On the other hand, if her opponent is a chess master, he is not likely to take too seriously any wins against a novice, but would react very strongly to a deafeat by one. In the same way, a CLA can also adjust the impact of evaluations based on some internal function.

Compensation policies can be very flexible and vary with a number of factors, such as experience and collection length. That said, unless researchers are specifically interested in the effects of the compensation policy on learning (Heckman 2004), the compensation policy is usually fixed on one that seems to maximize learning during operating point pilots.

Defintion 13: The compensation policy is fixed.

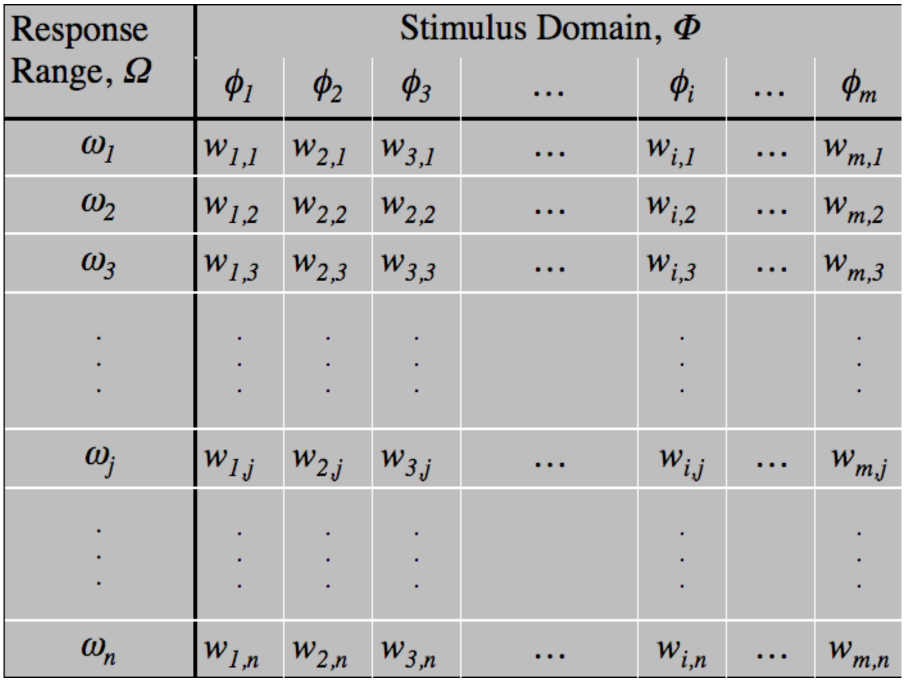
2.5.7 Update

The compensation policy interprets the evaluation from the environment for the CLA. The update function then takes the compensation, γ, and translates it into a numeric value that can be applied to the CLA's memory. The result of the update function is the update, υ, which is the amount that the memory elements in the history are to be changed.

Although the update is usually uniformly distributed over all the elements in the history, it can vary over the collection length. For example, the most recent events could be considered to be more important and therefore receive more of the update. Once the update has been applied to the weight matrix, the counts are incremented once for each time a stimulus/response pair appeared in the history. Then the history is cleared and the CLA is ready to handle another stimulus.

2.5.8 State Transition Matrix (STM)

The memory of a CLA exists in the State Transition Matrix (STM). The STM is a matrix which contains the knowledge about the correct mapping of inputs to outputs. The weights accumulate in the weight matrix over time as learning progesses. The STM can consist of more than one matrix for the purposes of computing the necessary statstics, but at a minimum there must exist one matrix, called the weight matrix. The weight matrix is the core knowledge matrix, containing numerical values which correspond to the strength between input/output pairs. Figure 2 shows a weight matrix for a hypothetical CLA.



Response Range, Ω	Stimulus Domain, Φ						
	ϕ_1	ϕ_2	ϕ_3	…	ϕ_i	…	ϕ_m
ω_1	$w_{1,1}$	$w_{2,1}$	$w_{3,1}$	…	$w_{i,1}$	…	$w_{m,1}$
ω_2	$w_{1,2}$	$w_{2,2}$	$w_{2,2}$	…	$w_{i,2}$	…	$w_{m,2}$
ω_3	$w_{1,3}$	$w_{2,3}$	$w_{3,3}$	…	$w_{i,3}$	…	$w_{m,3}$
⋮	⋮	⋮	⋮		⋮		⋮
ω_j	$w_{1,j}$	$w_{2,j}$	$w_{3,j}$	…	$w_{i,j}$	…	$w_{m,j}$
⋮	⋮	⋮	⋮		⋮		⋮
ω_n	$w_{1,n}$	$w_{2,n}$	$w_{3,n}$	…	$w_{i,n}$	…	$w_{m,n}$

Figure 2: State Transition Matrix (STM) The STM contains the knowledge about all stimulus/response pairs. The stimulus domain, Φ, represents all the stimuli that have been encountered by the CLA. The response range, Ω, is usually given at instantiation

Defintion 14: The **State Transition Matrix** (**STM**) is the knowledge matix that maps the input domain into the output range.

For the purposes of clarity, it is necessary to distinguish between the elements in the STM, which are stored permanently, and the information being passed between the environment and the CLA. The elements in the stimulus domain are called **stimulants** (ϕ) and the input elements from the environment are called stimuli. The elements in the response range are called **respondents** (ω) and the output elements from the CLA to the environment are called responses.

Defintion 15: a **stimulant** is a permanent element in the stimulus domain of the STM.

Defintion 16: a **stimulus** is an input from the environment to the CLA.

Defintion 17: a **respondent** is a permanent element in the response range of an STM.

Defintion 18: a **response** is an output element passed from the CLA to the environment.

Initially, all weights are set to a baseline number (usually zero or one). The weights are changed periodically according to the update policy as the responses are evaluated by the environment. For certain probability measures, it is useful to also use a count matrix which keeps track of the number of times a given stimulus/response pair was used.

2.5.9 Response Selection

As the weight matrix becomes populated, useful probabilities can be computed and used as the basis for choosing the next response. In order to choose a respondent to a given stimulant, the STM must be provided with a selection policy.

Defintion 19: A **selection policy** dictates how respondents are selected for a given stimulant.

Just as with the evaluation, compensation, and update policies, the selection policy may be fixed or varied over time and circumstance. There is no one uniform selection policy for CLS. Although there is no hard rule about which selection policy to choose, Bayes' Rule is often chosen as a default because, given independent classes (respondents), there exists no better classifier (Mitchell 1997). However, other options to consider are the maximum likelihood or a stochastic approach. In this research, maximum likelihood is the statistical basis of the clasifier.

It is not enough to simply choose the respondant with the largest probability. Some statistical tests must be used to decide if the largest probability is in fact significantly larger than the others. This is generally done by using the standard difference of two proportions to calculate the confidence that the largest probabilty is different than the other probabilities (the **reject confidence**) and then to calculate the confidence that the largest probability is different from the second largest probability (the **tie confidence**).

Defintion 20: The **reject confidence** is the confidence that, for a given stimulant, it is possible to reject the hypothesis that there is no difference between the largest posterior probability and the *apriori* probability.

Defintion 21: The **tie confidence** is the confidence that, for a given stimulant, the hypothesis that there is no difference between the largest posterior probability and the second largest posterior probability *can be* rejected.

If the reject and tie confidences are sufficiently high, then it is possible to consider the acceptance of the hypothesis that the largest posterior probability is significantly larger the other probabilities and that the associated respondent should be chosen.

Thresholds must be set in the selection policy to decide at what confidence level to stop exploring the response range and start exploiting the knowledge about a given stimulant/respondent pair. If the rejection confidence is not above the set threshold, then a random choice is made from all the possible respondents. If the reject confidence is high enough, but the tie confidence is not, then a random choice is made between the primary and secondary posterior probabilities. If the reject and tie confidences are above their respective thresholds, then the respondent with the highest posterior probability is chosen.

Recall that these confidences are all calculated using the standard difference of two proportions. The standard difference of two proportions assumes that the probabilities for a stimulant are normally distributed. This may or may not be the case, but it is an assumption must be made in order to calculate a confidence.

Assumption 03: the posterior probabilities of all stimulants are normally distributed.

Once a respondent is chosen, a response is generated and passed to the environment. If appropriate, the response may change the environment's state. Once the environment receives a response from the CLA, it sends back either an evaluation or another stimulus.

2.6 Applications of Collective Learning Systems

Collective Learning Systems are used to study different aspects of learning behavior (Armstrong & Bock 2005, Heckman 2004) in on-line learning situations. However, CLS can also be used for presupervised learning, classification without the algedonic cycle of learning. ALISA, a powerful tool for image and video processing that uses presupervised learning, was developed by Peter Bock (Bock 1998). In ALISA, the CLA is shown several examples of correct or preclassified stimulus/response pairs, so there is no need for a weight matrix. If all the examples are correct, then only the counts need to be recorded to calculate to necessary statistics.

Defintion 22: This research does not consider presupervised learning scenarios.

2.7 Maximum Likelihood

As has been discussed in earlier sections, most problems faced by a CLS do not meet the criteria for Bayes' Rule to provide optimal results, which requires that all classes (respondents) must be disjoint and exhaustive. The degree to which classes are not disjoint may be partially inferred from column-vector distributions for the feature vectors (stimulants) in the stimulus domain. If a stimulant can significantly trigger more than one respondent, then these respondents cannot be completely disjoint.

Most interesting classification problems specify classes that are patently and knowingly <u>not</u> disjoint. Consider the examples of lists of classes presented in Table 1, which are even *intuitively* far from disjoint, because they share significant features in common.

Table 1: Non-disjoint Classes. Rocks eventually become sand, words may belong to more than one part of speech, and mental health disorders often share many of the same symptoms.

Problem Domain	Possible Classes
pixels in images of outdoor scenes	{rock, sand, ice, water, clouds, sky, grass, deciduous trees, evergreens, dirt, flowers, snow, rabbits, *etc.*}
words with multiple meanings, such as “drive, father, gold”	{noun, verb, adjective, *etc.*}
mental health disorders	{depression, bipolar disorder, hypomania, anorexia, anxiety, phobias, *etc.*}

Given the possible and frequent shortcomings of Bayesian selection, it seems prudent to find another policy for response selection. Maximum Likelihood Estimation is a method developed by R.A. Fischer in 1950 (Fischer 1950). Its main purpose was to estimate the unknown parameters of a probability density function, PDF, that were most likely to have produced a given set of observed values. A concise definition of such classifiers comes from the Canada Centre for Remote Sensing:

Defintion 23: **Maximum Likelihood Estimation (MLE)** is “a statistical decision rule that examines the probability function of a [stimulus] for each of the classes [responses], and assigns the [stimulus] to the class [response] with the highest probability.”

Maximum Likelihood classifiers are commonly used to analyze images of common scenes on our planet, often acquired from airborne or spaceborne platforms (Short 2006). Bayesian classifiers perform very poorly in this context, as shown in the examples listed in Table 1. Classes found in natural scenes are rarely disjoint, often having very large intersections in feature space, which implies that Bayes’ rule will not perform anywhere near its optimal capability.

2.8 Primary Research Objective

After investigating five major areas of related work: psychological theories of child development, mathematical and statistical approaches, applications of child development theories to machine learning paradigms, reinforcement learning, and Collective Learning Systems, it is now possible to state a primary research objective.

The primary objective of this research is to develop and assess the effectiveness of a new selection policy for a CLA and the effect that the new policy has on the CLA's learning and behavior, subject to the following constraints:

- **The new selection policy must not require any feature analysis or comparison;**
- **Psychological theories of early learning provide a valid basis for developing a new selection policy;**
- **The CLS uses invariant evaluation, compensation, and update functions;**
- **Performance is measured on a simple solitaire game and then extended to a few representative applications;**
- **The CLA has a fixed response range (number of classes).**

CHAPTER 3: SOLUTION METHOD

3.1 Introduction to Solution Method

As stated at the end of the previous chapter, the objective of this research was to develop and assess the effectiveness of a new selection policy for a CLA and the effect that the new policy has on the CLA's learning and behavior, subject to the following constraints:

- The new selection policy must not require any feature analysis or comparison;
- Psychological theories of early learning provide a valid basis for developing a new selection policy;
- The CLS uses invariant evaluation, compensation, and update functions;
- Performance is measured on a simple solitaire game and then extended to a few representative applications;
- The CLA has a fixed response range (number of classes).

This chapter presents the details of the solution that was hypothesized to achieve this objective, as well as the details of the design of the goals and experiments necessary to accomplish in order to validate the research objective.

3.2 Tactic-Based Learning

The mechanism is called **Tactic-Based Learning (TBL)**. The main idea behind TBL is to bias the CLA in favor of solutions which it has already discovered to be effective. In order to do this, a CLA behaves exactly as it would without TBL until one stimulant-respondent pair emerges as a statistically significant and effective solution. At that point, the respondent becomes a tactic and all other stimulants select the tactic response, becoming a follower of the tactic. Each follower stimulant measures the effectiveness of the tactic for itself and remains a follower of the tactic if it is effective or abandons the tactic very quickly if it does not prove to be effective. In order to avoid becoming stuck in local maxima, when follower stimulants become somewhat confident, they abandon their tactics. When a follower stimulant abandons its tactic, it becomes an independent stimulant. An independent stimulant either becomes confident in a respondent, becoming a supporter of a tactic, or it loses confidence and reverts to follower status.

This is only a brief overview of the mechanism, but the general idea is to direct stimulants toward respondents that are known to be effective for other stimulants. By forcing stimulants to stay with effective tactics for some period of time, the CLA is biased towards an effective solution, but by allowing for some exploration later, the CLA avoids becoming stuck in suboptimal solutions.

3.2.1 Life Cycle of a Stimulant

A stimulant in a CLA that employs TBL assumes one of four roles at any given moment: **seeker**, **follower**, **independent**, or **supporter**. The transitions between these roles are regulated by two factors: the existence of tactics and the selection confidence of the stimulant. Figure 3 shows the life cycle of a hypothetical stimulant.

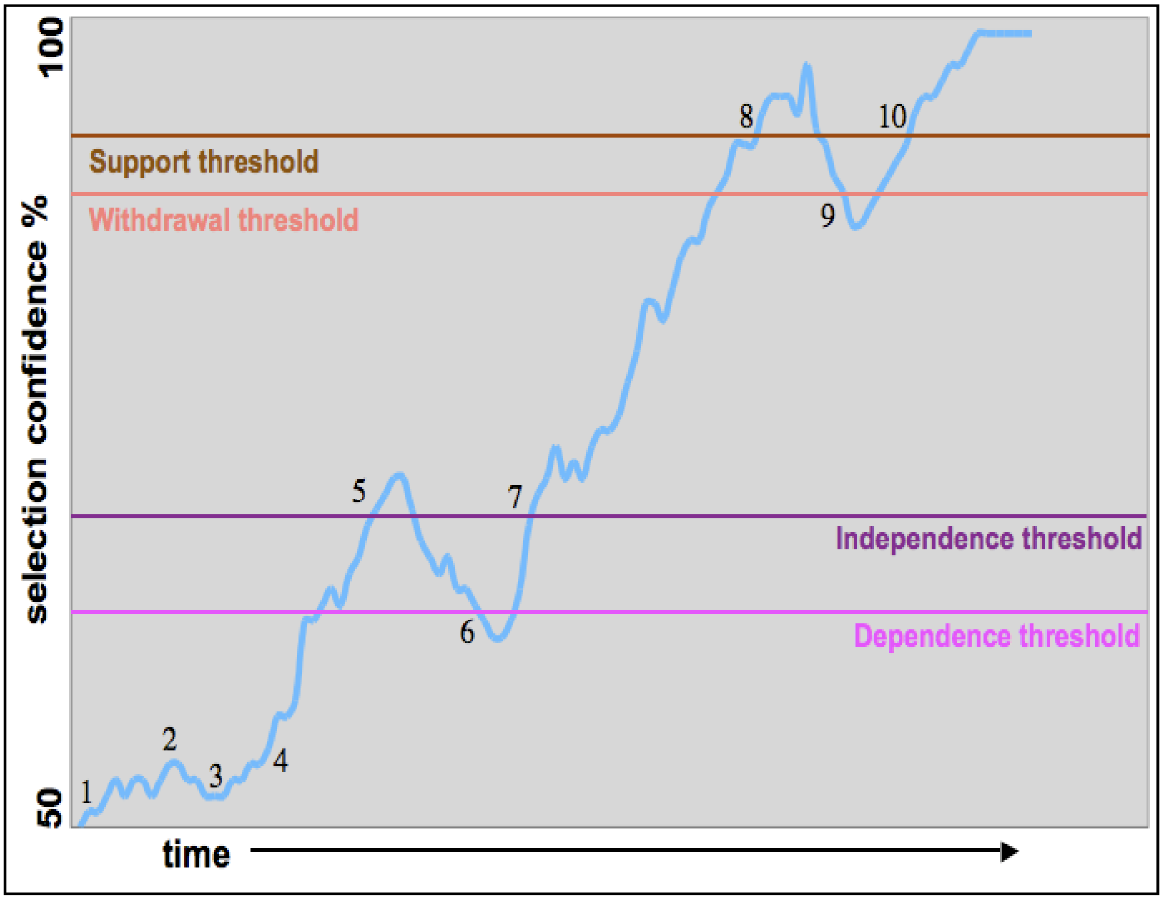


Figure 3: The Life Cycle of a Stimulant, ϕ. (1) The stimulant, ϕ, is encountered for the first time. In this example, there are not yet any tactics available in the STM, so ϕ is a seeker. (2) The first tactic appears in the STM and ϕ becomes a follower of it. (3) The first tactic is not effective, so ϕ abandons it and returns to being a seeker stimulant. (4) A second tactic appears in the STM and ϕ becomes a follower of this new tactic. (5) The stimulant's selection confidence quickly reaches the independence threshold. ϕ becomes independent and explores its response range. (6) ϕ does not become confident in a respondent and ϕ's selection confidence drops below the dependence threshold, causing ϕ to become follower again of its most effective tactic. (7) ϕ recovers its selection confidence and becomes an independent stimulant again. (8) ϕ has become confident enough to become a supporter of its own confident respondent. If this respondent is not already a tactic, a new tactic is available to other stimulants in the STM. (9) ϕ loses confidence and withdraws its support from its confident respondent. If ϕ was the only stimulant supporting that respondent as a tactic, the tactic is no longer available to other stimulants. (10) Once again, ϕ supports its confident respondent. This time ϕ remains confident.

Postulate 01: The **selection confidence** of a stimulant is the minimum of its tie and reject confidences (worst-case assumption).

When there are no tactics available, then all stimulants are seekers. When tactics exist, but none have proven effective for a given stimulant, then that stimulant is a seeker stimulant and follows the Standard selection policy.

Defintion 24: A **seeker stimulant** has no effective tactic and uses the Standard selection policy.

Defintion 25: A **tactic** is a respondent that follower stimulants may select.

In the beginning of the learning process, all stimulants are seekers, so how does the first tactic arise? The first tactic comes into existence when a stimulant meets the following criteria: its selection confidence is above the **support threshold** and it has only one **primary respondent**. When a stimulant becomes confident enough to support a tactic, it is called a supporter stimulant.

Defintion 26: The **primary respondent** for a stimulant is a respondent with the largest weight.

Defintion 27: The **secondary respondent** for a stimulant is a respondent with the second largest weight.

Defintion 28: A **supporter stimulant** is a stimulant whose primary respondent is a tactic.

Defintion 29: The **support threshold** of a CLA specifies the minimum selection confidence required of a stimulant to support a tactic.

It is possible for a confident stimulant to lose confidence in its respondent. This happens frequently when the statistics in the STM are changing rapidly during the early stages of learning. When a confident stimulant's selection confidence drops below the **withdrawal threshold**, the stimulant withdraws its support from the tactic respondent and becomes an independent stimulant. A supporter stimulant may also withdraw its support from a tactic even if its selection confidence has not dropped below the withdrawal threshold if it acquires more than one primary respondent.

Defintion 30: The **withdrawal threshold** of a CLA specifies the minimum selection confidence of a supporter stimulant required to maintain support for a tactic. (By implication, the withdrawal threshold cannot be higher than the support threshold.)

Defintion 31: A **follower stimulant** is a stimulant that uses a tactic to guide its response selection.

At the moment the first tactic is supported in the STM, all stimulants except the first supporter become followers of the tactic. A follower stimulant stays with the first tactic as long as the tactic proves effective. If the tactic is not effective for a follower stimulant, the stimulant ceases to be a follower of the tactic and reverts to being a seeker stimulant.

The effectiveness of a tactic, known as its **potency**, φ, is the average compensation that the stimulant has received while using a particular tactic. If the potency of a stimulant's tactic drops below a specified threshold, then the tactic is deemed ineffective, and the stimulant abandons the tactic.

Postulate 02: A tactic's **potency**,φ, is the average compensation that a follower stimulant has received while being a follower of the tactic.

If a tactic proves effective for a follower stimulant, the stimulant stays with its tactic until the stimulant's selection confidence is greater than the **independence threshold** and becomes an independent stimulant.

Defintion 32: An **independent stimulant** is a stimulant that has an effective tactic, but follows the Standard selection policy.

Defintion 33: The **independence threshold** of a CLA is the minimum selection confidence required for a follower stimulant become independent. (By implication, the independence threshold cannot be higher than the support threshold.)

An independent stimulant's selection confidence may be higher than the support threshold if the stimulant has more than one primary respondent. A stimulant may have a selection confidence higher than the independence threshold, but still be considered a seeker stimulant if it does not yet have at least one effective tactic. The independence threshold must always be less than or equal to the support threshold.

An independent stimulant uses the Standard selection policy and explores the full response range. There is an important reason for this period of exploration to exist: a tactic may provide a suboptimal solution. Certainly, a suboptimal solution is better than a completely incorrect one, but one of the criteria of the research objective is that the solution must avoid settling in local maxima.

If an independent stimulant's selection confidence drops below the **dependence threshold**, then the independent stimulant returns to being a follower stimulant. The new follower stimulant returns to using its effective local tactic unless that tactic no longer exists. If its previous tactic no longer exists, the stimulant selects a new tactic from the global list. If no tactics are available in the STM, then the stimulant becomes a seeker stimulant.

Defintion 34: The **dependence threshold** of a CLA is the minimum selection confidence an independent stimulant must maintain to remain independent. (By implication, the dependence threshold cannot be higher than the independence threshold.)

3.2.2 Life cycle of Tactics

As has been described in Section 2.5, a Collective Learning Automaton (CLA) is a statistical learning device whose behavior is defined by a State Transition Matrix (STM) whose domain is populated by **stimulants** and whose range is populated by **respondents**. The elements of the STM are the weights (probabilities) that map each stimulant to one or more of the respondents. These weights are acquired through interaction with a supervising **Environment** that periodically issues evaluations of the CLA's behavior, which the CLA uses to update the weights in the STM.

In a TBL-CLA, all stimulants begin without being allied with any of the available respondents; however, eventually the confidence that one of the respondents is a valid response for an individual stimulant may exceed the **support threshold**. Such a respondent then becomes a **tactic respondent**, or simply **tactic**. Once a tactic exists in the STM, other stimulants may become followers of that tactic. Followers of a tactic always select the tactic response instead of following the Standard selection policy (see Section 3.3.6).

Every *follower stimulant* keeps track of the **local potency**. Each *tactic* also keeps track of its overall effectiveness, or **global potency**, for all of its followers past and present.

Postulate 03: The **local potency**, φ_l, of a tactic is a measure of effectiveness of a tactic for an individual follower stimulant and is calculated as follows:

Equation 1

$$\varphi_l = \sum_{\mathrm{i}=1}^{n} \frac{\gamma_i}{n_l}$$

Where n_l is the number of times that the tactic has been used and γ_i is compensation value received.

Postulate 04: The **global potency**, φ_g, of a tactic is a measure of effectiveness of a tactic for stimulants that have followed the tactic and is calculated as follows:

Equation 2

$$\varphi_g = \sum_{\mathrm{i}=1}^{n} \frac{\gamma_i}{n_g}$$

Where n_g is the number of times that the tactic has been used and γ_i is compensation value received.

The local potency is used by a follower to decide if it should continue to follow a specific tactic. If the local potency falls blow the **minimum local potency**, the follower abandons the tactic. The global potency is used to rank tactics in the STM. When a stimulant is choosing a tactic, it starts with the most globally potent tactic and works its way down the list of tactics until the stimulant discovers a tactic that proves to be locally potent.

Postulate 05: The **minimum local potency** of a tactic is 1, the minimum non-penalty.

A respondent remains a tactic until it no longer has any supporters. When all supporters of a tactic withdraw their support, a tactic **resigns** and is removed from the global list of tactics and any local tactic lists kept by stimulants. Note that when a tactic resigns, it might still have followers which must find other tactics or return to being seekers.

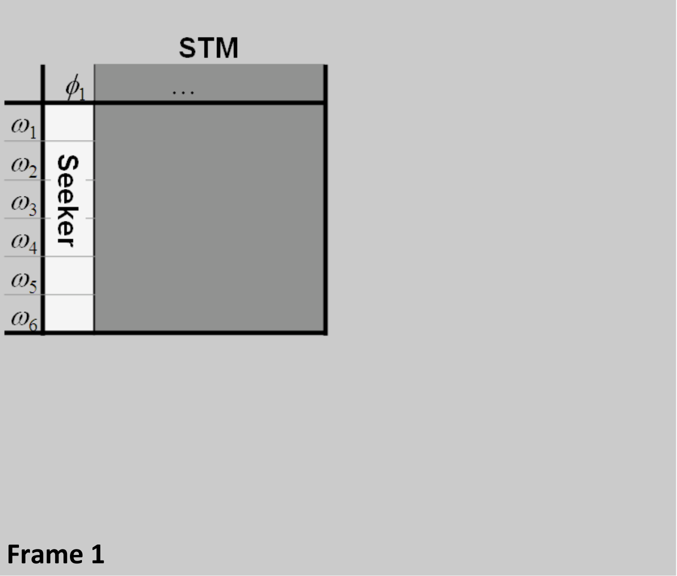
Postulate 06: A tactic **resigns** when it no longer has any supporters. After a tactic resigns, it is no longer available globally or locally for any stimulants to follow.

Tactic resignation occurs most frequently during the early stages of learning when stimulant selection confidences are changing rapidly. By allowing tactics to resign, Tactic-Based Learning attempts to avoid biasing the CLA in unproductive directions.

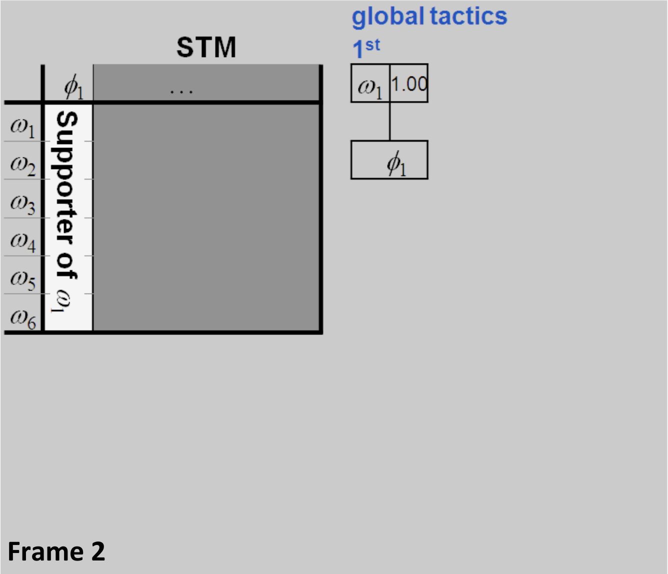
Once a tactic has resigned, it may be *reinstated* by any stimulant. There is no required waiting time between resignation and reinstatement. When a tactic is reinstated all potency values are reset to the default initial values. Followers must relearn the local potencies of a reinstated tactic and the reinstated tactic must establish its global potency again.

The preceding discussion described the roles of stimulants and respondents in the learning process. The role that a stimulant can be in at a given point in time is dictated by the presence, absence, or strength of the tactics that exist (or do not exist) in the CLA. It is necessary, however, to explain in more detail how tactics come into being, are adopted and rejected by individual stimulants, and the important differences between a global and a local tactic. In order to accomplish this, a series of sequential frames are presented below. Each frame has a number and all explanations of the process from frame to frame are given in the text after each frame.

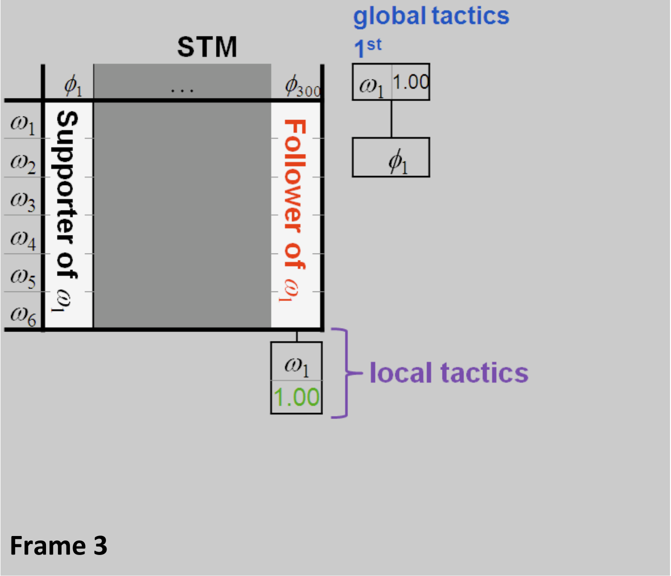
This sequence of frames starts when there are no tactics in the STM and demonstrates how tactics appears and disappear in the CLA. Since this illustration focuses on the rules that govern the process of the instantiation and resignation of tactics, no weights or statistics are provided for individual stimulants.



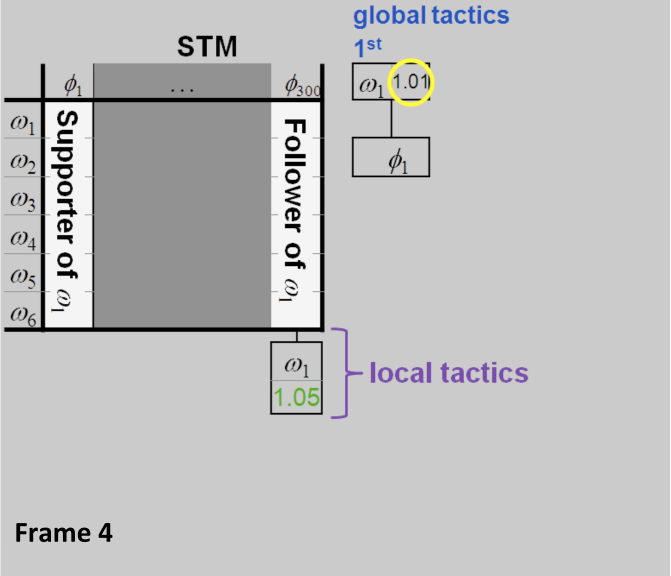
Frame 1 shows an STM before any tactics are present. Although the first stimulant, ϕ_1 is the only stimulant shown, the ellipses in the domain imply that much of the rest of the STM is in use, *i.e.* there are already many stimulants. The absence of any tactics in the CLA implies that ϕ_1 has not yet crossed the support threshold.



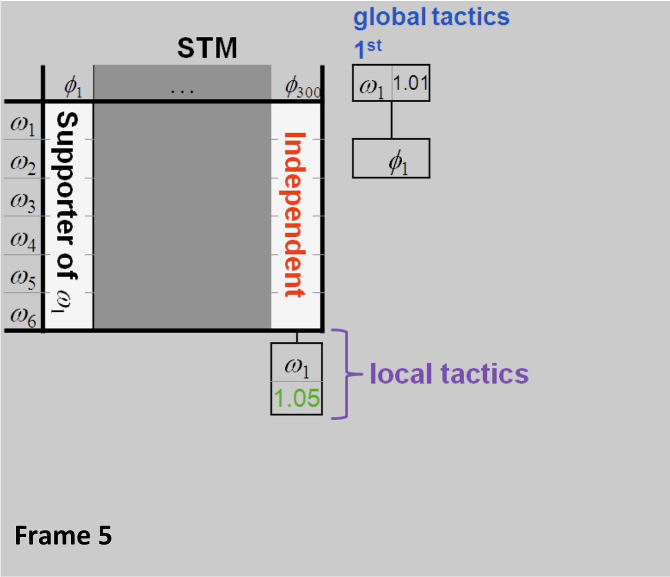
Frame 2 shows the CLA after ϕ_1 has crosses the support threshold. Stimulant ϕ_1 supports ω_1 as a tactic. When this happens, ω_1 is placed on the global tactics list with a global potency of 1.00. In this and subsequent frames, the global tactics consist of the tactic name and its current global potency. The line that extends down from the global tactic points to the list of stimulants that support the tactic, *i.e.* the supporters. As stimulants support or withdraw their support from tactics, they are added or removed from a global tactic's list of supporters.



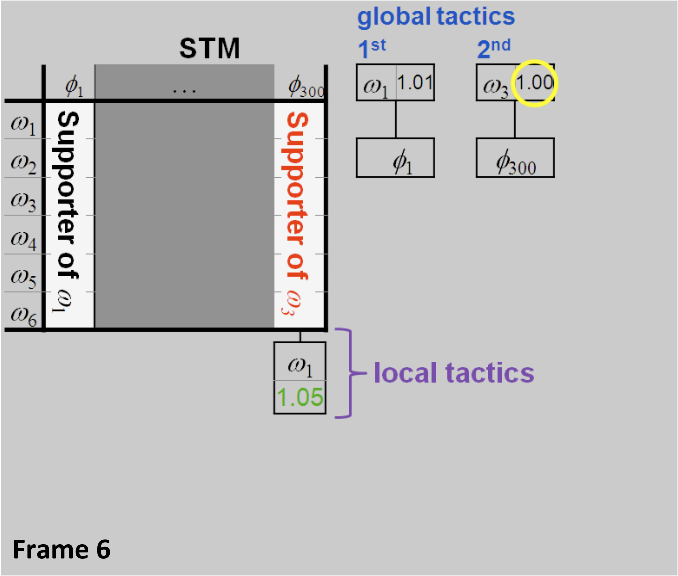
Later, other stimulants become followers. Stimulant ϕ_{300} is shown as a follower of ω_1. When a stimulant follows a tactic, the stimulant establishes its own list of local tactics.



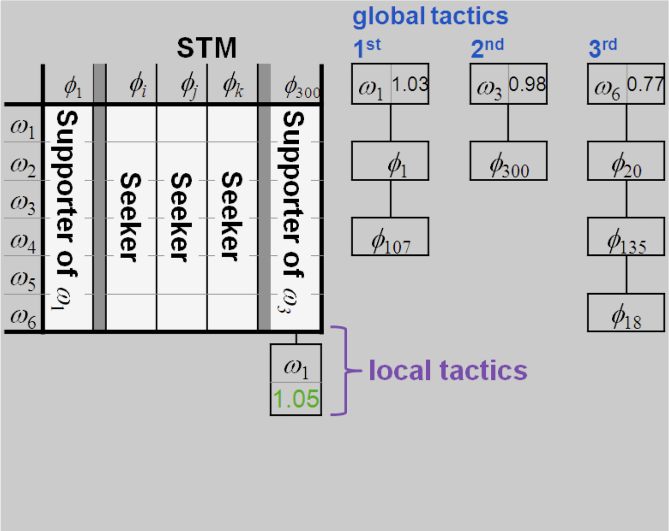
The local potency of a tactic reflects the average compensation that the stimulant has received while being a follower of the tactic. In the case of ϕ_{300}, its tactic has been effective and its local potency is at 1.05, which is above the minimum potency threshold of 1.00. Every time a stimulant uses a tactic, it updates its local potency *and* the global potency. Note that the global potency has changed from 1.00 to 1.01. This means that there are other stimulants that are followers of ω_1. If ϕ_{300} were the only stimulant following ω_1, the global potency would also be 1.05.



As stimulant ϕ_{300} becomes more confident, it becomes independent and stops following its local tactic, even though the local tactic is still potent. Stimulant ϕ_{300} takes this opportunity to explore its response range which helps avoid local maxima. When a stimulant is independent, it retains the list of local tactics even though it is not currently following a tactic.

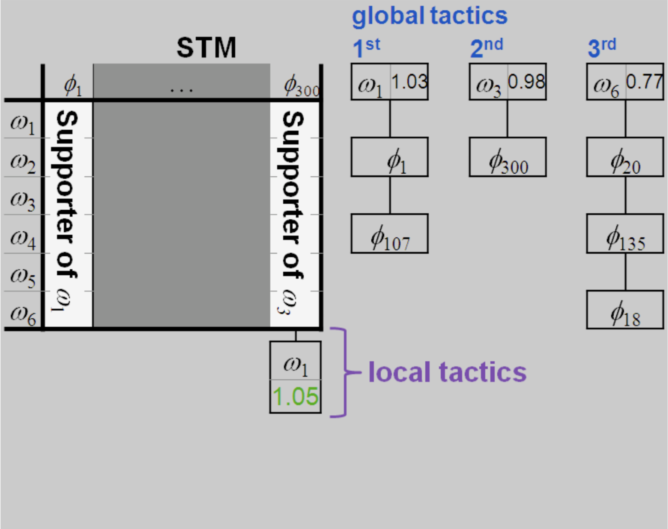


After a period of independence, ϕ_{300} becomes confident enough to support its own primary respondent as a tactic. Note that ϕ_{300} has come to support a different respondent than it successful tactic. When a new tactic is supported, it is added to the list of global tactics in order of global potency. All new tactics start with a potency of 1.00, the minimum potency. Because ω_3's potency is less than that of ω_1, it is placed second on the global tactics list. Note that even though ϕ_{300} is now a supporter of a tactic, it retains its local tactics list.



Frame 7

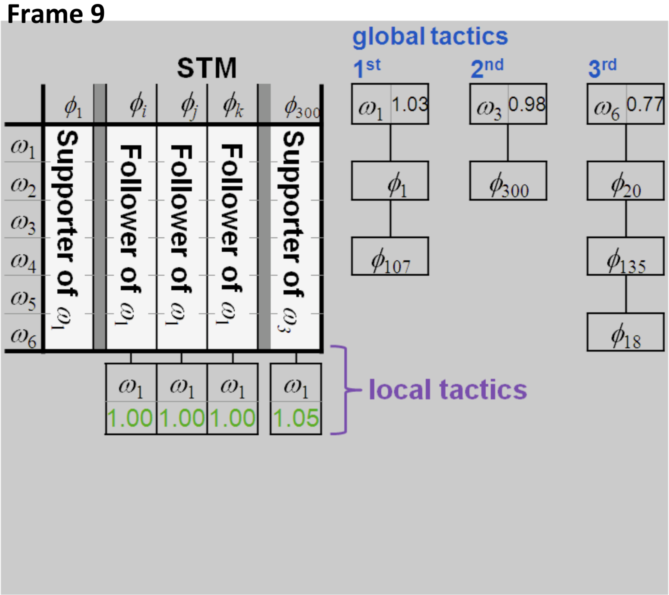
Frame 7 shows the CLA at a point in time farther along in the learning process. Another global tactic has been added and other stimulants which are not pictured have also emerged as supporters. Also all the global tactics' potencies have changed slightly. Note that more than one stimulant can support the same tactic. The global potency is not calculated based on the number of supporting stimulants, but it is important to track supporters because a tactic must resign if it no longer has any supporters.



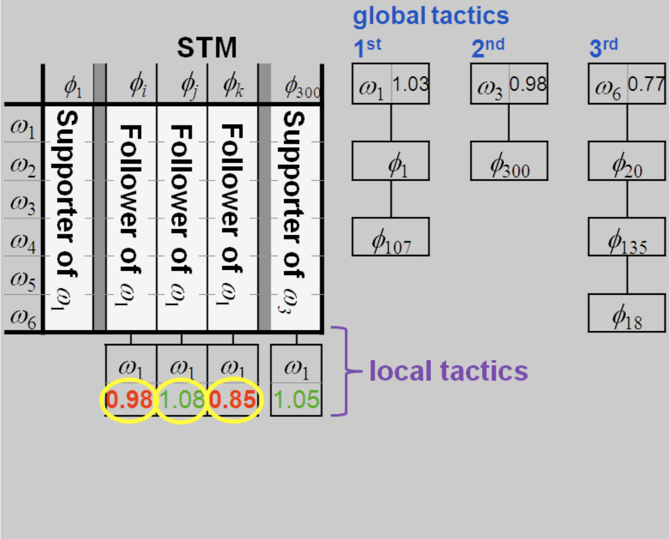
Frame 8

Later, three new stimulants appear in the STM. All stimulants are seekers when they first appear; however, they immediately seek a tactic the very first time they are called upon to choose a respondent.

Frame 9

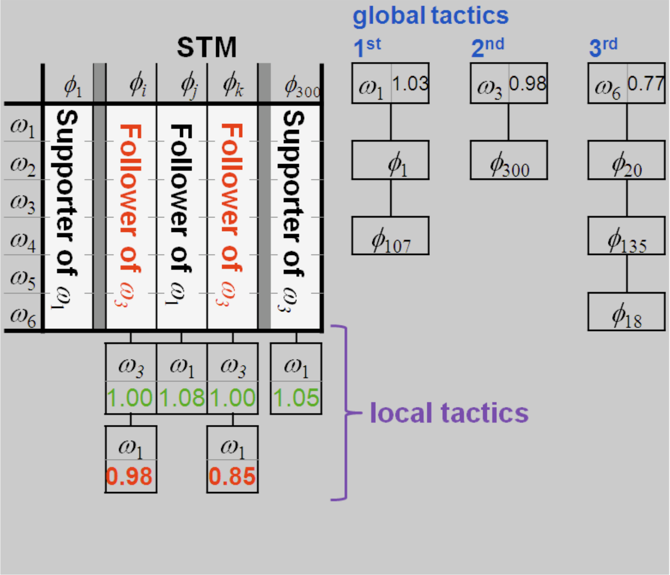


When the seekers are called upon to choose a respondent, they seek out a tactic. Since ω_1 is the most potent tactic on the global tactic list, all three become followers of it.

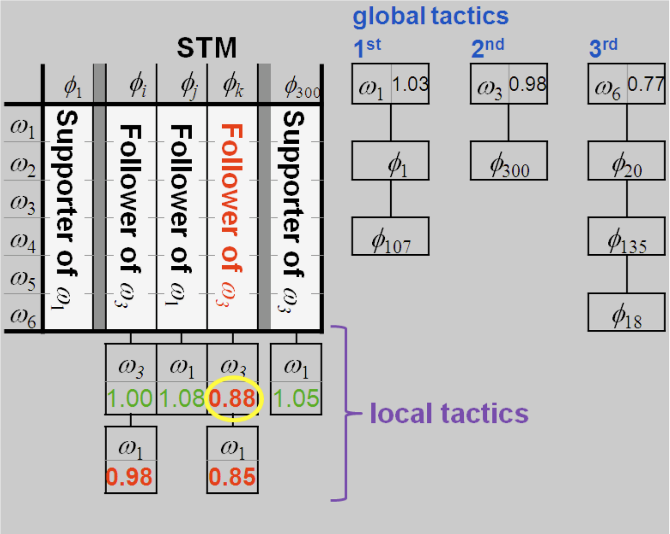


Frame 10

As time goes on, the new stimulants adjust their local potencies. In this example ϕ_i and ϕ_k have tactics whose local potencies have dropped below the minimum potency threshold. On the other hand, ϕ_j has a tactic that has become more potent.

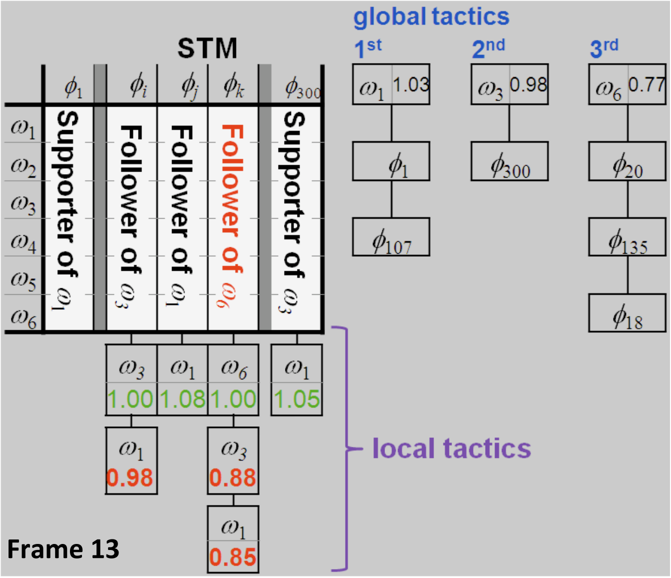


Both ϕ_i and ϕ_k now follow the next tactic from the global tactic list. All new local tactics start with a potency of 1.00. This allows for a quick move away from an ineffective tactic.

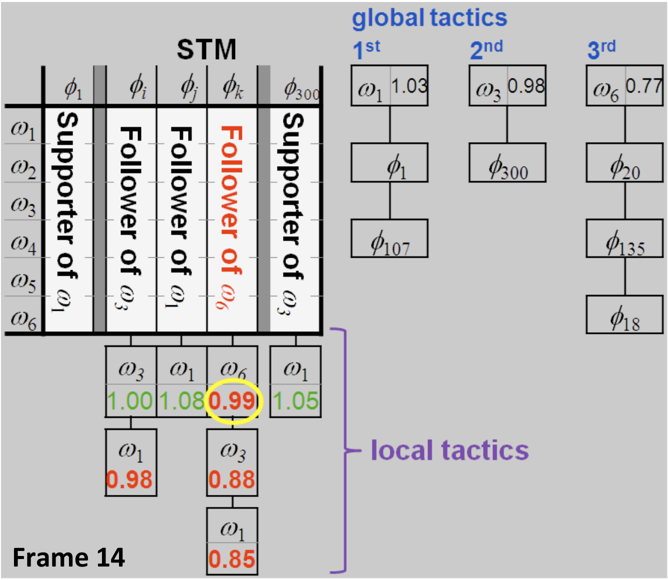


Frame 12

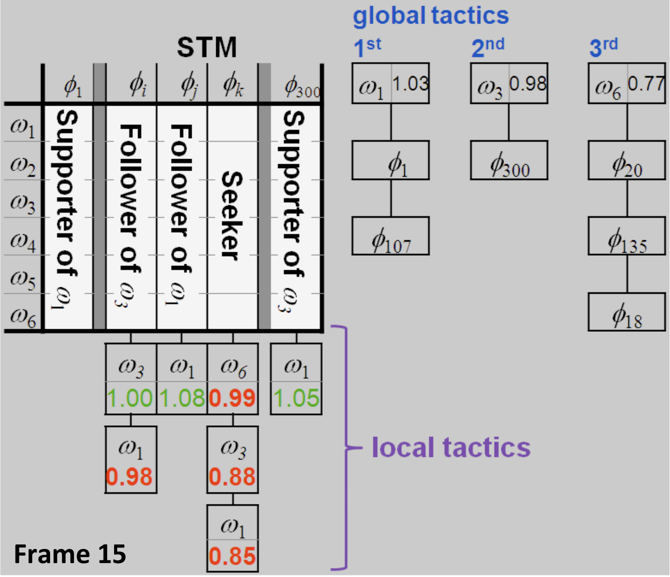
In this case, ϕ_k does not find its second tactic effective and so seeks another tactic the next time it is called on to choose a respondent.



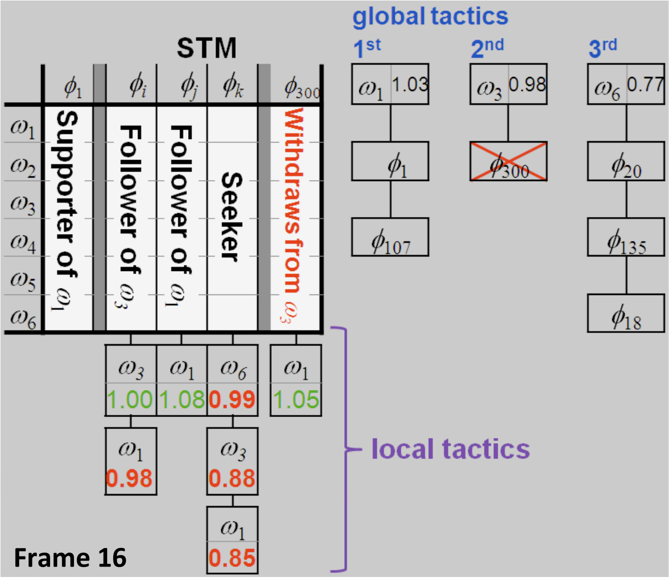
The next time ϕ_k chooses a respondent, it checks the list of global tactics and finds the only other tactic that it has not yet tried and becomes a follower of ω_6. Note that ϕ_k retains its list of previously used tactics.



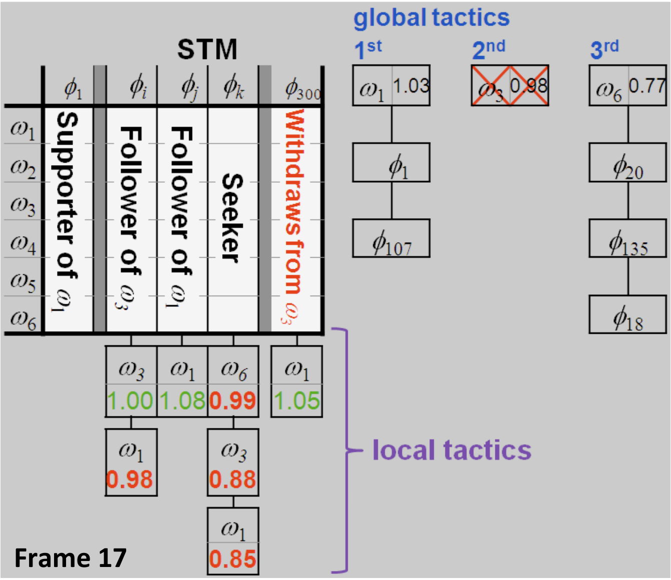
In this example, ω_6 is not an effective tactic for ϕ_k.



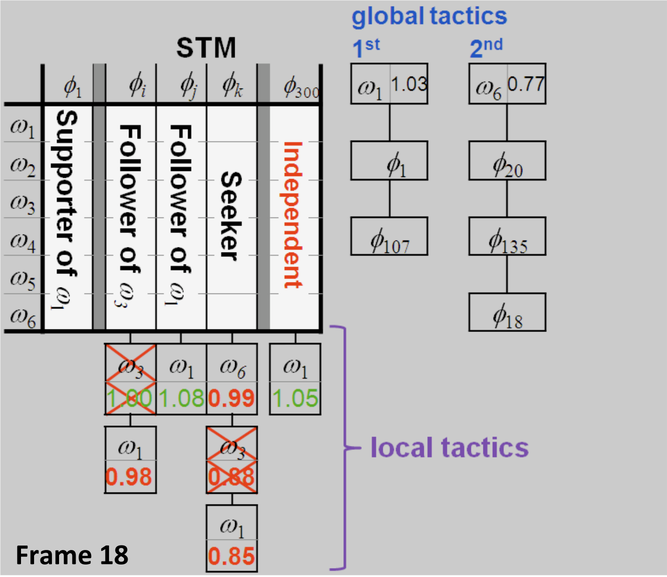
Since there are currently no more tactics available in the global tactic list, ϕ_k becomes a seeker stimulant.



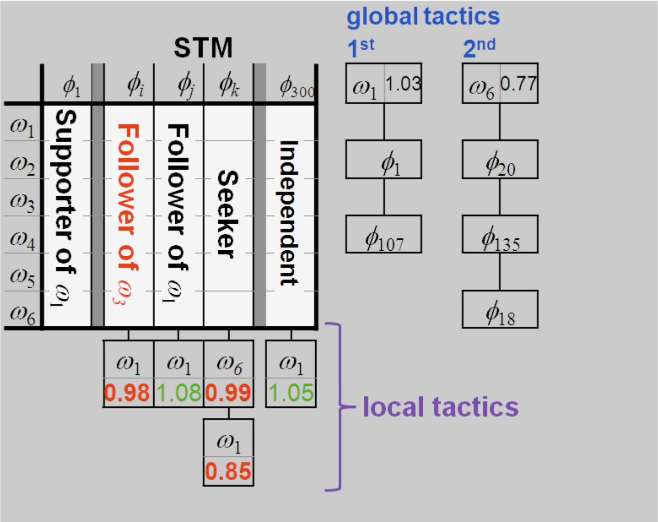
In the early stages of learning, it is sometimes the case that a supporter loses confidence and withdraws its support from the respondent it had supported as a tactic. When this happens, the supporter stimulant is removed from the global tactic's list of supporters.



When a global tactic no longer has any supporters, it resigns from the global tactic list.

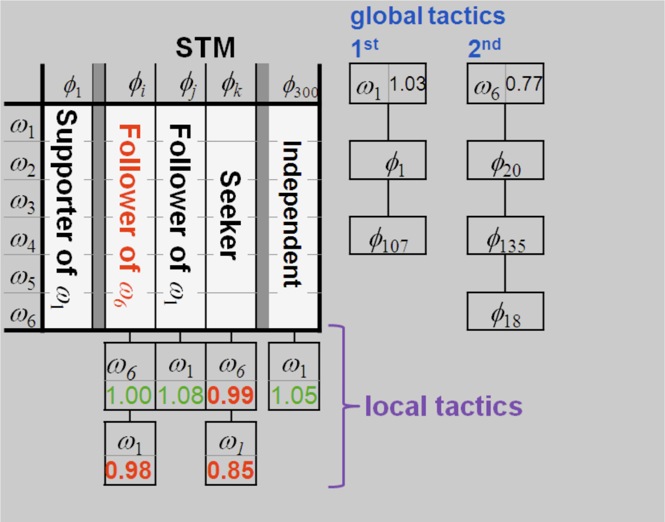


Once a supporter stimulant has withdrawn it support, it returns to independent status. In this example, when ϕ_k withdrew its support of ω_3, it caused ω_3 to resign from the global tactic list. When a tactic resigns, it can no longer be used as a tactic by any stimulants and must be removed from all local tactic lists.



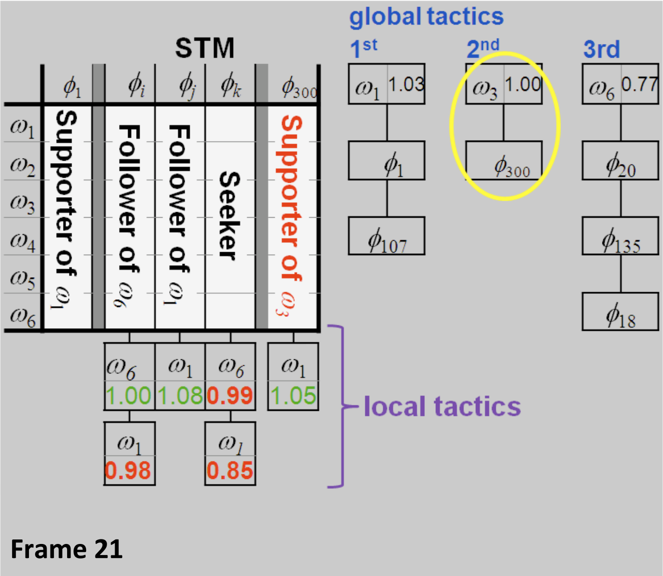
Frame 19

Now that ω_6 has been removed from the local tactic lists, stimulant ϕ_i must find another tactic. It does not consider ω_1 because that has already been proven an ineffective tactic.

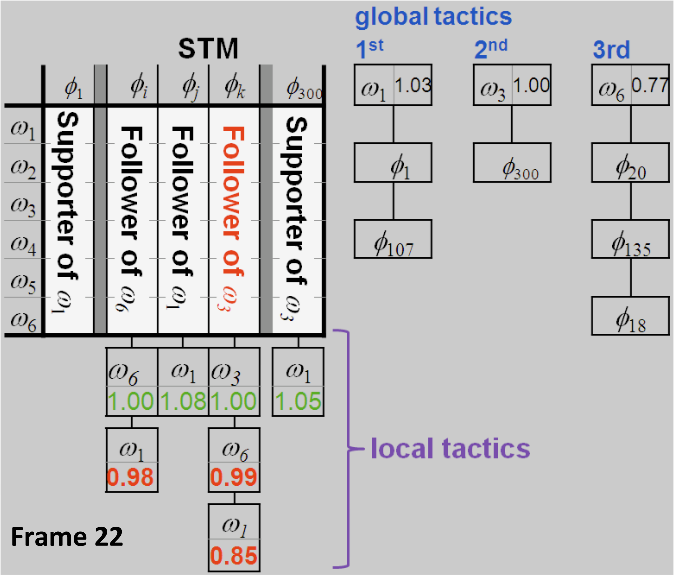


Frame 20

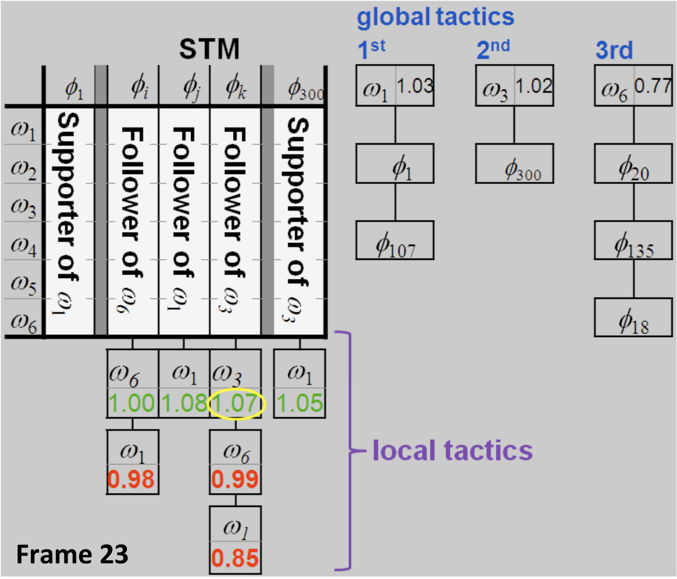
Stimulant ϕ_i becomes a follower of ω_6.



As learning progresses, it is quite possible for a stimulant to withdraw support from a tactic and later regain confidence and support the same tactic again.



Note that in Frame 12 the stimulant ϕ_k found the tactic ω_3 ineffective. When ω_3 resigned from the global tactic list, it was also removed from any and all local tactic lists. Now that ω_3 appears again as a tactic, ϕ_k no longer has a record that it has tried it before. This policy attempts to balance the fact that some tactics are unfairly eliminated from local tactic lists because they have been evaluated in histories which contain other incorrect responses. This counteracts some of the unfortunate side effects of longer collection lengths.



Occasionally, stimulants may find that a previously ineffective tactic is actually quite effective. This is because all local tactics start with a local potency of 1.00, the minimum potency. A new follower may fall in a history that is negatively compensated. Early compensations have a stronger effect on the local potency than later ones which can cause a follower to abandon its tactic even though it might actually be effective.

3.3 The TBL Algorithm and Data Structures

This section presents the TBL algorithm and necessary data structures to implement it in a CLA. The complete source code is provided in the Digital Appendix.

Tactic-Based Learning is implemented in a CLA, and so it is necessary to include some of the algorithms connected with CLS theory, although very little of CLS theory has been altered. It is easier to discuss the algorithm in terms of modules that represent the different parts of a CLS. In this section, *ModuleNames* are italicized. The flow of control is outlined first then the algorithms are presented in pseudo-code. In Sections 3.3.3 through 3.3.11, the algorithms and data structures are discussed in more detail. Section 3.3.12 presents a brief description of the time complexity implications of TBL. All references in this section to the Standard CLS algorithms are from (Bock 1993).

3.3.1 Flow of Control

The *Manager* gathers parameters from the human user, loads the correct *TruthTable*, generates the appropriate random stimuli, generates the fixed test set, and instantiates the *CLA* and *STM* with the appropriate parameters. Once everything is initialized, the *Manager* passes the stimuli, *Responses*, and evaluations between the *Environment* and the *CLA*. The *Manager* handles the training and testing of the *CLA* and records all the data that is used to calculate the preliminary results and conclusions.

The *Environment*, once initialized with a *TruthTable*, is responsible for periodically evaluating the responses generated by the *CLA*. The *Environment* does this using its evaluation policy, ξ. The evaluation policy may vary depending on the *TruthTable*.

The *CLA* receives random stimuli from the *Manager* and sends the stimuli to the *STM*. The *STM* returns *Responses* to the *CLA*, which sends the responses back to the *Manager*. When the *Manager* sends the *CLA* an evaluation, the *CLA* interprets the evaluation using its compensation policy to generate a compensation value, γ. The *CLA* then passes the compensation and the history of *Responses* to the *STM*.

The bulk of the TBL algorithm is implemented in the *STM*; however, the *STM* can handle both TBL and non-TBL learning paradigms. The *STM* receives stimuli from the *CLA* and sends back *Responses*. The *STM* stores all the *Stimulants* and maintains their learning statistics and calculates their reject, tie, and selection confidences. If the *STM* is using the TBL algorithm for selection respondents, it also stores the global tactic list.

Some of the modules function largely as data structures. These modules are *Stimulant, Response, TruthTable,* and *Tactic*. These modules are mentioned in the pseudo-code, but they are described in further detail in Sections 3.3.8 through 3.3.11.

3.3.2 Pseudo-code

```
Main()
   GetUsersInputs(RETURN userInputs {seed, matchLength, …})
   GenerateRandomStimuli(SEND seed; RETURN stimuli)
   FOR pre-selected states of the truthTable
      InitializeTruthTable(SEND stimuli; RETURN truthTable)
      InstantiateCLA(SEND userInputs; RETURN initializedCLA)
      TrainCLA(SEND initializedCLA, userInputs, truthTable, stimuli;
               RETURN trainedCLA)
      MeasurePerformance(SEND trainedCLA; RETURN performanceMeasures)
   END FOR
END Main
```

```
TrainCLA(SEND initializedCLA, userInputs, truthTable, stimuli; RETURN trainedCLA)
    FOR contest k = 1 to matchLength
        EmptyHistory(RETURN history)
        FOR 1 to collectionLength
            GetResponse(SEND $stimulus_k$; RETURN $response_k$)
            AppendHistory(SEND $stimulus_k$, $response_k$, history)
        END FOR
        CalculateEvaluation(SEND history, RETURN evaluation)
        EvaluateCLA(SEND evaluation, history)
    END FOR
END TrainCLA
```

```
GetResponse (RECIEVE stimulus; RETURN response)
    IF stimulus is not already a stimulant in STM THEN
        CreateStimulant(SEND stimulus, RETURN newStimulant)
        AddStimulant(SEND newStimulant)
        extent = extent + 1
        IF TBL-CLA THEN
            TBL(SEND newStimulant; RETURN response)
            RETURN response
        ELSE [Standard-CLA]
            Random(SEND newStimulant; RETURN any respondent)
            CreateResponse(SEND newStimulant, respondent; RETURN response)
            RETURN response
        END IF
    ELSE [stimulus is already a stimulant in STM]
        GetStimulant(SEND stimulus; RETURN stimulant)
        RecalculatePosteriorProbabilities (SEND stimulant)
        RecalculateConfidences (SEND stimulant)
        IF Standard-CLA THEN
            IF stimulant’s rejectConfidence < rejectThreshold THEN
                Random(SEND stimulant, RETURN any respondent)
                CreateResponse(SEND stimulant, respondent; RETURN response)
                RETURN response
            ELSE [stimulant’s rejectConfidence ≥ rejectThreshold]
                IF stimulant’s tieConfidence < tieThreshold THEN
                    Random(SEND stimulant;
                                RETURN primaryRespondent OR secondaryRespondent)
                    CreateResponse(SEND stimulant, respondent; RETURN response)
                    RETURN response
                ELSE [stimulant has a confident respondent]
                    CreateResponse(SEND stimulant, primaryRespondent;
                                RETURN response)
               RETURN response
            END IF
         END IF
      ELSE [TBL-CLA]
         IF stimulant is confident THEN
            CreateResponse(SEND stimulant, primaryRespondent;
                            RETURN response)
            RETURN response
         ELSE
            IF stimulant is an independent stimulant THEN
               IF stimulant's rejectConfidence < rejectThreshold THEN
                  Random(SEND stimulant, RETURN any respondent)
                  CreateResponse(SEND stimulant, respondent;
                                 RETURN response)
                  RETURN response
               ELSE [stimulant is tied]
                  Random(SEND phi,
                         RETURN primaryRespondent OR secondaryRespondent)
                  CreateResponse(SEND phi, respondent; RETURN response)
                  RETURN response
               END IF
            ELSE [stimulant is seeker OR a follower]
               TBL(SEND stimulant, RETURN response)
               RETURN response
            END IF
         END IF
      END IF
   END IF
END GetResponse
```

```
TBL(RECEIVE stimulant, RETURN response)
   IF stimulant has no local tactics THEN
      IF CLA has no global tactics THEN
         Random(SEND stimulant; RETURN any respondent)
         CreateResponse(SEND stimulant, respondent; RETURN response)
         RETURN response
      ELSE [CLA has global tactics]
         GetMostPotentTactic(RETURN tactic)
         AddLocalTactic(SEND tactic)
         ResetTactic(SEND initialValues)
         CreateResponse(SEND stimulant, tacticRespondent; RETURN response)
         RETURN response
      END IF
   ELSE [stimulant has local tactics]
      IF stimulant has an effective local tactic THEN
         CreateResponse(SEND stimulant, tacticRespondent; RETURN response)
         RETURN response
      ELSE [stimulant does not have an effective local tactic]
         IF there are no new global tactics THEN
            Random(SEND stimulant; RETURN any respondent)
            CreateResponse(SEND stimulant, respondent; RETURN response)
            RETURN response
         ELSE [there is a new global tactic]
            GetMostPotentTactic(RETURN tactic)
            AddLocalTactic(SEND tactic)
            ResetTactic(SEND initialValues)
            CreateResponse(SEND stimulant, tacticRespondent;
                                RETURN response)
            RETURN response
         END IF
      END IF
   END IF
END TBL
```

```
RecalculateConfidences(RECEIVE stimulant)
    CalculateTieConfidence(SEND stimulant)
    CalculateRejectConfidence(SEND stimulant)
    IF TBL-CLA THEN
        CreateTactic(SEND stimulant’s primaryRespondents, RETURN tempM)
        IF selectionConfidence ≥ supportThreshold
            AND stimulant has only one primaryRespondent THEN
            SetIndependentStatus(SEND stimulant, FALSE)
            SetSupporterStatus(SEND stimulant, TRUE)
            AddSupporter(SEND stimulant, tempM)
            IF tempM is not on the global tactic list THEN
                AddGlobalTactic(SEND tempM)
            END IF
        ELSE IF stimulant is a supporter AND
            (selectionConfidence < withdrawalThreshold OR
             stimulant has more than one primaryRespondent) THEN
            SetSupporterStatus(SEND stimulant, FALSE)
            IF selectionConfidence ≥ independenceThreshold AND
                stimulant has at least 1 useful tactic THEN
                SetIndependentStatus(SEND stimulant, TRUE)
            ELSE
                SetIndependentStatus(SEND stimulant, FALSE)
            END IF
            RemoveSupporter(SEND stimulant, tempM)
            IF tempM has no supporters THEN
                RemoveGlobalTactic(SEND tempM)
                RemoveLocalTactic(SEND tempM, all stimulants in domain)
            END IF
        ELSE IF stimulant is a supporter AND stimulant's supported tactic != tempM THEN
            RemoveSupporter(SEND stimulant, previousTactic)
            AddSupporter(SEND stimulant, tempM)
            IF previousTactic has no supporters THEN
                RemoveGlobalTactic(SEND previousTactic)
                RemoveLocalTactic(SEND previousTactic, all stimulants in domain)
            END IF
        ELSE IF stimulant is not a supporter AND stimulant is not a independent AND
            stimulant has at least 1 useful tactic AND
            selectionConfidence ≥ independenceThreshold THEN
            SetIndependentStatus(SEND stimulant, TRUE)
            SetSupporterStatus(SEND stimulant, FALSE)
        ELSE IF stimulant is independent AND
            selectionConfidence < dependenceThreshold THEN
            SetIndependentStatus(SEND stimulant, FALSE)
```

```
      SetSupporterStatus(SEND stimulant, FALSE)
    END IF
  END IF
END RecalculateConfidences
```

```
CompensateCLA(RECEIVE evaluation, history)
  confidentCount = 0
  followerCount = 0
  FOR every response in history
    IF $rejectConfidence \geq rejectThreshold$ THEN
      confidentCount = confidentCount + 1
    ELSE IF response was follower THEN
      followerCount = followerCount + 1
    END IF
  END FOR
  expectedEvaluation = $(confidentCount + followerCount)/collectionLength$
  IF $evaluation = 1$ OR ($evaluation > 0$ AND $evaluation > expectedEvaluation$) THEN
    compensationConfident = 1 + 0.1(evaluation)
    compensationOther = compensationConfident
  ELSE IF $evaluation > 0$ AND $evaluation = expectedEvaluation$ THEN
    IF $averageSelectionConfidence < compensationThreshold$ THEN
      compensationConfident = 1 + 0.1(evaluation)
      compensationOther = compensationConfident
    ELSE
      compensationConfident = 0.999
      compensationOther = 0.975
    END IF
  ELSE IF $evaluation > 0$ AND $evaluation < expectedEvaluation$ THEN
    IF $averageSelectionConfidence < compensationThreshold$ THEN
      compensationConfident = 1 + 0.1(evaluation)
      compensationOther = compensationConfident
    ELSE
      compensationConfident = 0.96
      compensationOther = 0.98
    END IF
  ELSE [$evaluation = 0$]
    compensationConfident = 0.85
    compensationOther = 0.90
  END IF
  UpdateSTM(SEND compensationConfident, compensationOther, history)
END CompensateCLA
UpdateSTM (receive compensationConfident, compensationOther, history)
  FOR every response in history
    GetRespondent(SEND response, RETURN respondent)
```

```
        $respondentCount = respondentCount + 1$
        IF tieConfidence AND rejectConfidence are at maximumValue THEN
            $compensation = 1$
        ELSE [tieConfidence OR rejectConfidence < maximumValue]
            IF $tieConfidence \geq tieThreshold$ AND $rejectConfidence \geq rejectThreshold$ THEN
                $compensation = compensationConfident$
            ELSE [nextResponse not_confident]
                $compensation = compensationOther$
            END IF
        END IF
        $newWeight_{respondent} = oldWeight_{respondent}(compensation)$
        IF TBL-CLA AND newResponse used a tactic THEN
            UpdateTacticPotency(SEND globalTactic, compensation)
            UpdateTacticPotency(SEND localTactic, compensation)
        END IF
    END FOR
END UpdateSTM
```

```
UpdateTacticPotency(RECEIVE tactic, compensation)
    $potency_{current} = tacticPotency$
$potency_{temporary} = potency_{current}(success + fail) + compensation$
    IF $compensation \geq minCompensation$ THEN
        $success = success + 1$
    ELSE
        $fail = fail + 1$
    END IF
$tacticPotency = potency_{current} \div (success + fail)$
END UpdateTacticPotency
```

3.3.3 *Environment*

The *Environment*’s main purpose is to evaluate the *Responses* produced by the *STM*. In order to provide an evaluation of the *Responses*, the *Environment* must be provided with the correct *TruthTable*. A *TruthTable* is given to the *Environment* by the *Manager* at the beginning of the game.

The *Environment* does not have any significant data structures to discuss; however, it does have the evaluation policy, ξ. Because this research deals only with categorical outputs, a *CLA* is evaluated based on the number of correct responses. The evaluation of a single *Response* has only two possible options: correct and incorrect.

Postulate 07: The evaluation policy for the game is the average evaluation for the collection length,

$$\frac{r_{correct}}{n}$$

where $r_{correct}$ is the number of correct *Responses* and *n* is the total number of *Responses*.

3.3.4 *CLA*

The *CLA* module, like the *Environment*, does not have any significant data structures. It received stimuli from the *Manager* and passes them to the *STM* where a *Response* is generated. The *CLA* takes the *Response* from the *STM* and passes it back to the *Manager*. The main responsibility of the *CLA* is to interpret the evaluations that are handed down from the *Environment* through the *Manager*. The evaluations are interpreted using a compensation policy which generates a compensation value, γ.

Using Tactic-Based Learning allows the *CLA* to compensate stimulants differently. The compensation policy postulated in pseudo-code in Section 3.3.4 calculates an anticipated evaluation based on the number of confidence stimulants and the number of follower stimulants. It should be noted that the compensation policy is applicable, and indeed is applied, to both Standard and TBL-CLAs. The purpose of generating a more nuanced compensation policy is to help a TBL-CLA from settling into local maxima. This would be very easy for a TBL-CLA to do because it is more likely to receive more positive evaluations early on because of the stimulants that use tactics. At longer collection lengths, it is possible for some incorrect follower stimulants to be positively evaluated because they are in histories with many other correct stimulants. This situation can also occur in a Standard-CLA, but in a TBL-CLA these incorrect stimulants are following tactics, and they continue to select the same respondent. In a Standard-CLA, a non-confident stimulant that chooses an incorrect respondent is not likely to make the same selection the next time because it is still selecting responses at random.

Defintion 35: A stimulant whose tie and reject confidences are greater than or equal to the tie and reject thresholds, respectively, generates a **confident response**.

A generic compensation policy would simply scale the compensation with the evaluation. This is a good policy for a Standard-CLA because it rewards any correct responses that were made over the course of the history; however, when some stimulants are follower stimulants, problems can occur. In the early stages of learning, the TBL-CLA performs very well and receives large amounts of positive compensation. This positive compensation strengthens the local potency of the tactics being used by follower stimulants. Some of the follower stimulants might be followers of tactics that are actually incorrect for them, but the stimulants receive some positive compensation in general because many other stimulants are correct. As follower stimulants of incorrect tactics become independent, they make mistakes and return to their incorrect tactic. Over time the counts behind the incorrect response get very large, which makes it very difficult for the stimulant to choose another respondent when it becomes independent. Once that happens, stimulants can become very confident in an incorrect response. By the time this happens, most of the other stimulants have been associated with their correct respondent and the CLA never receives the maximum evaluation, but it is a good enough evaluation to reinforce the incorrect respondents along with the correct ones, leading the CLA to settle in a local maxima.

The compensation policy in this research scales with the evaluation until the CLA becomes highly confident, on average. After that point, the compensation becomes much harsher and this change encourages the CLA to make corrections and helps avoid local maxima. The appropriate setting of the compensation threshold is determined by OP Pilots for the different experiments and is not a factor of this research, although informal observation suggests that its influence on learning behavior is a worthwhile research question.

Defintion 36: The **compensation threshold**, κ_γ**,** is a parameter in the compensation policy. When a CLA's average selection confidence crosses this threshold, the compensation policy becomes more stringent.

Postulate 08: The compensation policy for the unordered game generates the compensation value, γ, is as follows:

```
IF $\xi = 1$ OR ($\xi > 0$ AND $\xi > \xi_{anticipated}$) THEN
  $\gamma_{confident} = 1 + 0.1(\xi)$
  $\gamma_{other} = \gamma_{confident}$
ELSE IF $\xi > 0$ AND $\xi = \xi_{anticipated}$ THEN
  IF $avgerageSelectionConfidence < \kappa_{\gamma}$ THEN
    $\gamma_{confident} = 1 + 0.1(\xi)$
    $\gamma_{other} = \gamma_{confident}$
  ELSE
    $\gamma_{confident} = 0.999$
    $\gamma_{other} = 0.975$
  END IF
ELSE IF $\xi > 0$ AND $\xi < \xi_{anticipated}$ THEN
  IF $avgerageSelectionConfidence < \kappa_{\gamma}$ THEN
    $\gamma_{confident} = 1 + 0.1(\xi)$
    $\gamma_{other} = \gamma_{confident}$
  ELSE
    $\gamma_{confident} = 0.96$
    $\gamma_{other} = 0.98$
  END IF
ELSE
  $\gamma_{confident} = 0.85$
  $\gamma_{other} = 0.9$
END IF
```

3.3.5 *STM*

The *STM* receives stimuli from the *CLA* and uses a selection policy to choose a respondent from which it generates a *Response*. The *STM* periodically receives a compensation value, γ, and a history of *Responses* from the *CLA*. When the *STM* receives a compensation value, it uses its update policy to adjust the weights and statistics in the

Stimulants that make up the actual state transition matrix.

If a *CLA* is using the TBL selection policy, then the *STM* also handles the TBL process by determining when a *Stimulant* has elected or abandoned a *Tactic* and when a *Stimulant* is ready to become independent. This is where most of the procedures involved in the TBL algorithm are located.

The *STM* also has two data structures that are important to the algorithm: a list of all the stimulants and the list of global tactics.

`List stimulantDomain`

The `stimulantDomain` is a list containing *Stimulants*. The `stimulantDomain` is dynamically allocated so that the *STM* can be scaled in size easily, but this is not a requirement of CLS theory.

`List globalTactics`

`The globalTactics` list contains the global tactics. *Stimulants* in search of a new *Tactic* look to `globalTactics` to ascertain if any tactics are available.

The procedures **`GetResponse`**, **`TBL, RecalculateCondifences`**, and **`UpdateSTM`** all fall under the auspices of the *STM* module. Section 3.3.4 discusses how the *STM* selects a respondent to use in a *Response* using either the Standard selection policy or the TBL selection policy. Section 3.3.7 discusses how the *STM* incorporates a compensation value by using its update policy.

3.3.6 Discussion of the selection policy algorithms

One of the *STM*'s jobs is to apply a selection policy to a *Stimulant* and choose a respondent that becomes part of a *Response*. To generate a response for a stimulus, the *STM* first checks that the corresponding *Stimulant* exists in the *STM*. If it does not exist, a new *Stimulant* is created and added to the stimulus domain. If the *STM* is using TBL and there are global tactics available, then the new *Stimulant* is assigned its first tactic and it selects that tactic respondent. If the *STM* is using the Standard selection policy, a random respondent is selected.

If the *STM* is using the **Standard selection policy** and the stimulus corresponds to a known *Stimulant*, policy dictates that if the primary respondent is significant (*i.e.* if the tie and reject confidences are above their respective thresholds), the primary respondent is chosen and returned to the *CLA*. If the primary respondent and the secondary respondent are tied (*i.e.* if the reject confidence is greater than the reject threshold, but the tie confidence is not above the tie threshold), then the *STM* chooses randomly between the primary and secondary respondents. If there is no clear choice (*i.e.* neither the reject nor tie confidence is above its threshold), then the *STM* randomly chooses a respondent from among all the possible respondents.

If the *STM* is using TBL and the stimulus corresponds to a known *Stimulant*, then the *STM* attempts to apply the TBL selection policy. If the *Stimulant* has no local tactics, then the *STM* checks the list of global tactics. If there are no global tactics, then the *STM* follows the Standard selection policy. If global tactics *do* exist, then the *STM* selects the global tactic with the highest global potency, makes a copy of it, adds the new tactic to the *Stimulant's* local tactic list, and uses the tactic's respondent to create a *Response* to return to the *CLA*.

If the *Stimulant* exists in the *STM* and has local tactics, the *STM* searches for an effective local tactic. If an effective local tactic is available, the *STM* uses the local tactic respondent to create a *Response* to return to the *CLA*. If there is no effective local tactic, then the *STM* searches for a tactic in the global list that has not been tried by the current

Stimulant. If a new global tactic is available, it is added to the *Stimulant's* list of local tactics and the *STM* uses the new tactic respondent to create a *Respondent* to return to the *CLA*. If there is no effective local tactic and no new global tactic, then the *STM* follows the Standard selection policy.

When a *CLA* uses the TBL selection policy, the status of tactic respondents can change every time the confidence is recalculated. The confidence needs to be recalculated every time **`GetResponse`** is called. Instead of calculating the confidences for all stimulants after every call to **`UpdateSTM`**, individual *Stimulants* have their confidences recalculated one at a time as they are needed. This saves a great deal of execution time as the stimulus domain gets larger.

The details of calculating the confidences are not presented in **`GetResponse`** because the calculations are the same whether or not the *CLA* uses TBL; however, if the *CLA* is using TBL, the status of a *Stimulant* (*i.e.* follower, independent, *etc.*) must be reconsidered every time the confidences are recalculated.

A *Stimulant* may support a new *Tactic* to the global list of *Tactics* if the *Stimulant* is the first to become confident in a respondent which is not present on the global tactic list. The *Stimulant* becomes the *Tactic's* first supporter. Once a *Tactic* has been added to the global list, it is available to all seeker *Stimulants*, that is, *Stimulants* without an effective tactic.

As the weights and statistics change in the *STM*, a *Stimulant* may lose confidence in its respondent. When that happens, the *Stimulant* withdraws its support from the *Tactic* and the *Stimulant* returns to being either an independent *Stimulant* or a seeker *Stimulant* (this happens if the *Stimulant* does not have a previously effective local *Tactic*).

As *Stimulants* withdraw their support from a *Tactic*, it is possible that a *Tactic* could be left with no supporters at all. If this happens, the *Tactic* must resign. When a *Tactic* resigns, it removes itself from the global tactic list and all local tactic lists it was on, even if it was an effective *Tactic* for a certain follower *Stimulant*. If a Stimulant becomes confident in a former *Tactic* respondent, the respondent may be supported again as a *Tactic*.

When a follower *Stimulant's* selection confidence rises above the independence threshold, it becomes an independent *Stimulant.* Even though the *Stimulant* has an effective *Tactic*, it reverts to the Standard selection policy and explores the response range.

