International Journal of Semantic Computing

Using the Hurwicz Criterion to Optimise Selection Queries Under Partial Ignorance*

Sven Helmer
Department of Informatics, University of Zurich, 8050 Zurich, Switzerland

Peter T. Wood
School of Computing and Mathematical Sciences, Birkbeck, University of London, London WC1E 7HX, United Kingdom

Manuel Stuefer
Faculty of Computer Science, Free University of Bozen-Bolzano, 39100 Bolzano, Italy

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Optimising queries in real-world situations under imperfect conditions is still a problem that has not been fully solved. We consider finding the optimal order in which to execute a given set of selection operators under partial ignorance of their selectivities. The selectivities are modelled as intervals rather than exact values and we apply various methods from decision theory in order to measure optimality. On the one hand, we evaluate the different techniques theoretically, showing which axioms of decision theory they satisfy and for which cases they efficiently provide an optimal solution. On the other hand, we also conduct an empirical evaluation, illustrating the differences of the approaches in terms of the relative error to the optimal solution. We demonstrate that a hybrid algorithm combining optimistic and pessimistic aspects performs best.

Keywords: Database query optimisation; selection ordering; Hurwicz criterion.

1. Introduction

Although query optimisation in database management systems (DBMSs) has been a topic of research for decades, there are still important unresolved issues. In a blog post [18], Guy Lohman highlights errors made in estimating cardinalities as a crucial factor. These kinds of errors cause optimisers to generate query execution plans that are way off the target in terms of efficiency. Consequently, an optimiser should try to avoid potentially bad plans rather than strive for an optimal plan based on unreliable information.

For typical workloads, a DBMS can compile statistical data over time to obtain a fairly accurate picture. For instance, estimating the selectivities of simple predicates on base relations in a relational database is fairly well understood and can be done quite accurately

[13, 15]. However, the situation changes once systems are confronted with very unevenly distributed data values or predicates that are complex.

Trying to estimate selectivities in dynamic settings, such as data streams [29], or in non-relational contexts, such as XML databases [27, 34], also poses challenges. It may even be impossible to obtain any statistical data, because the query is running on remote servers [33]. Detailed information may also not be available because a user issues an atypical ad-hoc query or utilises parameter markers in a query. We propose to use techniques from decision theory for making decisions under ignorance[a], meaning that we know what the alternatives and their outcomes are, but we are unable to assign concrete probabilities to them [26].

In our approach we propose to build a robust query optimiser that is aware of the unreliability of database statistics and considers this during optimisation. When executing a query, the DBMS encounters a particular instance of concrete parameter values: we call this a *scenario*. The problem is that, during the prior optimisation step, the optimiser does not know which scenario the DBMS will face during plan execution. We propose to use well-known techniques from decision theory to guide an optimiser in choosing a suitable plan.

In summary, we make the following contributions:

- We define what it means for an ordering of selection operators to be optimal according to the optimism-pessimism rule specified in decision theory.
- Additionally, we prove a number of properties, such as dominance, that help in optimising plans following the notion of optimism-pessimism.
- As the search space is too large to be traversed exhaustively, we develop a heuristic for optimising plans according to optimism-pessimism.
- Finally, in an experimental evaluation we show that our heuristic produces plans that come close to the optimum and are far better than those generated by competitors.

2. Related Work

In the following, we review different approaches for dealing with uncertain parameters during query optimisation. A common approach of many optimisers is to use the mean or modal value of the parameters and then find the plan with least cost under the assumption that this value remains constant during query execution, an approach called Least Specific Cost (LSC) in [8]. As Chu et al. point out in [8], if the parameters vary significantly, this does not guarantee finding the plan of least expected cost.

An alternative is to use probabilistic information about the parameters fed into the database optimiser, an approach known as Least Expected Cost (LEC) [8]. (A discussion regarding the circumstances under which LEC or LSC is best appears in [9].) In decision-theoretic terms, we are making decisions under risk, maximising the expected utility. How-

[a]Sometimes these are also called decisions under uncertainty. We refer to them as decisions under ignorance to distinguish them from probability-based methods.

ever, probability distributions for the possible parameter values are needed to make this approach work, whereas in our case we do not have these prerequisites.

In parametric query optimisation, several plans can be precompiled and then, depending on the query parameters, be selected for execution [12]. However, if there is a large number of optimal plans, each covering a small region of the parameter space, this becomes problematic. First of all, we have to store all these plans. In addition, constantly switching from one plan to another in a dynamic environment (such as stream processing) just because we have small changes in the parameters introduces a considerable overhead. In order to amend this, researchers have proposed reducing the number of plans at the cost of slightly decreasing the quality of the query execution [10]. Our approach can be seen as an extreme form of parametric query optimisation by finding a single plan that covers the whole parameter space.

Another approach to deal with the lack of reliable statistics is adaptive query processing, in which an execution plan is re-optimised while it is running [4, 6, 16, 20]. It is far from trivial to determine at which point to re-optimise and adaptive query processing may also involve materialising large intermediate results. More importantly, this means modifying the whole query engine; in our approach no modifications of the actual query processing are needed. A gentler approach is the incremental execution of a query plan [25]. Deciding on how to decompose a plan into fragments and putting them together is still a complex task, though.

Estimates based on intervals arise explicitly in [6] and implicitly in [23]. Babu et al. [6] use intervals to model uncertainty in the accuracy of a single-point estimate. Uncertainty is represented by a value from 0 (none) to 6 (very high). Upper and lower bounds for the single-point estimate are then calculated using the estimate and the uncertainty value. During optimisation, three scenarios, those using the low estimates, the exact estimates and the high estimates, are considered. Moerkotte et al. [23] study histograms which provide so-called q-error guarantees. Given an estimate $\hat{s}$ for s, the q-error of $\hat{s}$ is $\max(s/\hat{s}, \hat{s}/s)$. An estimate is q-acceptable if its q-error is at most q. So, if an estimate $\hat{s}$ is q-acceptable, the true value s lies in the interval $1/q \times \hat{s} \leq s \leq q \times \hat{s}$.

Notions of robustness in query optimisation have been considered in [5, 6, 20, 32]. Babcock and Chaudhuri [5] use probability distributions derived from sampling as well as user preferences in order to tune the predictability (or robustness) of query plans versus their performance. For Markl et al. [20], robustness means not continuing to execute to completion a query plan which is found to be suboptimal during evaluation; instead re-optimisation is performed. On the other hand, Babu et al. [6] consider a plan to be robust only if its cost is within e.g. 20% of the cost of the optimal plan. All of the techniques discussed so far need additional statistical information to work. Wolf et al. [32] introduce a parametric cost function (PCF) for cardinality or selectivity estimation on edges in query execution plans. This function measures the cost modelled on one cost parameter, typically the cardinality or selectivity. For commonly used cost models, PCFs are linear functions and the gentler their slopes, the more robust a plan is. This is a very interesting way to model robustness, although it is not used actively in a query optimisation step yet. At the moment it is used to test the robustness of plans generated by some other process. In summary, however, none

of the papers above consider robustness in the sense of decision theory.

3. Decision Theory

Generally speaking, *decision theory* is the theory of rational decision making where a *decision maker* chooses an *act* from a set of *alternatives*. The *outcome* depends on the *state of the world*, which often is only partially known to the decision maker. Each possible state of the world is also known as a *scenario*. Any decision matrix can now be represented by enumerating the possible states/scenarios as a set $S : \{s_1, s_2, \ldots, s_n\}$ and the possible alternatives as a set $A : \{a_1, a_2, \ldots, a_m\}$.

3.1. *Decisions under Ignorance*

Depending on the knowledge of the probabilities for each possible outcome, we can distinguish between decisions under *risk*, *ignorance*, and *uncertainty*. Making a decision under risk means that the decision maker knows the probabilities of the possible outcomes, while in a decision under ignorance, the probabilities are unknown or non-existent. The term uncertainty is either used as a synonym for ignorance or as a broader term referring to both, risk and ignorance. In our work, we focus on decisions under ignorance.

Example 1. Let us look at an example involving executing queries in a database system. Before being able to execute a query, we have to build a concrete query execution plan. Usually, there are countless equivalent plans that all produce the same output, albeit at different levels of quality regarding the efficiency. The quality of an execution plan depends on the state of the database, e.g. the number of tuples satisfying certain predicates, the presence (or absence) of indexes, and the size of the available memory. Let s_i denote a database state and a_i an execution plan (which is an action available to a decision maker here). A (hugely simplified) decision matrix may look as follows:

	s_1	s_2	s_3	s_4	s_5
a_1	8	9	6	1	3
a_2	6	8	3	4	7
a_3	9	10	8	5	8
a_4	6	8	7	7	6

This means that running plan a_1 in scenario s_1 is better than running plan a_2, as a_1 has a higher quality level. In scenario s_4 the situation changes, though: here plan a_2 is better. $\Diamond$

3.2. *Preference and Dominance*

Clearly, a decision maker needs to be able to compare different alternatives in order to choose one of them. In decision theory, the relation between alternatives is called a *preference relation* or *preference ordering*.

Definition 1. A *weak* preference relation, denoted by $\preceq$, means that given two alternatives a_i and a_j with $a_i \preceq a_j$, a decision maker considers a_j to be at least as preferable as a_i.

Two alternatives a_i and a_j are considered *equivalent* if a decision maker is indifferent about them: $a_i \sim a_j$, iff $a_i \preceq a_j$ and $a_j \preceq a_i$. For a *strict* preference relation, $\prec$, the following must hold: $a_i \prec a_j$, iff $a_i \preceq a_j$ and $a_j \npreceq a_i$.

