Probabilistic analysis of optimization algorithms: some aspects from a practical point of view

Karl-Heinz Borgwardt

Angaben zur Veröffentlichung / Publication details:

Borgwardt, Karl-Heinz. 1987. "Probabilistic analysis of optimization algorithms: some aspects from a practical point of view." *Acta Applicandae Mathematica* 10 (2): 171–210. https://doi.org/10.1007/BF00046618.

Nr. 15

Probabilistic Analysis of Optimization Algorithms-Some Aspects From a Practical Point of View

BY
K.H. Borgwardt

Preprint Nr. 134 1987

März 1987

Institut für Mathematik
Memminger Str. 6
D-8900 Augsburg

Probabilistic Analysis of Optimization Algorithms Some Aspects From a Practical Point of View

Abstract

In this paper the utility and the difficulty of probabilistic analysis for optimization algorithms arediscussed. Such an analysis is expected to deliver valuable criteria- better than the worst case complexity - for the efficiency of an algorithm in practice.

The author has done much work of that kind in the field of linear programmig. Based on that experience he gives some insight into the general principles for such an approach. He reports on some typical and representative attempts to analyze algorithms resp. problems of linear and combinatorial optimization. For each case he describes the problem, the stochastic model under consideration, the algorithm, the results and tries to give a brief idea of the way how these results could be obtained. He concludes with a discussion of some drawbacks and difficulties in that field of research. Among these are the strong sensibility with respect to the chosen model, the restriction of results to the asymptotic case, the restriction to somehow inefficient algorithms etc. These points are the reasons why probabilistic analysis is of limited value for practice today. On the other hand they show which principal problems should be attacked in the future to obtain the desired utility.

- How much storage space is needed?

Here we restrict our interest to those questions, although there are many other important criteria like

- How easily can the algorithm be implemented on a computer?
- How difficult is it to understand the algorithm?
- How sensitively does the algorithm react on possible deviations in the input data?

Of course, A may not behave on all the Π - instances in the same manner and equally well. But the user needs rather general information in advance. We expect that difficult instances take more computation-time, storage-space and lead to a greater risk of deviations between output and theoretical solution. But we do not have a reliable measure for "difficulty". On the other side, we have the experience and the opinion that mostly, difficulty increases with the size of the given instance. So it is plausible to classify the problem-instances according to their size. Often the size of an instance (especially in usual complexity-theory) is measured by the encoding length L(I) of the given problem-instance I.
L(I) can be defined as the number of bits which are necessary to give a specific problem-description to the computer. So we obtain certain problem-subclasses

$$\Pi_n := \{I \in \Pi \mid L(I) = n\} \quad \text{for} \quad n \in \mathbb{N}$$

i.e., the class of Π - instances with encoding length n.
We should remark that in many special cases a classification according to certain problem-typical dimension-parameters is more appropriate.
For example: Problems dealing with the structure of graphs are often classified according to the number of nodes or to the number of edges. If this is the case in the following, we shall give a hint which classification-criterion is used for the problem under consideration.
Now we want to measure the efficiency in dependency of n.

2. Worst case analysis and polynomiality

A very conservative and cautious way to judge about the quality of an algorithm A for solving the Π - instances is based on studying its worst-case-behaviour.
Let us introduce the following notation for the result of application of A on I.

$d_A(I)$:= deviation of output from theoretical solution
$t_A(I)$:= time required for production of output
$s_A(I)$:= storage space required for job-run.

We define the worst-case complexity functions from IN into IR (or a corresponding one-dimensional space) by

$$\overline{d}_{A,\Pi}(n) = \text{Sup}\{d_A(I) \mid I \in \Pi_n\}$$

$$\overline{t}_{A,\Pi}(n) = \text{Sup}\{t_A(I) \mid I \in \Pi_n\}$$

$$\overline{s}_{A,\Pi}(n) = \text{Sup}\{s_A(I) \mid I \in \Pi_n\}$$

Often it is sufficient to have upper-and lower-bound-functions for these complexity-functions

$$f(n) \leq \overline{t}_{A,\Pi}(n) \leq g(n) \quad \text{for all} \quad n \in \mathbb{N} .$$

For practical purposes such bounds are preferred that can be represented in a simple form.

A very interesting question concerns the order of growth of the functions $\overline{t}_{A,\Pi}$, $\overline{s}_{A,\Pi}$ with n. Let us concentrate on time-complexity.

Definition

A is called a (time-) polynomial algorithm for Π if there is a polynomial function $g(n)$ such that

$$\overline{t}_{A,\Pi}(n) \leq g(n) \quad \text{for all} \quad n \in \mathbb{N} .$$

Edmonds has introduced the term "good algorithm" for such methods because only those have a chance to solve "large" instances.

Consequently, one can prove the nonpolynomiality of A on Π by showing that

$$n^k = o(\overline{t}_{A,\Pi}(n)) \quad \text{for all} \quad k \in \mathbb{N}$$

Here we use o and 0 in the usual sense, i.e. a sequence (a_n) is $0(b_n)$ if there is a constant $C \in \mathbb{R}$ such that

$$|a_n| \leq C b_n \quad \text{for all} \quad n \in \mathbb{N}$$

and (a_n) is called $o(b_n)$ iff

$$\frac{|a_n|}{|b_n|} \underset{n \to \infty}{\longrightarrow} 0 \quad .$$

Notice that all these are asymptotical criteria which describe the behaviour for large n.

A nonpolynomial algorithm may behave much better than a polynomial one in the case of moderate-sized n (for example: the polynomial n^3 is greater than 2^n as long as $2 \leq n \leq 9$) .

So the polynomiality-criterion is of limited value when we have to solve problems of moderate size.

3. Complexity of problems

Until now we have dealt with the behaviour of a certain algorithm A on a problem Π . Of course, the behaviour of A is not sufficient to judge about the difficulty of Π. Instead consider the class $\mathcal{A}$ of algorithms which are applicable to Π and guarantee an exact solution.

Then the best-behaving member of $\mathcal{A}$ can be used for describing the time-complexity of Π .

Definition

Π is solvable in $h(n)$-time if there is an algorithm $A \in \mathcal{A}$ such that A solves all $I \in \Pi$ exactly and

$$\bar{t}_{A,\Pi}(n) \leq h(n) \qquad \text{for all} \quad n \in \mathbb{N} ,$$

if such an estimation is possible with a polynomial $h(n)$, then Π is said to be solvable in polynomial time.