Independent *Stimulants* are highly volatile. Some may become confident in a respondent and go on to support a *Tactic*. Other independent *Stimulants* lose enough confidence that they cross the dependence threshold and are converted back to follower *Stimulants*. When a *Stimulant* becomes a follower again, it purges its local tactic list. By doing this, it is free to consider all the available *Tactics* again. This is useful because a more effective *Tactic* may have been added to the global *Tactic* list during the period that the *Stimulant* was independent. This is an opportunity for the follower to rediscover an effective *Tactic*.

3.3.7 *Discussion of* ***`UpdateSTM`***

When the *STM* receives a compensation value and the history from the *CLA* it must apply the update function to the compensation in order to calculate the new weight of the respondents in the history. The update policy to be used in this research is given below. This update policy is designed to keep the weights in the *STM* in a reasonable balance. This update function also distinguishes between the *Stimulants* that were very confident in their responses and those that were not. A full description of the compensation policy is given in Section 3.3.4, but briefly, confident *Stimulants* should receive smaller positive updates and larger negative updates than *Stimulants* that are still learning and not yet confident in a respondent.

Postulate 09: The **update policy** for both games is

```
IF ($\phi$ is confident) THEN
    $w^1 \leftarrow w$
ELSIF ($\phi$'s tie confidence >= tie threshold AND
    $\phi$'s reject confidence >= reject threshold)
    $w^1 \leftarrow w\gamma_{confident}$
ELSE
    $w^1 \leftarrow w\gamma_{normal}$
END IF
```

3.3.8 *Tactic*

The *Tactic's* main purpose is to provide a data structure for the global and local tactics. A *Tactic* contains the number of the respondent that it represents, its potency (global or local), and a list of supporters, if it is a global *Tactic*. A follower *Stimulant* does not need to know which other *Stimulants* happen to be supporters of its *Tactic*; it is only interested in the local potency of the *Tactic*. Aside from the data structure, the *Tactic* module contains one important piece of the algorithm: the policy for updating a *Tactic's* global or local potency. In this section, the data structure is presented first and then the potency update policy is discussed in more detail.

`ID`

A *Tactic* has a unique and comparable identification that is usually the same as the number of the respondent that the *Tactic* represents.

`integer response`

`response` is the index of the respondent to which *Tactic* corresponds.

`double potency`

This is the current value of the *Tactic's* potency. If the *Tactic* is stored in the global tactic list, then this number is the global potency of the tactic. If the *Tactic* is stored locally with a *Stimulant*, then this number is the local potency of the *Tactic*.

`List supporters`

Only global Tactics use this list. It contains the identifications of *Stimulants* that support a given *Tactic*. If the *Tactic* is local, then this list remains empty.

`integer success`

This is the number of times the *Tactic* has been used by a *Stimulant* and received positive compensation. It is used in calculating the potency of the *Tactic*.

`integer fail`

This is the number of times the *Tactic* has been used by a *Stimulant* and received negative compensation. It is used in calculating the potency of the *Tactic*.

`UpdateTacticPotency``(`**`RECEIVE`** *`compensation`*`)`

This procedure is responsible for updating the *Tactic's* potency. The algorithm is the same whether the *Tactic* is global or local. To update a *Tactic's* potency, the compensation is first averaged with the current potency. If the compensation received while using this tactic was greater than or equal to the minimum positive compensation value (set as a parameter), the use of the tactic is considered a success. The number of successes or failures is then updated accordingly.

By averaging the current compensation with the potency, greater influence is given to the compensation the *Tactic* received earlier in learning. This process is important because earlier in learning, more *Stimulants* are followers and are therefore behaving in a more stable way. If negative compensation is received early in learning, it is very likely that this *Tactic* is not effective for its current follower. If the compensation received early in learning is positive, it is very likely that this *Tactic* is, in fact, effective for its current follower.

Later on in learning, more *Stimulants* become independent. If the collection length is longer than one, then the erratic behavior of independent *Stimulants* may negatively affect the compensation received by a follower *Stimulant*. By weighting the first few experiences a follower *Stimulant* has with a *Tactic* more heavily, the quicker a *Stimulant* abandons an ineffective *Tactic* and the more "trust" it develops in an effective *Tactic*. This trust can help followers stay with a *Tactic* through the times when the follower's performance is being evaluated in the same collection as another independent *Stimulant*.

3.3.9 *Stimulant*

Although the STM is usually visualized as a matrix, it is more practical, from a programming standpoint, to keep each column of the *STM* with its associated *Stimulant*. The *Stimulant* module provides important data structures that are discussed in this section.

ID

A unique and comparable identifier.

integer intentSize

`intentSize` is the size of the *effective* intent. In some cases, certain respondents in the intent are not legal for a given *Stimulant*. Consider a game like chess. In general, the queen is free to move in all directions; however, the queen may not take the place of another piece on the queen's team, nor may the queen move off of the board. To compensate for this, each *Stimulant* keeps track of its own effective intent.

List pProb

`pProb` is a list of all the respondents currently sharing the highest posterior probability. It is necessary to have a list and not just a single respondent's ID because it is possible, especially early on in learning, more than one respondent may have the same posterior probability.

List sProb

`sProb` is a list of all the respondents currently sharing the second highest posterior probability. It is necessary to have a list and not just a single respondent's ID because many times, especially early on in learning, more than one respondent may have the same posterior probability.

integer reject

`reject` is the confidence that the following null hypothesis can be rejected

> *H_0*: the value of the primary posterior probability is significantly different from the *apriori* probability.

`integer tie`

`tie` is the confidence that the following null hypothesis can be rejected

> *H_0*: the value of the primary posterior probability is significantly different from the value of the secondary posterior probability.

`integer selection`

The selection confidence, which is not a true statistical confidence, is the minimum of the reject and tie confidences. This value is used as a shorthand measure of the confidence a *Stimulant* has in its knowledge.

`double[][] column`

This double array holds the weights, counts, and statistics for a given *Stimulant* in the *STM*. When using Maximum Likelihood to calculate the posterior probabilities, three arrays are necessary. The first array holds the weights for each respondent that have accrued over the learning process. The second array holds the count of the number of times the stimulant has chosen a given respondent. The third array holds the posterior probabilities.

`List tactics`

`tactics` is a list of the *Tactics* this *Stimulant* has considered.

`Boolean independent`

`independent` takes the value TRUE if this is an independent *Stimulant* and takes the value FALSE otherwise. This variable always takes the value FALSE if the *Stimulant* is in a Standard-CLA.

`Boolean supporter`

`supporter` takes the value TRUE if this *Stimulant* is confident and supports a *Tactic*. This variable takes the value FALSE if this *Stimulant* does not support a tactic. This variable always takes the value FALSE if the *Stimulant* is in Standard-CLA.

`Boolean usefulTactic`

`usefulTactic` takes the value TRUE if this *Stimulant* has a potent local tactic and takes the value FALSE if it does not. This variable retains the value TRUE even when the *Stimulant* becomes independent. This variable takes the value FALSE if the *Stimulant* is in a Standard-CLA.

3.3.10 *Response*

A *Response* provides the data structure that is passed to the *Environment* from the *CLA* through the *Manager* and stored in the history. This section describes the data structure in more detail.

`StimulantID`

The unique and comparable ID for the *Stimulant* in the stimulus-response pair.

`integer respondent`

The identification number of the respondent chosen by the *CLA*.

`Boolean usedTactic`

`usedTactic` takes that value TRUE if the *Stimulant* was a follower and FALSE otherwise. The variable `usedTactic` has the value FALSE in a Standard-CLA.

`Tactic t`

The local *Tactic* used in this interaction. If no *Tactic* was used, then `t` takes the value `null.`

`integer tie`

`tie` is the tie confidence for this interaction.

`integer reject`

`reject` is the reject confidence for this interaction.

`integer selectionConfidence`

`selectionConfidence` is the selection confidence for this interaction.

`Boolean independent`

`independent` takes that value TRUE if the *Stimulant* was an independent *Stimulant* and FALSE otherwise. The variable `independent` has the value FALSE in a Standard-CLA.

`Boolean isSupporter`

`isSupporter` takes that value TRUE if the *Stimulant* supports a *Tactic* and FALSE if the *Stimulant* does not. The variable `isSupporter` has the value FALSE in a Standard-CLA.

`Boolean TBL`

`TBL` takes that value TRUE if the *CLA* is using the TBL selection policy and FALSE if the *CLA* is using the Standard selection policy.

3.3.11 *TruthTable*

The *TruthTable* module holds information about the domain and range sizes of the different *TruthTables,* their names, and the correct responses for each state of a given *TruthTable*.

`String names[]`

`names` is an array of the names of the TruthTables.

`integer intentExtentState[][]`

`intentExtentState` is an array of arrays of integers describing the dimensions of the *TruthTables*. The inner arrays are integer triplets that give the size of the intent, the size of the extent, and the number of states in each *TruthTable*. For the purposes of this research, *TruthTables* have the same intent and extent sizes for all states.

`integer aTruthTable[][]`

`aTruthTable` is a sample *TruthTable*. An array of integer arrays, each *TruthTable* contains an inner array for each state. Each position in an inner array represents a *Stimulant* and the value at that position in the array represents the correct respondent for that *Stimulant*. For example, `aTruthTable[3][]` may hold the array `{2,3,4,5,6`}. This means that for the third state of `aTruthTable` the correct stimulus-response pairs are (1, 2); (2, 3); (3, 4); (4, 5); and (5, 6).

To generate *TruthTables* that are not square, meaning that the extent is larger than the intent, cycling through the state array creates stimulus-response pairs that are not explicitly stated in the state array. This can be done by the *Manager* or the *Environment*. For example, if `aTruthTable[3][]` actually had an extent of size 10 instead of 5, then the remaining stimulus-response pairs would be (6,2); (7,3); (8,4); (9,5); and (10, 6).

3.3.12 Time Complexity of the TBL Algorithm

Almost all of the additional operations that take place when using TBL occur at the local level of a Stimulant. In the worst case scenario, a Seeker Stimulant, a stimulant that is in need of a tactic, would have to search through the entire list of global tactics only to discover that there are no new tactics available. The global list of tactics is limited by the size of the intent (*M*), and the intent is guaranteed to be significantly less than the size of the extent (*N*). Therefore the TBL algorithm does not increase *O(N)*, the time complexity based on the number of stimulants in the STM.

The only operation introduced by TBL that does require “touching” each stimulant is the resignation of a global tactic. When a global tactic resigns, it must be removed from any and all local tactic lists. This means that every Stimulant must be visited to insure that the resigning tactic is removed. Assuming that the STM has been implemented with some data structure with *O(N)* = log*N* search time, the removal of a global tactic requires *O*(*N*) = *N*log*N* time.

Postulate 10: The time complexity of the removal of a global tactic has

Equation 3

$$O(N) = \ N \log N$$

Postulate 11: The average time complexity increase for using TBL is

Equation 4

$$O(N)_{TBL} = (0, N]O(N)_{standard}$$

3.4 The TruthTable Game

A game, called the TruthTable game, is used for the experiments in this research. This section describes the game and introduces the variations that were used in the research.

3.4.1 Overview of the game

The TruthTable game is a solitaire classification task. The TruthTable is represented by an *n* by *m* array where there are *n* inputs and *m* outputs. For each input there must be at least one correct output, but there may be more than one correct output. The game has several states. Each state is a different organization of the correct input-output pairs.

Defintion 37: A **state** of the TruthTable game is one of its arrangements of input-output pairs.

As long as all inputs have the same number of correct outputs, the configuration of the correct outputs does not matter to a Standard-CLA. Each configuration is equivalent because a Standard-CLA treats each input as an independent entity with no relationship to any other input. For a Standard-CLA, the game only gets harder as the number of inputs increases. It is important to note that neither the order of the inputs nor the order of the outputs is significant. All four game states in Figure 4 are equivalent to a Standard-CLA; however, the states in Figure 5 are not equivalent.

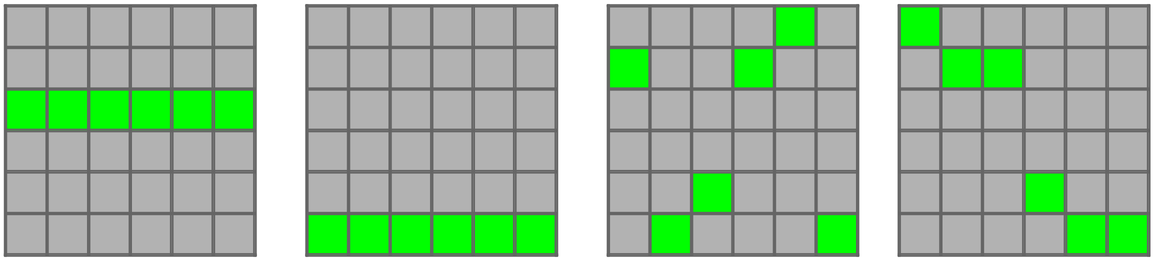


Figure 4: Sample Game States Each grid represents a single game state. In each grid, the inputs are the columns and the outputs are the rows. Each green cell represents the correct output for that input column. The object of the game is to find the correct output for every input.

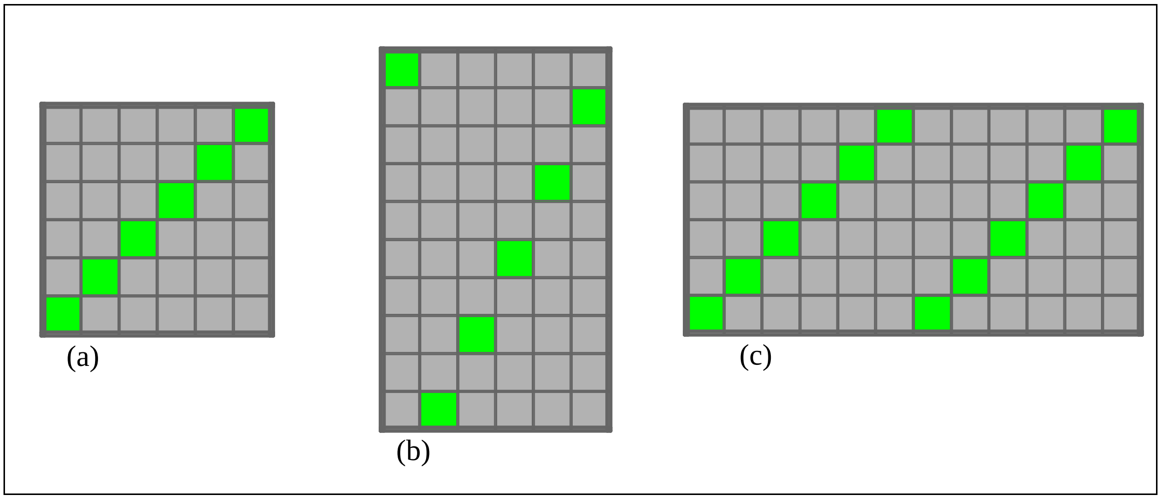


Figure 5: Non-equivalent Game States These three game states are different from the perspective of a Standard-CLA. State (b) has more outputs than (a); therefore, it takes a CLA longer to learn the correct output for each input for state (b). State (c) has more inputs than state (a), so while a CLA does not need more time to learn than it needs in state (a), there are twice as many inputs to be learned.

While the configuration of the outputs does not significantly affect the learning behavior of a Standard-CLA, a TBL-CLA is designed specifically to take advantage of the fact the some inputs might share the same output. Some configurations are more advantageous to a TBL-CLA than others. This advantage is called the **Tactic-Based Learning Advantage**, **TBL$_\alpha$**, of a game state. The more inputs share the same output and the lower the number of outputs used, the higher the TBL$_\alpha$.

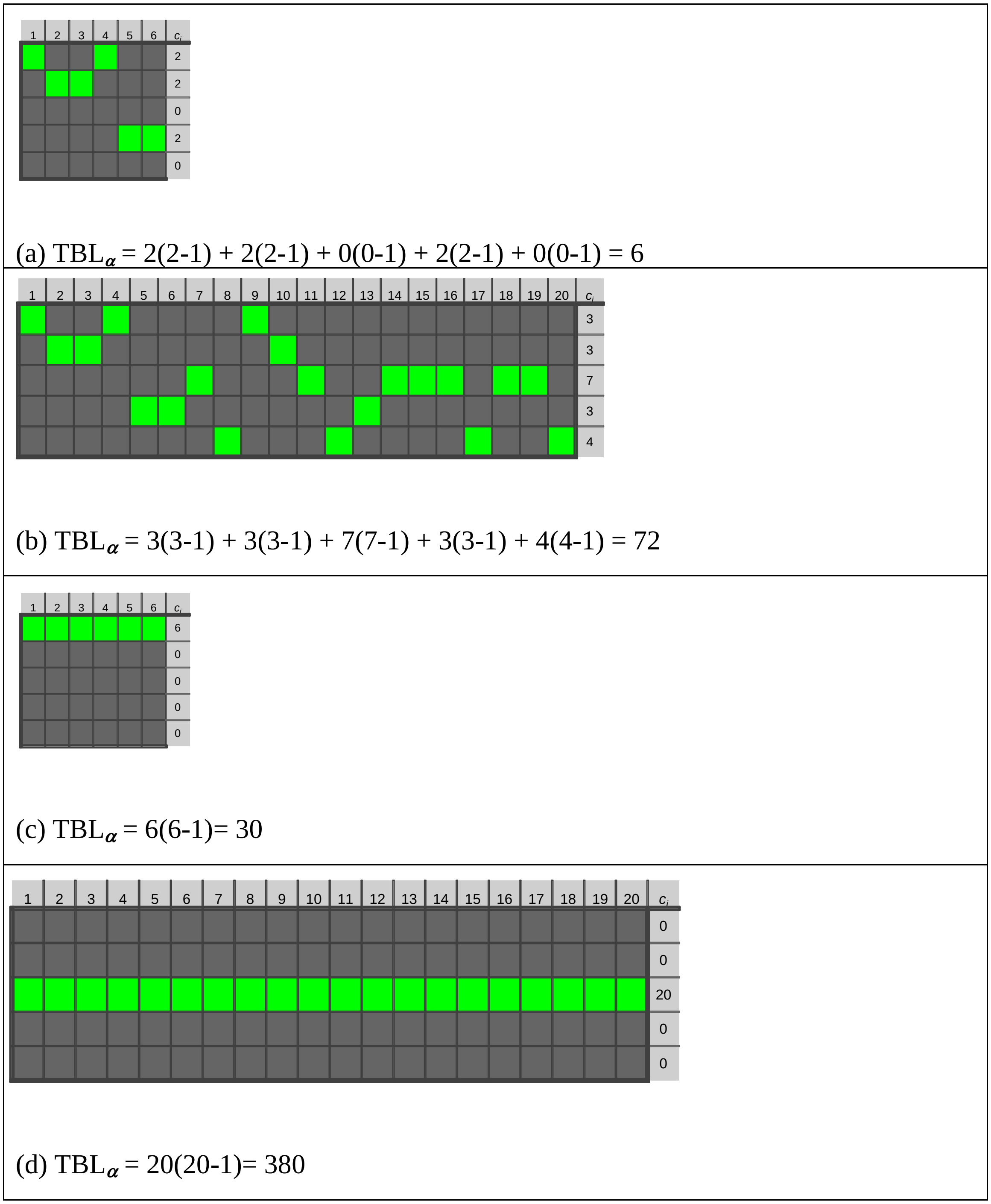
Postulate 12: The **Tactic-Based Learning advantage** (**TBL$_\alpha$**) of a given game state is computed as follows:

Equation 5

$$TBL_{\alpha} = \sum_{i=1}^{n} c_i(c_i - 1)$$

where *n* = the size of the range (the number of classes) and c_i = the number of inputs assigned to a given output

The TBL$_\alpha$, of a game state is simply a way to rank game states to determine the potential utility of using a TBL-CLA. It is expected that as the TBL$_\alpha$ rises, a reasonably configured TBL-CLA would outperform a Standard-CLA to a greater degree. Figure 6 shows some sample game states and their TBL$_\alpha$.


1 2 3 4 5 6 c_i
2
2
0
2
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 c_i
3
3
7
3
4
1 2 3 4 5 6 c_i
6
0
0
0
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 c_i
0
0
20
0
0

(a) $\text{TBL}_{\alpha} = 2(2\text{-}1) + 2(2\text{-}1) + 0(0\text{-}1) + 2(2\text{-}1) + 0(0\text{-}1) = 6$

(b) $\text{TBL}_{\alpha} = 3(3\text{-}1) + 3(3\text{-}1) + 7(7\text{-}1) + 3(3\text{-}1) + 4(4\text{-}1) = 72$

(c) $\text{TBL}_{\alpha} = 6(6\text{-}1)= 30$

(d) $\text{TBL}_{\alpha} = 20(20\text{-}1)= 380$

Figure 6: TBL$_{\alpha}$ for sample states (a) and (c) are equivalent states for a Standard-CLA as are states (b) and (d), however, all four states have different TBL$_{\alpha}$. The TBL$_{\alpha}$ increases as the number of inputs increases and as the correct outputs become less disperse.

3.4.2 Rules of the game

The object of the TruthTable game is to identify the correct output for each input. The TruthTable game is played as follows. The environment presents the CLA with a series of random inputs. The CLA chooses outputs and presents them to the environment. The environment periodically evaluates the CLA by scoring a group of outputs (the current history). The game stops when the CLA has reached an appropriate level of **confident accuracy** or the CLA runs out of time.

Postulate 13: The **confident accuracy** of a CLA is the average selection confidence for a given test point multiplied by the average score for the test point expressed as a percentage.

Postulate 14: The **scoring function** of the TruthTable game is the fraction of correct outputs in the current history.

3.4.3 Basic game play

For this research, the game is played in two phases. The first phase is called the **initial phase**. During the initial phase, a CLA learns to play on a small subsection of the eventual game state. The initial phase contains 6 inputs and 6 outputs. Once the CLA has reached a high level on confident accuracy on the initial phase (> 99.9%), the state is **extended** out along the domain (inputs) to its full size. The extended phase of the game is called the **secondary phase** and contains the 6 by 6 section of the initial phase and adds 294 new inputs for a 6 by 300 full sized state. In this research, the extension is achieved by repeating the pattern of correct input-output pairs several times until the domain is 300 inputs long. This process is shown in Figure 7.

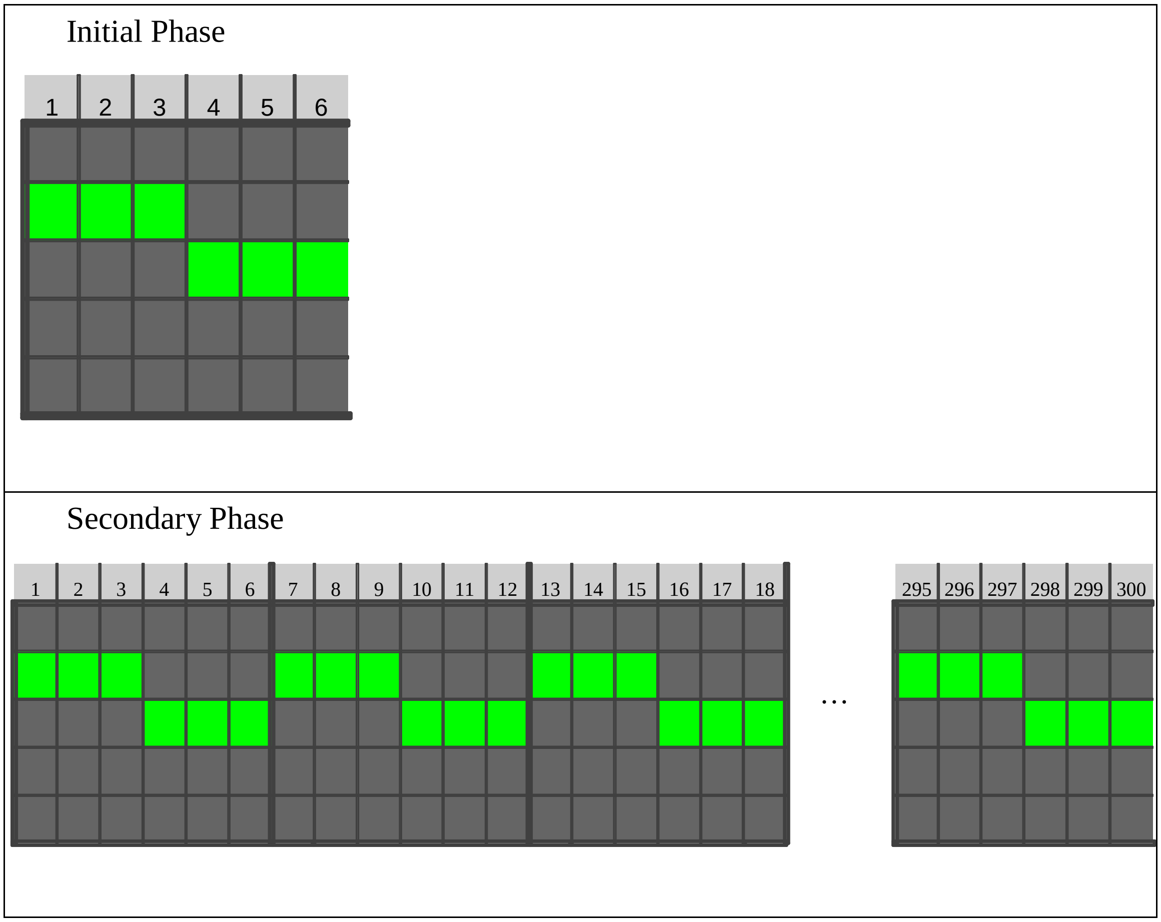


Figure 7: Sample Initial and Secondary Phases In the initial phase, a CLA trains with a small subsection of the game state. After achieving a confident accuracy > 99.9%, the secondary phase of the game begins and the remainder of the inputs are introduced. Note that the section of the game state used in the initial phase remains as part of game state. For this research, all game states are created by simply repeating the initial section of the state until there are 300 inputs in the state.

3.4.4 Justification of phased game play

Some may question the validity of phased game play because it would seem to give an additional advantage to the TBL-CLA. Allowing the Standard-CLA to train on a small section of the game state first gives it the opportunity to learn that section quickly because each input are seen more often, but that initial advantage is small once the secondary phase is entered. On the other hand, the TBL-CLA has not only mastered the inputs in the initial phase, it has identified all of the tactics it needs to succeed in the game. When the TBL-CLA starts the secondary phase, it only need identify the appropriate tactic for a given input and it has won the game.

Why not simply do away with the initial phase and start a CLA on the secondary phase? QD pilots have shown that doing so lessens the impact of tactic-based learning. By the time tactics have been identified, there are usually few stimulants left with a selection confidence low enough to take advantage of a tactic for long. The phased structure of the game can be justified from a social perspective and biological perspective. There are many examples in nature of adults shielding their young from the full range of experiences they will eventually have to face. Many species are born without the capacity to provide themselves with food even after a weaning period. Parents spend a great deal of time and energy to provide food for their young and to protect their young from threats they cannot handle yet.

From a biological perspective, human beings rely most heavily on vision to navigate and explore the world. In human infants, however, the vision system is not fully developed at birth. Infants' eyes do not develop the full range of photoreceptors on the retina until they are about 6 months old and it takes about 2 years for infants to be able to perform all visual tasks at the level of an adult. Clearly, many of these developments are delayed because there is little to stimulate vision in the womb, but this also serves as a protective measure, allowing the neonate to manage the violent transitions of birth in gentle stages by slowly adding more visual stimulation. There is also evidence that neonates that are overstimulated can become overwhelmed which causes them to withdraw from their environments. If this happens repeatedly, these children also suffer developmental delays and difficulties (Berk 2003).

By structuring the game play into stages, there is an acknowledged advantage being given to the TBL-CLA; however, this advantage has strong parallels in both biological and social strategies for supporting young learners.

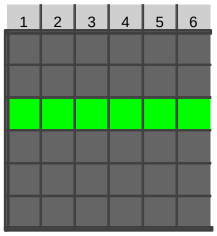
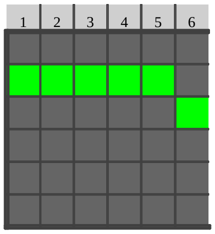
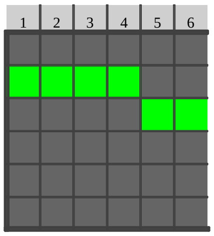
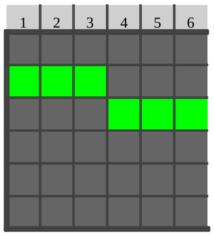
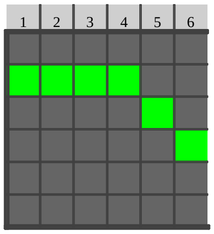
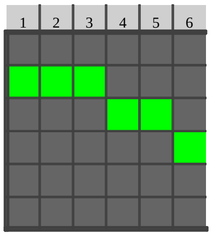
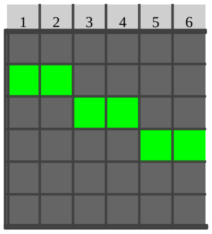
3.4.5 Game states used

All the canonical experiments in the research use a 6 by 6 subsection of the game state in the initial phase and then add 49 repetitions of the subsection for a total game state of 6 by 300. As was pointed out in Section 3.4.1, any arrangement of the target cells is equivalent to any other arrangement from a Standard-CLA's perspective, but the arrangement does affect how much advantage there is to be gained from using TBL. Different arrangements are used to explore the effect that the number of possible tactics available to a CLA has on its learning behavior. In order to minimize any extraneous advantage that might be given to a TBL-CLA by a fortunate arrangement of the target cells, only the arrangements with the lowest possible TBL$_{\alpha}$ and an equal number of target cells per target response were chosen. Figure 8 shows all possible game states for a 6 by 6 substate. The game states shown in Figure 9 are used as factors of the research.

Defintion 38: A **target cell** is a cell in a TruthTable, which signifies a correct input-output pair.

Defintion 39: A **target response** is an output in a TruthTable that is associated with at least one target cell.

Defintion 40: A **substate** is a small section of a state; it usually has all of the target responses in it.

1 target output  initial TBL$_{\alpha}$ = 30 secondary TBL$_{\alpha}$ = 89,700		
2 target outputs  initial TBL$_{\alpha}$ = 25 secondary TBL$_{\alpha}$ = 64,700	 initial TBL$_{\alpha}$ =14 secondary TBL$_{\alpha}$ = 49,700	 initial TBL$_{\alpha}$ =12 secondary TBL$_{\alpha}$ = 44,700
3 target outputs  initial TBL$_{\alpha}$ = 12 secondary TBL$_{\alpha}$ = 44,700	 initial TBL$_{\alpha}$ = 8 secondary TBL$_{\alpha}$ = 34,700	 initial TBL$_{\alpha}$ = 6 secondary TBL$_{\alpha}$ = 29,700

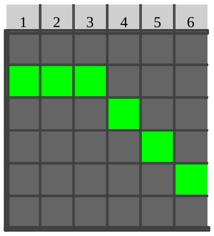
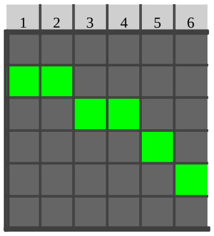
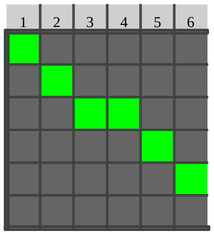
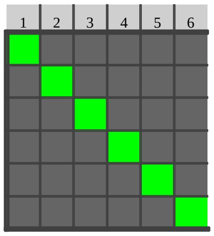
4 target outputs  initial $\text{TBL}_\alpha = 6$ secondary $\text{TBL}_\alpha = 29{,}700$	 initial $\text{TBL}_\alpha = 4$ secondary $\text{TBL}_\alpha = 24{,}700$	
5 target outputs  initial $\text{TBL}_\alpha = 2$ secondary $\text{TBL}_\alpha = 19{,}700$		
6 target outputs  initial $\text{TBL}_\alpha = 0$ secondary $\text{TBL}_\alpha = 14{,}700$		

Figure 8: All Possible Initial States All arrangements that have only one target output per input are equivalent to a Standard-CLA, but the arrangement does make a difference to a TBL-CLA. All possible arrangements of the initial state are given along with the TBL_α for both the initial and the secondary phases.

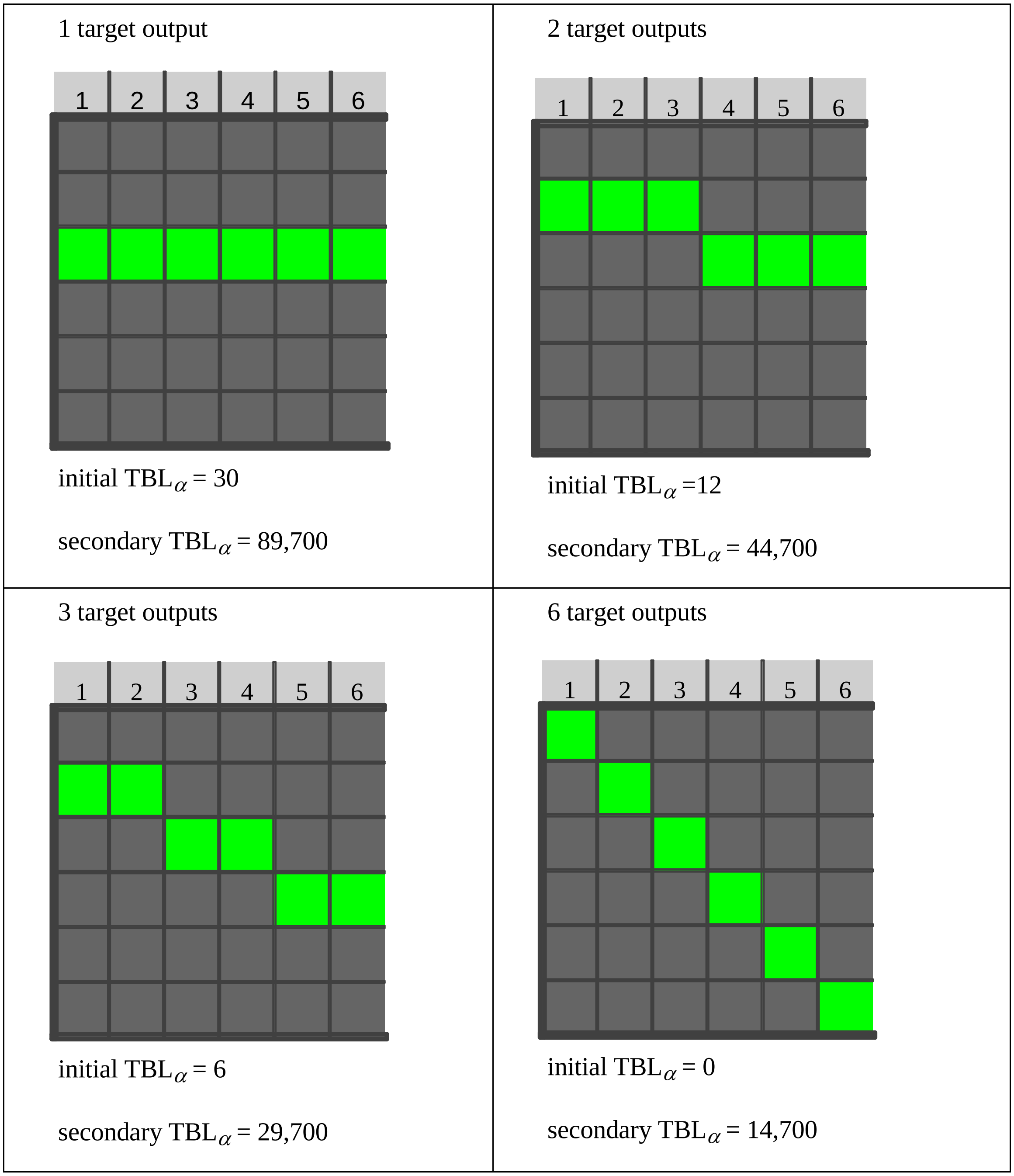


Figure 9: Initial States Used as Factors The initial states were chosen as the subset of all possible initial states to be used as factors. These states all have an equal number of inputs assigned to each target output and they each have the lowest possible TBL$_\alpha$ for the number of target outputs.

The details of the game that have been presented in this section represent the most basic version of the game. The game must be altered slightly to meet goals. All changes made to the basic rules of the game are explained thoroughly in Sections 4.1.4 and 4.1.5, which describe those experiments.

3.5 Goals

The goal structure in this section indicates how the application of the solution method achieves the research objective. The primary goal of the research is to measure the performance of a CLA using TBL. In order to achieve this primary goal, many subgoals must first be achieved. These subgoals include conducting operating point pilots to set the values of fixed parameters and conditions and to select appropriate ranges and increments for the factors (Goal 1), conducting formal, canonical experiments to measure the performance of a TBL-CLA in general (Goal 2), and the performance of a TBL-CLA in detail over a few representative cases (Goal 3). The factors and performance metrics for the formal experiments are defined in this section with their related goal and postulated in Section 3.6 (Performance Metrics).

Primary Goal: Measure a TBL-CLA's performance. The primary goal of the research is to measure the performance of a TBL-CLA.

G1: To specify all fixed parameters and conditions. There are many parameters and conditions that are not factors in the research. These parameters and conditions are fixed at reasonable levels and settings. These values are found through informal searches of the factor space.

Factors: collection length, *c*

reject threshold K_r

tie threshold K_t

minimum level of tactic potency, *p*

G2: To measure the general canonical behavior of a TBL-CLA. There is a need to demonstrate the general patterns of behavior in canonical situations. Results are presented exclusively as footprints.

G2.1: To measure TBL behavior when the *Environment* is stable and there is only one correct response for each stimulus.

Factors: collection length, *c*

TBL thresholds: support κ_s

withdrawal κ_w

independence κ_i

dependence κ_d

TruthTable game state

Performance Metrics: Payoff, *P*

n-tile advantage

Expense

G2.1.1: Determine appropriate factor ranges (OP Pilot)

Factors: TBL thresholds: support κ_s

withdrawal κ_w

independence κ_i

dependence κ_d

Performance Metrics: Payoff, *P*

n-tile advantage

Expense

G2.1.2: Determine appropriate settings for fixed conditions and parameters (OP Pilot)

Factors: compensation threshold, κ_γ

Performance Metrics: Payoff, *P*
n-tile advantage
Expense

G2.2: To measure TBL behavior when the *Environment* is stable and there are 2 correct responses for every stimulus.

Factors: collection length, *c*
TBL thresholds: support κ_s
withdrawal κ_w
independence κ_i
dependence κ_d
TruthTable game state

Performance Metrics: Payoff, *P*
n-tile advantage
Expense

G2.2.1: Determine appropriate factor ranges (OP Pilot)

Factors: TBL thresholds: support κ_s
withdrawal κ_w
independence κ_i
dependence κ_d

Performance Metrics: Payoff, *P*
n-tile advantage
Expense

G2.3: To measure TBL behavior when the *Environment* is not stable and there is only one correct response for each stimulus.

Factors: collection length, *c*

TBL thresholds: support κ_s

withdrawal κ_w

independence κ_i

dependence κ_d

TruthTable game state

Performance Metrics: Payoff, *P*

n-tile advantage

Expense

G2.3.1: Determine appropriate factor ranges (OP Pilot)

Factors: TBL thresholds: support κ_s

withdrawal κ_w

independence κ_i

dependence κ_d

Performance Metrics: Payoff, *P*

n-tile advantage

Expense

G2.3.2: Determine appropriate settings for fixed conditions and parameters (OP Pilot)

Factors: compensation threshold, κ_γ

Performance Metrics: Payoff, *P*

n-tile advantage

Expense

G3: To explore specific-TBL treatments of interest more deeply

G3.1: To explore specific cases of the results from G2.1. Further exploration into specific cases of the canonical experiments (stationary game, one response per stimulus).

Factors: case (average, strong TBL performance, poor TBL performance, any others that stand out for some reason)

Performance Metrics: learning curves
TBL roles
random responses

G3.1.1: Determine treatments to select

Factors: TBL thresholds: support κ_s
withdrawal κ_w
independence κ_i
dependence κ_d

Performance Metrics: Payoff, *P*
n-tile advantage
Expense

G3.2: To explore specific cases of the results from G2.2.

Factors: case (average, strong TBL performance, poor TBL performance, any others that stand out for some reason)

Performance Metrics: learning curves
TBL roles
random responses

G3.2.1: Determine treatments to select

Factors: TBL thresholds: support κ_s
withdrawal κ_w
independence κ_i
dependence κ_d

Performance Metrics: Payoff, *P*
n-tile advantage
Expense

G3.3: To explore specific cases of the results from G2.3. Further exploration into specific cases of the canonical experiments (task-switching game, one response per stimulus).

Factors: case (average, strong TBL performance, poor TBL performance, any others that stand out for some reason)

Performance Metrics: learning curves
TBL roles
random responses

G3.3.1: Determine treatments to select

Factors: TBL thresholds: support κ_s
withdrawal κ_w
independence κ_i
dependence κ_d

Performance Metrics: Payoff, *P*
n-tile advantage
Expense

3.6 Performance Metrics

This section discusses the performance metrics used in this research. They are Payoff, *n*-tile advantage, Expense, changing TBL roles, learning curves, and the number of random selections made during a test period. The first three metrics are used to measure overall performance across a wide range of factors and their values to provide significant statistical conclusions under a rigorous Monte Carlo test protocol. The last three metrics are applied to single treatments only to illustrate some examples of special and unusual behavior. These two goals and their associated experiments, results, and conclusions are referred to as the formal and informal goals.

In all the experiments, the performance of the TBL-CLA is compared with the performance of a Standard-CLA, both of which are implemented under the same parameters and conditions (game state, collection length, etc.). Because many of the metrics may need to compare the performance of two players that finish the task at different times, a few important terms must be defined before the postulated performance metrics can be understood. To assist in this process, Figure 10 represents two hypothetical learning curves of the two types of CLA during a single treatment.

Because the summary performance is computed by comparing the individual performances of the two CLAs, which can complete the game at different times, it is important to record the results at the two different termination points. The first termination point is the contest at which either CLA first satisfies the stopping criterion. The second termination is the contest number at which the other CLA eventually satisfies the stopping criteria. Recall that the stopping criterion is satisfied when either one of the CLAs reaches a confident accuracy > 99.9%, or the match has reached 100,000 contests.

Defintion 41: The **first termination**, $\boldsymbol{t_1}$, is the contest at which the first CLA satisfies the stopping criterion.

Defintion 42: The **second termination**, $\boldsymbol{t_2}$, is the contest at which the second CLA reaches the stopping criterion.

Recall that the TruthTable game is played in at least two phases. In the initial phase, a CLA trains on a small subsection of the eventual problem. The performance metrics never include the initial phase, but are only computed for the secondary or tertiary phase as appropriate.

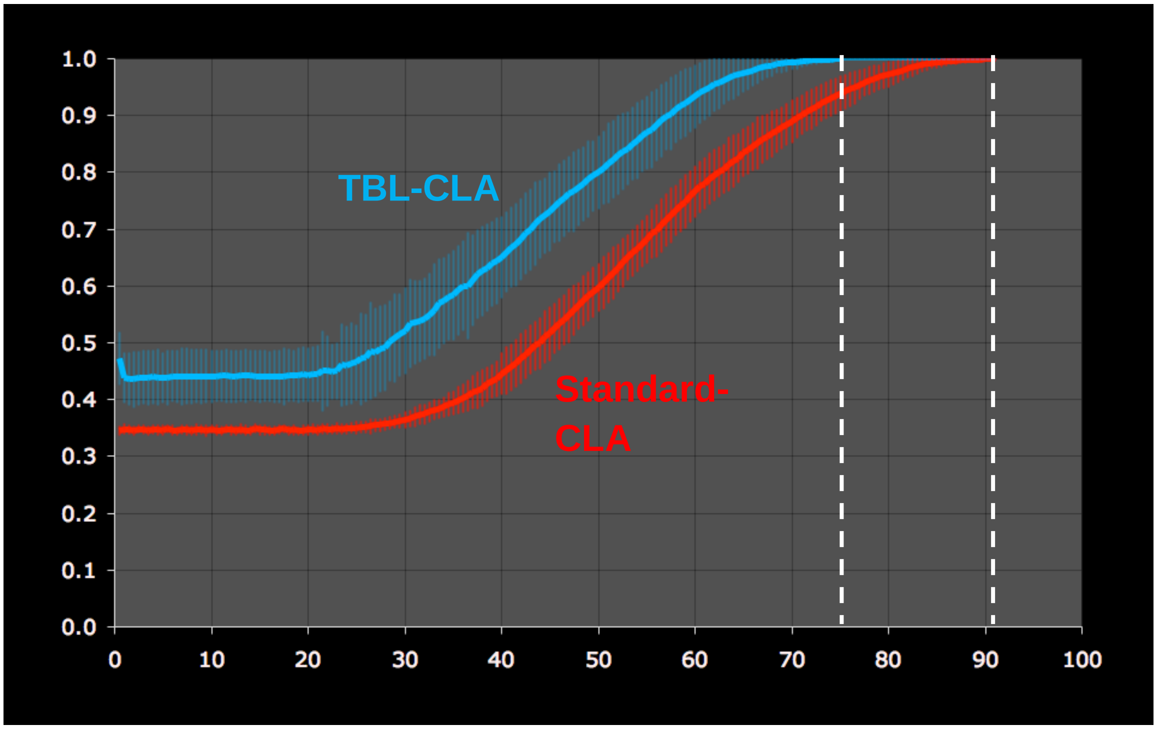


Figure 10: Learning curves for a single treatment The blue and red lines represent the Monte Carlo average score of the TBL-CLA and the Standard-CLA, respectively. The vertical error bars represent the 95% confidence interval. The first dotted line which extends down to about 75,000 contests marks the first termination. The second dotted line, extending down to about 91,000 contests, marks the second termination. A CLA trains until it achieves a confident accuracy > 99.9% or until it reaches 100,000 contests. For the CLA that finishes first, the last performance measures are to compute the remaining metrics for the CLA that finished second.

3.6.1 Payoff

At each test point, it is possible to measure the difference between the scores of the two CLAs. Because these differences are Monte Carlo averages of 30 instances of the same CLA started with different seeds for the random number generators, it is also possible to compute the confidence that the H_0 that there is no difference between the two scores can be rejected. The benefit at each test point is calculated as the confident difference between the two scores.

Postulate 15: the **benefit**, $\boldsymbol{b_i}$, at a given test point is computed as follows:

Equation 6

$$\boldsymbol{b_i} = \mathscr{R}_{H0} \left(s_{TBL} - s_{Standard}\right)$$

where s_{TBL} and $s_{Standard}$ are the scores of the TBL and Standard-CLAs, respectively, and $\mathscr{R}_{H0}$ is the confidence with which the null hypothesis H_0 that there is no difference between the two scores can be rejected.

The **Payoff**, ***P***, is the average of the benefits over all the test points to first termination. Using Payoff as a metric allows one treatment to be equitably compared with another treatment.

Postulate 16: the **Payoff**, ***P*,** of a treatment is:

Equation 7

$$P = \frac{\sum_{i=1}^{n} b_i}{n_{t_1}}$$

3.6.2 *n*-tile advantage

Payoff is a summary metric that assigns a single value to the entire learning process. Because it is an average of the benefit, it can necessarily hide some important aspects of the learning curve. For example, if a TBL-CLA and the Standard-CLA are never statically different in the learning curves, the Payoff is zero. It is also possible that the TBL-CLA has a positive benefit for about half of the test points and has a negative benefit of equal magnitude for the other half of the test points. In that case, the Payoff would also be close to 0.0.

The *n*-tile advantage is used to capture the learning curve in a way that is still easily visualized. The confidence that the two scores are different, $reject_{H_0}$, is measured at 10% of the first termination period, 20%, 30%, and so on. If the first termination period is 20,000 contests, then the confidence is measured at the following contests: 2,000; 4,000; 6,000; 8,000; 10,000; 12,000; 14,000; 16,000; 18,000; and 20,000. A confidence of 95% means that the TBL-CLA's score is greater than the Standard-CLA's score with confidence of 95%. A confidence of -95% means that the TBL-CLA's score is less than the Standard-CLA's score with a confidence of 95%. Because the *n*-tile advantage only reports a confidence, there is no indication of the size of the difference.

Postulate 17: The ***n*-tile advantage** of a treatment at a given contest is the two-tailed rejection confidence of H_0.

3.6.3 Expense

The **Expense** is the number of contests between the TBL-CLA's termination and Standard-CLA's termination. This is a way of measuring how much effort is saved *vis-à-vis* the Standard-CLA. The Expense is also presented in the footprint format to allow for comparison between treatments.

Postulate 18: The **Expense** is the difference between the number of contests the TBL-CLA needs to terminate and the number of contests the Standard-CLA needs to terminate.

With Expense, a negative number means that it took the TBL-CLA fewer contests to reach the termination conditions that the Standard-CLA required.

3.6.4 Footprints

Payoff, *n*-tile advantage, and Expense are all aggregate metrics that can be used to equitably compare different treatments. In order to see the interactions for the factors on the learning behavior, the aggregate metrics are presented in **footprints**. A footprint is a table in which performance measures are shown by changes in color.

A sample footprint is presented in Figure 11. Each cell in the footprint represents the sample measure for an individual treatment. For all experiments, only results from the final phase of the game are presented, so the initial phase of learning, when CLAs train on a subsection of the problem, is not included.

All footprints are presented in the same format. The metric name is the header of the footprint. The columns are divided by collection length and then subdivided by the number of target responses in the game state. The TBL factors, the thresholds, are the variables in the rows. The order of the thresholds is always, from left to right, the support threshold (κ_s), the withdrawal threshold (κ_w), the independence threshold (κ_i), and the dependence threshold (κ_d).

The scale that governs the coloring of the cells is not symmetrical around zero. Payoff, *n*-tile advantage, and Expense can all be positive or negative. In the footprints, pure white is always reserved for zero, but in order to highlight the important extremes, the distribution of the hues is not always proportional. It is important to note that the color scales are adjusted for each experiment, so care must be taken when reading footprints.

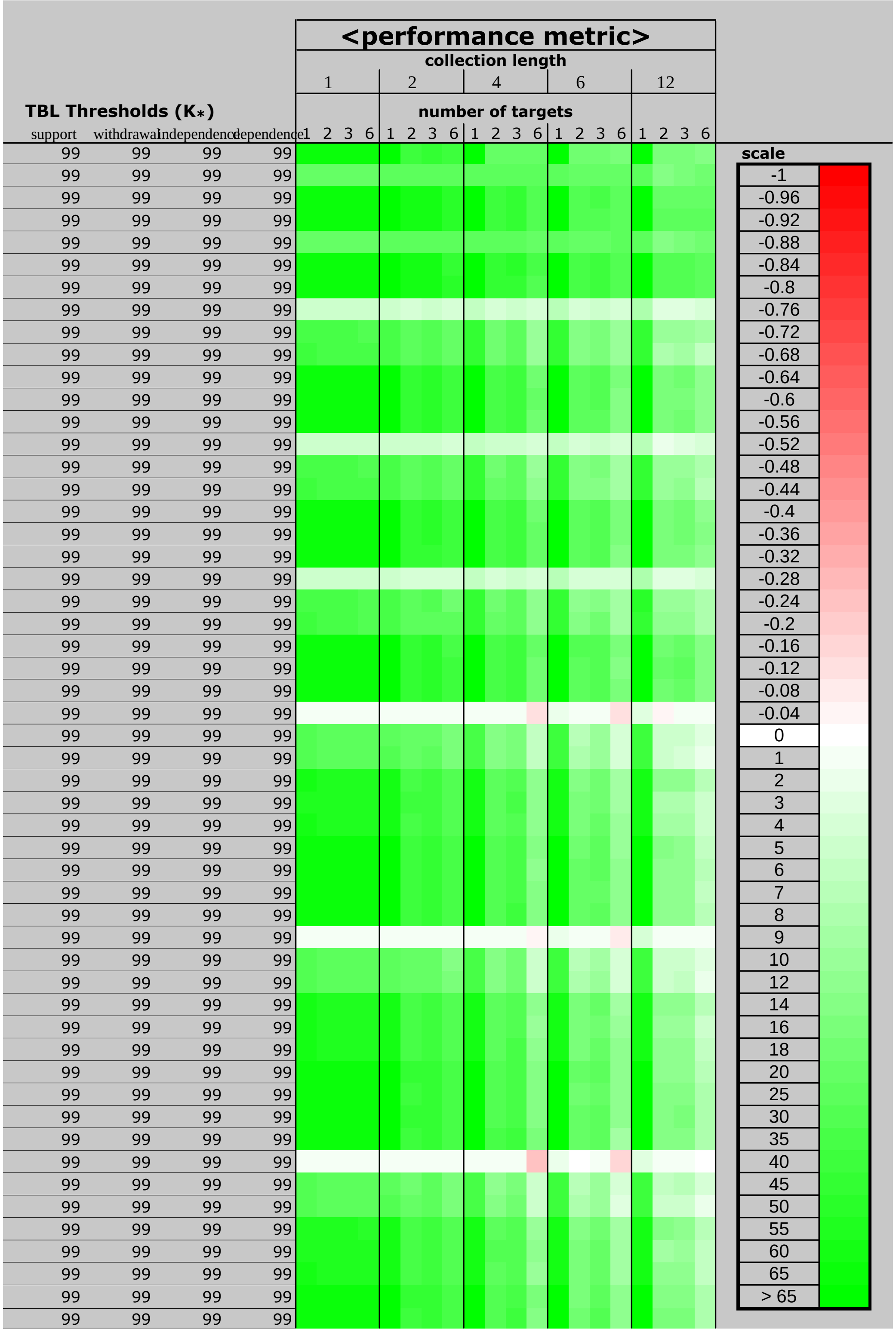


Figure 11: Sample Footprint the influence of the TBL factors (K*), collection length, and number of target responses is visualized by color shading. Note that the shading is not symmetrical around zero, nor are the positive or negative scales distributed linearly.

3.6.5 Learning curves

A learning curve is a graph which describes the scores of the CLA at each test point. The learning curve graph is useful for examining the behavior of the CLAs during a single treatment. The learning curves are only used for examining the results of a single treatment in greater detail. Each point on the lines of the graph represents a Monte Carlo average of 30 CLAs with different seeds for the necessary pseudo-random number generators. The error bars are the 95% confidence interval for the averages. Figure 12 shows a sample learning curve.

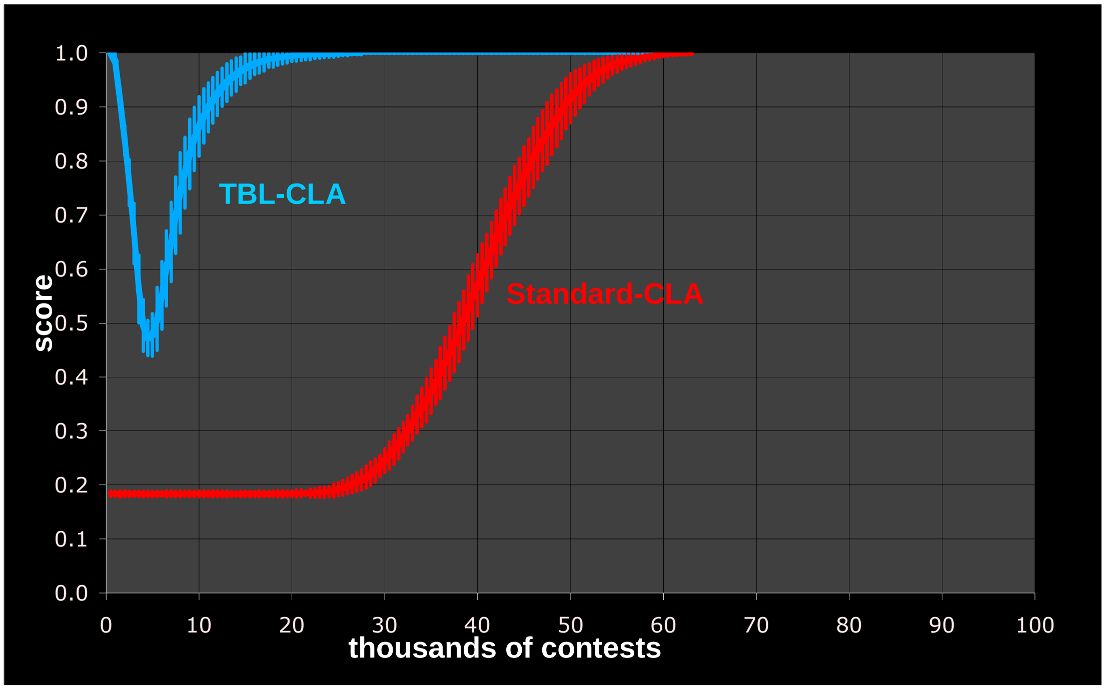


Figure 12: Sample Learning Curve the learning curve shows the Monte Carlo averages if the scores at each test point. The error bars represent the 95% confidence interval.

While the learning curve is not truly a metric in itself, it is a very useful tool for visualizing learning behavior. The error bars are 95% confidence intervals, making it is easy to see when the scores are significantly different.

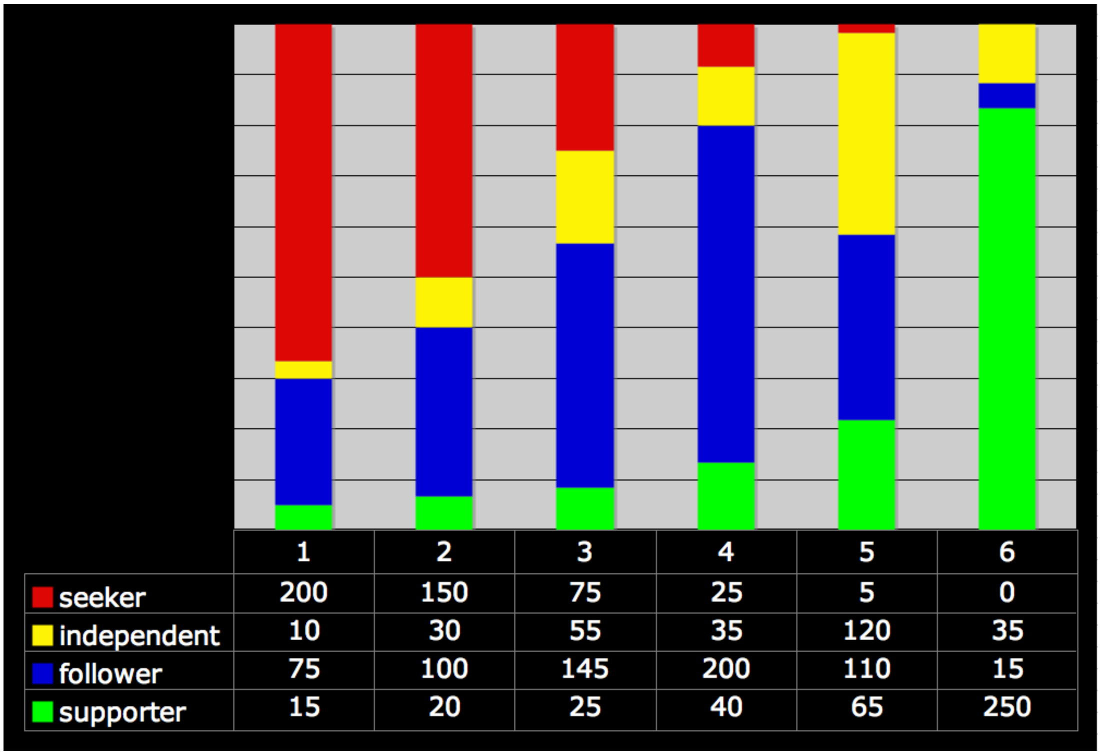
When one CLA reaches the termination condition before another, its final score is extended out until the second CLA reaches the termination conditions. This extension is visualized in a lighter color of the original line.

3.6.6 TBL roles

In order to assess and understand the behavior of a TBL-CLA, it is necessary to understand the impact of the different roles that a stimulant can be in at any given time. These roles – supporter, independent, follower, and seeker – determine how the stimulant selects its respondent. The TBL role graph displays the role of each stimulant at each test point during learning for a single treatment. This graph can only be used to show the results from one treatment.

The roles are presented in a stacked bar graph. Figure 13 contains a simplified version of the role graph. Each column always contains the same total number of stimulants, 300 stimulants in the canonical experiments. Each column represents the stimulants at a given test point. The distribution of stimulants in different roles is shown by the size of colored section corresponding to a role.

Figure 14 is a sample TBL role graph as it is presented in the results. In addition to the distribution of the roles, the TBL role graph also includes a line which represents the score, or learning curve, of the CLA. Observation of the relationship between the score of the TBL-CLA and the changing distribution of the roles is the basis for conclusions about the influence of the roles on the learning behavior.



	1	2	3	4	5	6
🟥 seeker	200	150	75	25	5	0
🟨 independent	10	30	55	35	120	35
🟦 follower	75	100	145	200	110	15
🟩 supporter	15	20	25	40	65	250

Figure 13: Simplified Stacked Bar Graph of TBL Roles each column represents all the stimulants in the STM at a given test point. The different colors in each column represent the distribution of roles that the stimulants are in at the test point. The number of stimulants is always the same, but the roles change. The data table shows to actual distribution of stimulants into the different roles. As time goes on, the number of seeker stimulants goes down with the number of supporters goes up.

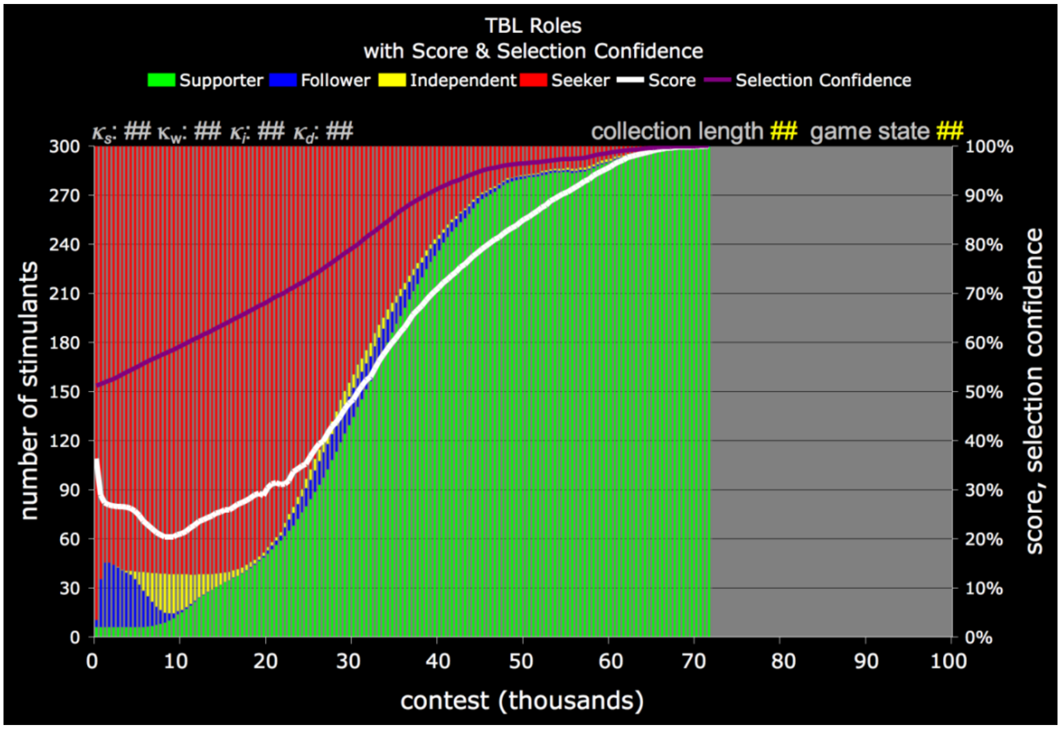


Figure 14: Sample TBL Role Graph the CLA uses the greatest number of follower stimulants early in learning. There is a dip in the score as the number of independent stimulants rises, causing an increase in the number of random choices being made. The increase in the number of supporter stimulants is consistent with the increase in the score. Throughout learning, a small number of follower and independent stimulants remain are extant. By the time the CLA has met the termination conditions, all of the stimulants are supporters.

3.6.7 Random responses

One of the objectives of this research is to develop a solution which reduces the reliance on pseudo-random number generators. The number of random responses made during a treatment by the CLAs is a way to assure that this is the case. A random response is a response which required a tie to be broken. These responses would include those in which a CLA did not have a confident respondent or when it was breaking the tie between a primary and secondary response. Follower stimulants do not make random responses because they are choosing the response indicated by a tactic. In a Standard-CLA, these followers would be following the Standard selection policy and selecting their responses at random. A sample graph is presented in Figure 15.

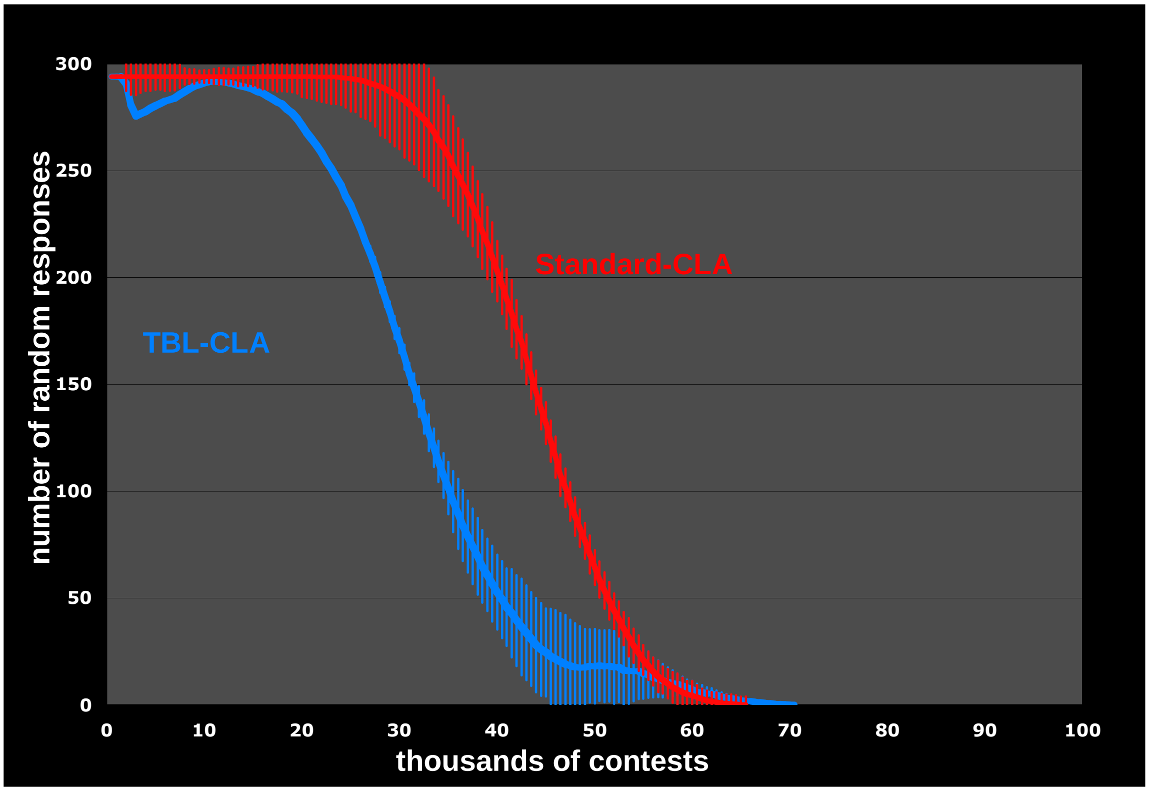


Figure 15: **Sample Graph of the Random Responses**.

CHAPTER 4: EXPERIMENTS

4.1 Experiment Design

This section describes the design of the experiments that are necessary to achieve the research objective. These include informal operating point pilots (OP pilots) as well as the formal experiments.

4.1.1 OP Pilots for fixed conditions and parameters

It was necessary to fix several conditions and parameters in order to limit the scope of the research. Informal operating point pilots were conducted to establish these fixed values, which are presented in **Table 2** below. These OP pilots were necessary to accomplish Goal 1, described in Section 3.5.

Table 2: Parameters and Conditions for the TruthTable OP Pilots

Category	Name	Selected Fixed Values
TruthTable Game conditions		
	initial phase: intent (size)	6
	initial phase: extent (size)	6
	secondary phase: intent (size)	6
	secondary phase: extent (size)	300
	game states	{1, 2, 3, 6} with lowest TBL$_\alpha$ (see Section 3.4 for details)
	maximum number of contests per phase	100,000
Standard CLA parameters		
	collection length, c	{1, 2, 4, 6, 12}
	reject threshold, κ_r	95%
	tie threshold, κ_t	98%
TBL-CLA parameters		
	minimum local potency	1
Experiment conditions		
	number of CLAs per treatment	30
	number of contests between test points	500

All of the TruthTable game conditions are described in Section 3.4.5. The Standard CLA parameters are discussed in Section 3.3.4. The TBL-CLA parameter, minimum local potency, is described in Section 3.3.6.

The experiment conditions deserve a little more explanation. Each treatment is run on 30 independent CLAs which have each been given different seeds for their required pseudo-random number generators. Each CLA is presented with 500 contests between test points. A contest consists of a single stimulus. If a CLA reaches 500 contests in the middle of the collection length, it completes the collection length and receives its evaluation before it begins testing. The periods between test points are known as training periods. During test points, learning is “turned off” by withholding evaluation. The CLA is presented with the same test set each time. The test set is made up of each input repeated 30 times. For example, if the CLA is in the initial phase, it is presented with a total of 180 test contests. If the CLA is beyond the initial phase, it is presented with 9000 test contests. The CLA does not receive any evaluation on the test set although all the data from the test set in recorded. Data is recorded during test points

The data collected during the experiments is presented in **Table 3**. All data collection is done during the test periods. Additionally, all the settings for the factors as well as the fixed parameters and conditions are recorded once, as part of the header to the data file. In the table below, experimental data from Standard-CLAs include the first two categories, while experimental data from TBL-CLAs include all three categories.

Table 3: Data collected during experiments at each test point

Category	Data Items
TruthTable game parameters	
Contest number of each test point	
Standard CLA results	
Score, *s*	
Selection confidence	
Number of respondents selected at random	
TBL-CLA results	
Number of global tactics	
Number of supporter stimulants	
Number of independent stimulants	
Number of follower stimulants	
Number of seeker stimulants	

4.1.2 Organization of Experiments

In order to accomplish the remaining goals, three canonical experiments are needed. The remaining goals are about measuring the performance of a TBL-CLA under different environmental conditions: a stationary game with a single target per input, a stationary game with two targets per input, and a task-switching game with a single target per input. The results from these three experiments can be used for the goals that deal with generalized behavior and the goals that deal with the behavior of specific factor combinations.

The following factors are influenced by the TruthTable game conditions and must be specified and fixed through informal operating point pilot experiments, known as OP pilot experiments, under each set of conditions: the TBL thresholds and the compensation threshold.

4.1.3 Stationary Game, One Target Cell per Input

This section describes the informal OP Pilots necessary to determine the appropriate TBL and compensation thresholds as well as the canonical experiment.

4.1.3.1 OP Pilot 1: TBL thresholds, Stationary Game, One Target Cell per Input

Informal observation shows that the TBL thresholds dramatically impact the behavior of a TBL-CLA; however, the behavior of the TBL-CLA is also affected by the environmental conditions of the TruthTable game. It was observed through informal pilots that one set of TBL threshold factors was not sufficient to provoke a sufficiently wide range of behavior across all game conditions.

In order to determine the appropriate factor range, the following OP pilot was conducted to accomplish Goal 2.1.1 (see Section 3.5). The collection length, *c*, was fixed at 12; the compensation threshold, κ_γ, was fixed at 60; and the state of the TruthTable game was set to state 6 (*i.e.* there were six target responses in the game; for more detail see Section 3.4.5). TBL-CLAs were then trained with the TBL thresholds listed below in a Monte Carlo trial (30 iterations). The performance measures were calculated using the Monte Carlo trial results of a Standard-CLA under the same game conditions.

Table 4: OP Pilot Experiment Design for TBL thresholds

	TBL threshold 4-tuples	**number of treatments**
OP Pilot factors	<($\kappa_s, \kappa_w, \kappa_i, \kappa_d$)> selected from the following set of values, subject to the specified constraints: {50.00, 51.00, 55.00, 60.00, 65.00, 70.00, 75.00, 80.00, 85.00, 90.00, 95.00, 98.00, 99.99}	4550
Results	<($\kappa_s, \kappa_w, \kappa_i, \kappa_d$)> selected from the following set of values, subject to the specified constraints: {50.00, 55.00, 70.00, 80.00, 95.00, 99.99}	266

Note that the number of treatments is not simply the set of all possible 4-tuples that could be constructed from the factor values for the TBL thresholds. The TBL thresholds are subject to the following constraints:

- The withdrawal threshold must be less than or equal to the support threshold.

  $\kappa_w \leq \kappa_s$

- The independence threshold must be less than or equal to the support threshold.

  $\kappa_i \leq \kappa_s$

- The dependence threshold must be less than or equal to the independence threshold.

  $\kappa_d \leq \kappa_i$

4.1.3.2 OP Pilot 2: Compensation Threshold, κ_γ, Stationary Game, One Target Cell per Input

Once the TBL threshold factor range was fixed, the following OP pilot was conducted to accomplish Goal 2.1.2 (see Section 3.5). One highly successful factors combination, one neutral combination, and one unsuccessful combination were selected. Each TBL threshold factor combination was applied to a TBL-CLA. Each TBL-CLA then completes Monte Carlo trials with different values for κ_γ. The OP pilot factors and the results are shown in Table 5.

Table 5: OP Pilot Experiment Design for κ_γ

OP Pilot factor values for κ_γ	**fixed value for κ_γ**
{50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 98, 99}	**50**

4.1.3.3 Experiment 1: Canonical, Stationary Game, One Target Cell per Input

After determining the appropriate factor values for the TBL threshold and the compensation threshold, the following experiment is conducted to accomplish Goal 2.1, described in Section 3.5. The TruthTable game environment is stationary throughout the learning process, that is, the arrangement of target cells never changes, and there is only one target cell per input. Below in **Table 6** is an experiment block design with all of the factors. Both Standard and TBL-CLAs are used. The results from the Standard-CLA are used as a baseline for computing the performance measures.

Table 6: Design of Experiment 1 (stationary game, 1 target cell per input)

Factor Name	**Values**	**Treatments**
TruthTable game state	{1, 2, 3, 6}	4
Collection length, *c*	{1, 2, 4, 6, 12}	5
TBL thresholds, κ*	<($\kappa_{s,}$ $\kappa_{w,}\kappa_{i,}$ κ_d)> selected from the following set of values, subject to the specified constraints: <50.00, 55.00, 70.00, 80.00, 95.00, 99.99>	266
Experiment resource requirements		
	Total treatments	**5340***
	Estimated time per treatment	2 CPU minutes
	Estimated total CPU time required	7.5 CPU days
	Total CPUs available	4
	Estimated time required	**2 days**
	* total number of treatments calculated as follows: Standard-CLA treatments = 4(5) TBL-CLA treatments = 4(5)(266) Total = Standard-CLA treatments + TBL-CLA treatments = 5340	

4.1.3.4 OP Pilot 3: Select factor combinations for close inspection, Stationary Game, One Target Cell per Input

Experiment 1, described in Section, generates sufficient data to accomplish Goal 3.1, described in Section 3.5. In this experiment, the behavior of a few specific TBL-CLAs is examined more closely.

Before Goal 3.1 can be accomplished, treatments must be selected for closer examination (Goal 3.1.1). An informal OP pilot was conducted in which the results of Experiment 1 were analyzed to determine which treatments presented an average case, strong case, and a weak case. The following TBL-threshold combinations are considered:

- Strong (70.00, 70.00, 70.00, 70.00)
- Average (99.99, 95.00, 50.00, 50.00)
- Weak (95.00, 95.00, 95.00, 50.00)

4.1.3.5 Experiment 2: Close inspection, Stationary Game, One Target Cell per Input

Experiment 2 accomplishes Goal 3.1, described in Section 3.5. The calculation of the individual performance measures is completed with the use of a spreadsheet, but the process of inspecting the results and drawing conclusions about each case requires significant time and attention from a person. **Table 7** below presents the block design for Experiment 2.

Table 7: Design for Experiment 2 (close inspection, 1 target cell per input)

Name	**Factor values**	**Treatments**
Target responses	{1, 2, 3, 6 }	4
Collection length, *c*	{1, 2, 4, 6, 12 }	5
TBL threshold 4-tuples <($\kappa_{S,}$ $\kappa_{W,}$$\kappa_{i,}$ κ_d)>	(70.00, 70.00, 70.00, 70.00) (99.99, 95.00, 50.00, 50.00) (95.00, 95.00, 95.00, 50.00)	3
Experiment resource requirements		
Total treatments		**80***
Estimated time per treatment		10 person-minutes
Estimated total time required		**14 hours**
* total number of treatments calculated as follows: Standard-CLA treatments = 4(5) TBL-CLA treatments = 4(5)(3) Total = Standard-CLA treatments + TBL-CLA treatments = 80		

4.1.4 Stationary Game, Two Target Cells per Input

This section describes the informal OP Pilots necessary to determine the appropriate TBL and compensation thresholds as well as the canonical experiment.

4.1.4.1 OP Pilot 4: TBL thresholds, Stationary Game, Two Target Cells per Input

In order to determine the appropriate factor range, the following OP pilot was conducted in order to accomplish Goal 2.2.1 (see Section 3.5). The collection length, *c*, was fixed at 12; the compensation threshold, κ_γ, was fixed at 60; and the TruthTable game was set to state 6 (*i.e.* there were six target responses in the game, for more detail see Section 3.4.5). TBL-CLAs were then trained with the TBL thresholds listed below in a Monte Carlo trial. The performance measures were calculated with Monte Carol trial results of a Standard-CLA under the same game conditions.

Table 8: OP Pilot Experiment Design for TBL thresholds

	TBL threshold 4-tuples	**number of treatments**
OP Pilot factors	<($\kappa_{s,}$ $\kappa_{w,}\kappa_{i,}$ κ_d)> selected from the following set of values, subject to the specified constraints: <50.00, 60.00, 70.00, 80.00, 90.00, 95.00, 98.00, 99.99>	750
Results	<($\kappa_{s,}$ $\kappa_{w,}\kappa_{i,}$ κ_d)> selected from the following set of values, subject to the specified constraints: <50.00, 70.00, 90.00, 95.00, 99.99>	66

Note that the number of treatments is not simply the set of all possible 4-tuples that could be constructed from the factor values for the TBL thresholds. The TBL thresholds are subject to the following constraints:

- The withdrawal threshold must be less than or equal to the support threshold.

  $\kappa_w \leq \kappa_s$

- The independence threshold must be less than or equal to the support threshold.

  $\kappa_i \leq \kappa_s$

- The dependence threshold must be less than or equal to the independence threshold.

  $\kappa_d \leq \kappa_i$

4.1.4.2 OP Pilot 5: Compensation Threshold, κ_γ, Stationary Game, Two Target Cells per Input

Once the TBL threshold factor range was fixed, the following OP pilot was conducted to accomplish Goal 2.1.2 (see Section 3.5). One highly successful factor combination, one neutral combination, and one unsuccessful combination were selected. Each TBL threshold factor combination was applied to a TBL-CLA. Each TBL-CLA then completed Monte Carlo trials with different values for κ_γ. The OP pilot factors and the results are shown in Table 9.

Table 9: OP Pilot Experiment Design for κ_γ

OP pilot factor values for κ_γ	**fixed value for κ_γ**
{50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 98, 99}	**50**

4.1.4.3 Experiment 3: Canonical, Stationary Game, Two Target Cells per Input

After determining the appropriate factor values for the TBL threshold and the compensation threshold, the following experiment is conducted. The TruthTable game environment is stationary throughout the learning process; that is, the arrangement of target cells never changes, and there is only one target cell per input. Below in Table 10 is an experiment block design with all of the factors. Both Standard and TBL-CLAs are used. The results from the Standard-CLA are used as a baseline for computing the performance measures.

Table 10: Design for Experiment 3 (canonical, stable game, 2 target cells per input)

Name	**Factor values**	**Treatments**
Target responses	{1, 2, 3, 6 }	4
Collection length, *c*	{1, 2, 4, 6, 12 }	5
TBL thresholds, κ*	$<(\kappa_{s}, \kappa_{w}, \kappa_{i}, \kappa_{d})>$ selected from the following set of values, subject to the specified constraints: <50.00, 70.00, 90.00, 95.00, 99.99>	66

Experiment resource requirements	
Total treatments	**1340***
Estimated time per treatment	2 CPU minutes
Estimated total CPU time required	1.9 CPU days
Total CPUs available	4
Estimated total time required	**1 day**
* total number of treatments calculated as follows: Standard-CLA treatments = 4(5) TBL-CLA treatments = 4(5)(66) Total = Standard-CLA treatments + TBL-CLA treatments = 1340	

4.1.4.4 OP Pilot 6: Select factor combinations for close inspection, Stationary Game, Two Target Cells per Input

Experiment 3 generates sufficient data to accomplish Goal 3.2, described in Section 3.5. In this experiment, the behavior of a few, specific TBL-CLAs is examined more closely.

Before Goal 3.2 can be accomplished, treatments must be selected for closer examination (Goal 3.2.1). An informal OP pilot was conducted in which the results of Experiment 1 were analyzed to determine which treatments presented an average case, strong case, and a weak case. The following TBL-threshold combinations are considered:

- Strong (70.00, 70.00, 70.00, 50.00)
- Average (99.99, 99.99, 99.99, 99.99)
- Weak (99.99, 50.00, 50.00, 50.00)

4.1.4.5 Experiment 4: Close inspection, Stationary Game, Two Target Cells per Input

Experiment 4 accomplishes Goal 3.2, described in Section 3.5. The calculation of the individual performance measures is completed with the use of a spreadsheet, but the process of inspecting the results and drawing conclusions about each case requires significant attention for a person.

