Applying Multi-Criteria Optimisation to Develop Cognitive Models

Peter C. R. Lane
School of Computer Science,
University of Hertfordshire,
College Lane, HATFIELD AL10 9AB,
Hertfordshire, England
peter.lane@bcs.org.uk

Fernand Gobet
School of Social Science and Law,
Brunel University,
UXBRIDGE UB8 3PH,
Middlesex, England
fernand.gobet@brunel.ac.uk

Abstract

A scientific theory is developed by modelling empirical data in a range of domains. The goal of developing a theory is to optimise the fit of the theory to as many experimental settings as possible, whilst retaining some qualitative properties such as 'parsimony' or 'comprehensibility'. We formalise the task of developing theories of human cognition as a problem in multi-criteria optimisation. There are many challenges in this task, including the representation of competing theories, coordinating the fit with multiple experiments, and bringing together competing results to provide suitable theories. Experiments demonstrate the development of a theory of categorisation, using multiple optimisation criteria in genetic algorithms to locate pareto-optimal sets.

1 Introduction

Cognitive science studies the behaviour of human (or animal) participants in different experimental settings, with the aim of devising explanations for the observed behaviour. An important aspect of this study is the development of computational models which can simulate the observed behaviour. Classes of models are collected together into *theories*, where a set of constraints or typical representations are used to define a collection of similar models. Empirical work collects specific items of observed behaviour in well-defined experimental settings. One aim of cognitive science is to locate models which do well in as many of the experimental settings as possible.

The problem confronting cognitive scientists may be viewed as a classic multi-criteria optimisation problem. There is a collection of models (the *variables*), perhaps drawn from multiple theories, and there is a collection of experimental evidence (the *constraints*). The task of the modeller is to select from the set of models those which best fit the multiple experiments. In particular, selecting a set of models which perform well on one experiment will affect the available set of models when considering a second experiment. The aim of this paper is to formalise the model-selection problem so that optimisation techniques, specifically genetic algorithms, may be effectively applied. We study a single class of experiments, from categorisation, and three different classes of models.

In brief, the structure of our approach is as follows. First, the target behaviour in each of the experimental settings is converted into a single quantitative measure: each measure is known as a *constraint*. Second, the classes of models which we are interested in are defined and *parameterised*. The space of models is defined by the possible types of models and also the range of parameter settings for each model. Third, we use genetic algorithms [3] to locate optimal sets of models. Finally, consideration of the models found by our technique enables us to explore the range of viable and robust cognitive theories given the experimental constraints.

We continue this paper in Section 2 by explaining what a cognitive theory is, and how cognitive models can be specified in terms of their experimental behaviour. Section 3 defines the process of finding a cognitive model as a multi-criteria optimisation problem, and explains how genetic algorithms can be applied. Section 4 describes in detail the constraints and variables (models) over which we wish to optimise. Section 5 gives details of the problem representation, and Section 6 describes our initial experimental results. Section 7 discusses some related work and avenues for further exploration.

Example	Attribute (A)			
	A0	A1	A2	A3
A examples				
E1	1	1	1	0
E2	1	0	1	0
E3	1	0	1	1
E4	1	1	0	1
E5	0	1	1	1
B examples				
E6	1	1	0	0
E7	0	1	1	0
E8	0	0	0	1
E9	0	0	0	0
Transfer items				
E10	1	0	0	1
E11	1	0	0	0
E12	1	1	1	1
E13	0	0	1	0
E14	0	1	0	1
E15	0	0	1	1
E16	0	1	0	0

Table 1: The 5-4 structure used in categorisation experiments (after [9, 12])

Example	P($R_A\|E_i$)		Time (s)
	1st	Avg	
E1	0.97	0.83	1.11
E2	0.97	0.82	1.34
E3	0.92	0.89	1.08
E4	0.81	0.79	1.27
E5	0.72	0.74	1.07
E6	0.33	0.30	1.30
E7	0.28	0.28	1.08
E8	0.03	0.15	1.13
E9	0.05	0.11	1.19
E10	0.72	0.62	
E11	0.56	0.40	
E12	0.98	0.88	
E13	0.34	0.34	
E14	0.27	0.40	
E15	0.39	0.55	
E16	0.09	0.17	

Table 2: Target behaviours in 5-4 structure: Probability of responding A and time to make a classification. (Classification times were not collected for the transfer items.)

