Principles of probabilistic query optimization
Waas, F.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
The term *cost distribution* refers to the distribution function of the occurring values of the objective function in the entire search space of a combinatorial optimization problem. This distribution in turn is defined by its *density*, i.e. the frequency of the single values; we will use both terms synonymously, context permitting.

Let us first motivate why cost distributions are an important characteristic of search spaces, and in particular preferable to a characterization by other means. Optimization problems are in general abstracted as finding the minimum, or maximum respectively, of an objective function over a multi-dimensional space. Notions such as "local-minimum", "landscape" or "hill-climbing" implicitly refer to the idea of "neighborhood" among solutions, i.e., a topology of the search space. However, such a topology is not intrinsic to the problem, but defined—intently or not—by the encoding of solutions, or by transformation mechanisms used in search algorithms, e.g. Simulated Annealing. In contrast, the cost distribution in the search space is natural to the problem and has no arbitrary component.

Our principal goal within this section is to identify characteristic features of cost distributions to obtain deeper insight into the mode of action of different optimization techniques. Features of interest include total range of costs, mean and deviation of cost values, concentrations of solutions within the search space and the parameters responsible for the particular effect.

So far, cost distributions received little attention within the database community when it comes to analyze combinatorial optimization problems. Instead, topological structures have been devised and used, mainly because they offer a very intuitive connection to a number of optimization algorithms.

The concept is apparently not restricted to query optimization or a certain class of optimization problems, but extends generally to any combinatorial optimization problem. And, if the principle is generally applicable we can also expect the conclusions drawn from it to hold with similar generality. To this end, we will scrutinize cost distributions of several well-known
and well-understood optimization problems. Indeed, they are classics in this field: Number Partitioning, the Traveling Salesman Problem, and the Knapsack Problem. Modeling the occurring cost distributions with analytical means will also give us a good impression of the limitations of such techniques indicating what assumptions are justified and which need further attention.

An analysis based on cost distributions is specifically desirable as it offers the possibilities to predict the success of optimization strategies without having to analyze the problem and the algorithm at a level of detail that is hard to cope with. Rather, it provides a black box approach for this assessment.

Before we study the abovementioned optimization problems in detail, we need to clarify what to expect from such an analysis and what not. The prospect of such a powerful and general tool for the assessment of optimization algorithms needs to be carefully contrasted with its limitations. Let us first have a look at the means we are going to deploy.

1. We model average cases by using random variables that are often normally distributed and discuss in what way other instances may differ.

2. We approximate distributions by normal distributions where appropriate, substituting originally discrete distributions with continuous ones.

The results are approximations for an increasing size of the problem instances with respect to the deviation of the underlying random variables. Moreover, it characterizes the distribution of a problem on average. As we will see, the mathematical model is usually very accurate for large problem sizes and reflects important trends, as far as the classical optimization problems are concerned. For the cross product optimization and join ordering problem we will not be able to provide as accurate a model as for the traditional problems.

In any case, these models do not map particular instances to distribution functions directly. For some of the problems it is not too difficult to come up with special instances that fall beyond our analysis leading to unpredictable distributions. The existence of such cases is of interest for two reasons. First, we need to discuss whether real world instances of a certain problem are usually closer to the average model or often include degenerated cases. Secondly, the susceptibility to such degeneration differs considerably from problem to problem.

This chapter is completed with cost distribution of fully fledged TPC-H queries extracted from Microsoft SQL Server, confirming our findings for the simplified models of cross product and join order optimization.
4.1 Topologies and Landscapes

Ever since the introduction of blind search algorithms, relentless effort has been devoted to characterizing the search space and its influence on the search algorithms. Usually, the terms topology and landscape are used to describe certain properties and relations among solutions. Though an important means for the interpretation of certain effects occurring in some optimization algorithms these structures are not intrinsic to the problem.

Given an optimization problem we are usually able to develop a certain notion of distance between solutions that expresses the degree of similarity. Consider for example the Traveling Salesman Problem where the shortest tour via a number of cities is sought. Two tours that differ only in so far that two cities are exchanged while keeping the order of the remaining could be considered similar. More general, the distance between tours could be expressed by the minimal number of pairwise exchanges needed to transform one tour into the other. A pair of tours with distance 1 could be referred to as neighbors. Ergo, we defined a topology on the search space.

This particular distance measure may appear somewhat coarse as it does not take into account any locality. A natural demand for a similarity measure could also involve similarity in terms of the value of the objective function for both neighbors. Exchanging arbitrary cities, however, may result in a neighborhood between the optimal tour and a tour which is nowhere near to be optimal with respect to its length. Thus, it seems reasonable to restrict our distance measure to pairs of cities that are neighbors within the tour, i.e., they visited subsequently without further vias in between. Consequently, we defined a new, fundamentally different topl-
ogy on exactly the same search space. Only for a very small set of tours the distances are the same under both distance measures. In Figure 4.1a, the optimal tour of a simple Traveling Salesman Problem is shown. Possible neighbors under exchange of subsequent and arbitrary cities are depicted in 4.1b and c, respectively.

Our definitions of distances can be used in a rather straight forward way to generate neighboring tours given one initial tour by swapping pairs of cities—either a pair of neighbored or non-neighbored cities. A large variety of different schemas for a guided exploration of neighborhoods has been suggested in the literature [FJMO95]. These search mechanisms proceed transitively in the sense that they investigate also neighbor of neighbors etc. if they are promising.

In order to conduct such a guided exploration of neighborhoods in the quest for the best tour, all tours have to be seen with respect to the objective function, the length of a tour. Combining both those components, a structure which is often referred to as landscape—specifically in the context of genetic algorithms as fitness landscape—is obtained. In case the topology defines a planar graph, the landscape can be visualized as a two-dimensional manifold in \( R^3 \).