Table 11 below presents the block design for Experiment 4.

Table 11: Design for Experiment 4 (close inspection, stationary game, two target cells per input)

Name	Factor values	Treatments
TruthTable game states	{1, 2, 3, 6 }	4
Collection length, *c*	{1, 2, 4, 6, 12 }	5
TBL threshold 4-tuples <($\kappa_{s,}$ $\kappa_{w,}$$\kappa_{i,}$ κ_{d})>	(70.00, 70.00, 70.00, 70.00) (99.99, 99.99, 99.99, 99.99) (99.99, 50.00, 50.00, 50.00)	3
Experiment resource requirements		
Total treatments		**80***
Estimated time per treatment		10 person-minutes
Estimated time required		**14 hours**
* total number of treatments calculated as follows: Standard-CLA treatments = 4(5) TBL-CLA treatments = 4(5)(3) Total = Standard-CLA treatments + TBL-CLA treatments = 80		

4.1.5 Task-switching Game, One Target Cell per Input

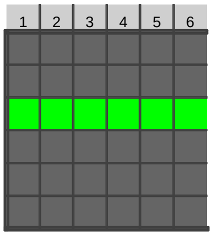
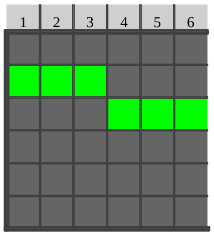
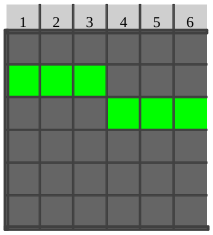
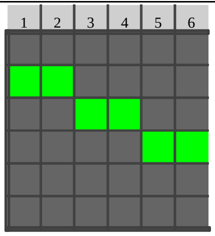
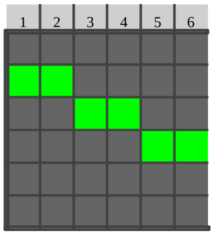
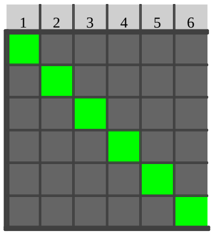
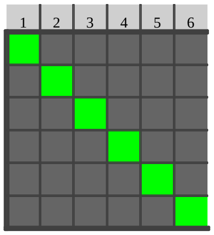
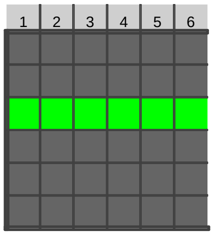
This section describes the informal OP Pilots necessary to determine the appropriate TBL and compensation thresholds as well as the canonical experiment. It also describes how a task-switching game in played.

4.1.5.1 A task-switching TruthTable game

The task-switching game with only one target cell per input is played exactly like the game described for Experiment 1, described above. The CLA trains in the initial phase on a 6-by-6 substage of the game until its confident accuracy is > 99.99% or it has trained for 100,000 contests, whichever happens first. Then, the game enters the secondary phase and extends out to the full 6-by-300 stage. The CLA continues to train until its confident accuracy is again >99.99% or until it has trained for 100,000 contests. In Experiments 1 through 4, CLAs only trained through the secondary phase, but in the remaining experiments, the CLAs trains for another, **tertiary phase**.

During the tertiary phase, the game changes partially as some of the target cells are reassigned to different outputs. **Table 12** below shows the changes that occur in the tertiary phase. The Standard-CLA is only affected by the number of inputs which are reassigned to new target cells, but the TBL-CLA is also affected by the change in TBL_{α}.

Table 12: Tertiary Phases of the TruthTable game

Stage	Secondary Phase	→	Tertiary Phase
1	 secondary TBL$_{\alpha}$ = 89,700	→	 tertiary TBL$_{\alpha}$ = 44,700 Δ 50% target responses Δ -45,000 TBL$_{\alpha}$
2	 secondary TBL$_{\alpha}$ = 44,700	→	 tertiary TBL$_{\alpha}$ = 29,700 Δ 50% target responses Δ -15,000 TBL$_{\alpha}$
3	 tertiary TBL$_{\alpha}$ = 29,700	→	 tertiary TBL$_{\alpha}$ = 14,700 Δ $66.\underline{6}\%$ target responses Δ -15,000 TBL$_{\alpha}$
6	 tertiary TBL$_{\alpha}$ = 14,700	→	 tertiary TBL$_{\alpha}$ = 89,700 Δ $83.\underline{3}\%$ target responses Δ +75,000 TBL$_{\alpha}$

4.1.5.2 OP Pilot 7: TBL thresholds, Task-Switching Game, One Target Cell per Input

In order to determine the appropriate factor range, the following OP pilot was conducted in order to accomplish Goal 2.3.1 (see Section 3.5). The collection length, *c*, was fixed at 12; the compensation threshold, κ_γ, was fixed at 60; and the TruthTable game was set to state 6 (*i.e.* there were six target responses in the game, for more detail see Section 3.4.5). TBL-CLAs were then trained with the TBL thresholds listed below in a Monte Carlo trial. The performance measures were calculated with Monte Carol trial results of a Standard-CLA under the same game conditions.

Table 13: OP Pilot Experiment Design for TBL thresholds

	TBL threshold 4-tuples	**number of treatments**
OP Pilot factors	<(κ_s, κ_w, κ_i, κ_d)> selected from the following set of values, subject to the specified constraints: {50.00, 60.00, 70.00, 80.00, 90.00, 95.00, 98.00, 99.99}	750
Results	<(κ_s, κ_w, κ_i, κ_d)> selected from the following set of values, subject to the specified constraints: {50.00, 70.00, 90.00, 98.00, 99.99}	66

Note that the number of treatments is not simply the factorial of the number of factors for the TBL threshold. The TBL thresholds are subject to the following constraints:

- The withdrawal threshold must be less than or equal to the support threshold.

  $\kappa_w \leq \kappa_s$

- The independence threshold must be less than or equal to the support threshold.

  $\kappa_i \leq \kappa_s$

- The dependence threshold must be less than or equal to the independence threshold.

$$\kappa_d \leq \kappa_i$$

4.1.5.3 OP Pilot 8: Compensation Threshold, κ_γ, Task-Switching Game, One Target Cell per Input

Once the TBL threshold factor range was fixed, the following OP pilot was conducted to accomplish Goal 2.3.2 (see Section 3.5). One highly successful factor combination, one neutral combination, and one unsuccessful combination were selected. Each TBL threshold factor combination was applied to a TBL-CLA. Each TBL-CLA then completes Monte Carlo trials with different values for κ_γ. The OP pilot factors and the results are shown in Table 14. The results of this pilot were interesting because the TBL-CLA was much more sensitive to the settings for κ_γ than the other pilots.

Table 14: OP Pilot Experiment Design for κ_γ

OP pilot factor values for κ_γ	**fixed value for κ_γ**
{50, 55, 60, 65, 70, 75, 80, 85, 90, 94, 95, 95.5, 96, 96.5, 97 98, 99}	**96**

4.1.5.4 Experiment 5: Canonical, Task-switching Game, One Target Cell per Input

After determining the appropriate factor values for the TBL threshold and the compensation threshold, the following experiment is conducted. The TruthTable game environment is *not* stationary throughout the learning process, that is, the arrangement of target cells changes after the secondary phase of the game, and there is only one target cell per input. Below in Table 15 is an experiment block design with all of the factors.

Both Standard and TBL-CLAs are used. The results from the Standard-CLA are used as a baseline for computing the performance measures.

Table 15: Design for Experiment 5 (canonical, task-switching game, 1 target cell per input)

Name	**Factor values**	**Treatments**
Target responses	{1, 2, 3, 6 }	4
Collection length, *c*	{1, 2, 4, 6, 12 }	5
TBL thresholds, κ*	<($\kappa_{s,}$ $\kappa_{w,}\kappa_{i,}$ κ_d)> selected from the following set of values, subject to the specified constraints: <50.00, 70.00, 90.00, 98.00, 99.99>	66
Experiment resource requirements		
	Total treatments	**1340***
	Estimated time per treatment	8 CPU minutes
	Estimated total CPU time required	7.5 CPU days
	Total CPUs available	4
	Estimated time required	**2 days**
	* total number of treatments calculated as follows: Standard-CLA treatments = 4(5) TBL-CLA treatments = 4(5)(66) Total = Standard-CLA treatments + TBL-CLA treatments = 1340	

4.1.5.5 OP Pilot 9: Select factor combinations for close inspection, Task-switching Game, One Target Cell per Input

Experiment 5, described in the previous section, generates sufficient data to accomplish Goal 3.3, described in Section 3.5. In this experiment, the behavior of a few specific TBL-CLAs is examined more closely.

Before Goal 3.3 can be accomplished, treatments must be selected for closer examination (Goal 3.3.1). An informal OP pilot was conducted in which the results of

Experiment 5 were analyzed to determine which treatments presented an average case, strong case, and a weak case. It was determined that the variances in behavior that occurred with the changes in the number of target responses and collection length were worth investigating at all settings. The following TBL threshold combinations are considered:

- Strong (90.00, 90.00, 90.00, 70.00)
- Average (99.99, 99.99, 70.00, 70.00)
- Weak (99.99, 50.00, 50.00, 50.00)

4.1.5.6 Experiment 6: Close Inspection, Task-switching Game, One Target Cell per Input

Experiment 6 accomplishes Goal 3.3, described in Section 3.5. The calculation of the individual performance measures is completed with the use of a spreadsheet, but the process of inspecting the results and drawing conclusions about each case requires significant attention from a person. Table 16 below presents the block design for Experiment 6.

Table 16: Design for Experiment 6

Name	Factor values	Treatments
TruthTable game states	{1, 2, 3, 6 }	4
Collection length, *c*	{1, 2, 4, 6, 12}	5
TBL threshold 4-tuples <($\kappa_{s}, \kappa_{w}, \kappa_{i}, \kappa_{d}$)>	(90.00, 90.00, 90.00, 70.00) (99.99, 99.99, 70.00, 70.00) (99.99, 50.00, 50.00. 50.00)	3
Experiment Resource Requirements		
	Total treatments	**80***
	Estimated time per treatment	10 person-minutes
	Total time required	**14 person hours**
	* total number of treatments calculated as follows: Standard-CLA treatments = 4(5) TBL-CLA treatments = 4(5)(3) Total = Standard-CLA treatments + TBL-CLA treatments = 80	

4.2 Experimentation & Reduced Results

The solution for this research is entirely software-based. All code was written and compiled in Java. Full code listings are available in the Digital Appendix. The experiments were run on an Apple MacPro Quad-Core computer. The results were analyzed using Microsoft Excel 2004 and Microsoft Excel 2007. The reduced results are available in the digital appendices as Microsoft Excel files.

CHAPTER 5: RESULTS AND CONCLUSIONS

5.1 Introduction

This chapter presents the results and conclusions of this research along with informal observations and suggestions for future work. The results and formal conclusions of each experiment are presented together in Sections 5.2 through 5.5.5. The informal observations about all of the experiments are presented in Section 5.8. Finally, suggestions for future work are presented and discussed in Section 5.9.

5.2 Experiment 1

Experiment 1 is a canonical experiment. The TruthTable game environment is stationary throughout the learning process, that is, the arrangement of target cells does not change during training, and there is only one target cell per input. Below in Table 6 taken from Section 4.1.3.3, is the experiment block design with all of the factors. Both Standard and TBL-CLAs are used. The results from the Standard-CLA are used as a baseline for computing the performance measures.

Table 6: Design of Experiment 1 (stationary game, 1 target cell per input)

Factor Name	Values	Treatments
TruthTable game state	{1, 2, 3, 6}	4
Collection length, *c*	{1, 2, 4, 6, 12}	5
TBL thresholds, κ^*	$<(\kappa_{s}, \kappa_{w}, \kappa_{i}, \kappa_{d})>$ selected from the following set of values, subject to the specified constraints: <50.00, 55.00, 70.00, 80.00, 95.00, 99.99>	266
Experiment resource requirements		
	Total treatments	**5340***
	Estimated time per treatment	2 CPU minutes
	Estimated total CPU time required	7.5 CPU days
	Total CPUs available	4
	Estimated time required	**2 days**
	total number of treatments calculated as follows: Standard-CLA treatments = 4(5) TBL-CLA treatments = 4(5)(266) Total = Standard-CLA treatments + TBL-CLA treatments = 5340	

5.2.1 Results

The results are presented in footprints in this section and have been sized to fit a single page. In many cases, this limits the legibility of the data labels. The footprints are intended to give an overview of trends in behavior and performance as the factors are varied. Each performance measure has a unique dynamic range, but they are all presented in the same color range (red to green). Because the footprints are used to visualize trends, the exact values are not as important. Bright green is always used for results that favor the TBL-CLA and bright red for those that favor the Standard-CLA. The footprints are also presented at a legible resolution with individual color scale information over several pages in the following appendices: APPENDIX A: EXPERIMENT 1, PAYOFF RESULTS; APPENDIX B: EXPERIMENT 1, EXPENSE RESULTS; and

APPENDIX C: EXPERIMENT 1 *N*-TILE ADVANTAGE RESULTS.

Table 17 shows the results of Experiment 1 sorted by the TBL thresholds. Section 3.6.4 describes the layout of the footprint in greater detail, but briefly, the columns and rows are organized in a hierarchical fashion.

The columns are divided first by the performance metric: Payoff, then Expense, then *n*-tile advantage. Within each performance metric, the columns are subdivided by collection length. Finally, within each collection length, the columns are again subdivided by the TruthTable game state, which corresponds to the number of target outputs in each game state.

The rows are organized hierarchically by TBL thresholds. The thresholds are presented in the following order, from left to right:

Support threshold, κ_s

Withdrawal threshold, κ_w

Independence threshold, κ_i

Dependence threshold, κ_d

The factor values for the TBL thresholds are presented from smallest to largest, according to the following rules:

The withdrawal threshold must be less than or equal to the support threshold.

$\kappa_w \leq \kappa_s$

The independence threshold must be less than or equal to the support threshold.

$\kappa_i \leq \kappa_s$

The dependence threshold must be less than or equal to the independence threshold.

$\kappa_d \leq \kappa_i$

Turning to the results in Table 17, it can be seen that the TBL thresholds do affect the performance of a TBL-CLA. The footprint can be divided into 2 major sections by looking at Payoff and *n*-tile advantage. In approximately the top third of the footprint, the Payoff is generally very high and the TBL-CLA has a clear advantage through the first 70% of the first termination. In general, the performance measures in the bottom two thirds of the footprint are very similar.

Another observation that can be drawn from the footprint in Table 17 is that the TBL-CLA performs exceptionally well in all cases for a collection length of one. A collection length of one is a trivial case, but it is included for completeness. At a collection length of one, the optimal strategy is a simple process of elimination because the CLA receives an evaluation on it responses individually rather than collectively. A second, less trivial observation is that the TBL-CLA does consistently well in all TruthTable games in state 1, where there is only one target response (see Section 3.4.5 for more details). This means that the TBL-CLA's performance is correlated with the TBL_{α} of the TruthTable game state. That is, the greater the TBL advantage, the better the TBL-CLA's performance.

In order to draw more specific observations about the performance of a TBL-CLA, it is necessary to reorder the footprint and examine smaller subsections of it. In order to identify those TBL threshold 4-tuples that are most effective, the results are sorted by the minimum Payoff value in each row. The resorted footprint is presented in Table 18. The TBL 4-tuples with a minimum Payoff greater than zero are shown in the blue square. There is not much new information to be gained from this view of the results. A closer view of the data is needed. Table 19 presents the top results from Table 18.

Table 17: Results of Experiment 1 sorted by TBL thresholds. The varying collection lengths and the TruthTable game state are the column headers. The TBL thresholds are the row headers. The thresholds are sorted from smallest to largest factors values in the following order, according to the rules for TBL thresholds: support threshold, withdrawal threshold, independence threshold, and dependence threshold.

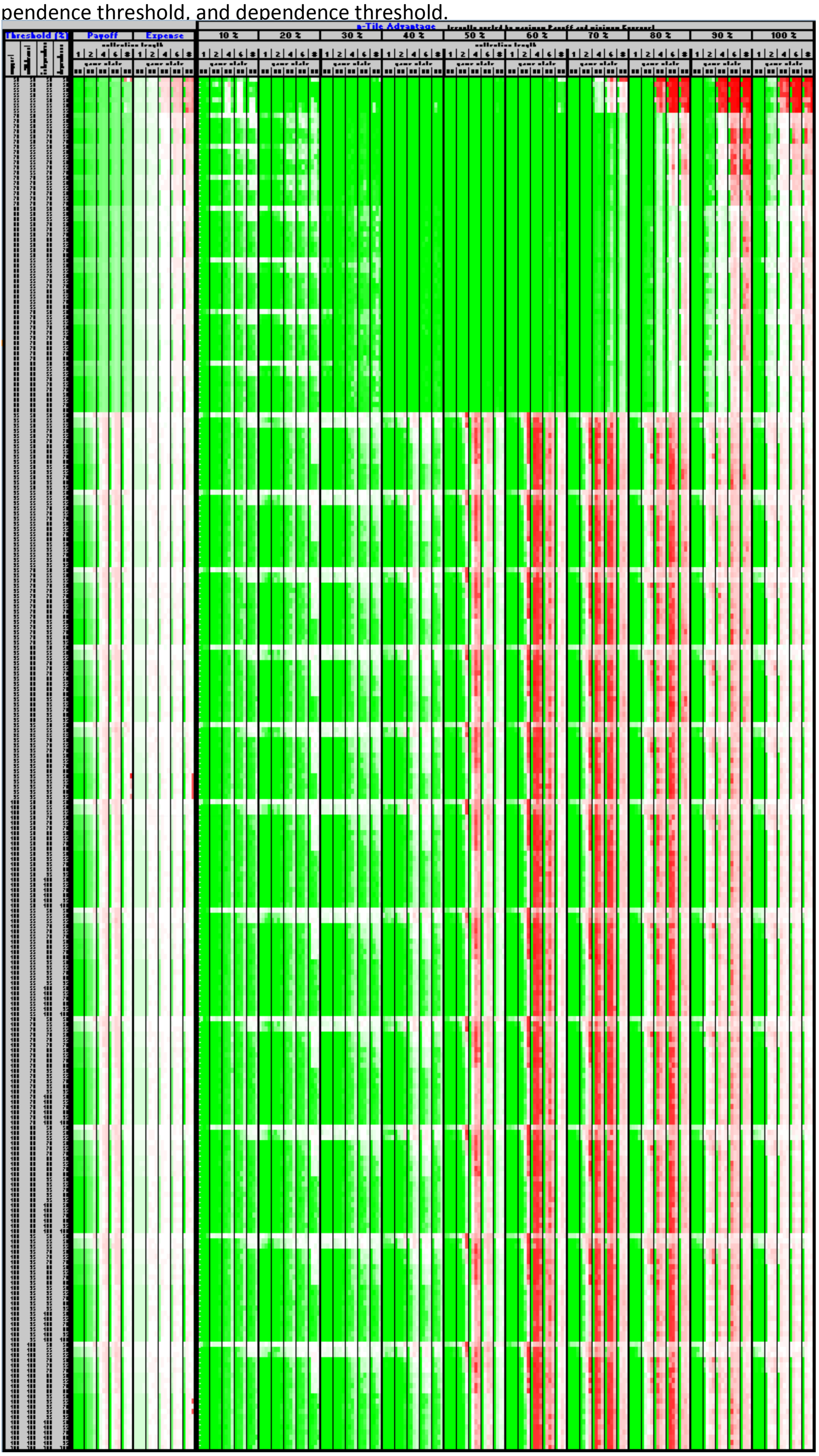

n-Tile Advantage
Threshold (%)
Payoff
Expense
10 %
20 %
30 %
40 %
50 %
60 %
70 %
80 %
90 %
100 %
collection length
1
2
4
6
8
game state

Table 18: Results of Experiment 1 sorted by minimum Payoff. The results have been reordered by the minimum Payoff value in each row. The results in the blue square are those results with a minimum Payoff value greater than

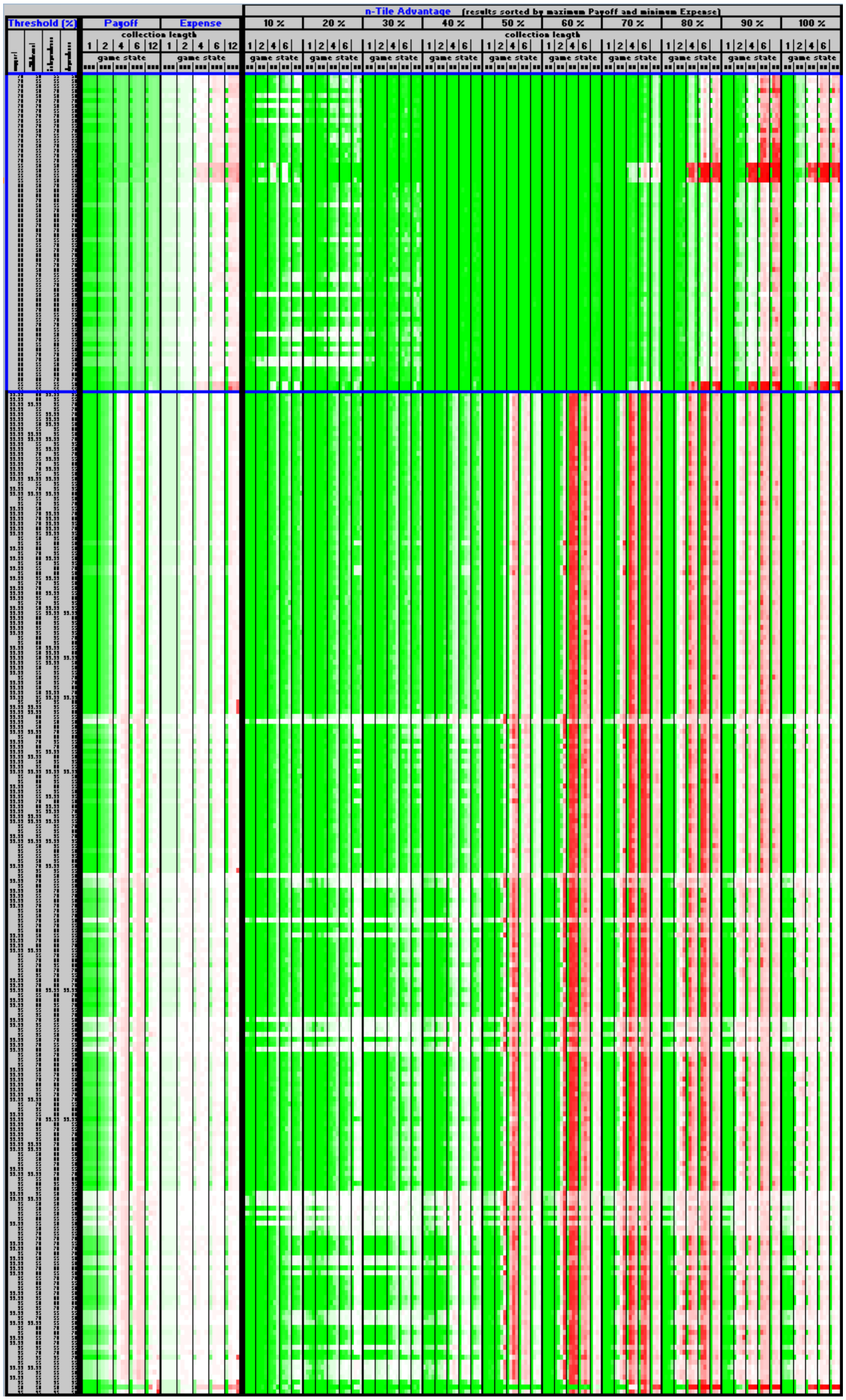
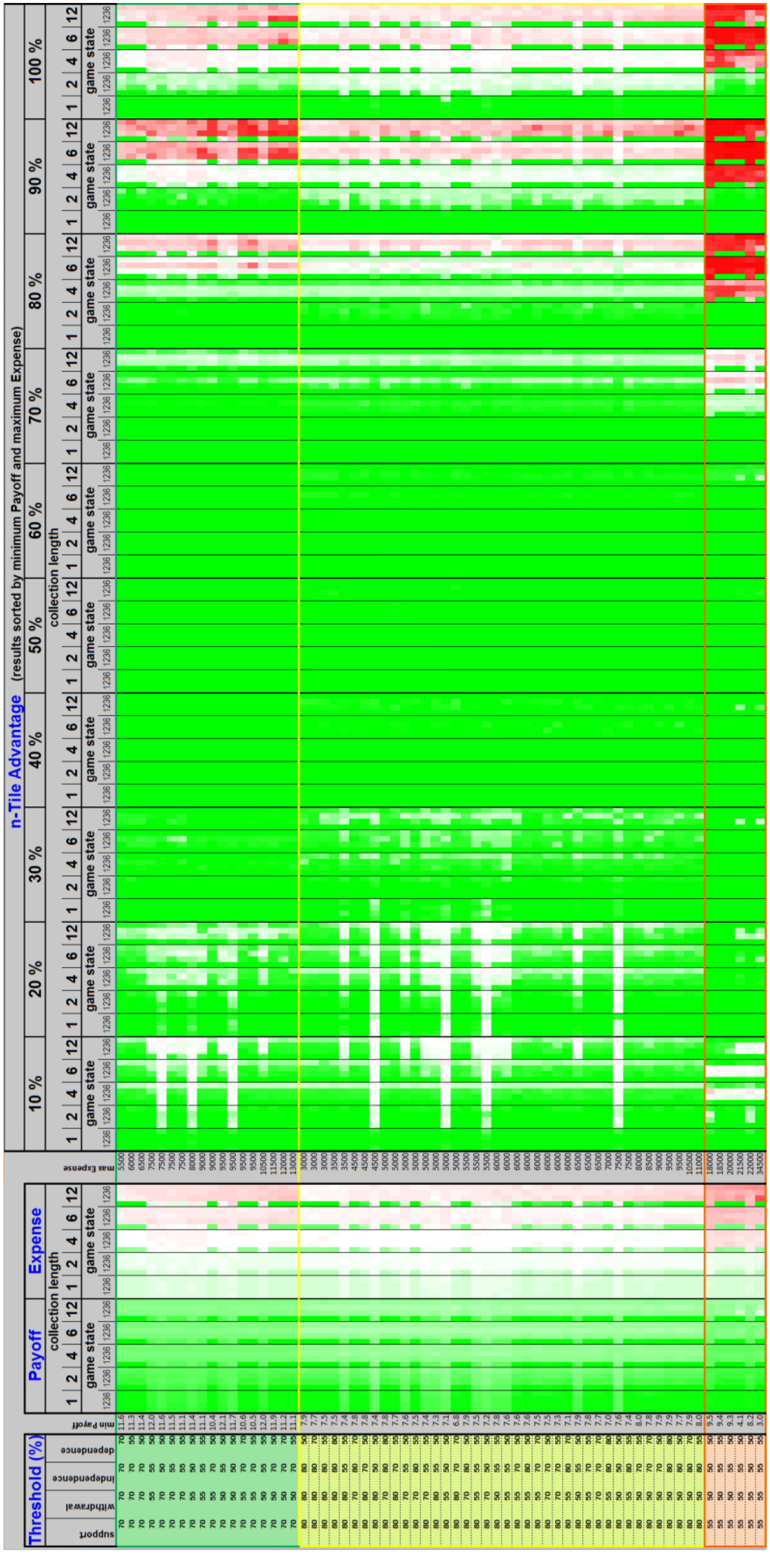

n-Tile Advantage (results sorted by maximum Payoff and minimum Expense)
Threshold (%)
Payoff
Expense
collection length
1 | 2 | 4 | 6 | 12
1 | 2 | 4 | 6 | 12
game state
10 %
20 %
30 %
40 %
50 %
60 %
70 %
80 %
90 %
100 %
collection length
1 | 2 | 4 | 6
game state

Table 19: Good Results from Experiment 1 sorted by Payoff and Expense. The good results were first sorted by minimum Payoff. They were then separated into two groups at a minimum Payoff of 10.0 and resorted by maximum Expense. The results then fell into groups based on the value of the support threshold.



The results presented in Table 19 are first sorted by minimum Payoff. There is a clear break in the distribution of minimum Payoff measures at 10.0, so the data is spilt into two groups. Each of the groups is internally sorted by the maximum Expense. The sorted data falls into three distinct groups, each governed by the support threshold. The best overall performing TBL 4-tuples were those with a support threshold of 70%. The Expense is lowest in the 4-tuple (70, 70, 70, 70) and increased as the withdrawal threshold is lowered. When the support threshold is set to 80%, both the minimum Payoff and the maximum Expense drop slightly. This means that there is the option of trading some Payoff for a lower Expense, if that is a concern. The last grouping is the 4-tuples with the support threshold of 55%. These 4-tuples have higher minimum Payoff values, but also higher Expense values and lower *n*-tile advantages in the latter *n*-tiles.

For best performance, the support, withdrawal, and independence thresholds should be set equal to each other and at lower values (between 70% and 80%), but not too low (less than 70%). Setting the support threshold low, but not too low, allows tactics to be supported early on and setting the withdrawal threshold low allows tactics to stay on the global list. A potential drawback to low support thresholds is that it forces stimulants into the independent role before they have gained the full benefit of following a tactic. This is why it is best to set the independence threshold as high as possible. However, having a low independence threshold also helps the TBL-CLA avoid harsh and confusing updates from the compensation policy.

Recall that the compensation policy relies on the CLA’s anticipated evaluation which is calculated based on the number of confident stimulants in the history (stimulants whose tie and reject confidences are sufficiently high to all them to consistently select their primary respondent) and the number of follower stimulants. It is possible for a follower to receive a very positive update one time and a very negative one the next, especially at longer collection lengths. This phenomenon adds noise to the signal. It is important to balance the benefits of receiving “unearned” positive updates with the overly harsh updates that come at the cost of occasionally inconsistent and harsh negative ones.

The remaining data not presented in Table 19 was sorted by Expense. The 4-tuples with higher support and withdrawal threshold values and very low independence and dependence values produced results that indicate that the TBL-CLA’s performance is not distinguishable from the Standard-CLA’s performance. This is a reasonable outcome because it follows that if stimulants are not allowed to be followers for very long, if at all, they will spend the bulk of their life cycles following the Standard selection policy. Table 20 shows the neutral results.

The remaining results that have not been discussed in detail are those 4-tuples which perform just slightly worse than the Standard-CLA. A small selection of these results is presented in Table 21. Note that even these cases show early gains over the Standard-CLA, though they fall behind in the later *n*-tiles.

The footprint results are only useful for analyzing the trends in the data. For a more detailed study, but with a limited cover, Experiment 2 examines the learning behavior of three 4-tuples in detail.

Table 20: Neutral Results for Experiment 1 sorted by Expense. By setting the independence and dependence thresholds very low and keeping the support and withdrawal thresholds high, the TBL-CLA performs like a Standard-CLA.

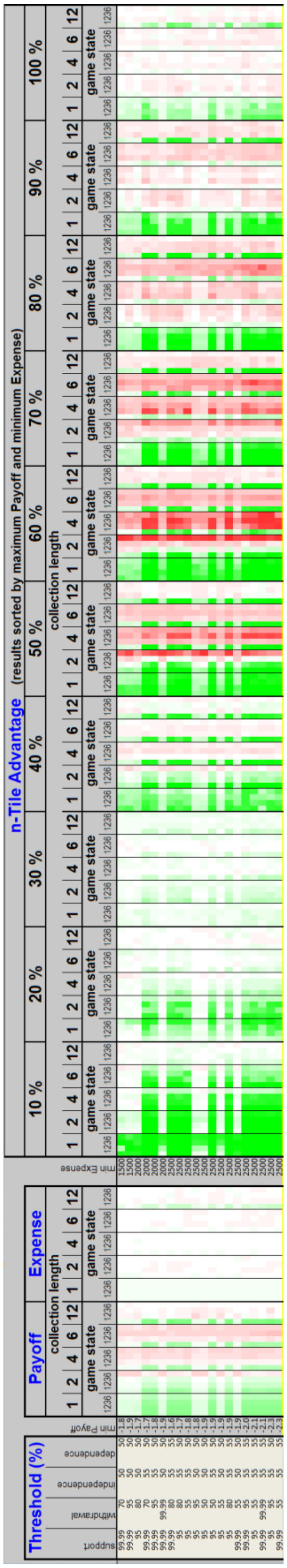
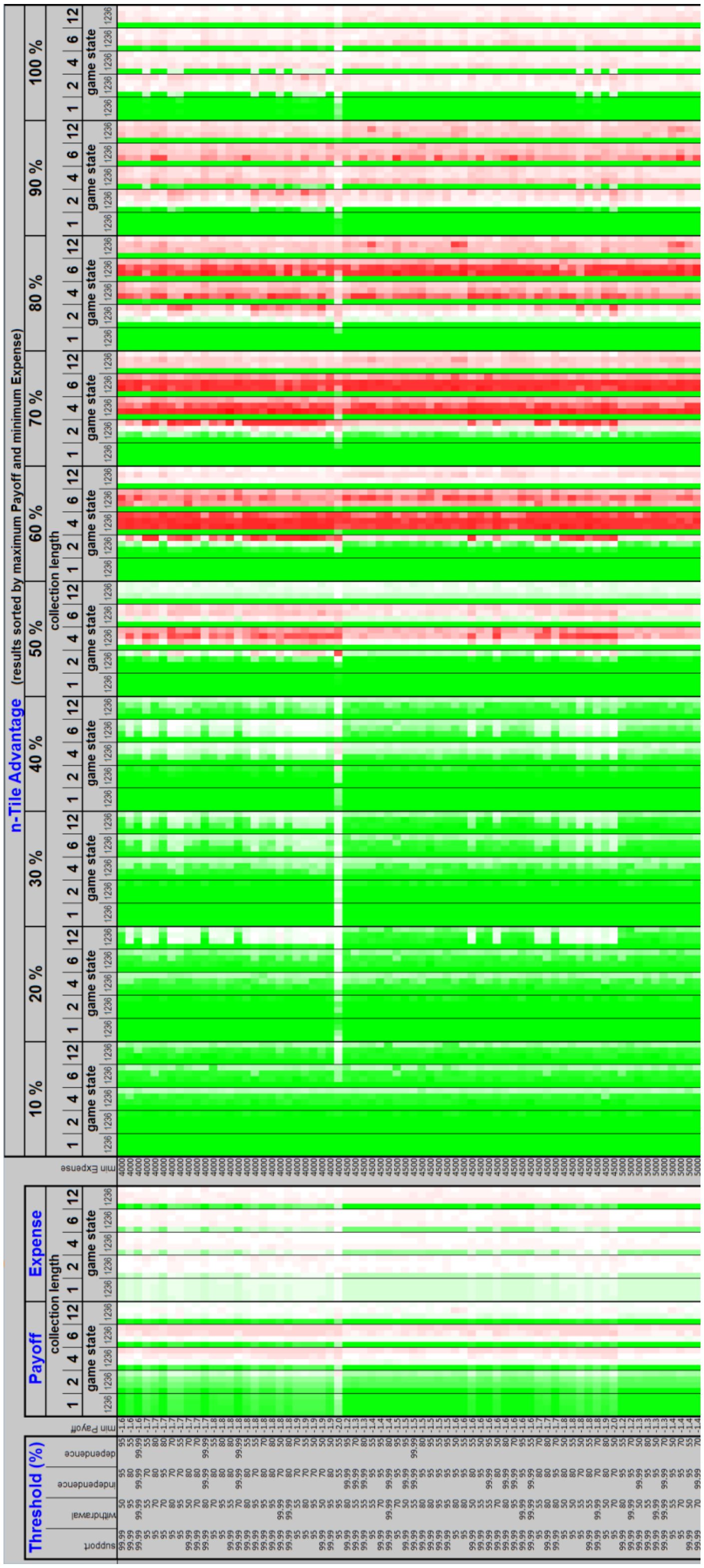


Table 21: Underperforming Results for Experiment 1 sorted by Expense. Most of the TBL 4-tuples produce slightly underperforming TBL-CLAs. This table presents only a subset of the full results.



5.2.2 Formal Conclusions from Experiment 1

A summary of the conclusions drawn from Experiment 1 is presented in this section. For a more detailed discussion of these conclusions, see the previous section. All conclusions presented in this section are only valid for the TruthTable game in an environment with *one target per output* and that is *stationary, deterministic, and correct* for the duration of the learning process. Informal conclusions, speculations, and predictions about the performance of the application of TBL to other games, including actual-life games, will be presented and discussed in Section 5.8. Suggestions for future research are presented and discussed in Section 5.9.

Conclusion 1: The learning behavior of a TBL-CLA is significantly affected by the settings of the TBL thresholds.

Conclusion 1a: For best performance, set the support, withdrawal, and independence thresholds equal to each other and at values that are between 70 and 80%.

Conclusion 1b: A TBL-CLA will behave like a Standard-CLA when the support and withdrawal thresholds are set high, greater than 90%, and the independence and dependence thresholds are set at or near the minimum (less than or equal to 55%).

Conclusion 1c: Even in 4-tuples that do not produce optimal or Standard-like behavior, using TBL provides significant advantages early in the learning process; however, the advantages are lost later on as the TBL-CLA significantly underperforms compared to the Standard-CLA.

Conclusion 2: TBL-CLA performance improves as the TBL advantage, TBL$_{\alpha}$, of the TruthTable game state is increased.

5.3 Experiment 2

Experiment 2 accomplishes Goal 3.1, described in Section 3.5, a close inspection of the behavior of a small selection of 4-tuples from Experiment 1. Table 7 below, taken from Section 4.1.3.5, presents the block design for Experiment 2.

Table 7: Design for Experiment 2 (close inspection, 1 target cell per input) [from Section 4.1.3.5]

Name	Factor values	Treatments
Target responses	{1, 2, 3, 6 }	4
Collection length, *c*	{1, 2, 4, 6, 12 }	5
TBL threshold 4-tuples <($\kappa_{s,}$ $\kappa_{w,}$$\kappa_{i,}$ κ_{d})>	(70.00, 70.00, 70.00, 70.00) (99.99, 95.00, 50.00, 50.00) (95.00, 95.00, 95.00, 50.00)	3
Experiment resource requirements		
Total treatments		**80***
Estimated time per treatment		10 person-minutes
Estimated total time required		**14 hours**
* total number of treatments calculated as follows: Standard-CLA treatments = 4(5) TBL-CLA treatments = 4(5)(3) Total = Standard-CLA treatments + TBL-CLA treatments = 80		

5.3.1 Results

This section presents selected results from Experiment 2. The full set of reduced results is available in the digital appendices as interactive Microsoft Excel spreadsheets.

5.3.2 Best Performance: 4-tuple (κ_s=70, κ_w=70, κ_i=70, κ_d=70)

It was observed in Experiment 1 that the 4-tuple of (70, 70, 70, 70) was the best performing combination of TBL thresholds. The results of close inspection of the behavior of a TBL-CLA with these settings are presented in this section. As the TBL thresholds are fixed, the purpose of this experiment is to understand the influence that the thresholds have on the behavior of the CLA under varying environmental conditions: the collection length and the TruthTable game state.

5.3.2.1 TruthTable state 1, Collection Length 6

In Experiment 1, the TBL-CLA generally dominated the Standard-CLA in TruthTable game state 1 (see Section 3.4.5 for a description of the game state). A closer inspection of the CLAs performance on TruthTable game state 1 with a collection length of 6 is presented in Figure 16, Figure 17, and Figure 18 below.

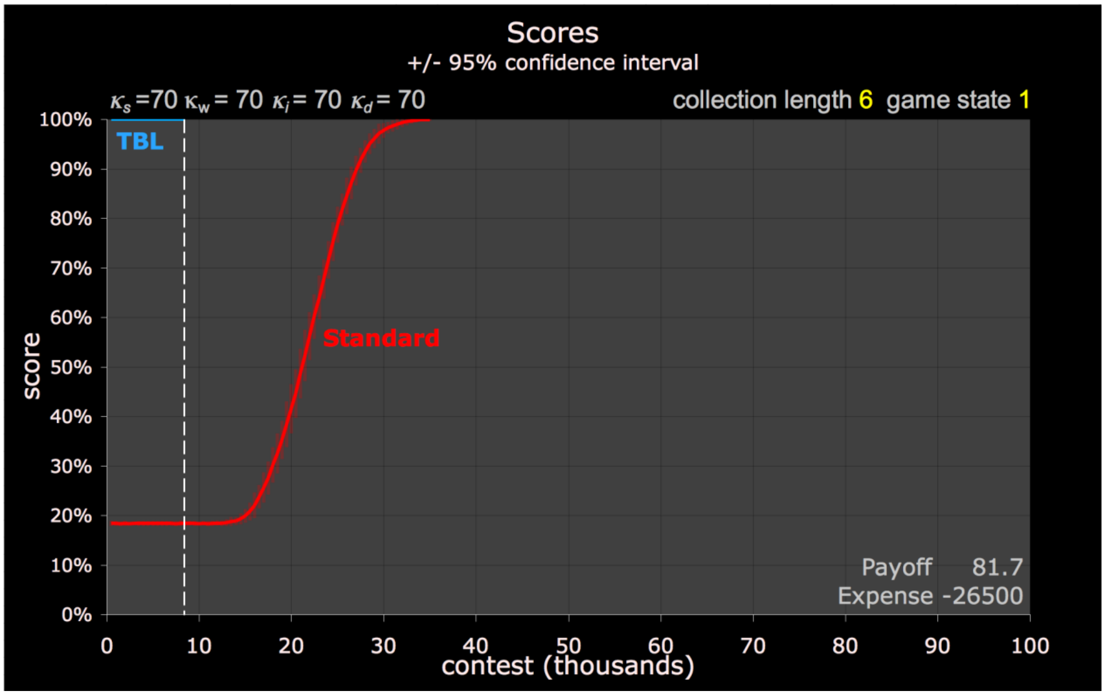


Figure 16: The TBL-CLA dominates the Standard-CLA on TruthTable game state 1. The dashed line marks the first termination. The Payoff and Expense measures for this treatment are given in the lower right hand corner.

The TBL-CLA dramatically outperforms the Standard-CLA on the TruthTable game state 1, with a Payoff of 81.7 and an Expense of -26,500. Figure 16 demonstrates how the TBL-CLA is able to quickly identify effective tactics and take advantage of those tactics. Of course, if there is only one target response in the game, then there only need be one tactic. Figure 18 shows that the TBL-CLA makes significantly fewer random selections during its learning process. There is a large spike in the number of random selections for the TBL-CLA, but this can be explained by recalling that the support threshold is set at 70. This means that stimulants are able to support a tactic even *before* they are consistently selecting that respondent themselves. In order to be considered a non-random selection, a stimulant must be a confident stimulant or a follower. Figure 17 shows that several stimulants are indeed supporters before they are confident stimulants, but that this spike in the number of random selections does not hinder the learning process.

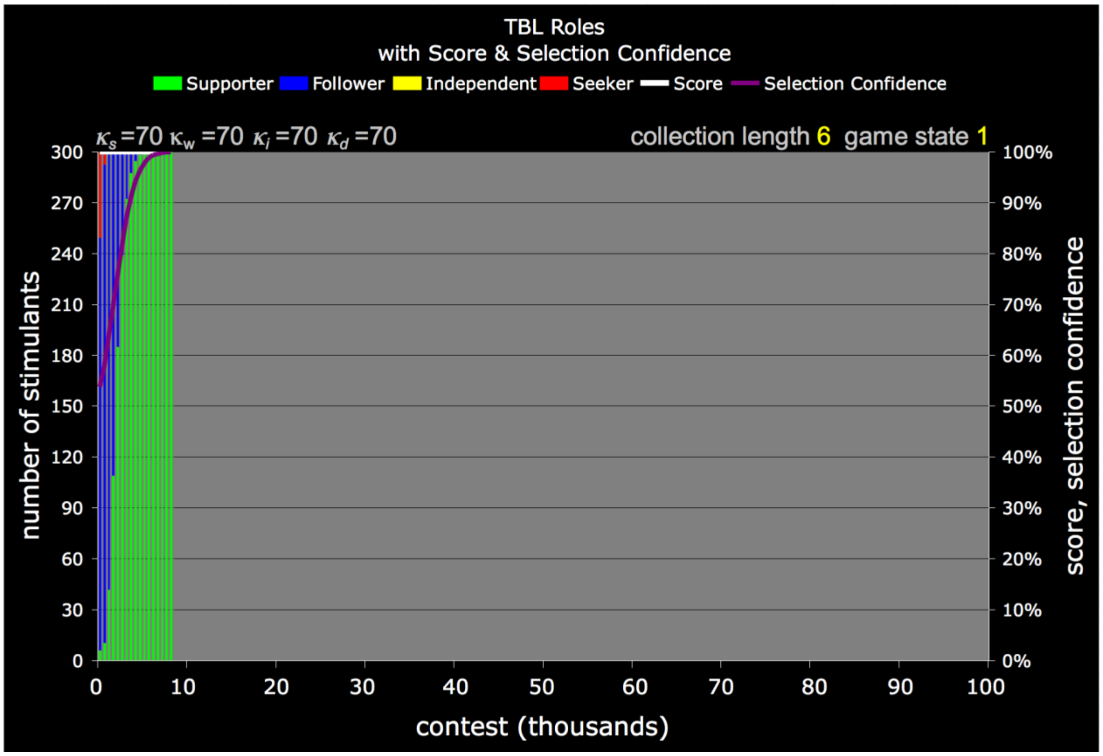


Figure 17: The TBL-CLA makes heavy use of tactics to quickly reach termination.

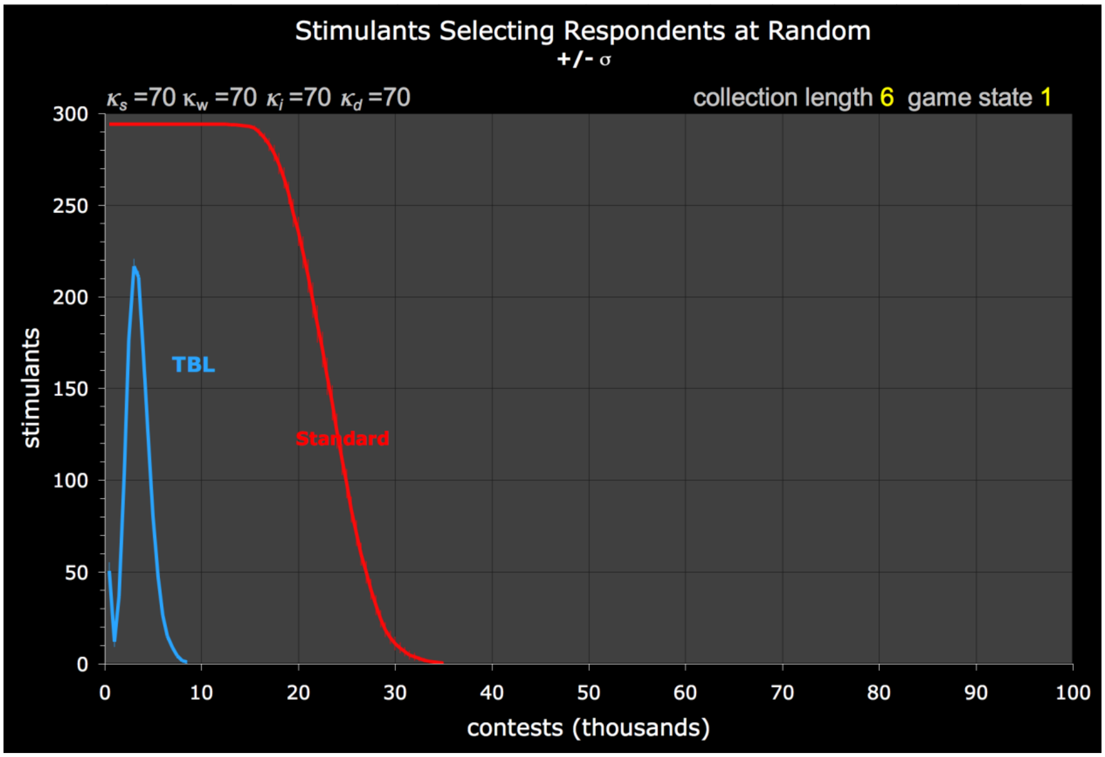


Figure 18: The TBL-CLA makes significantly fewer random selections than the Standard-CLA.

5.3.2.2 TruthTable state 2, Collection Length 6

When the number of target responses in a TruthTable game state increases, the Tactic-Based Learning advantage, TBL$_\alpha$, decreases (see Section 3.4.5). Figure 19 shows that as the TBL$_\alpha$ decreases, so does the performance of the TBL-CLA. When the TruthTable game state includes two target responses, the TBL-CLA still outperforms the Standard-CLA for most of the learning process, but the TBL-CLA gains, while still significant, are not nearly as large as they were in state 1.

Having more than one target response in the game makes it harder for the TBL-CLA to identify effective tactics. Recall that a local tactic's potency is the average of the update values that have been received while using the tactic. When the local potency drops below the minimum potency threshold, the local tactic is no longer considered effective. It is possible for a stimulant to incorrectly deem a local tactic ineffective and revert to the Standard selection policy. This means that overall, there are more seeker stimulants in the STM than there are when the TruthTable game state has only one target response (see Figure 20).

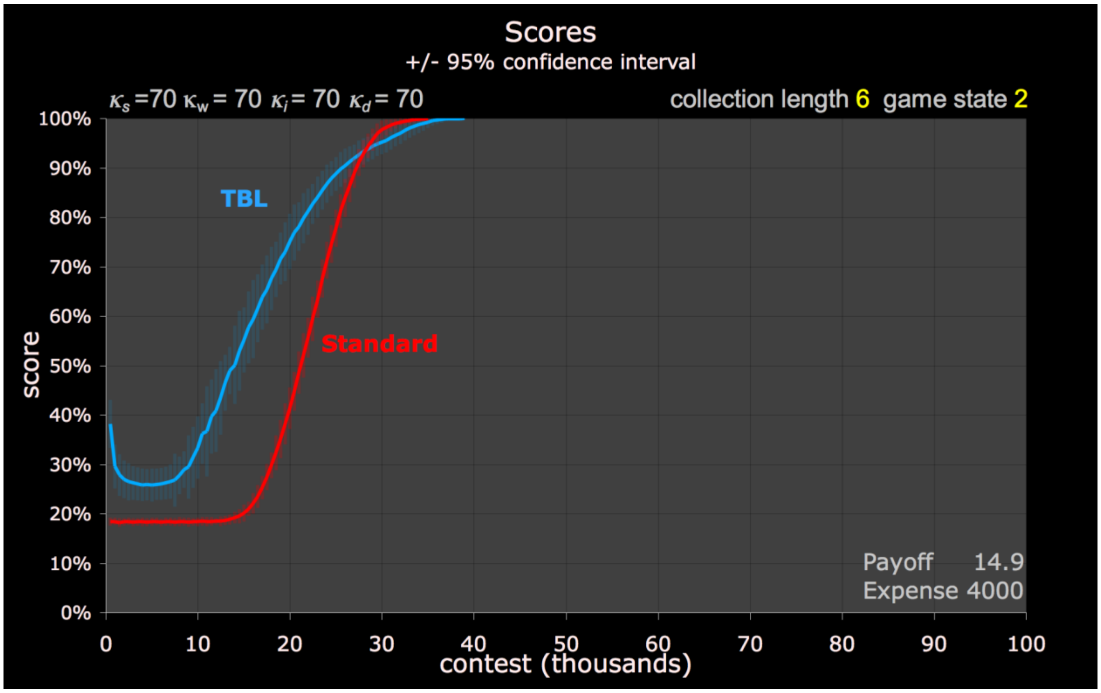


Figure 19: The TBL-CLA's score is significantly better than the Standard-CLA's score.

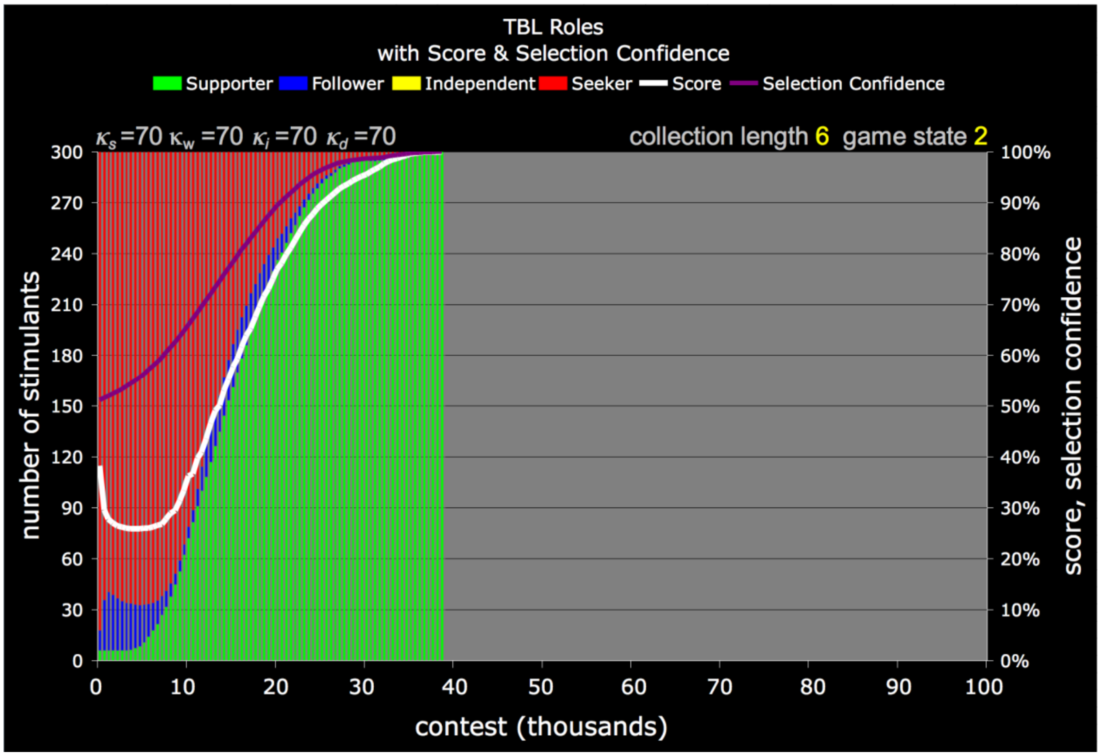


Figure 20: With two target responses and a lower TBL$_\alpha$, the TBL-CLA is not able to use tactics as long as it can for a game with only one target response.

5.3.2.3 TruthTable state 3, Collection Length 12

The conditions of TruthTable state 3 and collection length 12 were chosen because they demonstrate the way in which using TBL can be beneficial with a longer collection length. The longer the collection length means that there is a lower the signal-to-noise ratio, which slows the rate of learning for both CLAs. The TBL-CLA shows a clear advantage in the early contests (Figure 21). It ends in a tie with the Standard-CLA, but the early significant advantages are large enough for the TBL-CLA to earn a Payoff of 11.6 for this treatment and incur and Expense of only 5500 contests.

The improvement of the TBL-CLA's scores is caused by the fact that a fraction of the stimulants are able to remain followers throughout the learning process (shown in Figure 22). The TBL-CLA makes fewer random selections for most of the learning process (shown in Figure 23).

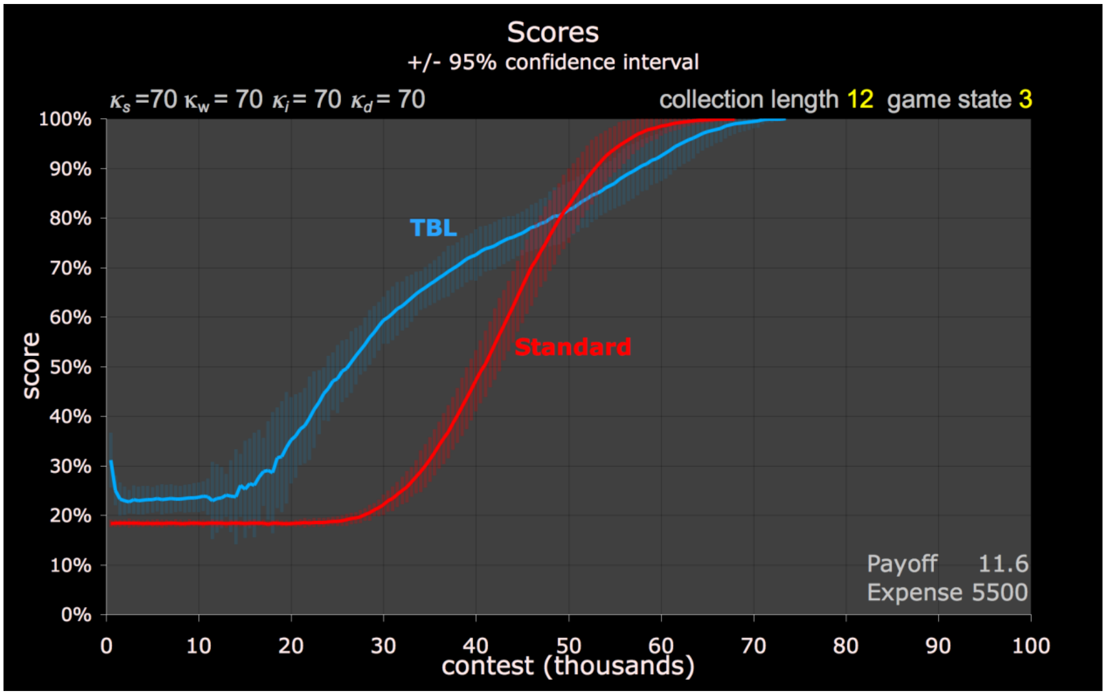


Figure 21: With a collection length of 12, the TBL-CLA earns a significantly higher score than the Standard-CLA for most of its learning process.

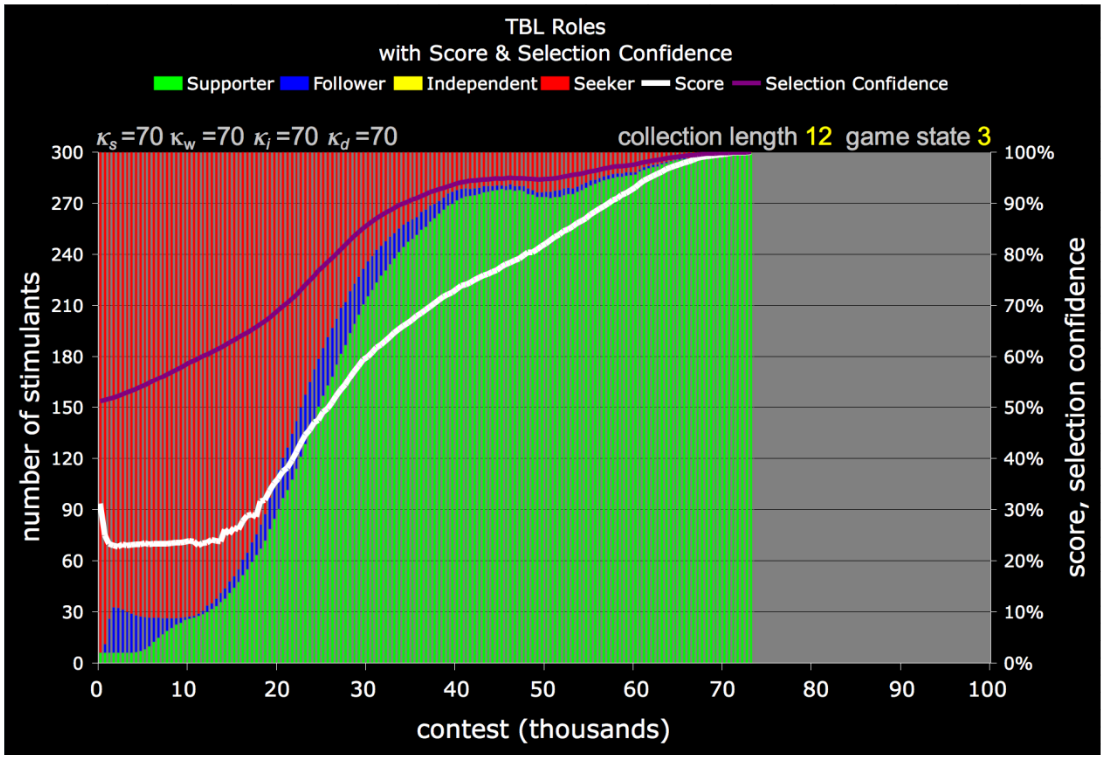


Figure 23: A fraction of the stimulants are able to remain followers throughout most of the learning process, which supports the TBL-CLA's score.

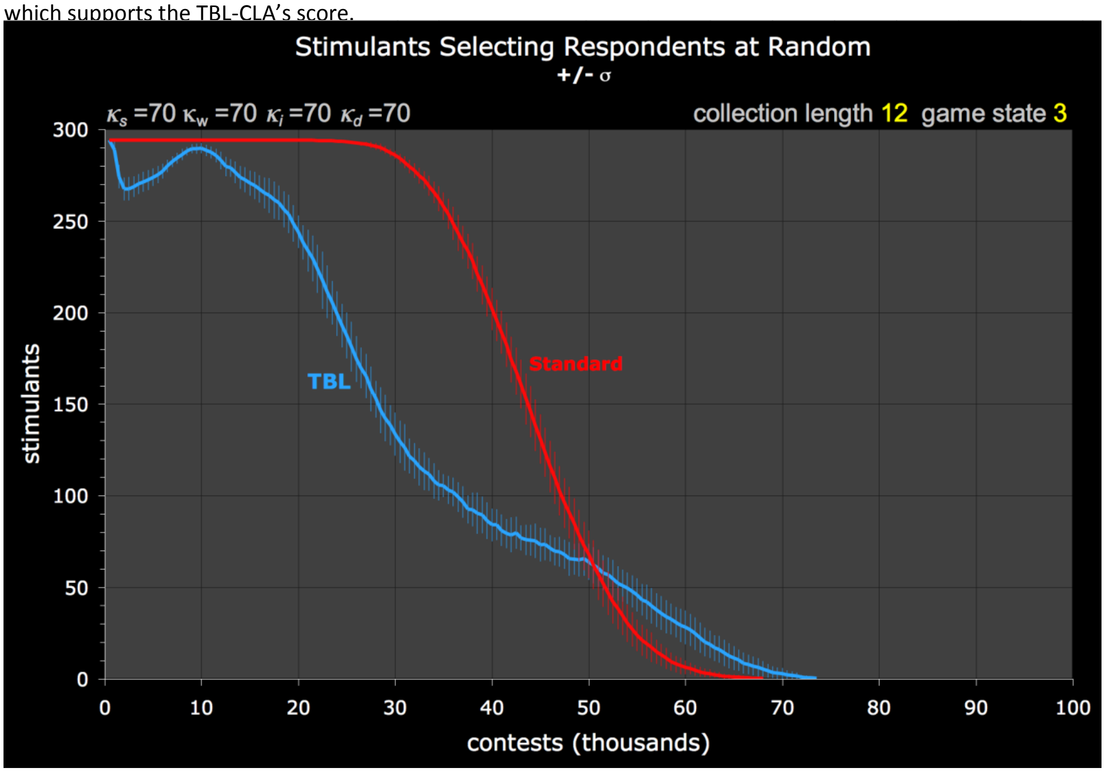


Figure 22: The TBL-CLA is able to maintain the use of tactics throughout the learning process and therefore the number of random selections made by the TBL-CLA is significantly lower that the Standard-CLA.

5.3.2.4 TruthTable state 6, Collection Length 4

This treatment is included to demonstrate that even a few follower stimulants early in the learning process can have a lasting and improving effect on learning behavior. Figure 24 shows the consistent and significant improvement in scores by the TBL-CLA. Figure 25 shows that only a small number of follower stimulants is ever in the STM and that those stimulants only appear early in the learning process. Finally, Figure 26 shows a corresponding reduction in the number of stimulants selecting their respondents at random.

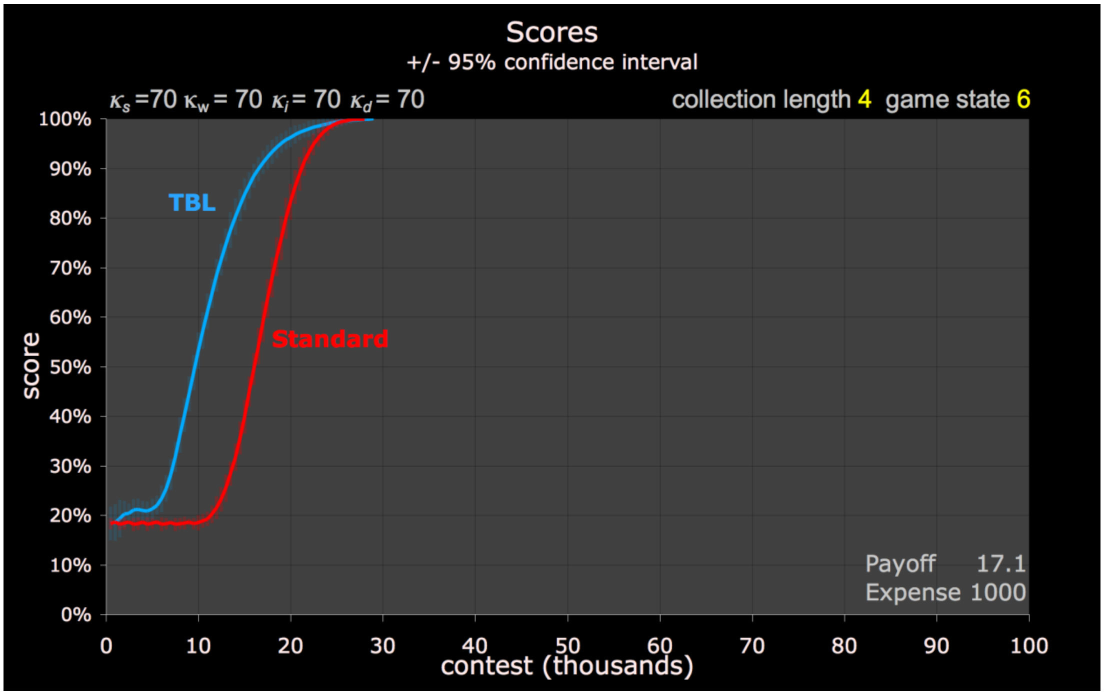


Figure 24: The TBL-CLA earns a consistently higher score than the Standard-CLA during learning.

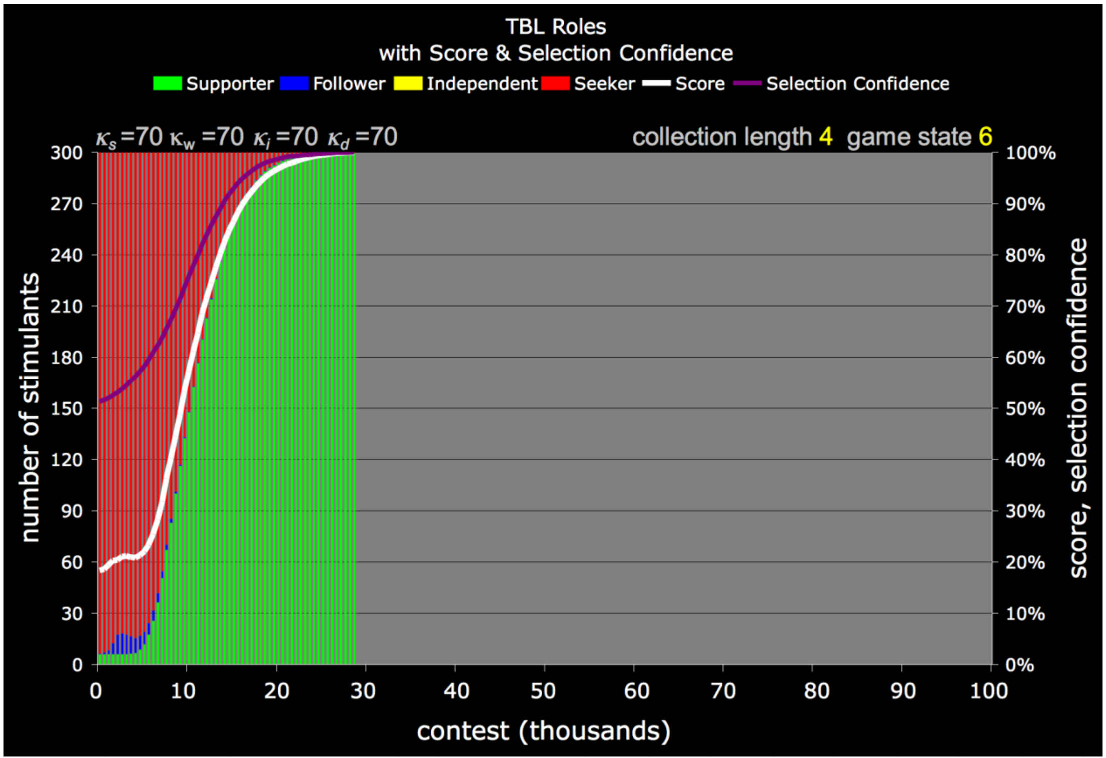


Figure 25: There are only a few follower stimulants and these only exist early on in the learning process, but, this is enough of a boost to the TBL-CLA that is score significantly higher than the Standard-CLA.

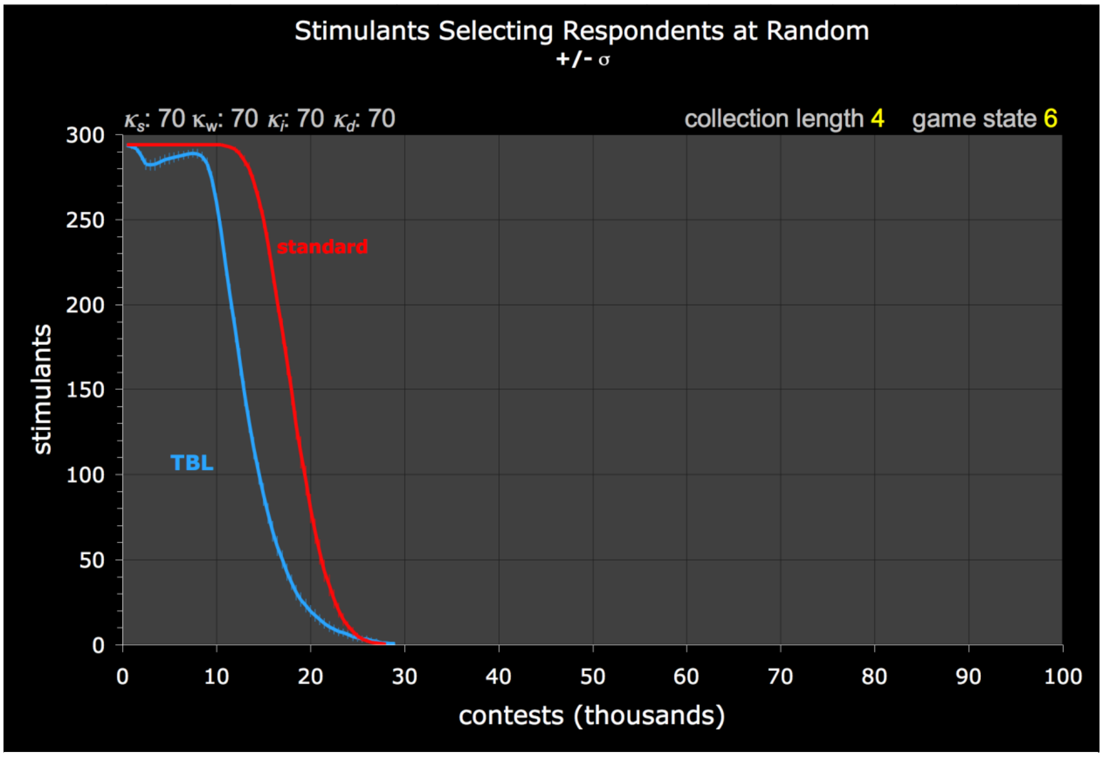


Figure 26: The early use of tactics significantly and consistently reduces the number of stimulants that select respondents at random.

5.3.3 Neutral Performance: 4-tuple (κ_s=99.99, κ_w=95, κ_i=50, κ_d=50)

It was observed in Experiment 1 that the 4-tuple of (99.99, 95, 50, 50) was a neutral performing combination of TBL thresholds. The results of close inspection of the behavior of a TBL-CLA with these settings are presented in this section. As the TBL thresholds are fixed, the purpose of this experiment is to understand the influence that the thresholds have on the behavior of the TBL-CLA under varying environmental conditions, namely the collection length and the TruthTable game state. Since the behavior of the TBL-CLA was nearly identical to that of the Standard-CLA, only one example will be discussed in this section. The example presented below has all of the attributes of the observed behavior of this 4-tuple.

5.3.3.1 TruthTable state 3, Collection Length 12

The TBL-CLA starts the learning process with a significant advantage which quickly disappears (see Figure 27). This early advantage may seem counterintuitive, as the independence threshold is set to the minimum possible value (*i.e.* there should be no follower stimulants in the STM ever) and the TBL-CLA and the Standard-CLA both start the secondary phase with the same level of mastery of the initial phase. Also note in Figure 28 that there are a large number of independent stimulants, but no followers. This also seems unusual as the setting of the independence threshold suggests an absence of independent stimulants.

Recall that an independent stimulant is a stimulant that has a selection confidence greater than the independence threshold *and* that has at least one effective tactic. While all stimulants will have a selection confidence greater than the independence threshold as soon as they receive their first update, they must also seek out an effective tactic before they can be classified as independent. This implies that all the new stimulants introduced in the secondary phase of the game must be followers for at least one stage (*i.e.* one evaluation). As soon as the follower identifies an effective tactic, it immediately abandons it and becomes independent. This can all happen in the contests between the test points so that even though the follower stimulants are not recorded in the test points, the effects of having made several correct responses early on is seen at the first few test points. Once the stimulants have either found an effective tactic and become independent or tried all the global tactics and have found them to be ineffective, all the stimulants in the TBL-CLA are following the Standard selection policy and the TBL-CLA's behavior mimics that of the Standard-CLA. This is also born out in Figure 29, which shows that the TBL-CLA and the Standard-CLA make that same number of random selections during the learning process.

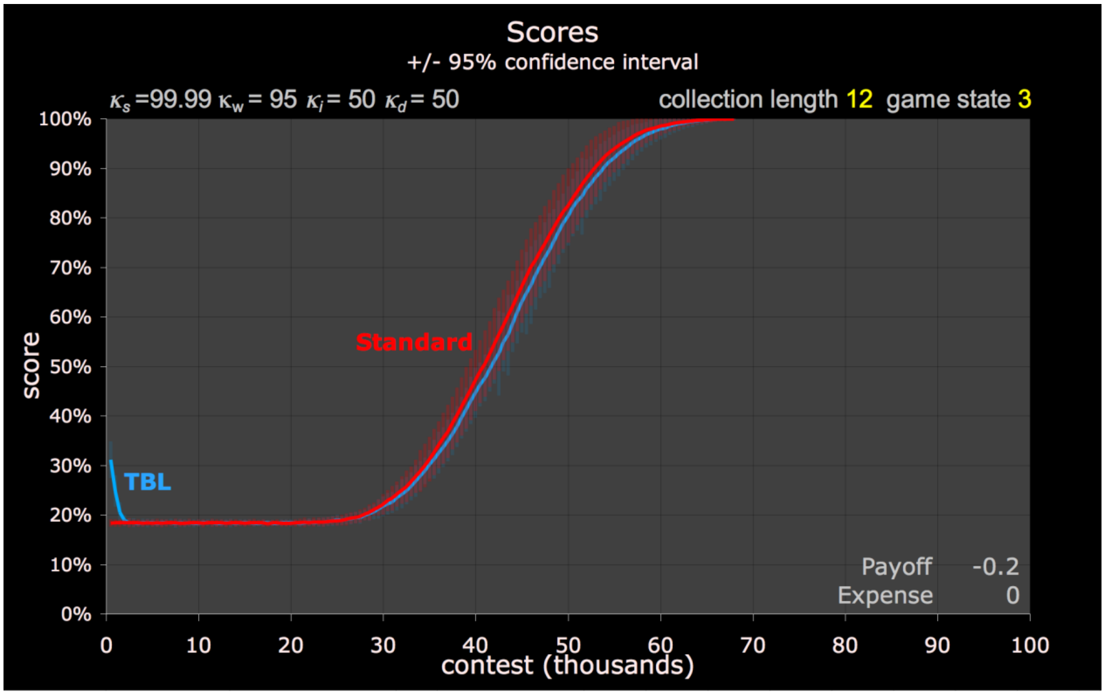


Figure 27: Despite a significant advantage during the very first contests, the TBL-CLA's learning behavior quickly become indistinguishable from the Standard-CLA's behavior.

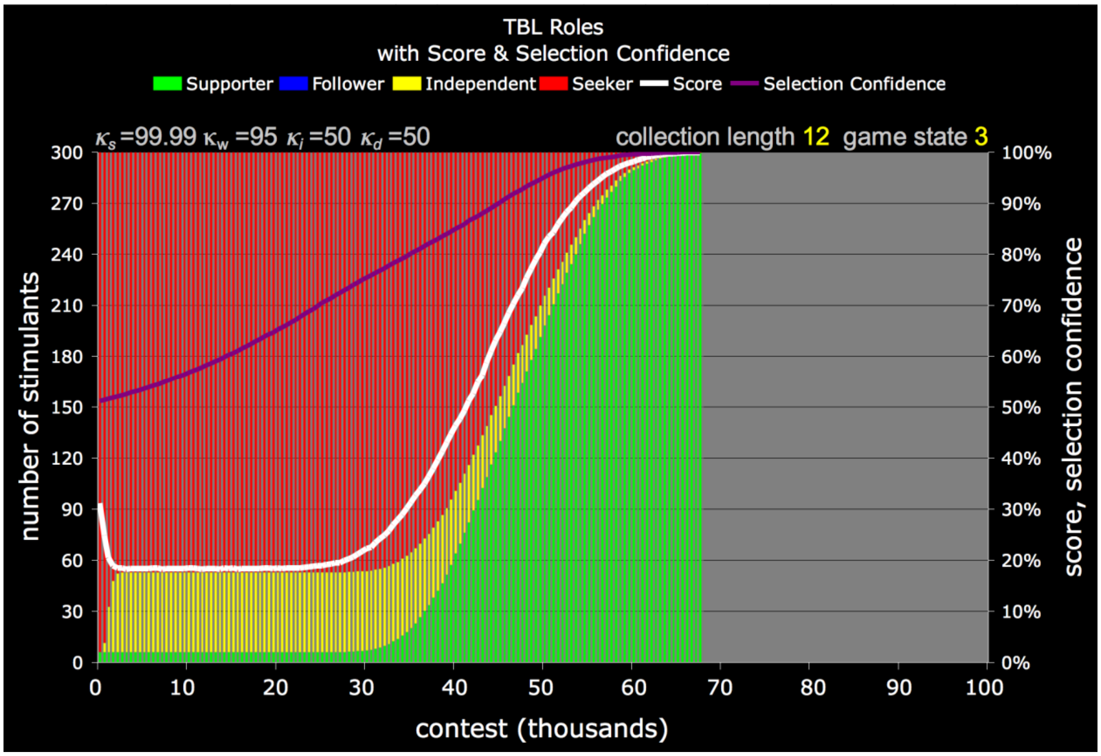


Figure 28: The TBL-CLA does not have any follower stimulants, which accounts for the fact that the TBL-CLA behaves like a Standard-CLA.

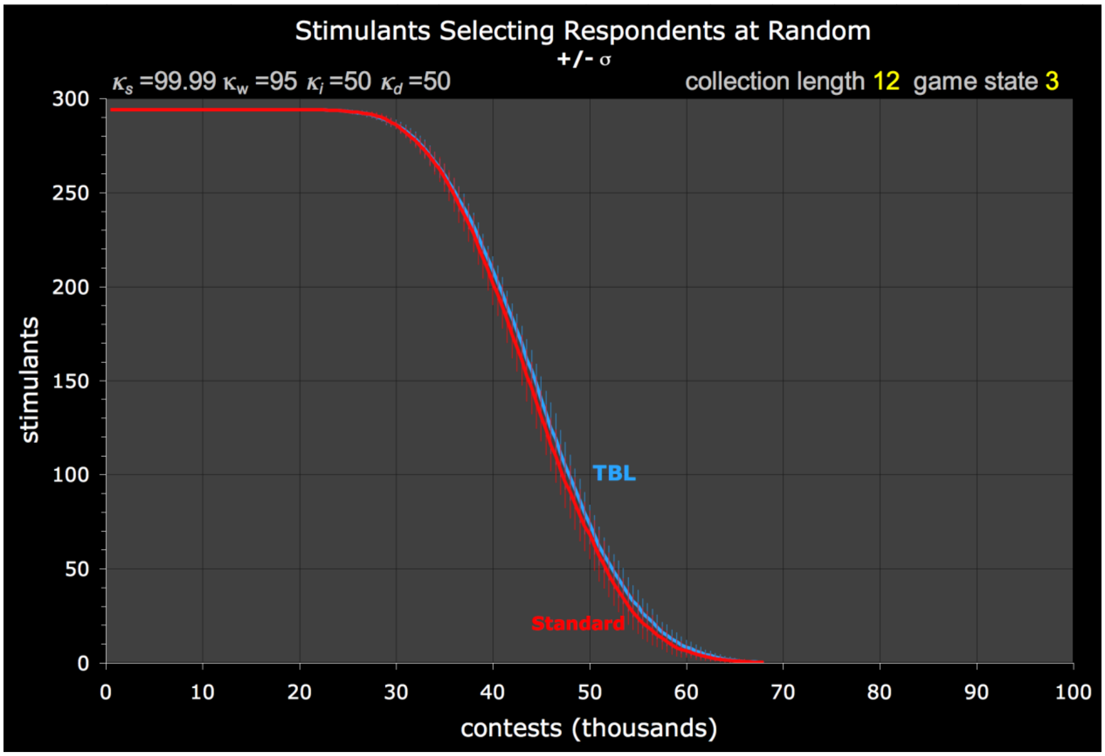


Figure 29: The TBL-CLA does not have any follower stimulants and so it makes the same amount of random selections as the Standard-CLA.