2 Developing Cognitive Theories

2.1 Psychological Experiments

Experimental data form the scaffolding upon which a cognitive theory is developed. Each experimental result forms a *constraint* on our understanding of the phenomenon. As an example, we consider the task of *categorisation*, which has been studied in intense detail by psychologists and modellers. Initial experiments by Medin and Smith [9] in categorisation spawned a large number of follow-up studies. Smith and Minda [12] describe a collection of thirty previous experimental results, and analyse a number of mathematical models of behaviour.

Table 1 illustrates the basic concepts used in these experiments, known as the 5-4 structure. There are four binary attributes. Examples of category A are typically those closer to having all four attribute values set, whereas examples of category B are typically those closer to having all four attribute values unset. Example E2 is interesting because it is the only one with two set values which is in category A.

These data can be used to create different categories of objects, depending on what interpretation is given to the attributes. For example, by making A0 eye height, A1 eye separation, A2 nose length, and A3 mouth height, we obtain the face experiment [9, 2]. The thirty different experiments reported by Smith and Minda [12] used different interpretations of the attributes, and varying instructions for the participants.

If this were a machine learning application, it would be relatively straightforward to select an algorithm, with the criterion for success being generalisation error on the transfer items. However, in a psychological experiment, what is of interest is the behaviour of human participants when carrying out the task. For example, having learnt the training items, what proportion of participants obtain a correct response, both for the training and the transfer items? How long does it take a participant to obtain a response for each item? How many errors are made whilst learning the training items?

Table 2 provides some figures taken from experimental data attempting to answer just these questions. The two columns for response probability are taken from Smith and Minda [12]. The first of the two columns is data from a single experiment, averaged over a number of participants. The second is the average result over 30 experiments, performed by different groups of researchers. The figures for mean classification time are taken from Gobet *et al.* [2]. As can be seen, there is a large amount of data to capture for what appears, computationally, a very simple categorisation problem.

2.2 Defining Cognitive Models

Once the experimentalist has gathered their data, it is now the turn of the theorist to use the data to create an explanation of the behaviour. An important tool for the theorist is the construction of computational models, simulations of the behaviour which can be used to generate predictions of how humans will perform. A good theory will produce models which accurately predict behaviour in a wide range of experiments, with the minimum of 'parameter tweaking'.

There is a wide variety of theories and models available to the cognitive scientist. However, there are few comparative studies of how different classes of models relate to one another, and especially how different models compare on the same empirical data. One of the missing tools in the theorist's toolkit is a mechanism for setting up and performing such comparative studies: the framework proposed in this paper aims to fill this gap.

Our framework begins with a behavioural specification for computational models. We extend a technique proposed by Lane and Gobet [5], which suggests that a computational model should be released in three components: (1) a well-documented implementation; (2) a set of tests illustrating each of the key processes within the model; and (3) a set of canonical results, for reproducing the model's predictions in important experiments.

Components 1 and 2 relate to how the model is defined. For example, the implementation might be for a neural network, and the tests would check that the network performed back-propagation correctly. Component 3 is more interesting, as it provides a rigorous regression test for the model. Lane and Gobet [5] intended the tests to ensure models developed using this framework could always be confirmed against their prior evidence, removing the (usually unvoiced) doubt that version N does everything that version (N-1) did.

The set of canonical tests proposed as component 3 form the foundation for the current work. A canonical test would be, for example, to compute the fit of a model against the data in the first column of Table 2. Before the test could be applied, the model would have to run through the entire experiment, like the human, i.e. be trained, and then predict responses for each of the examples. The test would then be whether the predicted results were 'close enough' to those observed in the experiments. We take the value of these tests further, by not restricting them to a single model. Instead, canonical results will be a requirement of all models, of whatever type, which claim to be models of, in this case, categorisation.