Viewing our landscapes for the Traveling Salesman Problem from this perspective we obtained one rugged and one relatively smooth landscape for the same problem. Kaufman, one of the pioneers in fitness landscapes developed several models and notions of landscapes along the line of different degrees of ruggedness including techniques how to transform or modify them [Kau93].

However, neither of the two landscapes sketched above is intrinsic or natural to the current problem. The problem is well-defined without any such component. So, the only characteristics of the search space are the size of the space and the costs of all its elements. To avoid any misunderstanding at this point, certain optimization algorithms need and define indeed such landscapes, and we will also devise some models for them in one of the next chapters. However, for the analysis of the search space independent of a search algorithm, these models are only of limited use.

Furthermore, we will avoid the notion of fitness but prefer the term costs as fitness can be interpreted—and is indeed often used this way—as relative fitness. In Section 4.2.3 we will give an example for such a relative fitness measure. As opposed to this, costs refer to the absolute value of the objective function.

### 4.2 An Excursion: Some NP-Complete Problems

In order to get an impression of the concept of cost distributions and its parameters we make a short excursion and analyze and model three fundamental problems. These examples will also be also be useful for further considerations of the difficulty of problems in the following chapters.
4.2.1 Number Partitioning

We define the problem POPT as the associated optimization problem of the PARTITIONING problem [GJ79]:

Let \( S \) be a set, and \( w \) a weight function that assigns every element of \( S \) a certain value. Find a partitioning of \( S \) into \( S_1 \) and \( S_2 \) such that the expression

\[
\left| \sum_{s \in S_1} w(s) - \sum_{s \in S_2} w(s) \right|
\]

is minimal. PARTITIONING in NP-hard and so is POPT.

We model the sizes of the partitions with random variables \( S_1 \) and \( S_2 \) denoting the size of \( S_1 \) and \( S_2 \) respectively. It is sufficient to focus on \( S_1 \) since \( S_2 = |S| - S_1 \). The set sizes are binomially distributed with equal probability, i.e., for every element, the probability to belong to either \( S_1 \) or \( S_2 \) is \( \frac{1}{2} \). For the expected sizes of the sets

\[
E(S_1) = \frac{1}{2} |S|
\]

holds. We model \( w \), the individual weights with a random variable \( \omega \). Let \( \mu \) and \( \sigma \) be mean and deviation of \( \omega \) and

\[
X := \sum_{i=1}^{S_1} \omega
\]

be the weight of a set. According to the central limit theorem of statistics we can approximate the distribution of the weight of a set by a normal distribution \( N(\mu, \sigma) \) with \( \mu = \frac{N}{2} \mu \) and \( \sigma = \sqrt{\frac{N}{2}} \). Without loss of generality, we assume \( \mu = 0 \). We denote the density of \( N(0, \sigma) \) with \( \phi \).

The density of the approximated cost distribution is then

\[
\psi(x) = \int_{-\infty}^{\infty} \phi(t) \cdot \phi(x + t) dt
\]

for \( x \in \mathbb{R}^+ \).

In Figure 4.2 an experimentally obtained cost distribution—i.e., its density—is contrasted with the approximated one. \( S \) is implemented as normally distributed set of random numbers with mean \( \mu = 0 \) and deviation \( \sigma = 10 \). The size of \( S \), \( |S| \) is 100. The experimental data was obtained by a sample of size 10000. The frequency of occurring differences and the analytically determined probability are shown as functions over the weight differences. The figure shows strong coincidence between the analytical and the experimental results.

The most important feature of the cost distribution is its monotonic decrease which implies that the difference zero, the optimal result, occurs with the highest probability of all solutions.
The accuracy of our analytical assessment relies on the abstraction according to the central limit theorem. The less resemblance the distribution of the sum of weights of the partitions bears with the normal distribution the less accurate our results are, or are to be expected. Apparently, some pathological cases are easy to construct. Consider for example an instance, where all elements have the same weight, or only very few different values. The resulting distribution consists of only one or a few points respectively. Consequently, the approximation by a continuous function is not very meaningful.

On the other hand, if the contribution of the single weights to the sum is sufficiently small, the abstraction by normal distributions is justified independently of the particular distribution of the weights. To assess the susceptibility of this assumption, we varied the distribution and deviation of the random numbers in the original set and also varied the number of elements.

As a measure of difference between experimentally and analytically obtained distribution, we use the Kulback-Leibler Divergence—also known as relative entropy—, which is defined as

$$D(P, Q) = \sum_x P(x) \cdot \ln \frac{P(x)}{Q(x)}$$
for two distributions $P$ and $Q$. The divergence is always non-negative and the smaller the value of this expression the stronger the distributions resemble each other. For instance, the divergence of the distributions in Figure 4.2 is less than 0.006.

In the following experiments we used randomly numbers generated according to Normal and Gamma distributions.

For the experiment with normally distributed numbers, we varied the deviation from 1 to 50 and the size of the original set between 2 and 25. In Figure 4.3, the divergence as function of size and deviation is shown. Very small set sizes, result in the highest divergence, which does not come as a surprise since the deviation for 2 elements for example can hardly compare to a continuous function. Distributions for very small sets contain mostly noise. However, the distributions become quickly stabilized with increasing size. For sets with 10 or more elements, the approximated distribution virtually coincides with the experimental one. As the plots show, this process is independent from the deviation of the underlying distribution of
the numbers.

One might argue, that using a symmetric distribution for the numbers is the reason for the latter effect. In the next experiment, we used numbers distributed according to a Gamma distribution with shape parameters $\alpha$ between 1 and 5. The Gamma distribution is asymmetric coinciding with an exponential distribution if $\alpha$ equals 1. In Figure 4.4, the divergence is plotted as a function of the set size and $\alpha$. As the figure shows, the influence of the underlying distribution is negligible. Again, the set size is the dominating factor: For a problem instance with more than 10 elements, the analytical approximation is highly accurate.