For certain cases, it is straightforward for a decision maker to choose between alternatives. Looking at the decision matrix in Example 1, rationality forbids us to choose a_1 over a_3, because the latter always leads to a better, or equally good outcome, no matter which state happens to be the final true state of the world. We say that alternative a_3 dominates alternative a_1. More concretely, we define the notions of *weak dominance* and *strong dominance* as follows:

Definition 2. Let $v(a_i, s_j)$ represent the value of alternative a_i in state/scenario s_j of the decision matrix. We say that a_i *weakly dominates* a_j, if all outcomes under all states for a_i are at least as good as those for a_j (i.e, $\forall k : v(a_i, s_k) \geq v(a_j, s_k)$.

Definition 3. We say that a_i *strongly dominates* (or *strictly dominates*) a_j, if a_i weakly dominates a_j and the value of a_i in every state is strictly greater than that of a_j (i.e., $\forall k : v(a_i, s_k) > v(a_j, s_k)$).

3.3. *Different Strategies*

We now discuss some well-known strategies that a decision maker can choose from when trying to make decisions under ignorance. Basically, a strategy utilises an ordinal utility function that imposes an order on the different alternatives, representing the preferences of a decision maker. More formally, let u be a utility function, then for any two alternatives a_i and a_j, $u(a_i) \leq u(a_j)$, iff $a_i \preceq a_j$. Choosing an option then boils down to finding the alternative that maximises u. We cover the *Maximin*, *Maximax*, *Optimism-Pessimism*, and *Minmax Regret* approaches [26].

3.3.1. *Maximin*

Using the maximin principle, a decision maker *maximises* the *minimal* value obtainable with each action. In other words, if the worst possible outcome of one alternative is better than that of another, the former should be chosen. Formally, this means the following.

Definition 4. $\text{Maximin}(A, S) = \max_{a \in A}(\min_{s \in S}(v(a, s)))$

When applying this technique to Example 1, we can see that a_4 is chosen, as its worst result (for scenario s_1 and s_5) is the best among the worst results of all alternatives. This is a rather pessimistic strategy as it optimises the worst case.

If the worst outcomes of some alternatives are equally good, we would invoke *lexical* maximin. In this case, the second-worst outcome acts as a tie-breaker. If this still does not uniquely determine an alternative, we look at the the third-worst outcome, and so on.

3.3.2. *Maximax*

In contrast to the maximin technique, maximax is a rather optimistic approach, trying to optimise the best case by choosing the alternative that can lead to the best possible outcome.

Definition 5. $\text{Maximax}(A, S) = \max_{a \in A}(\max_{s \in S}(v(a, s)))$

Applying this technique to the decision matrix in Example 1 would result in picking a_3, as it provides the best possible outcome among all alternatives (for scenario s_2).

3.3.3. *Optimism-Pessimism*

Hurwicz developed the Optimism-Pessimism Rule, also known as the *Hurwicz Method* or the *Alpha-Index Rule* [14]. Here, both, the *best* and the *worst* possible outcome of each alternative are considered and the outcomes are weighted according to our degree of optimism, denoted by α. The degree of optimism takes on values between 0 and 1, where 0 stands for a very pessimistic stance and 1 for a very optimistic one. In fact, setting α to 0 is equivalent to using the maximin strategy, while setting it to 1 is equivalent to maximax. The parameter α is assumed to be fixed throughout the evaluation of all alternatives.

Definition 6. $\text{H}_\alpha(A, S) = \max_{a \in A}(\alpha \cdot \max_{s \in S}(v(a, s)) + (1 - \alpha) \cdot \min_{s \in S}(v(a, s)))$

Applying the optimism-pessimism rule to our example with a moderately pessimistic parameter $\alpha = 0.2$ would lead to the following result:

alternative	weighted average	result
a_1	$0.2 \cdot 9 + (1 - 0.2) \cdot 1$	2.6
a_2	$0.2 \cdot 8 + (1 - 0.2) \cdot 3$	4
a_3	$0.2 \cdot 10 + (1 - 0.2) \cdot 5$	6
a_4	$0.2 \cdot 8 + (1 - 0.2) \cdot 6$	6.4

So, a_4 would be the correct choice under the optimism-pessimism rule, as it maximises the weighted average of the most optimistic and most pessimistic result for $\alpha = 0.2$.

3.3.4. *Minmax Regret*

The final strategy, using the minmax regret criterion, defines a regret or loss that we experience by comparing the results of the alternative we picked to the best result that we could have obtained by picking the best alternative for the scenario we encounter.

Definition 7. The (absolute) regret $r(a_i, s_j)$ for an action a_i given a scenario s_j is defined as $r(a_i, s_j) = \max_{a \in A}(v(a, s_j)) - v(a_i, s_j)$.

Minimising the maximal regret is choosing the alternative that has the smallest difference to the best possible outcome when faced with its worst-case scenario. Formally, the minmax regret or MR, is defined as follows.

Definition 8. $\text{MR}(A, S) = \min_{a \in A}(\max_{s \in S}(r(a, s)))$

Applying this to Example 1, yields the following results.

	$r(s_1)$	$r(s_2)$	$r(s_3)$	$r(s_4)$	$r(s_5)$	$\max(r)$
a_1	1	1	2	6	5	6
a_2	3	2	5	3	1	5
a_3	0	0	0	2	0	2
a_4	3	2	1	0	2	3

The alternative with the smallest maximal regret is a_3, so it would be chosen under the minmax regret criterion.

3.4. *Discussion*

As we have seen in the examples in Section 3.3, the four different strategies draw different conclusions as to what is considered the best alternative. This unsatisfactory situation led to research on identifying fundamental principles that a rational decision maker should generally follow when choosing between alternatives and initiated an axiomatic treatment of the matter. Unfortunately, there is no universally accepted set of axioms and there are still controversial discussions about this. Nevertheless, we think that it is a good idea also to analyse the strategies we introduced earlier from a theoretical viewpoint.

3.4.1. *Axiomatisation*

Seminal work on the axiomatisation of decision theory was done by von Neumann and Morgenstern, who proposed four axioms [31]. Later on, Savage, one of the most prominent researchers in decision theory, formulated six axioms [28]. However, since the approach by von Neumann and Morgenstern, and to a lesser extent Savage's work, is mainly based on probabilistic reasoning, we do not focus on them. We are interested in decisions under ignorance, meaning we follow an approach based on qualitative reasoning. Nevertheless, we come back to two of the axioms by von Neumann and Morgenstern that are not probabilistic in nature. Early work in the area of qualitative reasoning was done by Milnor, who proposed ten axioms [21, 22]. Compared to other axiomatisations, we find ten quite a high number for key principles and also think that they are not as intuitive as other frameworks. In fact, in his discussion on decision theory, Suppes mentions that none of the familiar decision theory strategies satisfies all of Milnor's axioms [30]. We decided to go with the four axioms proposed by Arrow and Hurwicz [3].

Before introducing the axioms by Arrow and Hurwicz, we briefly discuss two other axioms that are widely accepted (and were originally proposed by von Neumann and Morgenstern [31]):

- **Completeness:** the (weak) preference relation $\preceq$ defines a total ordering, i.e., for every a_i and a_j, $a_i \preceq a_j$ or $a_j \preceq a_i$, or both. If $a_i \preceq a_j$ and $a_j \preceq a_i$, we say that a decision maker is indifferent (denoted by $a_i \sim a_j$). If $a_i \preceq a_j$ and $a_j \npreceq a_i$, then there is a strict preference for a_j (denoted by $a_i \prec a_j$).
- **Transitivity:** for all a_i, a_j, a_k with $a_i \preceq a_j$ and $a_j \preceq a_k$, it follows that $a_i \preceq a_k$.

These two axioms are justified pragmatically [26]. If a decision maker has contradicting

preferences, e.g., $a_i \prec a_j$ and $a_j \prec a_i$ (or $a_i \prec a_j$, $a_j \prec a_k$, and $a_k \prec a_i$), then they could be trapped in an infinite swapping cycle, as there is always a more preferred option available. [b]

Arrow and Hurwicz propose four (additional) axioms [3]. Given a set of alternatives A, each of the strategies discussed previously determines a set of choices A^*, containing the largest elements of A according to the preference relation $\preceq$ we use. Essentially, Maximin, Maximax, H_α, and MR can be seen as a mapping $A \mapsto A^*$. Arrow and Hurwicz define four axioms that a strategy should satisfy (the following description is taken from [11]):

- **Axiom AH1:** the non-empty intersection of a decision problem, i.e., a set of alternatives, and the set of choices of a larger decision problem, i.e., a larger set of alternatives, is the set of choices of the former. Formally, if $A_1 \subset A_2$ and $A_2^* \cap A_1 \neq \emptyset$, then $A_1^* = A_2^* \cap A_1$.
- **Axiom AH2:** relabelling alternatives and scenarios does not change the outcome, i.e., swapping of rows and/or columns in a decision matrix has no impact on $\preceq$.
- **Axiom AH3:** deletion of a duplicate scenario does not change the outcome (s_i is a duplicate of s_j if $v(a_k, s_i) = v(a_k, s_j)$ for all k.
- **Axiom AH4:** if $a_i \in A^*$ and $a_i \preceq a_j$, then $a_j \in A^*$. If $a_i \notin A^*$ and $a_j \preceq a_i$, then $a_j \notin A^*$.