Note that we have one algorithm $A \in \mathcal{A}$ in mind which determines the worst-case-complexity of Π .It is not possible to make the algorithm choice for the single instances independently.
In order to relax that restriction a "nondeterministic" time-measure can be introduced. To be precise we should introduce a nondeterministic Turing machine, i.e. a machine which can duplicate its state at several points. Instead we explain this concept informally in the following way. Consider an algorithm which is able to repeat its search for the solution from the beginning with modifications of the search procedure.
Instead of counting the total computation time of that algorithm we count the time required for the shortest successful repetition. This is what we call nondeterministic solution time.
So we can define the nondeterministic complexity and nondeterministic polynomiality in a similar way as above. As far as decision-problems are concerned we distinguish between the class IP of all decision-problems Π which are solvable in deterministic polynomial time and the class INIP of all decision problems solvable in

nondeterministic time. It is an immediate consequence that $IP \subset INIP$, but it is an open question (very important for complexity theory) whether that inclusion is proper.
But we know that some problems in !NIP have a kind of "maximal difficulty", namely the INIP - complete problems.

A decision-problem Π is said to be INIP-complete if the following implication holds:

$$\Pi \in IP \Rightarrow \Pi \in INIP \quad .$$

Many decision-problems have been proven to be INIP-complete (see the list of such problems in [Garey/Johnson].We want to give some examples for provably INIP-complete decision problems:

- The satisfiability problem:
  Is a Boolean expression satisfiable by assignment of 0 or 1 to the logical variables occuring in the expression?

- the clique problem:
  Does an undirected graph (V,E) (V = set of nodes, E =set of edges)contain a clique (a complete subgraph where every pair of nodes is connected by an edge of E) of cardinality $k \in IN$?

- the Hamilton-circuit-problem:
  Does a graph have a Hamilton-circuit (a cycle containing every node just once)?

- the Colorability problem:
  Is an undirected graph k-colorable (can the nodes be assigned to k different colours, such that no two adjacent nodes have the same colour?)

Whereas usual complexity theory mainly deals with such decision problems with "yes-or-no"-answer, many other problems ask for certain values, elements, points etc. And many of those problems can be reformulated as decision problems. In our study we shall emphasize the analysis of optimization problems. Such an optimization problem can be defined as follows:

Definition
Given a set M and a function $f:M \to N$, with an ordered set N , find an element $\bar{m}$ of M such that $f(m) \geq f(m)$ for all $m \in M$ and the maximal possible value of f(m) .

Such an optimization problem can be relaxed to a decision problem in the following way:
Is there an element $\tilde{m}$ of M whose objective value $f(\tilde{m})$ is not less than a given bound $n \in N$.
We finish with a difficulty-measure for such optimization-problems.

Definition

An optimization problem-class Π is called INIP-hard if the corresponding decision problem is INIP-complete.

4. Drawbacks of worst-case analysis and the need for average-case analysis

Worst-case analysis gives a very pessimistic impression of the speed, the accuracy and the storage-requirements of an algorithm. We must be aware of the risk that the problem instance leading to the worst behaviour may be some kind of exception. So the information based only on worst case is rather sparse. Practice requires other criteria.

We have the feeling that worst-case analysis is closely related to traditional mathematical thinking, where highest-achievable generality is the or one of the most important aims. Practice - on the other hand - is not generality-oriented, but success-oriented. That means that a user wants his special problem-instance to be solved. The way how this can be managed does not matter. It would be typical for him to stop all his efforts of dealing with the theoretical problem as soon as the solution of the given instance is achieved. In addition, we observe the following effect, which is in strong contrast to the philosophy of mathematics. The user does not start with the application of the most efficient algorithms if they are highly sophisticated and complicated to understand and to implement. Only when simple methods have failed he becomes willing to get into the theory and to deal with more difficult algorithms.

So it seems to be necessary to avoid further drifting away of the two points of view. This aim could (partly) be achieved if the mathematician tries to find a formal structure to simulate the practitioner's preferences, his judgements and his behaviour such that this suits into mathematical logic.

One attempt in this direction lies in the assumption that the problem instances of Π are somehow distributed. So the different importance of instances because of their occurence in practice could be taken into regard. It is our hope that by underlying such realistic problem-distributions and by observing the induced average behaviour of the solution-method the mathematician comes to similar conclusions as the practitioner.

Also from the very theoretical point of view we see a need for average case analysis. There may be (we do not know whether this is true) a tendency of the following kind in the design of algorithms.

The more an algorithm is adapted (and improved in its efficiency) to the typical instance (or what the designer assumes as typical), the worse is its behaviour in untypical situations. If this is true then many of the improvements would be useless if we did not accept distribution models preferring the occurence of typical instances.

Of course, the introduction of such a distribution leads to a new dimension of uncertainty. We want to discuss that danger in IV.

Summarizing our arguments, we come to the conclusion, that besides the worst-case behaviour the average-case behaviour should be taken into regard before judging about algorithms and comparing their efficiency. Necessary for that information is a probabilistic analysis of the algorithms resp. the problems and this means a new challenge for mathematicians.

Π is called solvable in average polynomial time when there is an algorithm A which admits a corresponding estimation with a polynomial g .

In the following sections we shall restrict our considerations to such concepts of average case behaviour including the average analysis of possible deviations $d_A(I)$.

There are certain theorems of probability theory which turn out to be extremely useful for doing such an analysis. We list some of these tools

Theorem 1

If $X_1,...,X_n$ are random variables, then $E(X_1+...+X_n) = E(X_1)+...+E(X_n)$.

Theorem 2 Markov's inequality

If X is a nonnegative random variable and if $t > 0$, then $P(X \geq t) \leq \frac{E(X)}{t}$.

Corollary

If (X_n) is a sequence of random variables in a sequence of sample spaces $\{\Omega_n\}$ and if $X_n \geq 0$ for all n , then $P(X_n \geq 1) \leq E(X_n)$.

If, in addition, X is integer-valued, then $E(X_n) \underset{n \to \infty}{\to} 0$ implies $P(X_n \geq 1) \underset{n \to \infty}{\to} 0$

Theorem 3 Tschebyscheff 's inequality

For $t > 0$ and for a random variable X we have

$P(|X-E(X)| \geq t) \leq \frac{V(X)}{t^2}$, where V denotes the variance.