### 4.2.2 Traveling Salesman Problem

The next problem we analyze is the Traveling Salesman Problem, one of the classic problems in combinatorial optimization. Given a graph with nodes,
connected by edges of certain lengths, we are interested in the shortest possible Hamiltonian Circuit, i.e., a complete tour where every node is visited exactly once. The length of a tour is the sum of the lengths of its edges [GJ79].

The variant we consider here is known as Symmetric Euclidean Traveling Salesman Problem and is given by the co-ordinates of the nodes. Every node is connected to every other node by an edge the length of the Euclidean distance of the two nodes. In Figure 4.5, an example is shown. The nodes represent the 13509 cities of the United States with more than 500 inhabitants. This particular problem ranks among the largest Traveling Salesman Problems that have been solved to optimality to date [ABCC98].

In more formal terms, we can state the problem as finding a permutation $v_1, \ldots, v_n$ of cities given as two-dimensional vectors such that

$$l = |v_n - v_1| + \sum_{i=0}^{n} |v_i - v_{i-1}|$$

is minimal. The first part of the sum, results from the condition to connect first and last point of the tour.

In order to approximate the cost distribution of a problem instance we first determine mean $\mu_c$ and deviation $\sigma_c$ of the pairwise distances of the cities. For our example, the histogram of the associated distribution is given in Figure 4.6.

We model

$$Y := \text{"Length of random tour"}$$
by

\[ Y := \sum_{n} X, \]

where \( X \) is a normally distributed random variable with parameters \( \mu_c \) and \( \sigma_c \).

According to the central limit theorem of statistics, we can approximate the cost distribution of all tours through \( n \) cities by a normal distribution with mean

\[ \mu_a = n \cdot \mu_c \]

and deviation

\[ \sigma_a = \sqrt{n} \cdot \sigma_c. \]

Figure 4.6 shows both cost distributions obtained from a sample of size \( 10^7 \) and the approximated one for the example problem. The approximated distribution resembles the actual, obtained by sampling, setting off the symmetry as the major feature of the distribution. Despite the length of the sum and the seemingly coarse approximation by \( \mu_c, \mu_a \) appears highly accurate. Only the deviation \( \sigma_a \) is slightly too large.
Observation.

As with all minimization problems, the left most quantile of the cost distribution contains the optimal solution. However, the sample can be somewhat misleading since it does not unveil how far to the left the optimal tour is. Theoretically, the optimal tour could have length 199000 as well as 1990 without influencing the shape of the remaining distribution. In this particular example, the optimal tour is of length 57201.

As a consequence, the probability to find the optimum or any tour of length close to it in a random sample is almost zero, i.e., sampling limits us to an analysis of the majority of the plans, which have length close to the average tour. To correct the histogram in this respect we need to adjust the range to cover minimal and maximal solution. For nearly all TSP problems in TSPLIB, the optimum is known. For the few, very large instances that have not yet been solved to optimality, at least tight bounds for the optimum are known. For the longest possible tour, we use an approximation generated with a greedy heuristic. In Figure 4.7 the rectified histogram is shown with characteristic points marked up. We will discuss the effect of these characteristics on optimization approaches in the next chapter and focus on the shape only for the moment.

As it became clear from the abovementioned example, a comparison of the actual and the approximated distribution has to take heavy tails into ac-
count, that is, we need an appropriate measure to compute a distance. The integral over the difference is not suitable as the distributions may for example differ by this means by a value close to 2—the maximum distance of two densities under this metric—although both distributions have almost all their weight in the same quantile of one mere per cent. To overcome this problem we take two separate measures, the overlap and the similarity of shape. The first one is expressed by the difference of the means, the second by the Kullback-Leibler Divergence after centering the distributions on their means. In all experiments, the sampled tours were generated with uniform probability. All instances show distributions that are symmetric and largely coinciding with the normal distributions used for approximation as the small values of $\kappa$ indicate.

Another effect deserving special attention is the fact that the means sampled as well as approximated are in the right half of the total range. At first glance, one might attribute this skew to the fact that the left edge of the distributions is the absolute minimum whereas the right is only approximated with a greedy heuristic. However also the minimum obtained by greedy optimization is distinctly further away from the mean than its counterpart, the maximum. The explanation requires a look at the distribution of the pairwise distances between cities. This distribution is skewed as shown in Figure 4.6. Though not important for the Central Limit Theorem and the shape of the approximation, the skew determines the length of the distribution's tails. For very small problem sizes, this effect is almost negligible but gains importance with increasing problem size.

To test for the generality of our observation, we sampled and compared cost distributions for all 77 instances of the euclidean symmetric TSP in the TSPLIB [Waa99a]. The results are, without any exception perfectly in line with the findings above. In the appendix (page 177), a short overview of these extensive experiments is given in form of tables containing all characteristic values such as total range, sampled range, experimentally and analytically determined mean and deviation, as well as the divergence of approximated and actual distribution.

### 4.2.3 Knapsack Problem

The last problem in this excursion is the 0/1 Knapsack Problem. It is described by a set of items which have profit and weight associated with them. The optimization task is to find a subset with the highest possible accumulated value so that its total weight does not exceed a certain capacity [GJ79]. The problem owes its name to the analogy of packing a knapsack.
In formal terms:

\[
\begin{align*}
\text{maximize} & \quad \sum_i p_i x_i \\
\text{subject to} & \quad \sum_i w_i x_i \leq c \\
& \quad x_i \in \{0, 1\}
\end{align*}
\]

where \(x_i\) is a binary value that indicates whether item \(i\) is to be included in the subset or not.