3.4.2. *Analysing the Different Strategies*

Having introduced the axioms, we now check how the strategies defined in Section 3.3 hold up to them. Completeness, transitivity, and axioms AH2 to AH4 are satisfied by all four, as all of them impose a total order on the set of alternatives and relabelling/deduplication has no effect. For axiom AH1, the picture looks different. Minmax regret does not satisfy this axiom. In the following, we provide a counterexample (taken from [11]).

Example 2. Given a decision problem with the decision matrix shown below with $A_1 = \{a_1, a_2, a_3\}$, A_1^* contains a single alternative: a_1 (the maximal regrets of a_1, a_2, and a_3 are 26, 45, and 27, respectively).

	s_1	s_2	s_3
a_1	49	70	2
a_2	4	96	1
a_3	22	76	25

We now create a set $A_2 = A_1 \cup \{a_4\}$ with a_4 having the values 0, 100, 0 for s_1, s_2, and s_3, respectively. The maximal regrets of a_1, a_2, a_3, and a_4 are now 30, 45, 27, and 49. Consequently, $A_2^* = \{a_3\}$ and $A_2^* \cap A_1 = \{a_3\} \neq A_1^*$, violating axiom AH1. After the introduction of a_4, which is far from being optimal, a_1 has ceased to be the preferred choice. $\Diamond$

[b] Coupled with a tiny fee for swapping, it would eventually force a decision maker into bankruptcy. This is called the *money-pump* argument.

What conclusions can we draw from the axiomatic analysis of the strategies? Minmax regret not satisfying axiom AH1 is actually problematic for us. As we will see later, the query optimisation problem we investigate, the optimal ordering of selection operators, has a huge number of alternatives. For n operators, there are $n!$ potential query execution plans, which makes it unrealistic to look at all of them. Violating axiom AH1 in this context means that depending on which other (suboptimal) alternatives we consider, this may have an effect on which query execution plan we choose in the end, which is not acceptable. Additionally, it turns out that the computation of a solution using minmax regret is prohibitively expensive. In our previous work we have shown that the problem is (at least) NP-hard [1, 2]. We suspect that it is even more complex: we conjecture that it might even be Π_2^P-complete like some other minmax optimisation problems [17]. This conjecture is based on our finding that checking a given solution to an instance of the problem seems to take exponential time. As Ko and Lin have also shown in [17], many of the corresponding c-approximation problems for Π_2^P-complete problems are either themselves Π_2^P-complete or NP-complete. We believe that this makes it unlikely to find good approximation algorithms or heuristics for minmax regret query optimisation. Due to the reasons discussed above, we do not investigate minmax regret further.

The remaining strategies, maximin, maximax, and optimism-pessimism, all satisfy the the six axioms introduced in Section 3.4.1, so from an axiomatic viewpoint they are equivalent. Nevertheless, we focus on the optimism-pessimism rule and do not look further at the maximin and maximax principles as separate alternatives, since the optimism-pessimism rule subsumes both. Setting $\alpha = 0$ yields the same result as maximin, while setting $\alpha = 1$ does so for maximax.

4. Problem Definition and Formalisation

Before defining the problem of ordering selection operators under ignorance, we briefly introduce the original version of the problem.

Definition 9. Given a relation r containing the tuples $t_1, t_2, \ldots, t_m$, a selection operator σ with (Boolean) predicate p selects those tuples t_i from r for which $p(t_i)$ is true:

$$\sigma_p(r) = \{t \in r \mid p(t) = \text{true}\}$$

For $1 \leq i \leq n$, each operator σ_{p_i} has a selectivity s_i and a cost c_i. The selectivity $s_i \in [0, 1]$ determines the ratio of tuples in r that pass the filter:

$$s_i = \frac{|\sigma_{p_i}(r)|}{|r|}$$

$c_i \in \mathbf{R}^+$ represents the cost of σ_{p_i} per tuple, i.e., the cost for processing a single input tuple.

We can apply an arbitrary number of selection operators with different predicates to relation r, e.g. $\sigma_{p_1}(\sigma_{p_2}(\sigma_{p_3}(r)))$ is a valid query. The result includes all tuples t_i in r that satisfy all three predicates, i.e., $p_1(t_i) \wedge p_2(t_i) \wedge p_3(t_i) = \text{true}$. From now on, to declutter

the notation, we do not mention the predicates explicitly anymore but only use their index, i.e., $\sigma_i = \sigma_{p_i}$.

Next, we define the concept of a *query execution plan* containing n selection operators via the permutations of these operators. Let π^n be the set of all possible permutations over $1, 2, \ldots, n$. For $\pi_j \in \pi^n$, $\pi_j(i)$ denotes the i-th element of π_j.

Definition 10. A query execution plan p_j is a permutation $\sigma_{\pi_j(1)}, \sigma_{\pi_j(2)}, \ldots, \sigma_{\pi_j(n)}$ of the n selection operators. The set of all possible query execution plans is given by

$$P = \{p \mid p = \sigma_{\pi(1)}, \sigma_{\pi(2)}, \ldots, \sigma_{\pi(n)} \text{ such that } \pi \in \pi^n\}.$$

The cost of evaluating plan p_j is

$$\begin{aligned} \text{Cost}(p_j) &= \Omega(c_{\pi(1)} + s_{\pi(1)}c_{\pi(2)} + s_{\pi(1)}s_{\pi(2)}c_{\pi(3)} + \cdots + \prod_{i=1}^{n-1} s_{\pi(i)}c_{\pi(n)}) \\ &= \Omega\left(\sum_{i=1}^{n}\left(\prod_{j=1}^{i-1} s_{\pi(j)}\right)c_{\pi(i)}\right) \end{aligned} \tag{1}$$

where Ω is the cardinality of the relation on which we execute the selection operators. Currently, we make the Attribute Value Independence (AVI) assumption that the selection predicates are stochastically independent. Extending our approach to situations in which (some) joint selectivities are known is a topic for future work.

For obtaining the correct answer to the query, it does not matter which of the permutations we run: all of them produce the same result. However, in terms of the execution time, they differ. The job of a query optimiser in a database system is to determine the query execution plan that minimises the costs.

Definition 11. The *selection ordering problem* is about finding the query execution plan $p_{\min} \in P$ having the smallest costs, i.e., $\forall p \in P : \text{Cost}(p_{\min}) \leq \text{Cost}(p)$.

If we know exact values for all the selectivities s_i (and the costs c_i), the selection ordering problems can be solved efficiently, using a ranking scheme devised by Monma and Sidney for sequencing and job scheduling [24]. For each σ_i, we compute a

$$rank(\sigma_i) = \frac{s_i - 1}{c_i}$$

and then apply the operators in increasing order of their ranks. This gives us the plan with minimal costs in $O(n \log n)$ time.

We now move on to the variant of the selection ordering problem where the selectivity of each operator is only partially defined. In particular, the selectivity of each operator is known to fall within an interval of values between 0 and 1. Since there is only a partial order defined on intervals (and not a total order), we cannot simply compute ranks and order the ranks as before. Instead, we try to find an optimal ordering for the operators by applying the optimism-pessimism rule. The exact costs of selection operators can also be unknown, but for the moment we restrict ourselves to partially defined selectivities. Note that since we try to optimise the costs of query execution plans rather than their general

level of quality, we are trying to *minimise* the costs rather than maximising their quality. Compared to Section 3 the roles of max and min will be reversed.

Definition 12. For the selection ordering problem under partial ignorance, given a set $S = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ of selection operators, each has a selectivity s_i defined by a closed interval, for $1 \leq i \leq n$, $s_i = [\underline{s}_i, \overline{s}_i]$ with $\underline{s}_i, \overline{s}_i \in [0, 1]$ and $\underline{s}_i \leq \overline{s}_i$, and a cost c_i.

Depending on their selectivity intervals selection operators may relate to each other in a special way. Later on we exploit this property in order to optimise selection orders.

Definition 13. Given two selection operators $\sigma_i, \sigma_j \in S$, we say that σ_i *dominates* σ_j, iff $\underline{s}_i \leq \underline{s}_j$ and $\overline{s}_i \leq \overline{s}_j$. The set S of operators is called *dominant* if for each pair $\sigma_i, \sigma_j \in S$ it is the case that either σ_i dominates σ_j or σ_j dominates σ_i.

Later on, it will be helpful to consider a special case of dominant sets of operators.

Definition 14. Given two selection operators $\sigma_i, \sigma_j \in S$, we say that σ_i *strictly dominates* σ_j, iff $\overline{s}_i < \underline{s}_j$. A *strictly dominant* set is defined analogously to a dominant set.

If for two selection operators $\sigma_i, \sigma_j \in S$, neither σ_i dominates σ_j nor σ_j dominates σ_i, then σ_i and σ_j form a *nested* pair of operators. So, operator σ_i is *nested* in σ_j if $\underline{s}_j < \underline{s}_i$ and $\overline{s}_i < \overline{s}_j$.

Example 3. Let $S = \{\sigma_1, \sigma_2, \sigma_3\}$ be a set of selection operators, with selectivities $s_1 = [.2, .8]$, $s_2 = [.3, .5]$ and $s_3 = [.1, .4]$. Operator σ_3 dominates both σ_1 and σ_2, but does not strictly dominate either of them. Because σ_2 is nested in σ_1, the set S is not dominant. $\Diamond$

Definition 15. An assignment of a concrete value to each of the n selectivities is called a *scenario* and is defined by a vector $x = (s_1, s_2, \ldots, s_n)$, with $s_i \in [\underline{s}_i, \overline{s}_i]$.