3 Finding Optimal Models

In Section 4.1 we describe some concrete examples of cognitive theories and the models they specify. For now, we assume the existence of a class of models, $\mathcal{M}$, which we want to search. Our aim is to find one or more $m \in \mathcal{M}$ satisfying one or more of the experimental constraints, the canonical results described above.

Each canonical result is used for computing a *constraint*, referred to as $f_i(m)$. The constraint is a function of m, which returns a measure of how well the model fits the constraint. Without loss of generality, we assume that we want to minimise $f_i(m) \geq 0$.

We now have a classic optimisation problem, with a set of constraints and a variable to modify. How we solve this problem depends, in part, on the question which the theorist wants to answer. For example, locating individual models which optimise their behaviour on specific constraints is a single-criterion optimisation problem, which may be solved using a number of techniques; we use Genetic Algorithms [4].

However, if we instead wish to locate models which satisfy a number of the experimental constraints simultaneously, the problem becomes one of multi-criteria optimisation. The difficulty with a multi-criteria problem is in finding an appropriate definition of optimality. In our domain, we typically find that one model performs optimally on some of the experiments, whereas another beats it elsewhere. In order to respect all of our criteria simultaneously, we would like to obtain the *set* of models which satisfy the constraints. We wish to find those models which are not worse in *all* constraints than any other model.

Formally, we define *dominated* and *non-dominated* solutions. Model m_1 dominates model m_2 if,

$$\forall i \bullet f_i(m_1) \leq f_i(m_2) \ \wedge \ \exists j \bullet f_j(m_1) < f_j(m_2)$$

In other words, m_1 does at least as well as m_2 everywhere, but there is at least one experiment in which m_1 does better. The set of non-dominated solutions to a multi-criteria optimisation problem is known as the *pareto-optimal* set.

As first suggested by Schaffer [11], Genetic Algorithms are natural candidates for finding pareto-optimal sets, because they manipulate a population of candidates. Goldberg [3] proposed a technique of nondominated sorting, which we adapt here.

The genetic algorithm maintains a separate population of individuals for each of the model classes. Each separate population is evolved using a standard genetic algorithm. The only contact between the populations is in the fitness function. Our fitness function gives an individual a perfect fitness if it is a non-dominated member of the entire set of individuals. This ensures that it is maintained within its own population, irrespective of its performance relative to its neighbours.

4 Specifying the Search Space

4.1 A Space of Cognitive Models

The motivation behind different models of cognitive behaviour varies across cognitive science. Three important classes are: mathematical models, which are motivated purely by trying to explain particular behaviours; discrimination-network models, which explain high-level patterns of human memory; and connectionist models, which loosely model low-level neuronal activity in the brain. We describe how the model works, the parameters which define the model, and how it is used to predict values for the main kinds of constraint: probability of responding with category A, time to respond, and average number of errors in training.

4.1.1 Mathematical Models

Smith and Minda [12] consider eight mathematical models, most taken from earlier psychological literature on categorisation. We describe one of these here to illustrate the kind of model being considered. The aim of all eight models is to compute, from the training examples, the probability of responding with category A to a given example. For the *context model*, the probability of responding with category A given exemplar E_i is defined as:

$$\mathrm{P}(A|E_i) = \frac{\sum_{j \in C_A} \eta_{ij}}{\sum_{j \in C_A} \eta_{ij} + \sum_{j \in C_B} \eta_{ij}}$$

where

$$d_{ij} = c\left[\sum_{k=0}^{3} w_k |x_{ik} - x_{jk}|\right]$$

x_{ik} and x_{jk} are the values of the item and exemplar on attribute k, w_k is the attentional weight to attribute k, c is the sensitivity parameter, $C_{A/B}$ are the training examples in category A/B, and $\eta_{ij} = e^{-d_{ij}}$.