Corollary

If $E(X) \neq 0$, then

$$P(X=0) \leq \frac{E(X^2)}{[E(X)]^2} - 1 \quad .$$

Of similar importance are asymptotic laws.

Theorem 4 Weak law of large numbers

Let $X_1, X_2, ...$ be mutually independent random variables with a common distribution F and with $E(X_k) = 0$ for all $k \in \mathbb{N}$.

Set $S_n = X_1+...+X_n$. Then for all $\varepsilon > 0$ we have

$$P(\frac{S_n}{n} > \varepsilon) \underset{n \to \infty}{\to} 0 \quad .$$

Theorem 5 Strong law of large numbers

Let $X_1, X_2, \ldots$ be independent identically distributed variables with $E(X_k) = 0$ for all k. Then

$P(\frac{Sn}{n} \to 0) = 1$

Theorem 6 Central limit theorem

Let $X_1, X_2, \ldots$ be mutually independent random variables with a common distribution F Assume that $E(X_k) = 0$, $Var\ (X_k) = 1$.

As $n \to \infty$, the distribution of the normalized sums

$S_n^* = (X_1 + \ldots + X_n)/\sqrt{n}$

converges to the normal distribution with density $e^{-\frac{1}{2}x^2} \cdot \frac{1}{\sqrt{2\pi}}$

Finally we should introduce three different concepts of convergence under stochastical view, because most of the known results in probabilistic analysis are of an asymptotic nature.

Convergence in probability

Let $\{X_n)$ n = 1,2, be a sequence of random variables and X be a random variable. We say that

$Y_n \to Y$ in probability, if for every $\varepsilon > 0$ and $n \to \infty$ $\lim P\{|y_n - y| > \varepsilon\} = 0$.

A stronger notion is

Almost sure convergence

We say that X_n converges to Y almost surely if

$P[\limsup_{n \to \infty} X_n = Y = \liminf_{n \to \infty} X_n] = 1$

And even stronger is

Complete convergence

We say that X converges to X completely, if

$\sum_{n=1}^{\infty} P[|X_n - X| > \varepsilon] < \infty \quad \forall \varepsilon > 0$.

We specialize X_n and X to be the indicator functions of certain events. We say that a property holds almost everywhere if

$$\sum_{n=1}^{\infty} P[X_n \neq 1] < \infty \quad .$$

Equipped with these tools, the attempt of a probabilistic analysis of an algorithm (resp. problem) should be done in the following stages:

1. Introduction of a reasonable stochastic model for the distribution of the instances.

2. Characterization of the algorithm in stochastic terms depending on the (random) input data only.

3. Derivation and evaluation of formulas for the desired stochastic parameters (as mean values, moments quantities, etc.)

6. Outline of the paper

In the following sections we try to give an impression how probabilistic analysis of algorithms works, how successful it is, where its complications and drawbacks lie and what can be expected in the future. We do not want to call this paper a survey because of several reasons. First the field ist too broad, to inhomogeneous and unstructured to have a general overview. And the author certainly emphasizes those questions of his individual preferences (e.g. optimization problems). The second reason is that we desire to point out the general principles of such an analysis rather then listing all the approaches and the corresponding results. This aim seems to be better achievable by concentration on some typical and representative cases.
The author has done much scientific work in the Linear Programming case. This problem had been one of the most important open and challenging problems of probabilistic analysis. Based on that experience and on the comparison with other approaches to that specific problem we want to figure out general principles and study whether similar approaches can be used for other problems as well. For that reason we give a rather detailed description of the LP-analysis in the second chapter. In the third chapter we show at typical examples how other problems (mainly of combinatorial optimization) have been attacked, which stochastic assumptions were made, which types of results have been obtained and which tools have been used.
In the fourth chapter we try to discuss the utility and the drawbacks from a more general point of view. We shall describe the difficulties to find "real-world-distributions" and we give reasons why most of the existing results are of an asymptotic nature.

II. Experience with the linear programming case

1. The linear programming problem

One of the most important types of mathematical problems is the linear programming problem. It appears in numerous applications as search for optimal mixtures, composition of activities, allocation problems, cautious waste of resources, assignment problems etc.

A general formulation for that problem is:

Maximize $v^T x$

subject to $a_1{}^T x \leq b^1, \ldots, a_m{}^T x \leq b^m$

where $x, v, a_1, \ldots, a_m \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $m \geq n$.

$m, n, v, a_1, \ldots, a_m, b$ are called the parameters of the problem. They can be specified by attribution of the input data to obtain a special instance.
Every special instance defines a feasibility region X (which is a polyhedron) of points satisfying the restrictions. The -until today - most efficient and powerful algorithm to handle such problems is the Simplex-Method which had been derived by George B. Dantzig in 1947.

Phase I: Calculate a vertex x_0 of the polyhedron X .
STOP if it turns out that such a point does not exist.

Phase II: Start from the given vertex x_0 and construct a sequence of vertices $x_1, x_2, \ldots, x_s$ such that x_i and x_{i+1} are adjacent and $v^T x_i < v^T x_{i+1}$ for $i=0,\ldots s-1$ as long as this sequence can be extended. The algorithm stops at x_s if it turns out that
x_s is the optimal vertex or that an optimal vertex does not exist.

The construction rule for the successor vertex determines the special variant of the method. Since Phase I can be done in an analoguos way to Phase II (only a slight modification of the problem-instance is necessary), it makes sense to concentrate on Phase II for the moment.
The data necessary for the pivot steps can be taken from a tableau containing $m+1$ rows and $n+1$ columns. For the usual variants the renewal of the tableau (necessary in each vertex-exchange or pivot step) requires $0(mn)$ additions and $0(mn)$ multiplications. Whereas we have that moderate bound on the number of elementary arithmetic operations per step, information on s , the number of pivot steps, is rare.
But such a-priori-information on s would be valuable. We must be aware that s depends highly on

- the dimensions of the problem instance m and n

- the input data $a_1,...,a_m,v,b$
- the variant.