Every time we actually run a query, we encounter one scenario. However, during the optimisation step we are unaware of which scenario we will face. The set of all possible scenarios can be described by $X = \{x \mid x \in [\underline{s}_1, \overline{s}_1] \times [\underline{s}_2, \overline{s}_2] \times \cdots \times [\underline{s}_n, \overline{s}_n]\}$.

As before, a query execution plan p_j is a permutation $\sigma_{\pi_j(1)}, \sigma_{\pi_j(2)}, \ldots, \sigma_{\pi_j(n)}$ of the n selection operators. However, the costs of a plan now depend on the scenario we are encountering.

Definition 16. The cost $\text{Cost}(p_j, x)$ of evaluating plan p_j under a given scenario $x = (s_1, s_2, \ldots, s_n)$ is computed according to Equation (1).

Because we focus on finding an optimal plan according to the optimism-pessimism rule, we do not need to consider all possible scenarios. Instead, only two scenarios are relevant: the scenario in which each operator takes its minimum selectivity value (called the *optimistic scenario*), and the one in which each operator takes its maximum selectivity value (called the *pessimistic scenario*). This is due to the way Arrow and Hurwicz formulate their axioms AH1 to AH4 (see Section 3.4.1 and [3]) and provides another argument for choosing the optimism-pessimism rule [11].

Example 4. Recall the set $S = \{\sigma_1, \sigma_2, \sigma_3\}$ of selection operators from Example 3, with selectivities $s_1 = [.2, .8]$, $s_2 = [.3, .5]$ and $s_3 = [.1, .4]$. The optimistic scenario is given by $x = (\underline{s}_1, \underline{s}_2, \underline{s}_3) = (.2, .3, .1)$, while the pessimistic scenario is given by $y = (\overline{s}_1, \overline{s}_2, \overline{s}_3) = (.8, .5, .4)$. One of the 6 possible plans for S is given by plan $p_1 = \sigma_1\sigma_2\sigma_3$. Assuming that Ω and each cost c_i is set to 1, we can calculate the cost of plan p_1 under scenario x, Cost(p_1, x), using Equation (1) as follows:

$$\text{Cost}(p_1, x) = (1 + .2 + .2 \times .3) = 1.26$$

$\Diamond$

5. Properties

In this section, we prove a number of properties that plans of selection operators satisfy, when optimised according to optimism-pessimism. In particular, we look at cases that can be solved optimally in an efficient way (dominant sets) and those that are hard to solve (nested intervals).

5.1. *Dominance*

We first introduce some notation which will be helpful in subsequent proofs. Consider a plan $p = (\sigma_1, \sigma_2, \ldots, \sigma_n)$. Let $\underline{C}_0(p) = \alpha$ and $\underline{C}_i(p) = \alpha \cdot \prod_{j=1}^{i} \underline{s}_j$, for $1 \leq i \leq n$. Similarly, let $\overline{C}_0(p) = (1 - \alpha)$ and $\overline{C}_i(p) = (1 - \alpha) \cdot \prod_{j=1}^{i} \overline{s}_j$, for $1 \leq i \leq n$. So $\underline{C}_i(p)$ and $\overline{C}_i(p)$ represent the costs of the i'th operator under the optimistic and pessimistic scenarios, respectively, also taking into account the degree of optimism α. We can also write $\underline{C}_i(p) = \underline{C}_{i-1}(p) \cdot \underline{s}_i$ and $\overline{C}_i(p) = \overline{C}_{i-1}(p) \cdot \overline{s}_i$, for $1 \leq i \leq n$. Then the cost of p can be written as

$$cost(p) = \underline{C}_0(p) + \cdots + \underline{C}_{n-1}(p) + \overline{C}_0(p) + \cdots + \overline{C}_{n-1}(p).$$

Now let $R_i(p) = \underline{C}_i(p)/\overline{C}_i(p)$, $0 \leq i \leq n$. Since $R_0(p) = \alpha/(1 - \alpha)$, $R_0(p)$ represents our relative level of optimism at the start of query planning, while $R_i(p)$ represents our relative level of optimism at each position in plan p. The following lemma shows that $R_i(p)$ never increases as i increases.

Lemma 1. *For any plan p, $R_i(p) \leq R_{i-1}(p)$, $1 \leq i \leq n$.*

Proof. Consider a plan p and some $R_i(p)$, $1 \leq i \leq n$. Now

$$\begin{aligned} R_i(p) &= \underline{C}_i(p)/\overline{C}_i(p) = (\underline{C}_{i-1}(p) \cdot \underline{s}_i)/(\overline{C}_i(p) \cdot \overline{s}_i) \\ &= R_{i-1}(p) \cdot \underline{s}_i/\overline{s}_i \end{aligned}$$

Since $\underline{s}_i \leq \overline{s}_i$, we have that $R_i(p) \leq R_{i-1}(p)$. □

In other words, we get relatively more pessimistic as we move from left to right in any plan. We first consider the case where we are given a set of dominant selection operators.

Theorem 1. *Let $S = \{\sigma_1, \ldots, \sigma_n\}$ be a set of selection operators such that σ_i dominates σ_{i+1}, $1 \leq i \leq n-1$ (i.e., S is a set of dominant operators). Then the optimal plan p for S is given by $p = (\sigma_1, \ldots, \sigma_n)$.*

Proof. We show that sequences of dominant operators satisfy the Adjacent Pairwise Interchange (API) property [7], from which the result follows. To prove that the sequences satisfy the API property, we need to show that, for all operators σ_i and σ_j and sequences u and v, if σ_i dominates σ_j, then $cost((u, \sigma_i, \sigma_j, v)) \leq cost((u, \sigma_j, \sigma_i, v))$.

Assume that the sequence u is of length k. Then

$$cost((u, \sigma_i, \sigma_j, v)) = \cdots + \underline{C}_k + \underline{C}_k\underline{s}_i + \underline{C}_k\underline{s}_i\underline{s}_j + \cdots + \overline{C}_k + \overline{C}_k\overline{s}_i + \overline{C}_k\overline{s}_i\overline{s}_j + \cdots$$

while

$$cost((u, \sigma_j, \sigma_i, v)) = \cdots + \underline{C}_k + \underline{C}_k\underline{s}_j + \underline{C}_k\underline{s}_i\underline{s}_j + \cdots + \overline{C}_k + \overline{C}_k\overline{s}_j + \overline{C}_k\overline{s}_i\overline{s}_j + \cdots$$

So

$$cost((u, \sigma_j, \sigma_i, v)) - cost((u, \sigma_i, \sigma_j, v)) = \underline{C}_k(\underline{s}_j - \underline{s}_i) + \overline{C}_k(\overline{s}_j - \overline{s}_i).$$

Since σ_i dominates σ_j, we have that $\underline{s}_i \leq \underline{s}_j$ and $\overline{s}_i \leq \overline{s}_j$. Hence both terms above are non-negative and $cost((u, \sigma_i, \sigma_j, v)) \leq cost((u, \sigma_j, \sigma_i, v))$. □

5.2. *Nested intervals*

The implications of Theorem 1 are very useful: given a set of dominant operators, we know how to order operators with selectivity intervals optimally. The hard case is ordering operators with nested selectivity intervals, which we look at next. We first investigate general properties of selection operators with nested selectivity intervals and then turn to a special case, in which all intervals share the same midpoint.

5.2.1. *General Properties*

Let us write $\sigma_i \sqsubseteq \sigma_j$ to mean that the selectivity interval of operator σ_i is nested inside that of operator σ_j. We say that a set of operators is *fully nested* if $\sqsubseteq$ defines a total order. Let $S = \{\sigma_1, \ldots, \sigma_n\}$ be a fully nested set of operators, with $\sigma_i \sqsubseteq \sigma_{i+1}$, $1 \leq i \leq n-1$. If $\alpha = 0$, then the plan $p = (\sigma_1, \ldots, \sigma_n)$ is optimal. We show below additional situations in which p is optimal. Given α, let $\sigma_i \lhd_\alpha \sigma_j$ denote that $\alpha(\underline{s}_i - \underline{s}_j) \leq (1-\alpha)(\overline{s}_j - \overline{s}_i)$.

Lemma 2. *Let $S = \{\sigma_1, \ldots, \sigma_n\}$ be a fully nested set of operators, with $\sigma_i \sqsubseteq \sigma_{i+1}$, $1 \leq i \leq n-1$. If $\sigma_i \lhd_\alpha \sigma_{i+1}$ for each $1 \leq i \leq n-1$, then the optimal plan for S is $(\sigma_1, \ldots, \sigma_n)$.*

Proof. We first show that the relation $\lhd_\alpha$ is transitive. Let $\sigma_i \lhd_\alpha \sigma_j$ and $\sigma_j \lhd_\alpha \sigma_k$ for some $1 \leq i, j, k \leq n$. From the definition of $\lhd_\alpha$, we have that $\alpha(\underline{s}_i - \underline{s}_j) \leq (1-\alpha)(\overline{s}_j - \overline{s}_i)$ and $\alpha(\underline{s}_j - \underline{s}_k) \leq (1-\alpha)(\overline{s}_k - \overline{s}_j)$. Adding left- and right-hand sides gives us

$$\alpha(\underline{s}_i - \underline{s}_j) + \alpha(\underline{s}_j - \underline{s}_k) \leq (1-\alpha)(\overline{s}_j - \overline{s}_i) + (1-\alpha)(\overline{s}_k - \overline{s}_j)$$

Simplifying yields

$$\alpha(\underline{s}_i - \underline{s}_k) \leq (1-\alpha)(\overline{s}_k - \overline{s}_i)$$

which is the definition of $\sigma_i \lhd_\alpha \sigma_k$.