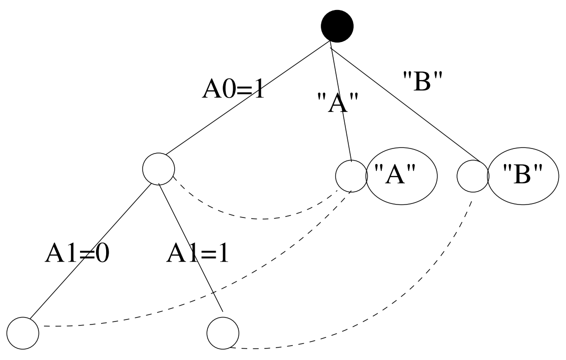

Figure 1: An example discrimination network, classifying the 5-4 structure

The model is completed with a ‘guessing’ parameter, used to model the fact that sometimes people simply guess a category, without doing any reasoning. The probability that the model will guess is γ, with a 0.5 probability of choosing category A.

The model provides a direct probability of responding with category A, which may be compared directly with the gathered experimental data. Although there is no learning, we may estimate the number of errors during training from the expected error. As the model does not capture the processes occurring within the experimental participant, no timing information is provided by the model. However, to facilitate the experiments in this paper, we add a parameter to the model, *classification time*, (*ct*), whose value gives the fixed time to make a classification.

Similar to the context model is the *prototype* model, which uses a similar set of formulae, but calculates similarity to a canonical example, not all the examples. The two mathematical models are defined by seven parameters: $\{ w_0, w_1, w_2, w_3, c, \gamma, ct \}$, where $\Sigma_{i=0}^{3} w_i = 1$

4.1.2 Discrimination-network models

Discrimination-network models, such as EPAM [2] or CHREST [1], capture the *processes* underlying behaviour. The core representation for learnt knowledge is a discrimination network, as illustrated in Figure 1. Information is stored as chunks at individual nodes. Tests on the links between nodes are used when sorting a pattern from the root node (the black disc). The dashed links indicate *naming links*, which are used by CHREST to associate categories to perceived information. The discrimination network is built up incrementally as the model is given each input in turn.

The model gives a definite categorisation for each stimulus, so, to obtain probalities of response, we need to simulate a number of models. Variations in the level of performance are achieved by making learning conditional. When a training example is given to each model, it will attempt to learn the pattern based on a probability of learning, ρ.[1] The probability of response is then obtained from the number of trained models which produce category A in response to each example.

Mean number of errors is given by counting the errors of all the models during learning. Time to categorise is given by parameters, which govern the amount of time it takes to sort a stimulus through the discrimination network. The times are computed as follows: a time to react to the stimulus, r, and a time to sort through each of the tests, t.

The CHREST model is defined by three parameters: $\{ \rho, r, t \}$

4.1.3 Connectionist models

The typical connectionist network [8] comprises a set of nodes interconnected by weighted links. The links pass activation between the nodes. Each node computes a simple function. Learning is a process of modifying the weights on the links between nodes. We use a simple, single-unit network (a perceptron) to model the categorisation experiment, similar to the mathematical models, except the decision function for choosing category A is:

$$\sum_{i=0}^{3} w_i \times x_i \; > \; \theta$$

The model is a *learning* model, and the weights, w_i, may be trained using the rule:

$$\Delta w_i = \eta \times x_i \times (target - output)$$

where *output* is the current output value, and *target* is the desired output. η is a parameter governing the learning rate.

Rather like the discrimination-network models, connectionist models have a definite output, and so, to get a population of models, we train 100 models to run the experiments. The model is defined by two independent parameters: η, the learning rate, and ρ, the probability of learning an example.

Errors during training are counted, to obtain the average error rate during learning. There is no equivalent of time to categorise, so, as with the mathematical models, we add a parameter for the *categorisation time*.