5.3.4 Poor Performance: 4-tuple (κ_s=95, κ_w=95, κ_i=95, κ_d=50)

Many of the 4-tuples from Experiment 1 produced TBL-CLAs that slightly underperformed when compared to a Standard-CLA. One such 4-tuple, (95, 95, 95, 50), is discussed in this section. The poor performing 4-tuples did not always do worse than the Standard-CLA, which makes the potentially even more hazardous than a 4-tuple that produced consistently poor performance. These poor performing 4-tuples can produce behavior that is similar to the best 4-tuples, or similar to the neutral 4-tuples, or behavior that is very unstable. These three cases are presented below.

5.3.4.1 TruthTable state 3, Collection Length 2

At shorter collection lengths, the TBL-CLA is able to perform in much the same way as the best 4-tuples (see Figure 30, Figure 31, and Figure 32). While this shows that TBL makes good use of the information provided in situations with a low signal-to-noise ratio, this is not a particularly useful benefit because learning is very easy in these situations.

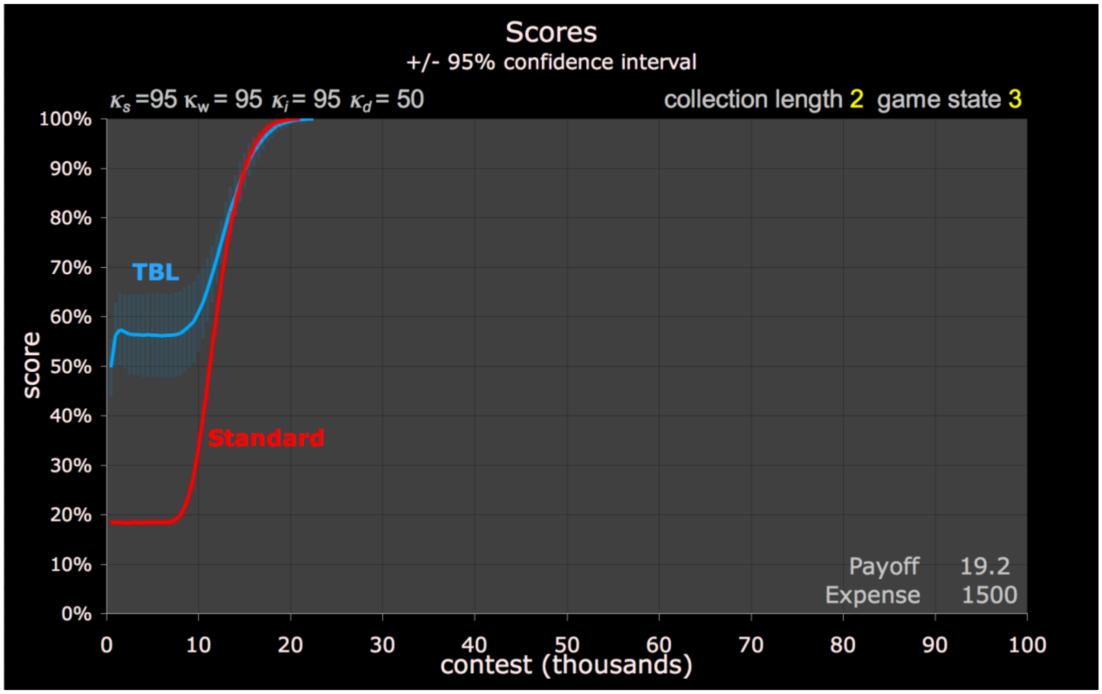


Figure 30: With a short collection length, the TBL-CLA can do significantly better then the Standard-CLA in early learning.

The early gains in learning come from the large number of follower stimulants that exist in the STM. When the support and independence thresholds are equal, it is not possible for a stimulant to ever be independent. In environments where there is only a single target response for each stimulus, this is advantageous because there is nothing to be gained from exploring the response range once an effective tactic is found; there are no other correct responses to be found.

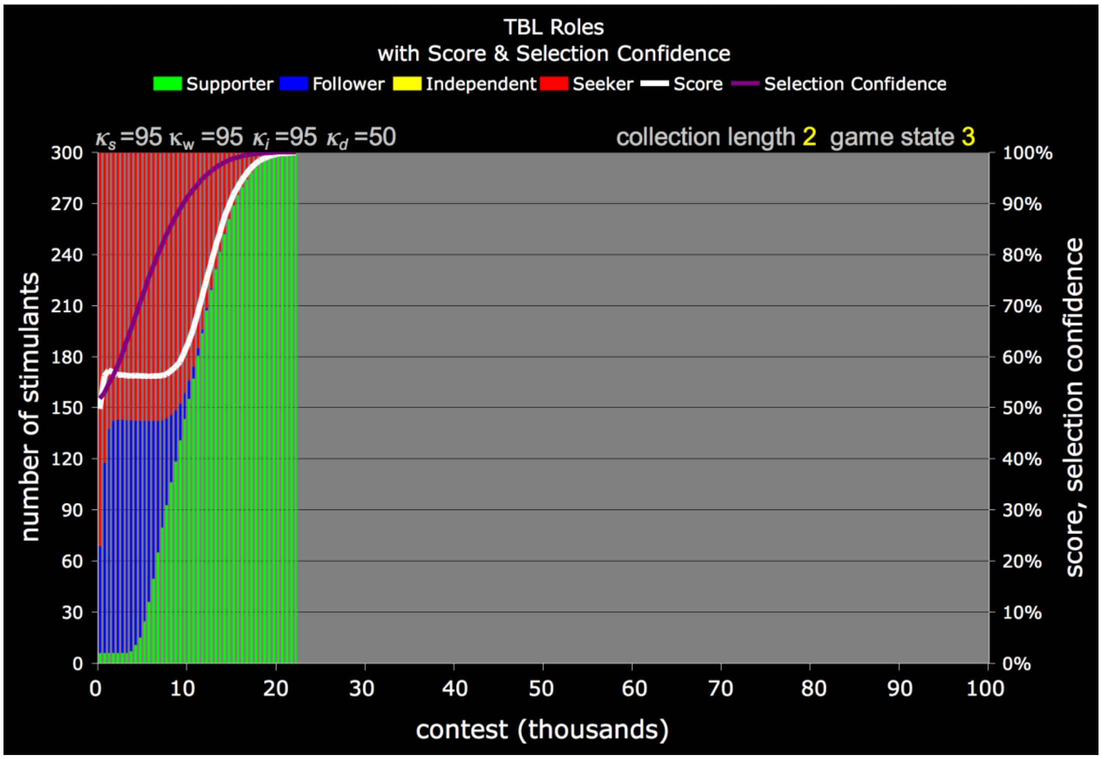


Figure 31: The TBL threshold settings allow the CLA to use effective tactics for a longer period of time. When the support and independence thresholds are equal, stimulants never become independent. Followers use their effective tactics until they become supporters.

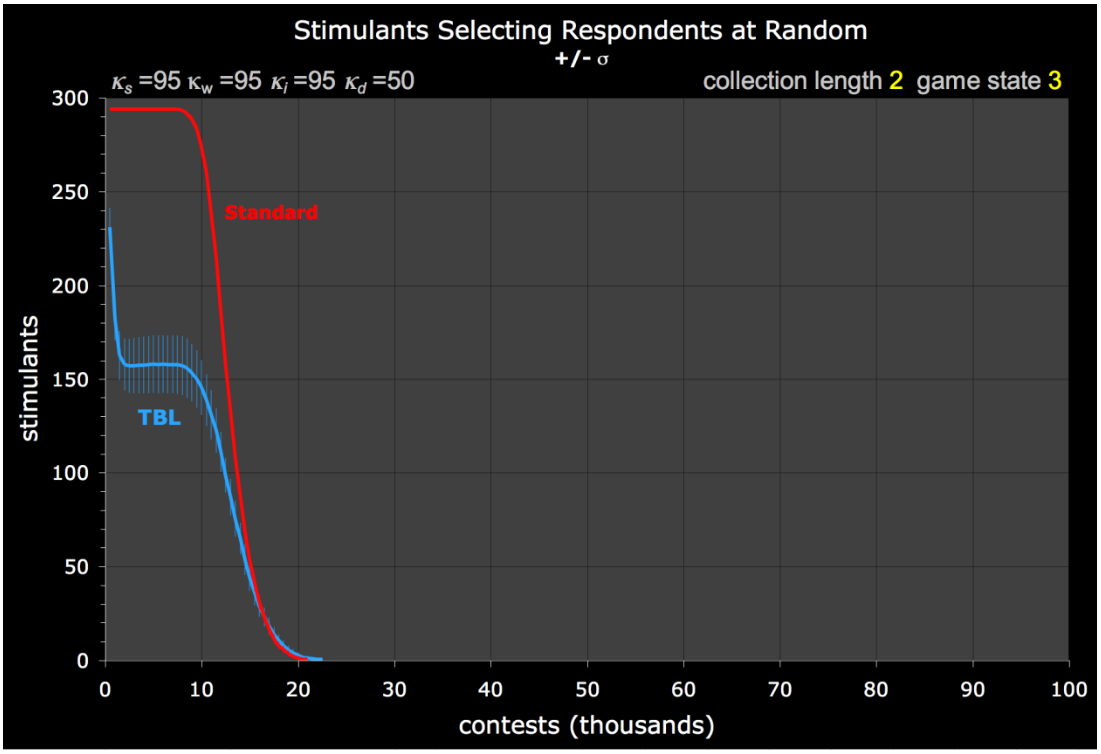


Figure 32: The TBL-CLA has many follower stimulants and therefore has significantly fewer stimulants that make random selections than the Standard-CLA.

5.3.4.2 TruthTable state 3, Collection Length 6

While the 4-tuple shows promise at a collection length of 2 contests, it causes a TBL-CLA to perform just like the neutral 4-tuples when the collection length is extended to 6 contests. The behavior shown in Figure 33 and Figure 35 is very similar to the behavior seen in the neutral performing TBL-CLA in Section 5.3.3.1.

As the collection length gets longer, the TBL-CLA is more likely to generate harsh update values early on. Recall that the compensation threshold was set to 50% for Experiments 1 and 2. This means that any time a follower stimulant that has found an effect tactic is in a history with a follower of an ineffective tactic, it generates a punitive update value. This can discourage the use of tactics and soon most stimulants become seekers and use the Standard selection policy. This behavior is shown in Figure 34.

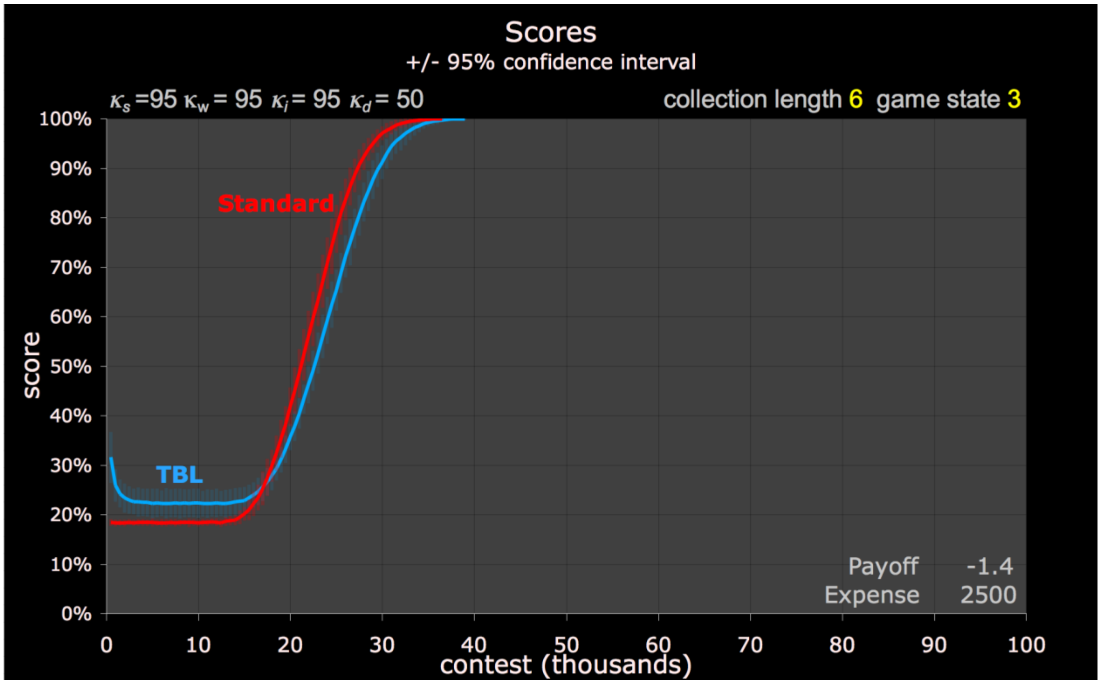


Figure 33: At a longer collection length, the TBL-CLA reverts to neutral performance under these 4-tuple settings.

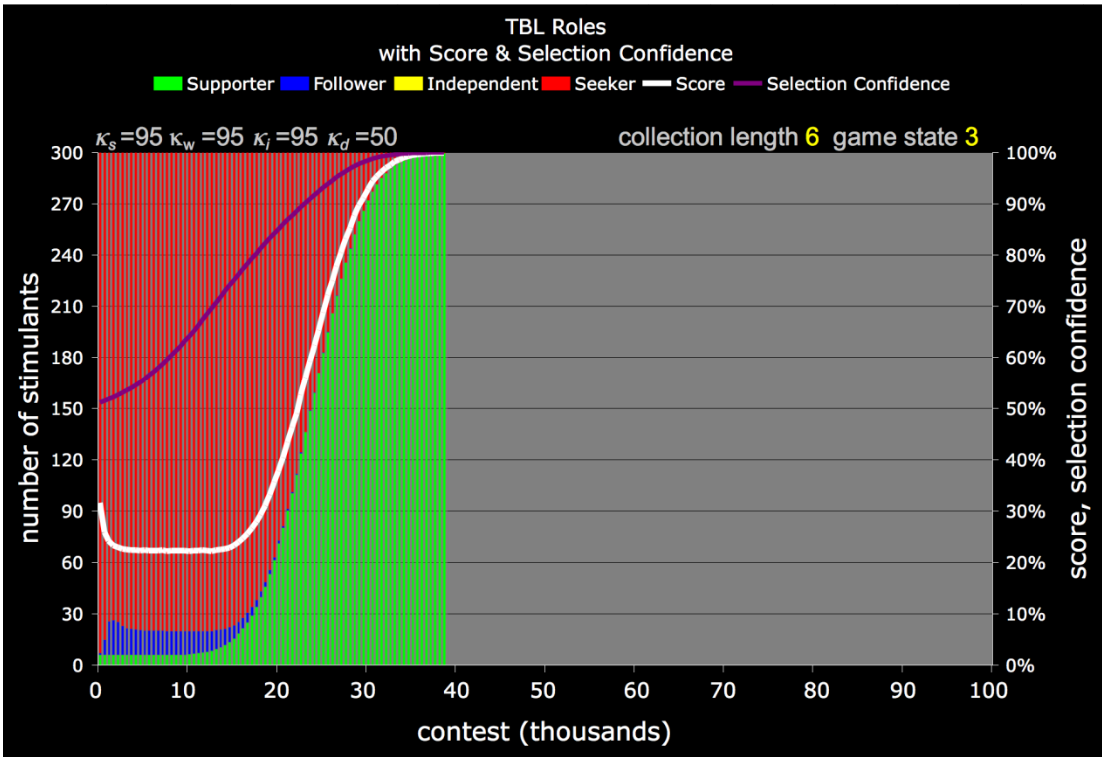


Figure 34: The TBL-CLA uses some follower stimulants early on, which account for the slight, but significant, advantage it has over the Standard-CLA.

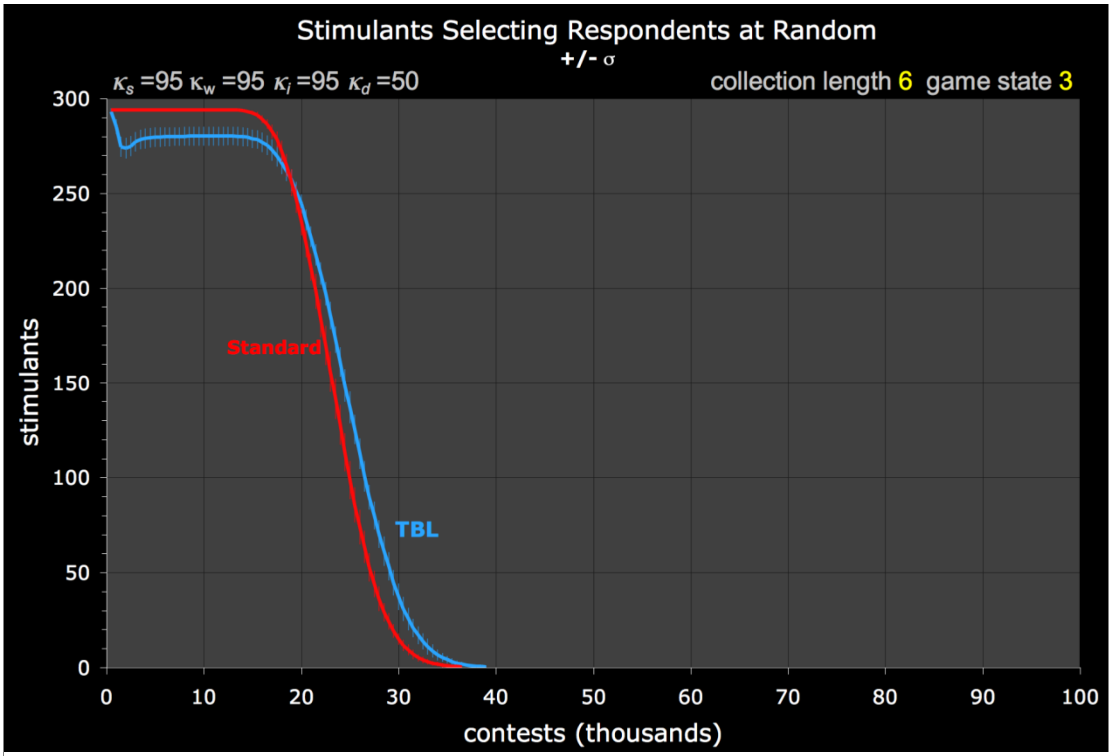


Figure 35: The TBL-CLA has a similar number of stimulants making random selection as the Standard-CLA.

5.3.5 Formal Conclusions from Experiment 2

A summary of the conclusions drawn from Experiment 2 is presented in this section. For a more detailed discussion of these conclusions, see the previous section. All conclusions presented in this section are only valid for the TruthTable game in an environment with *one target per output* and that is *stationary, deterministic, and correct* for the duration of the learning process. In formal conclusions, speculations, and predictions about the performance of the application of TBL to other games, including actual-life games, will be presented and discussed in Section 5.8. Suggestions for future research are presented and discussed in Section 5.9.

Conclusion 1: The learning behavior of a TBL-CLA is significantly affected by the settings of the TBL thresholds. (Conclusion 1 from Experiment 1)

Conclusion 1d: In an environment with only one target response per input, there is no need for independent stimulants because there is no alternate response which could provide positive evaluations; therefore, a TBL-CLA performance improves when the independent role is removed by setting the support and independence thresholds equal to each other.

Conclusion 3: When the TBL thresholds are set for good performance, the TBL-CLA has significantly fewer stimulants that make random selections than the Standard-CLA, meeting one of the performance criteria for this research.

Conclusion 4: Only a small percentage of the total stimulants need be follower stimulants for a TBL-CLA to perform significantly better than a Standard-CLA.

5.4 Experiment 3

Experiment 3 is a canonical experiment. The TruthTable game environment is stationary throughout the learning process, that is, the arrangement of target cells never changes, and there are two target cells per input. Below in Table 10, taken from Section 4.1.4.3, is the experiment block design with all of the factors. Both Standard and TBL-CLAs are used. The results from the Standard-CLA are used as a baseline for computing the performance measures.

Table 10: Design for Experiment 3 (canonical, stable game, 2 target cells per input) [from Section 4.1.4.3]

Name	**Factor values**	**Treatments**
Target responses	{1, 2, 3, 6}	4
Collection length, *c*	{1, 2, 4, 6, 12}	5
TBL thresholds, κ*	<($\kappa_{s,}$ $\kappa_{w,}\kappa_{i,}$ κ_d)> selected from the following set of values, subject to the specified constraints: <50.00, 70.00, 90.00, 95.00, 99.99>	66
Experiment resource requirements		
Total treatments		**1340***
Estimated time per treatment		2 CPU minutes
Estimated total CPU time required		1.9 CPU days
Total CPUs available		4
Estimated total time required		**1 day**
* total number of treatments calculated as follows: Standard-CLA treatments = 4(5) TBL-CLA treatments = 4(5)(66) Total = Standard-CLA treatments + TBL-CLA treatments = 1340		

5.4.1 Results

The results are presented in footprints in this section and have been sized to fit a single page. In many cases, this limits the legibility of the data labels. The footprints are intended to give an overview of trends in behavior and performance as the factors are varied. Each performance measure has a unique dynamic range, but they are all presented in the same color range (red to green). Because the footprints are used to visualize trends, the exact values are not as important. Bright green is always used for results that favor the TBL-CLA and bright red for those that favor the Standard-CLA. The footprints are also presented at a legible resolution with individual color scale information over several pages in the following appendices: APPENDIX D: EXPERIMENT 3, PAYOFF

RESULTS, APPENDIX E: EXPERIMENT 3, EXPENSE RESULTS, and APPENDIX F: EXPERIMENT 3 *N*-TILE RESULTS.

Table 22 shows the results of Experiment 3 sorted by the TBL thresholds. Section 3.6.4 describes the layout of the footprint in greater detail, but briefly, the columns and rows are organized in a hierarchical fashion.

The columns are divided first by the performance metric: Payoff, then Expense, then *n*-tile advantage. Within each performance metric, the columns are subdivided by collection length. Finally, within each collection length, the columns are again subdivided by the TruthTable game state, which corresponds to the number of target outputs in each game state.

The rows are organized hierarchically by TBL thresholds. The thresholds are presented in the following order, from left to right:

- Support threshold, κ_s
- Withdrawal threshold, κ_w
- Independence threshold, κ_i
- Dependence threshold, κ_d

The factor values for the TBL thresholds are presented from smallest to largest, according to the following rules:

- The withdrawal threshold must be less than or equal to the support threshold.

  $\kappa_w \leq \kappa_s$

- The independence threshold must be less than or equal to the support threshold.

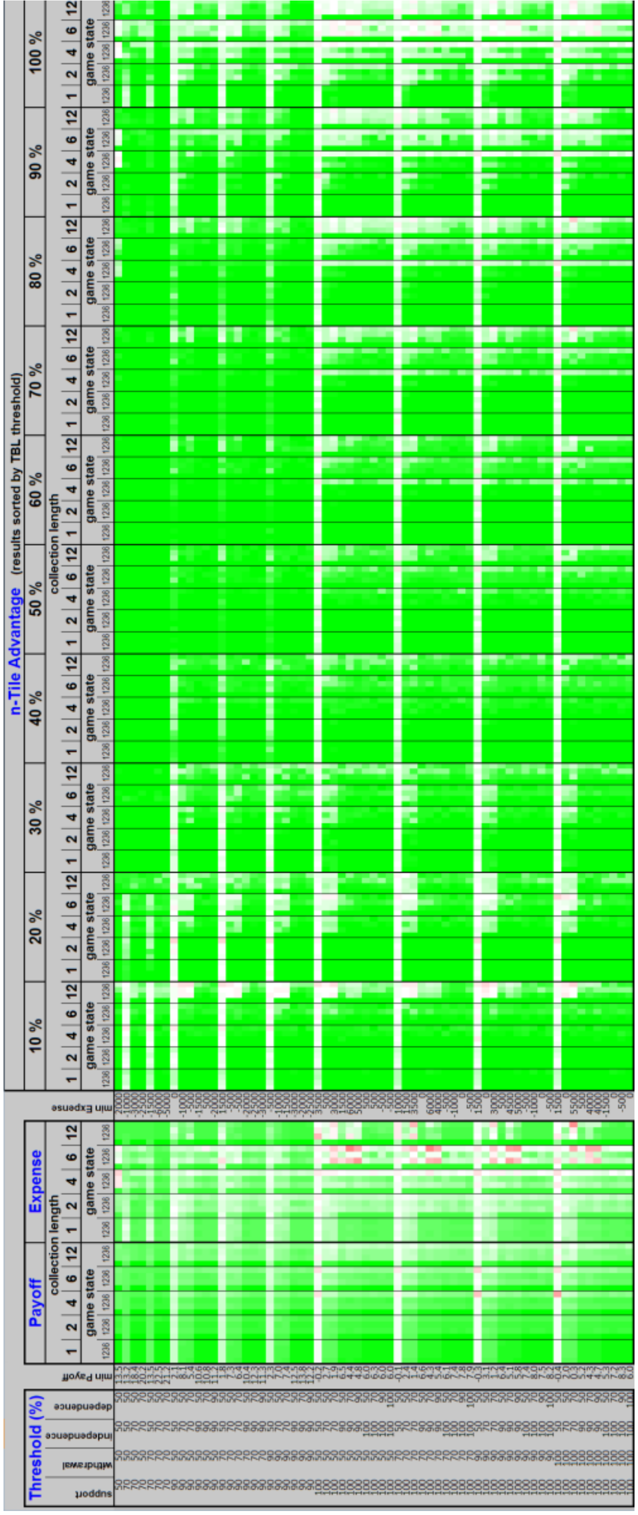
  $\kappa_i \leq \kappa_s$

- The dependence threshold must be less than or equal to the independence threshold.

  $\kappa_d \leq \kappa_i$

Turning to the results in Table 22, it can be seen that the TBL-CLA performs significantly better than the Standard-CLA across almost all the factors and metrics. Increasing the number of target responses per input from one to two is advantageous for the Standard-CLA, but the increase that that gives to the Tactic-Based Learning advantage, TBL$_{\alpha}$, allows the TBL-CLA to do consistently better.

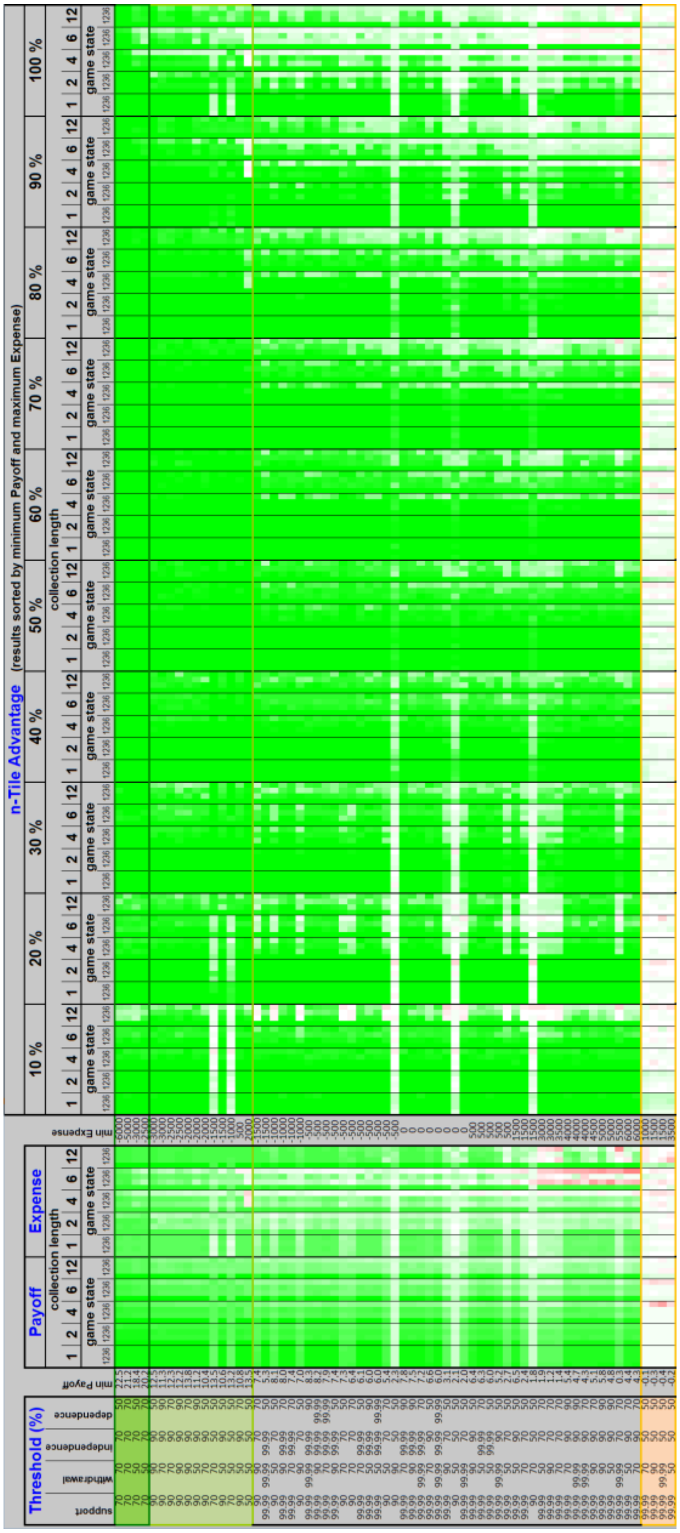
Table 22: Results from Experiment 3, sorted by TBL thresholds. The varying collection lengths and the TruthTable game state are the column headers. The TBL thresholds are the row headers. The thresholds are sorted from smallest to largest factors values in the following order, according to the rules for TBL thresholds: support threshold, withdrawal threshold, independence threshold, and dependence threshold.


Threshold (%)
support
withdrawal
independence
dependence
min Payoff
Payoff
Expense
collection length
1 2 4 6 12
game state
1236
min Expense
n-Tile Advantage (results sorted by TBL threshold)
10 %
20 %
30 %
40 %
50 %
60 %
70 %
80 %
90 %
100 %

In order to draw more specific observations about the performance of a TBL-CLA, it is necessary to reorder the footprint and examine smaller subsections of it. In order to identify those TBL threshold 4-tuples that are most effective, the results are sorted by the minimum Payoff value in each row. The minimum Payoff values fell into four distinct groups: values greater than 18.0, values between 10.0 and 18.0, values between 0.0 and 10.0, and values less than 0.0. These four groups were internally sorted by the maximum Expense value for each row. The resorted results are presented in Table 23.

While almost all of the TBL threshold 4-tuples resulted in a significant advantage for the TBL-CLA, the best results were in the same range as those from Experiment 1: those with the support threshold set at 70%. The TBL-CLA also does very well when the support threshold is set to 90%. The TBL-CLA’s performance degrades as the support threshold is set at 99.99%. The worst-case performance comes when the support threshold is set to 99.99% and the independence threshold is set at 50%. This produces behavior very similar to the Standard-CLA.

Table 23: Results from Experiment 3 sorted by Payoff and Expense. The results were first sorted by minimum Payoff. They were then separated into four groups and resorted by maximum Expense. The results then fell into groups based on the value of the support threshold.



5.4.2 Formal Conclusions for Experiment 3

A summary of the conclusions drawn from Experiment 3 is presented in this section. For a more detailed discussion of these conclusions, see the previous section. All conclusions presented in this section are only valid for the TruthTable game in an environment with *two targets per output* and that is *stationary, deterministic, and correct* for the duration of the learning process. Informal conclusions, speculations, and predictions about the performance of the application of TBL to other games, including actual-life games, will be presented and discussed in Section 5.8. Suggestions for future research are presented and discussed in Section 5.9.

Conclusion 5: The learning behavior of a TBL-CLA is significantly affected by the settings of the TBL thresholds.

Conclusion 5a: For best performance, set the support threshold at 70%.

Conclusion 5b: For strong performance, set the support threshold at 90%.

Conclusion 5c: Worst-case performance for a TBL-CLA is not statistically significantly different from that of a Standard-CLA.

Conclusion 5d: For worst-case performance, set the support threshold to 99.99% and the independence threshold to 50%.

Conclusion 6: TBL-CLA performance improves without bound as the TBL advantage, TBL_{α}, of the TruthTable game state is increased.

5.5 Experiment 4

Experiment 4 accomplishes Goal 3.2, described in Section 3.5. The calculation of the individual performance measures is completed with the use of a spreadsheet, but the process of inspecting the results and drawing conclusions about each case requires significant attention from a person. Table 11 below, from Section 4.1.4.5, presents the block design for Experiment 4.

Table 11: Design for Experiment 4 (close inspection, stationary, 2 target cells per input) [from Section 4.1.4.5]

Name	Factor values	Treatments
TruthTable game states	{1, 2, 3, 6}	4
Collection length, *c*	{1, 2, 4, 6, 12}	5
TBL threshold 4-tuples <(κ_s, κ_w, κ_i, κ_d)>	(70.00, 70.00, 70.00, 70.00) (99.99, 99.99, 99.99, 99.99) (99.99, 50.00, 50.00, 50.00)	3
Experiment resource requirements		
	Total treatments	**80***
	Estimated time per treatment	10 person-minutes
	Estimated time required	**14 hours**
	* total number of treatments calculated as follows: Standard-CLA treatments = 4(5) TBL-CLA treatments = 4(5)(3) Total = Standard-CLA treatments + TBL-CLA treatments = 80	

5.5.1 Results

This section presents selected results from Experiment 4. The full set of reduced results is available in the digital appendices as interactive Microsoft Excel spreadsheets.

5.5.2 Best Performance: 4-tuple (κ_s=70, κ_w=70, κ_i=70, κ_d=50)

It was observed in Experiment 3 that the 4-tuple of (70, 70, 70, 50) was the best performing combination of TBL thresholds. The results of close inspection of the behavior of a TBL-CLA with these settings are presented in this section. As the TBL thresholds are fixed, the purpose of this experiment is to understand the influence that the thresholds have on the behavior of the CLA under varying environmental conditions: the collection length and the TruthTable game state.

5.5.2.1 TruthTable state 6, Collection Length 12

It was observed in Experiment 3 that the TBL-CLA generally dominated the Standard-CLA across all factor conditions. A closer inspection of the CLAs performance on TruthTable game state 6 with a collection length of 12 is discussed in this section because it presents the most challenging combination of factors: a long collection and the lowest possible TBL$_\alpha$.

Figure 36 shows that the scores of the TBL-CLA are significantly and consistently higher than those of the Standard-CLA. Figure 37 shows that the TBL-CLA makes use of tactics all throughout the learning process, implying that it is the use of tactics which leads to such strong performance. Figure 38 shows that the TBL-CLA has significantly fewer stimulants making random selections during the learning process.

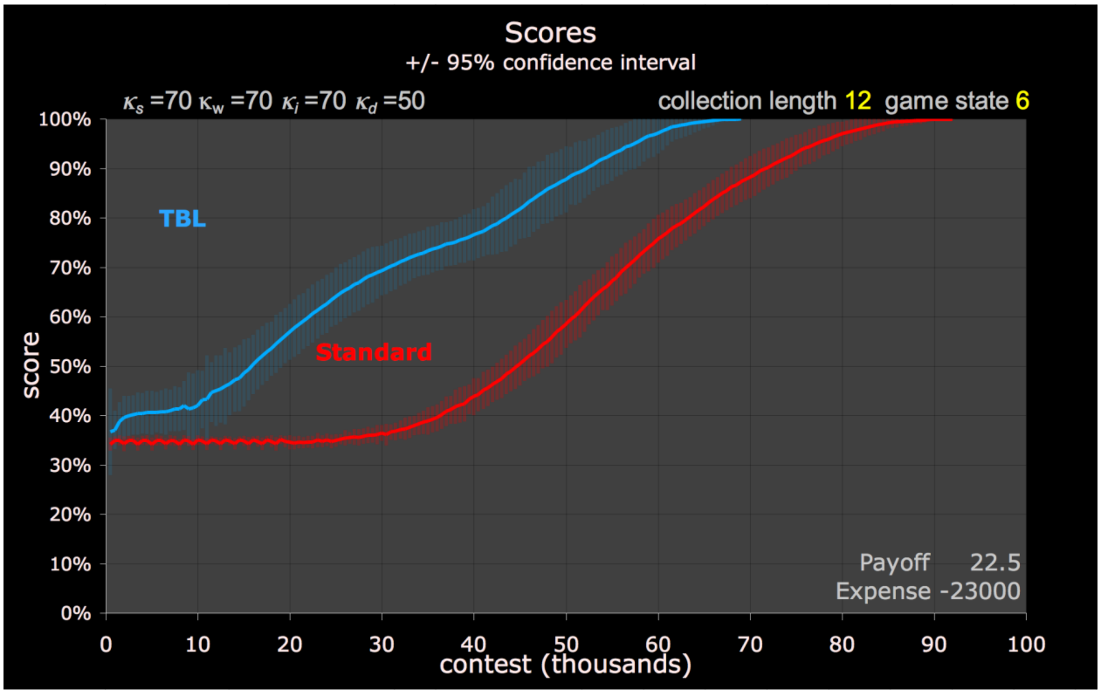


Figure 36: With two target responses for each stimulant, the TBL-CLA's score is significantly and consistently better than that of the Standard-CLA's.

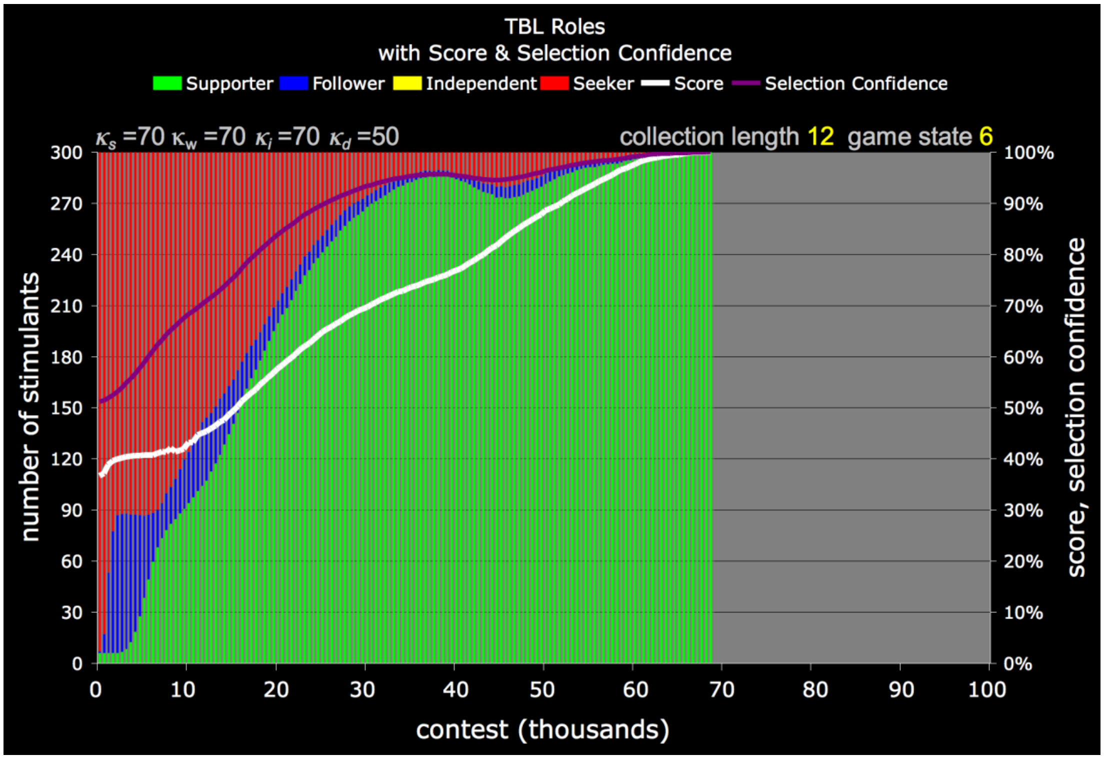


Figure 37: The TBL-CLA makes use of tactics throughout the learning process. When the support threshold and independence thresholds are equal, the TBL-CLA cannot have independent stimulants.

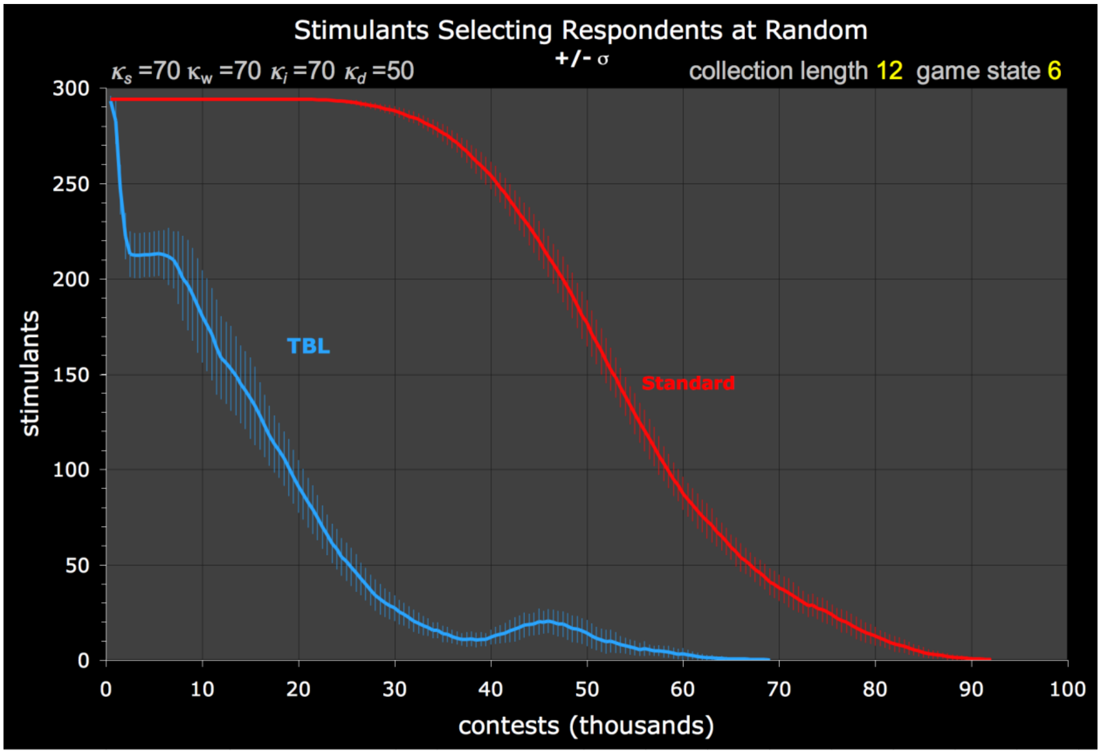


Figure 38: The TBL-CLA makes significantly fewer random selections that the Standard-CLA.

5.5.3 Average Performance: 4-tuple (κ_s= 99.99, κ_w= 99.99, κ_i= 99.99, κ_d= 99.99)

In an average 4-tuple, the TBL-CLA still does very well. The 4-tuple (99.99, 99.99, 99.99, 99.99) produced average and is interesting because it is at the extreme of the setting values.

5.5.3.1 TruthTable game state 2, Collection Length 4

In conditions with a shorter collection length, it is possible to get a better picture of how the TBL-CLA learns. When the TBL thresholds are all set to the maximum value, it creates a lull in the learning progress of the TBL-CLA, which can be seen in Figure 39. A side effect of setting the thresholds so high is that new support stimulants may struggle to maintain their selection confidence. If the selection confidence falls below 99.99%, the new supporters are forced to become seekers. At a selection confidence of 99.99%, these stimulants will already be confident stimulants and will consistently choose the same respondent; however, if some of these confidence stimulants continue to experience a drop in selection confidence, they may no longer be confident stimulants and may make some random selections until they regain their selection confidence. Figure 40 and figure 41 show that the plateau in the learning curve matches the increase in seeker stimulants and the increase in stimulants making random selections.

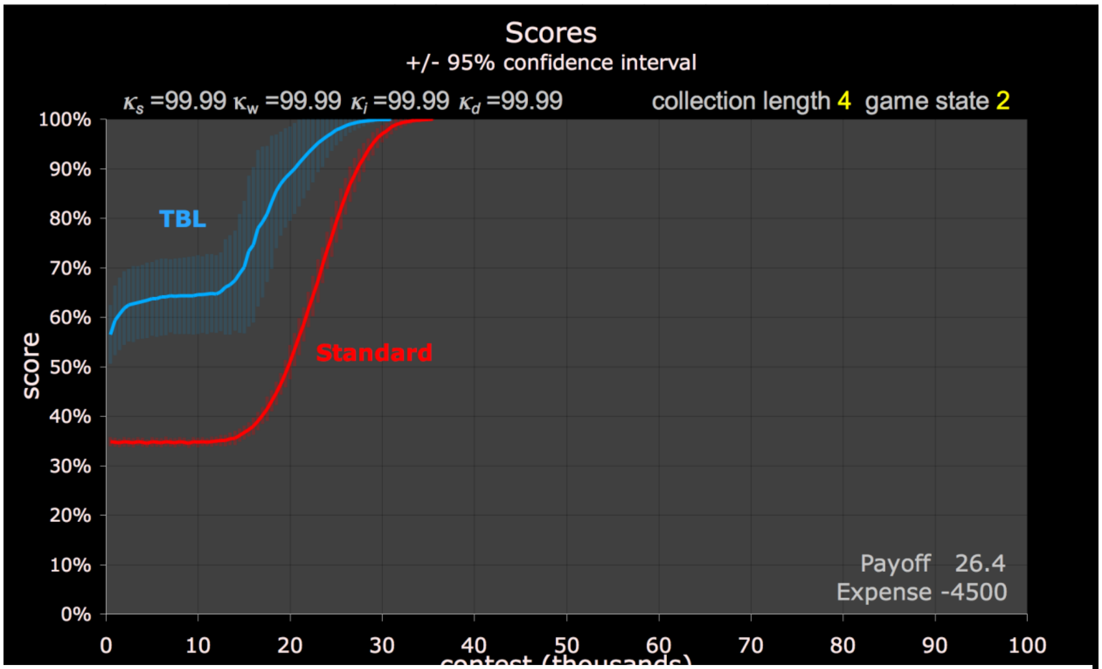


Figure 39: The TBL-CLA produces a staircase-style learning curve.

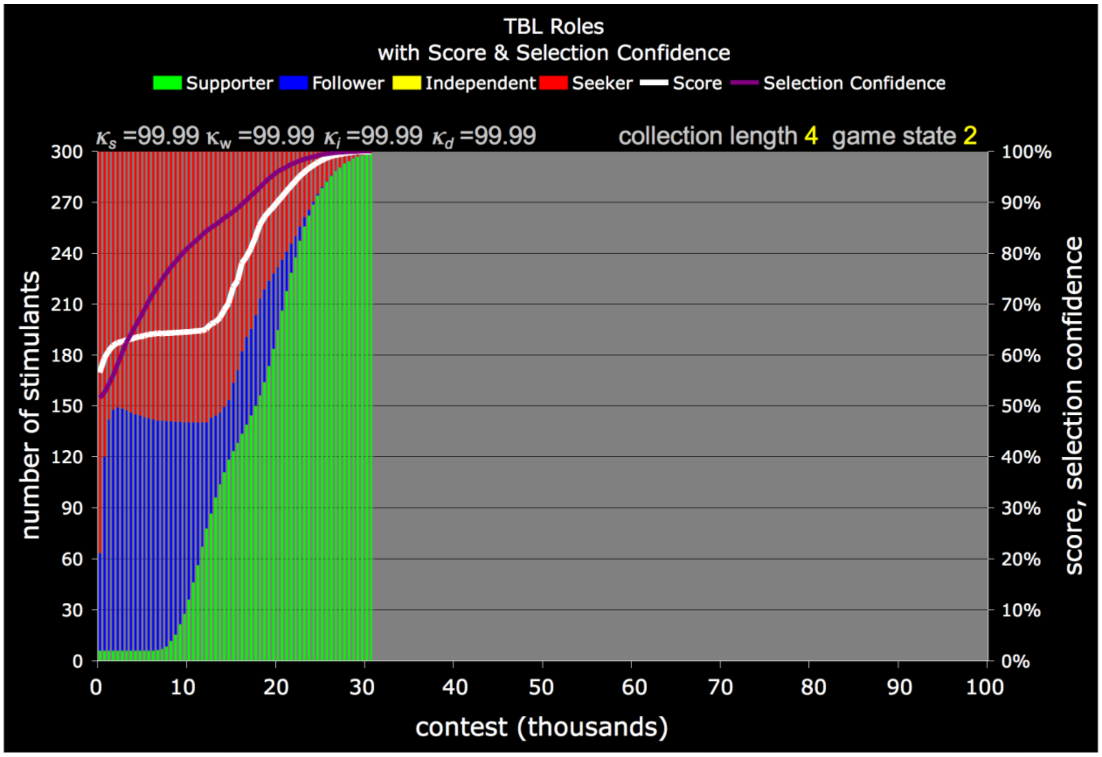


Figure 40: The TBL-CLA experiences a slight increase in the number of seeker stimulants halfway through the learning process. This accounts for the brief stagnation in score performance.

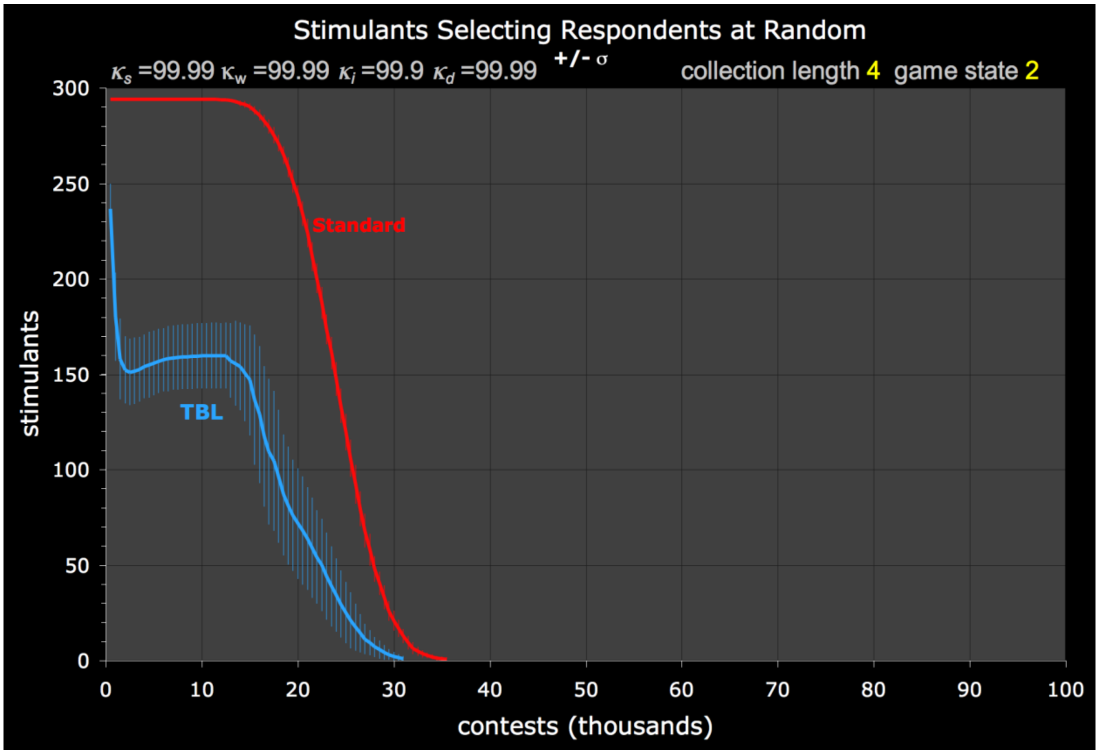


Figure 41: The number of stimulants making random selections increases with the number of seeker stimulants, but remains significantly fewer then the Standard-CLA

5.5.4 Poor Performance: 4-tuple (κ_s= 99.99, κ_w= 50, κ_i= 50, κ_d= 50)

The worst-case results in Experiment 3 were those where the TBL-CLA's performance was not significantly different than that of the Standard-CLA's. This section examines one of the 4-tuples that produces such performance.

5.5.4.1 TruthTable game state 2, Collection Length 12

When the TBL-CLA cannot use its tactics, it behaves like a Standard-CLA because it is using the Standard selection policy throughout the learning process. This behavior is very similar to the behavior that was seen in the neutral example in Experiment 2 (see Section 5.3.3). Figure 42 and Figure 44 show that the TBL-CLA's performance is not significantly different from the Standard-CLA's score and number of random selections. Figure 43 shows that the lack of follower stimulants accounts for this behavior.

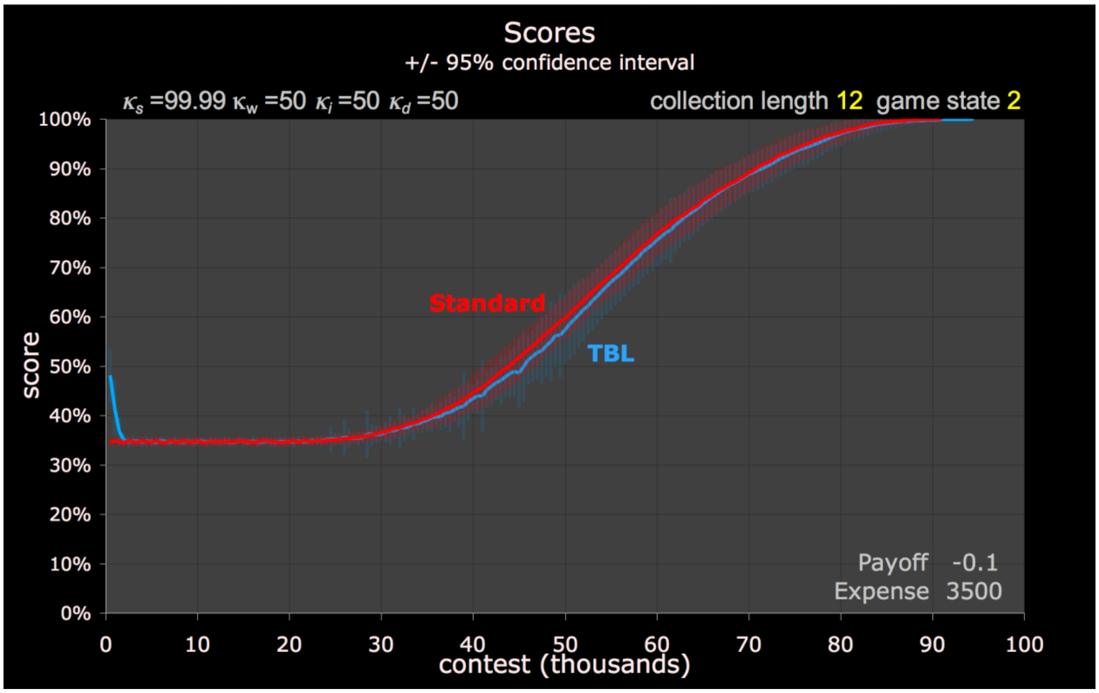


Figure 42: Despite a slight advantage in the first few test points, the TBL-CLA scores become indistinguishable from the Standard-CLA's scores.

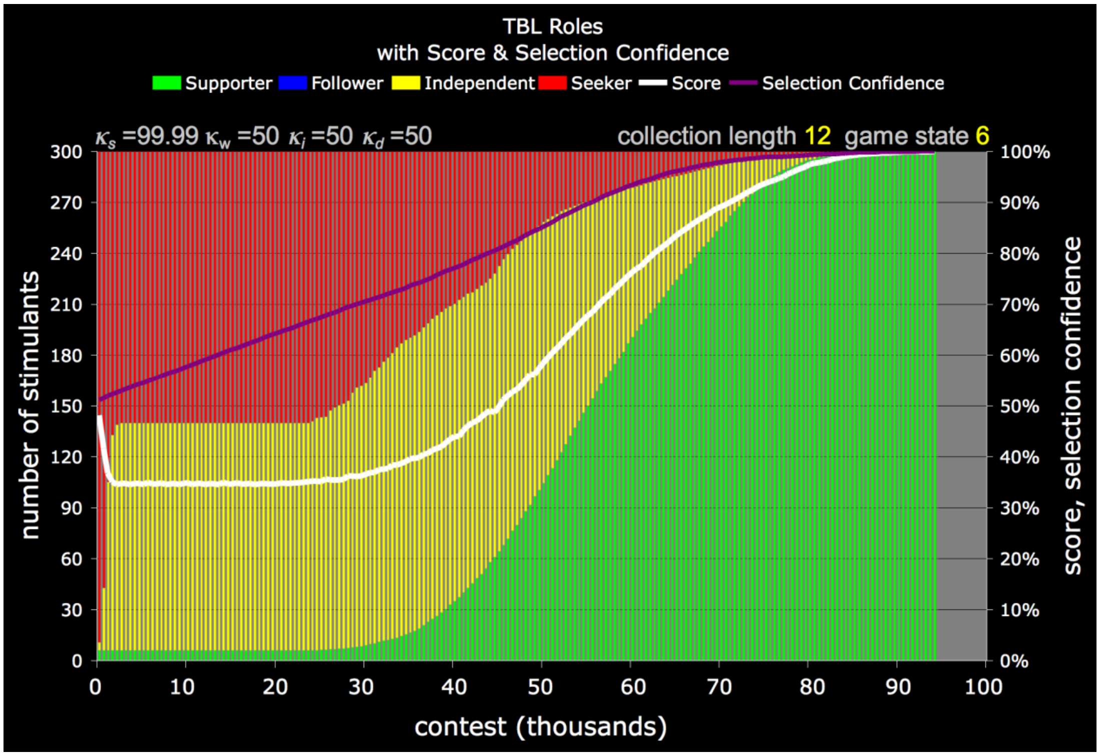


Figure 44: The TBL thresholds prevent the stimulants from using tactics for any significant period of time which has the consequence of forcing the TBL-CLA to follow the Standard selection policy during the learning process.

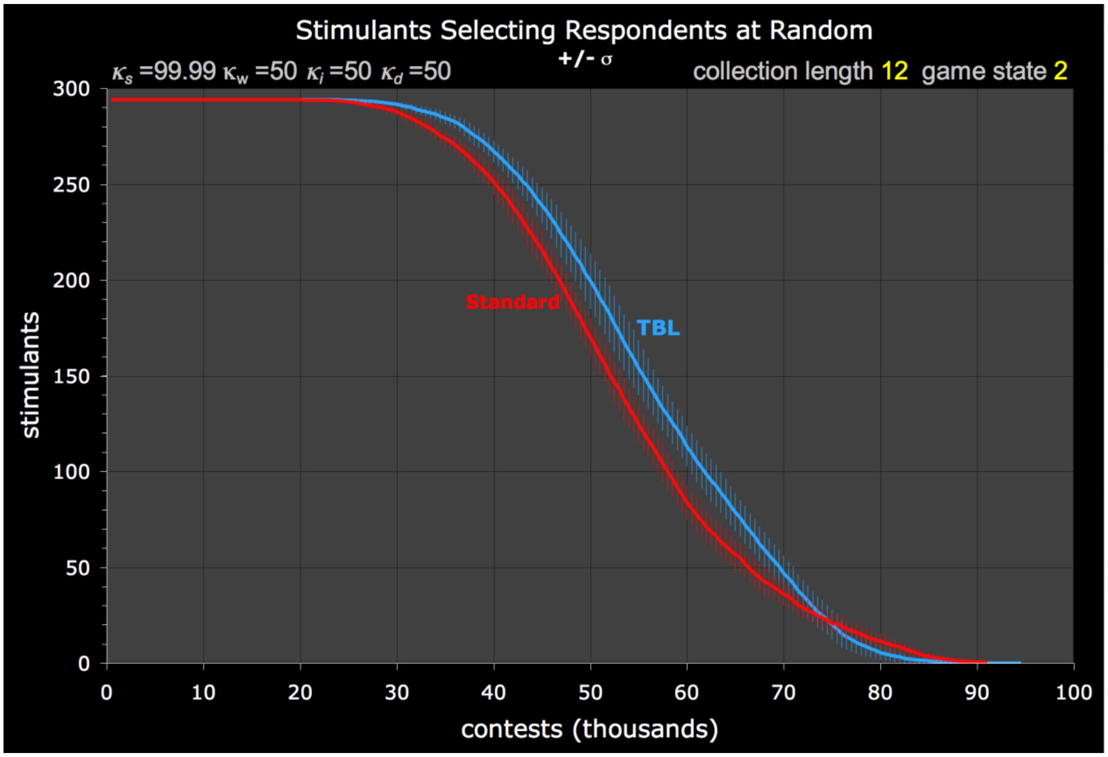


Figure 43 The TBL-CLA has as the same number of stimulants making random selections as the Standard-CLA.

5.5.5 Formal Conclusions for Experiment 4

A summary of the conclusions drawn from Experiment 4 is presented here in summary form. For a more detailed discussion of these conclusions, see the previous section. All conclusions presented in this section are only valid for the TruthTable game in an environment with *two targets per output* and that *is stationary, deterministic, and correct* for the duration of the learning process. In formal conclusions, speculations, and predictions about the performance of the application of TBL to other games, including actual-life games, will be presented and discussed in Section 5.8. Suggestions for future research are presented and discussed in Section 5.9.

Conclusion 5: The learning behavior of a TBL-CLA is significantly affected by the settings of the TBL thresholds. (Conclusion 5 from Experiment 3)

Conclusion 5e: When all the TBL thresholds are set to 99.99%, the TBL-CLA's performance plateaus briefly as the selection confidence rises.

5.6 Experiment 5

Experiment 5 is a canonical experiment. The TruthTable game environment is *not* stationary throughout the learning process, that is, the arrangement of target cells changes after completion of the secondary phase, and there is only one target cell per input. For more detail on the experiment design, see Section 4.1.5. The results presented in this section are for the tertiary phase of the game only. Below in Table 15, taken from Section 4.1.5.4 is an experiment block design with all of the factors. Both Standard and TBL-CLAs are used. The results from the Standard-CLA are used as a baseline for computing the performance measures.

Table 15: Design for Experiment 5 (canonical, task-switching, 1 target cell per input) [from Section 4.1.5.4]

Name	Factor values	Treatments
Target responses	{1, 2, 3, 6}	4
Collection length, *c*	{1, 2, 4, 6, 12}	5
TBL thresholds, κ*	$<(\kappa_S, \kappa_W, \kappa_i, \kappa_d)>$ selected from the following set of values, subject to the specified constraints: <50.00, 70.00, 90.00, 98.00, 99.99>	66

Experiment resource requirements	
Total treatments	**1340***
Estimated time per treatment	8 CPU minutes
Estimated total CPU time required	7.5 CPU days
Total CPUs available	4
Estimated time required	**2 days**
* total number of treatments calculated as follows: Standard-CLA treatments = 4(5) TBL-CLA treatments = 4(5)(66) Total = Standard-CLA treatments + TBL-CLA treatments = 1340	

5.6.1 Results

The results are presented in footprints in this section and have been sized to fit a single page. In many cases, this limits the legibility of the data labels. The footprints are intended to give an overview of trends in behavior and performance as the factors are varied. Each performance measure has a unique dynamic range, but they are all presented in the same color range (red to green). Because the footprints are used to visualize trends, the exact values are not as important. Bright green is always used for results that favor the TBL-CLA and bright red for those that favor the Standard-CLA. The footprints are also presented at a legible resolution with individual color scale information over several pages in the following appendices: APPENDIX G: EXPERIMENT 5, PAYOFF RESULTS;

APPENDIX H: EXPERIMENT 5, EXPENSE RESULTS; and APPENDIX I: EXPERIMENT 5 *N*-TILE RESULTS.

Table 24 shows the results of Experiment 5 sorted by the TBL thresholds. Section 3.6.4 describes the layout of the footprint in greater detail, but briefly, the columns and rows are organized in a hierarchical fashion.

The columns are divided first by the performance metric: Payoff, then Expense, then *n*-tile advantage. Within each performance metric, the columns are subdivided by collection length. Finally, within each collection length, the columns are again subdivided by the TruthTable game state, which corresponds to the number of target outputs in each game state.

The rows are organized hierarchically by TBL thresholds. The thresholds are presented in the following order, from left to right:

- Support threshold, κ_s
- Withdrawal threshold, κ_w
- Independence threshold, κ_i
- Dependence threshold, κ_d

The factor values for the TBL thresholds are presented from smallest to largest, according to the following rules:

- The withdrawal threshold must be less than or equal to the support threshold.

  $\kappa_w \leq \kappa_s$

- The independence threshold must be less than or equal to the support threshold.

  $\kappa_i \leq \kappa_s$

- The dependence threshold must be less than or equal to the independence threshold.

$$\kappa_d \leq \kappa_i$$

Table 24: Results from Experiment 5, sorted by TBL thresholds. The varying collection lengths and the TruthTable game state are the column headers. The TBL thresholds are the row headers. The thresholds are sorted from smallest to largest factors values in the following order, according to the rules for TBL thresholds: support threshold, withdrawal threshold, independence threshold, and dependence threshold.

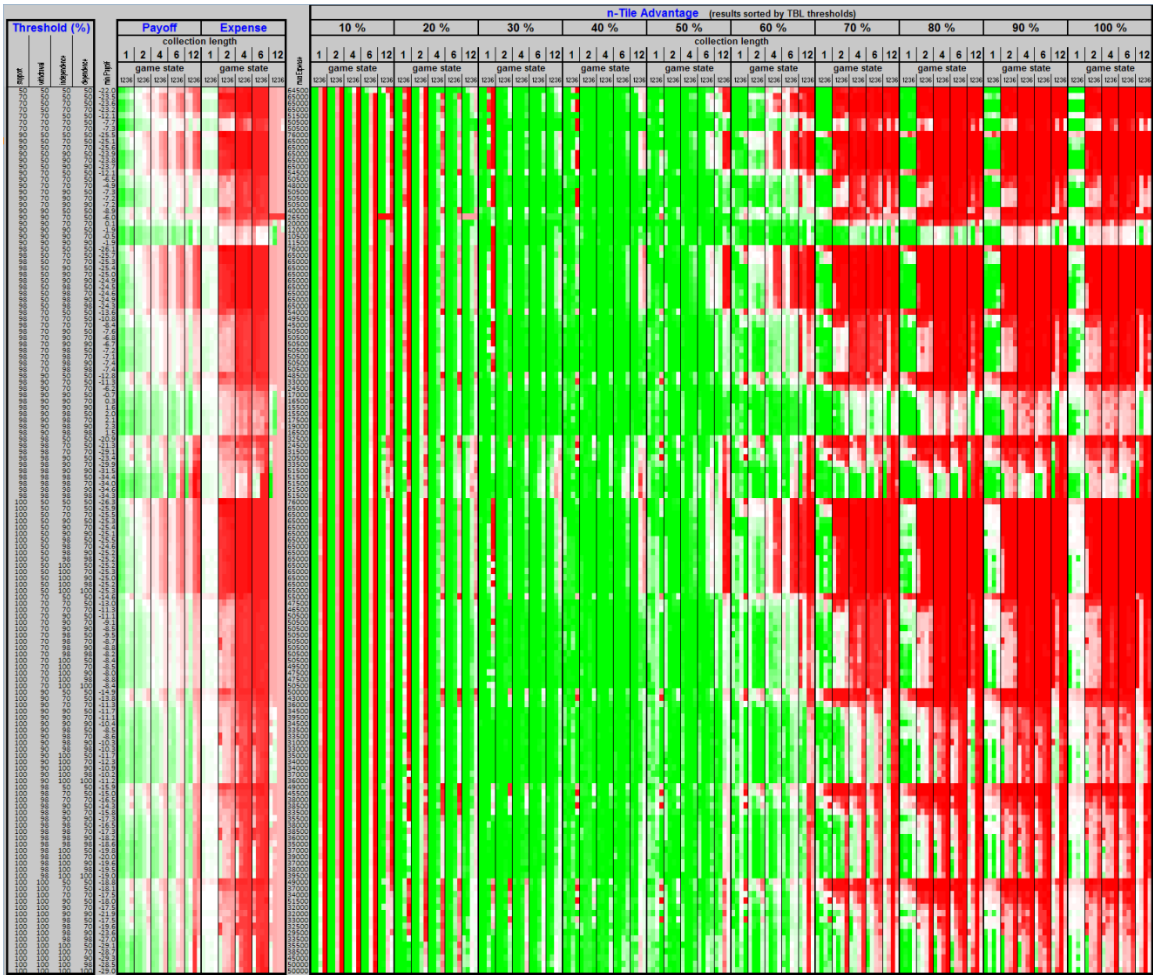
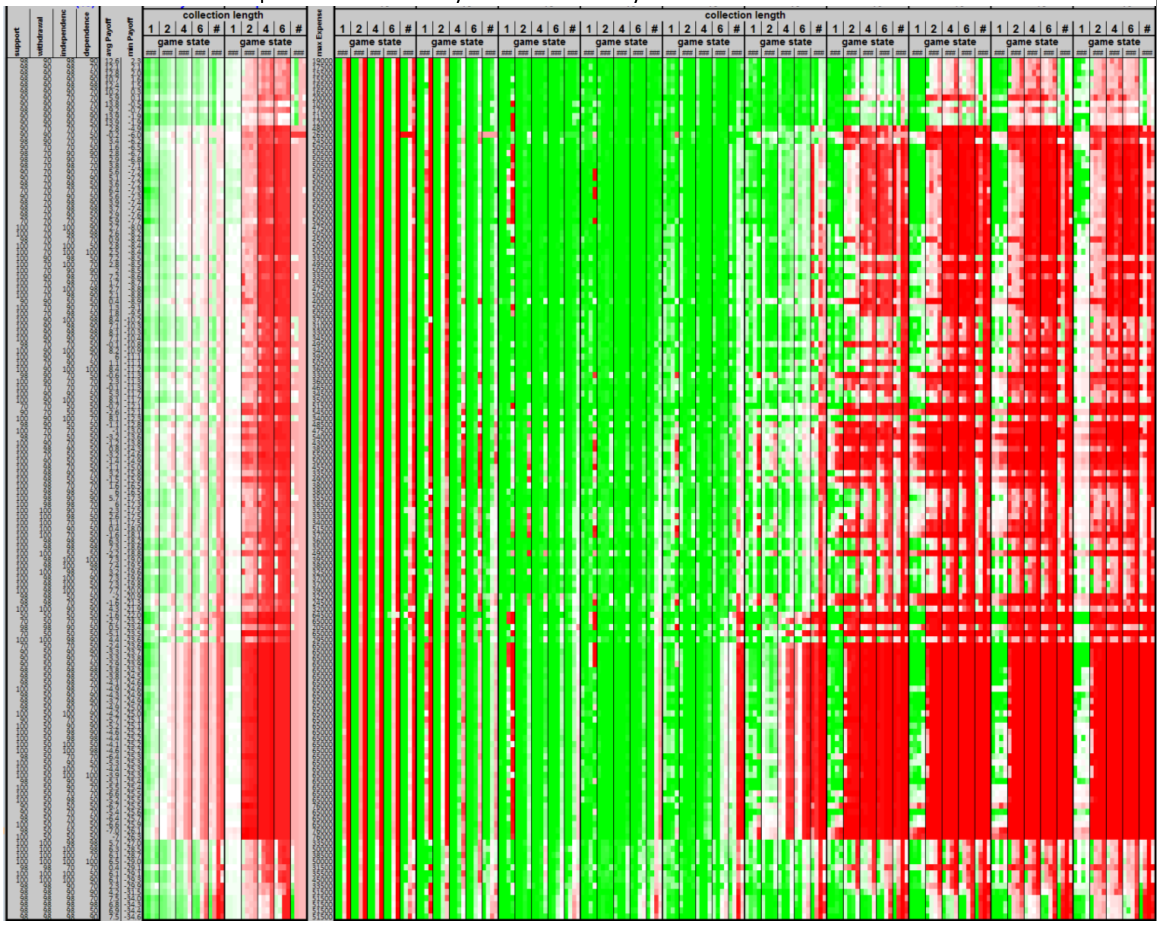


Table 25: The results of Experiment 5 sorted by the minimum Payoff in each row


support
withdrawal
independenc
dependence
avg Payoff
min Payoff
collection length
1
2
4
6
#
game state
max Expense
collection length
game state

Turning to the results in Table 24, it can be seen that the TBL-CLA does not have as strong as an advantage in a task-switching game as it is in the other two stationary games that are examined in Experiments 1 and 3. While it can also be seen that the TBL thresholds do have an effect on the behavior, it necessary to reorder the footprint in order to gain further insight into the relationship between the threshold settings and behavior.

Table 25 shows the results of Experiment 5 sorted by the minimum Payoff in each row. While this technique is useful for revealing patterns in the results in Experiment 1 and Experiment 3, it did not group the patterns of behavior very well. The data was then resorted by the maximum Expense and *average* Payoff in each row, and this did a better job of grouping the results in a useful way. Table 26 shows the sorted and grouped data.

Table 26: Results of Experiment 5 sorted first by the maximum Expense in each row. The results were then group and internally sorted by the average Payoff in each row. This revealed the major groupings.

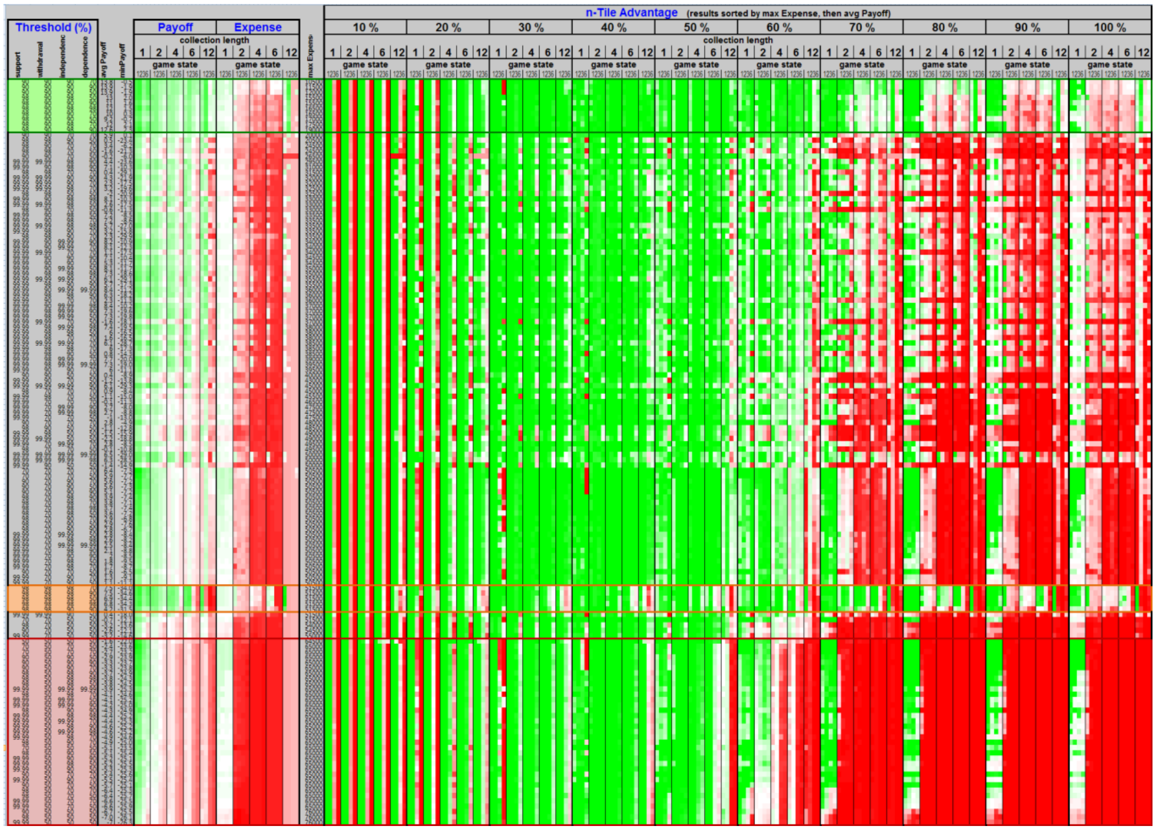
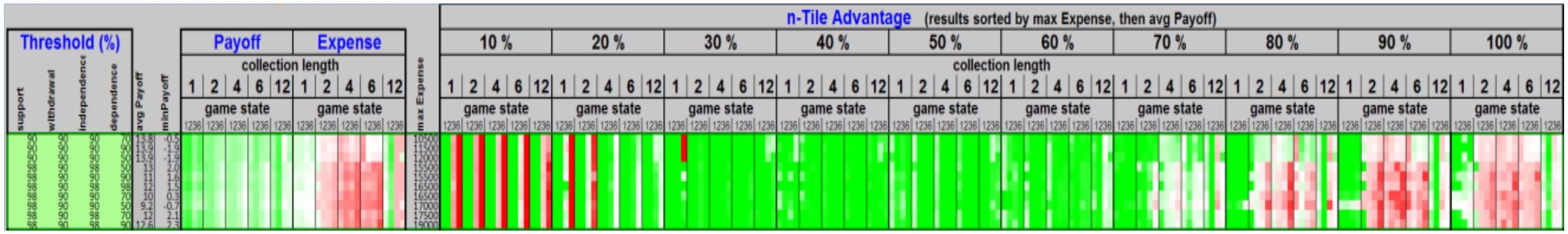

Threshold (%)
support
withdrawal
independence
dependence
avg Payoff
minPayoff
Payoff
Expense
collection length
1 2 4 6 12
game state
1236
max Expense
n-Tile Advantage (results sorted by max Expense, then avg Payoff)
10 %
20 %
30 %
40 %
50 %
60 %
70 %
80 %
90 %
100 %

Table 26 reveals some clearer groupings, which have been highlighted. These groups are examined more closely in Table 27 through Table 29. Recall that in the tertiary phase of learning, the results reflect the ability of the TBL-CLA to recover it performance after the game has suddenly changed from one state to another. In the footprints, the column for game state 1 records the results for the transition from game state 1 to game state 2, the column for game state 2 contains the results for the transition from state 2 to state 3, the column for state 3 contains the results for the transition from state 3 to state 6, and the column for game state 6 contains the results for the transition from state 6 back to state 1.

The best results from Experiment 5 are those TBL 4-tuples with the withdrawal threshold is set at 90% and the support and independence thresholds are set equal to each other at 90% or 98%. In Experiments 1 and 3, the environment was stationary and so the ideal learning curve is one that only has a positive slope. In Experiment 5, the environment is not stationary and; therefore, even the ideal learning curve must have some period of a negative slope as the learner unlearns old responses and relearns the new correct ones. In Experiments 1 and 3, the best thresholds were set near 70% anticipating that the learner's selection confidence starts low and should only become higher. In this experiment, in the tertiary phase the learner's selection confidence should start high and fall as a reaction to the influx of negative feedback that the learner will have to encounter as it learns that some of its old responses do not work anymore. By setting the TBL-thresholds higher in a task-switching environment, the TBL-CLA is able to withdraw support from those tactics that should no longer be used and have some of its stimulants become followers of tactics again to make the "course correction".

If these thresholds are set too high, the TBL-CLA becomes overly sensitive and if they are set too low, the TBL-CLA does not get to take advantage of using follower stimulants because the stimulants will have relearned their correct respondents by simply using the Standard selection policy. The worst results come from TBL 4-tuples settings with the withdrawal threshold set to 50%. In these cases, there is no possibility of the stimulants ever returning to follower status. Individual cases are discussed in greater detail in Experiment 6.

Table 27: Best results from Experiment 5, sorted by the maximum Expense value in each row.