Clearly, relation $\sqsubseteq$ is also transitive, so if we assume that $\sigma_i \lhd_\alpha \sigma_{i+1}$ for each $1 \leq i \leq n-1$, we have both $\sigma_i \sqsubseteq \sigma_j$ and $\sigma_i \lhd_\alpha \sigma_j$, for each $1 \leq i \leq j \leq n$.

We now show that plans formed from S satisfy the API property, from which the fact that $(\sigma_1, \ldots, \sigma_n)$ is the optimal plan for S follows. To prove this, we need to show that, for all operators σ_i and σ_j and sequences of operators u and v, if $\sigma_i \sqsubseteq \sigma_j$ and $\sigma_i \lhd_\alpha \sigma_j$, then $cost((u, \sigma_i, \sigma_j, v)) \leq cost((u, \sigma_j, \sigma_i, v))$.

Assume that the sequence u is of length k. Then

$$cost((u, \sigma_i, \sigma_j, v)) = \cdots + \underline{C}_k + \underline{C}_k\underline{s}_i + \underline{C}_k\underline{s}_i\underline{s}_j + \cdots + \overline{C}_k + \overline{C}_k\overline{s}_i + \overline{C}_k\overline{s}_i\overline{s}_j + \cdots$$

while

$$cost((u, \sigma_j, \sigma_i, v)) = \cdots + \underline{C}_k + \underline{C}_k\underline{s}_j + \underline{C}_k\underline{s}_i\underline{s}_j + \cdots + \overline{C}_k + \overline{C}_k\overline{s}_j + \overline{C}_k\overline{s}_i\overline{s}_j + \cdots$$

So

$$cost((u, \sigma_j, \sigma_i, v)) - cost((u, \sigma_i, \sigma_j, v)) = \underline{C}_k(\underline{s}_j - \underline{s}_i) + \overline{C}_k(\overline{s}_j - \overline{s}_i).$$

Because $\sigma_i \sqsubseteq \sigma_j$, $(\overline{s}_j - \overline{s}_i) \geq 0$ while $(\underline{s}_j - \underline{s}_i) \leq 0$. So

$$cost((u, \sigma_j, \sigma_i, v)) - cost((u, \sigma_i, \sigma_j, v)) = \overline{C}_k(\overline{s}_j - \overline{s}_i) - \underline{C}_k(\underline{s}_i - \underline{s}_j). \qquad (2)$$

If $k = 0$, the right-hand side of (2) becomes $(1-\alpha)(\overline{s}_j - \overline{s}_i) - \alpha(\underline{s}_i - \underline{s}_j)$. We assumed that $\sigma_i \lhd_\alpha \sigma_j$, so we have $\alpha(\underline{s}_i - \underline{s}_j) \leq (1-\alpha)(\overline{s}_j - \overline{s}_i)$, and the right-hand side of (2) must be non-negative.

For $k > 0$, Lemma 1 shows that the ratio $\underline{C}_k/\overline{C}_k$ does not increase as k increases, and hence the right-hand side of (2) must be non-negative for all k. We conclude that $cost((u, \sigma_i, \sigma_j, v)) \leq cost((u, \sigma_j, \sigma_i, v))$ and the API property holds. □

The following notation will be useful in subsequent results. Consider a plan $p = (\sigma_1, \sigma_2, \ldots, \sigma_n)$. We first introduce notation for the products of selectivities up to (but not including) operator σ_i in the plan. Let $\underline{A}_i(p) = \prod_{k=1}^{i-1} \underline{s}_k$, for $1 < i \leq n$. Similarly, let $\overline{A}_i(p) = \prod_{k=1}^{i-1} \overline{s}_k$, for $1 < i \leq n$. If $i = 1$, then we define $\underline{A}_i(p) = \overline{A}_i(p) = 1$.

Next we introduce notation for costs between two operators σ_i and σ_j, where $i < j$, in plan p. Let $\underline{B}_{ij}(p) = 1 + \underline{s}_{i+1} + \underline{s}_{i+1} \cdot \underline{s}_{i+2} + \cdots + \prod_{k=i+1}^{j-1} \underline{s}_k$, for $1 \leq i < j \leq n$. Similarly, let $\overline{B}_{ij}(p) = 1 + \overline{s}_{i+1} + \overline{s}_{i+1} \cdot \overline{s}_{i+2} + \cdots + \prod_{k=i+1}^{j-1} \overline{s}_k$, for $1 \leq i < j \leq n$. So if σ_i and σ_j are adjacent in p (i.e., $j = i+1$), then $\underline{B}_{ij}(p) = \overline{B}_{ij}(p) = 1$.

The following theorem provides a general condition under which nested operators can be swapped in a plan in order to produce a cheaper plan. (Since the A and B terms in the theorem are equal for both plans p and q, we drop the plan from the notation for readability.)

Theorem 2. *Let p be the plan $(\sigma_1, \ldots, \sigma_i, \ldots, \sigma_j, \ldots, \sigma_n)$, where σ_i and σ_j are nested operators. Let q be the plan in which σ_i and σ_j are swapped. Then $cost(p) \geq cost(q)$ if*

and only if either (i) $\overline{s}_i \geq \overline{s}_j$ *and*

$$\frac{(\overline{s}_i - \overline{s}_j)}{(\underline{s}_j - \underline{s}_i)} \geq \frac{\alpha \cdot \underline{A}_i \cdot \underline{B}_{ij}}{(1-\alpha) \cdot \overline{A}_i \cdot \overline{B}_{ij}} \tag{3}$$

or (ii) $\underline{s}_i \geq \underline{s}_j$ *and*

$$\frac{(\overline{s}_j - \overline{s}_i)}{(\underline{s}_i - \underline{s}_j)} \leq \frac{\alpha \cdot \underline{A}_i \cdot \underline{B}_{ij}}{(1-\alpha) \cdot \overline{A}_i \cdot \overline{B}_{ij}} \tag{4}$$

Proof. The cost formulas for plans p and q are as follows:

$$\begin{aligned} cost(p) = {} & \alpha(1 + X + \underline{s}_1 \cdots \underline{s}_{i-1}\underline{s}_i + \cdots + \underline{s}_1 \cdots \underline{s}_{j-1} + Y) + \\ & (1-\alpha)(1 + U + \overline{s}_1 \cdots \overline{s}_{i-1}\overline{s}_i + \cdots + \overline{s}_1 \cdots \overline{s}_{j-1} + V) \\ cost(q) = {} & \alpha(1 + X + \underline{s}_1 \cdots \underline{s}_{i-1}\underline{s}_j + \cdots + \underline{s}_1 \cdots \underline{s}_{i-1}\underline{s}_j\underline{s}_{i+1} \cdots \underline{s}_{j-1} + Y) + \\ & (1-\alpha)(1 + U + \overline{s}_1 \cdots \overline{s}_{i-1}\overline{s}_i + \cdots + \overline{s}_1 \cdots \overline{s}_{i-1}\overline{s}_j\overline{s}_{i+1} \cdots \overline{s}_{j-1} + V) \end{aligned}$$

where U, V, X and Y are additional terms. Now

$$\begin{aligned} cost(p) - cost(q) = {} & \alpha(\underline{A}_i(\underline{s}_i - \underline{s}_j) + \cdots + \underline{A}_i(\underline{s}_i - \underline{s}_j)\underline{s}_{i+1} \cdots \underline{s}_{j-1}) + \\ & (1-\alpha)(\overline{A}_i(\overline{s}_i - \overline{s}_j) + \cdots + \overline{A}_i(\overline{s}_i - \overline{s}_j)\overline{s}_{i+1} \cdots \overline{s}_{j-1}) \end{aligned}$$

where $\underline{A}_i = \underline{s}_1 \cdots \underline{s}_{i-1}$ and $\overline{A}_i = \overline{s}_1 \cdots \overline{s}_{i-1}$ as defined above. Using the notation $\underline{B}_{ij}$ and $\overline{B}_{ij}$ introduced above, this can be rewritten as

$$\alpha \cdot \underline{A}_i \cdot \underline{B}_{ij} \cdot (\underline{s}_i - \underline{s}_j) + (1-\alpha) \cdot \overline{A}_i \cdot \overline{B}_{ij} \cdot (\overline{s}_i - \overline{s}_j) \tag{5}$$

In case (i) we have $\overline{s}_i \geq \overline{s}_j$; hence $\overline{s}_i - \overline{s}_j$ is non-negative, while $\underline{s}_i - \underline{s}_j$ is non-positive since the operators are nested. So we get

$$(1-\alpha) \cdot \overline{A}_i \cdot \overline{B}_{ij} \cdot (\overline{s}_i - \overline{s}_j) - \alpha \cdot \underline{A}_i \cdot \underline{B}_{ij} \cdot (\underline{s}_j - \underline{s}_i)$$

We want this to be non-negative, so we need

$$(1-\alpha) \cdot \overline{A}_i \cdot \overline{B}_{ij} \cdot (\overline{s}_i - \overline{s}_j) \geq \alpha \cdot \underline{A}_i \cdot \underline{B}_{ij} \cdot (\underline{s}_j - \underline{s}_i)$$

Condition (3) then follows by dividing both sides by $(\underline{s}_j - \underline{s}_i)$ and $(1-\alpha) \cdot \overline{A}_i \cdot \overline{B}_{ij}$, both of which are positive (assuming $\alpha \neq 1$).