The perceptron is defined by four parameters: $\{ \theta, \eta, \rho, ct \}$

4.2 The Constraints

Given a model and an experiment, we can obtain the predicted behaviour of the model on the experiment. Considering the probability of response, we thus obtain 16 predicted and target figures for each (experiment, model) pair. Following standard practice in psychology, these 16 figures are converted into a single measure of fit between the target behaviour and the model. This measure may be computed in different ways, of which we consider two: the sum-squared error (SSE) and the average-absolute difference (AAD).

$$\begin{aligned} \text{SSE} &= \sum_{i=0}^{15} (p_i - t_i)^2 \\ \text{AAD} &= \left[\sum_{i=0}^{15} |p_i - t_i| \right] / 16 \end{aligned}$$

where p_i is the i^{th} predicted response, and t_i the i^{th} target behaviour.

The criteria are thus defined by selecting the experiment to perform, a target response to model, and a measure of fitness. Results are reported by describing the constraint. For example, in Table 3, the constraint 'SSE 1st' refers to using the first piece of experimental data, and fitting using sum-squared error. (The 1st, 2nd, 3rd etc refer to the data presented in Smith and Minda [12]. AVG is the average across all the thirty datasets considered there. Target figures for 1st and AVG can be found in Table 2.)

5 Representing the Models

In order to apply a genetic algorithm to search the space of models, the models must all be represented in a consistent manner. The models are represented as a list:

`(model-type param-1 param-2 ...)`

The model-type will define which of the classes of models the definition refers to, e.g. context-model, or discrimination-network. The parameters are simply the real numbers defining the model, as described above. Thus, a context model would be defined as:

(`context-model` w_0 w_1 w_2 w_3 c γ t)

To facilitate cross-over between models, we have a fixed size of list, so that every model is de-

1. This parameter is taken from Gobet *et al.* [2].

(a)	Mathematical		Connectionist	Discrimination-network
Constraint	Context	Prototype	Perceptron	CHREST
SSE 1st	0.053	0.186	0.180	0.893
SSE 2nd	0.033	0.124	0.267	0.528
SSE Avg	0.022	0.073	0.078	0.507
AAD 1st	0.045	0.085	0.099	0.191
AAD 2nd	0.030	0.065	0.083	0.144
AAD Avg	0.034	0.060	0.053	0.160
Time AAD	0.616	0.594	0.599	0.069

(b)	Mathematical		Connectionist	Discrimination-network
Constraint	Context	Prototype	Perceptron	CHREST
SSE 1st	0.143	0.270	0.443	1.117
SSE 2nd	0.117	0.187	0.136	0.556
SSE Avg	0.050	0.120	0.209	0.713
AAD 1st	0.064	0.099	0.142	0.206
AAD 2nd	0.064	0.081	0.075	0.136
AAD Avg	0.045	0.071	0.091	0.157
Time AAD	0.487	0.465	0.373	0.076

(c)	Mathematical		Connectionist	Discrimination-network
Constraint	Context	Prototype	Perceptron	CHREST
SSE 1st	0.118	0.249	0.339	1.183
SSE 2nd	0.126	0.198	0.222	0.570
SSE Avg	0.070	0.094	0.127	0.689
AAD 1st	0.067	0.100	0.091	0.218
AAD 2nd	0.046	0.080	0.151	0.160
AAD Avg	0.041	0.065	0.084	0.162

Table 3: Performance on each constraint separately: (a) model optimised on each constraint separately, (b) model optimised on sum of all constraints, (c) including non-dominated sorting.

fined with the same number of parameters; some model types will simply ignore the extra parameters. Cross-over is performed by simply picking a point in the list at random, and then creating two new individuals from the two halves. Mutation is performed by picking a parameter at random, and adding a random number between -1 and 1. (Or, for the model-type, picking a type at random from the list of possible models.) After any such operation, the new model is normalised, to ensure the parameters are suitable for the model type. For example, all the parameters should be positive numbers, and, some weights are rescaled to sum to 1.

6 Experiments

To illustrate the complexities and challenges with our approach, we discuss the results from three sets of experiments. The first develops optimal models for each criterion separately. These optimal models provide a base-line for considering the models found in the more complex, multi-criteria optimisation setting. The second uses a simple aggregative technique to pool all the criteria into a single measure of fitness. The third uses non-dominated sorting in a genetic algorithm to find a pareto-optimal set.