Since every variant V defines a separate algorithm , we should ask for the complexity of such a single variant. It makes sense to aggregate all instances with equal dimension-pair (m,n) and to study the behaviour of V on the class of (m,n) - problems. In addition we may count the elementary arithmetic steps without regard to the length of the occuring tableau-entries, because the calculation rules for the tableau guarantee that the entry-length remains polynomial in the input length.
So the crucial point for the complexity is the number of pivot steps s .
The worst-case step-number $\bar{s}_V(m,n)$ is defined as the maximal occuring length of such a vertex-sequence when V is applied to all (m,n) - problems. In contrast to the very good practical experience with the Simplex-Method, its worst-case complexity is rather bad.
The very certain opinion of the fifties that the algorithm is polynomial was deceived in the beginning seventies when Klee, Minty [Klee, Minty] and others derived exponentially increasing lower bounds on $\bar{s}_V(m,n)$ for almost all usual variants. So it was shown that every such variant has a practical solution gap, i.e. a set of problems which cannot be solved in acceptable time. Still there is a little hope that there could be a polynomial variant, but the chance to find such an algorithm and to prove polynomality does not seem to be high.
The "embarassing gap between what has been observed and what has been proved" [Gale] gave an additional motivation for studying the average behaviour of $s_V(m,n)$.
Consequently, one wants to know the expected value of $s_V(m,n)$, denoted by $E_{m,n}(s_V)$, under a given distribution of the input data.
Related but not identical investigations had been done by several authors in the field of stochastic geometry. They had been interested in properties of randomly generated polyhedra such as the average number of vertices, of faces, of facets etc.

But these investigations did not lead to results on the average number of pivot steps since it was not possible to develop an easy characterization of those vertices located on the Simplex-Path.

2. Borgwardt's approach

The first theoretical probabilistic analysis of such a variant was done by the author. He observed that the so-called shadow-vertex-algorithm (a parametric version of the Simplex-Algorithm) permitted such a simple characterization. The advantage was that a simple geometrical criterion for the inclusion of a vertex into the Simplex-Path could be given. This criterion could be used for the evaluation of expectation values.
Here is a brief description of the variant.
Suppose that Phase I is done and that a start vertex x_o is available and that certain nondegeneracy conditions are satisfied (an almost sure event in our stochastic model). Then there is convex cone $U \in \mathbb{R}^n$ such that for all $u \in U$ the linear function $u^T x$

is maximized in x_o. Now choose such an element $u \in U$, $u \neq 0$.
Project the polyhedron X orthogonally on the two-dimensional plane span (u,v). The result is a two-dimensional polyhedron $\Gamma(X)$. The vertices of $\Gamma(X)$ are images of certain vertices of X. Those vertices of X which are mapped into vertices of $\Gamma(X)$ are called shadow-vertices. x_o is such a shadow vertex and we can construct a path of shadow-vertices according to the requirements of Phase II leading to x_s. This construction is done by use of the shadow vertex algorithm.
We know that the number s cannot be greater than the number of shadow vertices S. So we are allowed to concentrate on that random variable. (Under the stochastic model given below we know that $E_{m,n}(s) = 1/4\, E_{m,n}(S)$).
For doing our probabilistic analysis we have to fix a stochastic model

Rotation-Symmetry-Model (RSM)

We assume that the vectors $a_1,\ldots,a_m$ are distributed independently, identically and symmetrically under rotations on $\mathbb{R}^n \setminus \{0\}$ and that $b = (1,\ldots,1)^T \in \mathbb{R}^m$.
The latter normalization and restriction guarantees that only feasible problem-instances occur (i.e. $X \neq \emptyset$), because $0 \in X$.
For this class of instances the only candidates for being vertices on the Simplex-Path are the solutions of the systems of equations

$$a_{\Delta^1}{}^T x = 1, \ldots, a_{\Delta^n}{}^T x = 1 \qquad \text{where } \Delta = \{\Delta^1,\ldots,\Delta^n\} \subset \{1,\ldots,m\} \text{ is an n-element set.}$$

We call the solution points x_Δ. Note that there are $\binom{m}{n}$ such points.

Now we make use of the following

Lemma

1) x_Δ is a vertex of X iff the hyperplane through the points $a_{\Delta^1},\ldots,a_{\Delta^n}$ is a supporting hyperplane for the convex hull of the set $(0,a_1,\ldots a_m)$

2) Let x_Δ be a vertex of X. Then it is even a shadow-vertex iff span(u,v) intersects the convex cone spanned by the vectors $a_{\Delta^1},\ldots,a_{\Delta^n}$ not only in the origin.

So we obtain an integral expression for the number of shadow vertices.

$$E_{m,n}(S) = \binom{m}{n} \int_{\mathbb{R}^n} \ldots \int_{\mathbb{R}^n} P(\{a_1,\ldots,a_n\} \text{ satisfies both the supporting and the intersection condition}) \, dF(a_1)\ldots dF(a_n)\, dF(v) \; .$$

with F denoting the distribution under consideration.
The evaluation of that expression turns out to be complicated. But the use of several integration tricks eventually leads to useful results.

Note that conditions 1 and 2 have the following effect on the probability for being a shadow vertex. Because of rotational symmetry the probability for 1 depends only on the distance of the hyperplane from the origin. The higher the distance - the higher is the probability. Condition 2 depends on the spherical measure of the respective cone. The greater the spherical measure - the greater the probability.

The first results on $E_{m,n}(S)$ described the asymptotic behaviour, i.e. they hold for $m \to \infty$ while n is fixed. For three special a_i - distributions according to the RSM we obtain the following growth rates.

Theorem 1 [Borgwardt 1977]
For $m \to \infty$ and fixed n the expected value of shadow vertices grows proportionally to

1) $\sqrt{\ln m}\ n^{3/2}$ under Gaussian distribution

2) $m^{\frac{1}{n+1}}\ n^2$ under uniform distribution on the unit ball on $\mathbb{R}^n$

3) $m^{\frac{1}{n-1}}\ n^2$ under uniform distribution on the unit sphere of $\mathbb{R}^n$

The asymptotic case is easier to deal with because condition 1 becomes extremely unlikely for small distances between hyperplane and origin. This enables us to exclude those cases in our integration formula. But the remaining cases allow rather sharp approximations of the spherical measures needed for condition 2.
The asymptotic results could be generalized in the following way.