In case (ii) we have $\underline{s}_i \geq \underline{s}_j$; hence $\underline{s}_i - \underline{s}_j$ is non-negative, while $\overline{s}_i - \overline{s}_j$ is non-positive since the operators are nested. So formula (5) becomes

$$\alpha \cdot \underline{A}_i \cdot \underline{B}_{ij} \cdot (\underline{s}_i - \underline{s}_j) - (1-\alpha) \cdot \overline{A}_i \cdot \overline{B}_{ij} \cdot (\overline{s}_j - \overline{s}_i)$$

We want this to be non-negative, so we need

$$\alpha \cdot \underline{A}_i \cdot \underline{B}_{ij} \cdot (\underline{s}_i - \underline{s}_j) \geq (1-\alpha) \cdot \overline{A}_i \cdot \overline{B}_{ij} \cdot (\overline{s}_j - \overline{s}_i)$$

Condition (4) then follows by dividing both sides by $(\underline{s}_i - \underline{s}_j)$ and $(1-\alpha) \cdot \overline{A}_i \cdot \overline{B}_{ij}$, both of which are positive (assuming $\alpha \neq 1$). □

In fact, in conditions (3) and (4) the formulas for the ratios between the selectivities give the same result because, in each case, both differences are either positive or negative. So, in a sense, we are only concerned with the absolute values of the differences.

(If we call the ratio on the left of (3) and (4) the *selectivity ratio* and the one on the right the *optimism ratio*, then the theorem says the following: if the selectivity ratio is greater than the optimism ratio, then the operator with the smaller maximum selectivity should appear earlier, while if the selectivity ratio is less than the optimism ratio, then the operator with the smaller minimum selectivity should appear earlier.)

The problem with the above condition is that it depends on the selectivities of operators other than σ_i and σ_j in the plan. The following corollary of the previous theorem gives a sufficient condition for swapping operators σ_i and σ_j, dependent only on α and the selectivities of σ_i and σ_j.

Corollary 1. *Given plans $p = (\sigma_1, \ldots, \sigma_i, \ldots, \sigma_j, \ldots, \sigma_n)$ and q in which σ_i and σ_j are swapped, $cost(q) \leq cost(p)$ if $\overline{s}_i \geq \overline{s}_j$ and*

$$\frac{(\overline{s}_i - \overline{s}_j)}{(\underline{s}_j - \underline{s}_i)} \geq \frac{\alpha}{1-\alpha} \tag{6}$$

Proof. In Theorem 2 we established that $cost(q) \leq cost(p)$ if $\overline{s}_i \geq \overline{s}_j$ and

$$\frac{(\overline{s}_i - \overline{s}_j)}{(\underline{s}_j - \underline{s}_i)} \geq \frac{\alpha \cdot \underline{A}_i \cdot \underline{B}_{ij}}{(1-\alpha) \cdot \overline{A}_i \cdot \overline{B}_{ij}}$$

Since it is always the case that $\underline{A}_i \leq \overline{A}_i$ and $\underline{B}_{ij} \leq \overline{B}_{ij}$, we have that

$$\frac{\alpha}{1-\alpha} \geq \frac{\alpha \cdot \underline{A}_i \cdot \underline{B}_{ij}}{(1-\alpha) \cdot \overline{A}_i \cdot \overline{B}_{ij}}$$

and the result follows. □

It is instructive to consider a special case in which Corollary 1 can be applied, namely when all the operators are nested and all have the same midpoint for their selectivity intervals.

Corollary 2. *Let S be a set of operators whose intervals each have the same midpoint. If $\alpha < 0.5$, then the optimal plan for S has operators ordered in terms of non-decreasing maximum selectivity.*

Proof. If operators s_i and s_j have the same midpoint, then

$$\frac{(\overline{s}_i - \overline{s}_j)}{(\underline{s}_j - \underline{s}_i)} = 1.$$

If $\alpha < 0.5$, then condition (6) always holds. This means that operators with smaller maximum selectivity should appear earlier in an optimal plan, so the optimal plan is one in which the operators are ordered by non-decreasing maximum selectivity. □

In the following subsection, we study nested intervals with the same midpoint further, finding some conditions under which the API property holds.

5.2.2. *Nested intervals with the same midpoint*

Let us assume that we are given a set S of operators with associated interval selectivities such that the *midpoint* of each selectivity is the same. Given a plan p, recall the definition of the ratio $R_i(p)$, for position i in p, given near the beginning of Section 5.1. The following lemma shows that, once the ratio $R_i(p)$ drops below one in plan p, the remaining operators in p should be in order of non-decreasing maximum selectivity.

Lemma 3. *Let S be a set of m operators whose intervals each have the same midpoint. Let p be a sequence of $0 < n < m$ operators such that $R_n(p) < 1$. Then the minimum cost plan for S which has p as a prefix has the operators in positions $n+1$ to m ordered in terms of non-decreasing maximum selectivity.*

Proof. We show that sequences of operators with p as a prefix satisfy the API property, from which the result follows. To prove that the sequences satisfy the API property, we need to show that, for all operators σ_i and σ_j and sequences u and v (where u has p as a prefix), if $\overline{s}_i < \overline{s}_j$, then $cost((u, \sigma_i, \sigma_j, v)) < cost((u, \sigma_j, \sigma_i, v))$.

Assume that the sequence u is of length $k \geq n$. Then

$$cost((u, \sigma_i, \sigma_j, v)) = \cdots + \underline{C}_k + \underline{C}_k \cdot \underline{s}_i + \underline{C}_k \cdot \underline{s}_i \cdot \underline{s}_j + \cdots + \overline{C}_k + \overline{C}_k \cdot \overline{s}_i + \overline{C}_k \cdot \overline{s}_i \cdot \overline{s}_j + \cdots$$

while

$$cost((u, \sigma_j, \sigma_i, v)) = \cdots + \underline{C}_k + \underline{C}_k \cdot \underline{s}_j + \underline{C}_k \cdot \underline{s}_i \cdot \underline{s}_j + \cdots + \overline{C}_k + \overline{C}_k \cdot \overline{s}_j + \overline{C}_k \cdot \overline{s}_i \cdot \overline{s}_j + \cdots$$

So

$$cost((u, \sigma_j, \sigma_i, v)) - cost((u, \sigma_i, \sigma_j, v)) = \underline{C}_k(\underline{s}_j - \underline{s}_i) + \overline{C}_k(\overline{s}_j - \overline{s}_i).$$

Since $\overline{s}_j > \overline{s}_i$, $(\overline{s}_j - \overline{s}_i)$ is positive while $(\underline{s}_j - \underline{s}_i)$ is negative (because of the assumption that intervals are nested). So

$$cost((u, \sigma_j, \sigma_i, v)) - cost((u, \sigma_i, \sigma_j, v)) = \overline{C}_k(\overline{s}_j - \overline{s}_i) - \underline{C}_k(\underline{s}_i - \underline{s}_j). \quad (7)$$

Because the intervals have equal midpoints, we know that $(\overline{s}_j - \overline{s}_i) = (\underline{s}_i - \underline{s}_j)$. Also, $\overline{C}_k > \underline{C}_k$ because $R_n(p) < 1$ and Lemma 1 ensures the ratio continues to be less than one for position k. Hence, the difference between the costs given in (7) is positive. □

Note that Corollary 2 above also follows from Lemma 3 by observing that if $\alpha < 0.5$, then $R_0(p) = \alpha/(1-\alpha) < 1$ for any plan p.

For a plan p comprising n operators, the notation $R_n(p)$ denotes the ratio between the optimistic and pessimistic costs of the last term in p. Given another plan q over the same operators and with the same degree of optimism α, it should be clear that $R_n(q) = R_n(p)$. Hence, for a *set* S of n operators, we introduce the notation $R_n(S)$ to represent the ratio for the last term of any plan using all the operators in S (but we only allow this notation where n is the cardinality of S).

Lemma 4. *Let S be a set of $n \geq 2$ operators, each with the same midpoint, such that $R_n(S) > 1$. Then the minimum cost plan for S has the operators ordered in terms of non-decreasing minimum selectivity.*

Proof. The proof is similar to that of Lemma 3 in that we show that sequences of all n operators from S satisfy the API property (but this time considering minimum selectivities). So we need to prove that for all operators σ_i and σ_j and sequences u and v, such that u, v, σ_i and σ_j comprise all of the operators in S, if $\underline{s}_i < \underline{s}_j$, then $cost((u, \sigma_i, \sigma_j, v)) < cost((u, \sigma_j, \sigma_i, v))$.