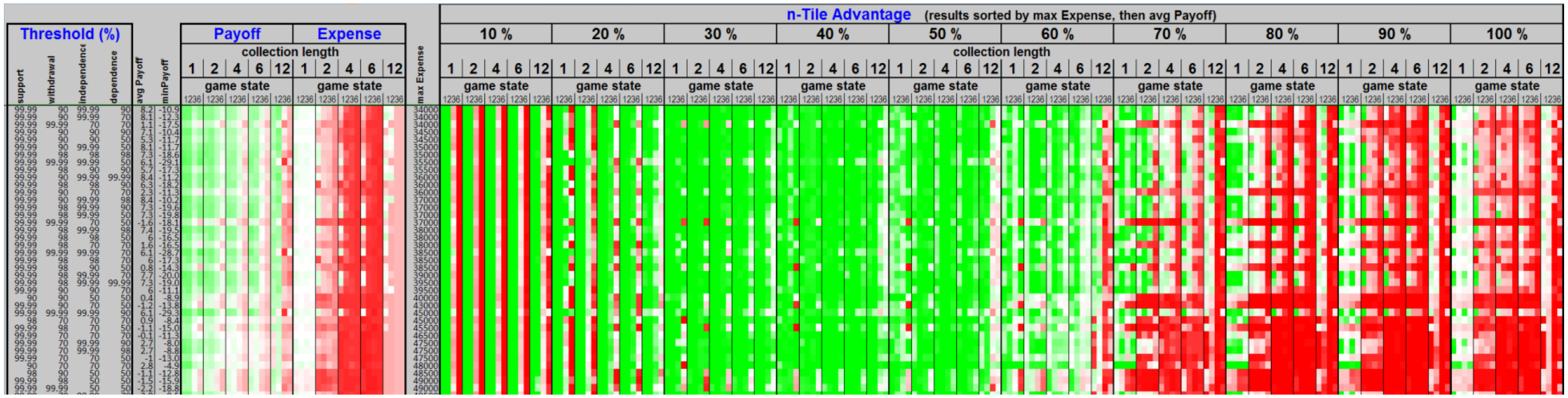


Threshold (%) support	withdrawal	independence	dependence	avg Payoff	minPayoff	max Expense
90	90	90	70	13.8	-0.5	10500
90	90	90	90	13.9	-1.9	11500
90	90	90	50	13.9	-1.9	12000
98	90	98	50	13	2.0	15500
98	90	90	90	11	1.6	15500
98	90	98	98	12	1.5	16500
98	90	90	70	10	0.3	16500
98	90	90	50	9.2	-0.7	17000
98	90	98	70	12	2.1	17500
98	90	98	90	12.6	2.3	19000

Payoff / Expense: collection length 1, 2, 4, 6, 12; game state 1236

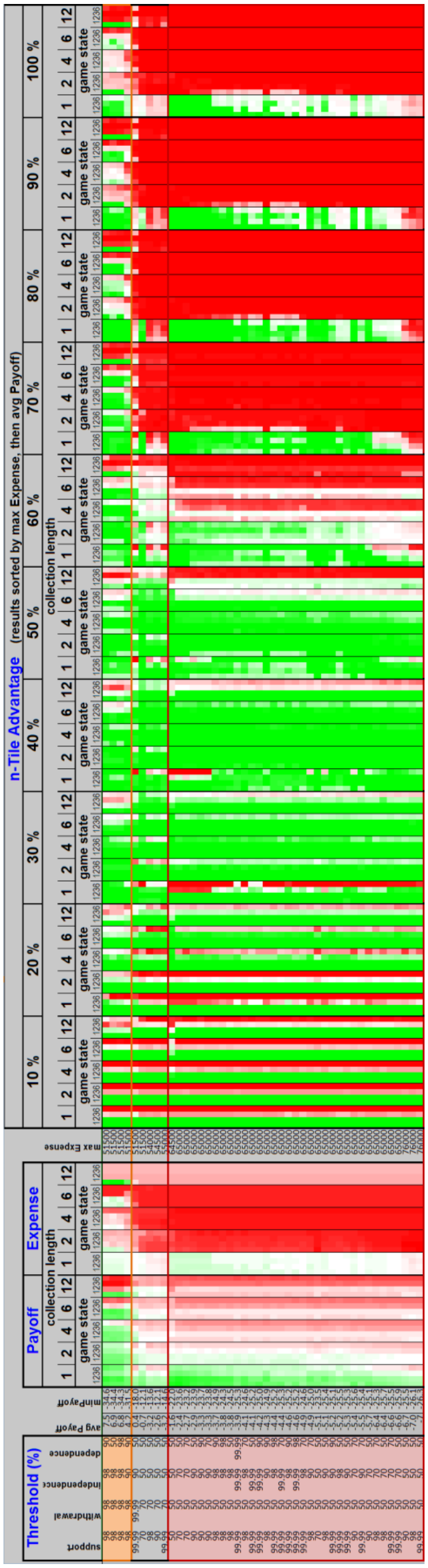
n-Tile Advantage (results sorted by max Expense, then avg Payoff): 10 %, 20 %, 30 %, 40 %, 50 %, 60 %, 70 %, 80 %, 90 %, 100 %; collection length 1, 2, 4, 6, 12; game state 1236

Table 28: Average results from Experiment 5 sorted by the maximum Expense in each row.


Threshold (%)
Payoff
Expense
collection length
1 | 2 | 4 | 6 | 12
game state
1236
n-Tile Advantage (results sorted by max Expense, then avg Payoff)
10 % | 20 % | 30 % | 40 % | 50 % | 60 % | 70 % | 80 % | 90 % | 100 %

support	withdrawal	independence	dependence	avg Payoff	minPayoff	max Expense
99.99	90	99.99	90	8.2	-10.9	34000
99.99	90	99.99	70	8.1	-12.3	34000
99.99	99.99	70	70	1.1	-17.5	34000
99.99	90	90	90	7.1	-10.4	34500
99.99	90	90	50	5.3	-11.7	34500
99.99	90	99.99	50	8.1	-11.7	35000
99.99	98	98	98	7.3	-18.6	35000
99.99	99.99	99.99	50	6.1	-29.1	35500
99.99	98	90	90	5.7	-17.3	35500
99.99	90	99.99	99.99	8.4	-11.2	36000
99.99	98	98	90	6.3	-18.2	36000
99.99	90	70	70	2.3	-11.3	36000
99.99	90	99.99	98	8.4	-10.2	37000
99.99	98	99.99	90	7.3	-19.6	37000
99.99	98	99.99	50	7.3	-19.8	37000
99.99	99.99	70	50	-1.6	-18.1	37000
99.99	98	99.99	98	7.4	-19.5	38000
99.99	98	98	50	6	-16.5	38000
99.99	98	70	70	1.6	-16.5	38000
99.99	99.99	99.99	70	6.1	-28.7	38500
99.99	98	98	70	6	-17.3	38500
99.99	98	90	50	0.8	-14.3	38500
99.99	98	99.99	70	7.7	-20.0	39000
99.99	98	99.99	99.99	7.3	-19.0	39500
99.99	90	90	70	6	-11.1	39500
90	90	50	50	0.4	-8.9	40000
99.99	90	70	50	-1.2	-13.8	43000
99.99	99.99	99.99	90	6.1	-29.3	45000
98	70	70	70	0.9	-8.4	45000
99.99	98	70	50	-1.1	-15.0	45500
99.99	70	70	70	-0.1	-11.3	46500
99.99	70	99.99	90	2.7	-8.0	47500
99.99	70	99.99	98	2.7	-8.8	47500
99.99	70	70	50	-1	-13.0	47500
90	70	70	70	2.8	-4.9	48000
98	90	50	50	-1.1	-12.8	48500
99.99	98	50	50	-1.5	-15.9	49000
99.99	99.99	50	50	-2.2	-18.8	49000

Table 29: The worst results from Experiment 5 sorted by the maximum Expense in each row.



5.6.2 Formal Conclusions for Experiment 5

A summary of the conclusions drawn from Experiment 5 is presented here in summary form. For a more detailed discussion of these conclusions, see the previous section. All conclusions presented in this section are only valid for the TruthTable game in an environment with *one target per output* and that *is task-switching, deterministic, and correct* for the duration of the learning process. The experimental results and conclusions are only relevant to the tertiary phase of the game, and are focus on the recovery of learning performance. Informal conclusions, speculations, and predictions about the performance of the application of TBL to other games, including actual-life games, will be presented and discussed in Section 5.8. Suggestions for future research are presented and discussed in Section 5.9.

Conclusion 6: The learning behavior of a TBL-CLA is significantly affected by the settings of the TBL thresholds.

Conclusion 6a: For best performance, set the withdrawal threshold at 90% and the support and independence equal to each other at values between 90 and 98%.

Conclusion 6b: For worst-case performance, set the support threshold to 99.99% and the withdrawal threshold to 50%.

Conclusion 7: When the TBL thresholds are at their optimal settings, the TBL-CLA significantly outperforms the Standard-CLA at no statistically significant Expense across all collection lengths and TruthTable game states. This implies that TBL is a useful strategy for learning and recovering learning performance in a changing environment.

5.7 Experiment 6

Experiment 6 accomplishes Goal 3.3, described in Section 3.5. The calculation of the individual performance measures is completed with the use of a spreadsheet, but the process of inspecting the results and drawing conclusions about each case requires significant attention from a person. Table 16, taken from Section 4.1.5.6, below presents the block design for Experiment 6.

Table 16: Design for Experiment 6 [from 4.1.5.6]

Name	Factor values	Treatments
TruthTable game states	{1, 2, 3, 6}	4
Collection length, *c*	{1, 2, 4, 6, 12}	5
TBL threshold 4-tuples <($\kappa_{s,}$ $\kappa_{w,}\kappa_{i,}$ κ_d)>	(90.00, 90.00, 90.00, 70.00) (99.99, 99.99, 70.00, 70.00) (99.99, 50.00, 50.00. 50.00)	3
Experiment Resource Requirements		
	Total treatments	**80***
	Estimated time per treatment	10 person-minutes
	Total time required	**14 person hours**
	* total number of treatments calculated as follows: Standard-CLA treatments = 4(5) TBL-CLA treatments = 4(5)(3) Total = Standard-CLA treatments + TBL-CLA treatments = 80	

5.7.1 Results

This section presents selected results from Experiment 6. The full set of reduced results is available in the digital appendices as interactive Microsoft Excel spreadsheets.

5.7.2 Best Performance: 4-tuple (κ_s=90, κ_w=90, κ_i=90, κ_d=70)

It was observed in Experiment 5 that the 4-tuple of (90, 90, 90, 70) was the best performing combination of TBL thresholds. The results of close inspection of the behavior of a TBL-CLA with these settings are presented in this section. As the TBL thresholds are fixed, the purpose of this experiment is to understand the influence that the thresholds have on the behavior of the CLA under varying environmental conditions: the collection length and the TruthTable game state.

5.7.2.1 TruthTable game state 1 to 2, Collection Length 12

At a collection length of 12, both CLAs face the most difficult challenge because the signal-to-noise ratio is at its lowest. The problem is compounded by the way in which the game changes. When the TruthTable game state changes from state 1 to state 2, 50% of all of the inputs are reassigned to new target outputs. The TBL-CLA needs to discover this new target response and identify it as a tactic. This takes time, during which some of the stimulants cross the support threshold. When the new tactic is identified, there is a smaller number of tactics that can become followers of it. Figure 45 shows that the TBL-CLA does not score significantly better than the Standard-CLA, but it does reach the termination condition before the Standard-CLA. Figure 46 shows that the follower stimulants appear later in the learning process and Figure 47 shows that the TBL-CLA does not make significantly fewer random selections than the Standard-CLA.

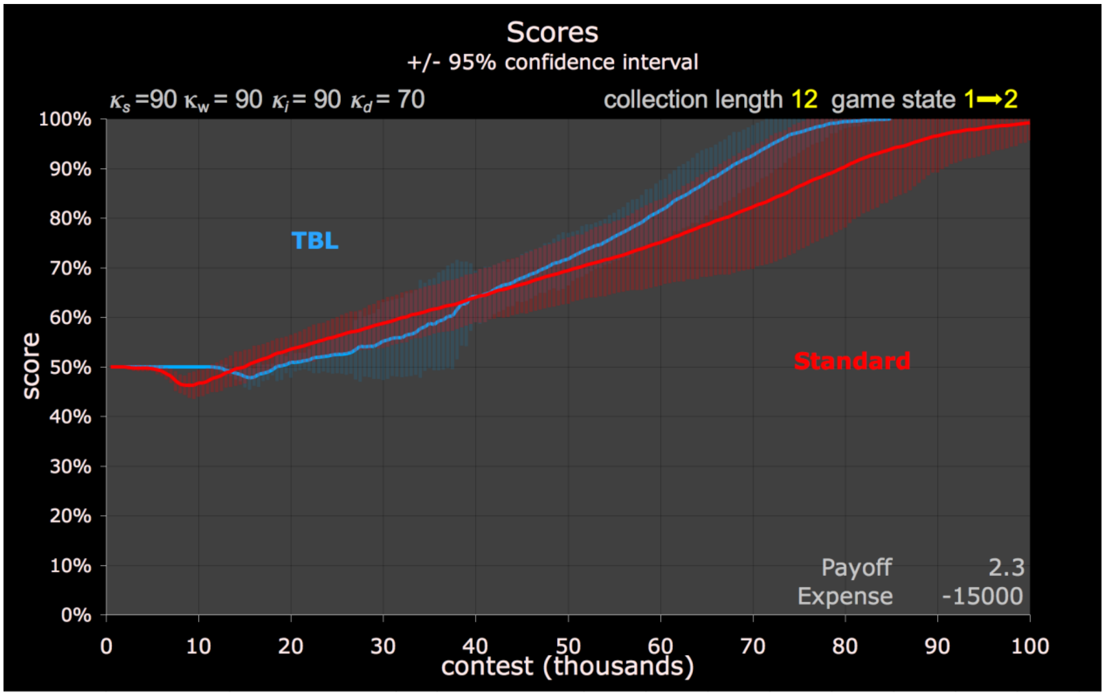


Figure 45: At a collection length of 12 and a 50% change in the game, the TBL-CLA garners a small Payoff, but earns a very low Expense

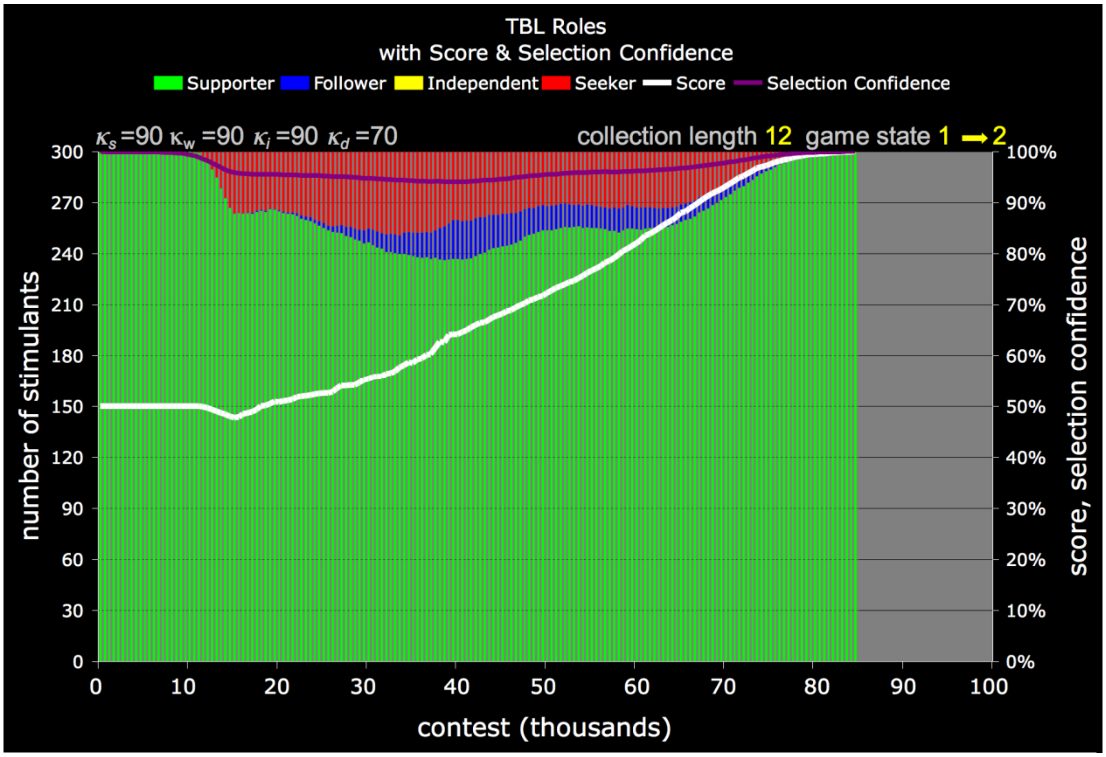


Figure 46: Because the CLA starts the tertiary phase with a high selection confidence, it takes time before the first followers appear in the system.

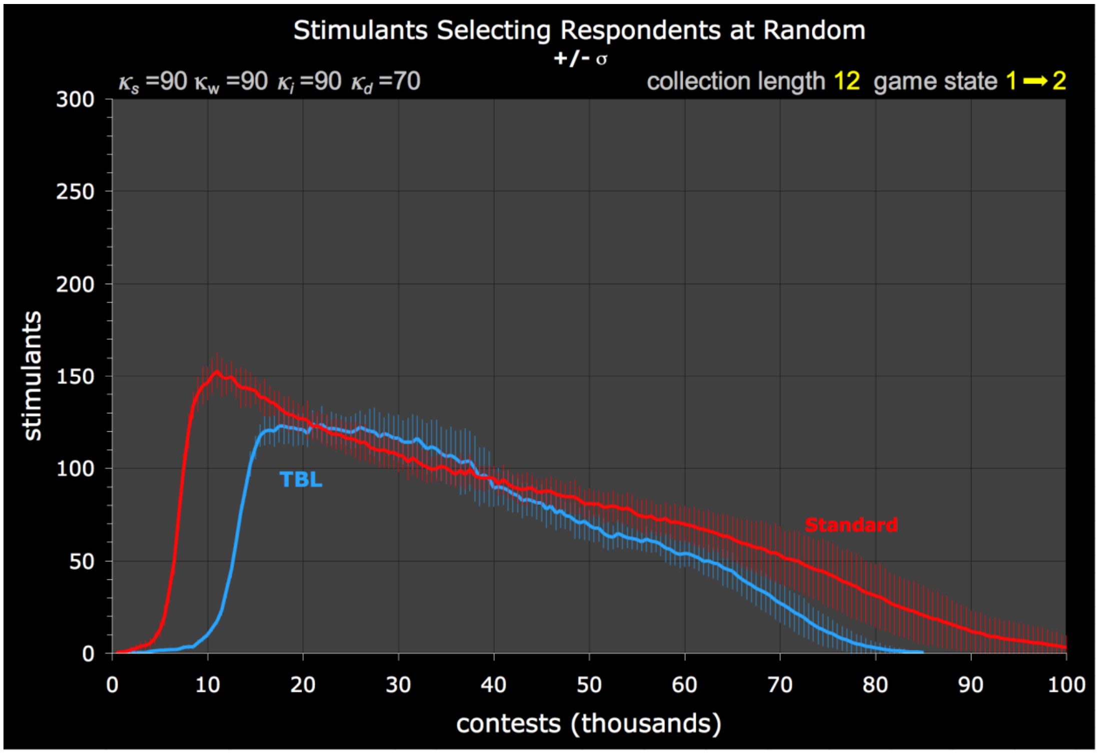


Figure 47: Even though the TBL-CLA has fewer stimulants selecting respondents at random early in the phase, the TBL-CLA's score is not significantly different from the Standard-CLA's score. This implies that the TBL-CLA is entrenched in its behavior and needs more time to unlearn old patterns.

5.7.2.2 TruthTable game state 2 to 3, Collection Length 12

When the TruthTable game state changes from state 2 to state 3, it is also the case that 50% of the inputs are reassigned to new target; however, Figure 48 shows that in this case, the TBL-CLA is able to score significantly better than the Standard-CLA. The advantage comes from the fact that when the game state changes from state 2 to state 3 one third of the inputs which are reassigned are reassigned to target responses that were already in use in state 2 (see Section 4.1.5.1 for more details). Because some of the reassigned inputs to target responses that the TBL-CLA has identified as tactics, these inputs can be followers earlier and take advantage of this guidance sooner than the other stimulants that must wait for the new tactic to be identified. Figure 49 demonstrates the fact that follower stimulants appear earlier, and Figure 50 shows that the TBL-CLA makes significantly fewer random selections.

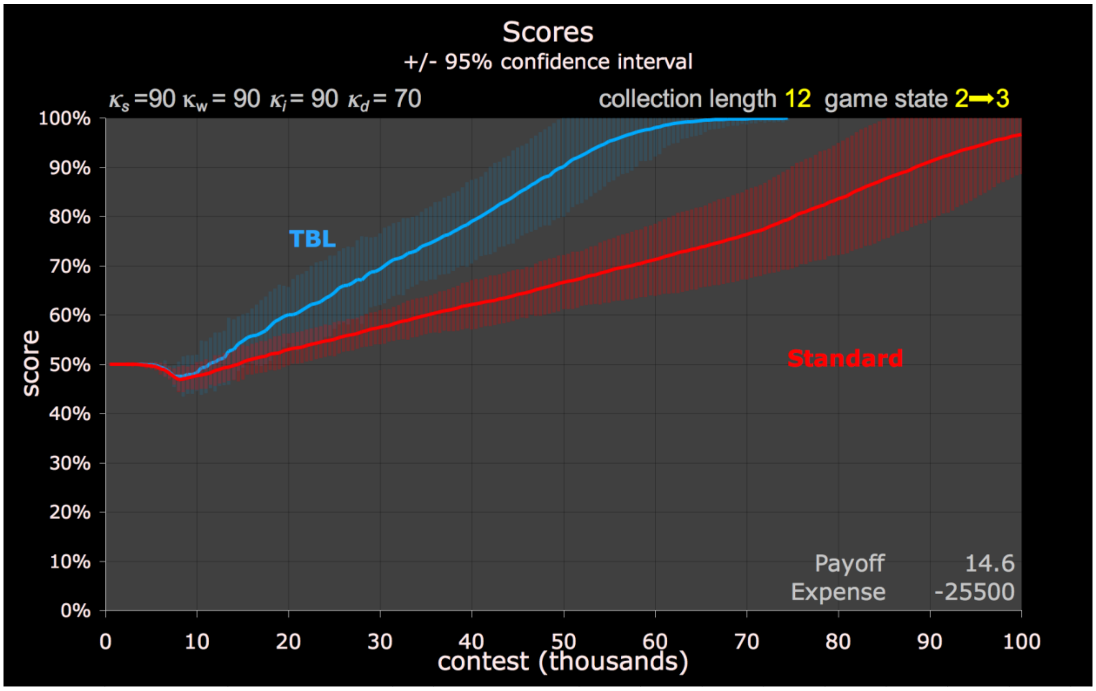


Figure 48: When the game changes from state 2 to state 3, it is also a 50% change. The difference in performance comes from the fact that 1/3 of the changed inputs are reassigned to a target response that was in use in the previous stage.

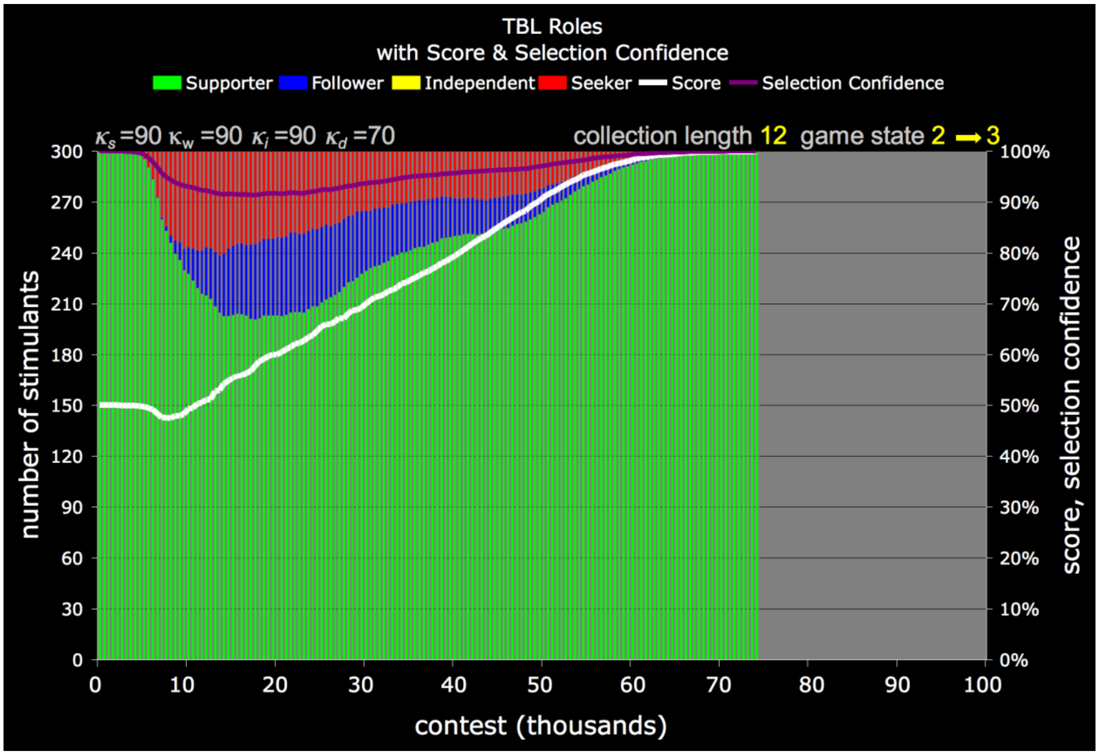


Figure 50: Some of the stimulants that are reassigned can become followers of an existing tactic early in the phase, instead of having to wait until a new tactic is discovered.

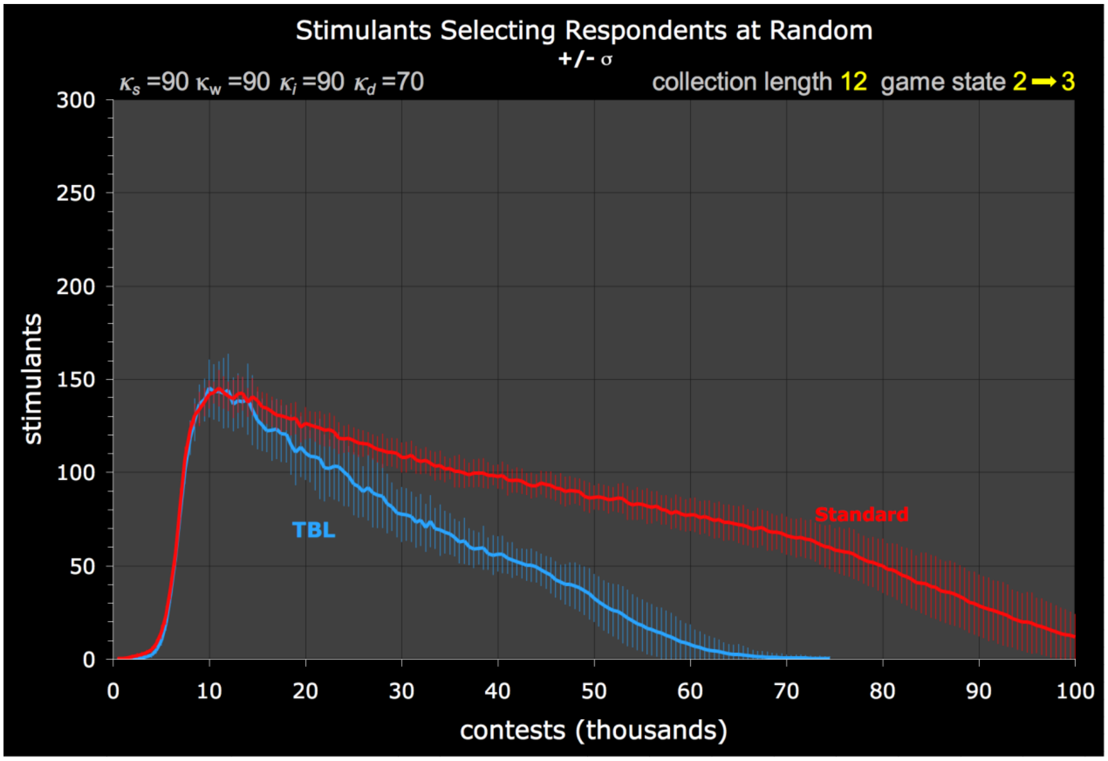


Figure 49: The TBL-CLA has few stimulants making random selections, which corresponds with the use of tactics.

5.7.2.3 TruthTable game state 6 to 1, Collection Length 12

Similarly to the transition from TruthTable game state 2 to state 3, the transition from state 6 to state 1 involves reassigning 83% of the inputs to new targets (see Section 4.1.5.1 for details). Moving from 6 target responses to one target response means that the TBL-CLA does not have to identify any new tactics and can begin immediately using an existing tactic. Figure 51 shows that the TBL-CLA scores significantly better than the Standard-CLA. Figure 52 shows that the TBL-CLA is able to have a large number of follower stimulants fairly early in the learning process. Figure 53 shows that the TBL-CLA has significantly few stimulants that are selecting responses at random.

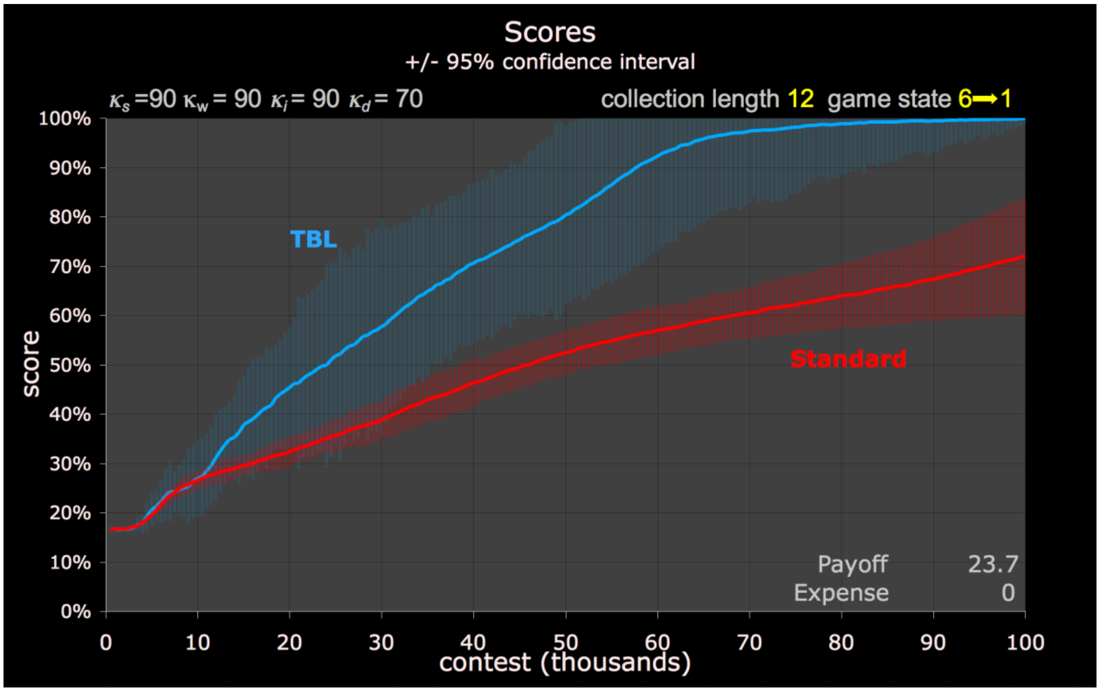


Figure 51: When the game state changes from 6 to 1, 83% of the inputs are reassigned to new target respondents; however, the new reassigned inputs are assigned to the same target response, which was already in use in the previous phase. This gives the TBL-CLA a significant advantage.

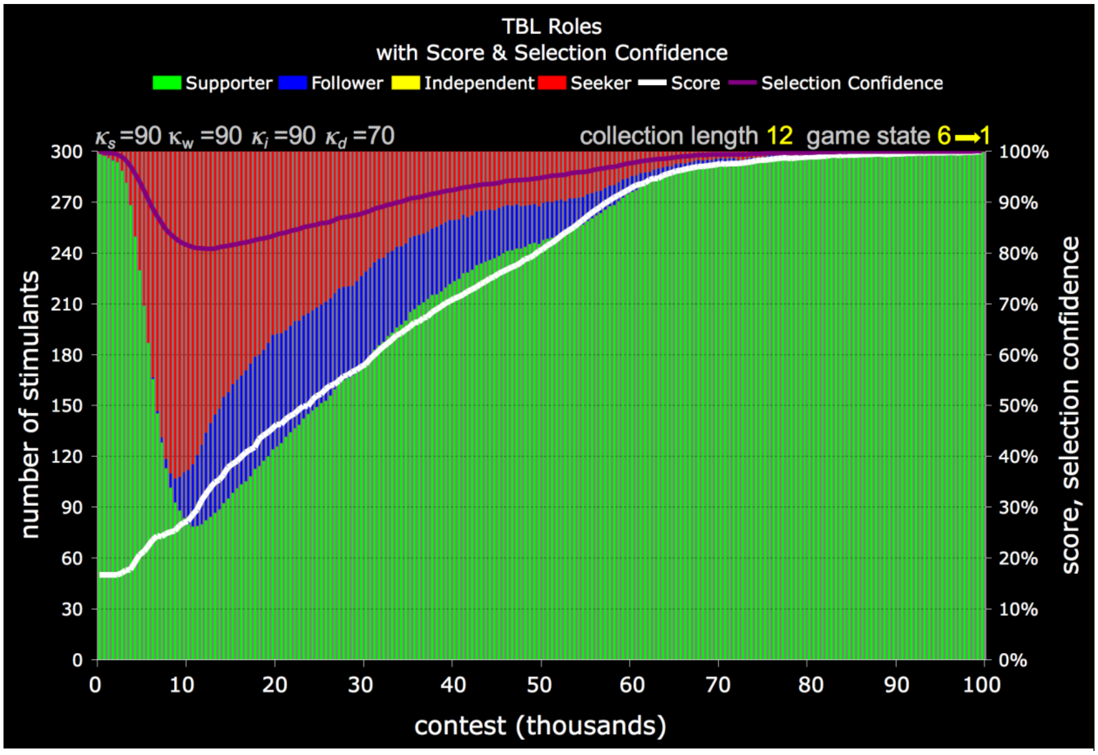


Figure 53: The TBL threshold settings prevent the TBL-CLA from using independent stimulants. The TBL-CLA has many follower stimulants which help to boost the scores.

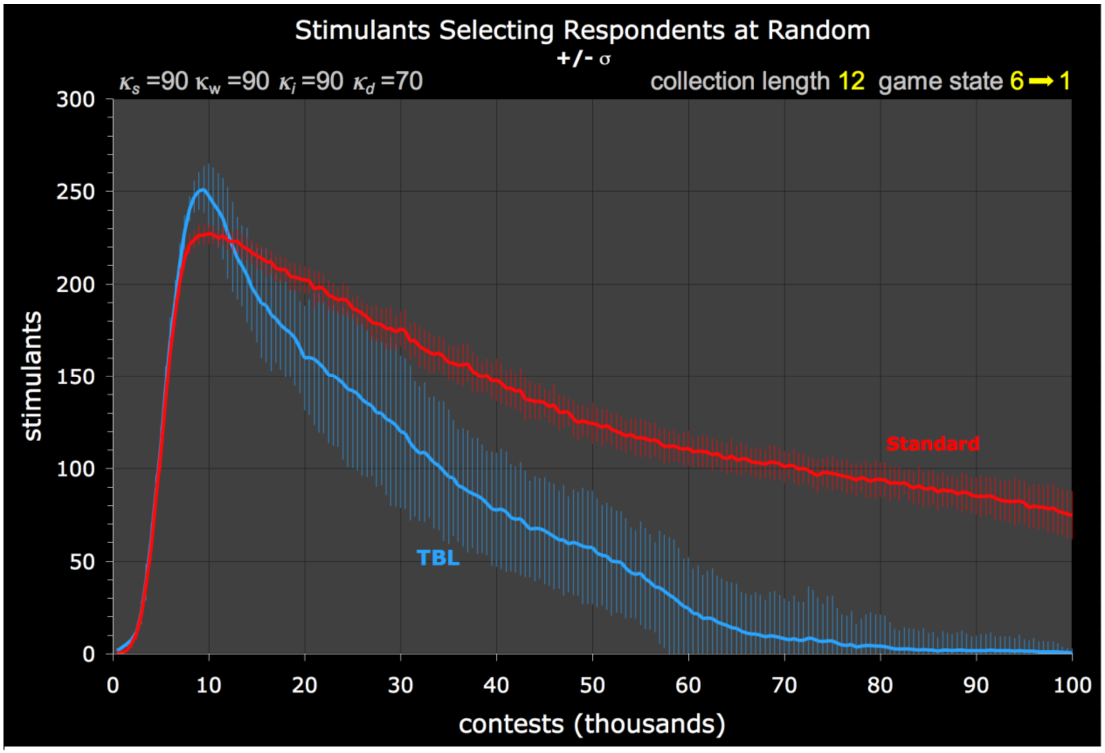


Figure 52: Despite the larger variance, the TBL-CLA has significantly fewer stimulants selecting random respondents than the Standard-CLA.

5.7.3 Average Performance: 4-tuple (κ_s=99.99, κ_w=99.99, κ_i=70, κ_d=70)

It was observed in Experiment 5 that the 4-tuple of (99.99, 99.99, 70, 70) was an average performing combination of TBL thresholds. The results of close inspection of the behavior of a TBL-CLA with these settings are presented in this section. As the TBL thresholds are fixed, the purpose of this experiment is to understand the influence that the thresholds have on the behavior of the CLA under varying environmental conditions: the collection length and the TruthTable game state.

5.7.3.1 TruthTable game state 3 to 6, Collection Length 6

In the average case, the TBL-CLA's recovery performance is better with a shorter collection length. Figure 54 shows that the TBL-CLA scores significantly better than the Standard-CLA and that it reaches the termination conditions before the Standard-CLA. Figure 55 shows that the TBL-CLA is able to make use of tactics early in the learning process. Figure 56 shows that the TBL-CLA has fewer stimulants that select respondents at random, which corresponds to the increase in follower stimulants.

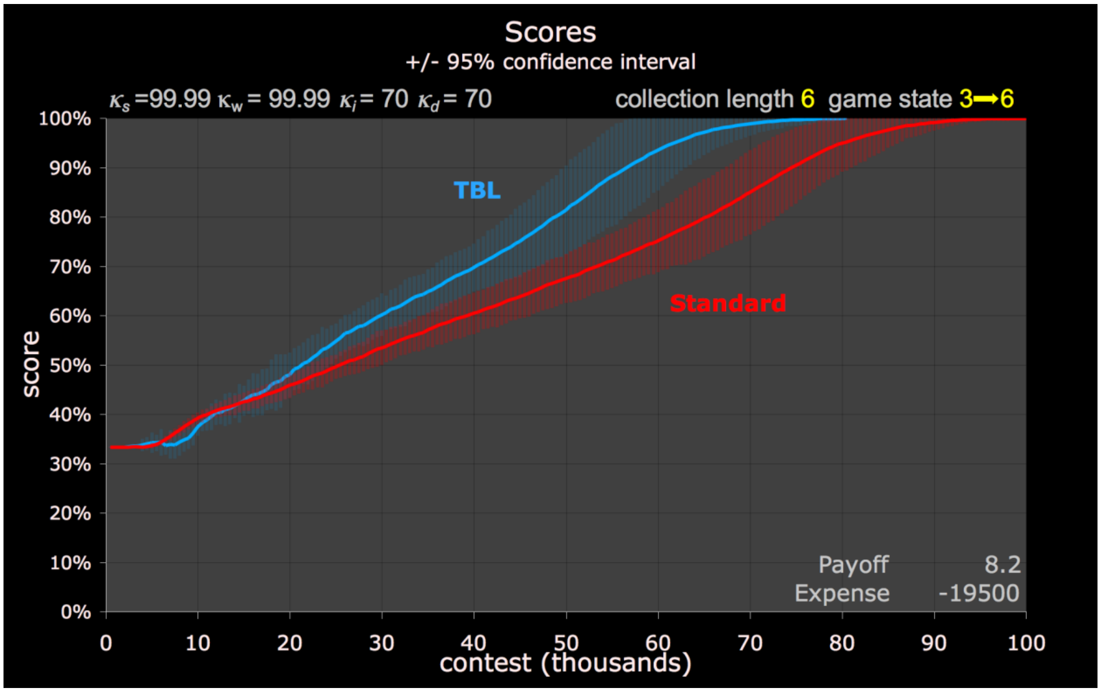


Figure 54: When the environment provides a shortened collection length, the TBL-CLA can regain its advantage, even though the state presents more reassigned inputs.

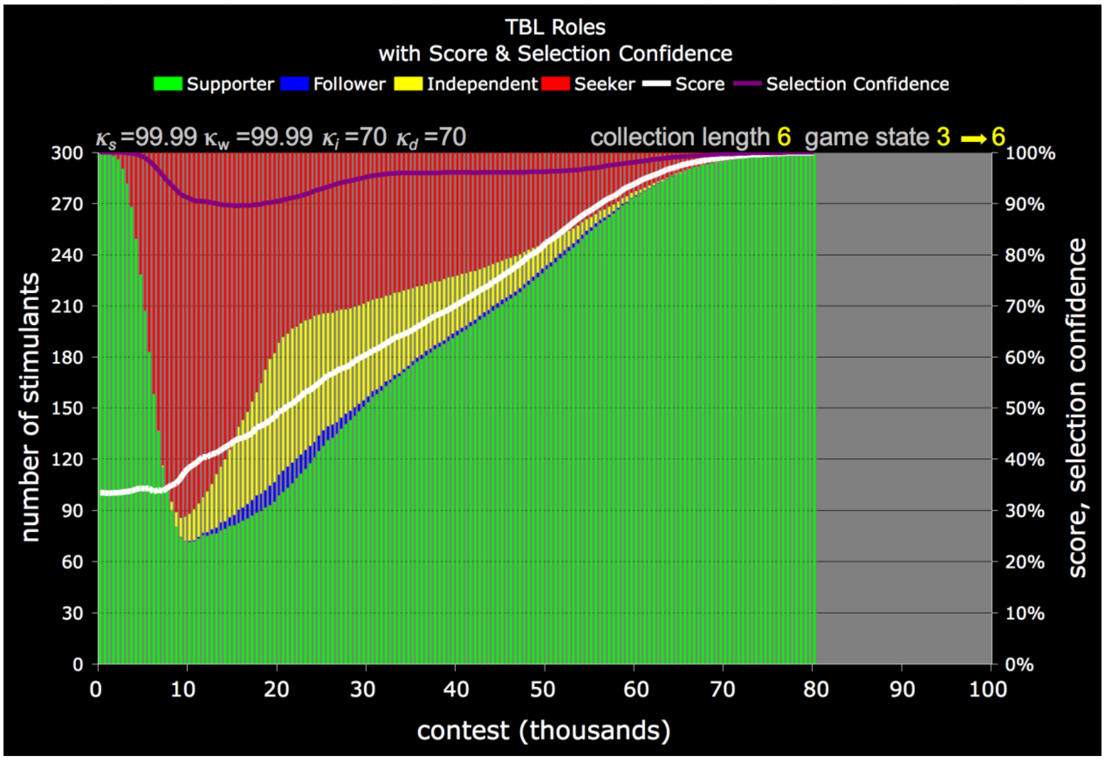


Figure 55: The TBL-CLA is able to leverage the follower stimulants to improve its score.

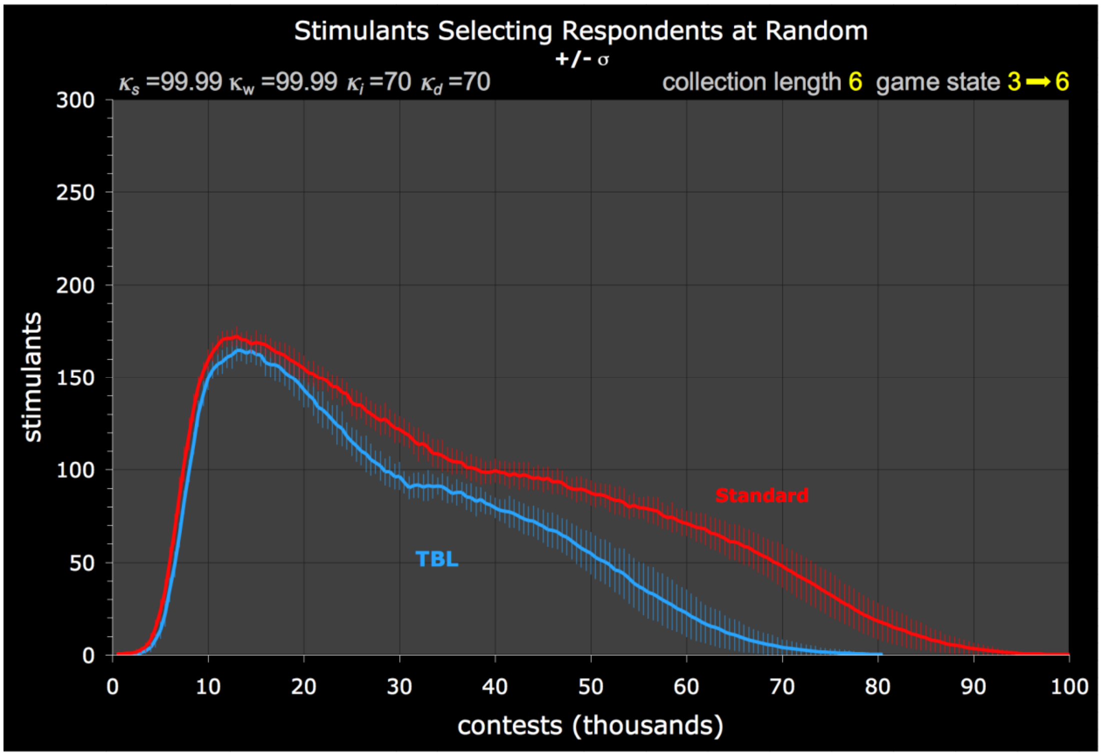


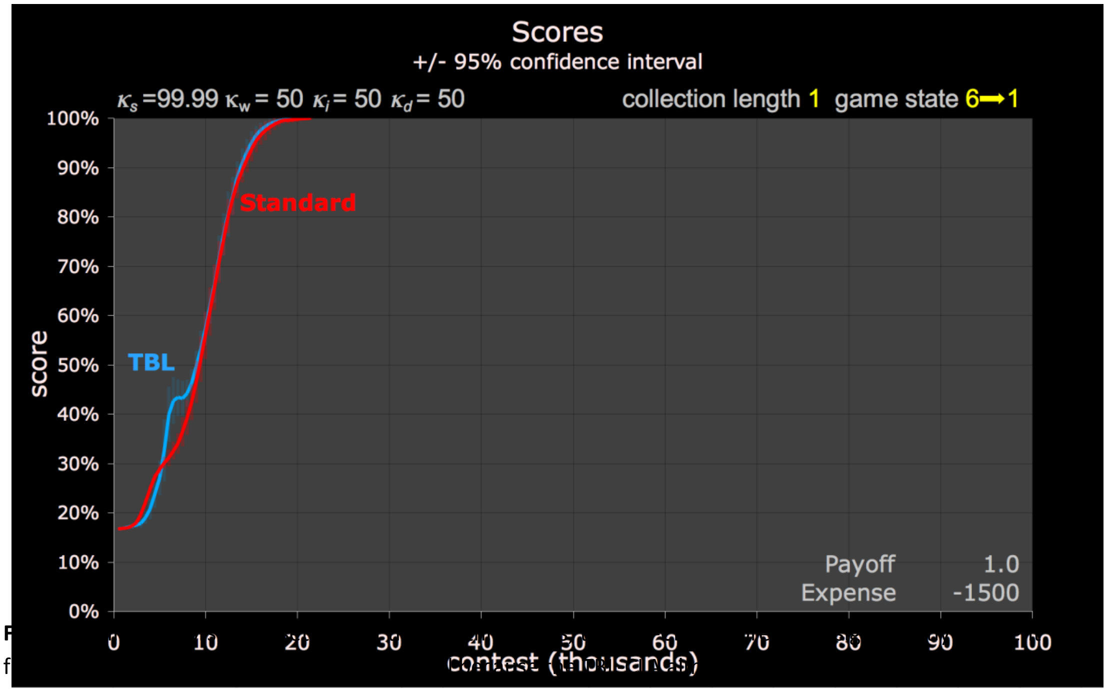
Figure 56: The TBL-CLA makes significantly few random selections than the Standard-CLA during the later part of the learning process.

5.7.4 Poor Performance: 4-tuple (κ_s=99.99, κ_w=50, κ_i=50, κ_d=50)

It was observed in Experiment 5 that the 4-tuple of (99.99, 50, 50, 50) was the worst performing combination of TBL thresholds. The results of close inspection of the behavior of a TBL-CLA with these settings are presented in this section. As the TBL thresholds are fixed, the purpose of this experiment is to understand the influence that the thresholds have on the behavior of the CLA under varying environmental conditions: the collection length and the TruthTable game state.

5.7.4.1 TruthTable game state 6 to 1, Collection Length 1

The worst-case settings of the TBL thresholds are such that the TBL-CLA behaves like a Standard-CLA even at a collection length of one (see Figure 57). This is a very unusual occurrence; generally, at a collection length of one the TBL-CLA shows some advantage. In this case, the TBL thresholds are set such that once a stimulant has become a supporter, the only way it can become independent is if it acquires more than one primary respondent. This only happens in a few instances and none of the stimulants ever become followers (see Figure 48). As a consequence, the TBL-CLA makes has as many stimulants selecting respondents randomly as the Standard-CLA (see Figure 49).



F

f

Standard-CLA with a collection length of one.

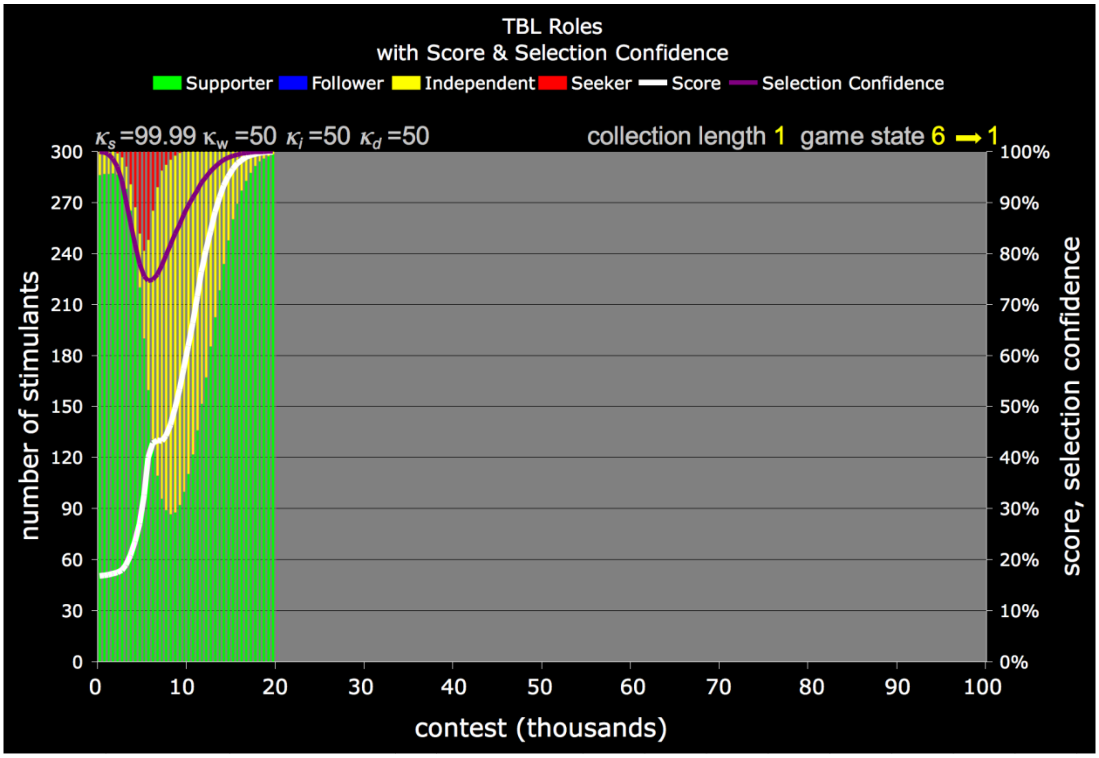


Figure 59 When the withdrawal threshold is set at 50%, the only stimulants that are able to become independent are those stimulants that withdraw their support from a tactic because they have more than one primary respondent.

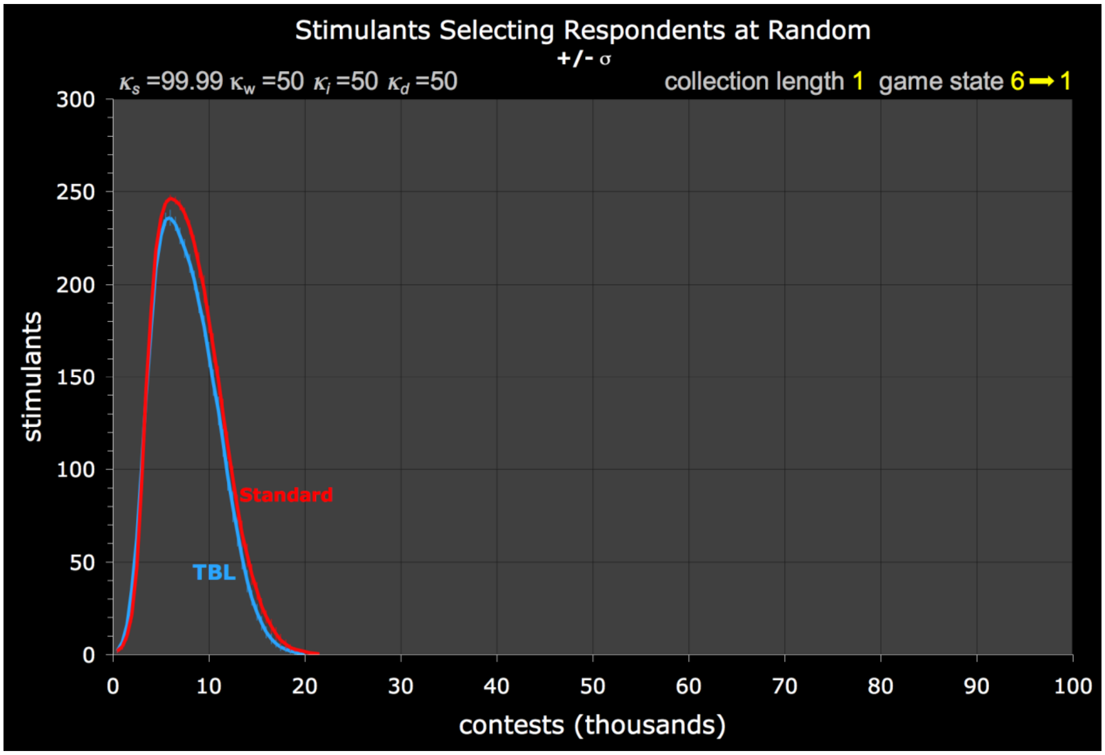


Figure 58: There is no significant difference between the number of random selections being made by the TBL-CLA and the Standard-CLA.

5.7.4.2 TruthTable game state 6 to 1, Collection Length 2

Figure 60 shows that at a collection length of two, the TBL-CLA starts to perform significantly worse than the Standard-CLA. As the collection length gets longer, the updates become more diffuse and the signal-to-noise ratio goes down, causing the TBL-CLA to take longer to correct itself. Figure 61 shows that very few stimulants are able to even change roles, and Figure 62 shows that the TBL-CLA makes significantly more random selections later in the learning process.

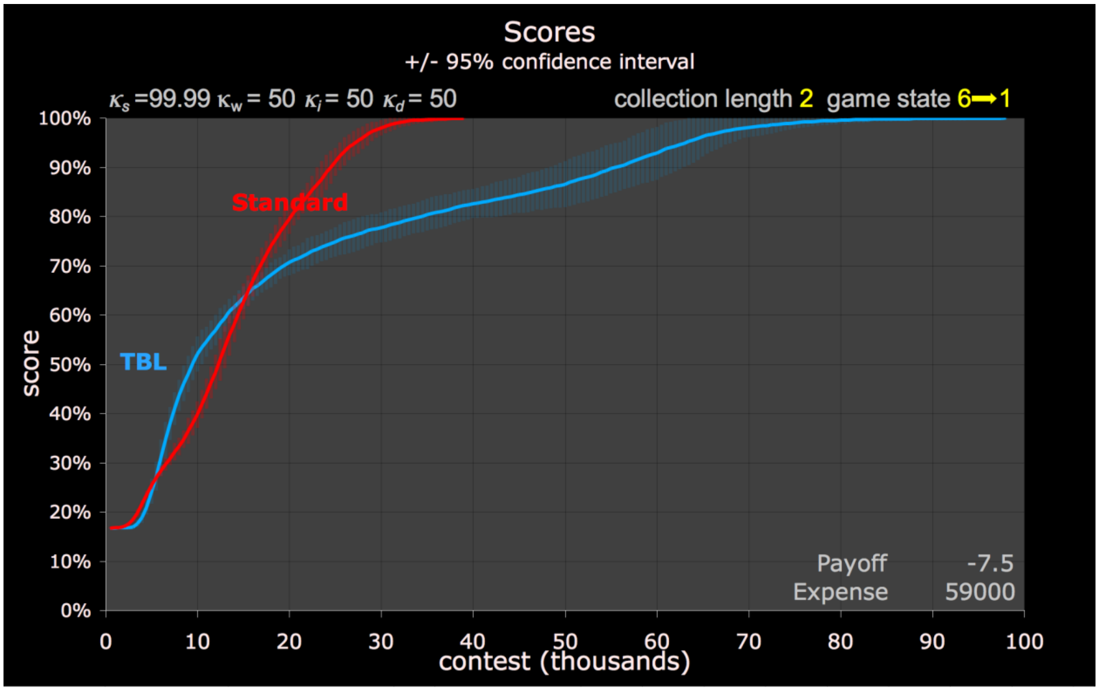


Figure 60: Even a collection length of two, the TBL-CLA significantly underperforms the Standard-CLA.

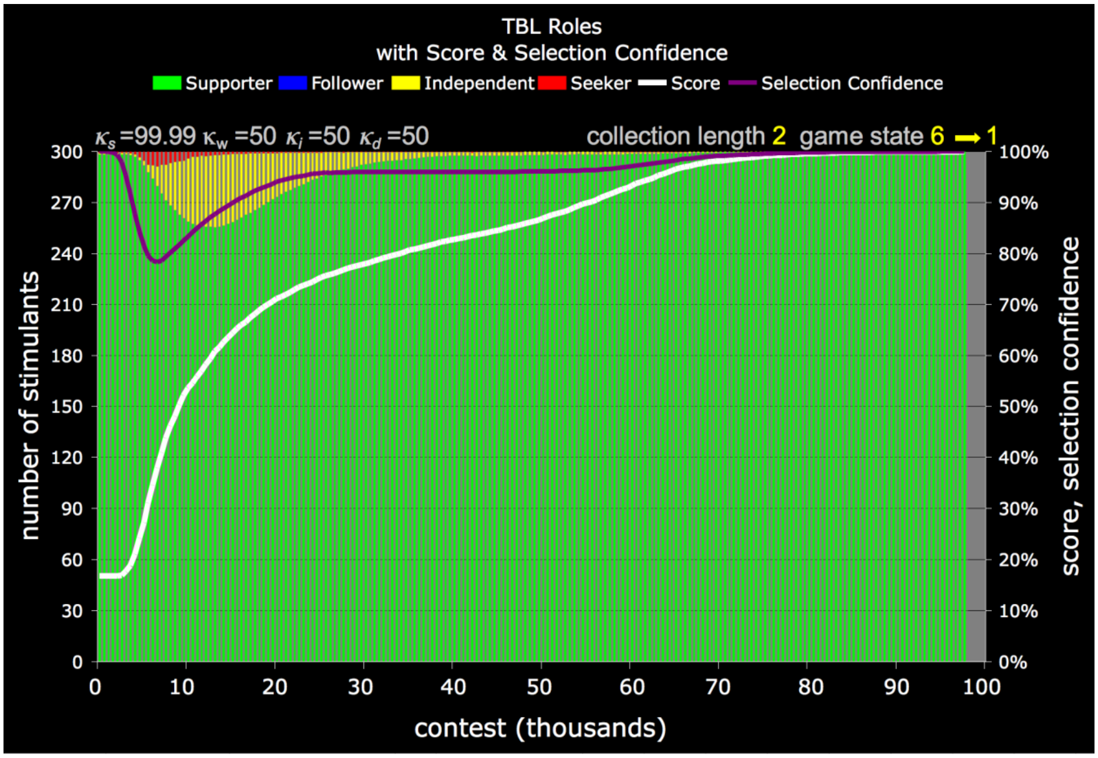


Figure 62: With the withdrawal, independence, and dependence thresholds all set at 50%, almost all of the stimulants remain in the role of supporter.

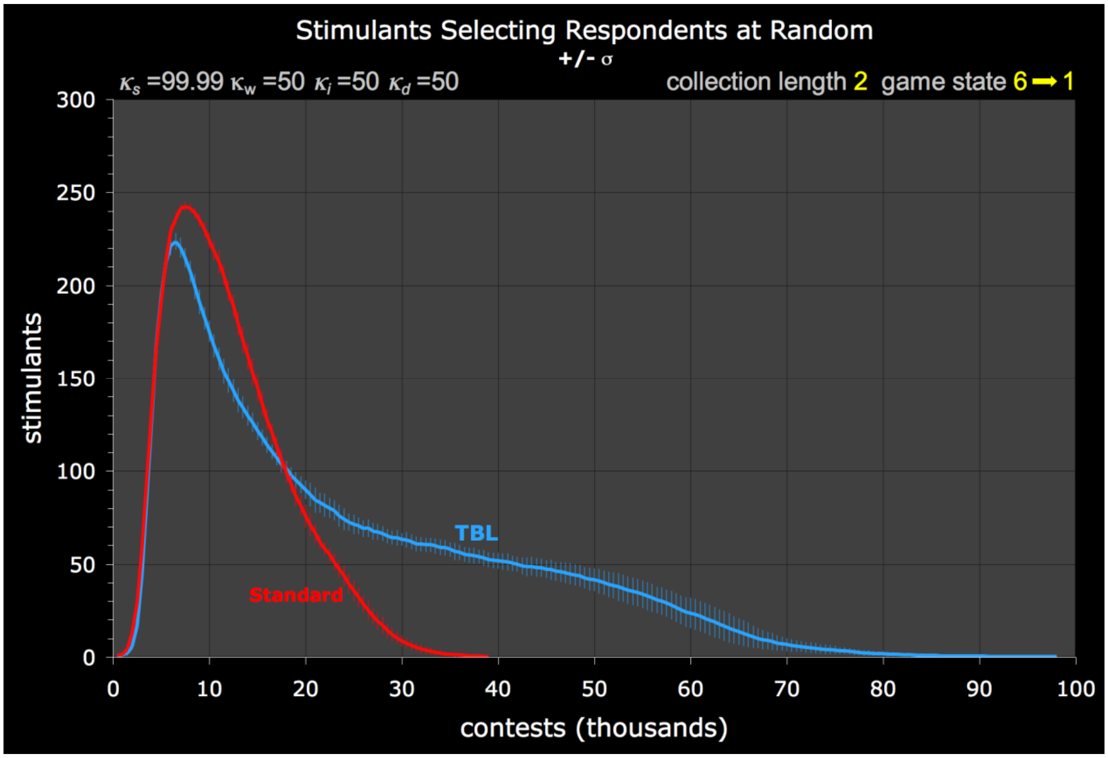


Figure 61: The TBL-CLA makes significantly more random selections than the Standard-CLA.

5.7.5 Formal Conclusion from Experiment 6

A summary of the conclusions drawn from Experiment 6 is presented here in summary form. For a more detailed discussion of these conclusions, see the previous section. All conclusions presented in this section are only valid for the TruthTable game in an environment with *one target per output* and that *is task-switching, deterministic, and correct* for the duration of the learning process. The experimental results and conclusions are only relevant to the tertiary phase of the game, and are focus on the recovery of learning performance. Informal conclusions, speculations, and predictions about the performance of the application of TBL to other games, including actual-life games, will be presented and discussed in Section 5.8. Suggestions for future research are presented and discussed in Section 5.9.

Conclusion 8: The effectiveness of Tactic-Based Learning to aide in recovery of learning behavior after a change in the environment is influenced more by the number of reassigned stimulants that can become followers of existing tactics than it is influenced by the percentage of stimulants that change or the number of new tactics that must also be identified.

Conclusion 9: When the TBL-CLA is able to reduce the number of stimulants that select their respondents at random, its performance improves.

5.8 Summary of Formal Conclusions

This section brings together all the formal conclusions that have been drawn from this research. The section number of the corresponding experiment from which the conclusion is drawn is indicated after each conclusion. All formal conclusions are only valid for the TruthTable game.

Conclusion 1: The learning behavior of a TBL-CLA in a *stationary, deterministic, and correct* environment with *one target cell per input* is significantly affected by the settings of the TBL thresholds. (Section 5.2)

Conclusion 1a: For best performance in a *stationary, deterministic, and correct* environment with *one target cell per input*, set the support, withdrawal, and independence thresholds equal to each other and at values between 70% and 80%. (Section 5.2)

Conclusion 1b: A TBL-CLA will behave like a Standard-CLA when the support and withdrawal thresholds are set high, greater than 90%, and the independence and dependence thresholds are set at or near the minimum (less than or equal to 55%) in a *stationary, deterministic, and correct* environment with *one target cell per input*. (Section 5.2)

Conclusion 1c: Even TBL 4-tuples that do not produce optimal or Standard-like behavior in a *stationary, deterministic, and correct* environment with *one target cell per input* provide significant advantages early in the learning process; however, the advantages are lost later on as the TBL-CLA significantly underperforms compared to the Standard-CLA. (Section 5.2)

Conclusion 1d: In a *stationary, deterministic, and correct* environment with *one target cell per input* a TBL-CLA performance improves when the independent role is removed by setting the support and independence thresholds equal to each other. (Section 5.3)

Conclusion 2: TBL-CLA performance improves as the TBL advantage, TBL$_{\alpha}$, of the TruthTable game state is increased in a *stationary, deterministic, and correct* environment with *one target cell per input*. (Section 5.2)

Conclusion 3: When the TBL thresholds are set for good performance in a *stationary, deterministic, and correct* environment with *one target cell per input*, the TBL-CLA has significantly fewer stimulants that make random selections than the Standard-CLA, meeting one of the performance criteria for this research. (Section 5.3)

Conclusion 4: Only a small percentage of the total stimulants need be follower stimulants for a TBL-CLA to perform significantly better than a Standard-CLA in a *stationary, deterministic, and correct* environment with *one target cell per input*. (Section 5.3)

Conclusion 5: The learning behavior of a TBL-CLA in a *stationary, deterministic, and correct* environment with *two target cells per input* is significantly affected by the settings of the TBL thresholds. (Section 5.4)

Conclusion 5a: For best performance of a TBL-CLA in a *stationary, deterministic, and correct* environment with *two target cells per input*, set the support threshold at 70%. (Section 5.4)

Conclusion 5b: For strong performance of a TBL-CLA in a *stationary, deterministic, and correct* environment with *two target cells per input*, set the support threshold at 90%. (Section 5.4)

Conclusion 5c: Worst-case performance for a TBL-CLA in a *stationary, deterministic, and correct* environment with *two target cells per input* is not statistically significantly different than that of a Standard-CLA. (Section 5.4)

Conclusion 5d: For worst-case performance for a TBL-CLA in a *stationary, deterministic, and correct* environment with *two target cells per input*, set the support threshold to 99.99% and the independence threshold to 50%. (Section 5.4)

Conclusion 5e: When all the TBL thresholds are set to 99.99% in a *stationary, deterministic, and correct* environment with *two target cells per input*, the TBL-CLA's performance plateaus briefly as the selection confidence rises. (Section 5.5)

Conclusion 6: TBL-CLA performance improves without bound as the TBL advantage, TBL_{α}, of the TruthTable game state is increased in a *stationary, deterministic, and correct* environment with *two target cells per input*. (Section 5.4)

Conclusion 7: The learning behavior of a TBL-CLA is significantly affected by the settings of the TBL thresholds in a *task-switching, deterministic, and correct* environment with *one target cell per input*. (Section 5.6)

Conclusion 7a: For best performance of a TBL-CLA in a *task-switching, deterministic, and correct* environment with *one target cell per input*, set the withdrawal threshold at 90% and the support and independence equal to each other at values between 90 and 98%. (Section 5.6)

Conclusion 7b: For worst-case performance of a TBL-CLA in a *task-switching, deterministic, and correct* environment with *one target cell per input*, set the support threshold to 99.99% and the withdrawal threshold to 50%. (Section 5.6)

Conclusion 8: When the TBL thresholds are at their optimal settings, the TBL-CLA significantly outperforms the Standard-CLA at no statistically significant Expense across all collection lengths and TruthTable game states in a *task-switching, deterministic, and correct* environment with *one target cell per input*. This implies that TBL is a useful strategy for learning and recovering learning performance in a changing environment. (Section 5.6)

Conclusion 9: The effectiveness of Tactic-Based Learning to aide in recovery of learning behavior after a change in the environment is more strongly influenced by the number of stimulants that can become followers of existing tactics than the percentage of the game that changes or the number of new tactics that must be identified. (Section 5.7)

Conclusion 10: When the TBL-CLA is able to reduce the number of stimulants that select their respondents at random, its performance improves. (Sections 5.3, 5.5, and 5.7)

5.8 Informal Observations

This section contains the informal observations and speculations that have arisen over the course of this research. These observations have not been formally tested or validated, but they are informed by several years of observation while conducting this research.

1. For almost all settings of the TBL thresholds under all of the environmental conditions tested in this research, the TBL-CLA exhibits some significant advantage during the learning process. This implies that there might be some way to turn off or change learning techniques before the disadvantages started to take over. TBL appears to provide a strong boost in early learning, but it is clearly not the only important component of learning in generalization.

While this research has identified the optimal settings for the TBL thresholds across very broad environmental conditions, it is possible that these optimal settings may not be appropriate for other environmental situations. The fact that the TBL-CLA shows some advantage in the early stages of learning suggests that is it a useful technique to apply to other, more complicated problems combined with a mechanism for moderating how and when Tactic-Based Learning is applied.

2. The independence threshold was included to cover situations in which a CLA would be stuck in a local maximum. In this research, the compensation policy was an integral part of working against this possibility. The TruthTable game does not include target responses that are differently weighted. That is to say, all target responses are equally correct, so once a stimulant has found one target cell, there is no reason to look for another, even if one exists. It still seems like a good idea to include the independent role, but its value has not been proved by these experiments.

Breaking from the patterns of behavior given to children by their parents is a vital part of human development. It is important to spend some time questioning the values with which one was raised because it helps form an individual's identity. This rebellion might even be considered a check against bad and abusive parenting, although it is clear that the scars of abuse go very deep and are often difficult to overcome. While this research has not made a strong case for the utility of the independent phase of learning, it should remain an important part of any complex artificially intelligent system.

3. While Tactic-Based Learning has only been applied to Collective Learning Systems, it seems that the underlying theory should be applicable to other reinforcement learning paradigms. With some adjustments, these ideas should translate into other areas of machine learning. It seems that the ability to bias the learning agent in the direction of effective solutions would be useful to other machine learning disciplines.

5.9 Future Directions

This section discusses suggestions for future research.

1. Set the TBL and compensation thresholds dynamically, using a CLA

In this research, the TBL thresholds were fixed throughout the learning process, but in order to build a more flexible system is seems that these thresholds should be adjustable and able to react to changes in the environment and growing experience. The best way to do this would be to have another CLA learn where to set the thresholds and when to change them.

2. Experiment with a more complex environment

The TruthTable game is a very simplified game that has been limited in scope. The benefits of using TBL should translate well to environments that are complex enough to require far more than 300 stimulants and 6 respondents.

3. Experiment with an environment that changes independently of the CLA preparedness

In the research, CLAs were allowed to train on each phase of the game until they were completely confident and accurate, but many learning environments change without consideration of the learner's readiness. Even if a TBL-CLA has not had a chance to become completely confident and accurate, it should still do better than a Standard-CLA.

4. Experiment with an environment that changes incrementally

In this research, when the environment changed from one state to the next, the change happened all at once; however, there are many real-world applications were the environment is changing slowing and continually. Over time, a TBL-CLA should be able to perform better in this kind of changing environment than a Standard-CLA.

5. Implementing a TBL-CLA in a real-world application, such a video game

Video games can provide rich environments for learning from human interaction. It is possible that a TBL-CLA could learn individual players’ behavior quickly enough to provide a more interesting and challenging opponent or non-player character ally. This would move the computer-generated components of the game away from predictable behavior and towards a more dynamic play experience.

6. Experiment with environments with non-uniformly weighted target responses

It was mentioned in the informal observations (Section 5.8) that this research has not found a strong justification for the inclusion of the independent role, although it appears to have a strong biological precedent. The environment used in this research had uniformly valued target responses, which explains why there is no benefit to exploring the response range once a target cell has been found. The usefulness of the independent role might become more apparent in an environment with multiple, but unequally valued target responses.

7. Experiment with combining TBL and feature analysis

One of the major contributions of this research is that it identifies a new method for improving learning without using any feature analysis or generalization; however, it has also been speculated that TBL can not be the only piece that is needed for complex learning. TBL could be combined with some kind of feature comparison to, perhaps, decide if a tactic was worth following or not before choosing it from the global tactic list.

8. Investigating learning pathologies that are introduced with TBL

While TBL can be very effective at improving the learning behavior of a CLA, it can also be made to hinder the learning process. If TBL is actually replicating or imitating the human learning process on some level, it is possible that there might be parallels in the problems it can introduce into the learning process. One area to look for parallel is the area of learning disabilities. It would be interesting to look for parallel between these behaviors and those of people with learning disabilities. If TBL is modeling some sort of behavior that is happening in the brain, then it might be true that it produces parallel patterns of failure. Additionally, it might be possible to model some of the therapies with a sub-optimal TBL-CLA to see if it is possible to correct the deficiencies introduced by incorrect settings of the TBL thresholds.

REFERENCES

Armstrong, A., & Bock, P., (2005) “Mentoring: An Intelligently Biased Selection Policy for Collective Learning Automata*”*, *Conference Proceedings, Intelligent Engineering Systems through Artificial Neural Networks (ANNIE ‘05)* Volume 15, ASME Press, New York.

Barto, A., Dietterich, T. G. (2004) “Reinforcement learning and its relationship to supervised learning”. In J. Si, A. G. Barto, W. B. Powell, D. Wunsch II (Eds.) *Handbook of Learning and Approximate Dynamic Programming*. pp. 47-64. Wiley Interscience/IEEE Press, Piscataway, NJ.

Beer, S., (1966) *Decision and Control: The Meaning of Operational Research and Management Cybernetics*, West Sussex, England.

Berk, L. (2003) *Child Development*, Boston, Allyn & Bacon.

Birk, A., Paul, W. (2000) “Schemas and genetic Algorithms”, *Prerational Intelligence, Volume 2,* Ritter, Cruse, Dean (editors), Kluwer.

Birk, A. (1996) “Learning Geometric Concepts with an Evolutionary Algorithm”, *Proceedings of the Fifth Annual Conference on Evolutionary Programming,* Cambridge, MA, MIT Press.

Bock, P., Heckman, K. (2002) "An Assessment of Personality in a Collective Learning System with the Five-Factor Model of Personality, *Proceedings of the 6th World Multi-Conference on Systemics, Cybernetics, and Informatics (SCI’2002),* Orlando, FL.

Bock, P., Riopka, T. P. (2000) “Intelligent Recombination Using Individual Learning in a Collective Learning Genetic Algorithm”, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000),* Las Vegas.

Bock, P., (1993) *The Emergence of Artificial Cognition: An Introduction to Collective Learning*, World Scientific, New Jersey.

Bock, P., Rovener, R., Kocinski, C. J. (1990) "A Performance Evaluation of ALIAS for the Detection of Geometric Anomalies on Fractal Images," *Advanced Neural Computers*, (R. Eckmiller, ed.), 237-246, Elsevier North-Holland, The Netherlands.

Bock, P. (1985) "The Emergence of Artificial Intelligence: Learning to Learn", *The AI Magazine* (AAAI), vol. 6, no. 3, 180-190.

Bock, P. (1976) "Observation of the Properties of a Collective Learning Stochastic Automaton", *Proceedings of the International Information Sciences Symposium*, Patras, Greece.

Bjorklund, D. F. (2000) *Children's Thinking* (3rd ed.), Belmont, CA, Wadsworth.

Blass, E. M., Ganchrow, J. R., Steiner, J. E. (1984) "Classical conditioning in newborn humans 2 – 48 hours of age", *Infant Behavior and Development*, *volume 7*, 223-235.

Brazelton, T.B., Nugent, J. K. (1995) *Neonatal Behavioral assessment Scale*. London: Mac Keith Press.

Canada Centre for Remote Sensing (March 2006) http://ccrs.nrcan.gc.ca/glossary/index_e.php?id=341

Casella, G. & Berger R. (2002) *Statistical Inference*. 2nd edition. Pacific Grove, California: Duxbury.

Cohen, D. (2002) *How the Child's Mind Develops*, New York, NY: Taylor & Francis Inc.

de Jong, E. D. (1999) "Autonomous concept formation", *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence* IJCAI'99.

DeLoache, J.S. (1987) "Rapid change in symbolic functioning of very young children", *Science*, 238, *1556-1557.*

Drescher, G. (1991) *Made up Minds: A constructivist Approach to Artificial Intelligence.* Cambridge, MA, The MIT Press.

Eiden, R. D., Reifman, A. (1996) "Effects of Brazelton demonstrations on later parenting: A meta-analysis", *Journal of Pediatric Psychology*, *volume 21*, 857-868.

Evans, R. (1973) *Jean Piaget; The man and his ideas.* New York: E.P. Dutton.

Fischer, R. A. (1950) *Contributions to Mathematical Statistics.* New York: Wiley.

Floccia, C., Christophe, A., Bertoncini, J. (1997) "High-amplitude sucking and newborns: the quest for underlying mechanisms", *Journal of Experimental Child Psychology, volume 64*, 174-198.

Hartshorn, K., Rovee-Collier, C., Gerhardstein, P., Bhatt, R. S., Klein, P. J., Aaron, F., Wondoloski, T.L., Wurtzel, N. (1998) "Developmental changes in the specificity of memory over the first year of life", *Developmental Psychology, volume 33*, 61-78.

Hayne, H., Boniface, J., & Barr, R. (2000) "The development of declarative memory in human infants: age-related changes in deferred imitation", *Behavioral Neuroscience, volume 114,* 77-83.

Heckman, K. (2004) *An assessment of the effect of personality and forward context on the expertise of a collective learning system using the FIVE-FACTOR model of personality*. Doctoral Dissertation, The George Washington University Press, Washington, DC.

Inhelder, B, Piaget, J., (1958) *The growth of logical thinking from childhood to adolescence: An essay on the construction of formal operational structures.* New York: Basic Books. (Original work published 1955)

Kaelbling, L. P., Littman, M. L., Moore, A. W. (1996) "Reinforcement learning: A survey", *Journal of Artificial Intelligence Research, volume 4,* 237-285.

Miller, P.H. (1993) *Theories of developmental psychology. (3rd ed.).* New York: Freeman.

Mitchell, T. (1997) *Machine Learning*. McGraw-Hill.

A.W. Moore, M.S. Lee, "Efficient algorithms for minimizing cross validation error", Proceedings of the 11th International Conference on Machine Learning. Morgan Kauffman, San Francisco, CA, 1994.

Ni, Y. (1998) "Cognitive structure, content knowledge, and classificatory reasoning". *Journal of Genetic Psychology*, volume 159*, 280-296.*

O'Reilly, A.W. (1995) "Using representations: Comprehension and production of actions with imagined objects", *Child Development,* 66, 999-1010.

Piaget, J. (1926) *Judgment and reasoning in the child*. New York: Harcourt, Brace & World. (Original work published 1926)

Piaget, J., (1952) The *origins of intelligence in children*. New York: International Universities Press. (Original work published 1936)

Piaget, J., Inhelder, B., & Szeminska, A. (1960) *The child's conception of geometry*. New York: Basic Books. (Original work published 1948)

Portnoy, D., Bock, P. (2005) "Unsupervised Fuzzy-Membership Estimation of Terms in Semantic and Syntactic Lexical Classes", Proceedings of the 33rd IEEE Applied Imagery Pattern Recognition Workshop. Los Alamitos, CA

Pulaski, M. A. S. (1980) *Understanding Piaget: An Introduction to Children's Cognitive Development*. New York: Harper & Row.

Ross, K. A., Wright, C. R. B. (1988) *Discrete Mathematics 2nd ed.*, Englewood Cliffs, NJ, Prentice Hall.

Russell, S., Norvig, P. (2003) *Artificial Intelligence: A Modern Approach 2nd ed.*, Englewood Cliffs, NJ, Prentice Hall.

Shields, P. J., Rovee-Collier, C. K. (1992) "Long-term memory for context-specific category information at six months", *Child Development, volume 3*, 745-753.