6.1 Optimising Single Models

Table 3(a) lists the optimised performance of individual models on selected criteria. The genetic algorithm was run separately for each class of models with a population of 100 individuals in each case, and for 100 cycles. The best performing individual model of each class was used in the evaluation.

6.2 Optimising Across All Criteria

6.2.1 Single Theory Optimisation

A simplistic technique for optimising across multiple criteria is to create a single fitness function which includes all the criteria in one. As all our criteria are similar, minimising a function towards zero, the simplest combined fitness function is to sum the individual criteria together. The fitness function is: $\sum_{i=0}^{N} f_i(m)$, where N is the total number of constraints.

A simple genetic algorithm was used to evolve a population of 100 individuals, over 100 cycles. The algorithm was run separately for each of the model types. The performance of the model which performed best on the summed fitness function is reported in Table 3(b) against each of the separate criteria.

Cycle	Cont	Prot	Perc	CHREST
(a)				
1	100	100	100	100
2	0	0	0	400
(b)				
1	100	100	100	100
2	0	28	350	22
3	0	0	400	0

Table 4: Evolution of population proportion: (a) over all constraints, (b) without the time constraint.

6.2.2 Multiple Theory Optimisation

By creating our initial population with examples from all classes of models, we can make the different models compete with one another, and attempt to evolve examples of those theories. We seeded our population with 100 examples of each of the four example model classes, and tracked the proportion of each model class in subsequent generations. Each generation was created from the top 10% of the preceding generation. Table 4(a) shows the number of models of each type in the first few evolutionary cycles across all constraints. As can be seen, the CHREST family of models immediately dominates the field, as the algorithm prematurely converges. Table 4(b) shows the result when the criterion for time was removed from the fitness function; again, premature convergence is seen, this time to the connectionist models.

6.3 Multi-Criteria Optimisation

We maintain four independent populations of models, each population evolving separately with 100 examples, maintaining the top 10% of each population to generate each subsequent population. The fitness of an individual is 0 if it is a non-dominated member of the total four populations, or the computed sum of fitness against all the constraints, if not. Table 3(c) shows how the best models from each population perform against the separate criteria.

6.4 Discussion of Experiments

The experiments emphasise the value of using optimisation techniques when developing cognitive models. In particular, several of the optimum values obtained improved on those in the original publications. Table 3(a) gives results for optimising single model types on specific criteria. The context-model has a typical average sum-squared error of 0.02, whereas Smith and Minda [12] give a value of 0.065 for the same model with the same data. The CHREST model gives an average difference in the times of 69 ms, whereas Gobet *et al.* [2] report a difference of closer to 300 ms. The problem of multiple competing constraints is highlighted by the results in Table 3(b). For all constraints except time, the context model performs best. The other model types vary a bit in their performance, with the Perceptron and Prototype models being very close in their fits, but outperforming each other on different tasks. For the time to make a classification, the CHREST model outperforms the other classes quite comfortably.

From a cognitive perspective, the mathematical models have been heavily tailored towards giving a good performance on the 5-4 structure. They do this by using many parameters, which must be carefully tuned. The connectionist and discrimination-network models, however, begin from a more general conception of the underlying processes occurring within a human learner. In particular, CHREST can capture the changes in the response timings quite closely. The general sensitivity of the mathematical models to their parameter settings is sharply demonstrated in Table 4, where they cannot directly compete with the process models, and soon drop out of the population. This result has important general applicability, as it suggests that optimisation involving different types of individuals must take into account the rate of convergence of individual kinds of models. This result has led us to adopt separate populations when evolving models from multiple theories; for details see [6].

Table 3(c) demonstrates that the performance of individual models does not deteriorate (compared with Table 3(b)) when non-dominated sets of models are preferred. Obtaining non-dominated sets enables conclusions to be drawn about the validity of models against all criteria simultaneously.