The difference between the costs is the same as in Lemma 3, namely

$$cost((u, \sigma_j, \sigma_i, v)) - cost((u, \sigma_i, \sigma_j, v)) = \underline{C}_k(\underline{s}_j - \underline{s}_i) + \overline{C}_k(\overline{s}_j - \overline{s}_i).$$

However, now $\underline{s}_j > \underline{s}_i$, so $(\underline{s}_j - \underline{s}_i)$ is positive while $(\overline{s}_j - \overline{s}_i)$ is negative (because of the assumption of nested intervals). So

$$cost((u, \sigma_j, \sigma_i, v)) - cost((u, \sigma_i, \sigma_j, v)) = \underline{C}_k(\underline{s}_j - \underline{s}_i) - \overline{C}_k(\overline{s}_i - \overline{s}_j). \tag{8}$$

Let us denote $(u, \sigma_j, \sigma_i, v)$ by p and $(u, \sigma_i, \sigma_j, v)$ by q. The costs of the $k + m$'th terms in $cost(p)$ and $cost(q)$ are equal for $2 \leq m \leq n - k$ (so including the term for σ_j in p and that for σ_i in q). Also, by assumption, $R_n(p) = R_n(q) > 1$, and by Lemma 1, $R_i(p) > 1$ and $R_i(q) > 1$, $0 \leq i \leq n$. Hence, $R_{k+2}(p) = R_{k+2}(q) > 1$ and $R_k(p) = R_k(q) > 1$, and so $\underline{C}_k > \overline{C}_k$. Since $(\underline{s}_j - \underline{s}_i) = (\overline{s}_i - \overline{s}_j)$, the difference between the costs given in (8) is positive. □

Unfortunately, the API property does not always hold, even for fully nested intervals all with the same midpoint. As α increases from 0.5, operators with wider intervals (larger maximum selectivity, but smaller minimum selectivity) appear at the beginning of the optimal plans. Through experimentation we have observed that each such optimal plan appears to comprise two parts: in the first part, operators are ordered by non-decreasing minimum selectivity, and in the second part by non-decreasing maximum selectivity. In other words, we start off being optimistic but at some point switch to being pessimistic. However, discovering which operators should appear in the "first part" of an optimal plan seems to be a difficult problem.

5.3. *Maximum error*

Let p^M (respectively, p^m) be the plan in which the selection operators are ordered by increasing maximum (respectively, minimum) selectivity. For plan p, let $C_0(p)$ (respectively, $C_1(p)$) denote the cost of p when α has the value 0 (respectively, 1). So, p^M is the plan which minimises C_0, while p^m is the plan which minimises C_1.

The equation for the cost of a plan p is:

$$cost(p) = C_0(p) - \alpha(C_0(p) - C_1(p))$$

As shown in Figure 1, the equations for $cost(p^M)$ and $cost(p^m)$ must intersect. The value of α for the intersection point, denoted by α^*, is given by:

$$\alpha^* = \frac{C_0(p^M) - C_0(p^m)}{(C_0(p^M) - C_1(p^M)) - (C_0(p^m) - C_1(p^m))}$$

Now if we choose plan p^M when $\alpha \leq \alpha^*$ and plan p^m when $\alpha > \alpha^*$, then the maximum error in the cost is given by:

$$\alpha^*(C_1(p^M) - C_1(p^m))$$

Letting $D_0 = (C_0(p^M) - C_0(p^m))$ and $D_1 = (C_1(p^M) - C_1(p^m))$, substituting for α^* and reordering terms, we get:

$$\frac{D_0 \cdot D_1}{D_0 - D_1}$$

Example 5. Consider the selection operators σ_1, σ_2, and σ_3 with interval selectivities $[0.3, 0.7]$, $[0.2, 0.8]$ and $[0.1, 0.9]$, respectively. Note that the operators are nested in this case. Figure 1 shows the cost of the plans $p^M = (\sigma_1, \sigma_2, \sigma_3)$ and $p^m = (\sigma_3, \sigma_2, \sigma_1)$ as α changes. In this example, the costs of the two plans are equal when $\alpha = 0.6$ and the maximum error is 0.144. ◊

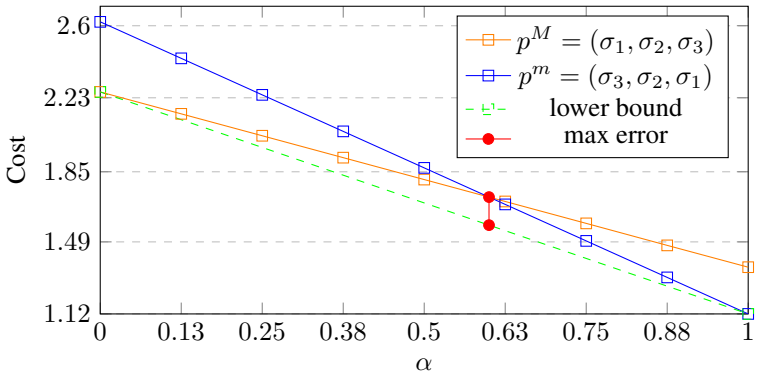


Fig. 1: Maximal error in cost between plans p^M and p^m

While this technique allows us to bound the error we are making, we cannot achieve results better than the best plan for the completely optimistic and pessimistic scenario. As we will see with the introduction of our heuristic in the following section and its experimental evaluation in Section 7, it is possible to find plans that are better than both.

6. Algorithms

In this section, we describe five algorithms for finding an order (plan) in which to execute a given number of selection operators, based on their interval selectivities and an optimism/pessimism ratio α. The five algorithms are as follows:

(1) *Optimal*: this is a brute-force algorithm that, given α, simply calculates the cost for each permutation of the operators and returns the plan of minimum cost. As this algorithm has exponential runtime, it is not practical for large problem instances. We use it in our experiments on small instances to determine the quality of the other algorithms.

Algorithm 1.

Input: α and set of operators $S = \{\sigma_1, \ldots, \sigma_n\}$
Output: a plan p
$p \leftarrow [\,]$
while p contains fewer than n operators **do** {
  $c \leftarrow \infty$
  for all σ_i not in p **do** {
    $q \leftarrow S \setminus (\{\sigma_i\} \cup p)$
    order q by non-decreasing maximum selectivity
    if cost$(p \circ \sigma_i \circ q) < c$ **then** {
      minop $\leftarrow \sigma_i$
      $c \leftarrow$ cost$(p \circ \sigma_i \circ q)$
    }
  }
  append minop to plan p
}
return p

Fig. 2: Heuristic plan finding algorithm

(2) *Pessimistic*: this algorithm simply orders the operators by non-decreasing values of their maximum selectivities. It is therefore independent of α and is optimal when $\alpha = 0$.
(3) *Optimistic*: this algorithm simply orders the operators by non-decreasing values of their minimum selectivities. It is therefore independent of α and is optimal when $\alpha = 1$.
(4) *Midpoint*: this algorithm calculates the midpoint of the selectivity interval for each operator and then orders the operators by non-decreasing values of their midpoint selectivities. It is therefore also independent of α.
(5) *Heuristic*: our heuristic algorithm for finding a plan of minimum cost is shown as Algorithm 1 in Figure 2 and is described below.

The algorithm starts by trying each operator in turn in the first position of the plan, with the remaining operators ordered by non-decreasing maximum selectivity values (i.e., pessimistically). In each case it calculates the cost of the resulting plan, and chooses for the first position the operator whose plan has the smallest cost. It then considers each subsequent position k ($2 \leq k \leq n$) in the plan, and repeats this process (each time excluding the operators already in the first $k-1$ positions). So the algorithm is based on the observations made at the end of Section 5.2.1.

A naive implementation of Algorithm 1 runs in time $O(n^3)$, using $O(n^2)$ swaps of operators and $O(n)$ time to calculate the cost of each plan. However, we can reduce the time from $O(n^3)$ to $O(n^2)$ by storing partial plan costs at the start, and then updating them incrementally and using them to compute the cost of each plan in constant time.

Example 6. Consider the selection operators σ_1, σ_2, σ_3 and σ_4, with interval selectivities $[0.4, 0.6]$, $[0.3, 0.7]$, $[0.2, 0.8]$ and $[0.1, 0.9]$, respectively. Our algorithm finds the optimal plan for this setting for $\alpha = 0.5$. Indeed, when choosing which operator to place in the first position, it considers the four plans $(\sigma_1, \sigma_2, \sigma_3, \sigma_4,)$, $(\sigma_2, \sigma_1, \sigma_3, \sigma_4)$, $(\sigma_3, \sigma_1, \sigma_2, \sigma_4)$, and $(\sigma_4, \sigma_1, \sigma_2, \sigma_3)$, where the third is optimal. $\Diamond$

7. Experimental Evaluation

In this section we provide the results of an empirical study comparing the performance of the different algorithms mentioned in Section 6, i.e., optimal, pessimistic, optimistic, midpoint, and our heuristic.

7.1. *Set-up*

All algorithms were implemented in Python and run on a laptop with 8 GB of RAM and four Intel Core i7-6500U processors running at 2.5 GHz. We used two different datasets for the experiments.

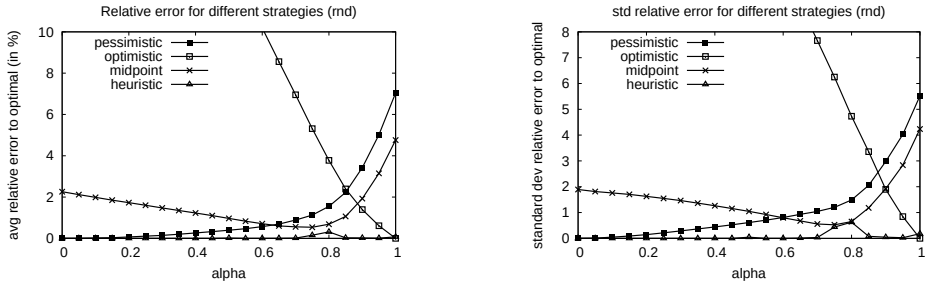
For the first dataset we generated twenty sets of nine selection operators for which the lower and upper bounds of the selectivities were uniformly distributed in $[0, 1]$. We call this dataset 'rnd'. As most operators within each set dominate each other, it is fairly easy to find good solutions: according to Theorem 1 we only have to sort them in order of dominance.