Short, N. (2006) *The Remote Sensing Tutorial* (awarded the NASA Goddard "Excellence in Outreach Award" 2003) http://rst.gsfc.nasa.gov/Homepage/Homepage.html

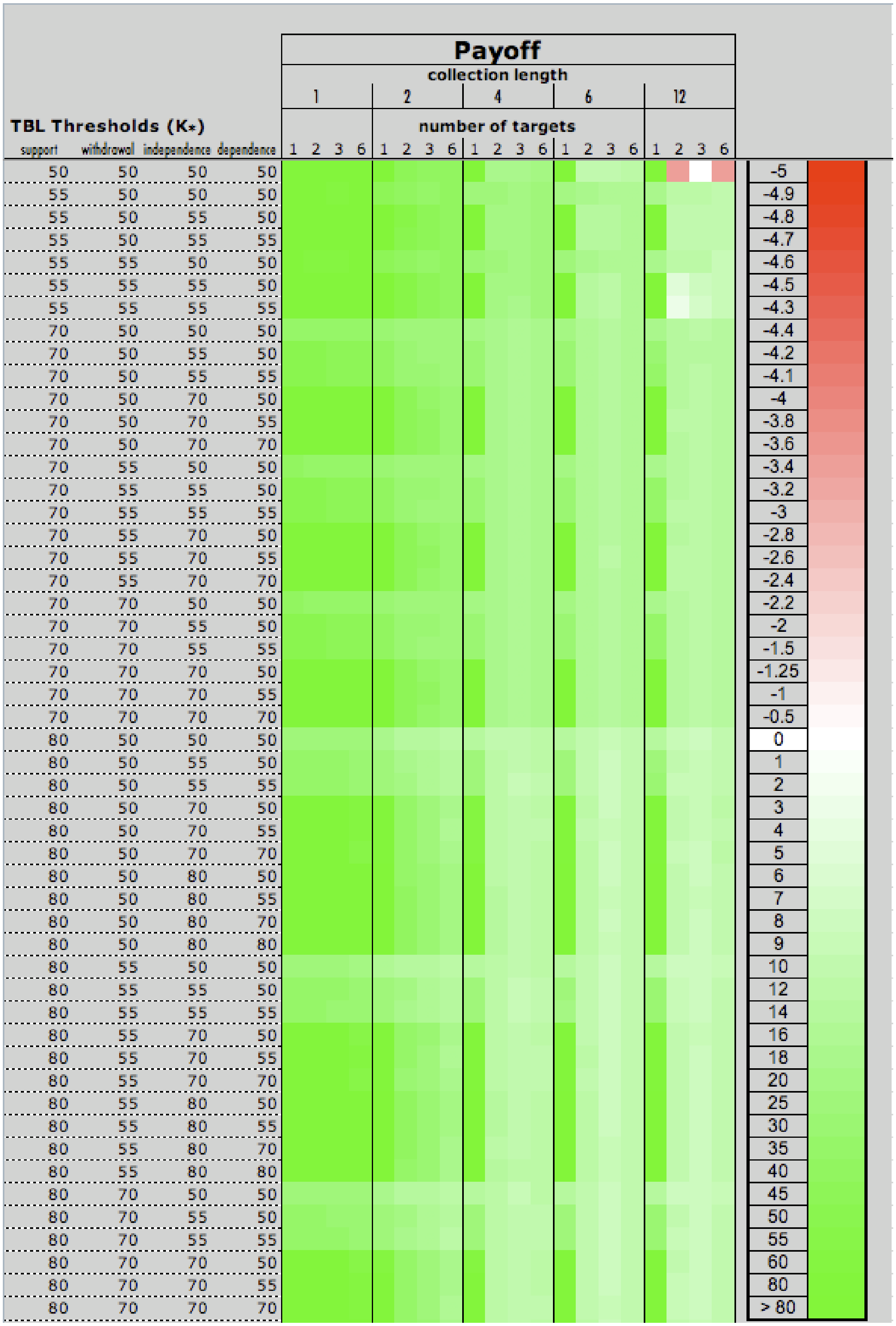
Siegler, R.S., Ellis, S. (1996) Piaget on childhood. *Psychological Science*, 7, *211-215.*

Sutton, R. S., & Barto, A. G. (1998) *Reinforcement Learning: An Introduction*. Cambridge, MA: The MIT Press.

Sycara, K., Guttal, R., Koning, J., Narasimhan, S., Navinchandra, D. (1992) "CADET: A case-based synthesis tool for engineering design". *International Journal of Expert Systems*, 4(2), 157-188

APPENDIX A: EXPERIMENT 1, PAYOFF RESULTS

This appendix presents the entire Payoff footprint, sorted by the TBL thresholds.



Payoff

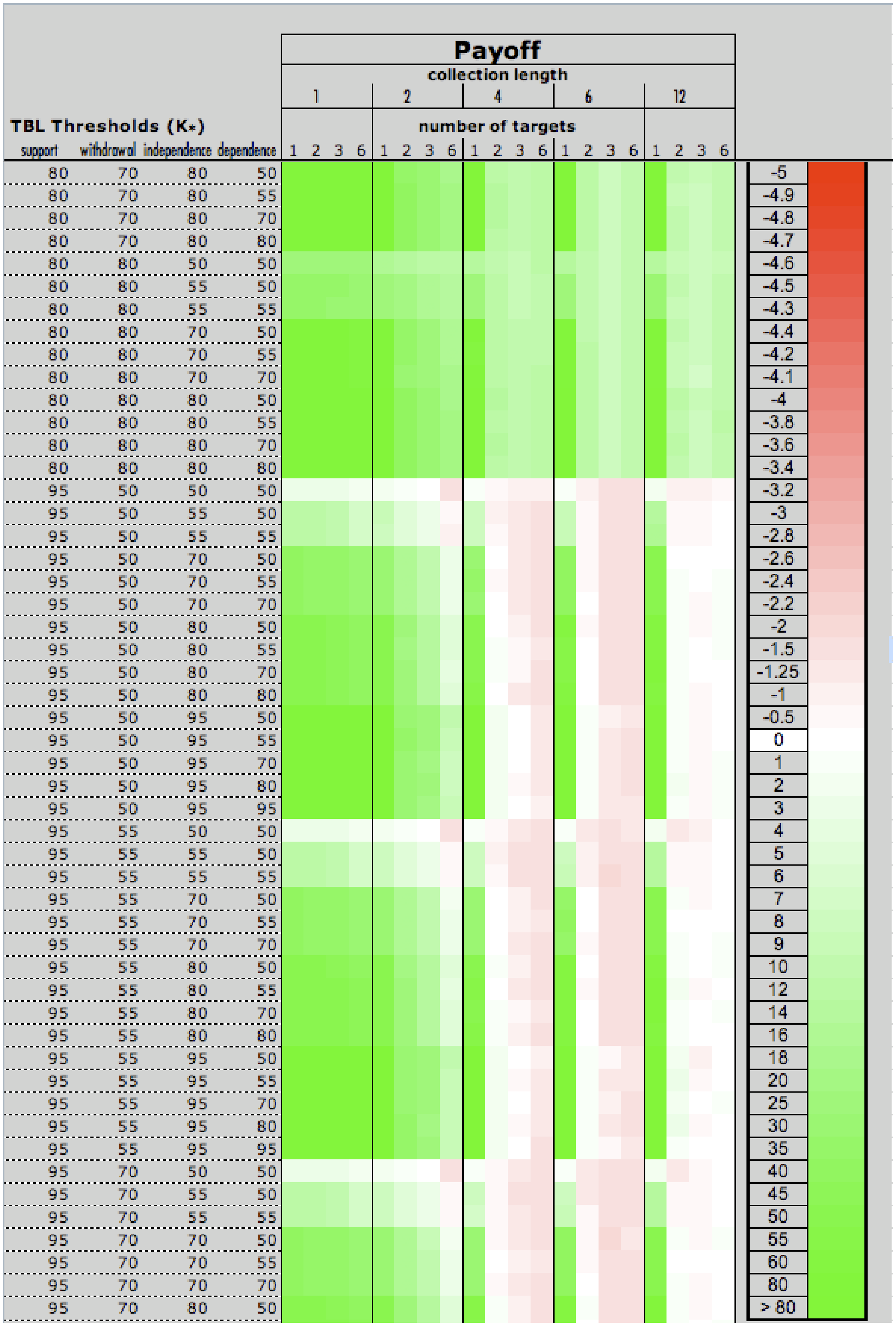
collection length: 1, 2, 4, 6, 12

number of targets: 1 2 3 6 | 1 2 3 6 | 1 2 3 6 | 1 2 3 6 | 1 2 3 6

TBL Thresholds (K*)

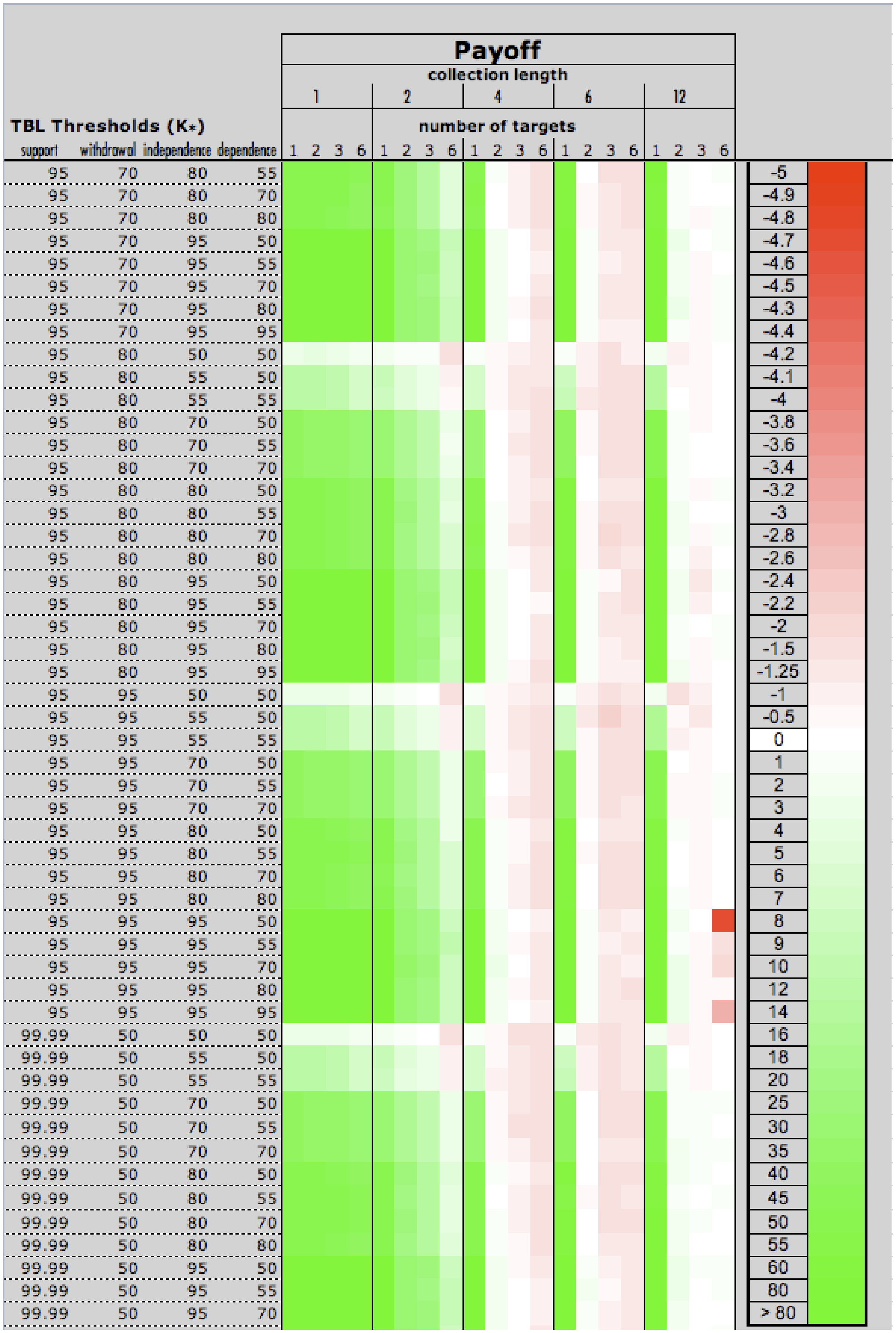
support	withdrawal	independence	dependence
50	50	50	50
55	50	50	50
55	50	55	50
55	50	55	55
55	55	50	50
55	55	55	50
55	55	55	55
70	50	50	50
70	50	55	50
70	50	55	55
70	50	70	50
70	50	70	55
70	50	70	70
70	55	50	50
70	55	55	50
70	55	55	55
70	55	70	50
70	55	70	55
70	55	70	70
70	70	50	50
70	70	55	50
70	70	55	55
70	70	70	50
70	70	70	55
70	70	70	70
80	50	50	50
80	50	55	50
80	50	55	55
80	50	70	50
80	50	70	55
80	50	70	70
80	50	80	50
80	50	80	55
80	50	80	70
80	50	80	80
80	55	50	50
80	55	55	50
80	55	55	55
80	55	70	50
80	55	70	55
80	55	70	70
80	55	80	50
80	55	80	55
80	55	80	70
80	55	80	80
80	70	50	50
80	70	55	50
80	70	55	55
80	70	70	50
80	70	70	55
80	70	70	70

-5	🟥
-4.9	🟥
-4.8	🟥
-4.7	🟥
-4.6	🟥
-4.5	🟥
-4.3	🟥
-4.4	🟥
-4.2	🟥
-4.1	🟥
-4	🟥
-3.8	🟥
-3.6	🟥
-3.4	🟥
-3.2	🟥
-3	🟥
-2.8	🟥
-2.6	🟥
-2.4	🟥
-2.2	🟥
-2	🟥
-1.5	🟥
-1.25	🟥
-1	🟥
-0.5	🟥
0	⬜
1	🟩
2	🟩
3	🟩
4	🟩
5	🟩
6	🟩
7	🟩
8	🟩
9	🟩
10	🟩
12	🟩
14	🟩
16	🟩
18	🟩
20	🟩
25	🟩
30	🟩
35	🟩
40	🟩
45	🟩
50	🟩
55	🟩
60	🟩
80	🟩
> 80	🟩



TBL Thresholds (K*)

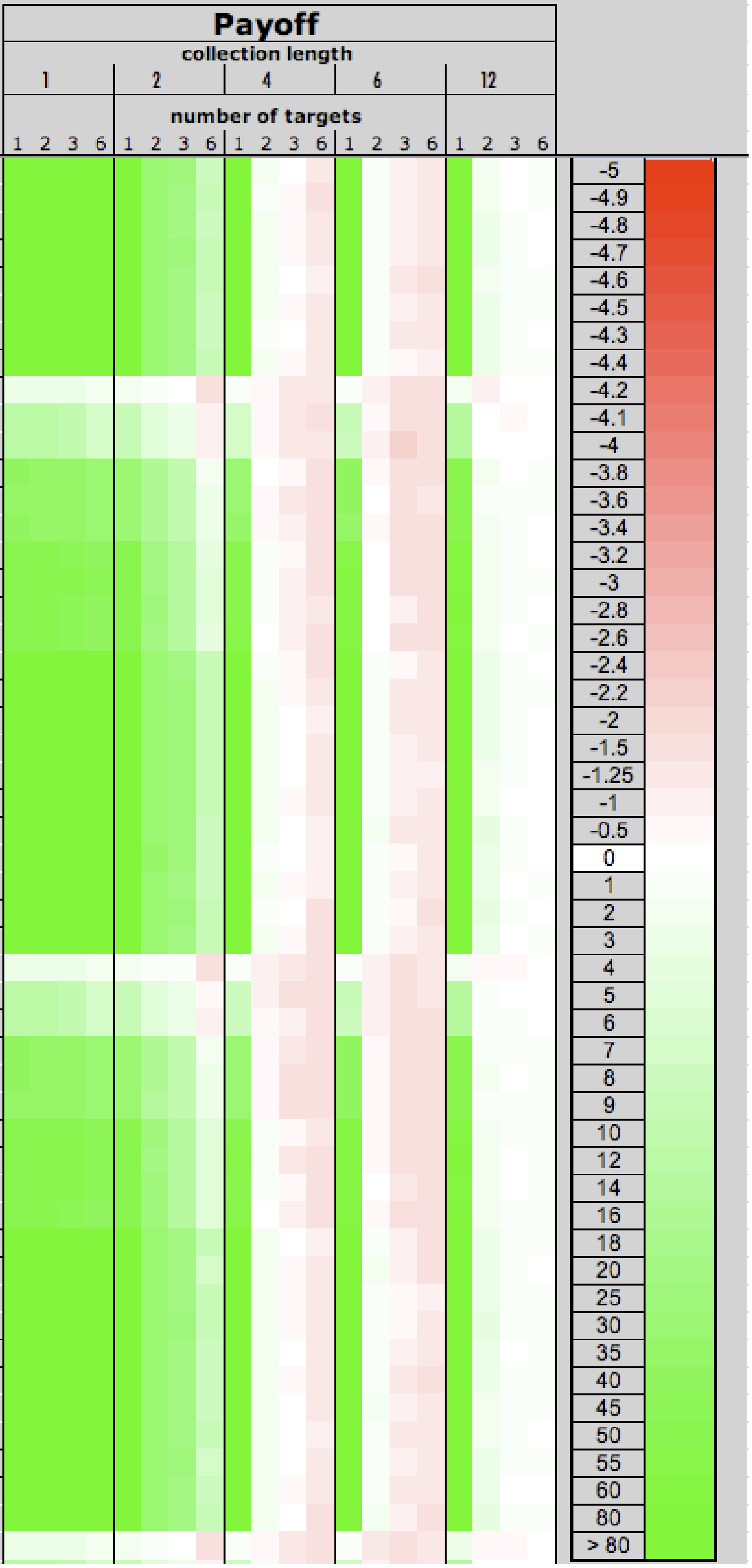
support	withdrawal	independence	dependence
80	70	80	50
80	70	80	55
80	70	80	70
80	70	80	80
80	80	50	50
80	80	55	50
80	80	55	55
80	80	70	50
80	80	70	55
80	80	70	70
80	80	80	50
80	80	80	55
80	80	80	70
80	80	80	80
95	50	50	50
95	50	55	50
95	50	55	55
95	50	70	50
95	50	70	55
95	50	70	70
95	50	80	50
95	50	80	55
95	50	80	70
95	50	80	80
95	50	95	50
95	50	95	55
95	50	95	70
95	50	95	80
95	50	95	95
95	55	50	50
95	55	55	50
95	55	55	55
95	55	70	50
95	55	70	55
95	55	70	70
95	55	80	50
95	55	80	55
95	55	80	70
95	55	80	80
95	55	95	50
95	55	95	55
95	55	95	70
95	55	95	80
95	55	95	95
95	70	50	50
95	70	55	50
95	70	55	55
95	70	70	50
95	70	70	55
95	70	70	70
95	70	80	50

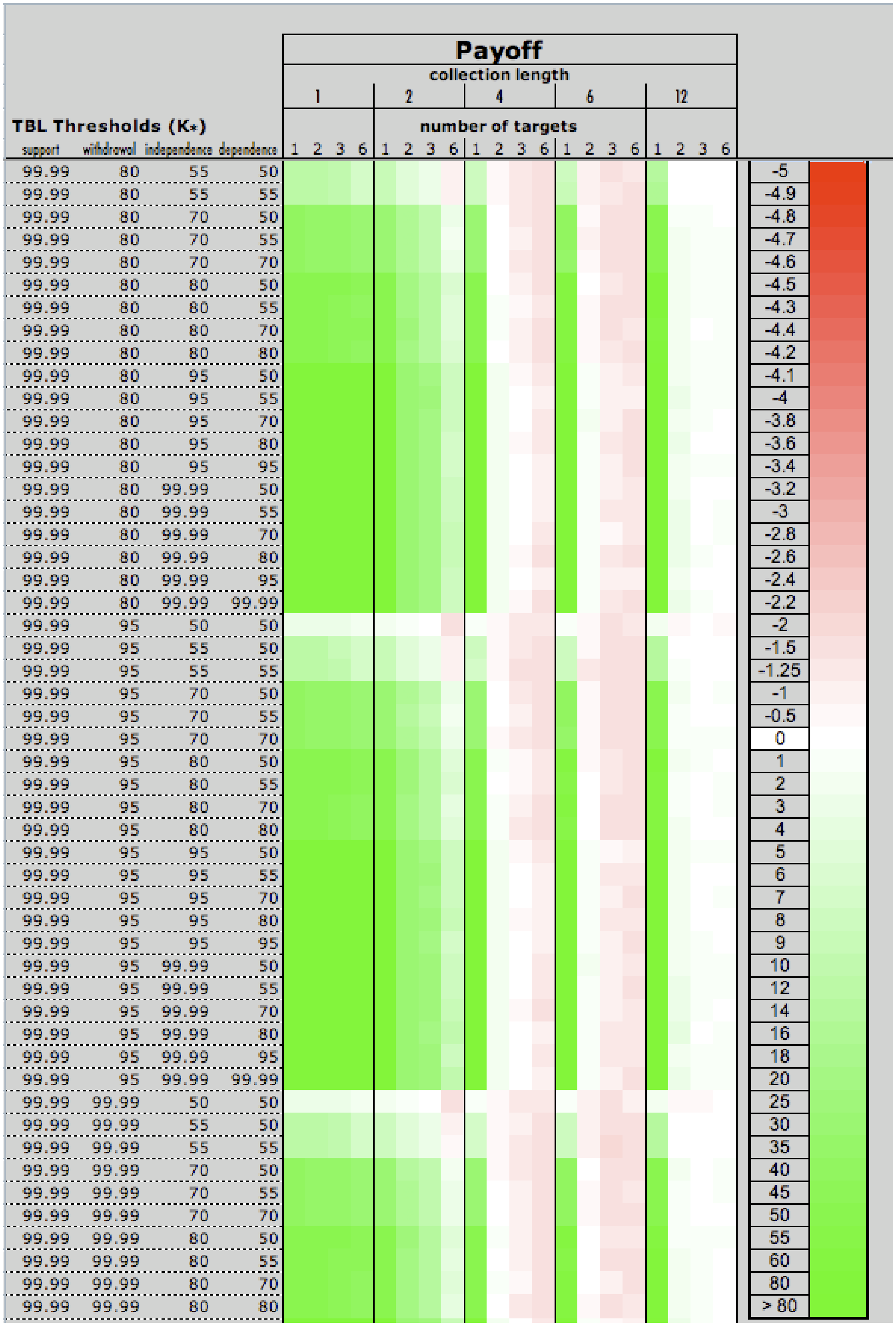


TBL Thresholds (K*)			
support	withdrawal	independence	dependence
95	70	80	55
95	70	80	70
95	70	80	80
95	70	95	50
95	70	95	55
95	70	95	70
95	70	95	80
95	70	95	95
95	80	50	50
95	80	55	50
95	80	55	55
95	80	70	50
95	80	70	55
95	80	70	70
95	80	80	50
95	80	80	55
95	80	80	70
95	80	80	80
95	80	95	50
95	80	95	55
95	80	95	70
95	80	95	80
95	80	95	95
95	95	50	50
95	95	55	50
95	95	55	55
95	95	70	50
95	95	70	55
95	95	70	70
95	95	80	50
95	95	80	55
95	95	80	70
95	95	80	80
95	95	95	50
95	95	95	55
95	95	95	70
95	95	95	80
95	95	95	95
99.99	50	50	50
99.99	50	55	50
99.99	50	55	55
99.99	50	70	50
99.99	50	70	55
99.99	50	70	70
99.99	50	80	50
99.99	50	80	55
99.99	50	80	70
99.99	50	80	80
99.99	50	95	50
99.99	50	95	55
99.99	50	95	70

TBL Thresholds (K*)

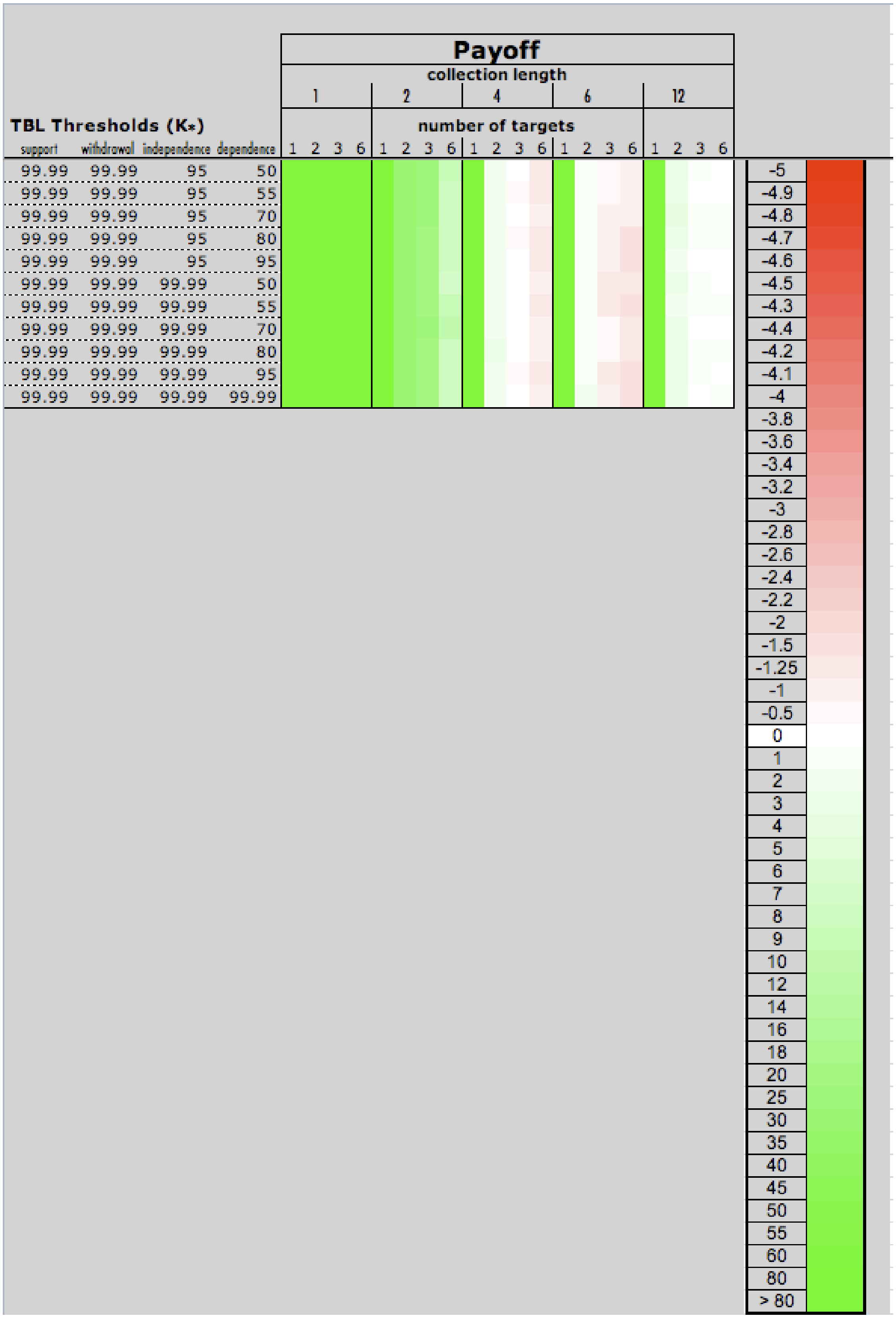
support	withdrawal	independence	dependence
99.99	50	95	80
99.99	50	95	95
99.99	50	99.99	50
99.99	50	99.99	55
99.99	50	99.99	70
99.99	50	99.99	80
99.99	50	99.99	95
99.99	50	99.99	99.99
99.99	55	50	50
99.99	55	55	50
99.99	55	55	55
99.99	55	70	50
99.99	55	70	55
99.99	55	70	70
99.99	55	80	50
99.99	55	80	55
99.99	55	80	70
99.99	55	80	80
99.99	55	95	50
99.99	55	95	55
99.99	55	95	70
99.99	55	95	80
99.99	55	95	95
99.99	55	99.99	50
99.99	55	99.99	55
99.99	55	99.99	70
99.99	55	99.99	80
99.99	55	99.99	95
99.99	55	99.99	99.99
99.99	70	50	50
99.99	70	55	50
99.99	70	55	55
99.99	70	70	50
99.99	70	70	55
99.99	70	70	70
99.99	70	80	50
99.99	70	80	55
99.99	70	80	70
99.99	70	80	80
99.99	70	95	50
99.99	70	95	55
99.99	70	95	70
99.99	70	95	80
99.99	70	95	95
99.99	70	99.99	50
99.99	70	99.99	55
99.99	70	99.99	70
99.99	70	99.99	80
99.99	70	99.99	95
99.99	70	99.99	99.99
99.99	80	50	50





TBL Thresholds (K*)

support	withdrawal	independence	dependence
99.99	80	55	50
99.99	80	55	55
99.99	80	70	50
99.99	80	70	55
99.99	80	70	70
99.99	80	80	50
99.99	80	80	55
99.99	80	80	70
99.99	80	80	80
99.99	80	95	50
99.99	80	95	55
99.99	80	95	70
99.99	80	95	80
99.99	80	95	95
99.99	80	99.99	50
99.99	80	99.99	55
99.99	80	99.99	70
99.99	80	99.99	80
99.99	80	99.99	95
99.99	80	99.99	99.99
99.99	95	50	50
99.99	95	55	50
99.99	95	55	55
99.99	95	70	50
99.99	95	70	55
99.99	95	70	70
99.99	95	80	50
99.99	95	80	55
99.99	95	80	70
99.99	95	80	80
99.99	95	95	50
99.99	95	95	55
99.99	95	95	70
99.99	95	95	80
99.99	95	95	95
99.99	95	99.99	50
99.99	95	99.99	55
99.99	95	99.99	70
99.99	95	99.99	80
99.99	95	99.99	95
99.99	95	99.99	99.99
99.99	99.99	50	50
99.99	99.99	55	50
99.99	99.99	55	55
99.99	99.99	70	50
99.99	99.99	70	55
99.99	99.99	70	70
99.99	99.99	80	50
99.99	99.99	80	55
99.99	99.99	80	70
99.99	99.99	80	80



TBL Thresholds (K*)

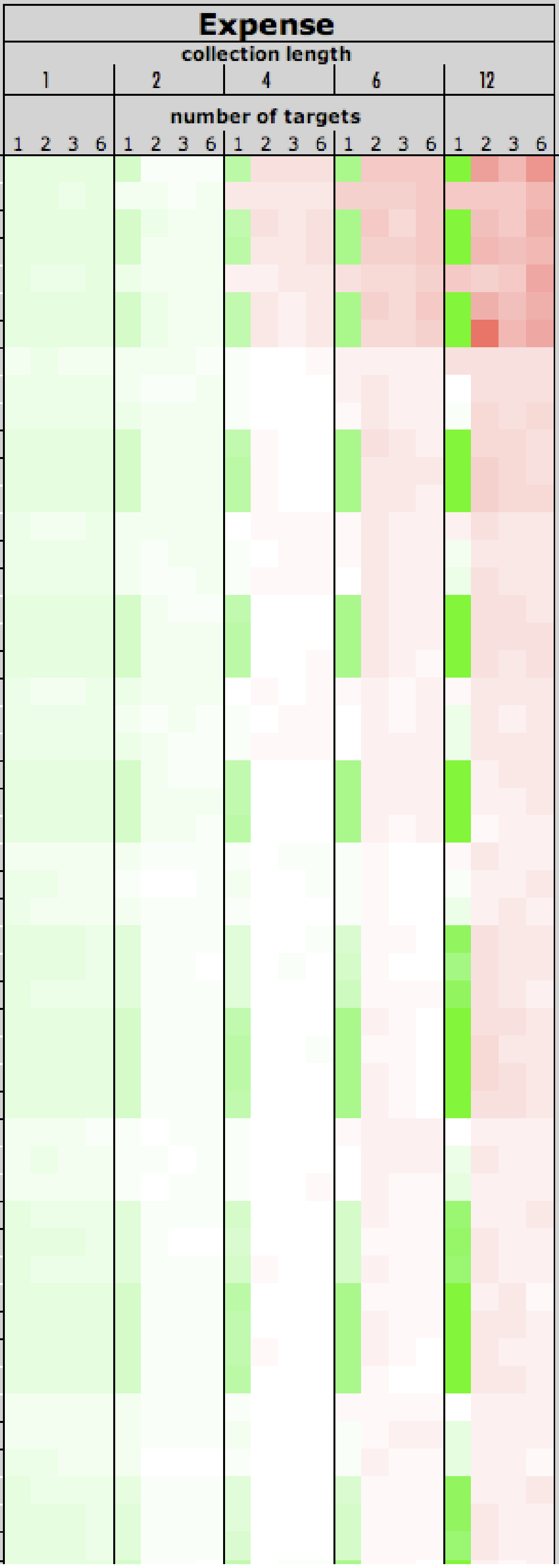
support	withdrawal	independence	dependence
99.99	99.99	95	50
99.99	99.99	95	55
99.99	99.99	95	70
99.99	99.99	95	80
99.99	99.99	95	95
99.99	99.99	99.99	50
99.99	99.99	99.99	55
99.99	99.99	99.99	70
99.99	99.99	99.99	80
99.99	99.99	99.99	95
99.99	99.99	99.99	99.99

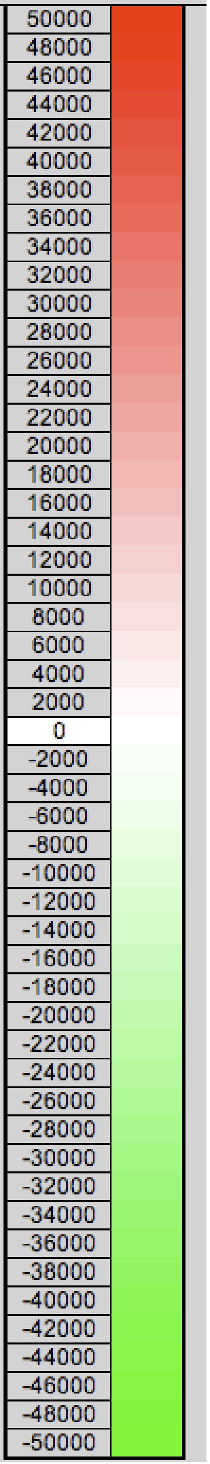
APPENDIX B: EXPERIMENT 1, EXPENSE RESULTS

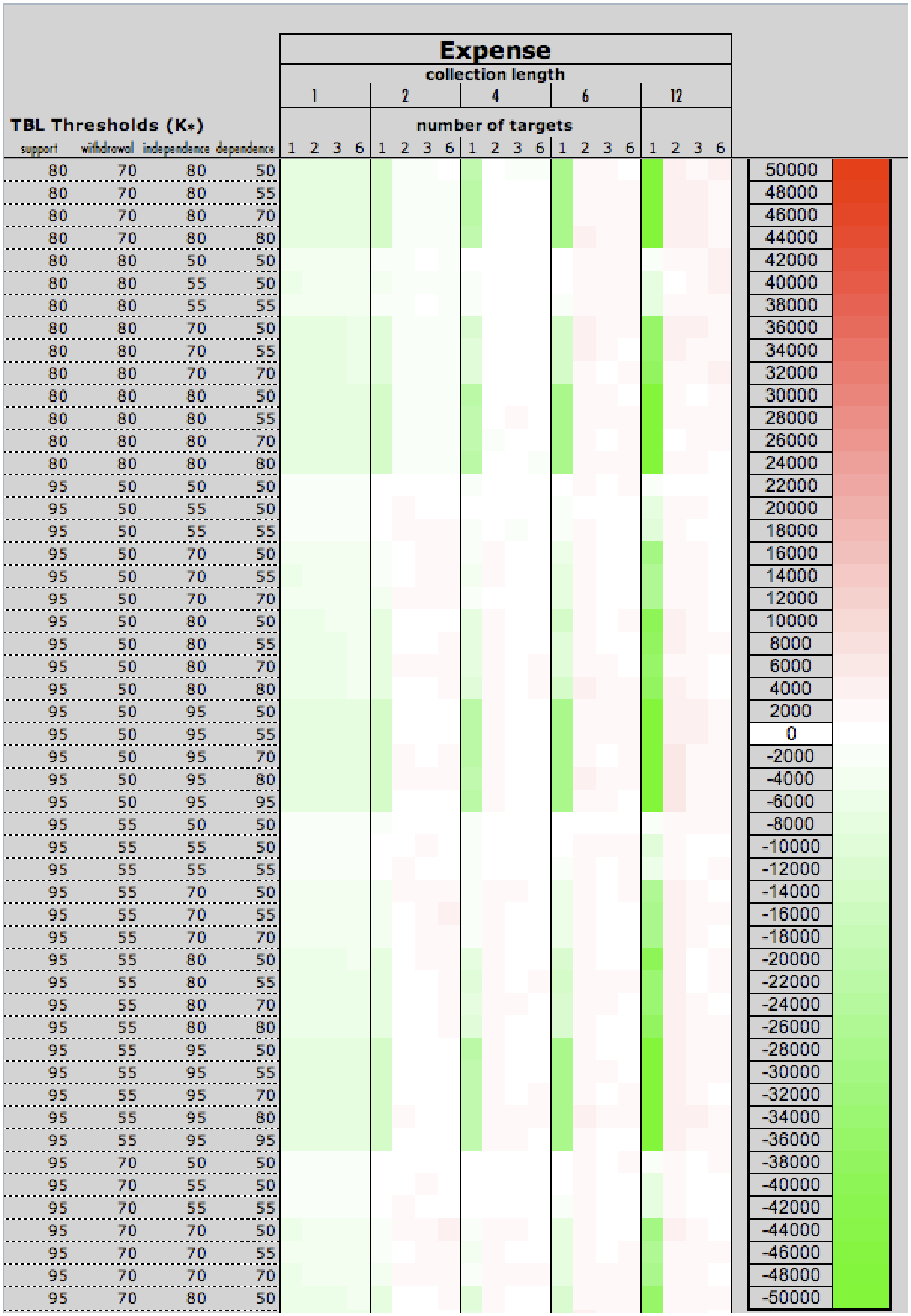
This appendix presents the full Expense results from Experiment 1, sorted by TBL threshold.

TBL Thresholds (K*)

support	withdrawal	independence	dependence
50	50	50	50
55	50	50	50
55	50	55	50
55	50	55	55
55	55	50	50
55	55	55	50
55	55	55	55
70	50	50	50
70	50	55	50
70	50	55	55
70	50	70	50
70	50	70	55
70	50	70	70
70	55	50	50
70	55	55	50
70	55	55	55
70	55	70	50
70	55	70	55
70	55	70	70
70	70	50	50
70	70	55	50
70	70	55	55
70	70	70	50
70	70	70	55
70	70	70	70
80	50	50	50
80	50	55	50
80	50	55	55
80	50	70	50
80	50	70	55
80	50	70	70
80	50	80	50
80	50	80	55
80	50	80	70
80	50	80	80
80	55	50	50
80	55	55	50
80	55	55	55
80	55	70	50
80	55	70	55
80	55	70	70
80	55	80	50
80	55	80	55
80	55	80	70
80	55	80	80
80	70	50	50
80	70	55	50
80	70	55	55
80	70	70	50
80	70	70	55
80	70	70	70







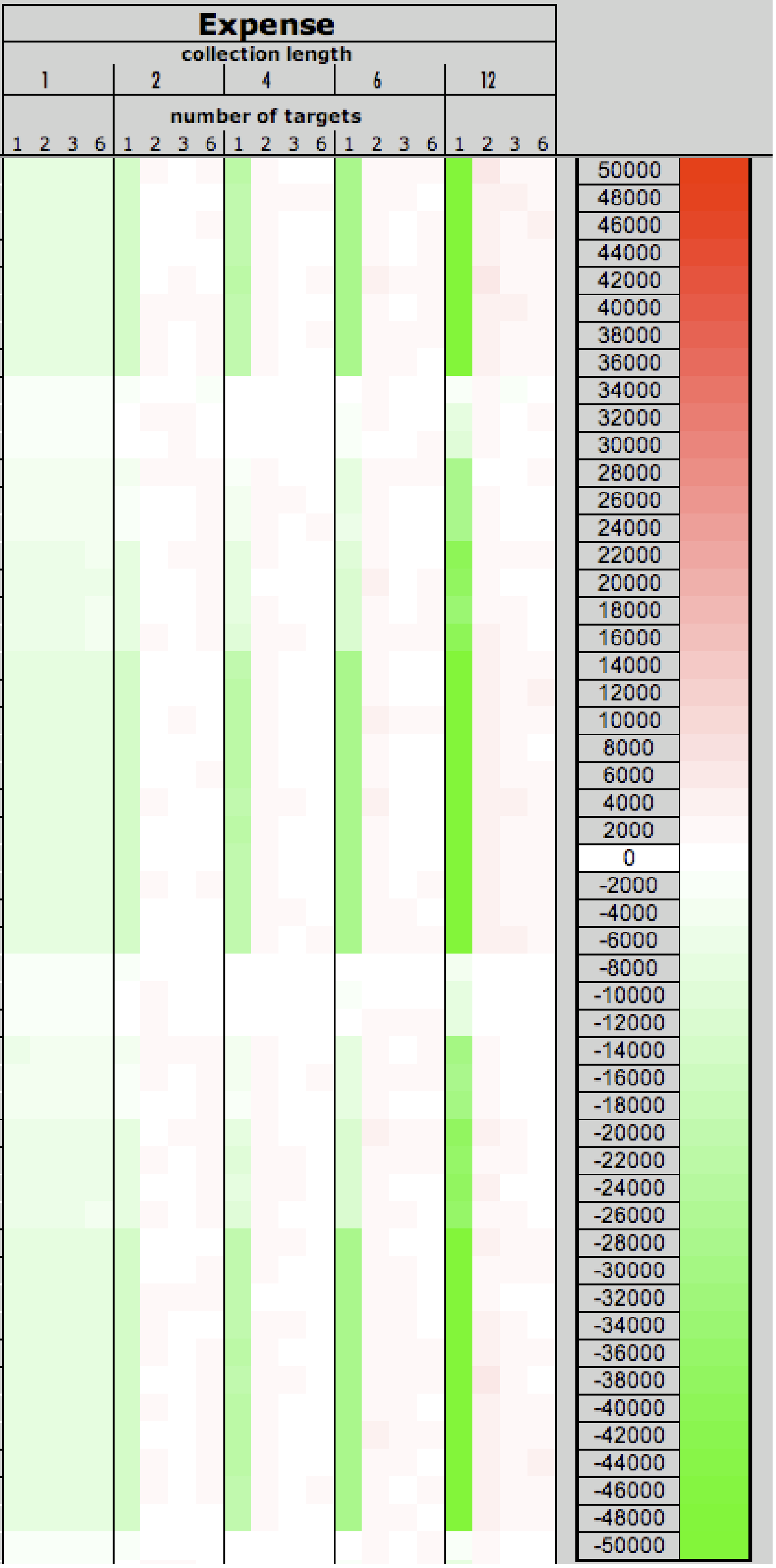
TBL Thresholds (K*)

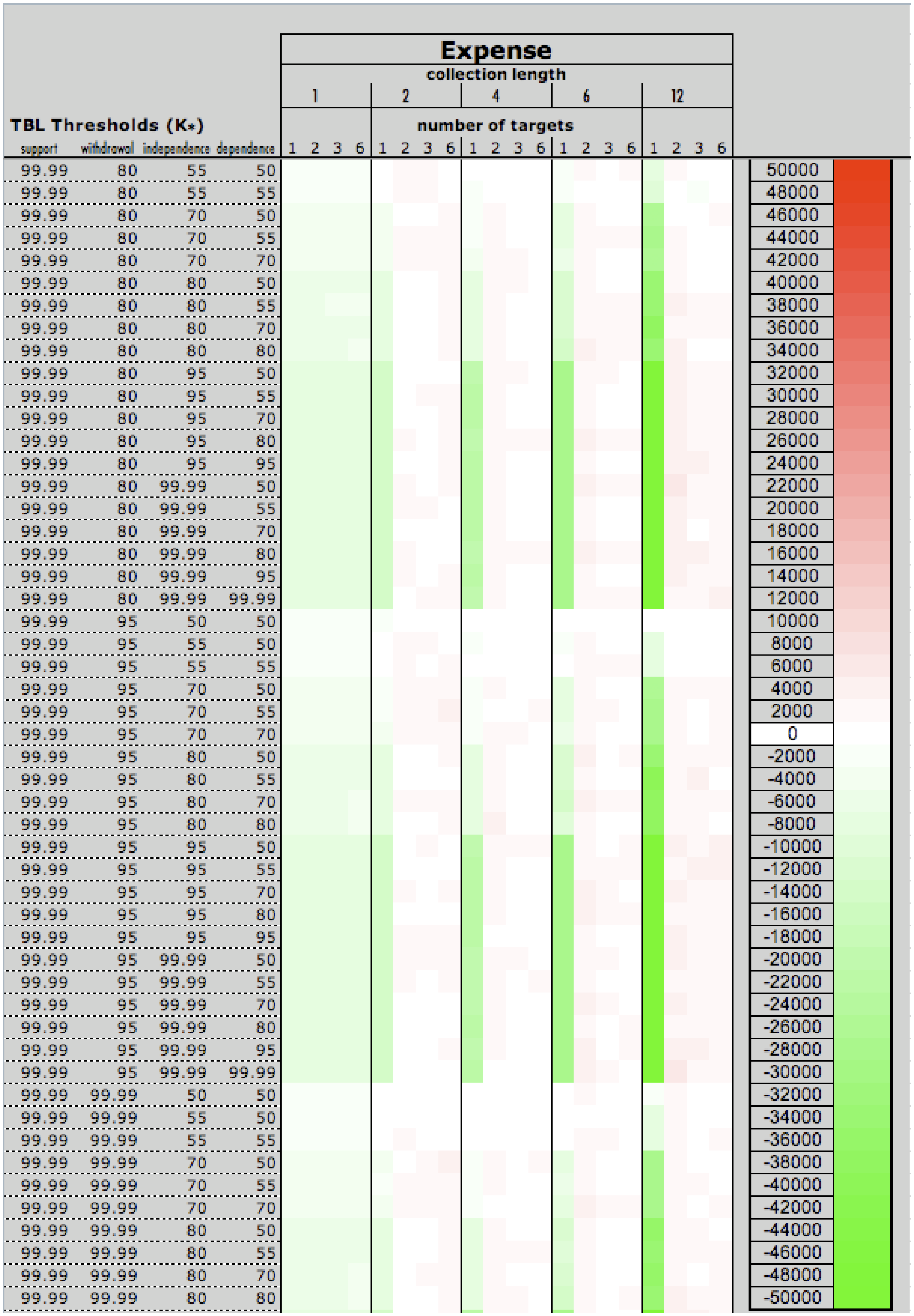
support	withdrawal	independence	dependence
80	70	80	50
80	70	80	55
80	70	80	70
80	70	80	80
80	80	50	50
80	80	55	50
80	80	55	55
80	80	70	50
80	80	70	55
80	80	70	70
80	80	80	50
80	80	80	55
80	80	80	70
80	80	80	80
95	50	50	50
95	50	55	50
95	50	55	55
95	50	70	50
95	50	70	55
95	50	70	70
95	50	80	50
95	50	80	55
95	50	80	70
95	50	80	80
95	50	95	50
95	50	95	55
95	50	95	70
95	50	95	80
95	50	95	95
95	55	50	50
95	55	55	50
95	55	55	55
95	55	70	50
95	55	70	55
95	55	70	70
95	55	80	50
95	55	80	55
95	55	80	70
95	55	80	80
95	55	95	50
95	55	95	55
95	55	95	70
95	55	95	80
95	55	95	95
95	70	50	50
95	70	55	50
95	70	55	55
95	70	70	50
95	70	70	55
95	70	70	70
95	70	80	50

50000	🟥
48000	🟥
46000	🟥
44000	🟥
42000	🟥
40000	🟥
38000	🟥
36000	🟥
34000	🟥
32000	🟥
30000	🟥
28000	🟥
26000	🟥
24000	🟥
22000	🟥
20000	🟥
18000	🟥
16000	🟥
14000	🟥
12000	🟥
10000	🟥
8000	🟥
6000	🟥
4000	🟥
2000	🟥
0	□
-2000	🟩
-4000	🟩
-6000	🟩
-8000	🟩
-10000	🟩
-12000	🟩
-14000	🟩
-16000	🟩
-18000	🟩
-20000	🟩
-22000	🟩
-24000	🟩
-26000	🟩
-28000	🟩
-30000	🟩
-32000	🟩
-34000	🟩
-36000	🟩
-38000	🟩
-40000	🟩
-42000	🟩
-44000	🟩
-46000	🟩
-48000	🟩
-50000	🟩

TBL Thresholds (K*)

support	withdrawal	independence	dependence
99.99	50	95	80
99.99	50	95	95
99.99	50	99.99	50
99.99	50	99.99	55
99.99	50	99.99	70
99.99	50	99.99	80
99.99	50	99.99	95
99.99	50	99.99	99.99
99.99	55	50	50
99.99	55	55	50
99.99	55	55	55
99.99	55	70	50
99.99	55	70	55
99.99	55	70	70
99.99	55	80	50
99.99	55	80	55
99.99	55	80	70
99.99	55	80	80
99.99	55	95	50
99.99	55	95	55
99.99	55	95	70
99.99	55	95	80
99.99	55	95	95
99.99	55	99.99	50
99.99	55	99.99	55
99.99	55	99.99	70
99.99	55	99.99	80
99.99	55	99.99	95
99.99	55	99.99	99.99
99.99	70	50	50
99.99	70	55	50
99.99	70	55	55
99.99	70	70	50
99.99	70	70	55
99.99	70	70	70
99.99	70	80	50
99.99	70	80	55
99.99	70	80	70
99.99	70	80	80
99.99	70	95	50
99.99	70	95	55
99.99	70	95	70
99.99	70	95	80
99.99	70	95	95
99.99	70	99.99	50
99.99	70	99.99	55
99.99	70	99.99	70
99.99	70	99.99	80
99.99	70	99.99	95
99.99	70	99.99	99.99
99.99	80	50	50

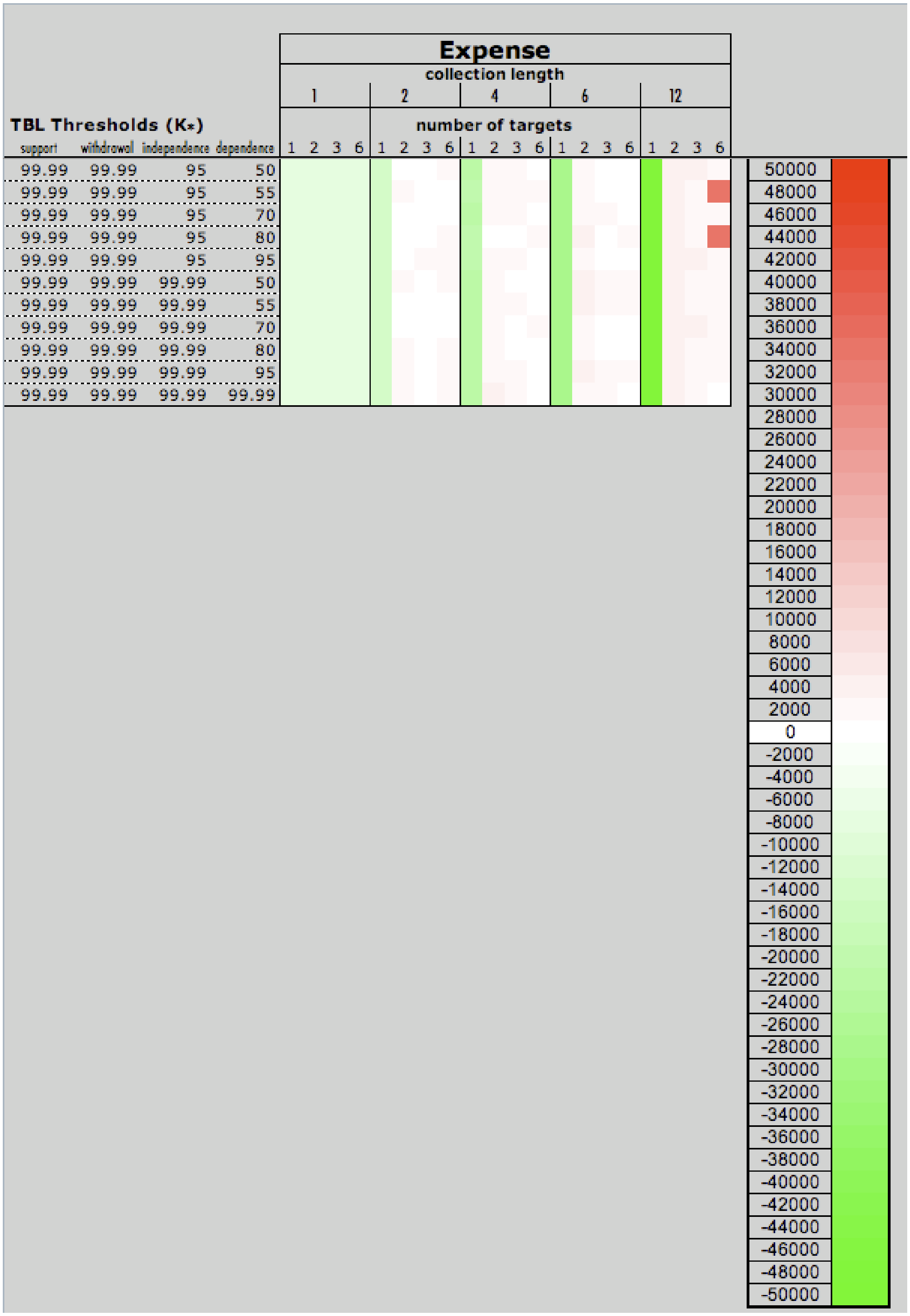




TBL Thresholds (K*)

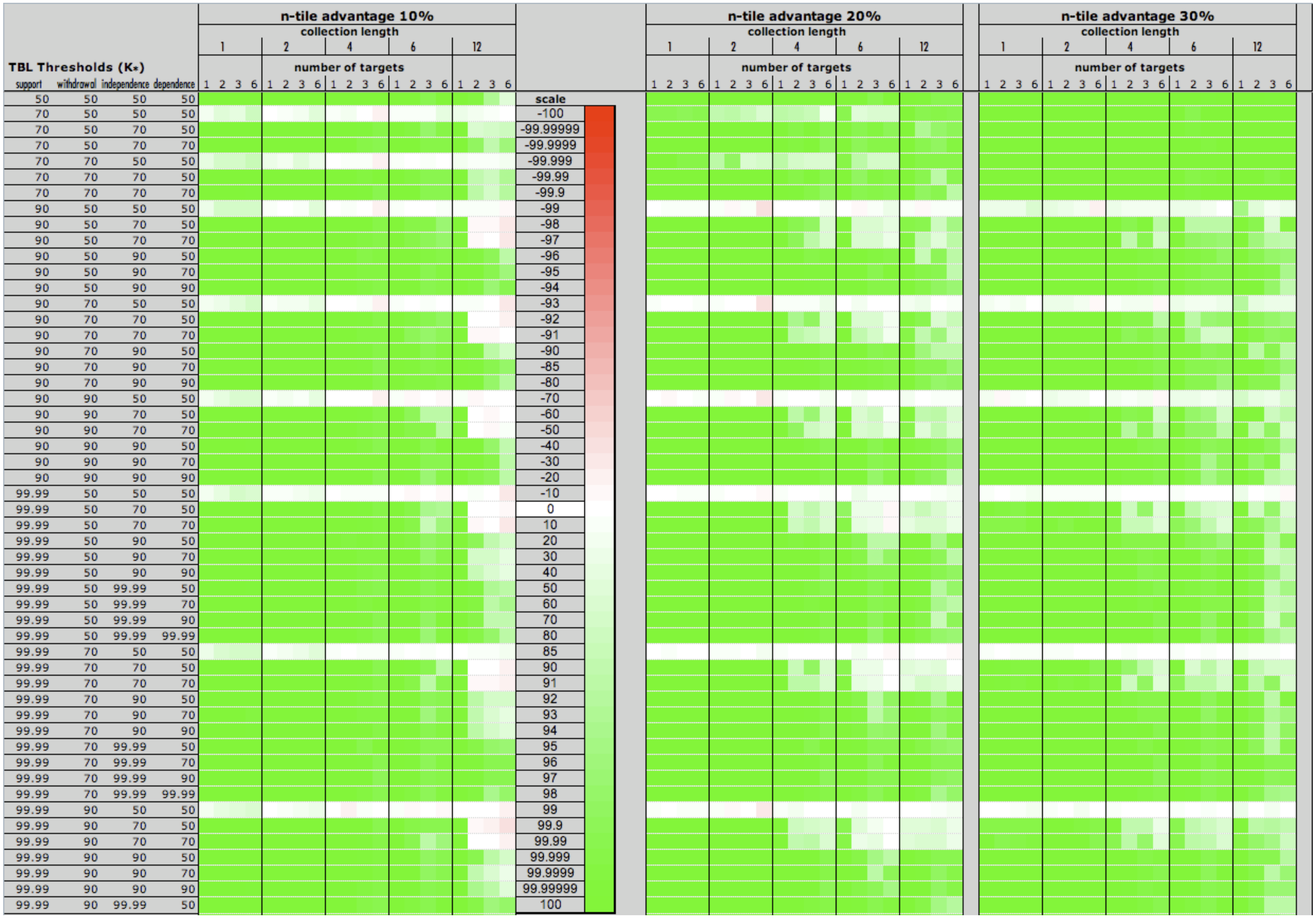
support	withdrawal	independence	dependence
99.99	80	55	50
99.99	80	55	55
99.99	80	70	50
99.99	80	70	55
99.99	80	70	70
99.99	80	80	50
99.99	80	80	55
99.99	80	80	70
99.99	80	80	80
99.99	80	95	50
99.99	80	95	55
99.99	80	95	70
99.99	80	95	80
99.99	80	95	95
99.99	80	99.99	50
99.99	80	99.99	55
99.99	80	99.99	70
99.99	80	99.99	80
99.99	80	99.99	95
99.99	80	99.99	99.99
99.99	95	50	50
99.99	95	55	50
99.99	95	55	55
99.99	95	70	50
99.99	95	70	55
99.99	95	70	70
99.99	95	80	50
99.99	95	80	55
99.99	95	80	70
99.99	95	80	80
99.99	95	95	50
99.99	95	95	55
99.99	95	95	70
99.99	95	95	80
99.99	95	95	95
99.99	95	99.99	50
99.99	95	99.99	55
99.99	95	99.99	70
99.99	95	99.99	80
99.99	95	99.99	95
99.99	95	99.99	99.99
99.99	99.99	50	50
99.99	99.99	55	50
99.99	99.99	55	55
99.99	99.99	70	50
99.99	99.99	70	55
99.99	99.99	70	70
99.99	99.99	80	50
99.99	99.99	80	55
99.99	99.99	80	70
99.99	99.99	80	80

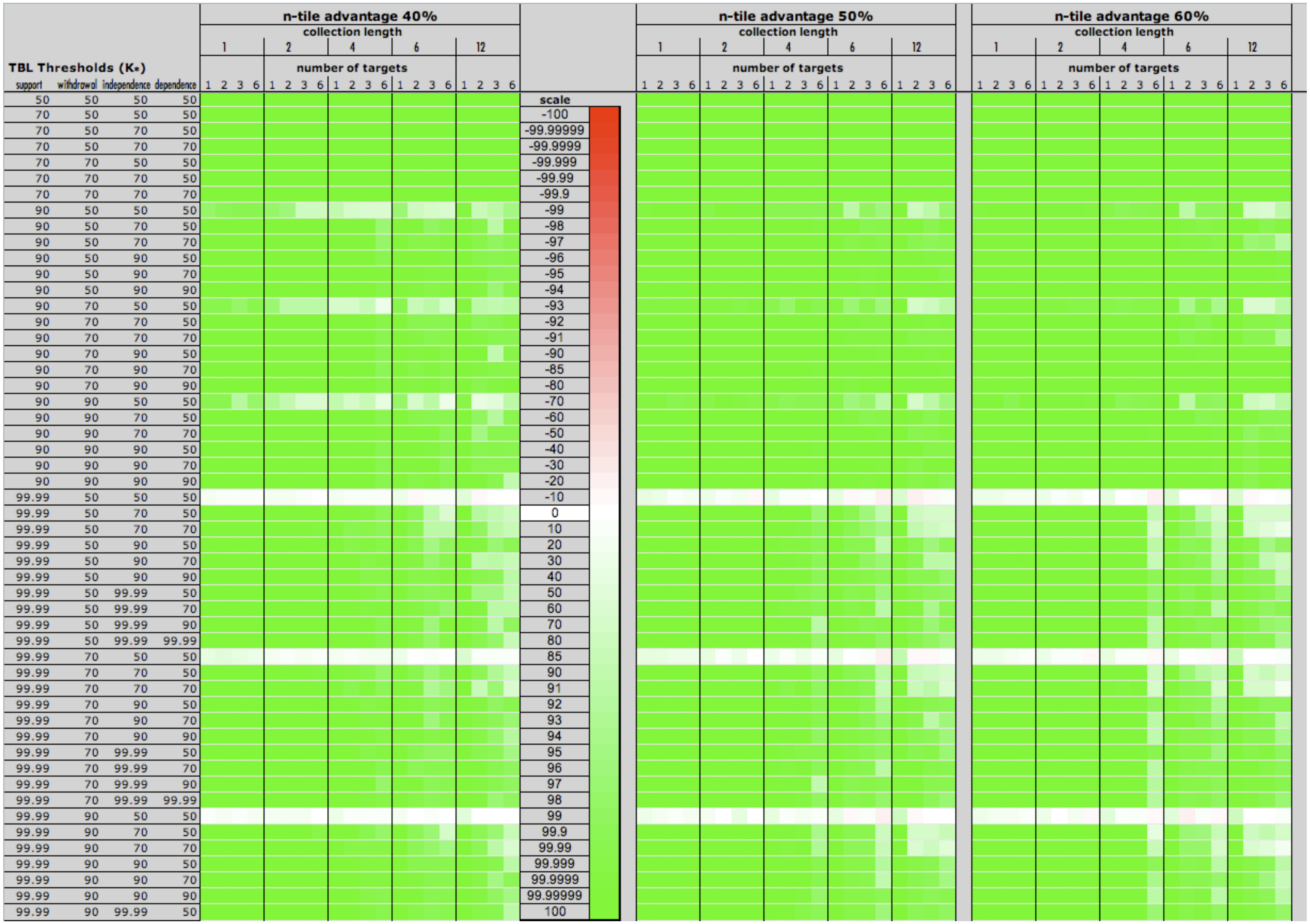
50000	🟥
48000	🟥
46000	🟥
44000	🟥
42000	🟥
40000	🟥
38000	🟥
36000	🟥
34000	🟥
32000	🟥
30000	🟥
28000	🟥
26000	🟥
24000	🟥
22000	🟥
20000	🟥
18000	🟥
16000	🟥
14000	🟥
12000	🟥
10000	🟥
8000	🟥
6000	🟥
4000	🟥
2000	🟥
0	
-2000	🟩
-4000	🟩
-6000	🟩
-8000	🟩
-10000	🟩
-12000	🟩
-14000	🟩
-16000	🟩
-18000	🟩
-20000	🟩
-22000	🟩
-24000	🟩
-26000	🟩
-28000	🟩
-30000	🟩
-32000	🟩
-34000	🟩
-36000	🟩
-38000	🟩
-40000	🟩
-42000	🟩
-44000	🟩
-46000	🟩
-48000	🟩
-50000	🟩

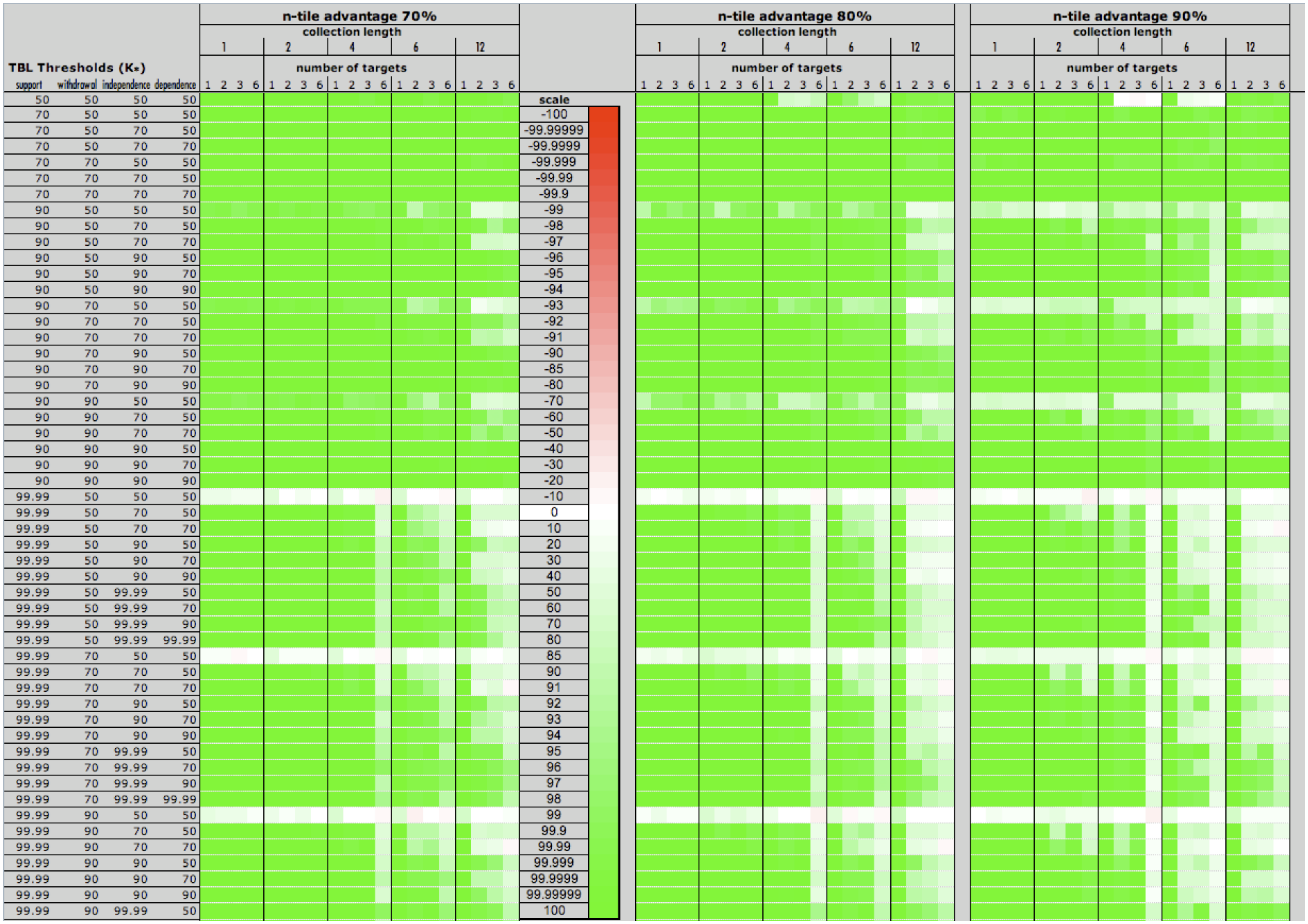

Expense
collection length
1
2
4
6
12
TBL Thresholds (K*)
number of targets
support withdrawal independence dependence
1 2 3 6 1 2 3 6 1 2 3 6 1 2 3 6 1 2 3 6
99.99 99.99 95 50
99.99 99.99 95 55
99.99 99.99 95 70
99.99 99.99 95 80
99.99 99.99 95 95
99.99 99.99 99.99 50
99.99 99.99 99.99 55
99.99 99.99 99.99 70
99.99 99.99 99.99 80
99.99 99.99 99.99 95
99.99 99.99 99.99 99.99
50000
48000
46000
44000
42000
40000
38000
36000
34000
32000
30000
28000
26000
24000
22000
20000
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
-2000
-4000
-6000
-8000
-10000
-12000
-14000
-16000
-18000
-20000
-22000
-24000
-26000
-28000
-30000
-32000
-34000
-36000
-38000
-40000
-42000
-44000
-46000
-48000
-50000

APPENDIX C: EXPERIMENT 1 *N*-TILE ADVANTAGE RESULTS

This section presents the full *n*-tile advantage results for Experiment 3. The results are sorted by TBL thresholds and are presented across the *n*-tiles and then down the TBL threshold factors.

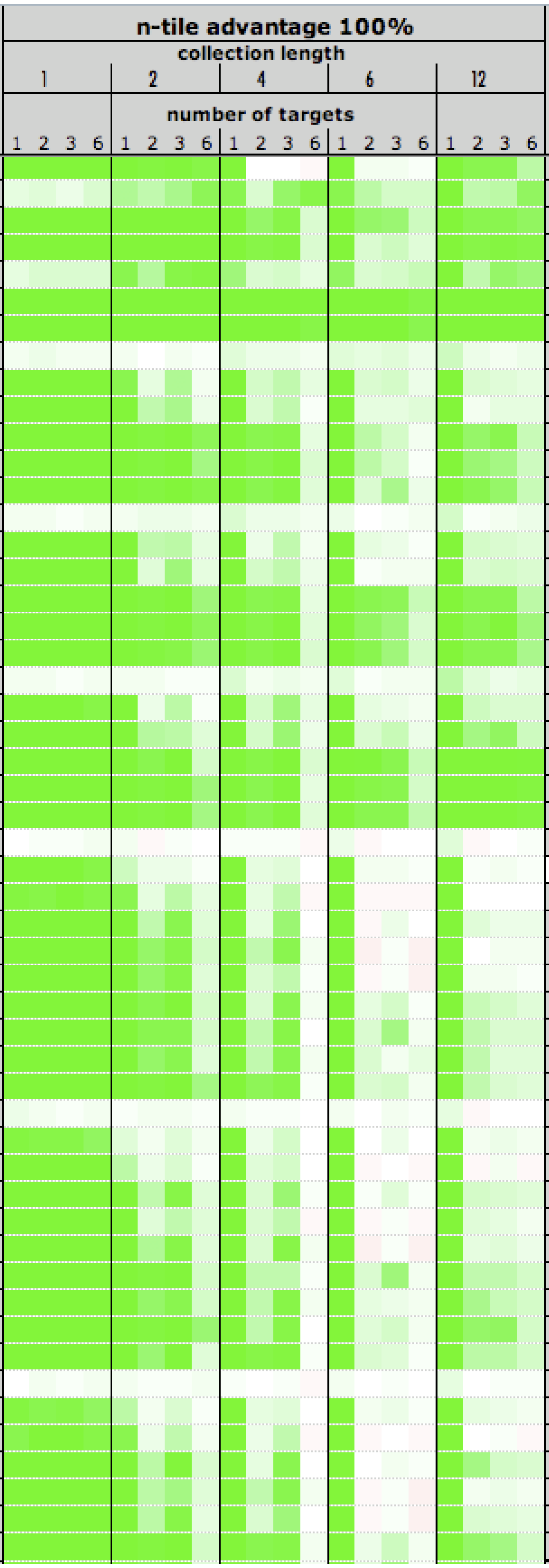

TBL Thresholds (K*)
support withdrawal independence dependence
n-tile advantage 10%
n-tile advantage 20%
n-tile advantage 30%
collection length
1 2 4 6 12
number of targets
1 2 3 6
scale
50 50 50 50
70 50 50 50
70 50 70 50
70 50 70 70
70 70 50 50
70 70 70 50
70 70 70 70
90 50 50 50
90 50 70 50
90 50 70 70
90 50 90 50
90 50 90 70
90 50 90 90
90 70 50 50
90 70 70 50
90 70 70 70
90 70 90 50
90 70 90 70
90 70 90 90
90 90 50 50
90 90 70 50
90 90 70 70
90 90 90 50
90 90 90 70
90 90 90 90
99.99 50 50 50
99.99 50 70 50
99.99 50 70 70
99.99 50 90 50
99.99 50 90 70
99.99 50 90 90
99.99 50 99.99 50
99.99 50 99.99 70
99.99 50 99.99 90
99.99 50 99.99 99.99
99.99 70 50 50
99.99 70 70 50
99.99 70 70 70
99.99 70 90 50
99.99 70 90 70
99.99 70 90 90
99.99 70 99.99 50
99.99 70 99.99 70
99.99 70 99.99 90
99.99 70 99.99 99.99
99.99 90 50 50
99.99 90 70 50
99.99 90 70 70
99.99 90 90 50
99.99 90 90 70
99.99 90 90 90
99.99 90 99.99 50
-100
-99.99999
-99.9999
-99.999
-99.99
-99.9
-99
-98
-97
-96
-95
-94
-93
-92
-91
-90
-85
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
85
90
91
92
93
94
95
96
97
98
99
99.9
99.99
99.999
99.9999
99.99999
100


n-tile advantage 40%
n-tile advantage 50%
n-tile advantage 60%
collection length
1
2
4
6
12
number of targets
1 2 3 6
TBL Thresholds (K*)
support withdrawal independence dependence
50 50 50 50
70 50 50 50
70 50 70 50
70 50 70 70
70 70 50 50
70 70 70 50
70 70 70 70
90 50 50 50
90 50 70 50
90 50 70 70
90 50 90 50
90 50 90 70
90 50 90 90
90 70 50 50
90 70 70 50
90 70 70 70
90 70 90 50
90 70 90 70
90 70 90 90
90 90 50 50
90 90 70 50
90 90 70 70
90 90 90 50
90 90 90 70
90 90 90 90
99.99 50 50 50
99.99 50 70 50
99.99 50 70 70
99.99 50 90 50
99.99 50 90 70
99.99 50 90 90
99.99 50 99.99 50
99.99 50 99.99 70
99.99 50 99.99 90
99.99 50 99.99 99.99
99.99 70 50 50
99.99 70 70 50
99.99 70 70 70
99.99 70 90 50
99.99 70 90 70
99.99 70 90 90
99.99 70 99.99 50
99.99 70 99.99 70
99.99 70 99.99 90
99.99 70 99.99 99.99
99.99 90 50 50
99.99 90 70 50
99.99 90 70 70
99.99 90 90 50
99.99 90 90 70
99.99 90 90 90
99.99 90 99.99 50
scale
-100
-99.99999
-99.9999
-99.999
-99.99
-99.9
-99
-98
-97
-96
-95
-94
-93
-92
-91
-90
-85
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
85
90
91
92
93
94
95
96
97
98
99
99.9
99.99
99.999
99.9999
99.99999
100



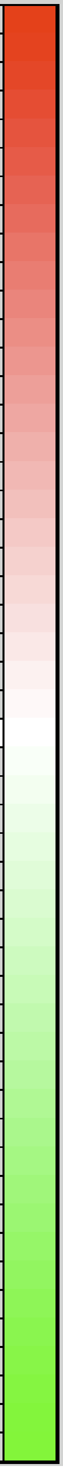
TBL Thresholds (K*)

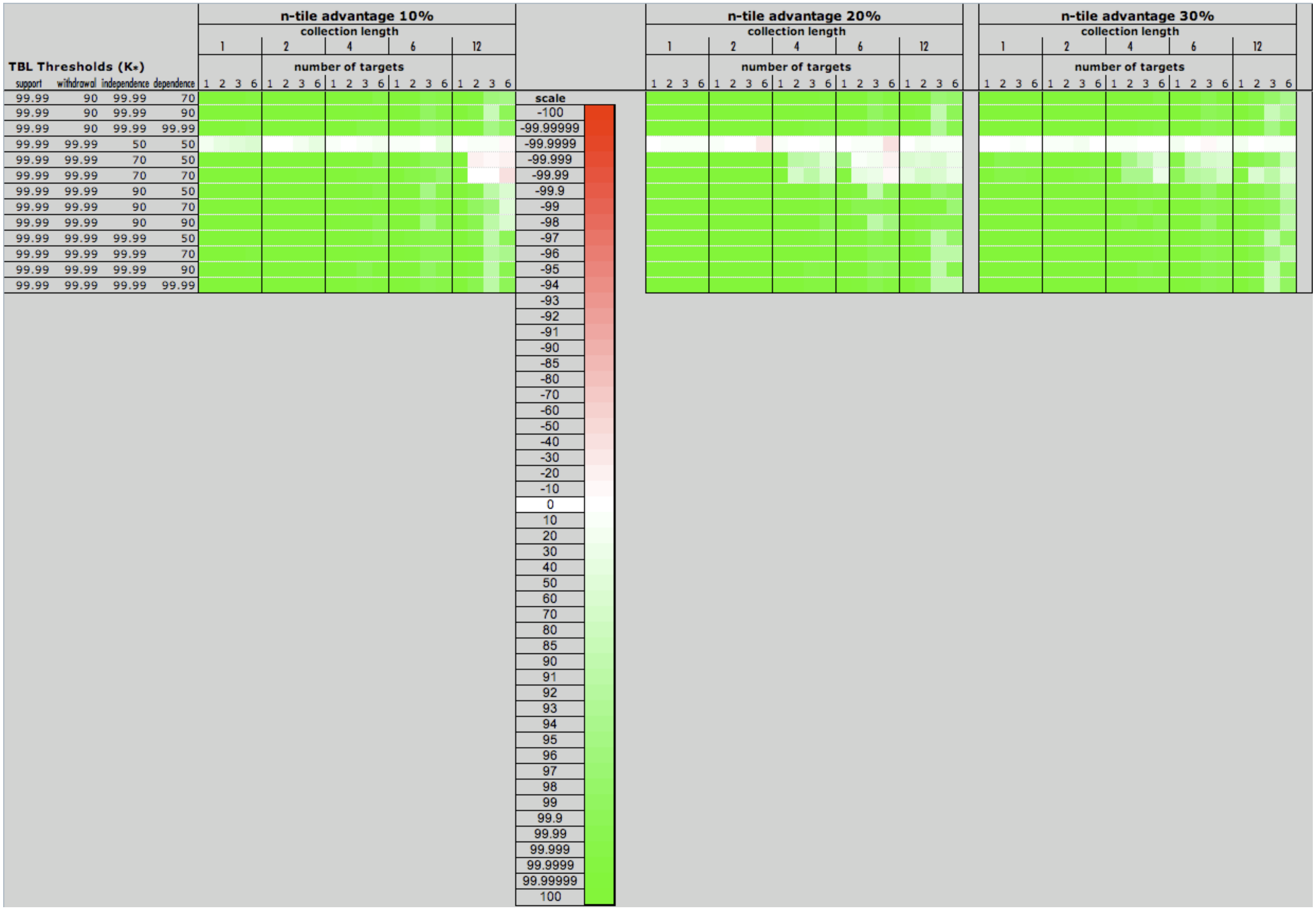
support	withdrawal	independence	dependence
50	50	50	50
70	50	50	50
70	50	70	50
70	50	70	70
70	70	50	50
70	70	70	50
70	70	70	70
90	50	50	50
90	50	70	50
90	50	70	70
90	50	90	50
90	50	90	70
90	50	90	90
90	70	50	50
90	70	70	50
90	70	70	70
90	70	90	50
90	70	90	70
90	70	90	90
90	90	50	50
90	90	70	50
90	90	70	70
90	90	90	50
90	90	90	70
90	90	90	90
99.99	50	50	50
99.99	50	70	50
99.99	50	70	70
99.99	50	90	50
99.99	50	90	70
99.99	50	90	90
99.99	50	99.99	50
99.99	50	99.99	70
99.99	50	99.99	90
99.99	50	99.99	99.99
99.99	70	50	50
99.99	70	70	50
99.99	70	70	70
99.99	70	90	50
99.99	70	90	70
99.99	70	90	90
99.99	70	99.99	50
99.99	70	99.99	70
99.99	70	99.99	90
99.99	70	99.99	99.99
99.99	90	50	50
99.99	90	70	50
99.99	90	70	70
99.99	90	90	50
99.99	90	90	70
99.99	90	90	90
99.99	90	99.99	50



TBL Thresholds (K*)

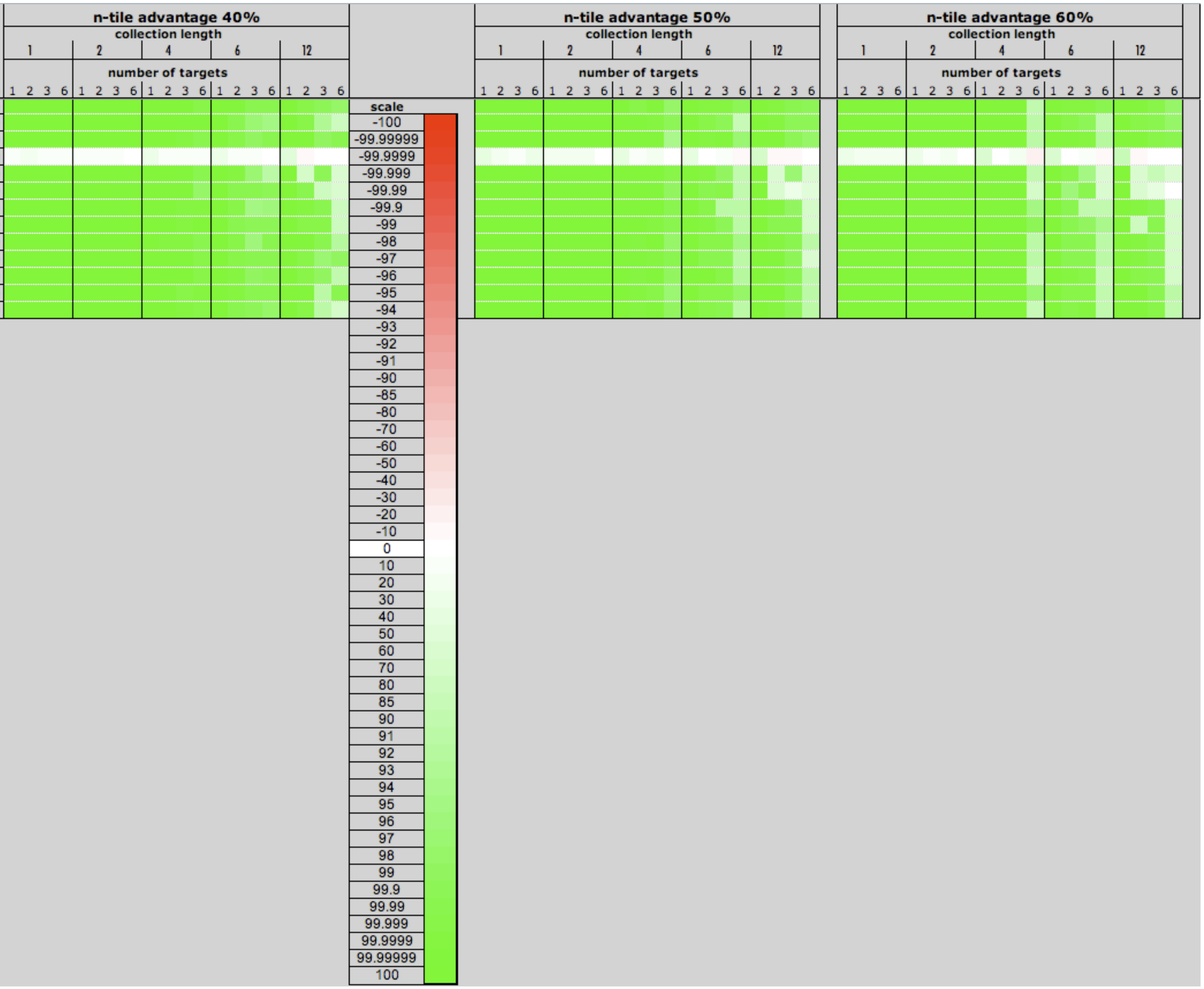
support	withdrawal	independence	dependence	scale
50	50	50	50	scale
70	50	50	50	-100
70	50	70	50	-99.99999
70	50	70	70	-99.9999
70	70	50	50	-99.999
70	70	70	50	-99.99
70	70	70	70	-99.9
90	50	50	50	-99
90	50	70	50	-98
90	50	70	70	-97
90	50	90	50	-96
90	50	90	70	-95
90	50	90	90	-94
90	70	50	50	-93
90	70	70	50	-92
90	70	70	70	-91
90	70	90	50	-90
90	70	90	70	-85
90	70	90	90	-80
90	90	50	50	-70
90	90	70	50	-60
90	90	70	70	-50
90	90	90	50	-40
90	90	90	70	-30
90	90	90	90	-20
99.99	50	50	50	-10
99.99	50	70	50	0
99.99	50	70	70	10
99.99	50	90	50	20
99.99	50	90	70	30
99.99	50	90	90	40
99.99	50	99.99	50	50
99.99	50	99.99	70	60
99.99	50	99.99	90	70
99.99	50	99.99	99.99	80
99.99	70	50	50	85
99.99	70	70	50	90
99.99	70	70	70	91
99.99	70	90	50	92
99.99	70	90	70	93
99.99	70	90	90	94
99.99	70	99.99	50	95
99.99	70	99.99	70	96
99.99	70	99.99	90	97
99.99	70	99.99	99.99	98
99.99	90	50	50	99
99.99	90	70	50	99.9
99.99	90	70	70	99.99
99.99	90	90	50	99.999
99.99	90	90	70	99.9999
99.99	90	90	90	99.99999
99.99	90	99.99	50	100

