Title
Simple Ways to Construct Search Orders

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 26(26)

ISSN
1069-7977

Authors
Dieckmann, Anja
Todd, Peter M.

Publication Date
2004

Peer reviewed

Simple Ways to Construct Search Orders

Anja Dieckmann (dieckmann@mpib-berlin.mpg.de)
Center for Adaptive Behavior and Cognition, Max Planck Institute for Human Development, Lentzeallee 94, 14195 Berlin, Germany

Peter M. Todd (ptodd@mpib-berlin.mpg.de)
Center for Adaptive Behavior and Cognition, Max Planck Institute for Human Development, Lentzeallee 94, 14195 Berlin, Germany

Abstract

Simple decision heuristics that process cues in a particular order and stop considering cues as soon as a decision can be made have been shown to be both accurate and quick. But one criticism of heuristics such as Take The Best is that these owe much of their simplicity and success to the not inconsiderable computations necessary for setting up the cue search order before the heuristic can be used. The criticism, though, can be countered in two ways: First, there are typically many cue orders possible that will achieve good performance in a given problem domain. And second, as we will show here, there are simple learning rules that can quickly converge on one of these useful cue orders through exposure to just a small number of decisions. We conclude by arguing for the need to take into account the computation necessary for not only the application but also the setup of a heuristic when talking about its simplicity.

One-Reason Decision Making and Ordered Search

In the book *Simple heuristics that make us smart*, Gigerenzer and colleagues (1999) propose several decision making heuristics for predicting which of two objects or options, described by multiple binary cues, scores higher on some quantitative criterion. These heuristics have in common that information search is stopped once one cue is found that discriminates between the alternatives and thus allows an informed decision. No integration of information is involved, leading these heuristics to be termed "one-reason" decision mechanisms. These heuristics differ only in the search rule that determines the order in which information is searched. But where do these search orders come from?

"Take the Best" (TTB; Gigerenzer & Goldstein, 1996, 1999) is the heuristic that has received most attention to date, both theoretically and empirically. TTB consists of three building blocks:

1. Search rule: Search through cues in the order of their validity. Validity is the proportion of correct decisions made by a cue out of all the times that cue discriminates between pairs of options.

2. Stopping rule: Stop search as soon as one cue is found that discriminates between the two options.

3. Decision rule: Select the option to which the discriminating cue points, that is, the option that has the cue value associated with higher criterion values.

The performance of TTB has been tested on several real-world data sets, ranging from professors' salaries to fish fertility (Czerlinski, Gigerenzer & Goldstein, 1999). Cross-validation comparisons have been made against other more complex strategies, such as multiple linear regression, by training on half of the items in each data set to get estimates of the relevant parameters (e.g., cue order based on validities for TTB, beta-weights for multiple linear regression) and testing on the other half of the data. Despite only using on average a third of the information employed by multiple linear regression, TTB outperformed regression in accuracy when generalizing to the test set (71% vs. 68%).

The even simpler heuristic Minimalist was tested in the same way. It is another one-reason decision making heuristic that differs from TTB only in its search rule. Minimalist searches through cues randomly, and thus requires even less knowledge and precomputation than TTB – all it needs to know are the directions in which the cues point. Again it was surprising that this heuristic performed reasonably close to multiple regression (65%). But the fact that Minimalist lagged behind TTB by a noticeable margin of 6 percentage points indicates that part of the secret of TTB's success lies in its ordered search.

In this paper, we explore how such useful cue orders can be constructed in the first place, by testing a variety of simple order-learning rules in simulation. We find that simple mechanisms at the learning stage can enable simple mechanisms at the decision stage, such as one-reason decision heuristics, to perform well.

Experimental Evidence for Ordered Search

From an adaptive point of view, the combination of simplicity and accuracy makes one-reason decision making with ordered search, as in TTB, a plausible candidate for human decision processes. Consequently, TTB has been subjected to several empirical tests. Because TTB explicitly specifies information search as one aspect of decision making, it must be tested in situations in which cue information is not laid out all at once, but has to be searched for one cue at a time, either in the external environment or in memory (Gigerenzer & Todd, 1999).