Like with the Traveling Salesman Problem, branch and bound algorithms in various forms became the standard techniques to solve this problem and its derivatives. For a survey on developments in this area see e.g. [Pis95].

The Knapsack problem offers further interesting insights as it can be interpreted as a multi-objective optimization problem with the additional condition

\[
\text{minimize} \quad \sum_i w_i x_i,
\]

i.e., while maximizing the profit, at the same time the weight is to be minimized.

The valid solutions are to be interpreted in a two-dimensional space which is not ordered, i.e., two elements \((W_1, P_1), (W_2, P_2)\) are incomparable if the pairwise comparisons of weight and profit are contradictory. For example, neither of the solutions \((1000, 1000)\) and \((2000, 2000)\) is superior. We say the solution \((W_1, P_1)\) dominates the solution \((W_2, P_2)\) if \(W_1 < W_2\) and \(P_1 > P_2\). In general there is no single optimal solution that dominates all the others.

When discussing fitness functions in Section 4.1 we already mentioned the term relative fitness, however, without giving a description yet. We can close this gap with the example at hand. An immediate definition of fitness is apparently its two dimensional cost value. However, many optimization algorithms need a one dimensional cost value. One way to obtain such a value is to define the fitness of a solution \(s\) relative to a set of \(S\) of solutions as the number of solutions in \(S\) that dominate \(s\). A solution with fitness \(0\) is then preferable to a solution with higher values.

The cost distribution is independent of whether or not we analyze the single- or the multi-objective variant.

To model the problem, we first consider the case

\[
\sum_i x_i = k, \quad \text{with} \ k > 0
\]

where a knapsack has to contain exactly \(k\) items. We can approximate both dimensions with normal distributions. If we assume the weight independent of the profit, we obtain the density:

\[
\phi_k(w, p) = N(k \cdot \mu_w, \sqrt{k} \cdot \sigma_w) \cdot N(k \cdot \mu_p, \sqrt{k} \cdot \sigma_p).
\]
To determine the density for the general case, we need to sum over all possible values of \( k \) and multiply with the probability to have \( k \) items in the knapsack:

\[
\phi(w, p) = \frac{1}{2^n} \sum_{k=1}^{n} \binom{n}{k} \cdot \phi_k(w, p)
\]

In Figure 4.8 and 4.9, the cost distribution obtained by sampling and its analytical approximation are shown. The plots exhibit the weak spot of our modeling. The assumption that weight and profit are completely uncorrelated is hardly ever fulfilled but the effects of the correlation are significant. In case that weight and profit are strongly correlated, e.g. weight equals profit, the distribution reduces in width the extreme of which is a one dimensional distribution along the diagonal. For decreasing correlation, the distribution grows increasingly wider until it reaches the shape depicted in Figure 4.9. This behavior could be modeled by also taking the covariance of
both random variables, weight and profit, into account. However, for our purposes, it is sufficient to understand the parameters and the quality of the shape.

4.2.4 Discussion

The lessons learned from this excursion can be summarized as

- Cost distributions are characteristic for optimization problems, as they reflect the basic properties of the cost function.

- For instances of non-trivial size, the distributions appear often very stable; Cost distribution of small instances may contain a considerable ratio of noise though.

- For most problems, pathological cases can be found, where the cost distribution collapses. For instance, all weights and values in the Knapsack Problem are equal or all cities in the Traveling Salesman Problem have the same coordinates.
We have seen two distinctly different types of cost distributions: With POPT, the optimum had the highest frequency, whereas with napsack and Traveling Salesman Problem the probability of sampling the optimum or any solution near it was practically zero. Yet, the latter two can be distinguished further by the distance of the optimum to the average solution. In Figure 4.10, the three types are sketched qualitatively. We have to be aware, that such a classification can cover real cost distributions only in a qualitative but not a quantitative way, i.e., cost distributions of other problems may appear to be of a shape that does not fit any of the three but is rather between two types. This differences need to be taken into account when discussing the effects on optimization algorithms and may require an “interpolation”. We will get back to this point in Chapter 6 when discussing the effects of cost distributions on evolutionary algorithm.

Finally, a point that is important for a transfer of the results to query optimization is the nature of the cost functions scrutinized. In all three cases, the cost functions are additive in the sense that the total cost of a solution is the sum of certain input parameters. Accordingly the central limit theorem applies allowing a fairly accurate approximation.

4.3 Cross Product Optimization

The problem \( \text{XOPT} \) of cross product optimization is a close relative of \( \text{JOPT} \) — in fact it can be expressed as a special case of the latter where all occurring selectivity parameters equal 1. Although it might seem easier than \( \text{JOPT} \) on first sight, it is also NP-hard as Scheufele and Moerkotte showed in [SM97].

The problem is defined as follows: Let \( M_n \) be the set of labeled, binary trees with \( n \) leaves as defined in 3.1. For a tree \( t \in M_n \), \( K(t) \) denotes the set
of all subtrees, including the complete tree \( t \). A function \( \gamma \) assigns every leaf a rational number. Find a tree \( t \) such that

\[
c(t) = \sum_{k \in K(t)} \prod_{v \in k} \gamma(v)
\]

is minimal, where \( v \in k \) means \( v \) is a leaf of \( k \).

A model for the cost distributions of non-isomorphic cross product trees is far more difficult than the ones presented in previous sections. Especially, the lack of a meaningful analytical description of the product of two normal distributed random variables confines us to strong simplifications already at an early stage. Nevertheless, we can derive trends and corroborate them experimentally.