Theorem 2 [Borgwardt 1982a]
For $m \to \infty$ and fixed n we have for distributions according to RSM:

1) $E_{m,n}(S) \leq m^{\frac{1}{n-1}}\ n^2\ \sqrt{2\pi}$ and $E_{m,n}(S) \to \infty$
if the support of the distribution is bounded.

2) There are such distributions (with unbounded support) with
$E_{m,n}(S) \leq C(n)$,i.e. the expected value remains bounded by a value independent upon m .

The reason for the observed discrepancies lies in the different redundancy rates.

Definition
A restriction $a_i^T x \leq 1$ is called redundant for X , iff $a_i^T x < 1$ for all $x \in X$. The redundancy rate is the share of redundant constraints among the total number of constraints m .
An equivalent criterion for redundancy is

The restriction is redundant iff a_i is located in the interior of the convex hull of the points $(0, a_1, \ldots, a_m)$.

Note that the redundancy rate is 0 for uniform distribution on the unit sphere. On the other side we can force the redundancy rate to be arbitrarily close to 1 by choosing a distribution like

$P(\|a_i\| \leq \gamma) = 1 - \frac{1}{\gamma^k}$ with k sufficiently great

High redundancy rates yield a small number of constraints forming the feasibility region. So we obtain only few vertices and short Simplex-Paths.

Later we succeeded in deriving corresponding results without use of the advantages of the asymptotic case and in proving the polynomiality in both parameters m and n .

Theorem 3 [Borgwardt 1982b]
For all pairs (m,n) with $m \geq n$ and all RSM-distributions we have

$$E_{m,n}(s) \leq m^{\frac{1}{n-1}} n^3 e\pi(\tfrac{\pi}{2}+\tfrac{1}{e}) \quad .$$

For solving the complete problem including Phase I we constructed a new algorithm which is based on application of the shadow-vertex algorithm n-1-times in increasing dimensions. This algorithm could be analyzed, too, and allowed the following result.

Theorem 4 [Borgwardt 1982 b]
For all pairs (m,n) with $m \geq n$ and all RSM-distributions the total number s^t of pivot steps for solving the problem completey satisfies

$$E_{m,n}(s^t) \leq m^{\frac{1}{n-1}} (n+1)^4 \frac{2}{5} \pi (1 + \frac{e\pi}{2}) \quad .$$

3. Smale's approach

Steve Smale [Smale 1982, 1983] tried a different approach. He deals with problems of the type

Maximize $v^T x$
subject to $a_1{}^T x \leq b^1, \ldots, a_m x \leq b^m$ and $x \geq 0$
where $v,x,a_1,\ldots a_m \in \mathbb{R}^n$, $b \in \mathbb{R}^m$

His stochastic model demands that

1) The distribution of the random variables A,b,v is absolutely continuous.
2) The vectors b and v and the columns of A are independent random vectors.
3) The probability measure of A and b is invariant under permutations of coordinates in single columns of A or in b .

Smale imbeds the LP into the more general Linear Complementarity Problem which specializes to:

For a given matrix $M = \begin{bmatrix} 0 & -A \\ A^T & 0 \end{bmatrix}$ and for a given vector $q = \begin{bmatrix} b \\ v \end{bmatrix}$

find vectors $w \geq 0$ and $z \geq 0$ $(\in \mathbb{R}^{n+m})$ such that $w-Mz=q$ and that $w^T z = 0$.

These problems are solved by applications of the self-dual algorithm of Lemke. Here one tries to represent the vectors

$q_\lambda = \lambda q + (1-\lambda)e$ with $e \in \mathbb{R}^{n+m}$ for growing $\lambda \in [0,1]$

in the form $q_\lambda = w - Mz$. One starts with $w = e$ for $\lambda = 0$. $(e = (1,\ldots,1^T)$.

During that procedure the index - set of positive entries in w may change and every such change requires a pivot step in the computation.
The question for the number of pivot steps amounts to the question for the number of certain cones (spanned by unit vectors and column vectors of -M) which are intersected by [e,q] .

Note that this geometrical concept is closely related to the geometrical question in Borgwardt' s analysis. This is a result of the use of parametric variants in both studies.

The question for the number of cones gets simpler when dominance events (one column not less then another in every entry) occur very often. And this becomes very likely for $m >> n$ because then M is rather sparse.
Using estimations concerning the spherical measures of such cones and of combinatorial sums, Smale obtains

Theorem 5 [Smale]
For fixed n and m tending to infinity
$E_{m,n}(s^L) \leq C(n)\ (1+\ \ln\ (m+1))^{n(n+1)}$.

Since C(n) may be exponential in n , we have polynomiality only in m .
Charles [Blair] showed that a slightly worse bound could under similar assumptions be derived for the number of existing vertices (without analysis of the algorithm).
This shows that the choice of the stochastic model alone may have a dramatic effect on the results of an analysis.

3.The Sign-Invariance Model

A different stochastic model, first proposed by [May and Smith] lead to much more optimistic results.

Consider again the problem type

Maximize $v^T x$

subject to $a_1{}^T x \leq b^1, \ldots, a_m{}^T x \leq b^m$

where $v, x, a_1, \ldots, a_m \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $m \geq n$.

Besides nondegeneracy no other assumptions on the data a_i ,b,v are made. Suppose that the input data $v,a_1...,a_m$,b are fixed. Now flip a coin for i=1,...,m (i.e. m times) to decide whether the restriction

$a_i^T x \leq b^i$ or $a_i^T x \geq b^i$

shall be included in the definition of the feasibility region. So we obtain 2^m different problem-instances each of same probability out of one set of input vectors. Averaging over those instances yields expectation values on the number of pivot steps.
[Haimovich] and [Adler]have analyzed the behaviour of the same parametric Phase II variant as Borgwardt's but under this different stochastic model. More precisely: They study the length of cooptimal paths. A point x is called cooptimal with respect to two objectives $u^T x$ and $v^T x$, if for fixed value of $v^T x$ the value $u^T x$ cannot be improved on X .
Since we have to deal with a finite number of instances, the mean value can be calculated by applications of combinational formulas.
Every instance of the 2^m creates a cell (its feasibility region). Those cells may be empty. A rather simple proof shows that the expected number of steps under the conditions that a cooptimal path exists, is rather low.
For that purpose we count the number of optimal vertices per cell (at most one because of nondegeneracy), the maximal number of a cooptimal paths per cell (one), the number of cells with cooptimal paths $(\binom{m}{n} + \binom{m}{n-1})$.
And we take into regard that each of the $\binom{m}{n}$ basic solution-points is optimal in exactly one cell. So we conclude that the number of edges to be traversed until every problem is solved is $\binom{m}{n-1}(m-n+2)$.
Division by the number of cells with cooptimal path yields:

Theorem 6 [Adler, Haimovich]
Under the sign-invariance model we have
$E_{m,n}(s|$ cooptimal path nonempty$) \leq n \frac{m-n+2}{m+1}$

This algorithm can be extended to an algorithm for Phase I and Phase II in one. The result of that extension is a Constraint - by - Constraint algorithm which is similar to the dual of Borgwardt 's algorithm.
The idea is as follows. Define the feasibility regions for $n \leq k \leq m$
$X_k = \{x|a_1^T x \leq 1,...,a_k^T x \leq 1\}$.

Initial stage: Calculate x_0 which solves $a_1^T x = 1,...,a_n^T x = 1$
Choose a vector $u \neq 0$ such that x_0 is optimal
with respect to $u^T x$ on X_n . Set $k = 1$.

Typical stage:

1) Given the point x_{n+k-1} which is optimal on X_{n+k-1} with respect to $u^T x$, check whether x_{n+k-1} satisfies $a_{n+k}^T x \leq b^{n+k}$. If yes set $x_{n+k} = x_{n+k-1}$. Go to 4) .

2) If not, use the parametric variant to decrease $a_{n+k}{}^T x$ until $a_{n+k}{}^T x \leq b_{n+k}$. If this is impossible STOP.

3) If a certain pivot step leads to satisfaction of the restriction, the corresponding edge contains a point with $a_{n+k}{}^T x = b^{n+k}$. Call that point x_{n+k} .

4) If $k < m-n$ set $k = k+1$ and go to 1).

Last stage: Start with x_m and apply the parametric Simplex variant to improve $v^T x$. STOP with the optimal point or with the information that an optimal point does not exist.

Summation over all stages and steps yields the following result which was derived independently and under different notation in three papers

Theorem 7 [Adler/ Karp/Shamir 1983 b] [Adler/Megiddo] [Todd]
Under the sign-invariance-model the average number of steps for the complete algorithm satisfies
$E_{m,n}(s^t) = 0(n^2)$.

This result looks sharper than the results mentioned above. But we must take care of the following facts. Only a small share of the instance set has a nonempty cell. This holds in particular when $m >> n$.
The average number of vertices per cell is rather low. Both figures tend to 0 (rather quickly) when $m \to \infty$ and when n remains fixed. Even conditioning on nonempty cells leads to the moderate average number of 2^n vertices per cell. And the redundancy rates (both for the unconditional case and for the case where feasibility is demanded) tend to 1 very quickly. So this model seems to admit reliable comparisons with Borgwardt's hard models (e.g. uniform distributions on unit sphere) only if $m \approx n$.
For $m >> n$ the properties of the sign-invariance model seem to have similarities with the harmless examples in Borgwardt's model (compare Theorem 2 part 2) because of high infeasibility chance and high redundancy rate. Besides that, that analysis seems to reflect properties of the model rather than the efficiency of the algorithm (compare [Adler, Karp, Shamir 1983 b]).
So a lot of questions arise:

How can we recognize the "real-world-distribution"?
Can we compare different results on different stochastic models?
Are the results so sensitive with respect to the distributions, that probabilistic analysis does not make sense at all?

These and other questions arise also for other problems and algorithms. We shall come back to that issue in chapter IV. In the following chapter we want to show some typical examples of probabilistic analysis of algorithms for combinatorial optimization.

III. Some Examples from Combinatorial Optimization
1. Knapsack and Related Problems

One of the most famous problems of integer resp. combinatorial optimization is the knapsack problem. Its most illustrative formulation is the following:
A man plans to make a long-distance walk and he wants to take some items (objects) with him. But he cannot carry more than a certain amount of weight. Every item under consideration has a specific weight and a specific value for the walker. Now the walker wants to maximize the value of his bag content under the restriction that the bag must not be overpacked.
The formal description is
Let each of n items $j = 1,2,...,n$ require a certain amount of weight a_j . Let c_j denote the value or profit of including item j . Let x_j be the knapsack-indicator, i.e. $x_j = 1$ if item j is included, $x_j = 0$ else.

Then we have the following integer optimization problem

$$\text{Maximize } \phi(x) = \sum_{j=1}^{n} c_j x_j$$

$$\text{subject to } \sum_{j=1}^{n} a_j x_j \leq b \text{ and } x_j \in \{0,1\} \text{ for } j = 1,...,n \ .$$

[Lifschitz] carried out a time-analysis of an exact enumerative method for solving that problem.
First he introduces the partial order $\lesssim$ defined on $\{1,...,n\}$.
Here $i \lesssim j$ if $a_i > a_j$ and $c_i \leq cj$
or $a_i \geq a_j$ and $c_i < c_j$
It can easily be seen that a vector $x = (x_1,...,x_n)^T \in \{0,1\}^n$ cannot be optimal if there is a pair (i,j) such that $x_i = 1$ and $x_j = 0$ whereas $i \lesssim j$.

Points with such a property can be excluded from the set of 2^n optimality-candidates. Denote the remaining subset of $\{0,1\}^n$ by B . An enumerative procedure on B for finding the optimum will require $0(\#(B))$ time.

Lifschitz introduces the following stochastic model
Let $a_j, c_j (j=1,...,n)$ be independent, identically distributed, nonnegative variables with density functions.
Now he calculates the mean value of $\#(B)$.

Theorem 8
Under Lifschitz's stochastic model we have

$$E(\#(B)) = 0(e^{2\sqrt{n}}) \ .$$

The result is still exponential, but much smaller than the former upper bound 2^n .

We should notice that this is again a property due to the stochastic model, while a very inefficient optimum-search-algorithm is used. So this analysis does not help very much

for comparing algorithms.

[Meanti, Rinnooy Kan, Stougie and Vercellis] made a probabilistic analysis of the Multi-Knapsack-problem.

Here every item requires space or wastes resources in several dimensions and for every dimension a limit is given.