In situations where information must be searched for sequentially in the external environment, particularly when there are direct search costs for accessing each successive cue, considerable use of TTB has been demonstrated (Bröder, 2000, experiments 3 & 4; Bröder, 2003). This also holds for indirect costs, such as from time pressure (Rieskamp & Hoffrage, 1999), as well as for internal search in memory (Bröder & Schiffer, 2003). The particular search order used has not always been tested separately, but when such an analysis at the level of building blocks has been done, search by cue validity order has received support (Newell & Shanks, 2003; Newell, Weston & Shanks, 2003).

However, none of these experiments tested search rules other than validity ordering. One other very important dimension on which cues can be ordered is discrimination rate, which refers to the proportion of all possible decision pairs in which a cue has different values for (i.e., discriminates between) the two alternatives[1]. A closer look into the experimental designs of the studies cited above reveals that they all used systematically constructed environments in which discrimination rates of the cues were held constant. Now, when the discrimination rates of cues are all the same, there are not many orders besides validity that make sense. To put it differently, identical discrimination rates make several alternative ordering criteria that combine discrimination rate and validity (e.g., Martignon & Hoffrage, 2002) all lead to the same (validity) order. Examples for such criteria are *success*, which is the proportion of correct discriminations that a cue makes plus the proportion of correct decisions expected from guessing in the non-discriminating trials (success = $v \cdot d + 0.5 \cdot (1-d)$, where v = validity and d = discrimination rate of the cue), and *usefulness,* the portion of correct decisions not including guessing (usefulness = $v \cdot d$).

Because these criteria collapse to a single order (validity) in the reported experiments, nothing can be said about how validity and discrimination rate may interact to determine the search orders that participants apply. It remains unclear what information participants would base their decisions on when both validity and discrimination rate vary. There are hints that when information is costly, making it sensible to consider both how often a cue will enable a decision (i.e., its discrimination rate) and the validity of those decisions, other criteria such as success that combine the two measures show a better fit to empirical data (e.g., Newell, Rakow, Weston & Shanks, in press; Läge, Hausmann, Christen & Daub, submitted). But these studies, too, remain silent about how these criteria, or an order based on these criteria, could possibly be derived by participants.

In sum, despite accumulating evidence for the use of one-reason decision making heuristics, the basic processes that underlie people's search through information when employing such heuristics remain a mystery. While some clues can be had by considering the size of the overlap or correlations between the search orders people use and various standard search orders (as reported by Newell et al., in press, and Läge et al., submitted), they do not come close to telling us how cue orders could possibly be learned.

Search Order Construction – the Hard Way

But how can the search order of TTB be constructed? Although TTB is a very simple heuristic to apply, the set-up of its search rule requires knowledge of the ecological validities of cues. This knowledge is probably not usually available in an explicit precomputed form in the environment, and so must be computed from stored or ongoing experience. Gigerenzer at al. (1999) have been relatively silent about the process by which people might derive validities and other search orders, a shortfall several peers have commented on (e.g., Lipshitz, 2000; Wallin & Gärdenfors, 2000). The criticism that TTB owes much of its strength to rather comprehensive computations necessary for deriving the search order cannot easily be dismissed. Juslin and Persson (2002) pay special attention to the question of how simple and informationally frugal TTB actually is, debating how to take into account the computation of cue validities for deriving the search order. They differentiate two main possibilities on the basis of when cue validities are computed: precomputation during experience, and calculation from memory when needed.

When potential decision criteria are already known at the time objects are encountered in the environment, then relevant validities can be continuously computed and updated with each new object seen. But if it is difficult to predict what decision tasks may arise in the future, this pre-computation of cue validities runs into problems. In that case, at the time of object exposure, all attributes should be treated the same, because any one could later be either a criterion or a cue depending on the decision being made. To use the well-known domain of German cities (Gigerenzer & Goldstein, 1996, 1999), the task that one encounters need not be the usual prediction of city populations based on cues such as train connections, but could just as well be which of two cities has an intercity train line based on cues that include city population. To keep track of all possible validities indicating how accurately one attribute can decide about another, the number of precomputed validities would have to be $C^2 - C$, with C denoting to the number of attributes available. In the German cities example, there are 10 attributes (9 cues plus the criterion population size), thus 90 validities would have to be pre-computed. This number rises rapidly with increasing number of attributes. Even ignoring computational complexity, this precomputation approach is not frugal in terms of information storage.