### 4.3.1 Constant Relation Sizes

We represent the leaves of a tree by independent identically distributed (iid) random variables \( X_i \), i.e., we assume the leaves normally distributed with the same parameters. For the moment, let us assume the deviation of \( X_i \) is zero, that is, all leaves are constant and have the same value. Thus, the cost is

\[
\sum_{k \in K(t)} X_{|k|}.
\]

In \( M_n \), trees with the following ranks are distinguished. For \( j \leq \lfloor \frac{n}{2} \rfloor \)

\[
l_{n,j} = \left( \sum_{1 \leq i \leq j} B(n-i) \cdot B(i) \right) - 1
\]

and

\[
b_{n,j} = l_{n,j} + B(n-j) \cdot B(j) - 1
\]

denote the lowest and the highest rank of trees with \( n \) leaves in total and \( j \) in the right subtree.

**Lemma 4.3.1**

For all trees with rank \( r > l_{n,j} \), the following holds

\[
h(t(r)) < h(l_{n,j}),
\]

where \( h(t) \) denotes the height of a tree.

\( t(l_{n,j}) \) consists of a linear tree with \( j \) leaves and one with \( n - j \) leaves; \( t(b_{n,j}) \) consists of two balanced trees (see Figure 4.11).

**Proposition 4.3.2**

Let \( X_i = X \) be iid random variables with deviation zero, then the following holds:
Figure 4.11. Trees $t(l_{n,j}), t(b_{n,j})$ for $n = 12$ and $j = 4$

a) $c(t(l_{n,j})) \geq c(t(r)) \geq c(t(b_{n,j}))$, with $l_{n,j} \leq r \leq b_{n,j}$,

b) $c(t(l_{n,j})) \geq c(t(l_{n,j-1}))$

c) $c(t(b_{n,j})) \geq c(t(b_{n,j-1}))$

Proof:
Consider the following algorithm. Given the tree $t(l_{n,j})$

1. Remove a leaf at depth $d$ from $t$.

2. Insert the removed leaf anywhere in the tree at a depth less than $d$.

Applying this algorithm repeatedly we can generate any tree $t$ from $l_{n,1}$.

The cost associated with a tree is

$$\sum_i a_i X^i.$$ 

For every removal and insertion of a leaf we identify the closest common ancestor $v_c$ for the removed and the inserted leaf. Costs outside the subtree below $v_c$ are not changed by the maneuver. Assume this subtree has $k + 1$ leaves. The costs of the root, i.e., $a_{k+1}X^{k+1}$ can be ignored as they appear in both trees. The costs for the new tree are bound by

$$bX^j + \sum_{i=1}^{k-1} a_i X^i \quad \text{with } j \leq k, b = a_j + a_k$$

Thus, the cost difference between the original and the restructured tree is less than

$$\sum_{i=1}^k a_i X^i - \left( bX^j + \sum_{i=1}^{k-1} a_i X^i \right)$$
which can be rewritten into
\[
a_jX^j + a_kX^k - (a_j + a_k)X^j = \frac{1}{X^j}(a_j + a_kX^{k-j} - (a_j + a_k)) = \frac{1}{X^j}(a_kX^{k-j} - a_k),
\]
which is greater than zero for \( X > 1 \).

Notice, this does not imply that all trees are monotonic in costs, rather there is a sequence of trees \( l_{n,1}, t', t'', \ldots, t \) such that \( c(l_{n,1}) \leq c(t') \leq c(t'') \leq \cdots \leq c(t) \). All three propositions follow immediately from application of the above algorithm.

The proposition provides upper and lower bounds for sets of trees in \( M_n \) with the same number of leaves in their left subtree. Since our ranking function was developed on a similar criteria (see Sec. 3.1), we derive implicitly tight upper and lower bounds for ranges of ranks. Both upper and lower bounds are monotonic across the groups.

Figure 4.12 shows upper and lower bounds for trees with 20 leaves and \( \mu = 2, 5, 10, 100 \) respectively. The curves show a steep descend narrowing the cost range of plans quickly. For larger values of \( \mu \), this effects is intensified. Plots for larger values are, besides the scale, indistinguishable from the last one.

### 4.3.2 Variable Relation Sizes

In a next step we loosen the restriction of constant relation sizes which were due to a zero deviation. As a consequence a tree is no longer associated with one single cost value but rather a distribution. We use \( \phi_r \) to denote the distribution belonging to tree \( r \). The distributions of different trees typically overlap in range.

The results obtained in the previous paragraph extend to the general case as the bounds of the total costs in the previous setting are now the bounds of the means of the distributions. The shift of the bounds implies here a shift of the complete distribution.

In Figure 4.13, these distributions and their shift is illustrated for a tree with 10 leaves and \( \mu = 2, \sigma = 0.1 \). We use this small relation size to keep the whole distribution narrow so that details are clearly visible. We will discuss larger ranges later.

The diagram is to be read column-wise, that is, the ranks of the trees are given on the one axis (0 is the linear tree, 97 is the bushy most) and the according distribution is plotted along the other axis. For illustration, the distribution of tree 10 is highlighted. Because of the small relation size, the single distributions are almost symmetric which makes the means of each distribution intuitively accessible as it coincides approximately with the maximum of the distribution.
The figure clearly shows the trends analogous to the previous section: The means shift from the middle of the cost range for tree 0 to the left side with increasing rank, i.e. bushyness. As pointed out before, this shift is not strictly monotonic. Using a larger mean for $X$ has two consequences: The distributions (1) extend to a broader range and (2) show stronger asymmetry. This means for the shape of the result distribution that the “ridge” is further shifted to the left. The quality of the shape stays the same but is scaled in its extent. A larger deviation in $X$ also causes stronger asymmetries. The mean of the distributions per tree is independent of the deviation of the distribution used for $X$. 