So the problem is
Given the values $c_j (j = 1,...,n)$ and the dimension weights a_{ij} $(i=1,...,m;\ j=1,...,n)$

$$\text{Maximize } \phi(x) = \sum_{j=1}^{n} a_{ij} x_j \leq b_i \text{ for } i = 1,...,m$$

$$\text{subject to } \sum_{j=1}^{n} a_{ij} x_j \leq b_i \text{ for } i = 1,...,m$$

$$\text{and } x_j \in \{0,1\} \text{ for } j = 1,...,n \ .$$

Here the authors are mainly interested in the behaviour of the optimal value of $\phi(x)$ in dependency upon the values b_i , and not in a time analysis.

For that purpose they analyze the following stochastic model. Let the c_j be random variables which are independently and identically distributed over a bounded nonnegative interval of IR . Let the coefficients a_{ij} also be independently and identically distributed over [0,1] (without loss of generality). Let all these distributions be absolutely continuous. And let for all $i = 1,...,m$ $b_i = n\beta_i$ with $0 < \beta_i < E[a_{ij}]$.

Now the authors prove that the random variable $1/n\ Z_n$,where Z_n is the optimal value, converges with probability one towards a value $L(\lambda^*)$ which is itself an (almost surely) limit of a sequence $L_n(\lambda_n^*)$ defined as follows

$$L_n(\lambda) = \sum_{i=1}^{m} \lambda_i \beta_i + \frac{1}{n} \sum_{j=1}^{n} [c_j - \sum_{i=1}^{m} \lambda_i a_{ij}] x_j^L(\lambda)$$

where λ_n^* is the multiplier vector $(\lambda_1,...\lambda_n)^T$ minimizing $L_n(\lambda)$ over $\{\lambda | \lambda \geq 0\}$ and where

$$x_j^L(\lambda) = \begin{cases} 1 & \text{if } [c_j - \sum_{i=1}^{m} \lambda_i a_{ij}] \geq 0 \\ 0 & \text{else} \end{cases}$$

Actually, $L(\lambda^*)$ is a function of the b_i's, which is implicitly defined by minimization, but difficult to describe in a closed-form expression. $L_n(\lambda)$ has the same valuable

properties,which support our ability to find the minimum. A second result says that $\lambda_n^* \to \lambda^*$ for $n \to \infty$ almost surely and that $x^L(\lambda^*)$ is infinitely often optimal when n tends to infinity.
The tools for these derivations are Lagrangean-multiplier-theory, law of large numbers and convex analysis. Besides that the authors give an approximation algorithm belonging to the family of generalized greedy heuristics. Here the items are ordered according to decreasing ratios

$$c_j \Big[\sum_{i=1}^{m} y_i \, a_{ij} \Big]^{-1} .$$

where y denotes weights and should be chosen as $y = \lambda^*$ (which had been calculated by numerical minimization). Using this choice the heuristic yields asymptotical optimality (for $n \to \infty$) almost surely.

[Tinhofer] deals with the Subset-Sum-Problem. Here we have $a_j = c_j$ for $j = 1,...,n$ and our aim is to fill as much weight into the Knapsack as possible or to close the gap between weight of the load and upper bound. So we have to

$$\text{Maximize } \sum_{j=1}^{n} a_j x_j$$

$$\text{subject to } \sum_{j=1}^{n} a_j x_j \le b \text{ and } x_j \in \{0,1\} \text{ for } j \in \{1,...,n\} .$$

In his stochastic model the values $a_j (j=1,...,n)$ are independently, identically and uniformly distributed over $\{1,2,...,d(n)\}$, where d depends on n.
The right hand bound may be independent of the a_i's and uniformly distributed over $\{1,2,...,nd(n)\}$.

So a set of $[n\, d(n)][d(n)]^n$ problem-instances is defined and he averages over these instances.

His greedy-algorithm works as follows

Initial Step: Set SUM = 0 and i = 1

Typical Step: 1) Set $x_i = 0$
2) If SUM + $a_i \le b$ then SUM = SUM + a_i, $x_i = 1$.
3) if $i < n$ set $i = i+1$, go to 1)
4) STOP

Clearly, the actual optimal value $\overline{SUM}$ is not greater then b .
The algorithm delivers an output SUM $\le \overline{SUM}$. So the gap $\overline{SUM}$ - SUM has the upper bound b - SUM. And we try to analyze the distributions of b - SUM as an upper bound variable for the error made by the algorithm.

Stochastically, the algorithm proceeds as a Markoff-process. The gap will be reduced with certain probabilities.

The initial state of the gap has the following distribution

$$P_0(\text{gap}=k) = \begin{cases} 0 & \text{for } k = 0 \\ \frac{1}{nd(n)} & \text{for } 1 \leq k \leq n-d(n) \end{cases}$$

The probability for a change from gap = k to gap = ℓ ($\ell \leq k$) (the transition probability) has the following value:

$$\begin{cases} 0 & \text{for } \ell > k \\ \frac{\text{Max}\{d(n) - k, 0\}}{d(n)} & \text{for } \ell = k \\ \frac{1}{d(n)} & \text{for Max}\{k-d(n), 0\} \leq \ell \leq k-1 \\ 0 & \text{for } 0 \leq \ell < \text{Max}\{k-d(n), 0\} \end{cases}$$

So we are able to calculate the transition matrix of the process and the distribution of the variable gap after i = 1,...,n steps. The n-th distribution is the final one.

Result: After step n we have a probability for gap = 0 of

$$\frac{1}{2} + \frac{1}{d(n)} - \frac{1}{nd(n)} \binom{d(n)}{2} \left(1 - \left(\frac{d(n)-1}{d(n)}\right)^n\right)$$

This implies that for $n \to \infty$, $d(n) \to \infty$ and $d(n) = o(n)$ the algorithm ends up at gap = 0 with probability 1/2 .

Theorem 9

Under the condition that $\sum_{j=1}^{n} a_j \geq b$

(which has probability $\frac{1}{2}(1+\frac{1}{n})$) we know that asymptotically

$$P(\text{gap becomes } 0) = \frac{\frac{1}{2}}{\frac{1}{2}(1+\frac{1}{n})} = 1 - \frac{1}{n+1} = 1 - o(\tfrac{1}{n}) \quad .$$

This follows because the complementary event $\sum_{j=1}^{n} a_j < b$ excludes gap = 0 .