For that reason, we generated a second dataset consisting of twenty sets of nine selection operators. However, for each set the intervals of all the operators were nested. The values were uniformly distributed in $[0, 1]$ and then assigned to the operators in the following way: the first operator was assigned the smallest and largest value, the second operator the second-smallest and second-largest value, and so on. We call this dataset 'nested'. In both datasets, 'rnd' and 'nested', we restricted ourselves to nine selection operators to be able to run the optimal algorithm as well.

7.2. *Results*

First, we look at the result for running the algorithms on the dataset 'rnd'. Figure 3(a) shows the results for the different algorithms compared to the optimal algorithm by depicting the relative error. Figure 3(b) looks at the variation of this relative error, showing the standard deviation of the relative errors.

As expected, the pessimistic algorithm performs worse the higher the value of α and only achieves an optimal result for $\alpha = 0$. For the optimistic algorithm it is the other way around: it only achieves an optimal result for $\alpha = 1$. Overall, the pessimistic strategy seems to be the better one of the two. Interestingly, the midpoint strategy is able to obtain good results for midrange values of α and does worse for the extreme ends. However, our heuristic is able to stay very close to the optimal result and outperforms all the other algorithms (except for the optimal one), with an average relative error over the 20 datasets of at most 0.2%. In fact, the worst relative error of our heuristic we have seen over the 20 datasets and all the different values of α is 2.1%. Only for α-values between 0.7 and 0.8



(a) relative error (b) standard deviation

Fig. 3: Comparison with optimal algorithm (dataset 'rnd')

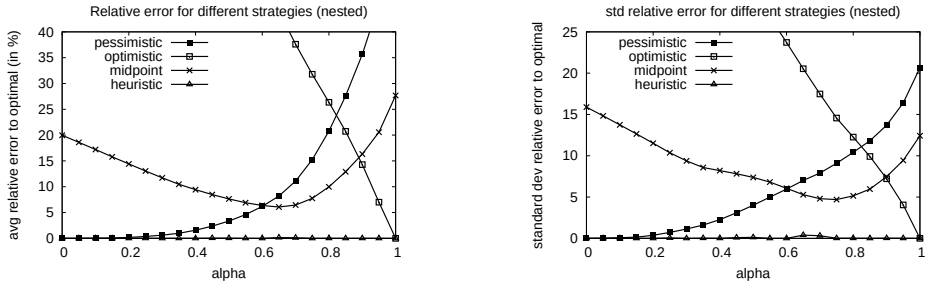
does the midpoint strategy come close, although its average relative error can be as high as 4.8% and its maximum relative error as high as 22.9%. Our heuristic also exhibits the smallest standard deviation for the relative errors, meaning that it outputs good results in a reliable way. Again, only the midpoint strategy comes close for values of α between 0.7 and 0.8.

Next, we look at the result of running the algorithms on the dataset 'nested'. Again, we show the average relative error (in Figure 4(a)) and the standard deviation (in Figure 4(b)). The general shapes of the curves in Figure 4 are very similar to those found in Figure 3, meaning that α affects the behaviour of the algorithms in a similar way. However, all algorithms, with the notable exception of our heuristic, perform much worse. In this case, the worst relative error (rather than the average as shown in Figure 4(a)) of our heuristic we observed was 1.3%, while that for the midpoint strategy, e.g., was 66.6%. Overall, this dataset illustrates that a set of non-dominant selection operators is much harder to optimise. Nevertheless, our heuristic is able to cope well with this kind of situation.

8. Conclusion and Outlook

Even though query optimisation has been studied for decades now, generating efficient plans for queries is still a challenging problem. One major stumbling block is the estimation of cardinalities. In many cases, it is very difficult to approximate the cardinalities of intermediate results accurately and the estimation errors lead a query optimiser to pick a suboptimal plan. Consequently, researchers have started to look into the notion of robustness of query execution plans: Lohman emphasises that it is more important to build robust query plans than to strive for an unattainable optimal one [19].

In our work we apply concepts of decision theory, which is the study of the reasoning underlying an entity's choices. In particular, we are interested in the branch that considers uncertainty and missing information and apply typical approaches from this area to query optimisation in order to build more robust plans. We have shown that the optimism-



(a) relative error (b) standard deviation

Fig. 4: Comparison with optimal algorithm (dataset 'nested')

pessimism rule is well-suited to this task and have developed a heuristic that generates query execution plans very close to the optimal ones under this rule.

For future work we would like to investigate the complexity of the problem further. At the moment we are neither able to find an exact polynomial-time algorithm for finding optimal plans under the optimism-pessimism rule, nor to prove the NP-hardness of the problem. Identifying the complexity class could either lead us to an efficient optimal algorithm or give us some indication of which approximation strategies to pursue. Another potential avenue of research is to extend our work to other operators, such as joins.

References

[1] K. H. Alyoubi, S. Helmer, and P. T. Wood. Ordering selection operators under partial ignorance. In *CIKM*, pages 1521–1530, 2015.

[2] K. H. Alyoubi, S. Helmer, and P. T. Wood. Ordering selection operators using the minmax regret rule. CoRR abs 1507.08257, arXiv.org, 2015.

[3] K. J. Arrow and L. Hurwicz. An optimality criterion for decision-making under ignorance. *Uncertainty and expectations in economics*, 1, 1972.

[4] R. Avnur and J. M. Hellerstein. Eddies: continuously adaptive query processing. In *SIGMOD*, pages 261–272, 2000.

[5] B. Babcock and S. Chaudhuri. Towards a robust query optimizer: a principled and practical approach. In *SIGMOD*, pages 119–130, 2005.

[6] S. Babu, P. Bizarro, and D. DeWitt. Proactive re-optimization. In *SIGMOD*, pages 107–118, 2005.

[7] K. A. Baker and D. Trietsch. *Principles of Sequences and Scheduling*. Wiley, 2009.

[8] F. Chu et al. Least expected cost query optimization: an exercise in utility. In *PODS*, pages 138–147, 1999.

[9] F. Chu et al. Least expected cost query optimization: what can we expect? In *PODS*, pages 293–302, 2002.

[10] H. D, P. N. Darera, and J. R. Haritsa. On the production of anorexic plan diagrams. In *VLDB*, pages 1081–1092, 2007.

[11] T. Denœux. Decision-making with belief functions: A review. *International Journal of Approx-*

imate Reasoning, 109:87–110, 2019.

[12] S. Ganguly. Design and analysis of parametric query optimization algorithms. In *VLDB*, pages 228–238, 1998.

[13] M. Garofalakis and P. B. Gibbons. Wavelet synopses with error guarantees. In *SIGMOD*, pages 476–487, 2002.

[14] L. Hurwicz. Some specification problems and application to econometric models. *Econometrica*, 19:343–344, 1951.

[15] Y. Ioannidis. The history of histograms (abridged). In *VLDB*, pages 19–30, 2003.

[16] N. Kabra and D. J. DeWitt. Efficient mid-query re-optimization of sub-optimal query execution plans. In *SIGMOD*, pages 106–117, 1998.

[17] K.-I. Ko and C.-L. Lin. On the complexity of min-max optimization problems and their approximation. In D.-Z. Du and P. M. Pardalos, editors, *Minimax and Applications*, pages 219–239. Springer US, Boston, MA, 1995.

[18] G. Lohman. Is query optimization a "solved" problem? `http://wp.sigmod.org/?p=1075`, 2014.

[19] G. Lohman. Query optimization – are we there yet? In *BTW*, pages 25–26, 2017.

[20] V. Markl et al. Robust query processing through progressive optimization. In *SIGMOD*, pages 659–670, 2004.

[21] J. W. Milnor. Games against nature. RAND Research Memorandum RM-679, RAND Corporation, 1951.

[22] J. W. Milnor. Games against nature. In R. M. Thrall and C. H. Coombs, editors, *Decision Processes*, pages 49–59. Wiley, New York, 1954.

[23] G. Moerkotte et al. Exploiting ordered dictionaries to efficiently construct histograms with q-error guarantees in SAP HANA. In *SIGMOD*, pages 361–372, 2014.

[24] C. L. Monma and J. B. Sidney. Sequencing with series-parallel precedence constraints. *Mathematics of Operations Research*, 4(3):215–224, 1979.

[25] T. Neumann and C. A. Galindo-Legaria. Taking the edge off cardinality estimation errors using incremental execution. In *BTW*, pages 73–92, 2013.

[26] M. Peterson. *An Introduction to Decision Theory*. Cambridge University Press, 2009.

[27] N. Polyzotis and M. Garofalakis. Statistical synopses for graph-structured XML databases. In *SIGMOD*, pages 358–369, 2002.

[28] L. Savage. *The Foundations of Statistics*. Wiley, 1954.

[29] U. Srivastava et al. Operator placement for in-network stream query processing. In *PODS*, pages 250–258, 2005.

[30] P. Suppes. The philosophical relevance of decision theory. *The Journal of Philosophy*, 58(21):605–614, 1961.

[31] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[32] F. Wolf, M. Brendle, N. May, P. R. Willems, K.-U. Sattler, and M. Grossniklaus. Robustness metrics for relational query execution plans. *Proc. VLDB Endow.*, 11(11):1360–1372, July 2018.

[33] V. Zadorozhny et al. Efficient evaluation of queries in a mediator for websources. In *SIGMOD*, pages 85–96, 2002.

[34] N. Zhang et al. Statistical learning techniques for costing XML queries. In *VLDB*, pages 289–300, 2005.