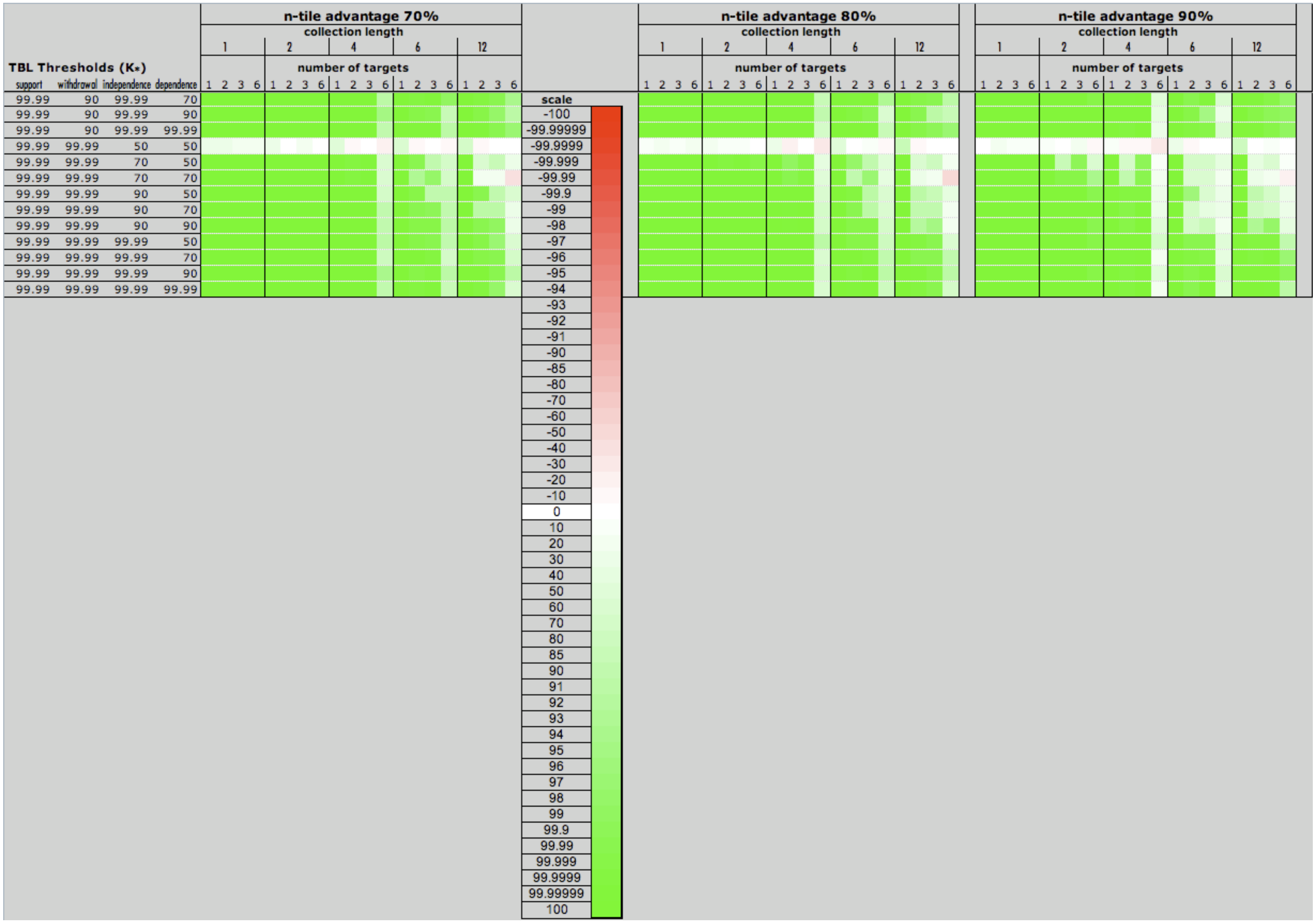




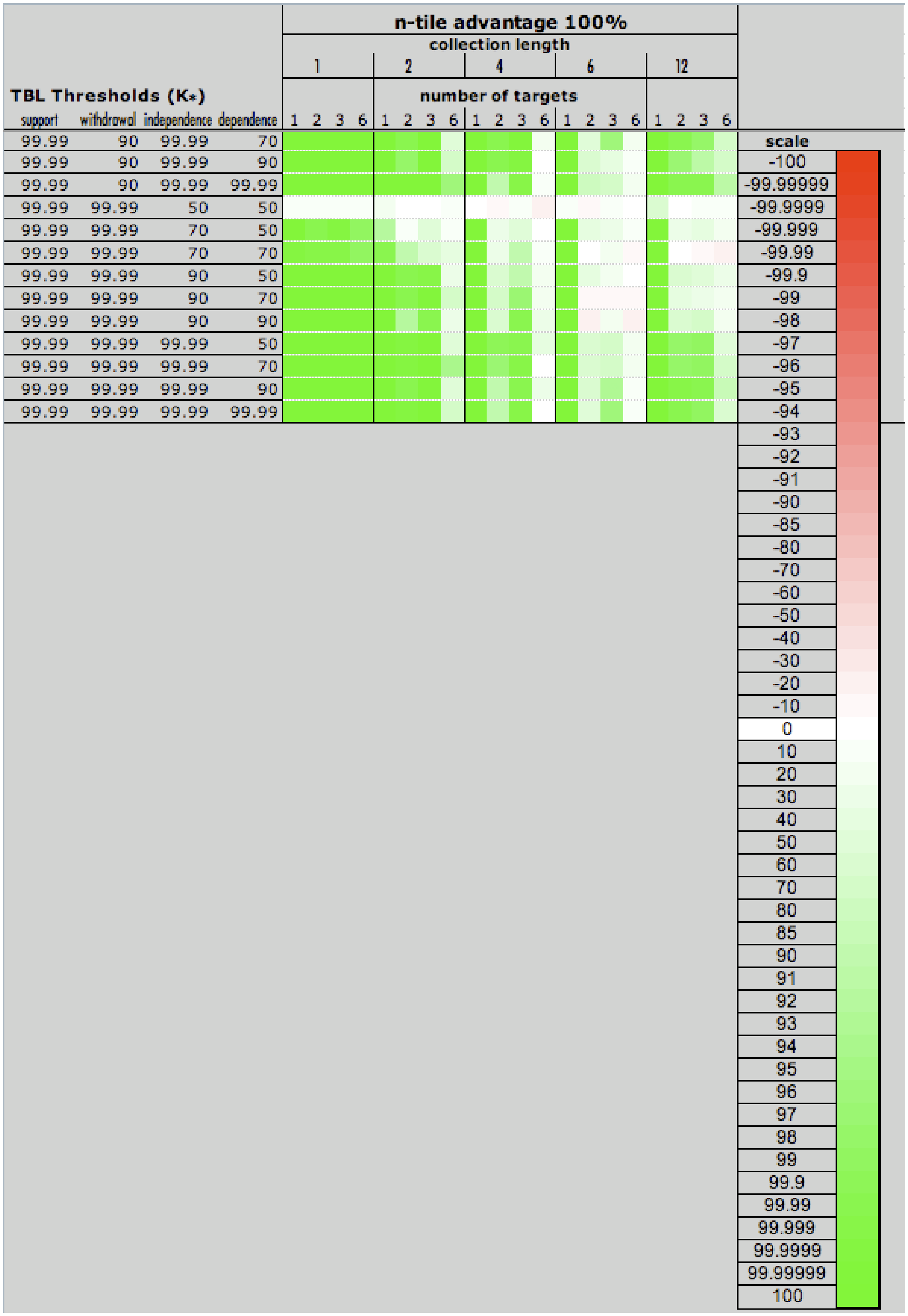
TBL Thresholds (K*)
support withdrawal independence dependence
99.99 90 99.99 70
99.99 90 99.99 90
99.99 90 99.99 99.99
99.99 99.99 50 50
99.99 99.99 70 50
99.99 99.99 70 70
99.99 99.99 90 50
99.99 99.99 90 70
99.99 99.99 90 90
99.99 99.99 99.99 50
99.99 99.99 99.99 70
99.99 99.99 99.99 90
99.99 99.99 99.99 99.99
n-tile advantage 10%
n-tile advantage 20%
n-tile advantage 30%
collection length
1
2
4
6
12
number of targets
1 2 3 6
scale
-100
-99.99999
-99.9999
-99.999
-99.99
-99.9
-99
-98
-97
-96
-95
-94
-93
-92
-91
-90
-85
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
85
90
91
92
93
94
95
96
97
98
99
99.9
99.99
99.999
99.9999
99.99999
100

TBL Thresholds (K*)			
support	withdrawal	independence	dependence
99.99	90	99.99	70
99.99	90	99.99	90
99.99	90	99.99	99.99
99.99	99.99	50	50
99.99	99.99	70	50
99.99	99.99	70	70
99.99	99.99	90	50
99.99	99.99	90	70
99.99	99.99	90	90
99.99	99.99	99.99	50
99.99	99.99	99.99	70
99.99	99.99	99.99	90
99.99	99.99	99.99	99.99



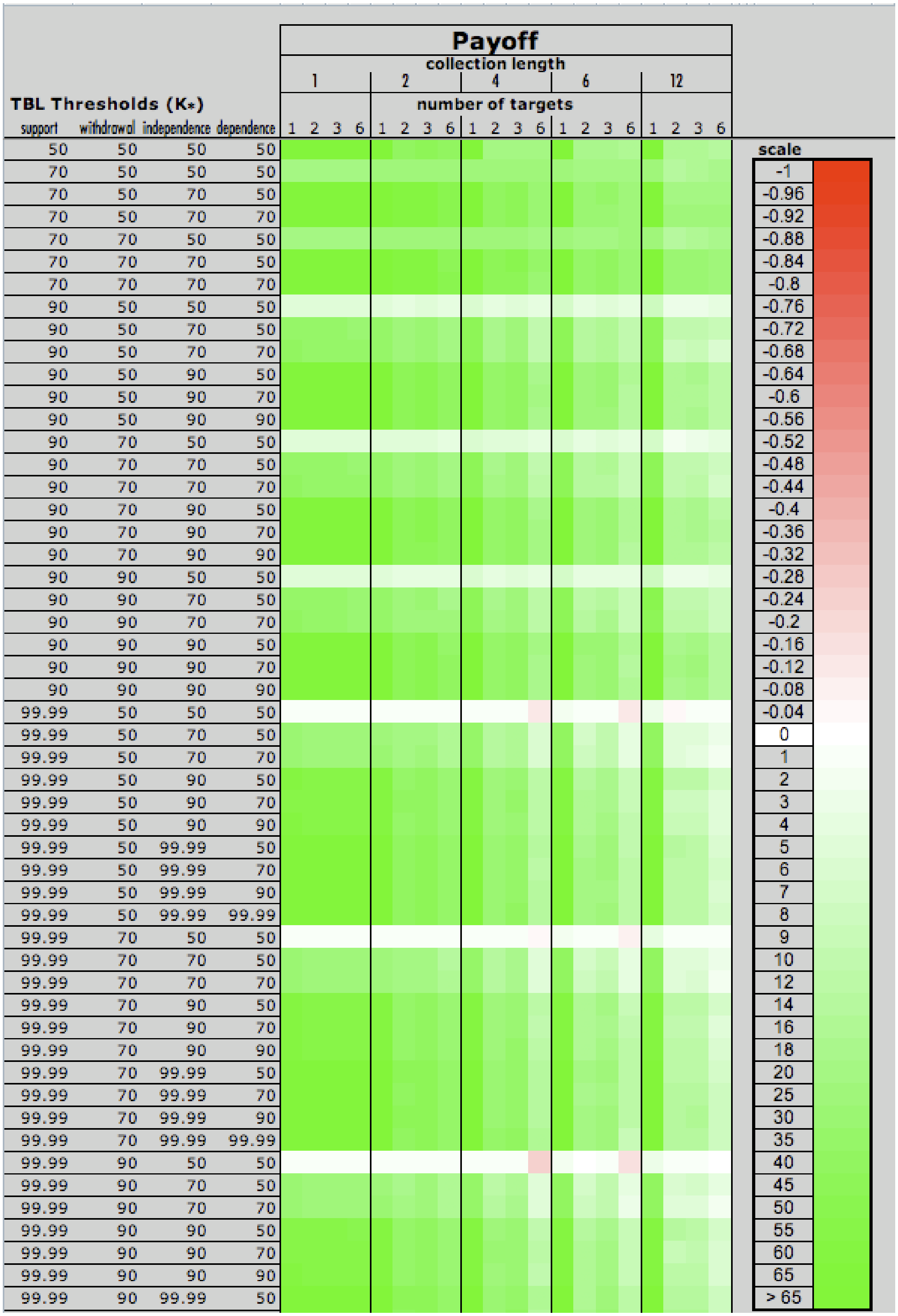


TBL Thresholds (K*)			
support	withdrawal	independence	dependence
99.99	90	99.99	70
99.99	90	99.99	90
99.99	90	99.99	99.99
99.99	99.99	50	50
99.99	99.99	70	50
99.99	99.99	70	70
99.99	99.99	90	50
99.99	99.99	90	70
99.99	99.99	90	90
99.99	99.99	99.99	50
99.99	99.99	99.99	70
99.99	99.99	99.99	90
99.99	99.99	99.99	99.99


n-tile advantage 100%
collection length
1
2
4
6
12
number of targets
1 2 3 6
1 2 3 6
1 2 3 6
1 2 3 6
1 2 3 6
TBL Thresholds (K*)
support withdrawal independence dependence
99.99 90 99.99 70
99.99 90 99.99 90
99.99 90 99.99 99.99
99.99 99.99 50 50
99.99 99.99 70 50
99.99 99.99 70 70
99.99 99.99 90 50
99.99 99.99 90 70
99.99 99.99 90 90
99.99 99.99 99.99 50
99.99 99.99 99.99 70
99.99 99.99 99.99 90
99.99 99.99 99.99 99.99
scale
-100
-99.99999
-99.9999
-99.999
-99.99
-99.9
-99
-98
-97
-96
-95
-94
-93
-92
-91
-90
-85
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
85
90
91
92
93
94
95
96
97
98
99
99.9
99.99
99.999
99.9999
99.99999
100

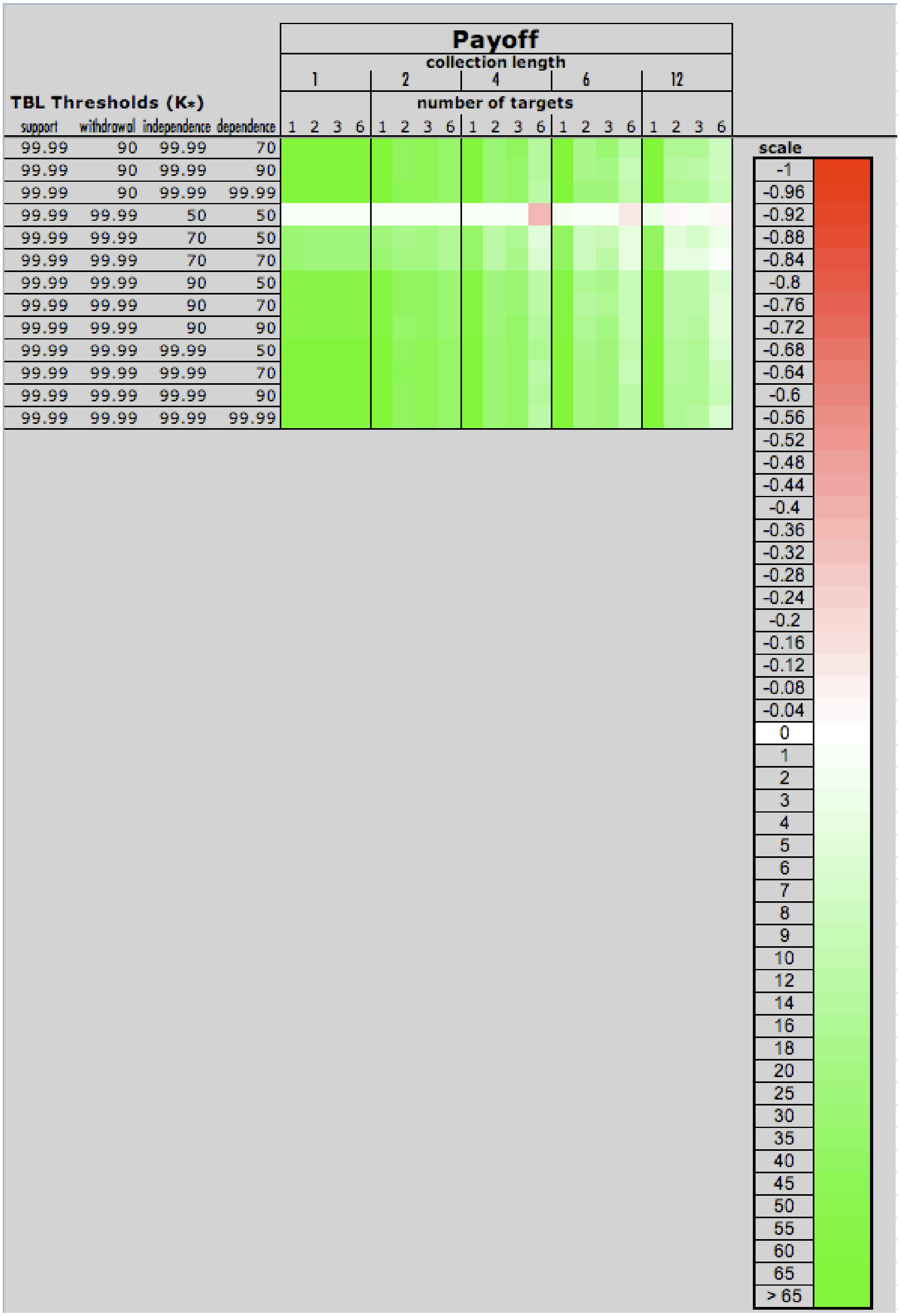
APPENDIX D: EXPERIMENT 3, PAYOFF RESULTS

This appendix contains the entire Payoff foot print for Experiment 3, sorted by TBL thresholds.



TBL Thresholds (K*)

support	withdrawal	independence	dependence
50	50	50	50
70	50	50	50
70	50	70	50
70	50	70	70
70	70	50	50
70	70	70	50
70	70	70	70
90	50	50	50
90	50	70	50
90	50	70	70
90	50	90	50
90	50	90	70
90	50	90	90
90	70	50	50
90	70	70	50
90	70	70	70
90	70	90	50
90	70	90	70
90	70	90	90
90	90	50	50
90	90	70	50
90	90	70	70
90	90	90	50
90	90	90	70
90	90	90	90
99.99	50	50	50
99.99	50	70	50
99.99	50	70	70
99.99	50	90	50
99.99	50	90	70
99.99	50	90	90
99.99	50	99.99	50
99.99	50	99.99	70
99.99	50	99.99	90
99.99	50	99.99	99.99
99.99	70	50	50
99.99	70	70	50
99.99	70	70	70
99.99	70	90	50
99.99	70	90	70
99.99	70	90	90
99.99	70	99.99	50
99.99	70	99.99	70
99.99	70	99.99	90
99.99	70	99.99	99.99
99.99	90	50	50
99.99	90	70	50
99.99	90	70	70
99.99	90	90	50
99.99	90	90	70
99.99	90	90	90
99.99	90	99.99	50



TBL Thresholds (K*)

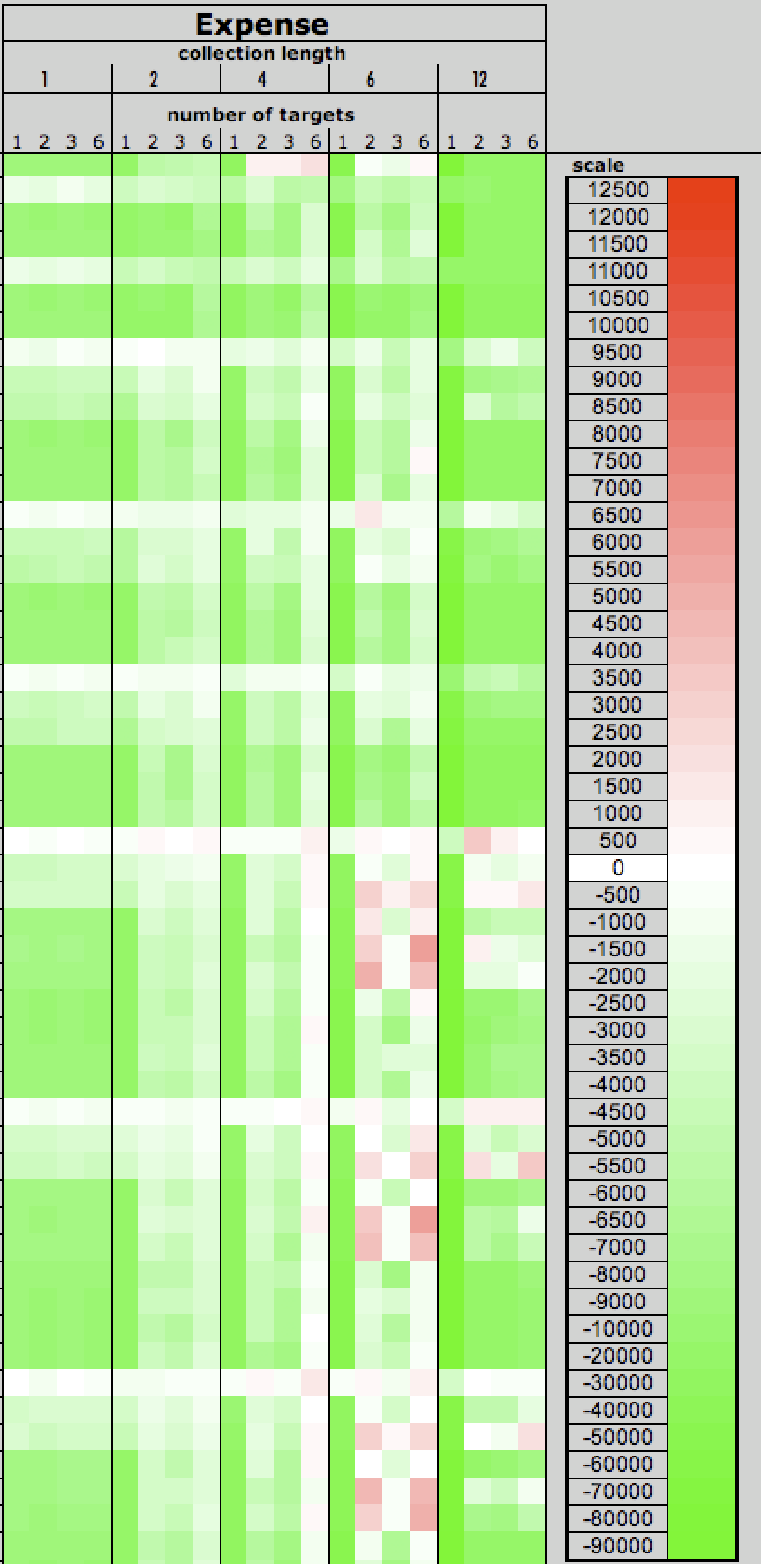
support	withdrawal	independence	dependence
99.99	90	99.99	70
99.99	90	99.99	90
99.99	90	99.99	99.99
99.99	99.99	50	50
99.99	99.99	70	50
99.99	99.99	70	70
99.99	99.99	90	50
99.99	99.99	90	70
99.99	99.99	90	90
99.99	99.99	99.99	50
99.99	99.99	99.99	70
99.99	99.99	99.99	90
99.99	99.99	99.99	99.99

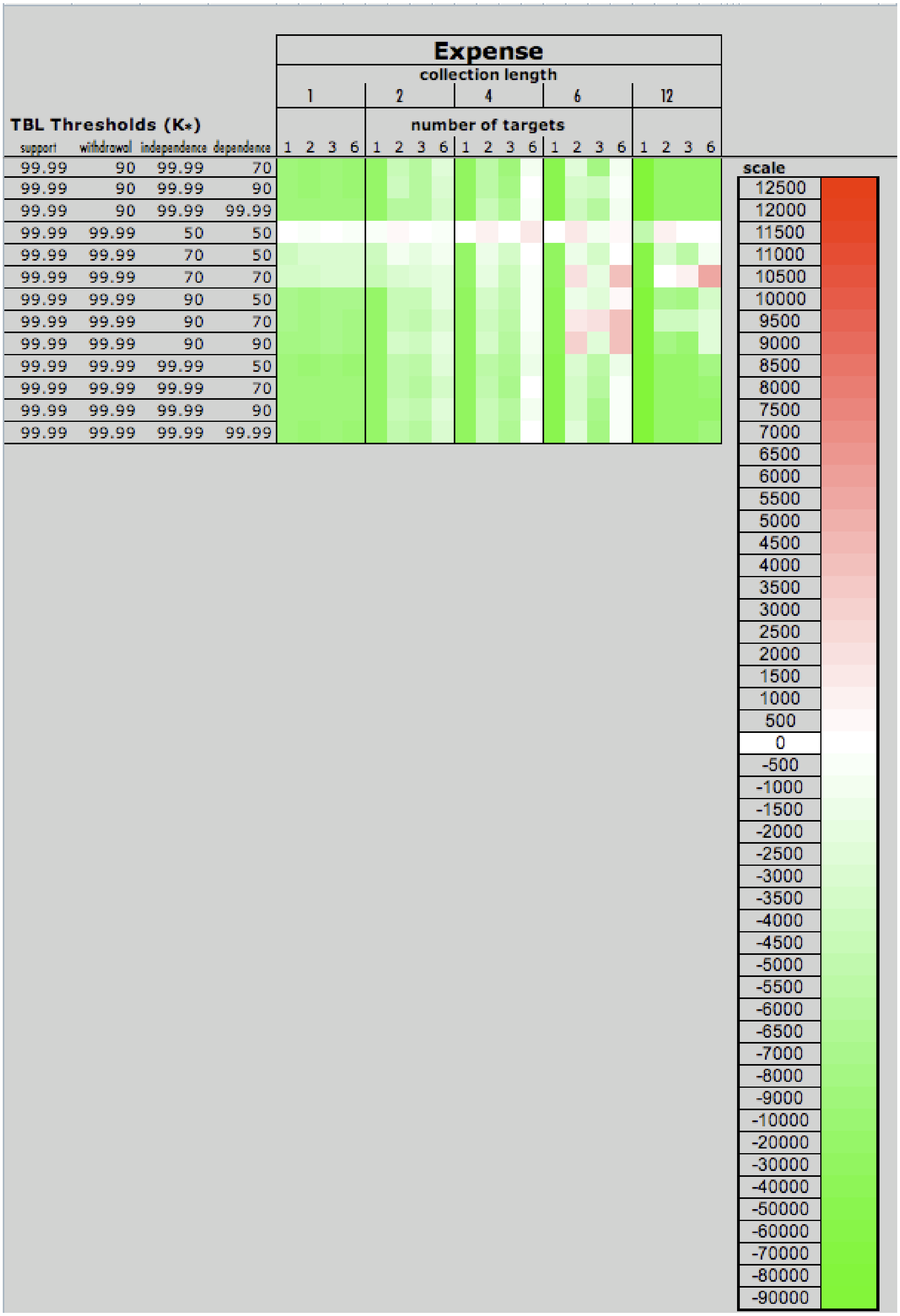
APPENDIX E: EXPERIMENT 3, EXPENSE RESULTS

This appendix contains the entire Expense footprint for Experiment 3, sorted by TBL thresholds.

TBL Thresholds (K*)

support	withdrawal	independence	dependence
50	50	50	50
70	50	50	50
70	50	70	50
70	50	70	70
70	70	50	50
70	70	70	50
70	70	70	70
90	50	50	50
90	50	70	50
90	50	70	70
90	50	90	50
90	50	90	70
90	50	90	90
90	70	50	50
90	70	70	50
90	70	70	70
90	70	90	50
90	70	90	70
90	70	90	90
90	90	50	50
90	90	70	50
90	90	70	70
90	90	90	50
90	90	90	70
90	90	90	90
99.99	50	50	50
99.99	50	70	50
99.99	50	70	70
99.99	50	90	50
99.99	50	90	70
99.99	50	90	90
99.99	50	99.99	50
99.99	50	99.99	70
99.99	50	99.99	90
99.99	50	99.99	99.99
99.99	70	50	50
99.99	70	70	50
99.99	70	70	70
99.99	70	90	50
99.99	70	90	70
99.99	70	90	90
99.99	70	99.99	50
99.99	70	99.99	70
99.99	70	99.99	90
99.99	70	99.99	99.99
99.99	90	50	50
99.99	90	70	50
99.99	90	70	70
99.99	90	90	50
99.99	90	90	70
99.99	90	90	90
99.99	90	99.99	50





Expense

collection length: 1, 2, 4, 6, 12

number of targets: 1 2 3 6 | 1 2 3 6 | 1 2 3 6 | 1 2 3 6 | 1 2 3 6

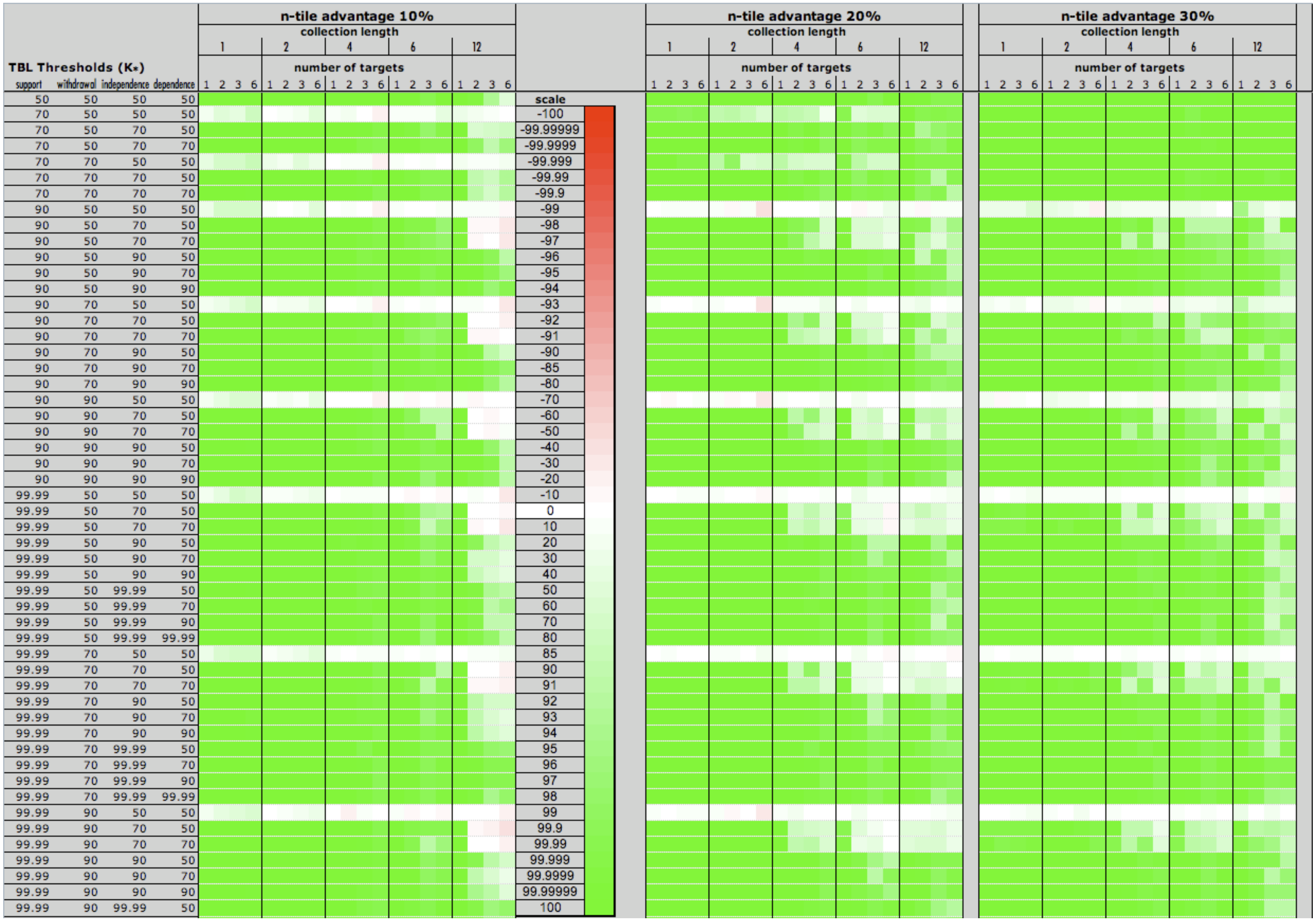
TBL Thresholds (K*)

support	withdrawal	independence	dependence
99.99	90	99.99	70
99.99	90	99.99	90
99.99	90	99.99	99.99
99.99	99.99	50	50
99.99	99.99	70	50
99.99	99.99	70	70
99.99	99.99	90	50
99.99	99.99	90	70
99.99	99.99	90	90
99.99	99.99	99.99	50
99.99	99.99	99.99	70
99.99	99.99	99.99	90
99.99	99.99	99.99	99.99

scale
12500
12000
11500
11000
10500
10000
9500
9000
8500
8000
7500
7000
6500
6000
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
-500
-1000
-1500
-2000
-2500
-3000
-3500
-4000
-4500
-5000
-5500
-6000
-6500
-7000
-8000
-9000
-10000
-20000
-30000
-40000
-50000
-60000
-70000
-80000
-90000

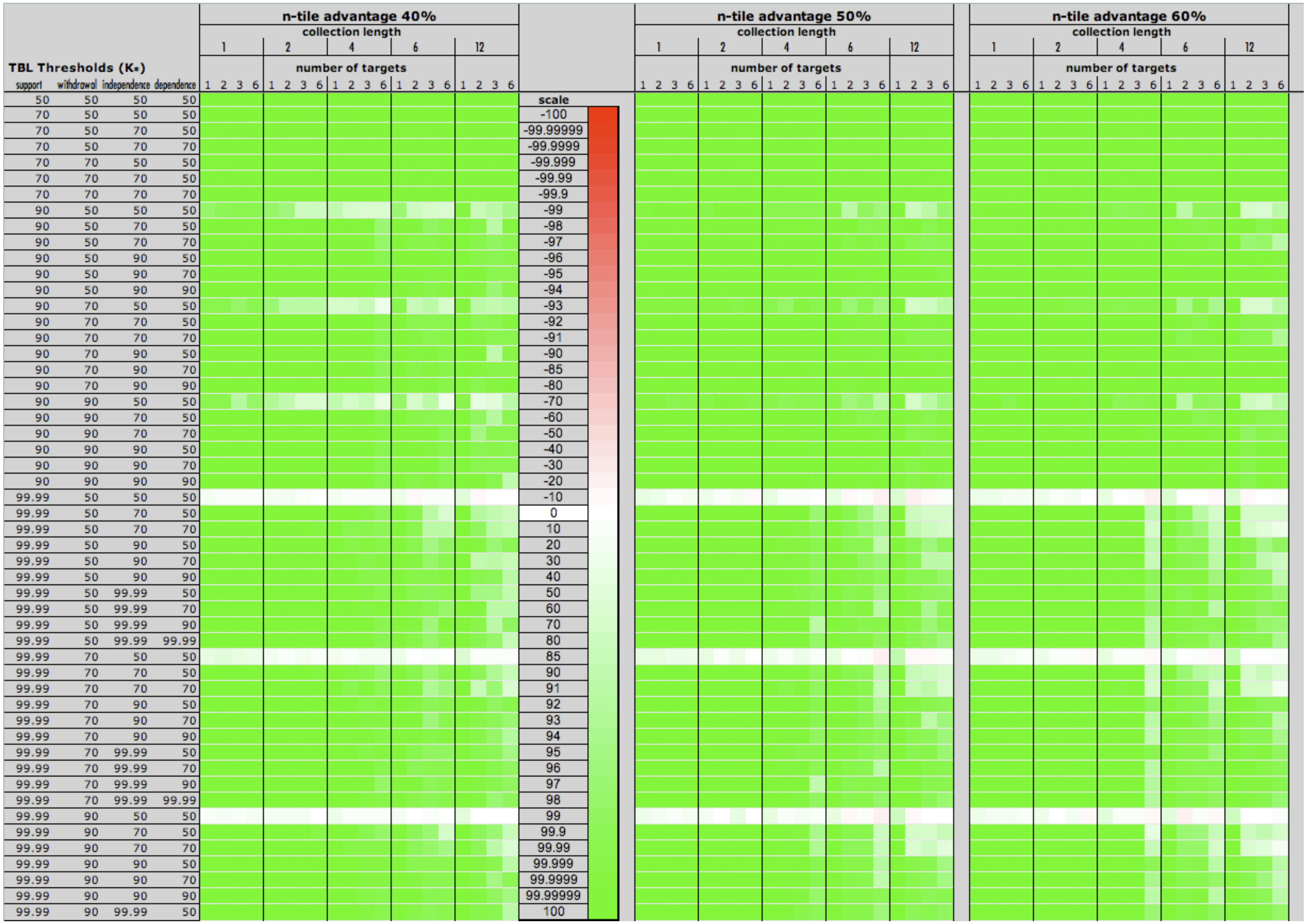
APPENDIX F: EXPERIMENT 3 *N*-TILE RESULTS

This section presents the full *n*-tile advantage results for Experiment 3. The results are sorted by TBL thresholds and are presented across the *n*-tiles and then down the TBL threshold factors.



TBL Thresholds (K*)

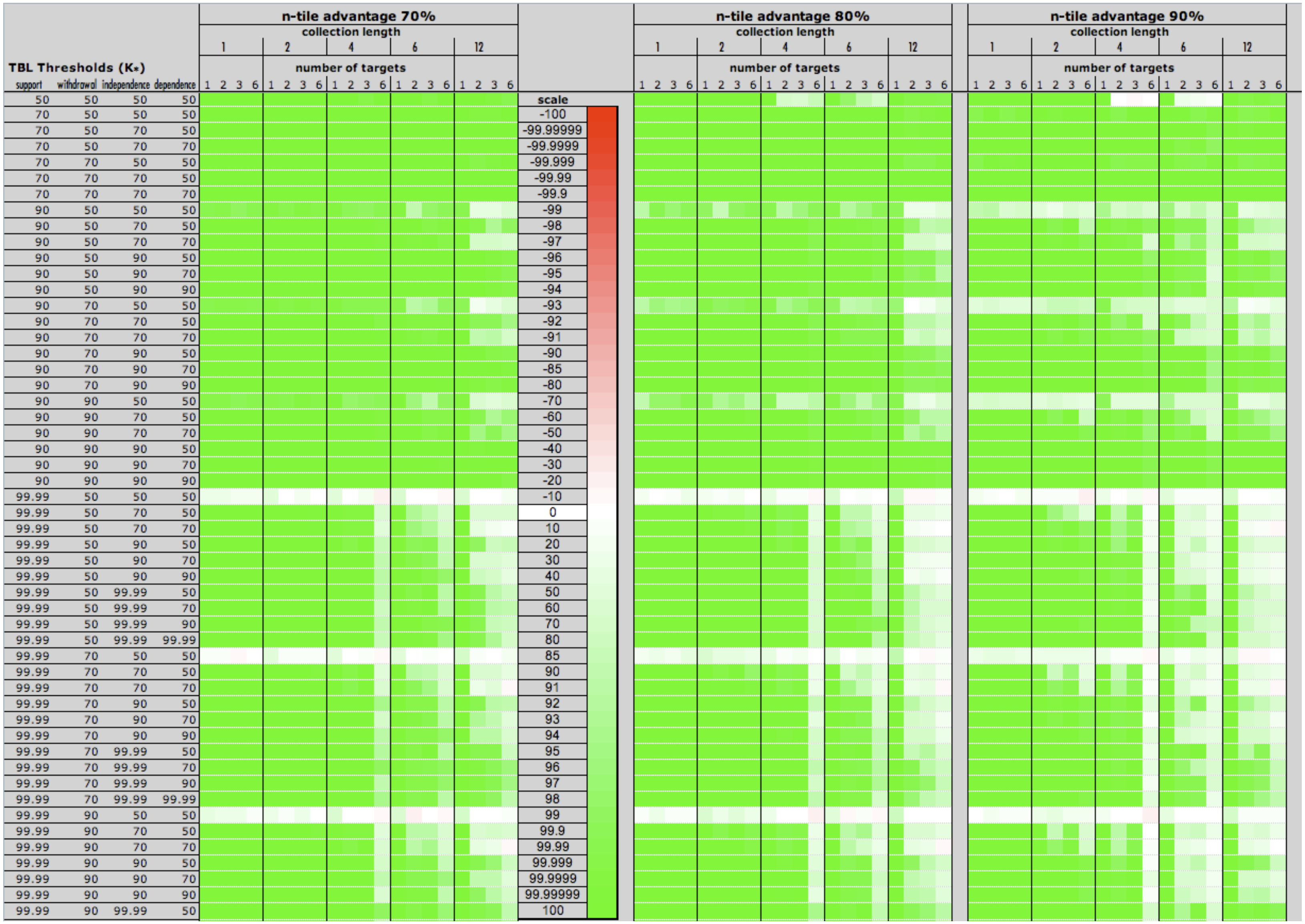
support	withdrawal	independence	dependence
50	50	50	50
70	50	50	50
70	50	70	50
70	50	70	70
70	70	50	50
70	70	70	50
70	70	70	70
90	50	50	50
90	50	70	50
90	50	70	70
90	50	90	50
90	50	90	70
90	50	90	90
90	70	50	50
90	70	70	50
90	70	70	70
90	70	90	50
90	70	90	70
90	70	90	90
90	90	50	50
90	90	70	50
90	90	70	70
90	90	90	50
90	90	90	70
90	90	90	90
99.99	50	50	50
99.99	50	70	50
99.99	50	70	70
99.99	50	90	50
99.99	50	90	70
99.99	50	90	90
99.99	50	99.99	50
99.99	50	99.99	70
99.99	50	99.99	90
99.99	50	99.99	99.99
99.99	70	50	50
99.99	70	70	50
99.99	70	70	70
99.99	70	90	50
99.99	70	90	70
99.99	70	90	90
99.99	70	99.99	50
99.99	70	99.99	70
99.99	70	99.99	90
99.99	70	99.99	99.99
99.99	90	50	50
99.99	90	70	50
99.99	90	70	70
99.99	90	90	50
99.99	90	90	70
99.99	90	90	90
99.99	90	99.99	50



TBL Thresholds (K*)

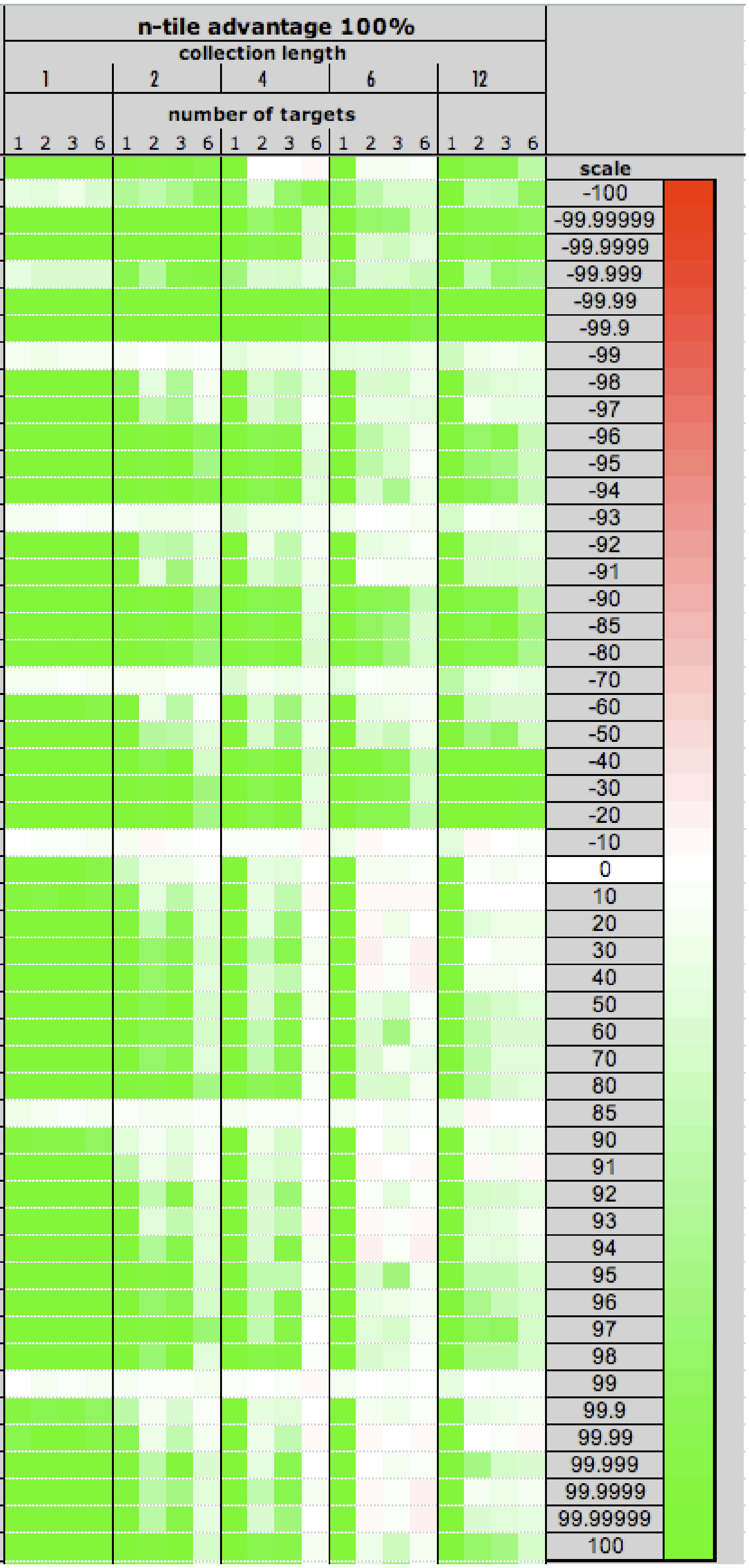
support	withdrawal	independence	dependence
50	50	50	50
70	50	50	50
70	50	70	50
70	50	70	70
70	70	50	50
70	70	70	50
70	70	70	70
90	50	50	50
90	50	70	50
90	50	70	70
90	50	90	50
90	50	90	70
90	50	90	90
90	70	50	50
90	70	70	50
90	70	70	70
90	70	90	50
90	70	90	70
90	70	90	90
90	90	50	50
90	90	70	50
90	90	70	70
90	90	90	50
90	90	90	70
90	90	90	90
99.99	50	50	50
99.99	50	70	50
99.99	50	70	70
99.99	50	90	50
99.99	50	90	70
99.99	50	90	90
99.99	50	99.99	50
99.99	50	99.99	70
99.99	50	99.99	90
99.99	50	99.99	99.99
99.99	70	50	50
99.99	70	70	50
99.99	70	70	70
99.99	70	90	50
99.99	70	90	70
99.99	70	90	90
99.99	70	99.99	50
99.99	70	99.99	70
99.99	70	99.99	90
99.99	70	99.99	99.99
99.99	90	50	50
99.99	90	70	50
99.99	90	70	70
99.99	90	90	50
99.99	90	90	70
99.99	90	90	90
99.99	90	99.99	50

scale
-100
-99.99999
-99.9999
-99.999
-99.99
-99.9
-99
-98
-97
-96
-95
-94
-93
-92
-91
-90
-85
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
85
90
91
92
93
94
95
96
97
98
99
99.9
99.99
99.999
99.9999
99.99999
100



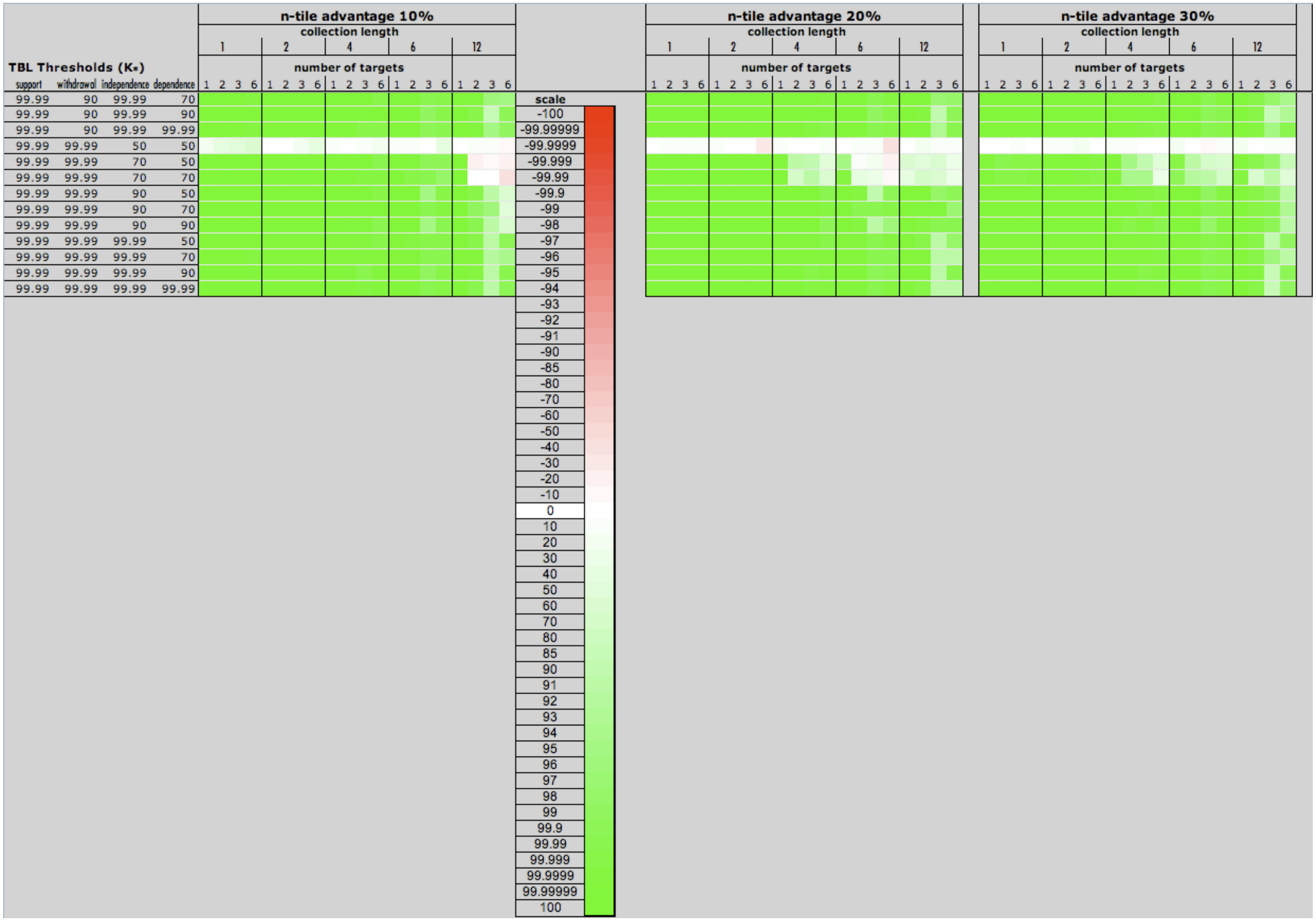
TBL Thresholds (K*)

support	withdrawal	independence	dependence
50	50	50	50
70	50	50	50
70	50	70	50
70	50	70	70
70	70	50	50
70	70	70	50
70	70	70	70
90	50	50	50
90	50	70	50
90	50	70	70
90	50	90	50
90	50	90	70
90	50	90	90
90	70	50	50
90	70	70	50
90	70	70	70
90	70	90	50
90	70	90	70
90	70	90	90
90	90	50	50
90	90	70	50
90	90	70	70
90	90	90	50
90	90	90	70
90	90	90	90
99.99	50	50	50
99.99	50	70	50
99.99	50	70	70
99.99	50	90	50
99.99	50	90	70
99.99	50	90	90
99.99	50	99.99	50
99.99	50	99.99	70
99.99	50	99.99	90
99.99	50	99.99	99.99
99.99	70	50	50
99.99	70	70	50
99.99	70	70	70
99.99	70	90	50
99.99	70	90	70
99.99	70	90	90
99.99	70	99.99	50
99.99	70	99.99	70
99.99	70	99.99	90
99.99	70	99.99	99.99
99.99	90	50	50
99.99	90	70	50
99.99	90	70	70
99.99	90	90	50
99.99	90	90	70
99.99	90	90	90
99.99	90	99.99	50



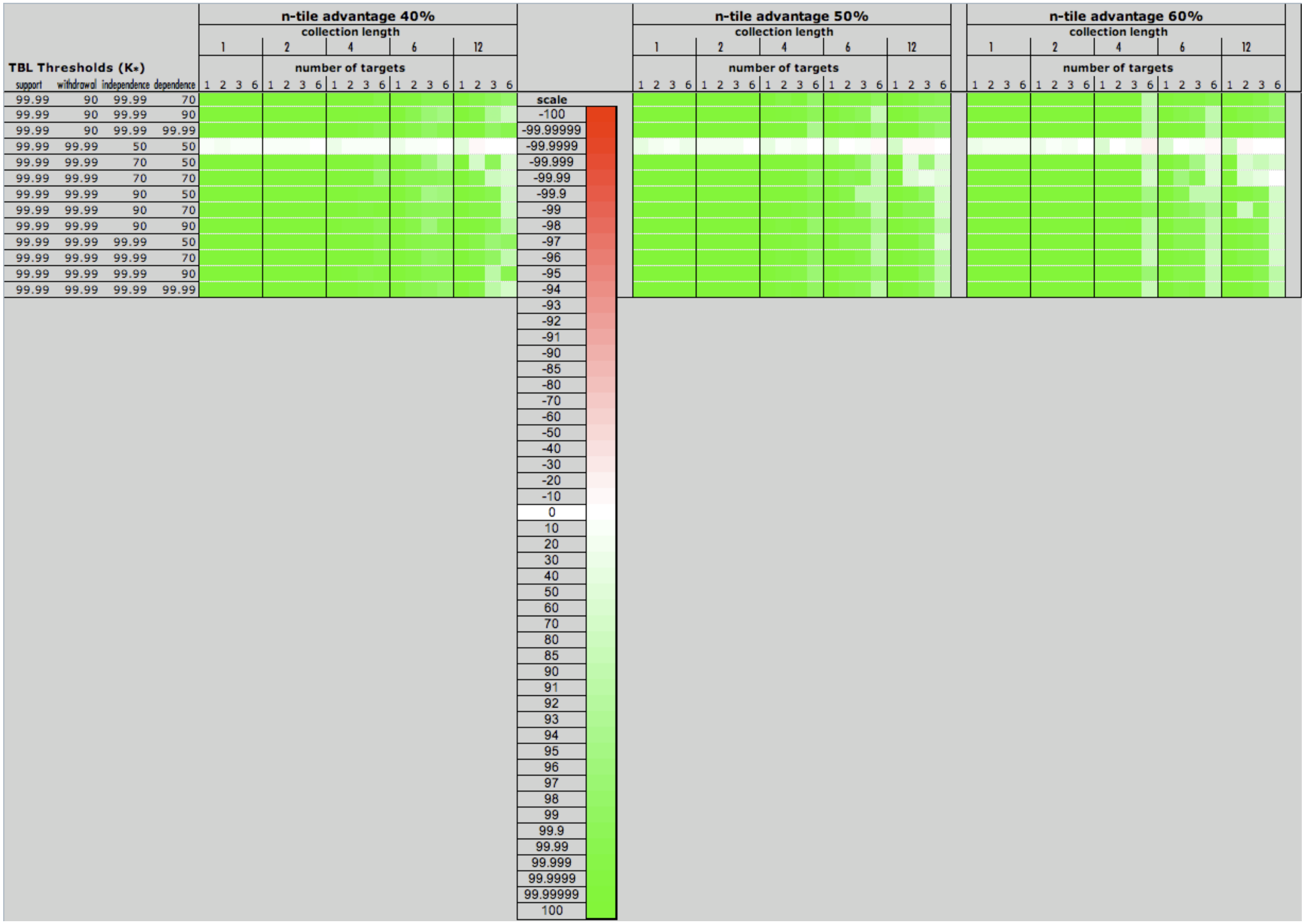
TBL Thresholds (K*)

support	withdrawal	independence	dependence
50	50	50	50
70	50	50	50
70	50	70	50
70	50	70	70
70	70	50	50
70	70	70	50
70	70	70	70
90	50	50	50
90	50	70	50
90	50	70	70
90	50	90	50
90	50	90	70
90	50	90	90
90	70	50	50
90	70	70	50
90	70	70	70
90	70	90	50
90	70	90	70
90	70	90	90
90	90	50	50
90	90	70	50
90	90	70	70
90	90	90	50
90	90	90	70
90	90	90	90
99.99	50	50	50
99.99	50	70	50
99.99	50	70	70
99.99	50	90	50
99.99	50	90	70
99.99	50	90	90
99.99	50	99.99	50
99.99	50	99.99	70
99.99	50	99.99	90
99.99	50	99.99	99.99
99.99	70	50	50
99.99	70	70	50
99.99	70	70	70
99.99	70	90	50
99.99	70	90	70
99.99	70	90	90
99.99	70	99.99	50
99.99	70	99.99	70
99.99	70	99.99	90
99.99	70	99.99	99.99
99.99	90	50	50
99.99	90	70	50
99.99	90	70	70
99.99	90	90	50
99.99	90	90	70
99.99	90	90	90
99.99	90	99.99	50



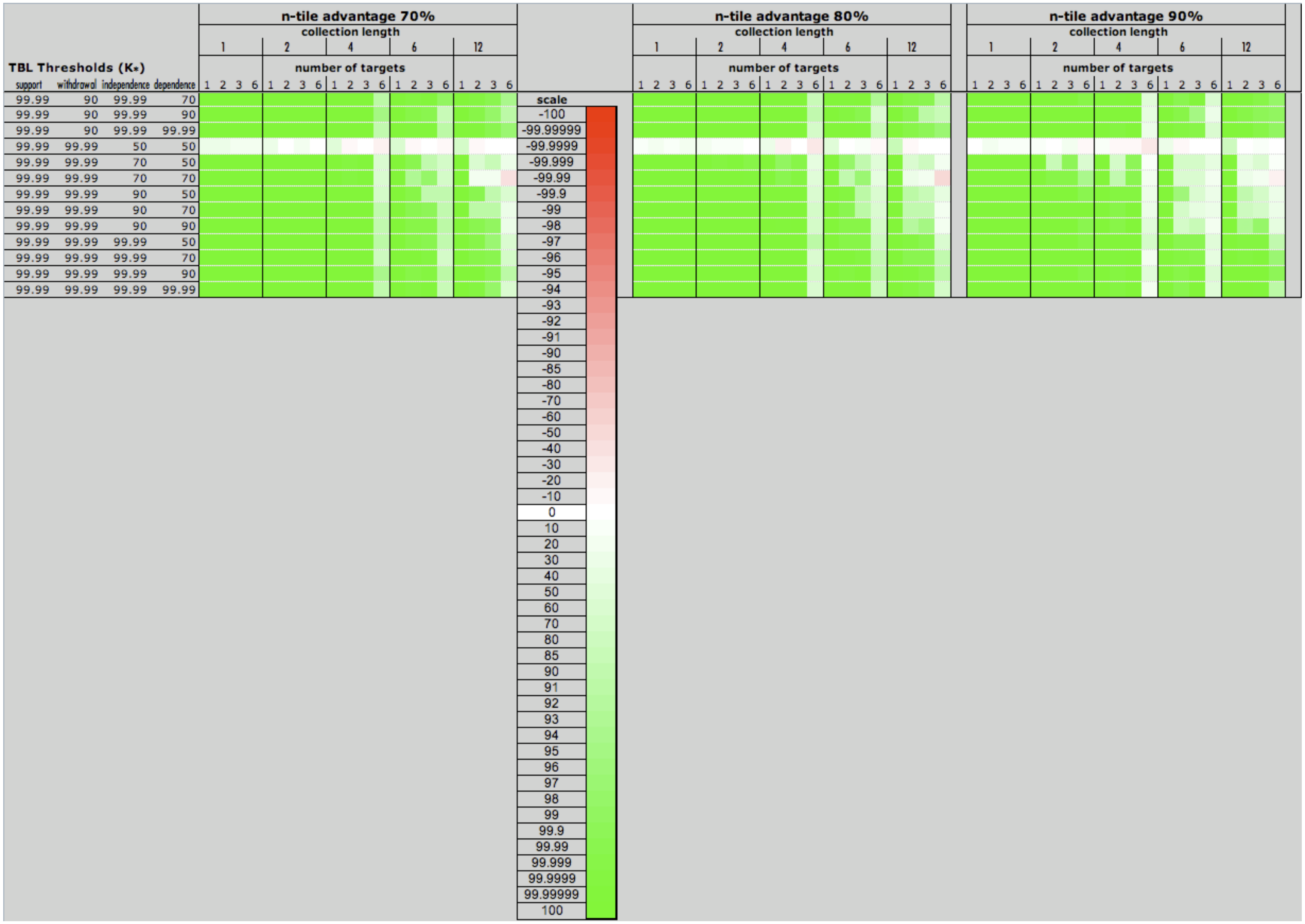
TBL Thresholds (K*)

support	withdrawal	independence	dependence
99.99	90	99.99	70
99.99	90	99.99	90
99.99	90	99.99	99.99
99.99	99.99	50	50
99.99	99.99	70	50
99.99	99.99	70	70
99.99	99.99	90	50
99.99	99.99	90	70
99.99	99.99	90	90
99.99	99.99	99.99	50
99.99	99.99	99.99	70
99.99	99.99	99.99	90
99.99	99.99	99.99	99.99



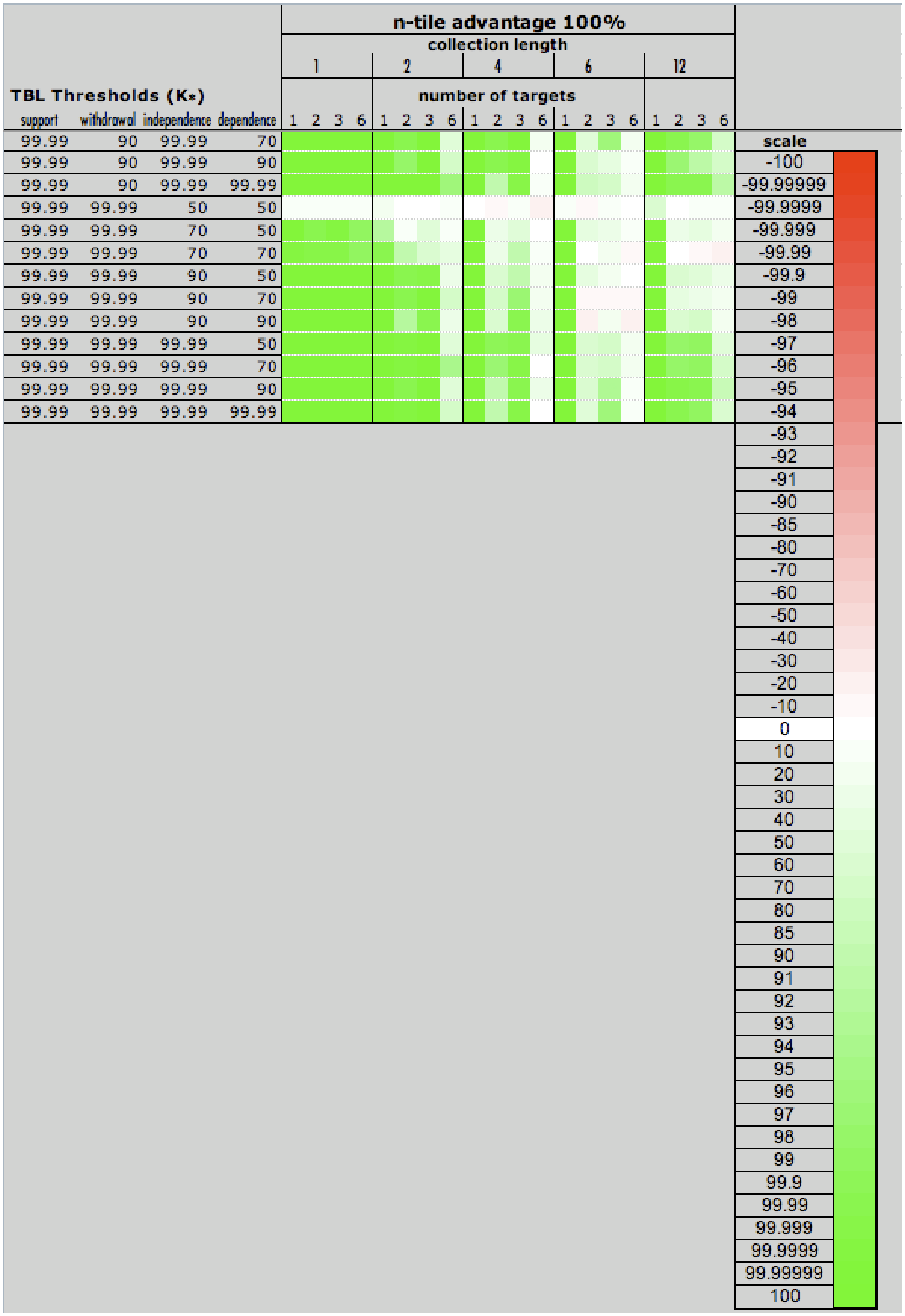
TBL Thresholds (K*)			
support	withdrawal	independence	dependence
99.99	90	99.99	70
99.99	90	99.99	90
99.99	90	99.99	99.99
99.99	99.99	50	50
99.99	99.99	70	50
99.99	99.99	70	70
99.99	99.99	90	50
99.99	99.99	90	70
99.99	99.99	90	90
99.99	99.99	99.99	50
99.99	99.99	99.99	70
99.99	99.99	99.99	90
99.99	99.99	99.99	99.99

scale
-100
-99.99999
-99.9999
-99.999
-99.99
-99.9
-99
-98
-97
-96
-95
-94
-93
-92
-91
-90
-85
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
85
90
91
92
93
94
95
96
97
98
99
99.9
99.99
99.999
99.9999
99.99999
100



TBL Thresholds (K*)

support	withdrawal	independence	dependence
99.99	90	99.99	70
99.99	90	99.99	90
99.99	90	99.99	99.99
99.99	99.99	50	50
99.99	99.99	70	50
99.99	99.99	70	70
99.99	99.99	90	50
99.99	99.99	90	70
99.99	99.99	90	90
99.99	99.99	99.99	50
99.99	99.99	99.99	70
99.99	99.99	99.99	90
99.99	99.99	99.99	99.99

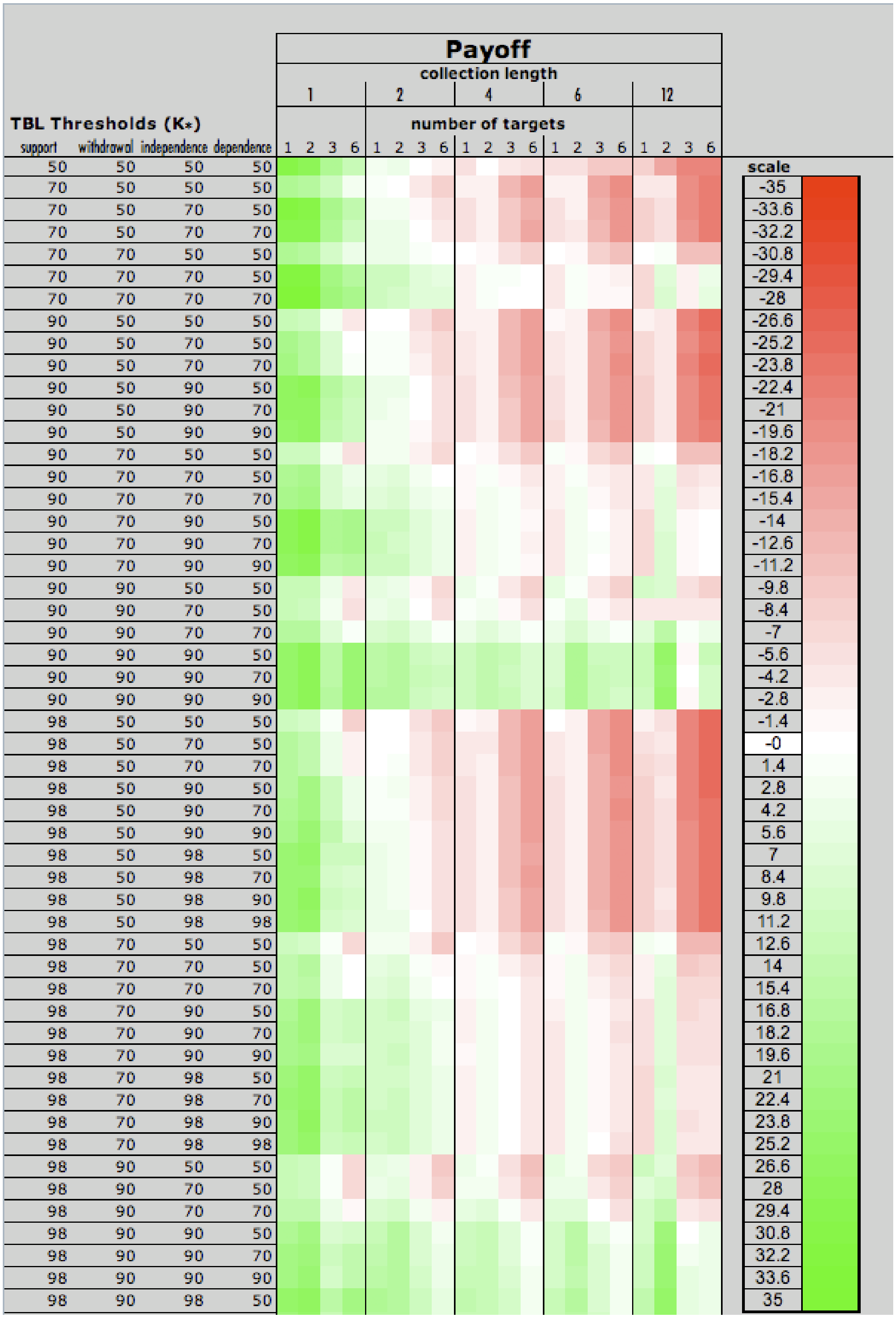


TBL Thresholds (K*)

support	withdrawal	independence	dependence
99.99	90	99.99	70
99.99	90	99.99	90
99.99	90	99.99	99.99
99.99	99.99	50	50
99.99	99.99	70	50
99.99	99.99	70	70
99.99	99.99	90	50
99.99	99.99	90	70
99.99	99.99	90	90
99.99	99.99	99.99	50
99.99	99.99	99.99	70
99.99	99.99	99.99	90
99.99	99.99	99.99	99.99

APPENDIX G: EXPERIMENT 5, PAYOFF RESULTS

This appendix contains the entire Payoff footprint for Experiment 5, sorted by TBL thresholds.