As a second possibility, Juslin and Persson (2002) consider storing all objects (exemplars) encountered along with their attribute values and postponing computation of validities to the point in time when an actual judgment is required. This, however, makes TTB considerably less frugal during its application. The number of pieces of information that would have to be accessed at the time of judgment is the number of attributes times the number of stored objects; in our city example, it is 10 times the number of known objects. With regard to computing validities for each of the $N \cdot (N-1)/2$ possible pairs that can be formed between the N known objects, each of the C cues has to be

[1] Other dimensions for ordering are possible, such as the temporal order of previous cue use, but we will not consider them here.

checked to see if it discriminated, and did so correctly. Thus the number of checks to be performed before a decision can be made is C·N·(N-1)/2, which grows with the square of the number of objects.

Although Juslin and Persson assume worst case scenarios in terms of computational complexity for the sake of their argument, they raise an important point, showing that one of the fundamental questions within the framework of the ABC research group (Gigerenzer et al., 1999) remains open: How can search orders be derived in relatively simple ways?

Many Roads Lead (Close) to Rome

From what we have said so far, the situation does not look too good for validity either in terms of empirical evidence or psychological feasibility. But what would be the consequence in terms of loss in accuracy if we drop the assumption that cue search follows the validity order? Simulation results can provide an answer. First of all, validity is usually not the best cue ordering that can be achieved. For the German city data set, Martignon and Hoffrage (2002) computed the performance of all possible orderings, assuming one-reason stopping and decision building blocks. The number of possible orders was 362,880 (9! orders of 9 cues). The mean accuracy of the resulting distribution corresponded to the performance expected from Minimalist, 70%, which was considerable above the worst ordering at 62%. Ordering cues by validity led to an accuracy of 74.2%, while the optimal ordering yielded 75.8% accuracy. More than half of all possible cue orders do better than the random order used by Minimalist, and 6,532 (1.8%) do better than the validity order. We can therefore conclude that many good orders exist. But how can one of these many reasonably good cue orders be constructed in a psychologically plausible way?

Search Order Construction – the Simple Way

A variety of simple approaches to deriving and continuously updating search orders can be proposed. Indeed, computer scientists have explored a number of self-organizing sequential search heuristics for the purpose of speeding retrieval of items from a sequential list when the relative importance of the items is not known a priori (Rivest, 1976; Bentley & McGeoch, 1985). The mechanisms they have focused on use transposition of nearby items and counting of instances of retrieval. Our problem of cue ordering is slightly different from that of the standard sequential list ordering, because cues can fail in ways that retrieved items cannot: a cue may not discriminate (necessitating the search for another cue before a decision can be made), or it may lead to a wrong decision. Still, the mechanisms of transposition and counting will be central to the heuristics we propose.

We focus on search order construction processes that are psychologically plausible by being frugal both in terms of information storage and in terms of computation. The decision situation we explore is different from the one assumed by Juslin and Persson (2002) who strongly differentiate between learning of (or about) objects and making decisions. Instead of assuming this unnecessary separation, we will explore a learning-while-doing situation. Certainly there are many occasions akin to Juslin and Persson's situation where individuals have to make decisions based on knowledge they have learned about objects encountered previously and in a different task context. But perhaps more common are tasks that have to be done repeatedly with feedback being obtained after each trial about the adequacy of one's decision. For instance, we can observe on multiple occasions which of two supermarket checkout lines, the one we have chosen or (more likely) another one, is faster, and associate this outcome with cues including the lines' lengths and the ages of their respective cashiers. In such situations, one can learn about the differential usefulness of cues for solving the task via the feedback received over time. It is this case – decisions made repeatedly with the same cues and criterion and the opportunity to learn from outcome feedback – which we will now look at more closely.