Figure 4.12. Upper and lower bounds in $M_{10}$ as function of number of leaves in right subtree
Fitting the puzzle pieces of the last two section together, we can now construct cost distribution for $\text{XOPT}$

$$\phi = \frac{1}{|M_n|} \sum_{i=1}^{M_n} \phi_i$$

The normalizing factor in front of the sum is necessary as all $\phi_i$ are distributions by themselves, i.e. $\int \phi_i = 1$. Multiplying with $\frac{1}{|M_n|}$ ensures $\int \phi = 1$.

In Figure 4.14 the dependency of $\phi$ from mean and deviation of $X$ is shown. The plots show distributions for $M_{10}$ with $\mu = 2, 5, 10, 100$. The deviation is given as fraction of $\mu$, varying between 0.05 and 0.5. We cut the cost range down to $[0; 2\mu^n]$ to exclude outlying but uninteresting data points—beyond the right edge of the diagram, the curves converge further to zero but stretch an enormous interval. Including them in a linear scale would shift all interesting details close to the left fringe of the cost range scrambling all features beyond recognizability; using a logarithmic scale on the other hand causes significant distortions on the shapes of the curves.
All plots show strong similarity: for small deviation of the underlying relation sizes, the distributions are rather symmetric resembling a normal distribution for large values of \( \mu \), but shifts quickly to the left, resembling an exponential distribution, with increasing deviation.

As opposed to the cost distributions analyzed in Section 4.2, \( \text{XOPT} \) displays a wider range of variety, i.e., it is more sensitive to its parameters. This sensitivity is due to the multiplicative character of the cost function where little differences are magnified by subsequent multiplications.

Finally, let us compute the mean of the resulting cost distribution.
Proposition 4.3.3
The mean $\mu(\phi)$ depends only on the mean of $X_i$ but not on the deviation.

Proof: Let $Y_k$ be a random variable

$$Y_k := \prod_{i=1}^{k} X_i$$

modeling the product of $k$ base tables. Moreover, consider

$$f_{n,i}(t) = |\{t' \in K(t) : |t'| = i\}|$$

which determines for a tree $t$ with $n$ leaves the number of subtrees with $i$ leaves. The costs of a tree can be formulated as

$$c(t) = \sum_{i=1}^{n} Y_i \cdot f_{n,i}(t).$$

The expected value of $Y_n$ computes to

$$E[Y_n] = \prod_{i=1}^{n} E[X_i] = \mu^n.$$

With

$$v_n(i) = \sum_{t \in M_n} f_{n,i}(t),$$

the number of subtrees in $M_n$ of size $i$, we determine

$$\mu(\phi) = E[\sum_{i=1}^{n} Y_i \cdot f_{n,i}(B)] = \sum_{i=1}^{n} E[Y_i] \cdot E[f_{n,i}(B)]$$

$$= \sum_{i=1}^{n} \mu^i \cdot E[f_{n,i}(B)] = \sum_{i=1}^{n} \mu^i \cdot \frac{v_n(i)}{|M_n|}.$$  

Notice, $v_n(i)$ refers to $M_n$ and not to a single tree alone. $v_n(i)$ computes to $v_n(i) = 0$ if $n < i$, $v_n(n) = t(n)$ and

$$v_n(i) = \begin{cases} 
\sum_{j=i}^{n-1} t(n-j) \cdot v_j(i), & \text{if } n \text{ is odd} \\
v_{n/2}(i) + \sum_{j=i}^{n-1} t(n-j) \cdot v_j(i), & \text{if } n \text{ is even} 
\end{cases}$$

otherwise.

On the other hand, we have seen the influence of the deviation of the $X_i$ above. The higher the deviation, the more distinct the skew. As a consequence of the above proposition, the cost range expands with increasing deviation, i.e., the cost distribution contains an increasing number of outliers with high costs. In Figure 4.14 we concealed this effect by cutting the right tail off to avoid the strong distortion caused by the very long tails otherwise.
4.4 Join Order Optimization

We already gave a detailed outline of the join order problem in Section 2.2. However, at this point we derive a definition on the basis of the previous tree models in order to transfer previous results.

The join order problem $J_{opt}$ extends $X_{opt}$ by predicates which define selectivity between pairs of relations. This selectivity is determined by statistic data about the involved relations $R_1, R_2$ and is given as a numerical value $p(R_1, R_2)$.

If $R_1$ and $R_2$ are connected nodes of the join graph, $p(R_1, R_2)$ is a value between 0 and 1 including both, otherwise $p(R_1, R_2) = 1$. This definition extends to selectivities between sets of relations with

$$p(R, S) = \prod_{i,j} p(R_i, S_j), \quad R_i \in R, S_j \in S.$$ 

Applying this definition recursively computes the selectivity of a tree $t$ as

$$p(t) = p(\{R_1, \ldots, R_n\}, \{S_1, \ldots, S_m\})$$

where $R_i \in t_l$ and $S_i \in t_r$. In case the join graph is tree-shaped, there is at most one predicate per subtree less than 1, in a clique graph there can be up to $|t|$. Using this notion of selectivity, we can define $J_{opt}$ to find a tree $t$ such that

$$c(t) = \sum_{k \in K(t)} c(k_l) \cdot c(k_r) \cdot p(k)$$

is minimal. $k_l$ and $k_r$ are left and right subtrees of $k$, and $K(t)$ as defined as above.

This definition also allows for the inclusion of cartesian products. Normally, one wants to exclude cartesian products as they—usually but not always—lead to more expensive plans and, which is especially important for exhaustive search algorithms, extend the search space by a large factor. Depending on the join graph the up-scale can be exponential.

A variant excluding cartesian products can be defined by the additional condition

$$p(t) = 1 \iff \exists (R, S) \in G, R \in t_l, S \in t_r$$

where $G$ denotes the join graph. This definition permits cartesian products only if they are forced by the user's query.

