Principles of probabilistic query optimization
Waas, F.

Citation for published version (APA):
Conclusion

The problem of query optimization differs from typical combinatorial optimization problems in two important aspects.

Firstly, a problem instance is strictly speaking always a problem instance with respect to a certain system configuration. In contrast to problems like the Traveling Salesman Problem, Knapsack etc. we lack a universal portable problem specification and, more severe, a universal cost model. Hardly any complex query will lead to identical query plans when optimized on two different database systems, yet, for each system they may very well be the best plans. These differences result from a different design and implementation techniques used. From one database system to another basically all components differ in more or less significant ways. For instance, the set of operators usually come in a large variety where each of them reflects some technicalities that are specific to the particular database system. Moreover the sets of operator differ widely; for example constructs like hash teams as an implementation technique for a group of subsequent hash joins is a singular specialty as is bitmap filtering in connection with hash joins \([GBC98, CHY93]\). Thus given a query there is not just one optimal plan but an optimal plan with respect to the cost model used.

Secondly, even within the framework of one database system the question of the optimal solution cannot be answered definitely if the query is of large or very large size. The estimate errors increasingly dominate the cost computation and the costs computed serve only as an approximation of the actual execution costs. Consequently, an optimization beyond the resolution of the cost model, i.e. the capability to distinguish two solutions conclusively in their costs, is not useful. As opposed to this situation, the Traveling Salesman Problem and other classical combinatorial optimization problems have an exact cost formula and are independent of any other background component, thus, these problems are exactly reproducible—the optimal tour of a Traveling Salesman Problem can be determined unambiguously unless there are several tours of the same optimal length.

Both these facts render the query optimization problem subjective and
approximative rather than a problem that could be solved to optimality in isolation. However, the problem also displays general trends that put bounds to the uncertainty, that allow us to postulate a series of basic properties.

In this work we sat out to analyze the problem's underlying structure and addressed specifically the issue of randomized or probabilistic optimization of large queries. In the following, we summarize the results achieved and discuss remaining open problems afterward.

9.1 Summary

Our analysis revolved around the concept of cost distributions and their effects on optimization techniques. Cost distributions determine the frequency of cost values in the complete search space identifying characteristic concentrations of cost values.

First of all, we provided the necessary means to obtain and verify such distributions for three differently sophisticated versions of the problem: cross product optimization, join order optimization with non-isomorphic processing trees, and finally, the unrestricted general case of query optimization. While the first two were developed on simplified models, the latter was devised and implemented in Microsoft SQL Server.

Equipped with this toolkit we extracted and analyzed cost distributions for the different models pointing out their close relationships and similarities. The distributions displayed the same trends—within certain ranges of variation—and are distinguished by their high stability. We contrasted the distributions found with distributions of other NP-hard combinatorial optimization problems including Traveling Salesman, Partitioning and Knapsack Problem. This comparison not only lend strong support to the idea that cost distributions are highly characteristic for a problem but also implied classifications of basic types of cost distributions.

Before discussing the effects of cost distributions on randomized optimization algorithms, we addressed the question of the problem's difficulty. Recent developments in the context of NP-complete decision problems suggest concentrations of difficult cases in a small range of a so-called order parameter. This phenomenon of phase transitions gained enormous popularity in the last decade. However, as our experiments with the Asymmetric Traveling Salesman Problem showed, there is no phase transition of similarly distinct kind in optimization though areas of higher and lower difficulty are clearly to spot. The changes of difficulty—except for trivial cases—are however strongly depending on the algorithm used. These findings put attempts to proof the existence of a phase transition on the lines of the $k$-Satisfiability problem like undertaken in [KRHM95] in a different light. We concluded our assessment of difficulty with the introduction of probabilistic difficulty, a measure of difficulty that takes a problem's cost distribution into account.
After this, we turned our attention to probabilistic query optimization techniques and evolutionary computing. Firstly, we analyzed evolutionary algorithm not only against the background of query optimization but used the previously introduced classification of cost distributions to give a comprehensive assessment complemented with a case study to verify our results in practice. Our findings explain where these techniques fall short of what is to expect and where they turn out to be well-suited instruments for optimization. Secondly, we scrutinized randomized algorithms like Simulated Annealing, Simple Improvement, Iterative Improvement, Hill Climbing, random sampling, and Two-Phase Optimization. As opposed to earlier studies, we identified the single principles a particular technique is composed of and studied the building blocks in isolation before assessing the compound method. That way, we were able to explain various effects observed previously, which was not fully understood in related work.

Piecing the parts together we finally presented an algorithm performing biased sampling using bottom-up random generation of plans with cost-bound pruning. These findings summarize our analysis best as “Good enough is easy”.

9.2 Open Problems

Finally, some thoughts where to go from here. Each of the chapters suggests one or more directions of further research either concerning the practicability of the ideas presented or the transfer to other areas of combinatorial optimization.

We detailed the basic properties of cost distributions found in query optimization. Further differentiation of the extent of the skew—i.e., the concentration around the optimum—could be helpful to determine the difficulty of a query compared to others. Such an assessment could be used to decide on what optimization strategy to use or how to combine several different ones, and how much effort to put into the optimization. It would be particularly challenging to devise means to predict the shape given the declarative query and the usual database statistics only. First steps in this direction could include the investigation of incrementally insertion of additional single joins and analyzing their impact on the distribution’s shape; a principle that would easily extend to complete sub-queries. In related work, cost models that take parallelism and main-memory resident base tables into account have been used to promote more sophisticated optimization techniques. While significantly more difficult to optimize with heuristics, these extended models do not appear to differ much from simpler models when addressed with blind search algorithms which raises hopes to unify some of these models.

The neighborhoods defined by transformation rules have a strong influence on the performance. But so far only little research has been devoted to investigate the implied topologies with the aim to generate landscapes
that are favorable to certain algorithms. First results in this field have been reported on by Spiliopoulou [Spi92]. Also the modification of the rule sets at run time in a way that it is able to adopt to the different areas of the search space could be an interesting target for further research.

Finally, the results of Chapter 8 offer the possibility for a combination of randomized and local exhaustive search. Using a framework like the MEMO structure but applying a very restricted set of transformations that fathom only the immediate vicinity of the initial plan together with a sampling phase that provides say 10 potential initial plans, the strengths of both approaches could be combined.