We consider several explicitly defined cue order learning rules that are designed to deal with probabilistic inference tasks. In particular, the task we use is forced choice paired comparison, in which a decision maker has to infer which of two objects, each described by a set of binary cues, is "bigger" on a criterion – the task for which TTB was formulated. Thus, in contrast to Juslin and Persson (2002), we assume individuals encounter decision situations instead of objects. After an inference has been made, feedback is given about whether a decision was right or wrong. Therefore, the learning algorithm has information about which cues were looked up, whether a cue discriminated, and whether a discriminating cue led to the right or wrong decision. There are different possibilities for taking these pieces of information into account. For example, correct decisions could be counted up for each cue (essentially keeping tallies). Or the information could be used to compute cue validities and discrimination rates based on the cases in which the cue has actually been looked up so far. These tallies, validity estimates, etc., would then be used for creating and adjusting the current cue order.

The rules we propose differ in the pieces of information they use and how they use them. We classify the learning rules based on their memory requirement – high versus low – and their computational requirements (see Table 1). The computational requirements include whether the entire set of cues is completely reordered after each decision or only adjusted locally via swapping of neighboring cue positions, and whether reordering is done on the basis of measures involving division, such as validity, or simple tallying.

The *validity rule* is the most demanding of the rules we consider in terms of both memory requirements and computational complexity. It keeps a count of all discriminations made by a cue so far (in all the times that the cue was looked up) and a separate count of all the correct discriminations. Therefore, memory load is comparatively high. The validity of each cue is determined by dividing its current correct discrimination count by its

Table 1: Learning rules classified according to memory and computational requirements

High memory load, complete reordering	High memory load, local reordering	Low memory load, local reordering
Validity: reorders cues based on their current validity Tally: reorders cues by number of correct minus incorrect decisions made so far	Tally swap: moves cue up (down) one position if it has made a correct (incorrect) decision if its tally of correct minus incorrect decisions is ≥ (≤) that of next higher (lower) cue	Simple swap: moves cue up one position if it has made a correct decision, and down if it has made an incorrect decision
Variants: - *reorder based on tally of discriminations so far* - *reorder based on tally of correct decisions only*	*Variants:* - *only upward swapping after correct decisions* - *tally of correct decisions only*	*Variants:* - *moving cues more than one position* - *only upward swapping after correct decisions*

total discrimination count. Based on these values computed after each decision, the rule reorders the whole set of cues from highest to lowest validity.

The *tally rule* only keeps one count per cue, storing the number of correct decisions made by that cue so far minus the number of incorrect decisions. So if a cue discriminates correctly on a given trial, one point is added to its tally. If it leads to an incorrect decision, one point is subtracted from its tally. The tally rule is less demanding both in terms of memory and computation: Only one count is kept, and no division is required.

While the validity and tally rules rely on a counting mechanism, the *simple swap rule* uses the principle of transposition (cf. Bentley & McGeoch, 1985). This rule has no memory of cue performance other than an ordered list of all cues, and just moves a cue up one position in this list whenever it leads to a correct decision, and down if it leads to an incorrect decision. In other words, a correctly deciding cue swaps positions with its nearest neighbor upwards in the cue order, and an incorrectly deciding cue swaps positions with its nearest neighbor downwards.

The *tally swap rule* is a hybrid of the simple swap rule and the tally rule. It keeps a tally of correct minus incorrect discriminations per cue so far (so memory load is high) but only locally swaps cues: When a cue makes a correct decision and its tally is greater than or equal to that of its upward neighbor, the two cues swap positions. When a cue makes an incorrect decision and its tally is smaller than or equal to that of its downward neighbor, the two cues also swap positions.

As indicated in table 1, many variants of these basic types of learning rules are possible. Here we will focus on these four rules spanning the space of possibilities, and look at how they perform in simulations. Elsewhere we consider evidence for their use in experimental decision settings, and use these simulation results to assess human performance.

Simulation Study of Simple Ordering Rules

To test the performance of these order learning rules, we use the German cities data set (Gigerenzer & Goldstein, 1996), consisting of the 83 highest-population German cities described on 9 cues. The question we want to address is, what would happen if a decision-maker does not search for cues in validity order from the beginning, but instead must construct a search order using feedback received about each decision made? We assume that cue directions are known. Furthermore, instead of allowing the decision maker to look up information about all 9 cues in each pair comparison, we assume that TTB's stopping and decision rule are used on all decisions. We do this because it is more natural to assume that learning happens in the ongoing context of decision making, which does not necessarily involve exhaustive information search. This runs counter the approach taken by Juslin and Persson (2002) who in their worst case scenarios assume exhaustive information search for validity computations. In our approach, only the limited information gathered until the first discriminating cue is found can be taken into account.