In order to understand which of the previous results offer the possibility to transfer to the more general case of $J_{opt}$ we first want to point out the differences and common grounds.

Essentially, $J_{opt}$ differs in three points: possible shapes of trees are limited to certain sets reflecting the join graph, the selectivities at each node in form of an additional factor have to be accounted for, and lastly, different algorithm may be used to implement the join operators.
Join Graphs

According to the join graph only a subset of $M_n$ can be chosen. The limitations in this respect vary drastically. The extremes of which are the complete graph and the star graph. For the complete graph, the $n$-clique, all trees are feasible, for the star graph, where all relations are connected to one central relation, only one single shape, the linear tree, is possible. Unfortunately, no concise way of describing the remaining graphs is possible.

For the complete cost distribution of the JOPT, this means, that it consists, compared to the XOPT, only of a subsequence of $(\phi_i)_i$. Rather than the slowly drifting sequence we studied above, we obtain only a few parts of the sum. Figure 4.15 shows an example where only 4 out of 98 possible tree shapes are used. Notice, the linear tree is always a feasible shape.

In the previous section we were mainly interested in the quality of the shapes and cut off the distribution at $2\mu^n$ to avoid distortion in the display and retain enough details of the distribution. However, for the further discussion it is also interesting to see the means—particularly the mean of $\phi_0$—with respect to the complete cost range. In Figure 4.16, $\mu(\phi_0)$ is plotted as function of the deviation of the underlying relation sizes. In our model, the linear tree formed an upper bound. Accordingly, using less than $|M_n|$ trees makes for a complete cost distribution similar to $\phi_0$, and exactly $\phi_0$ if only one tree is used. However, as all possibly omitted $\phi_i$ are left of $\phi_0$, $\phi$ cannot be further to the right than $\phi_0$ is.
Selectivities

The role of selectivities is more difficult to assess. If we consider them, for the moment, independent of the particular relation, their effect can be largely disregarded—it is merely an additional multiplication with a random variable in a larger product of random variables.

However, in combination with dependencies, their influence is much stronger. They make the complete cost distribution appear more rugged in general. Tree shaped join graphs imply that all joins use only one predicate. Join graphs with cyclic edges can have several predicates on one join, and more notable can have different predicate combination on a join, if the shape or labeling of the tree is changed. The consequences are, on one hand that the selectivity on the nodes is stronger, i.e., closer to zero, which also causes a stronger left shift of the complete cost distribution. On the other hand the combination of predicates on one join provides higher stability since a join then does not only rely on one single predicate, thus the effect of single predicates is reduced. The latter argument, however, applies only to join graphs with a sizeable number of cycles.

Join Implementations

The discussion of the role of join implementations has been enlivened lately by Graefe [Gra99], pointing out that improvements of sophisticated join algorithms are measurable but not significant in general. Especially in small queries, the right choice of join implementations can have severe impact—in larger queries only if they are dominated by one or very few large tables. However, in most practical cases differences are rather small.

Another point reducing the influence of join implementation on the complete cost distribution is the fact that implementations are optional but not excluding, i.e., nested-loop-join is always a possible implementation; in the event of an equality predicate additionally the implementation of a hash-join becomes also feasible. In total, different join implementations intensify or smoothen dents in the complete distribution.

Before presenting own results corroborating this line of argumentation we also include further operators and consider the case of holistic query optimization in the next section. At this point however, a look at related work confirms our observations: Ioannidis and Kang [IK90, IK91] conducted several experiments where they determined the cost distribution for particular instances of $\textsc{JOpt}$ by random walks in the search space. The number of solutions inspected was chosen large enough to obtain a fair sample. In this experiments they observed distributions similar to what they describe as Gamma distributions with shape parameter $\alpha$ between 1 and 2—the previous corresponds to exponential distributions, the latter is a strongly asymmetric distribution. In Figure 4.17, several instances of this distribution are shown. The statement of Ioannidis and Kang also applies to
the distributions we found in Section 4.3: for large deviation of the underlying random variables the distributions found resembled an exponential distribution having a significant fatter tail though; for smaller deviations, the complete distributions shifted to the right (see Figure 4.14). The same causality was reported by Ioannidis and Kang in terms of catalog variance. However, they do not specify the term catalog variance in detail. Particularly, this lack of specification lends strong support that the multitude of parameters they included additionally into their cost models are only of marginal influence to the shape of the distribution.

Further observations similar in trend have been reported on by Swami [Swa91], König-Ries et al. [KRHM95], Stillger and Spiliopoulou [SS96], as well as Steinbrunn et al. [SMK97]. However, the methods used for sampling were usually not fair, in the sense that the sample was not generated with uniform probability.

### 4.5 Meeting Reality

As sketched in Chapter 2, the relational algebra contains besides joins a large number of additional operators. In a practical implementation this number is further extended to allow for implementation details of both operators and underlying storage model.

Most of these operators are rather inexpensive compared to the join or cartesian product. Specifically unary operators like filters are of almost negligible cost in traditional I/O dominated cost models, though gaining higher importance in main-memory settings [BMK99]. The two most notable exceptions are the sort operator and its close relative aggregation.
Both can contribute severely to the total execution cost of a query however, without the variety of possible combination as seen with join and cartesian products. That is, the contribution of those operators is a significant additional cost with small deviation.

How does this multitude of operators—together with a larger number of additional parameters in an industry quality cost model—now influence the cost distribution?

We can expect a similar effect as seen with the three optimization problem in Section 4.2. Increasing complexity in form of larger instances stabilized the basic shape distribution. In query optimization a similar effect is to expect as a larger number of operators means also a larger number of very small changes—besides a few possibly more severe.