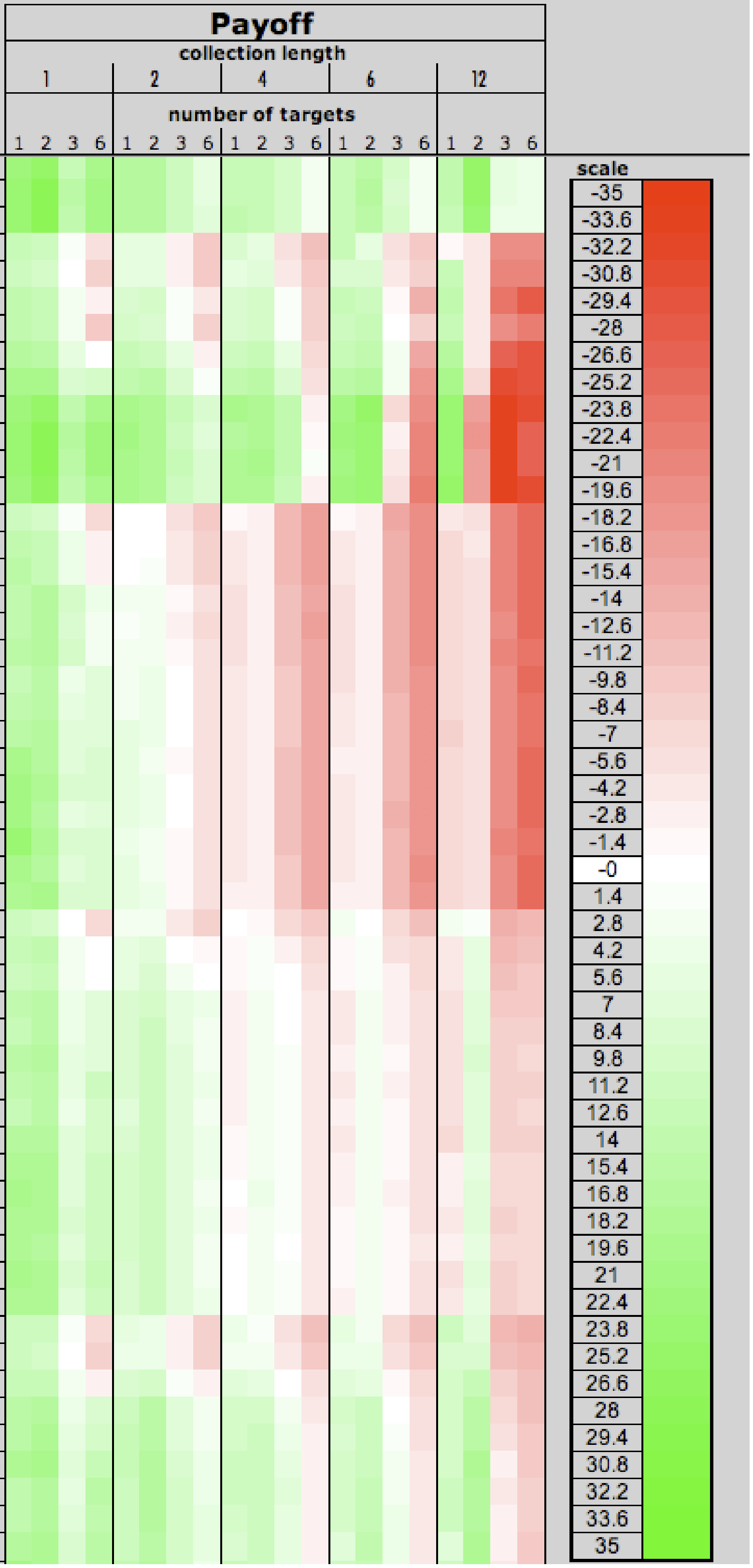
TBL Thresholds (K*)

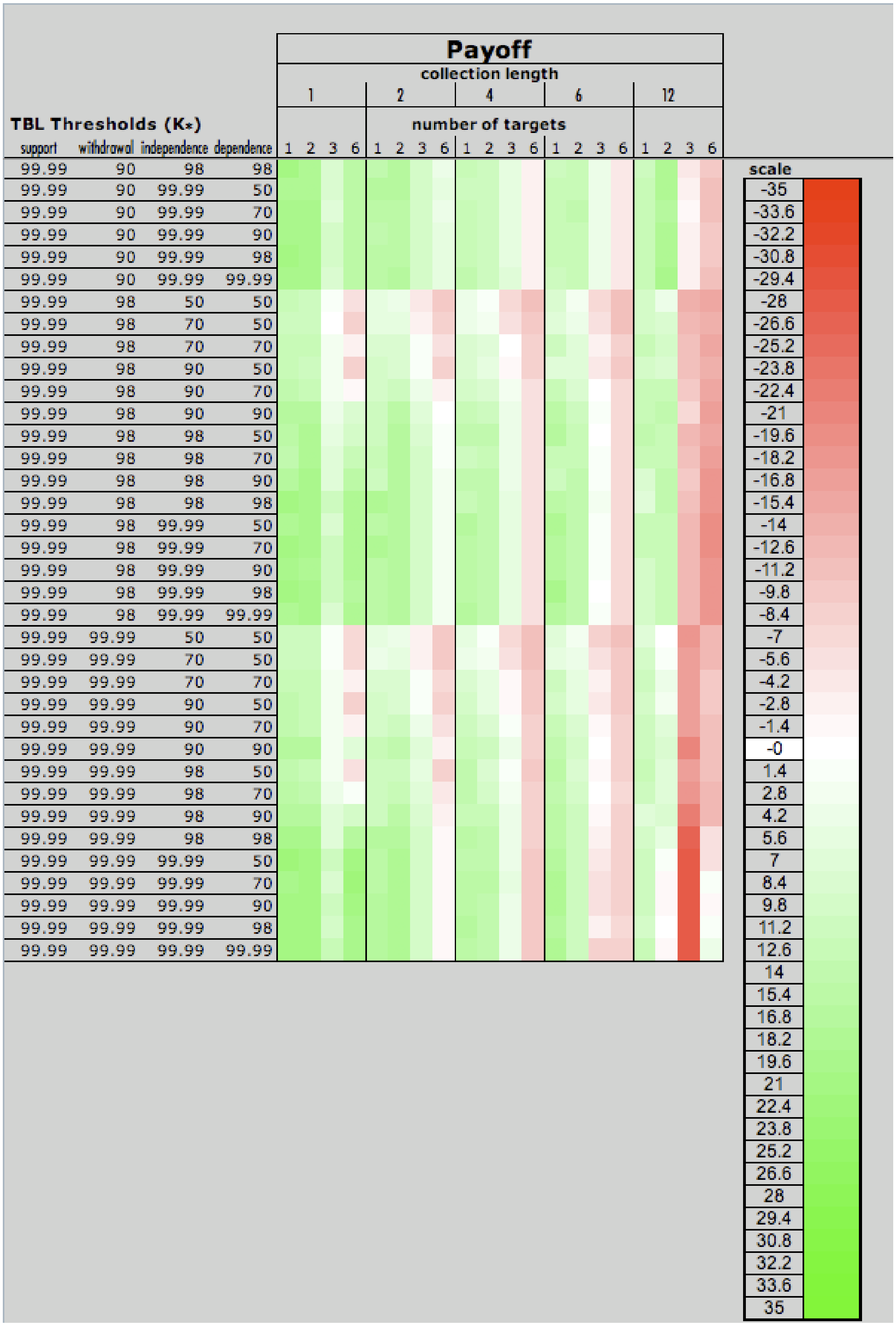
support	withdrawal	independence	dependence
50	50	50	50
70	50	50	50
70	50	70	50
70	50	70	70
70	70	50	50
70	70	70	50
70	70	70	70
90	50	50	50
90	50	70	50
90	50	70	70
90	50	90	50
90	50	90	70
90	50	90	90
90	70	50	50
90	70	70	50
90	70	70	70
90	70	90	50
90	70	90	70
90	70	90	90
90	90	50	50
90	90	70	50
90	90	70	70
90	90	90	50
90	90	90	70
90	90	90	90
98	50	50	50
98	50	70	50
98	50	70	70
98	50	90	50
98	50	90	70
98	50	90	90
98	50	98	50
98	50	98	70
98	50	98	90
98	50	98	98
98	70	50	50
98	70	70	50
98	70	70	70
98	70	90	50
98	70	90	70
98	70	90	90
98	70	98	50
98	70	98	70
98	70	98	90
98	70	98	98
98	90	50	50
98	90	70	50
98	90	70	70
98	90	90	50
98	90	90	70
98	90	90	90
98	90	98	50

scale
-35
-33.6
-32.2
-30.8
-29.4
-28
-26.6
-25.2
-23.8
-22.4
-21
-19.6
-18.2
-16.8
-15.4
-14
-12.6
-11.2
-9.8
-8.4
-7
-5.6
-4.2
-2.8
-1.4
-0
1.4
2.8
4.2
5.6
7
8.4
9.8
11.2
12.6
14
15.4
16.8
18.2
19.6
21
22.4
23.8
25.2
26.6
28
29.4
30.8
32.2
33.6
35

TBL Thresholds (K*)

support	withdrawal	independence	dependence
98	90	98	70
98	90	98	90
98	90	98	98
98	98	50	50
98	98	70	50
98	98	70	70
98	98	90	50
98	98	90	70
98	98	90	90
98	98	98	50
98	98	98	70
98	98	98	90
98	98	98	98
99.99	50	50	50
99.99	50	70	50
99.99	50	70	70
99.99	50	90	50
99.99	50	90	70
99.99	50	90	90
99.99	50	98	50
99.99	50	98	70
99.99	50	98	90
99.99	50	98	98
99.99	50	99.99	50
99.99	50	99.99	70
99.99	50	99.99	90
99.99	50	99.99	98
99.99	50	99.99	99.99
99.99	70	50	50
99.99	70	70	50
99.99	70	70	70
99.99	70	90	50
99.99	70	90	70
99.99	70	90	90
99.99	70	98	50
99.99	70	98	70
99.99	70	98	90
99.99	70	98	98
99.99	70	99.99	50
99.99	70	99.99	70
99.99	70	99.99	90
99.99	70	99.99	98
99.99	70	99.99	99.99
99.99	90	50	50
99.99	90	70	50
99.99	90	70	70
99.99	90	90	50
99.99	90	90	70
99.99	90	90	90
99.99	90	98	50
99.99	90	98	70
99.99	90	98	90





Payoff

collection length: 1, 2, 4, 6, 12

number of targets: 1 2 3 6 | 1 2 3 6 | 1 2 3 6 | 1 2 3 6 | 1 2 3 6

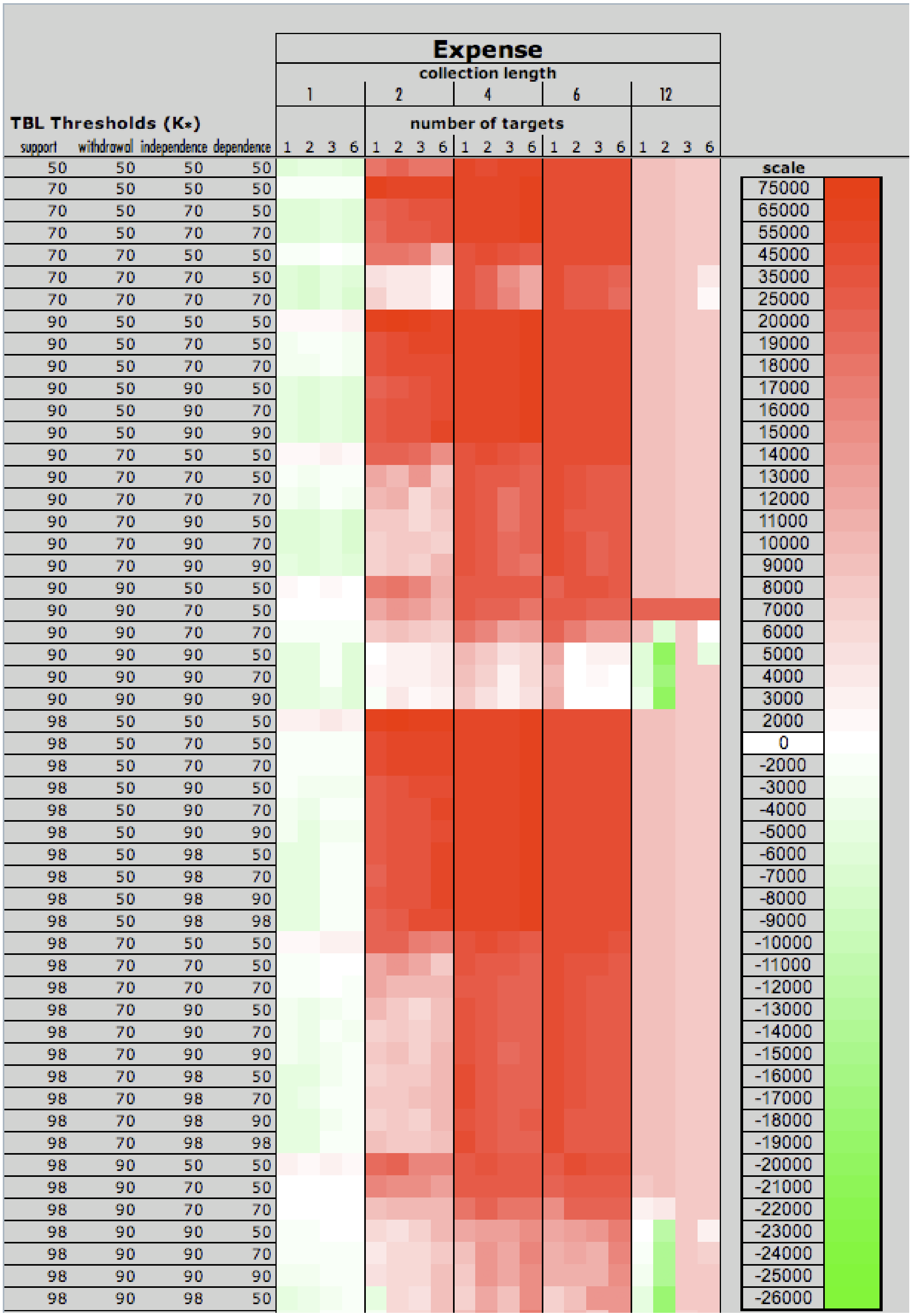
TBL Thresholds (K*)

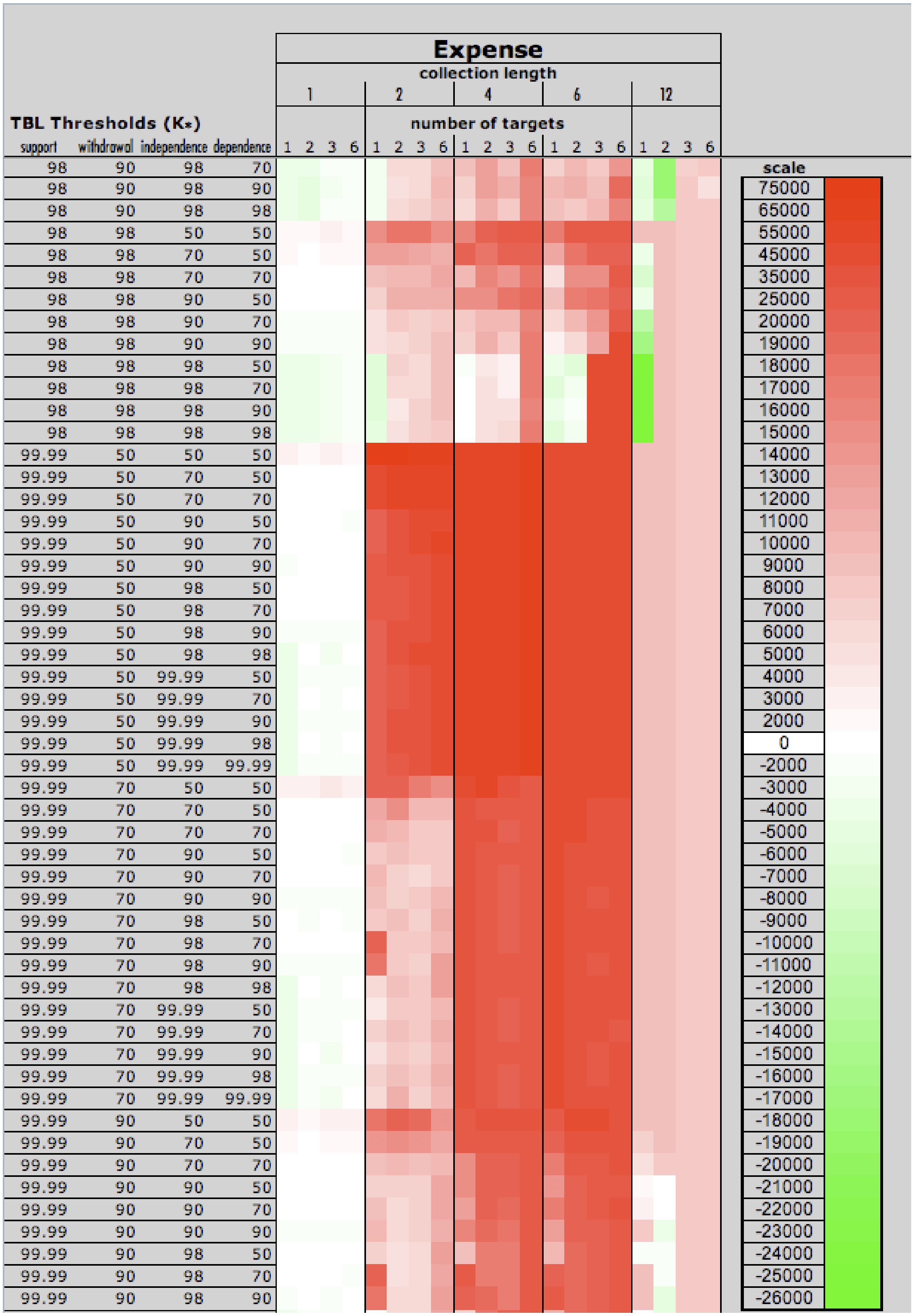
support	withdrawal	independence	dependence
99.99	90	98	98
99.99	90	99.99	50
99.99	90	99.99	70
99.99	90	99.99	90
99.99	90	99.99	98
99.99	90	99.99	99.99
99.99	98	50	50
99.99	98	70	50
99.99	98	70	70
99.99	98	90	50
99.99	98	90	70
99.99	98	90	90
99.99	98	98	50
99.99	98	98	70
99.99	98	98	90
99.99	98	98	98
99.99	98	99.99	50
99.99	98	99.99	70
99.99	98	99.99	90
99.99	98	99.99	98
99.99	98	99.99	99.99
99.99	99.99	50	50
99.99	99.99	70	50
99.99	99.99	70	70
99.99	99.99	90	50
99.99	99.99	90	70
99.99	99.99	90	90
99.99	99.99	98	50
99.99	99.99	98	70
99.99	99.99	98	90
99.99	99.99	98	98
99.99	99.99	99.99	50
99.99	99.99	99.99	70
99.99	99.99	99.99	90
99.99	99.99	99.99	98
99.99	99.99	99.99	99.99

scale
-35
-33.6
-32.2
-30.8
-29.4
-28
-26.6
-25.2
-23.8
-22.4
-21
-19.6
-18.2
-16.8
-15.4
-14
-12.6
-11.2
-9.8
-8.4
-7
-5.6
-4.2
-2.8
-1.4
-0
1.4
2.8
4.2
5.6
7
8.4
9.8
11.2
12.6
14
15.4
16.8
18.2
19.6
21
22.4
23.8
25.2
26.6
28
29.4
30.8
32.2
33.6
35

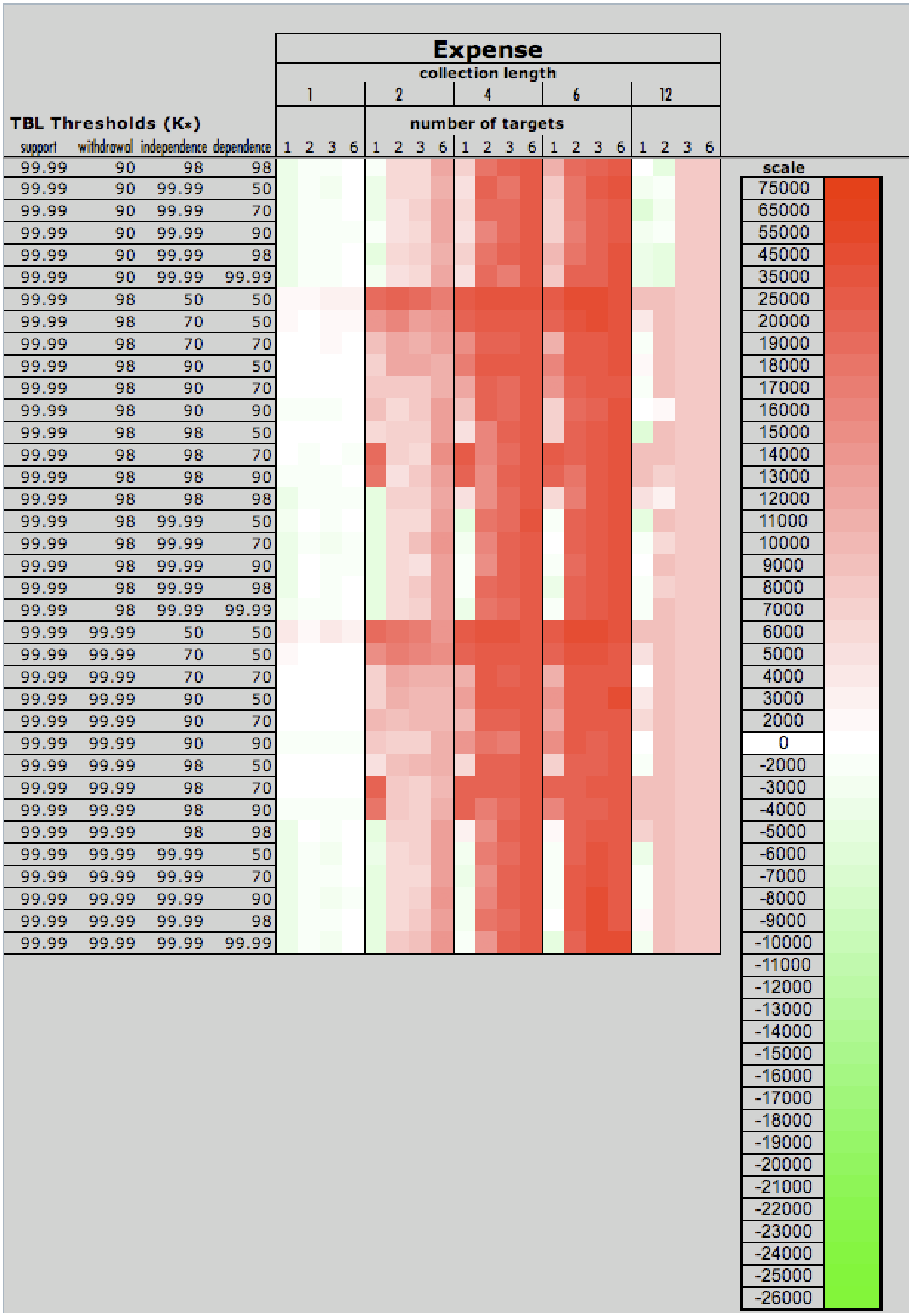
APPENDIX H: EXPERIMENT 5, EXPENSE RESULTS

This section presents the entire Expense footprint for Experiment 5, sorted by TBL thresholds.


Expense
collection length
1
2
4
6
12
number of targets
1 2 3 6 1 2 3 6 1 2 3 6 1 2 3 6 1 2 3 6
TBL Thresholds (K*)
support withdrawal independence dependence
50 50 50 50
70 50 50 50
70 50 70 50
70 50 70 70
70 70 50 50
70 70 70 50
70 70 70 70
90 50 50 50
90 50 70 50
90 50 70 70
90 50 90 50
90 50 90 70
90 50 90 90
90 70 50 50
90 70 70 50
90 70 70 70
90 70 90 50
90 70 90 70
90 70 90 90
90 90 50 50
90 90 70 50
90 90 70 70
90 90 90 50
90 90 90 70
90 90 90 90
98 50 50 50
98 50 70 50
98 50 70 70
98 50 90 50
98 50 90 70
98 50 90 90
98 50 98 50
98 50 98 70
98 50 98 90
98 50 98 98
98 70 50 50
98 70 70 50
98 70 70 70
98 70 90 50
98 70 90 70
98 70 90 90
98 70 98 50
98 70 98 70
98 70 98 90
98 70 98 98
98 90 50 50
98 90 70 50
98 90 70 70
98 90 90 50
98 90 90 70
98 90 90 90
98 90 98 50
scale
75000
65000
55000
45000
35000
25000
20000
19000
18000
17000
16000
15000
14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
0
-2000
-3000
-4000
-5000
-6000
-7000
-8000
-9000
-10000
-11000
-12000
-13000
-14000
-15000
-16000
-17000
-18000
-19000
-20000
-21000
-22000
-23000
-24000
-25000
-26000



TBL Thresholds (K*)			
support	withdrawal	independence	dependence
98	90	98	70
98	90	98	90
98	90	98	98
98	98	50	50
98	98	70	50
98	98	70	70
98	98	90	50
98	98	90	70
98	98	90	90
98	98	98	50
98	98	98	70
98	98	98	90
98	98	98	98
99.99	50	50	50
99.99	50	70	50
99.99	50	70	70
99.99	50	90	50
99.99	50	90	70
99.99	50	90	90
99.99	50	98	50
99.99	50	98	70
99.99	50	98	90
99.99	50	98	98
99.99	50	99.99	50
99.99	50	99.99	70
99.99	50	99.99	90
99.99	50	99.99	98
99.99	50	99.99	99.99
99.99	70	50	50
99.99	70	70	50
99.99	70	70	70
99.99	70	90	50
99.99	70	90	70
99.99	70	90	90
99.99	70	98	50
99.99	70	98	70
99.99	70	98	90
99.99	70	98	98
99.99	70	99.99	50
99.99	70	99.99	70
99.99	70	99.99	90
99.99	70	99.99	98
99.99	70	99.99	99.99
99.99	90	50	50
99.99	90	70	50
99.99	90	70	70
99.99	90	90	50
99.99	90	90	70
99.99	90	90	90
99.99	90	98	50
99.99	90	98	70
99.99	90	98	90



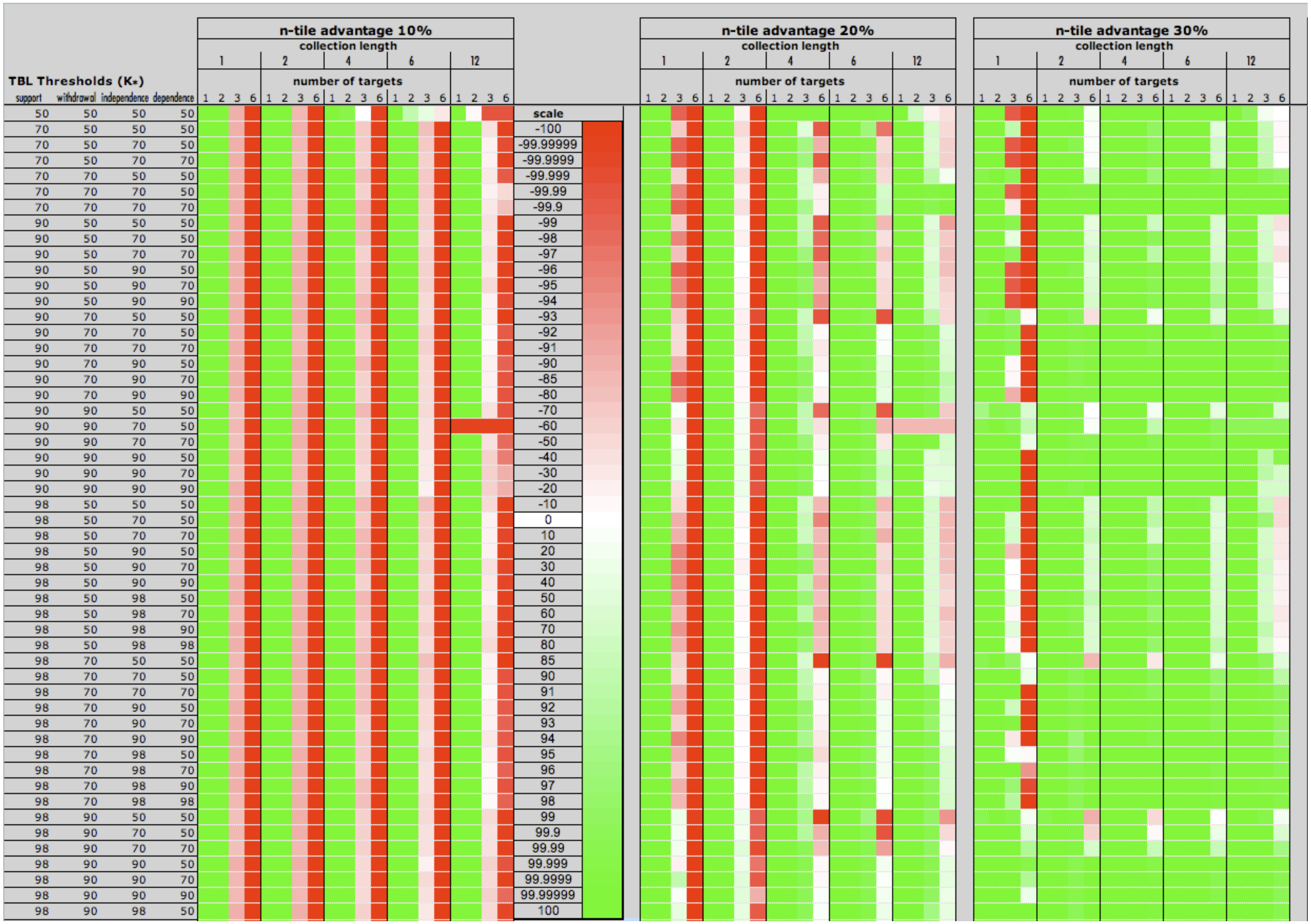
TBL Thresholds (K*)

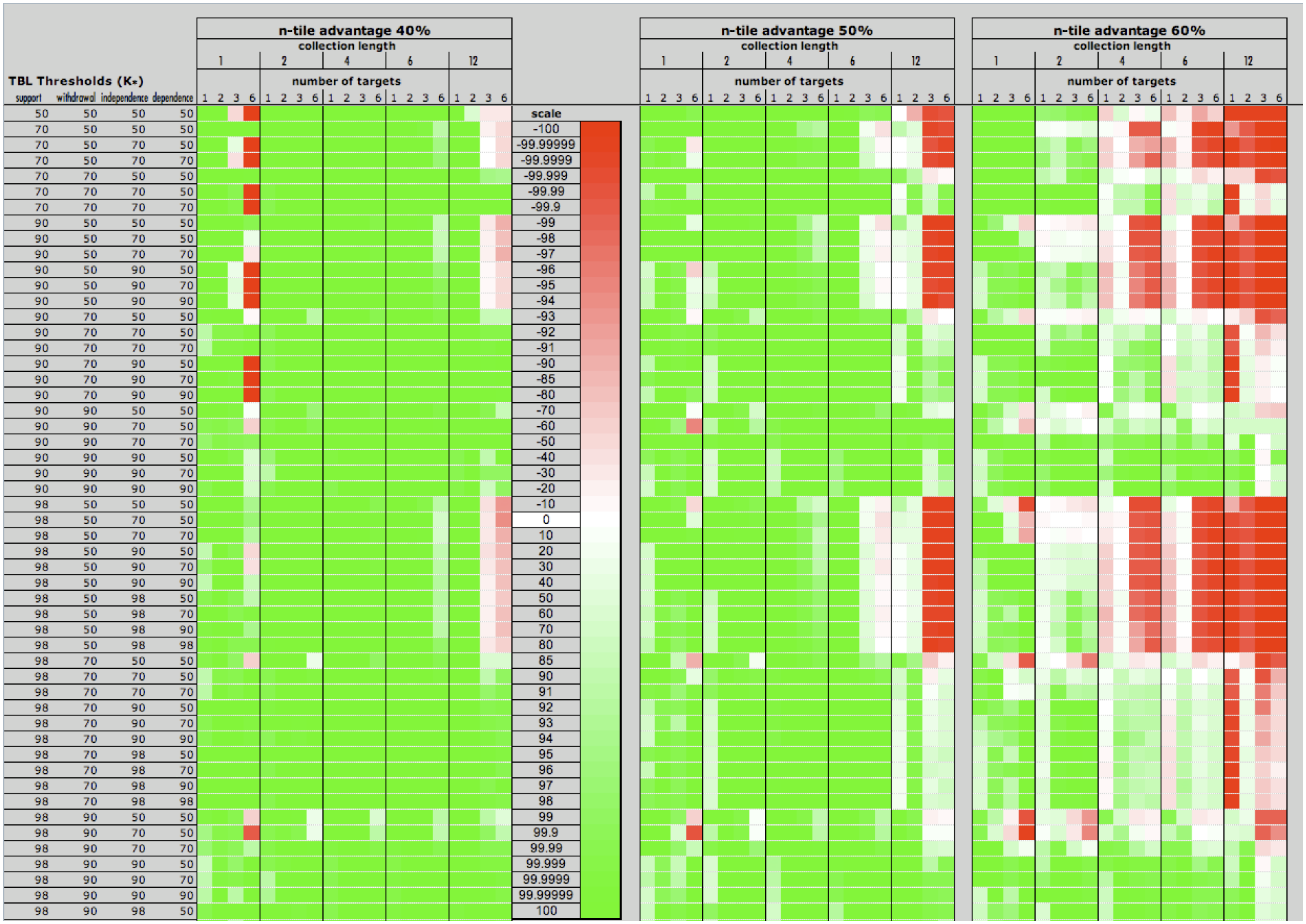
support	withdrawal	independence	dependence
99.99	90	98	98
99.99	90	99.99	50
99.99	90	99.99	70
99.99	90	99.99	90
99.99	90	99.99	98
99.99	90	99.99	99.99
99.99	98	50	50
99.99	98	70	50
99.99	98	70	70
99.99	98	90	50
99.99	98	90	70
99.99	98	90	90
99.99	98	98	50
99.99	98	98	70
99.99	98	98	90
99.99	98	98	98
99.99	98	99.99	50
99.99	98	99.99	70
99.99	98	99.99	90
99.99	98	99.99	98
99.99	98	99.99	99.99
99.99	99.99	50	50
99.99	99.99	70	50
99.99	99.99	70	70
99.99	99.99	90	50
99.99	99.99	90	70
99.99	99.99	90	90
99.99	99.99	98	50
99.99	99.99	98	70
99.99	99.99	98	90
99.99	99.99	98	98
99.99	99.99	99.99	50
99.99	99.99	99.99	70
99.99	99.99	99.99	90
99.99	99.99	99.99	98
99.99	99.99	99.99	99.99

scale
75000
65000
55000
45000
35000
25000
20000
19000
18000
17000
16000
15000
14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
0
-2000
-3000
-4000
-5000
-6000
-7000
-8000
-9000
-10000
-11000
-12000
-13000
-14000
-15000
-16000
-17000
-18000
-19000
-20000
-21000
-22000
-23000
-24000
-25000
-26000

APPENDIX I: EXPERIMENT 5 *N*-TILE RESULTS

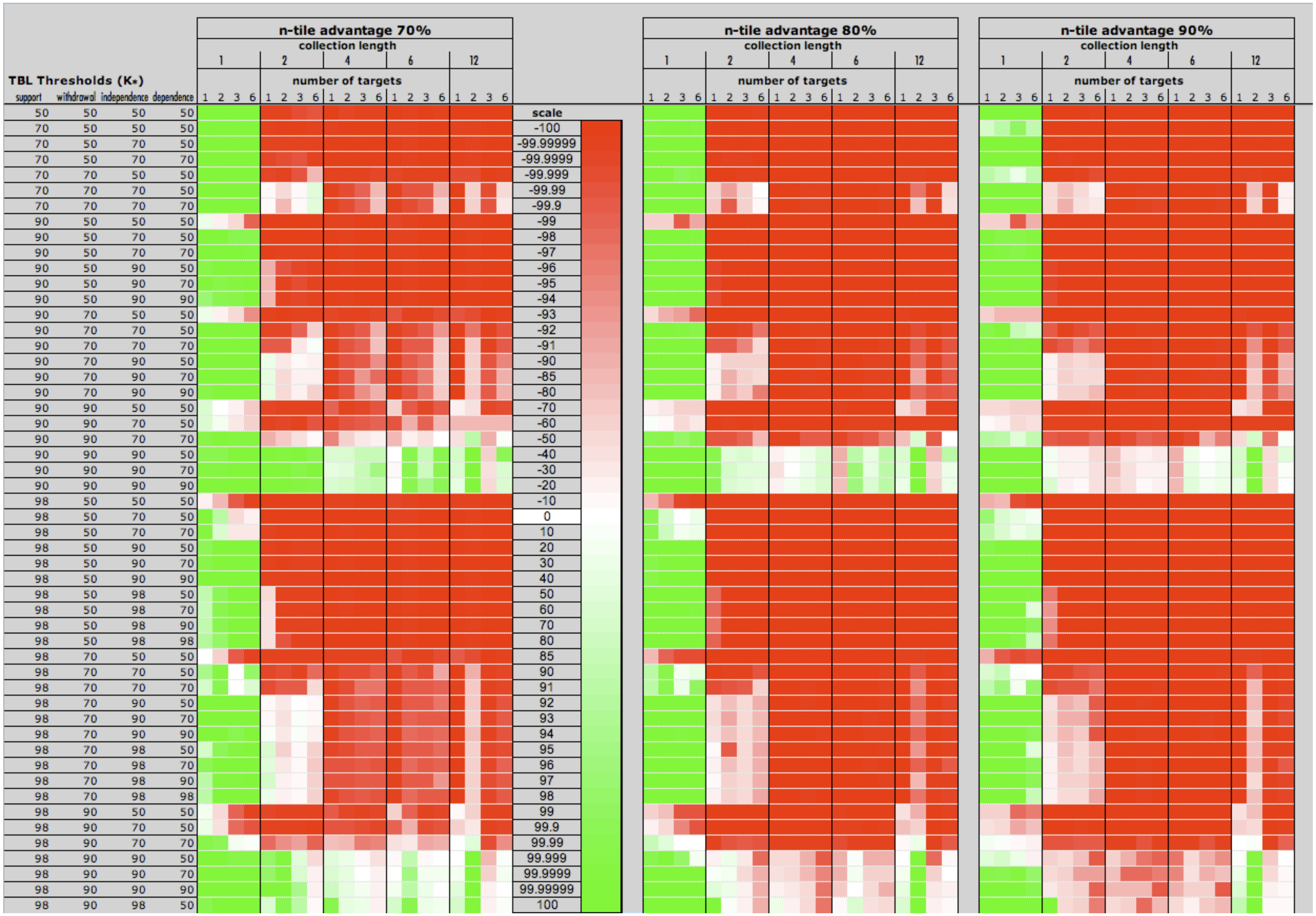
This section presents the full *n*-tile advantage results for Experiment 5. The results are sorted by TBL thresholds and are presented across the *n*-tiles and then down the TBL threshold factors.

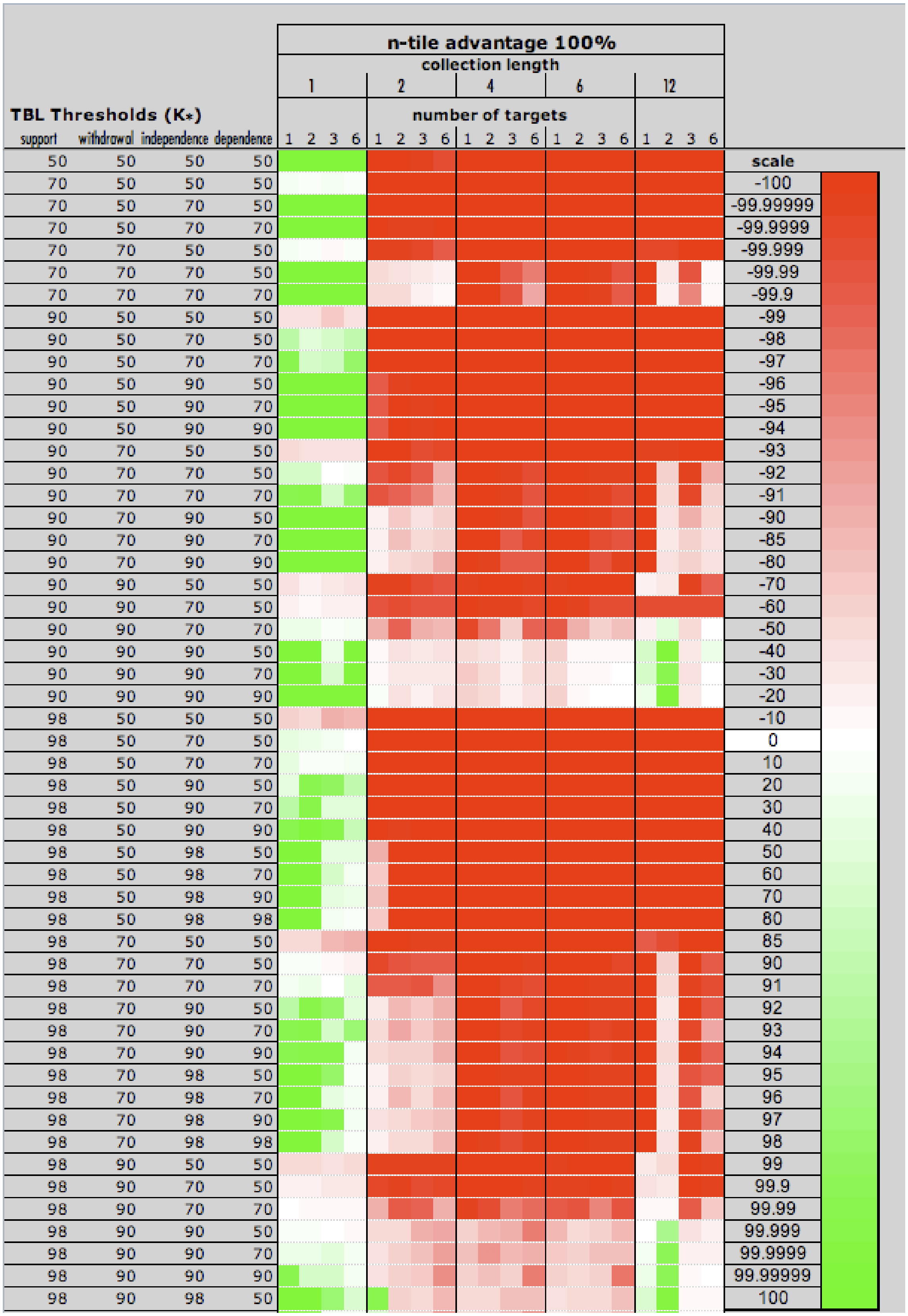

n-tile advantage 10%
n-tile advantage 20%
n-tile advantage 30%
collection length
1
2
4
6
12
number of targets
1 2 3 6
TBL Thresholds (K*)
support withdrawal independence dependence
50 50 50 50
70 50 50 50
70 50 70 50
70 50 70 70
70 70 50 50
70 70 70 50
70 70 70 70
90 50 50 50
90 50 70 50
90 50 70 70
90 50 90 50
90 50 90 70
90 50 90 90
90 70 50 50
90 70 70 50
90 70 70 70
90 70 90 50
90 70 90 70
90 70 90 90
90 90 50 50
90 90 70 50
90 90 70 70
90 90 90 50
90 90 90 70
90 90 90 90
98 50 50 50
98 50 70 50
98 50 70 70
98 50 90 50
98 50 90 70
98 50 90 90
98 50 98 50
98 50 98 70
98 50 98 90
98 50 98 98
98 70 50 50
98 70 70 50
98 70 70 70
98 70 90 50
98 70 90 70
98 70 90 90
98 70 98 50
98 70 98 70
98 70 98 90
98 70 98 98
98 90 50 50
98 90 70 50
98 90 70 70
98 90 90 50
98 90 90 70
98 90 90 90
98 90 98 50
scale
-100
-99.99999
-99.9999
-99.999
-99.99
-99.9
-99
-98
-97
-96
-95
-94
-93
-92
-91
-90
-85
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
85
90
91
92
93
94
95
96
97
98
99
99.9
99.99
99.999
99.9999
99.99999
100



TBL Thresholds (K*)

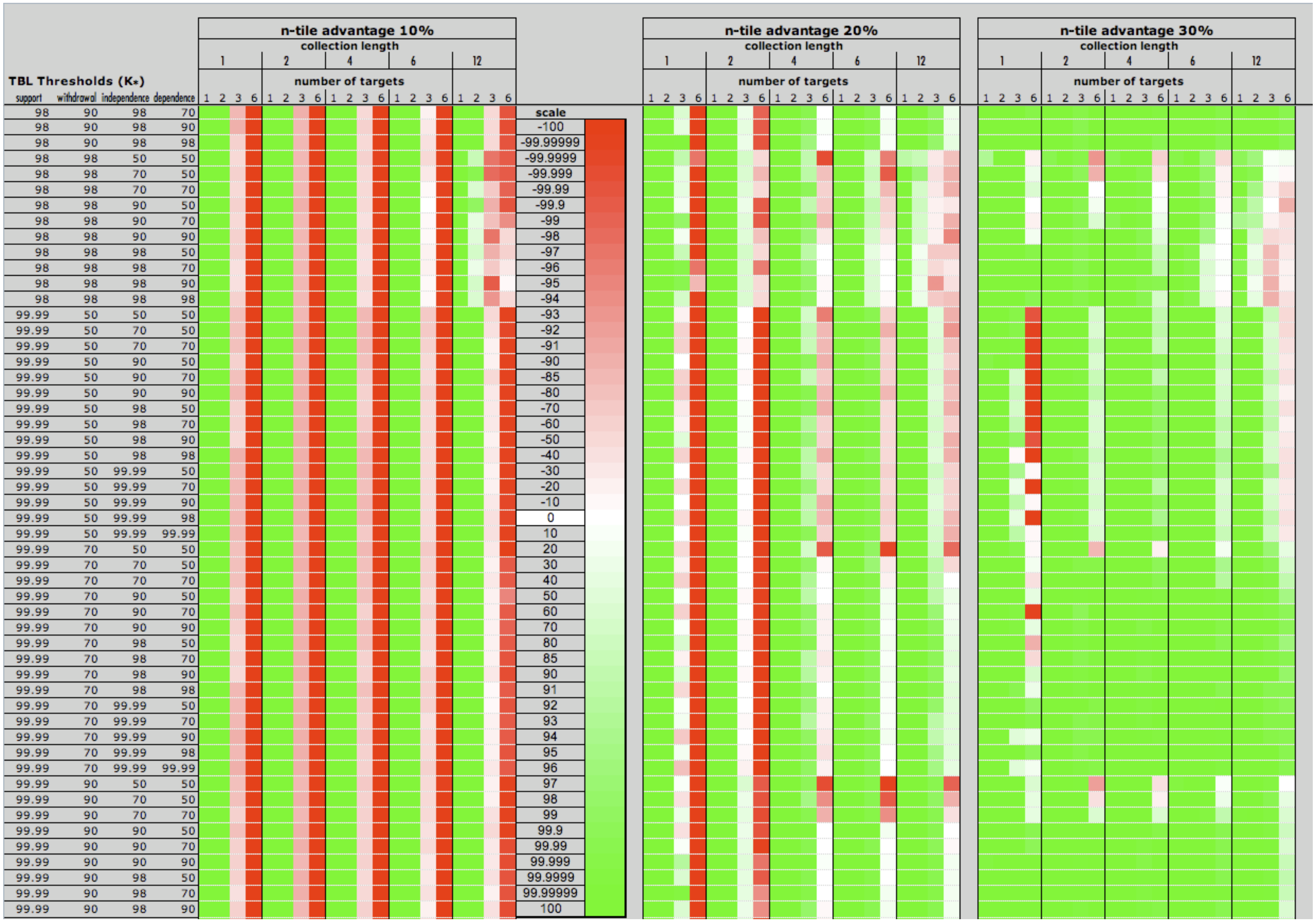
support	withdrawal	independence	dependence
50	50	50	50
70	50	50	50
70	50	70	50
70	50	70	70
70	70	50	50
70	70	70	50
70	70	70	70
90	50	50	50
90	50	70	50
90	50	70	70
90	50	90	50
90	50	90	70
90	50	90	90
90	70	50	50
90	70	70	50
90	70	70	70
90	70	90	50
90	70	90	70
90	70	90	90
90	90	50	50
90	90	70	50
90	90	70	70
90	90	90	50
90	90	90	70
90	90	90	90
98	50	50	50
98	50	70	50
98	50	70	70
98	50	90	50
98	50	90	70
98	50	90	90
98	50	98	50
98	50	98	70
98	50	98	90
98	50	98	98
98	70	50	50
98	70	70	50
98	70	70	70
98	70	90	50
98	70	90	70
98	70	90	90
98	70	98	50
98	70	98	70
98	70	98	90
98	70	98	98
98	90	50	50
98	90	70	50
98	90	70	70
98	90	90	50
98	90	90	70
98	90	90	90
98	90	98	50


TBL Thresholds (K*)
support withdrawal independence dependence
50 50 50 50
70 50 50 50
70 50 70 50
70 50 70 70
70 70 50 50
70 70 70 50
70 70 70 70
90 50 50 50
90 50 70 50
90 50 70 70
90 50 90 50
90 50 90 70
90 50 90 90
90 70 50 50
90 70 70 50
90 70 70 70
90 70 90 50
90 70 90 70
90 70 90 90
90 90 50 50
90 90 70 50
90 90 70 70
90 90 90 50
90 90 90 70
90 90 90 90
98 50 50 50
98 50 70 50
98 50 70 70
98 50 90 50
98 50 90 70
98 50 90 90
98 50 98 50
98 50 98 70
98 50 98 90
98 50 98 98
98 70 50 50
98 70 70 50
98 70 70 70
98 70 90 50
98 70 90 70
98 70 90 90
98 70 98 50
98 70 98 70
98 70 98 90
98 70 98 98
98 90 50 50
98 90 70 50
98 90 70 70
98 90 90 50
98 90 90 70
98 90 90 90
98 90 98 50
n-tile advantage 70%
collection length
1 2 4 6 12
number of targets
1 2 3 6
scale
-100
-99.99999
-99.9999
-99.999
-99.99
-99.9
-99
-98
-97
-96
-95
-94
-93
-92
-91
-90
-85
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
85
90
91
92
93
94
95
96
97
98
99
99.9
99.99
99.999
99.9999
99.99999
100
n-tile advantage 80%
collection length
1 2 4 6 12
number of targets
1 2 3 6
n-tile advantage 90%
collection length
1 2 4 6 12
number of targets
1 2 3 6



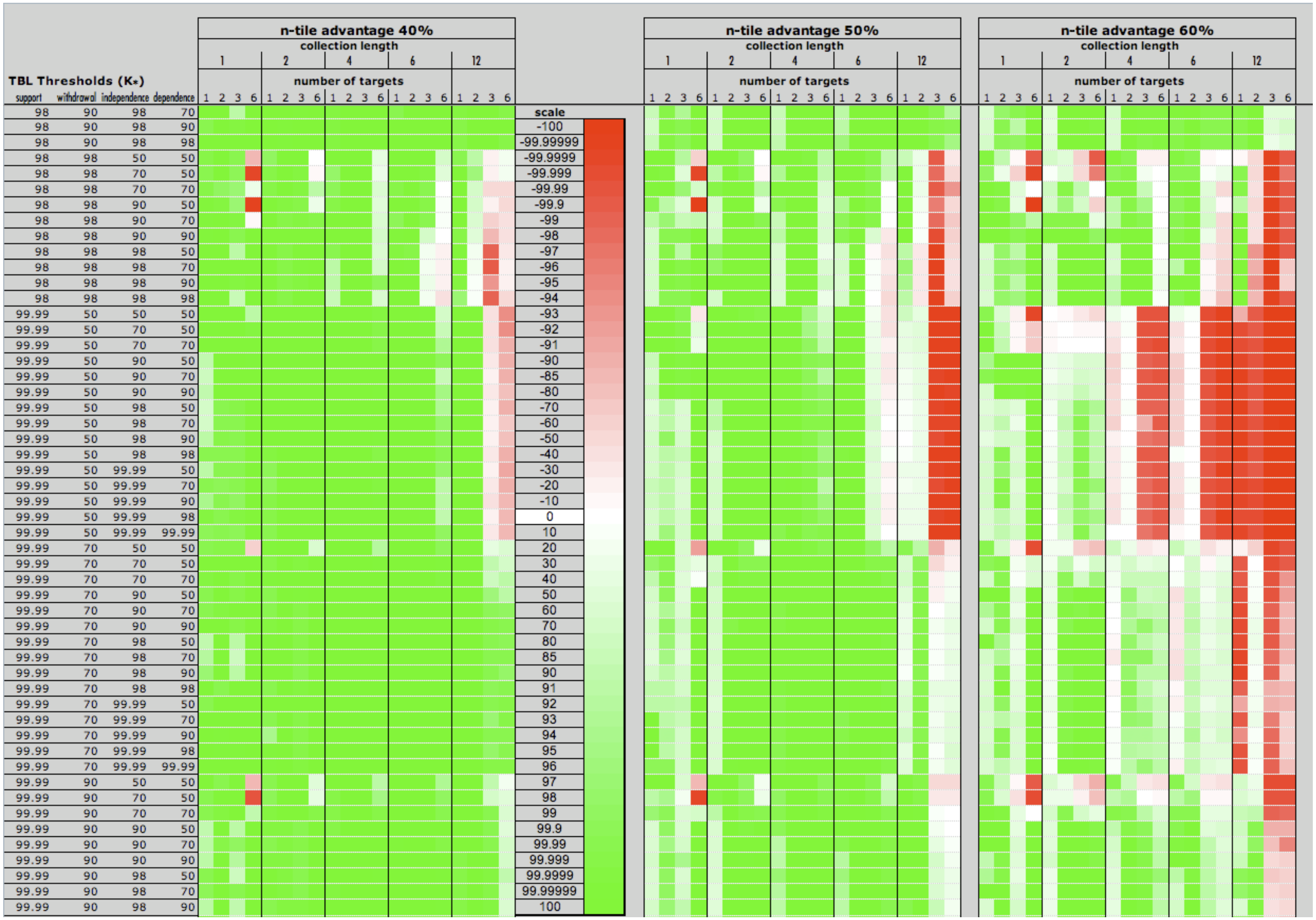
TBL Thresholds (K∗)

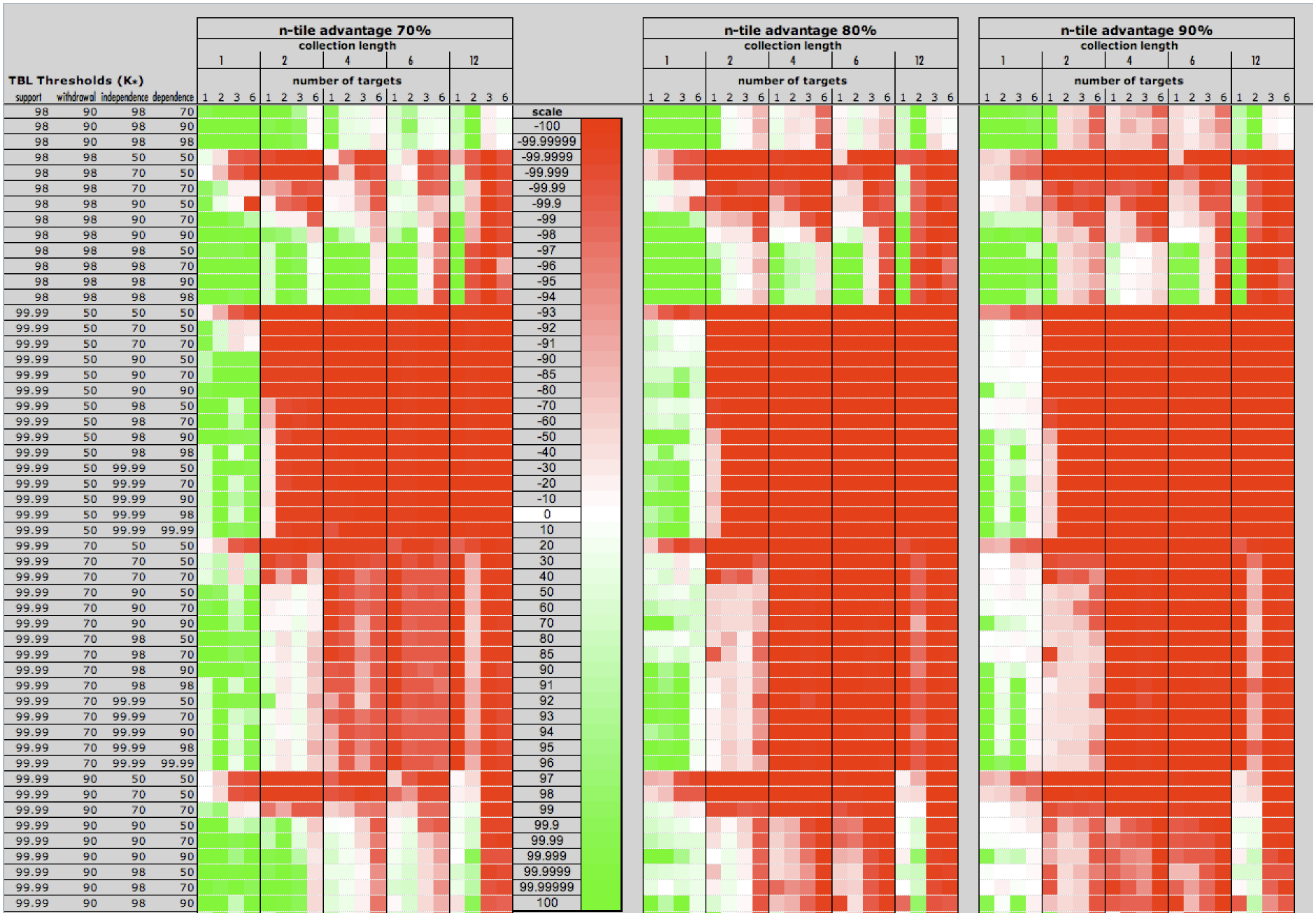
support	withdrawal	independence	dependence
50	50	50	50
70	50	50	50
70	50	70	50
70	50	70	70
70	70	50	50
70	70	70	50
70	70	70	70
90	50	50	50
90	50	70	50
90	50	70	70
90	50	90	50
90	50	90	70
90	50	90	90
90	70	50	50
90	70	70	50
90	70	70	70
90	70	90	50
90	70	90	70
90	70	90	90
90	90	50	50
90	90	70	50
90	90	70	70
90	90	90	50
90	90	90	70
90	90	90	90
98	50	50	50
98	50	70	50
98	50	70	70
98	50	90	50
98	50	90	70
98	50	90	90
98	50	98	50
98	50	98	70
98	50	98	90
98	50	98	98
98	70	50	50
98	70	70	50
98	70	70	70
98	70	90	50
98	70	90	70
98	70	90	90
98	70	98	50
98	70	98	70
98	70	98	90
98	70	98	98
98	90	50	50
98	90	70	50
98	90	70	70
98	90	90	50
98	90	90	70
98	90	90	90
98	90	98	50

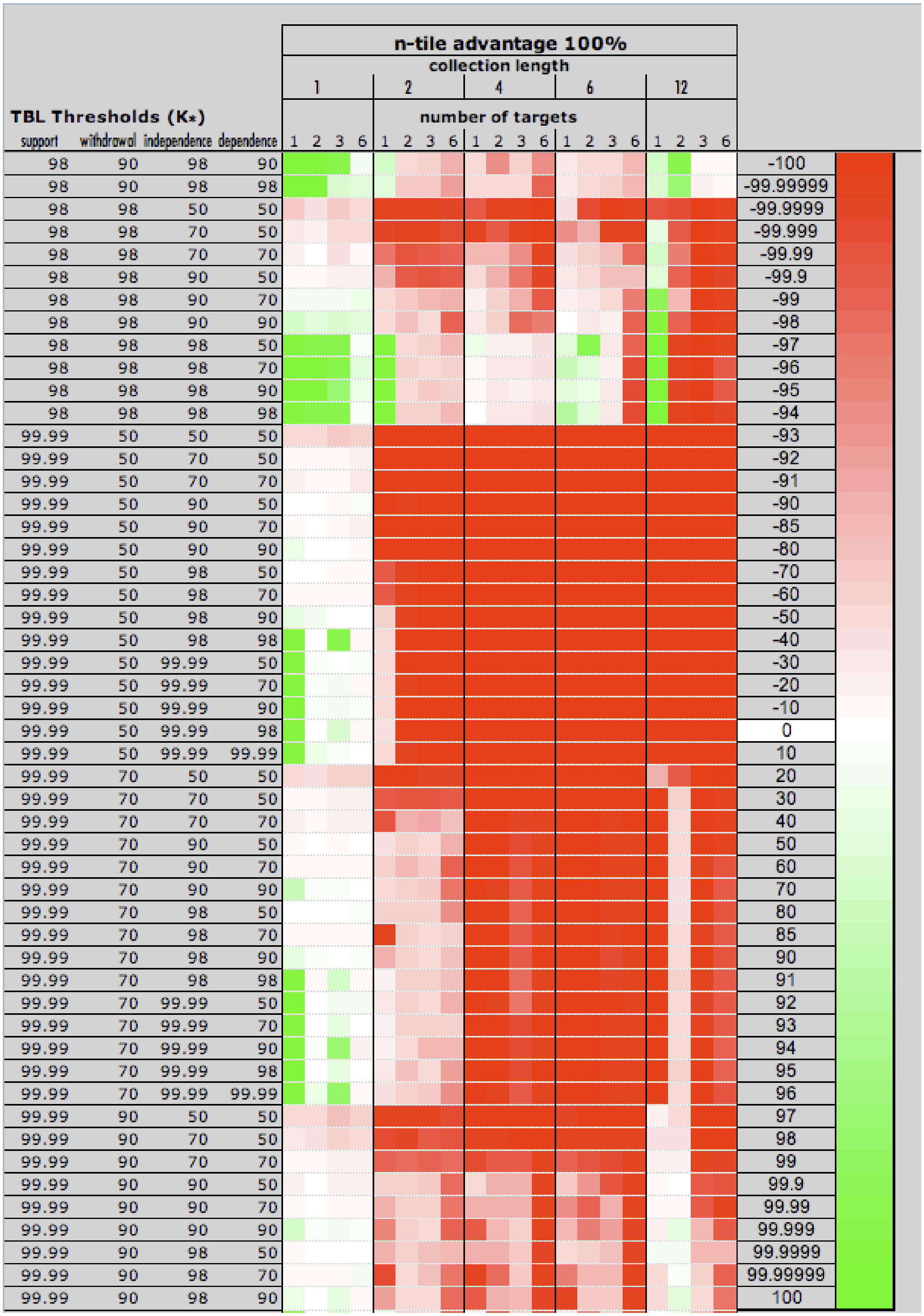


TBL Thresholds (K*)

support	withdrawal	independence	dependence
98	90	98	70
98	90	98	90
98	90	98	98
98	98	50	50
98	98	70	50
98	98	70	70
98	98	90	50
98	98	90	70
98	98	90	90
98	98	98	50
98	98	98	70
98	98	98	90
98	98	98	98
99.99	50	50	50
99.99	50	70	50
99.99	50	70	70
99.99	50	90	50
99.99	50	90	70
99.99	50	90	90
99.99	50	98	50
99.99	50	98	70
99.99	50	98	90
99.99	50	98	98
99.99	50	99.99	50
99.99	50	99.99	70
99.99	50	99.99	90
99.99	50	99.99	98
99.99	50	99.99	99.99
99.99	70	50	50
99.99	70	70	50
99.99	70	70	70
99.99	70	90	50
99.99	70	90	70
99.99	70	90	90
99.99	70	98	50
99.99	70	98	70
99.99	70	98	90
99.99	70	98	98
99.99	70	99.99	50
99.99	70	99.99	70
99.99	70	99.99	90
99.99	70	99.99	98
99.99	70	99.99	99.99
99.99	90	50	50
99.99	90	70	50
99.99	90	70	70
99.99	90	90	50
99.99	90	90	70
99.99	90	90	90
99.99	90	98	50
99.99	90	98	70
99.99	90	98	90

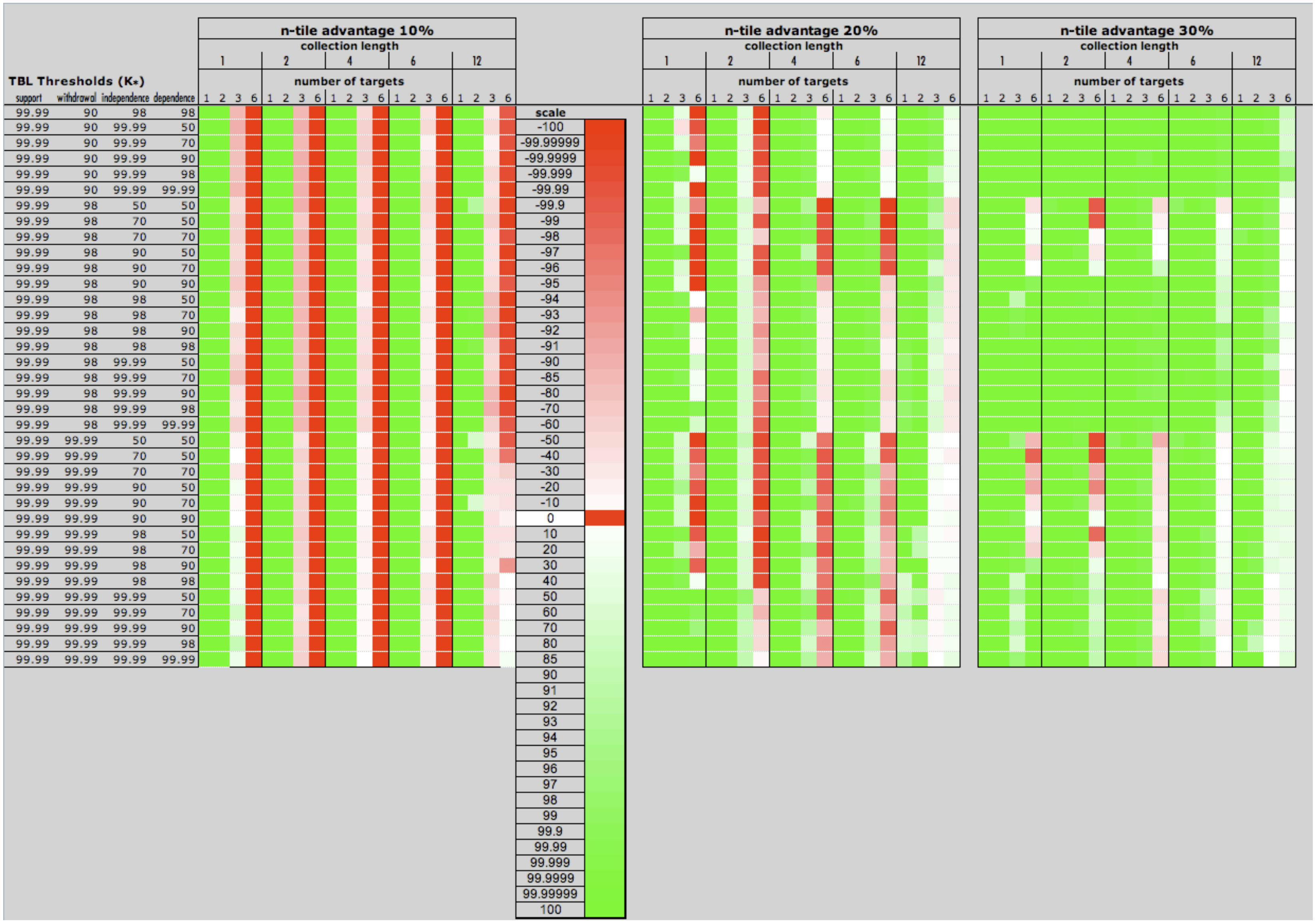

TBL Thresholds (K*)
support withdrawal independence dependence
98 90 98 70
98 90 98 90
98 90 98 98
98 98 50 50
98 98 70 50
98 98 70 70
98 98 90 50
98 98 90 70
98 98 90 90
98 98 98 50
98 98 98 70
98 98 98 90
98 98 98 98
99.99 50 50 50
99.99 50 70 50
99.99 50 70 70
99.99 50 90 50
99.99 50 90 70
99.99 50 90 90
99.99 50 98 50
99.99 50 98 70
99.99 50 98 90
99.99 50 98 98
99.99 50 99.99 50
99.99 50 99.99 70
99.99 50 99.99 90
99.99 50 99.99 98
99.99 50 99.99 99.99
99.99 70 50 50
99.99 70 70 50
99.99 70 70 70
99.99 70 90 50
99.99 70 90 70
99.99 70 90 90
99.99 70 98 50
99.99 70 98 70
99.99 70 98 90
99.99 70 98 98
99.99 70 99.99 50
99.99 70 99.99 70
99.99 70 99.99 90
99.99 70 99.99 98
99.99 70 99.99 99.99
99.99 90 50 50
99.99 90 70 50
99.99 90 70 70
99.99 90 90 50
99.99 90 90 70
99.99 90 90 90
99.99 90 98 50
99.99 90 98 70
99.99 90 98 90
n-tile advantage 40%
n-tile advantage 50%
n-tile advantage 60%
collection length
1 2 4 6 12
number of targets
1 2 3 6
scale
-100
-99.99999
-99.9999
-99.999
-99.99
-99.9
-99
-98
-97
-96
-95
-94
-93
-92
-91
-90
-85
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
85
90
91
92
93
94
95
96
97
98
99
99.9
99.99
99.999
99.9999
99.99999
100


TBL Thresholds (K*)
support withdrawal independence dependence
98 90 98 70
98 90 98 90
98 90 98 98
98 98 50 50
98 98 70 50
98 98 70 70
98 98 90 50
98 98 90 70
98 98 90 90
98 98 98 50
98 98 98 70
98 98 98 90
98 98 98 98
99.99 50 50 50
99.99 50 70 50
99.99 50 70 70
99.99 50 90 50
99.99 50 90 70
99.99 50 90 90
99.99 50 98 50
99.99 50 98 70
99.99 50 98 90
99.99 50 98 98
99.99 50 99.99 50
99.99 50 99.99 70
99.99 50 99.99 90
99.99 50 99.99 98
99.99 50 99.99 99.99
99.99 70 50 50
99.99 70 70 50
99.99 70 70 70
99.99 70 90 50
99.99 70 90 70
99.99 70 90 90
99.99 70 98 50
99.99 70 98 70
99.99 70 98 90
99.99 70 98 98
99.99 70 99.99 50
99.99 70 99.99 70
99.99 70 99.99 90
99.99 70 99.99 98
99.99 70 99.99 99.99
99.99 90 50 50
99.99 90 70 50
99.99 90 70 70
99.99 90 90 50
99.99 90 90 70
99.99 90 90 90
99.99 90 98 50
99.99 90 98 70
99.99 90 98 90
n-tile advantage 70%
collection length
1 2 4 6 12
number of targets
1 2 3 6 1 2 3 6 1 2 3 6 1 2 3 6 1 2 3 6
scale
-100
-99.99999
-99.9999
-99.999
-99.99
-99.9
-99
-98
-97
-96
-95
-94
-93
-92
-91
-90
-85
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
85
90
91
92
93
94
95
96
97
98
99
99.9
99.99
99.999
99.9999
99.99999
100
n-tile advantage 80%
collection length
1 2 4 6 12
number of targets
1 2 3 6 1 2 3 6 1 2 3 6 1 2 3 6 1 2 3 6
n-tile advantage 90%
collection length
1 2 4 6 12
number of targets
1 2 3 6 1 2 3 6 1 2 3 6 1 2 3 6 1 2 3 6



TBL Thresholds (K*)

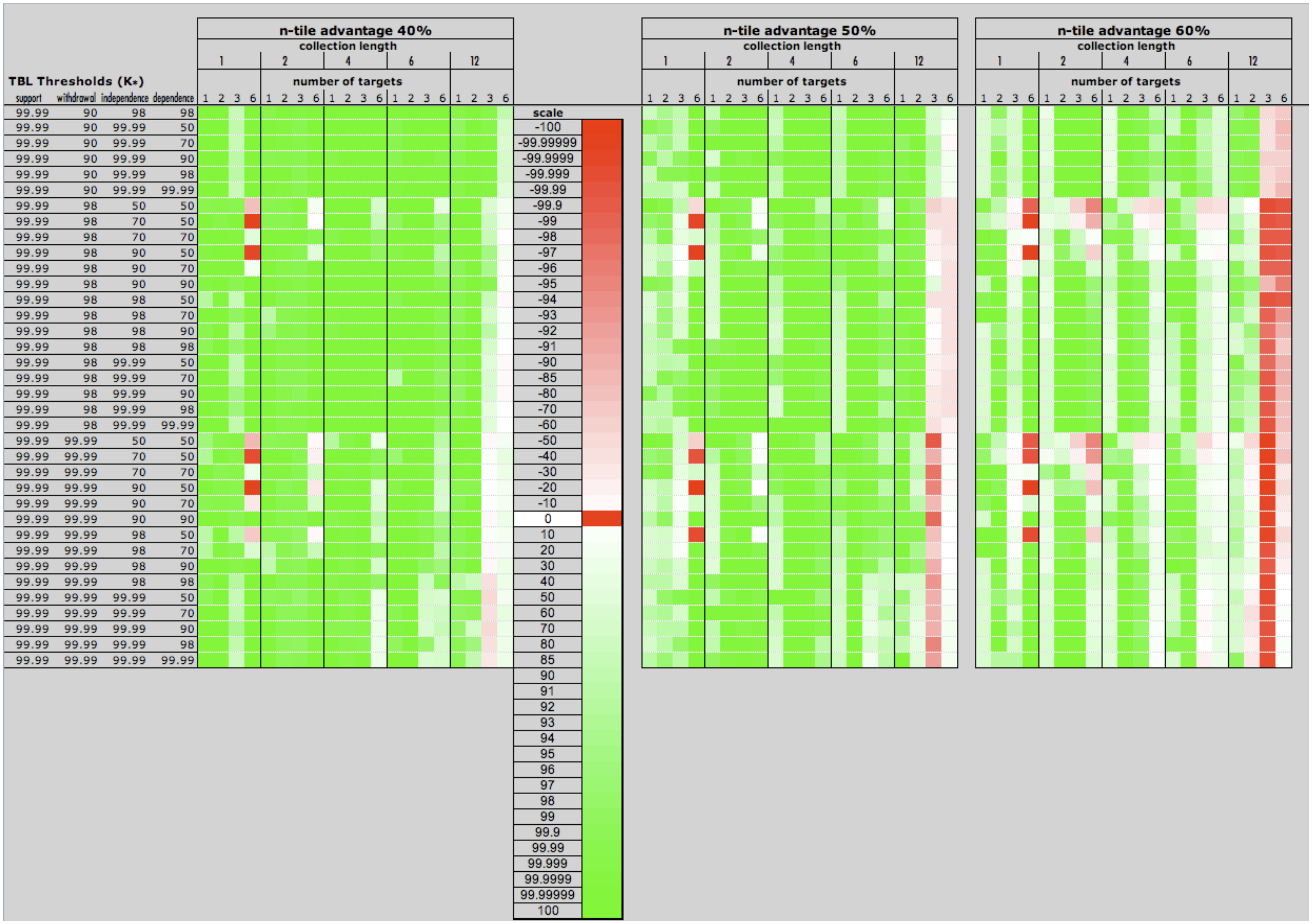
support	withdrawal	independence	dependence
98	90	98	90
98	90	98	98
98	98	50	50
98	98	70	50
98	98	70	70
98	98	90	50
98	98	90	70
98	98	90	90
98	98	98	50
98	98	98	70
98	98	98	90
98	98	98	98
99.99	50	50	50
99.99	50	70	50
99.99	50	70	70
99.99	50	90	50
99.99	50	90	70
99.99	50	90	90
99.99	50	98	50
99.99	50	98	70
99.99	50	98	90
99.99	50	98	98
99.99	50	99.99	50
99.99	50	99.99	70
99.99	50	99.99	90
99.99	50	99.99	98
99.99	50	99.99	99.99
99.99	70	50	50
99.99	70	70	50
99.99	70	70	70
99.99	70	90	50
99.99	70	90	70
99.99	70	90	90
99.99	70	98	50
99.99	70	98	70
99.99	70	98	90
99.99	70	98	98
99.99	70	99.99	50
99.99	70	99.99	70
99.99	70	99.99	90
99.99	70	99.99	98
99.99	70	99.99	99.99
99.99	90	50	50
99.99	90	70	50
99.99	90	70	70
99.99	90	90	50
99.99	90	90	70
99.99	90	90	90
99.99	90	98	50
99.99	90	98	70
99.99	90	98	90



TBL Thresholds (K*)

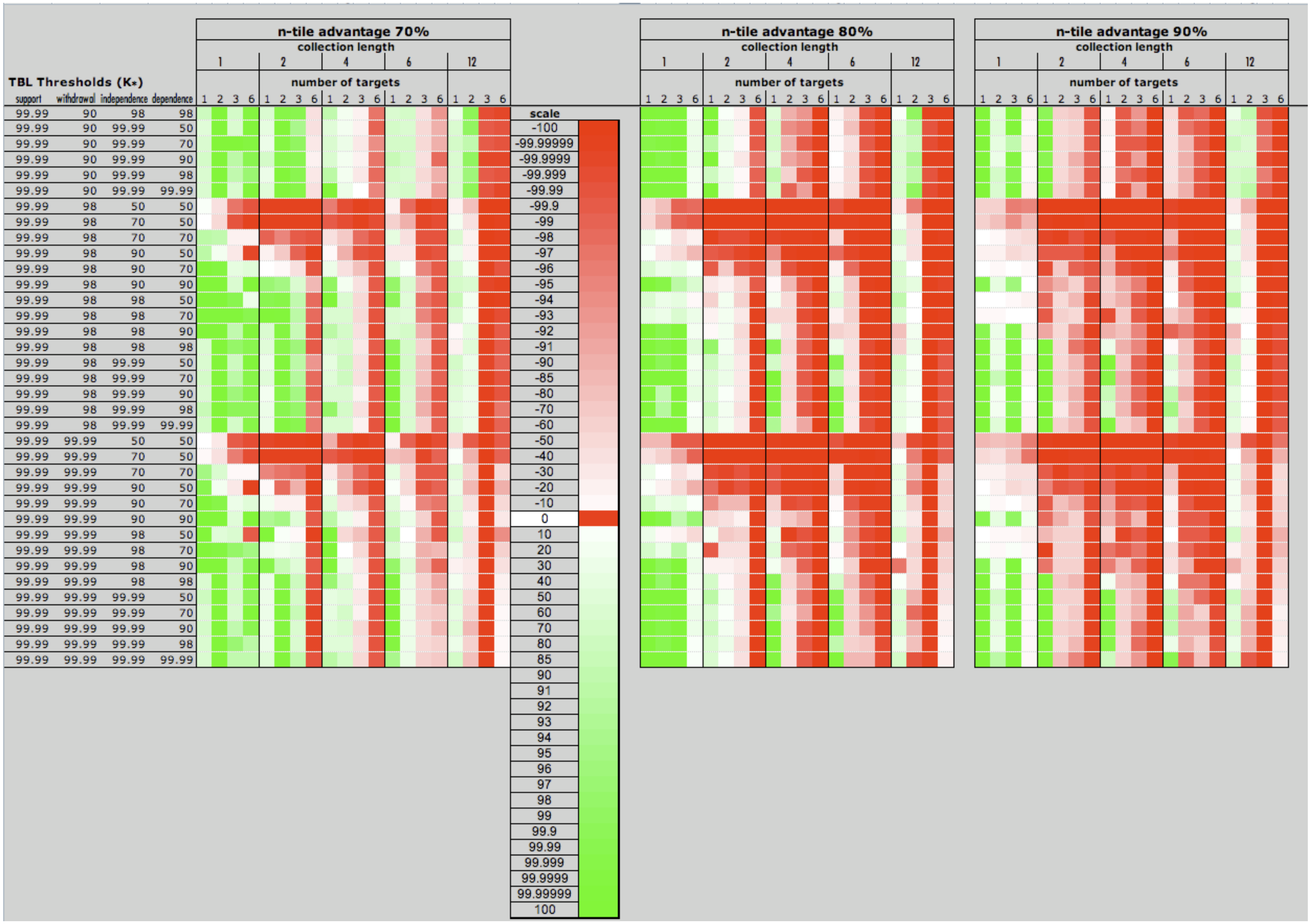
support	withdrawal	independence	dependence
99.99	90	98	98
99.99	90	99.99	50
99.99	90	99.99	70
99.99	90	99.99	90
99.99	90	99.99	98
99.99	90	99.99	99.99
99.99	98	50	50
99.99	98	70	50
99.99	98	70	70
99.99	98	90	50
99.99	98	90	70
99.99	98	90	90
99.99	98	98	50
99.99	98	98	70
99.99	98	98	90
99.99	98	98	98
99.99	98	99.99	50
99.99	98	99.99	70
99.99	98	99.99	90
99.99	98	99.99	98
99.99	98	99.99	99.99
99.99	99.99	50	50
99.99	99.99	70	50
99.99	99.99	70	70
99.99	99.99	90	50
99.99	99.99	90	70
99.99	99.99	90	90
99.99	99.99	98	50
99.99	99.99	98	70
99.99	99.99	98	90
99.99	99.99	98	98
99.99	99.99	99.99	50
99.99	99.99	99.99	70
99.99	99.99	99.99	90
99.99	99.99	99.99	98
99.99	99.99	99.99	99.99

scale
-100
-99.99999
-99.9999
-99.999
-99.99
-99.9
-99
-98
-97
-96
-95
-94
-93
-92
-91
-90
-85
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
85
90
91
92
93
94
95
96
97
98
99
99.9
99.99
99.999
99.9999
99.99999
100

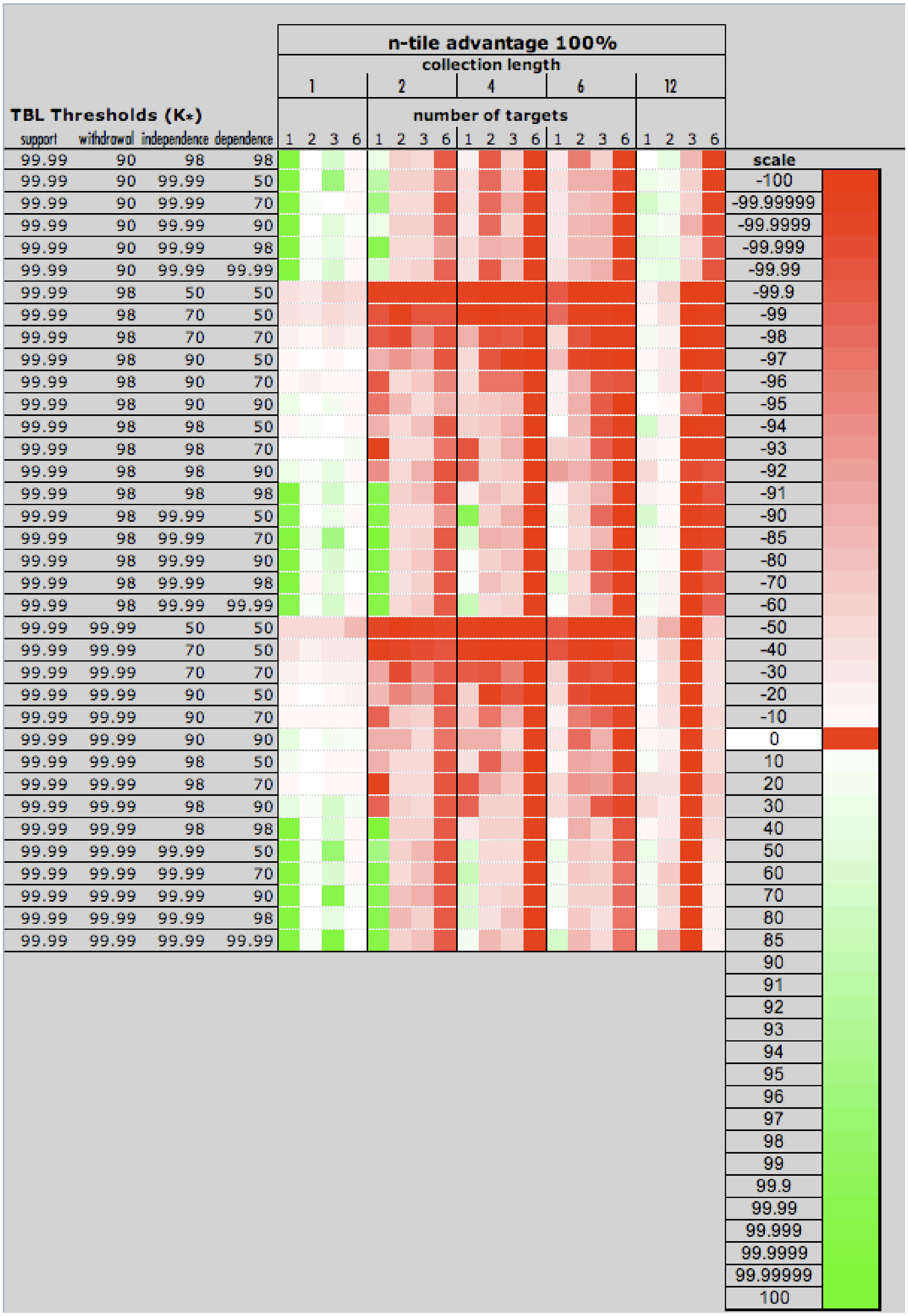


TBL Thresholds (K*)

support	withdrawal	independence	dependence
99.99	90	98	98
99.99	90	99.99	50
99.99	90	99.99	70
99.99	90	99.99	90
99.99	90	99.99	98
99.99	90	99.99	99.99
99.99	98	50	50
99.99	98	70	50
99.99	98	70	70
99.99	98	90	50
99.99	98	90	70
99.99	98	90	90
99.99	98	98	50
99.99	98	98	70
99.99	98	98	90
99.99	98	98	98
99.99	98	99.99	50
99.99	98	99.99	70
99.99	98	99.99	90
99.99	98	99.99	98
99.99	98	99.99	99.99
99.99	99.99	50	50
99.99	99.99	70	50
99.99	99.99	70	70
99.99	99.99	90	50
99.99	99.99	90	70
99.99	99.99	90	90
99.99	99.99	98	50
99.99	99.99	98	70
99.99	99.99	98	90
99.99	99.99	98	98
99.99	99.99	99.99	50
99.99	99.99	99.99	70
99.99	99.99	99.99	90
99.99	99.99	99.99	98
99.99	99.99	99.99	99.99



TBL Thresholds (K*)			
support	withdrawal	independence	dependence
99.99	90	98	98
99.99	90	99.99	50
99.99	90	99.99	70
99.99	90	99.99	90
99.99	90	99.99	98
99.99	90	99.99	99.99
99.99	98	50	50
99.99	98	70	50
99.99	98	70	70
99.99	98	90	50
99.99	98	90	70
99.99	98	90	90
99.99	98	98	50
99.99	98	98	70
99.99	98	98	90
99.99	98	98	98
99.99	98	99.99	50
99.99	98	99.99	70
99.99	98	99.99	90
99.99	98	99.99	98
99.99	98	99.99	99.99
99.99	99.99	50	50
99.99	99.99	70	50
99.99	99.99	70	70
99.99	99.99	90	50
99.99	99.99	90	70
99.99	99.99	90	90
99.99	99.99	98	50
99.99	99.99	98	70
99.99	99.99	98	90
99.99	99.99	98	98
99.99	99.99	99.99	50
99.99	99.99	99.99	70
99.99	99.99	99.99	90
99.99	99.99	99.99	98
99.99	99.99	99.99	99.99



n-tile advantage 100%

collection length: 1, 2, 4, 6, 12

number of targets: 1 2 3 6

TBL Thresholds (K*)

support	withdrawal	independence	dependence
99.99	90	98	98
99.99	90	99.99	50
99.99	90	99.99	70
99.99	90	99.99	90
99.99	90	99.99	98
99.99	90	99.99	99.99
99.99	98	50	50
99.99	98	70	50
99.99	98	70	70
99.99	98	90	50
99.99	98	90	70
99.99	98	90	90
99.99	98	98	50
99.99	98	98	70
99.99	98	98	90
99.99	98	98	98
99.99	98	99.99	50
99.99	98	99.99	70
99.99	98	99.99	90
99.99	98	99.99	98
99.99	98	99.99	99.99
99.99	99.99	50	50
99.99	99.99	70	50
99.99	99.99	70	70
99.99	99.99	90	50
99.99	99.99	90	70
99.99	99.99	90	90
99.99	99.99	98	50
99.99	99.99	98	70
99.99	99.99	98	90
99.99	99.99	98	98
99.99	99.99	99.99	50
99.99	99.99	99.99	70
99.99	99.99	99.99	90
99.99	99.99	99.99	98
99.99	99.99	99.99	99.99

scale

-100
-99.99999
-99.9999
-99.999
-99.99
-99.9
-99
-98
-97
-96
-95
-94
-93
-92
-91
-90
-85
-80
-70
-60
-50
-40
-30
-20
-10
0
10
20
30
40
50
60
70
80
85
90
91
92
93
94
95
96
97
98
99
99.9
99.99
99.999
99.9999
99.99999
100

DIGITAL APPENDICES

This section provides the list of the digital appendices. All Reduced Results are in Microsoft Excel spreadsheets which contain VBA macros.

d.A. Java Source Code (.java)

d.B. Experiment 1 Reduced Results

d.C. Experiment 2 Reduced Results for case (70, 70, 70, 70)

d.D. Experiment 2 Reduced Results for case (99.99, 95, 50, 50)

d.E. Experiment 2 Reduced Results for case (95, 95, 95, 50)

d.F. Experiment 3 Reduced Results

d.G. Experiment 4 Reduced Results for case (70, 70, 70, 50)

d.H. Experiment 4 Reduced Results for case (99.99, 99.99, 99.99, 99.99)

d.I. Experiment 4 Reduced Results for case (99.99, 50, 50, 50)

d.J. Experiment 5 Reduced Results

d.K. Experiment 6 Reduced Results for (90, 90, 90, 70)

d.L. Experiment 6 Reduced Results for (99.99, 99.99, 70, 70)

d.M. Experiment 6 Reduced Results for (99.99, 50, 50, 50)