We simulated 10,000 learning trials for each rule, starting from random initial cue orders. Each trial consisted of 100 decisions between randomly selected decision pairs. Below we report average values across the 10,000 trials.

Results

We start by considering the cumulative accuracies (i.e., online or amortized performance – Bentley & McGeoch, 1985) of the rules, defined as the total percentage of correct decisions made so far at any point in the learning process. (The contrasting measure of offline accuracy – how well the current learned cue order would do if it were applied to the entire test set – is a less psychologically useful indication of a real decision maker's performance using some rule.) The mean cumulative accuracies of the different search order learning rules when used with one-reason decision making are shown in Figure 1. Cumulative accuracies soon rise above that of the Minimalist heuristic (proportion correct = 0.70) which looks up cues in random order and thus serves as a lower benchmark. However, at least throughout the first 100 decisions, cumulative accuracies stay well below the (offline) accuracy that would be achieved by using TTB for all decisions (proportion correct = 0.74), looking up cues in the true order of their ecological validities.

The four learning rules all perform on a surprisingly similar level, with less than one percentage point difference in favor of the most demanding rule (i.e., validity) compared to the least (i.e., simple swap; mean proportion correct in 100 decisions: validity learning rule: 0.719; tally: 0.716; tally swap: 0.715; simple swap: 0.711). Importantly, though, the more demanding learning rules outperform Minimalist earlier. Whereas the tally swap and simple swap rule lead to accuracies that are significantly higher than Minimalist only after 48 and 61 decisions, respectively, the validity learning rule does significantly better already after 37 decisions, and the tally rule after 35 decisions ($z = 1.65$, $p = 0.05$).

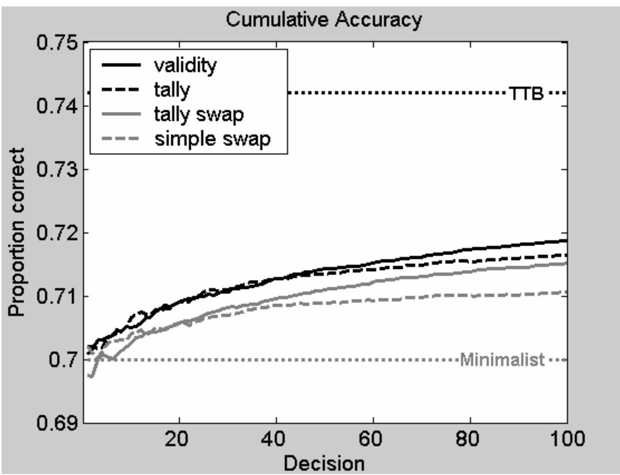

Figure 1: Mean cumulative accuracy of order learning rules

These four learning rules are, however, all more *frugal* than TTB, and even more frugal than Minimalist. On average, they look up fewer cues before reaching a decision (see Figure 2). Again, there is little difference between the rules (mean number of cues looked up in 100 decisions: validity learning rule: 3.17; tally: 3.07; tally swap: 3.13; simple swap: 3.18). The validity learning rule and the tally rule lead to cue orders that are significantly more frugal than Minimalist very early (after 16 and 14 decisions, respectively), whereas the two swapping rules take longer: The tally swap rule takes 27 decisions, and the simple swap rule 32 decisions.

Consistent with this finding, all of the learning rules lead to cue orders that show positive correlations with the discrimination rate cue order (reaching the following values after 100 decisions: validity learning rule: $r = 0.18$; tally: $r = 0.29$; tally swap: $r = 0.24$; simple swap: $r = 0.18$). This means that cues that often lead to discriminations are more likely to end up in the first positions of the order. In contrast, the cue orders resulting from all learning rules but the validity learning rule do not correlate with the validity cue order, and even the correlations of the cue orders resulting from the validity learning rule after 100 decisions only reach an average $r = 0.12$.