In the following we determine cost distributions of real-world examples as they are encountered in commercial query processing, i.e., using a full-blown set of relational algebra operators.

Using our framework we are able to perform a fair random sampling of costs in the search spaces that are not limited to join ordering only but may include arbitrary relational operators, various kinds of indexes and aggregates, and even cover parallel processing. We carried out numerous experiments with both standard benchmark queries like TPC-H and customer queries taken from various applications [Tra99]. Under the precondition that the queries were of sufficiently large size the distributions obtained were characterized by a relatively strong concentration of costs close to the optimum.

Figures presented here are the result of experiments with TPC-H queries 5, 7, 8, 9, which are the join-intensive queries of the benchmark, and have a larger search space. Table 1 summarizes some of the relevant values obtained. The first four rows consider a space of alternatives that does not allow cross products; while the last four rows allow cross products. Each experiment consists of a random sample of 10,000 plans from the space. The measure of a very large number of plans in the space does not imply

<table>
<thead>
<tr>
<th>Query</th>
<th>#Plans</th>
<th>Min*</th>
<th>Mean*</th>
<th>Max*</th>
<th>costs* ≤ 2</th>
<th>costs* ≤ 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q5</td>
<td>68572049</td>
<td>1.14</td>
<td>17098</td>
<td>4034135</td>
<td>0.47%</td>
<td>12.15%</td>
</tr>
<tr>
<td>Q7</td>
<td>22810757</td>
<td>1.15</td>
<td>3318</td>
<td>178720</td>
<td>0.11%</td>
<td>44.55%</td>
</tr>
<tr>
<td>Q8</td>
<td>20112521035</td>
<td>1.01</td>
<td>111</td>
<td>609</td>
<td>1.11%</td>
<td>14.70%</td>
</tr>
<tr>
<td>Q9</td>
<td>67503460</td>
<td>1.10</td>
<td>4107</td>
<td>109825</td>
<td>0.11%</td>
<td>4.08%</td>
</tr>
<tr>
<td>Q5*</td>
<td>455348910</td>
<td>1.23</td>
<td>105418</td>
<td>1287700</td>
<td>0.29%</td>
<td>5.70%</td>
</tr>
<tr>
<td>Q7*</td>
<td>390737772</td>
<td>1.48</td>
<td>1793052</td>
<td>1523086611</td>
<td>0.03%</td>
<td>2.79%</td>
</tr>
<tr>
<td>Q8*</td>
<td>4432829940185</td>
<td>1.31</td>
<td>28159718</td>
<td>32595091399</td>
<td>0.06%</td>
<td>1.83%</td>
</tr>
<tr>
<td>Q9*</td>
<td>250657568</td>
<td>1.30</td>
<td>38363213</td>
<td>35866936219</td>
<td>0.02%</td>
<td>7.00%</td>
</tr>
</tbody>
</table>

*as factor of the optimum; *including Cartesian products

| Table 4.1. Parameters of search spaces of TPC-H join queries. |
that a structure requires as many bytes—recall that the plans are obtained through composition and reuse of operators from the compact encoding of the MEMO structure. All costs are normalized to the optimum plan found by the optimizer, which has cost 1.0.\footnote{The actual values are proprietary information not published yet.} The "min" column shows that with a relatively small sample, we are able to find plans that are pretty close to the optimum. In fact, the percentage of plans that are within twice the optimum cost is non-trivial. Also, it should be noted that the results are slightly different for the different queries, which vary in their selectivity and other properties. But there are no dramatic differences, and the same trends can be seen in all the experiments.

Figures 4.18, 4.19, and 4.20 provide additional information about the shape of the distributions. The space with cartesian products is shown,
but the other is similar. Figure 4.18 shows the histogram of the complete sample, and it shows that most of the plans concentrate on the left, close to the cheap plans.

Figure 4.19 zooms in to the lower 50% sampled costs; that is, the part of the distribution that makes up for 50% of the space. Finally, Figure 4.20 shows a further zoom, to the points that are up to 50 times the cost of the optimum. In the "macro" view, we find that plans tend to be clustered to the left, close to the optimum solution. As we zoom in to the dense area, the histograms get less smooth, but they are still in the line of previous experiments.

The distributions of queries that contained few tables were of no particular shape but consisted only of random noise (e.g. TPC-H 6). Although it is hypothetically possible to devise queries of arbitrary size where the cost
distribution degenerates to a single point—e.g. the cross product of several instances of the same table, with a space restricted to be linear joins—we never observed any such tendency in practical instances or customer queries.

4.6 Summary

Cost distributions characterize a combinatorial optimization problem as they are solely reflecting the objective function, uninfluenced of any element that is part of the solution rather than the problem. To view cost distributions without any connection to topologies and particular optimization algorithms is a novel approach toward an analysis that is not limited to certain algorithmic elements used to tackle the problem.
Typically, similar analyzes in previous work superimposed a kind of topology in form of transformation rules, crossover operators or heuristics first and restricted itself to a particular perspective on the problem.

In this chapter we analyzed cost distributions independent of such limitations, for several archetypal NP-hard problems: Partitioning, Traveling Salesman, and Knapsack Problem. These problems display three basic types of cost distributions which we will resort to at a later stage again. This excursion gave an impression of the fundamental means needed to undertake such an analysis but also to assess stability and generality of cost distributions on a variety of distinctly different problems.

Using the enumeration techniques developed in the previous chapter we derived cost distributions for both a simple cartesian model based on cross products only and full-blown query optimization. Results presented show that basic properties found in the cartesian model extend to the much richer problem of query optimization as they have the fundamental property of multiplication as a dominant principle of cost computation in common.

In the further analysis we therefore use approximations based on Gamma distributions to model the query optimization problem and to study various effects in the optimization process.