7 Discussion

There have been some prior uses of optimisation to discover cognitive models. In particular, Ritter [10] analysed some connectionist models for grammar learning, and found that automated optimisation outperformed the hand generated models discussed in the literature. More recently, Tor and Ritter [13] used a genetic algorithm to optimise the parameters in a complex cognitive model of problem solving. There are two main differences between this earlier work and that presented here. First, we have provided a formal framework in which many different experimental results (constraints) and different models (variables) may be defined and manipulated in a multi-criteria optimisation problem. Second, by applying non-dominated sorting techniques to portions of this search space, we can answer more complex questions about suitable and optimised models.

In other work, we have further shown how the evolutionary process can be adapted to support the search for a wide range of models, as quickly as possible [6, 7]. Further work will expand the range of theories and domains, and develop tools to assist the modeller attempting to explain cognitive behaviour.

8 Conclusion

We have introduced a formal definition of the task of developing a cognitive model. Specifically, we have proposed that generic behavioural tests can be created for most kinds of empirical data gathered in psychological experiments. These generic tests can be used as constraints in an optimisation problem, with a parameterised space of models as the target variable. Our pilot experiment, developing a model of categorisation, demonstrated the value of optimisation within cognitive science. In particular, some of the models found by our system outperformed those in the published literature. The location of pareto-optimal sets is non-trivial in this domain, and instead locally optimal sets for specific criteria must be sought. Further work is needed to extend the range of models and experiments. Particularly important will be the construction of heuristics, or analysis tools, to target the evolution of appropriate models.

Acknowledgements

We thank Pat Langley for comments on an earlier version of this paper.

References

[1] F. Gobet, P. C. R. Lane, S. J. Croker, P. C-H. Cheng, G. Jones, I. Oliver, and J. M. Pine. Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, 5:236–243, 2001.

[2] F. Gobet, H. Richman, J. Staszewski, and H. A. Simon. Goals, representations, and strategies in a concept attainment task: The EPAM model. *The Psychology of Learning and Motivation*, 37:265–290, 1997.

[3] D. E. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning.* Reading, MA: Addison-Wesley, 1989.

[4] J. H. Holland. *Adaptation in natural and artificial systems.* Ann Arbor: The University of Michigan Press, 1975.

[5] P. C. R. Lane and F. Gobet. Developing reproducible and comprehensible computational models. *Artificial Intelligence*, 144:251–63, 2003.

[6] P. C. R. Lane and F. Gobet. Discovering predictive variables when evolving cognitive models. In *Proceedings of the Third International Conference on Advances in Pattern Recognition*, 2005.

[7] P. C. R. Lane and F. Gobet. Multi-task learning and transfer: The effect of algorithm representation. In *Proceedings of the ICML-2005 Workshop on Meta-Learning*, 2005.

[8] P. McLeod, K. Plunkett, and E. T. Rolls. *Introduction to Connectionist Modelling of Cognitive Processes.* Oxford, UK: Oxford University Press, 1998.

[9] D. L. Medin and E. E. Smith. Strategies and classification learning. *Journal of Experimental Psychology: Human Learning and Memory*, 7:241–253, 1981.

[10] F. E. Ritter. Towards fair comparisons of connectionist algorithms through automatically optimized parameter sets. In *Proceedings of the Annual Conference of the Cognitive Science Society*, pages 877–881. Hillsdale, NJ: Lawrence Erlbaum, 1991.

[11] J. D. Schaffer. *Some experiments in machine learning using vector evaluated genetic algorithms.* PhD thesis, Vanderbilt University, Nashville, 1984.

[12] J. D. Smith and J. P. Minda. Thirty categorization results in search of a model. *Journal of Experimental Psychology*, 26:3–27, 2000.

[13] K. Tor and F. E. Ritter. Using a genetic algorithm to optimize the fit of cognitive models. In *Proceedings of the Sixth International Conference on Cognitive Modeling*, pages 308–313. Mahwah, NJ: Lawrence Erlbaum, 2004.