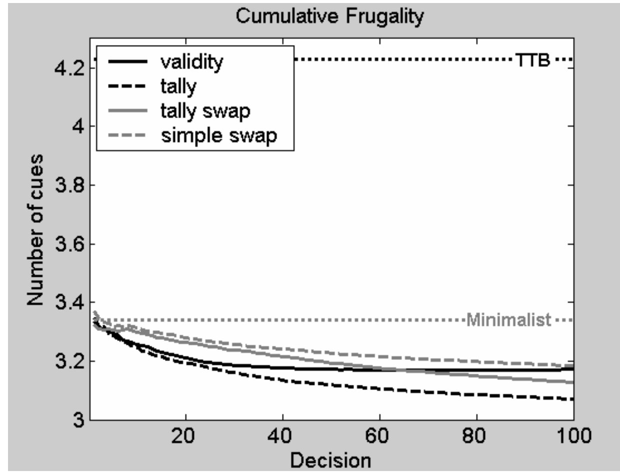

Figure 2: Mean cumulative frugality of order learning rules

But why would the discrimination rates of cues exert more of a pull on cue order than validity, even when the validity learning rule is applied? Part of the explanation comes from the fact that in the city data set we used for the simulations, validity and discrimination rate of cues are negatively correlated. Having a low discrimination rate means that a cue has little chance to be used and hence to demonstrate its high validity. Whatever learning rule is used, if such a cue is displaced downward to the lower end of the order by other cues, it may never be able to escape to the higher ranks where it belongs. The problem is that when a decision pair is finally encountered for which that cue would lead to a correct decision, it is unlikely to be checked because other, more discriminating although less valid, cues are looked up before and already bring about a decision. Thus, because one-reason decision making is intertwined with the learning mechanism and so influences which cues can be learned about, what mainly makes a cue come early in the order is producing a high *number* of correct decisions and not so much a high *ratio* of correct discriminations to total discriminations regardless of base rates.

In sum, all of the learning rules lead to accuracies between that of the heuristics TTB and Minimalist, but some rules reach orders that are better than Minimalist sooner. The rules are highly frugal, with a (slight) tendency to change the order in the direction of discrimination rate.

Discussion

The simpler cue order learning rules we have proposed do not fall far behind a validity learning rule in accuracy. This holds even for the simplest rule, which only requires memory of the last cue order used and moves a cue one position up in that order if it made a correct decision, and down if it made an incorrect decision. All of the rules considered here make one-reason decision heuristics perform above the level of Minimalist in the long run.

On the other hand, the four rules, even the validity learning rule, stay below TTB's accuracy across a relatively high number of decisions. But often it is necessary to make good decisions without much experience. Therefore, learning rules should be preferred that quickly lead to orders with good performance. Both the validity and tally learning rules quickly beat Minimalist. At the same time, the tally rule leads to considerably more frugal cue orders.

Remember that the tally rule assumes full memory of all correct minus incorrect decisions made by a cue so far. But this does not make the rule implausible. There is considerable evidence that people are actually very good at remembering the frequencies of events. Hasher and Zacks (1984) conclude from a wide range of studies that frequencies are encoded in an automatic way, implying that people are sensitive to this information without intention or special effort. Estes (1976) pointed out the role frequencies play in decision making as a shortcut for probabilities. Further, the tally rule is comparatively simple, not having to keep track of base rates or perform divisions as does the validity rule. From the other side, the simple swap rule may not be much simpler, because storing a cue order may be about as demanding as storing a set of tallies. We therefore conclude that the tally rule should not be discounted on grounds of implausibility without further empirical evidence. Of course, a necessary next step (currently underway) will be to test how well these and other rules predict people's information search when they have to make cue-based inferences without knowing validities.

Our goal in this paper was to argue for the necessity of taking into account the set-up costs of a heuristic in addition to its application costs when considering the mechanism's overall simplicity. As we have seen from the example of the validity search order of TTB, what is easy to apply may not necessarily be so easy to set up. But simple rules can also be at work in the construction of a heuristic's building blocks. We have proposed such rules for the construction of one building block, the search order. We have seen that these simple learning rules enable a one-reason decision heuristic to perform only slightly worse than if it had full knowledge of cue validities from the very beginning. Giving up the assumption of full a priori knowledge for the slight decrease in accuracy seems like a reasonable bargain: Through the addition of learning rules, one-reason decision heuristics might lose some of their appeal to decision theorists who were surprised by the performance of such simple mechanisms compared to more complex algorithms, but they gain psychological plausibility and thus become more attractive as explanations for human decision behavior.

References

Bentley, J.L. & McGeoch, C.C. (1985). Amortized analyses of self-organizing sequential search heuristics. *Communications of the ACM, 28(4),* 404-411.

Bröder, A. (2000). Assessing the empirical validity of the "Take-The-Best" heuristic as a model of human probabilistic inference. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 26 (5)*, 1332-1346.

Bröder, A. (2003). Decision making with the "adaptive toolbox": Influence of environmental structure, intelligence, and working memory load. *Journal of Experimental Psychology: Learning, Memory, & Cognition, 29,* 611-625.

Bröder, A. & Schiffer, S. (2003). "Take The Best" versus simultaneous feature matching: Probabilistic inferences from memory and effects of representation format. *Journal of Experimental Psychology: General, 132,* 277-293.

Czerlinski, J., Gigerenzer, G., & Goldstein, D.G. (1999). How good are simple heuristics? In G. Gigerenzer, P.M. Todd & The ABC Research Group, *Simple heuristics that make us smart*. New York: Oxford University Press.

Estes, W.K. (1976). The cognitive side of probability learning. *Psychological Review, 83*, 37-64.

Gigerenzer, G., & Goldstein, D.G. (1996). Reasoning the fast and frugal way: Models of bounded rationality. *Psychological Review, 103 (4)*, 650-669.

Gigerenzer, G., & Goldstein, D.G. (1999). Betting on one good reason: The Take The Best Heuristic. In G. Gigerenzer, P.M. Todd & The ABC Research Group, *Simple heuristics that make us smart*. New York: Oxford University Press.

Gigerenzer, G., & Todd, P.M. (1999). Fast and frugal heuristics: The adaptive toolbox. In G. Gigerenzer, P.M. Todd & The ABC Research Group, *Simple heuristics that make us smart*. New York: Oxford University Press.

Gigerenzer, G., Todd, P.M., & The ABC Research Group (1999). *Simple heuristics that make us smart.* New York: Oxford University Press.

Hasher , L., & Zacks, R.T. (1984). Automatic Processing of fundamental information: The case of frequency of occurrence. *American Psychologist, 39*, 1372-1388.

Juslin, P., & Persson, M. (2002). PROBabilities from EXemplars (PROBEX): a "lazy" algorithm for probabilistic inference from generic knowledge. *Cognitive Science, 26*, 563-607.

Läge, D., Hausmann, D., Christen, S. & Daub, S. (submitted). Take The Best: How much do people pay for validity?

Lipshitz, R. (2000). Two cheers for bounded rationality. *Behavioral and Brain Sciences, 23*, 756-757.

Martignon, L., & Hoffrage, U. (2002). Fast, frugal, and fit: Simple heuristics for paired comparison. *Theory and Decision, 52*, 29-71.

Newell, B.R., Rakow, T., Weston, N.J., & Shanks, D.R. (in press). Search strategies in decision-making: the success of 'success'. *Journal of Behavioral Decision Making.*

Newell, B.R., & Shanks, D.R. (2003). Take the best or look at the rest? Factors influencing 'one-reason' decision making. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 29*, 53-65.

Newell, B.R., Weston, N.J., & Shanks, D.R. (2003). Empirical tests of a fast and frugal heuristic: Not everyone "takes-the-best". *Organizational Behavior and Human Decision Processes, 91*, 82-96.

Rieskamp, J., & Hoffrage, U. (1999). When do people use simple heuristics, and how can we tell? In G. Gigerenzer, P.M. Todd & The ABC Research Group, *Simple heuristics that make us smart*. New York: Oxford University Press.

Rivest, R. (1976). On self-organizing sequential search heuristics. *Communications of the ACM, 19(2),* 63-67.

Wallin, A. & Gärdenfors, P. (2000). Smart people who make simple heuristics work. *Behavioral and Brain Sciences, 23*, 765.