The following modified algorithm GOLOSONE was analyzed by [D'Atri and Puech]. The modification lies in the fact that the items are first ordered according to their weight (in decreasing order). After that we start the greedy-algorithm as above. But as soon as SUM + $a_i > b$ is reached we admit at most one more item, i.e. the greatest one which still fits into the gap. Afterwards we stop.

Theorem 10 [D'Atri]

For $d(n) = o(n)$ we have

P(GOLOSONE solves the subset - sum problem exactly) $\to 1$ for $n \to \infty$

where $P(\text{gap} = 0) = 1 - \frac{1}{2} e^{\frac{-n}{d(n)}}$.

Remark
A GOLOSONE-version without preordering has the same limit, but slower convergence $P(\text{gap} = 0) \sim 1 - 1/2 \frac{d(n)}{n}$.

For the Knapsack-Problem we know $O(nb) = O(n^2 d(n))$ running time exact solution algorithms, but they are not polynomial in the encoding length of b !
In contrast, Knapsack and even Subset-Sum-Problems are INIP-hard.

§ 2 Scheduling problems

Of great importance for production and service-planning is the machine-scheduling-problem.
The definition is as follows:

> n jobs $J_1,...,J_n$ have to be processed by one machine which cannot handle more than one job at a time. Preemptions (i.e. interruptions) of jobs are not allowed. Job J_i arrives at time γ_i ($\gamma_1 \leq \gamma_2 \ldots \leq \gamma_n$) and requires $p_i \geq 0$ time-units for processing. The task is: Schedule the starting times for the jobs in such a way that the sum of completion times (completion time = waiting time + processing time) is minimized. The data of the problem instance are known in advance.

This problem is known to be INIP-hard .

[Gazmuri] makes a probabilistic analysis for certain algorithms under the following stochastic model.

i) the processing times p_i are identically, independently distributed integer variables bounded from above by a constant M.
ii) the distances between arrival times $t_i = \gamma_{i+1} - \gamma_i$ are identically, independently distributed integer variables with a moment generating function $E(e^{\tau t})$ which becomes finite for at least one $\tau < 0$.
III) the random variables $p_1,...,p_n,t_1,...,t_n$ are independent.

Gazmuri uses different algorithms for the respective cases

a) $E(p) < E(t)$.This is the undersaturated case. It must be expected that sometimes the machine will be idle because no job is present.
b) $E(p) > E(t)$. This is the oversaturated case. The jobs arrive (on the average) faster than they can be operated. Here jobs have to be postponed.

For the undersaturated case the algorithm partitions the set of jobs into smaller sets whose size is $a = \lfloor \ln n \rfloor$. These subsets are in a first stage independently scheduled in an optimal way. This can be done by enumerative methods. The resulting list of starting times for the total set of jobs may be infeasible, because jobs from different subsets may

be working at the same time. To transform the list into a feasible one we define so-called "breaking points" c between 0 and a . For $0 < i \leq c$ the schedule for the first subset is taken. Then we take into regard all jobs (from all sublists) which are present at the end of job c . They are ordered according to the "shortest job first"-rule and put onto our final list. Then we take the remaining jobs according to the "shortest job first rule" until the machine is (the first time again) clear of jobs. After that we repeat this process starting with the next group of a jobs.

In the oversaturated case one starts by computing an optimal preemptive solution (which would allow interruptions) by use of the Shortest Remaining Time-Rule. After that one takes the first started but interrupted job and collects all the remaining parts. They are put behind the remaining part of the first job. Now at least the first job runs without interruption. After that we apply the same procedure to the next interrupted job.

Theorem 11

Both algorithms are asymptotically optimal. The relative error tends to 0 for $n \to \infty$ with probability tending to 1.

In the first case the relative error will be at most $0[(\ln \ln n)^{-\alpha}]$ for an α with $0 < \alpha < 1$ with probability $1-0[(\ln \ln n)^{-\alpha}]$

In the second, a relative error of at most $0(1/n)$ has probability of more than $1-0(1/n)$.

Now we turn to a related problem, the parallel processor scheduling problem.

Definition

Let n jobs $J_1,...J_n$ be present at the beginning and let m parallel working machines be available. Let the processing tiimes be $p_1,...,p_n \in [0,\infty)$.
The task is: Minimize the completion time for the last finished job by assigning every job to one of the machines.

This problem is also known in a more abstract formulation; the Partitioning Problem:
Given a list of n nonnegative numbers $p_1,...,p_n$, partition this set in m sublists such that the length of the largest sublist is minimized.

This problem is also known to be INIP-hard.

[Loulou] studies two heuristics for the solution:

a) the LPT-heuristic (Longest Processing Time):
Sort the jobs in decreasing order with respect to p_i . Assign the (next) occuring job to the currently smallest loaded machine (for i=1,...,n).

b) the RLP-heuristic (Random List Processing):

Assign in the same way as above but avoid sorting.

His stochastic model requires that we have an infinite list of random variables $X_1,\ldots,X_n,\ldots$ which are all nonnegative, independently and identically distributed with finite mean. Regard the first n of those variables as processing times p_i .

Theorem 12 [Loulou]

1) The total running times are

$$C_1 \sum_{i=1}^{n} \log X_i + C_2\, n \log n + C_3\, n \log m \quad \text{for LPT}$$

$$C_1 \sum_{i=1}^{n} \log X_i + n\, C_2 \log m \quad \text{for RLP} .$$

2) for LPT and $m \geq 2$: there is a finite random variable Z which satisfies the following inequality almost surely

$$\text{LPT-solution} - \text{optimal solution} \leq Z < \infty \quad \forall n .$$

3) for RLP and $m > 2$: The absolute error is stochastically bounded by a finite random variable Z , which is independent of n , i.e.

$$P(\text{RLP-solution} - \text{optimal solution} > \alpha) \leq P(Z > \alpha)\ \forall\, \alpha \geq 0 \text{ and for all } n .$$

These results imply that the relative errors tend to 0 for $n \to \infty$.

3. Allocation and Assignment problems

For scheduling the locations for branches of a company it is very important to determine the optimal assignment of branches to the elements of a given set of places. Our first problem occurs when we try to distribute k warehouses or stores to serve n cities. We assume that all cities are equally important.
This problem can be formalized as follows:

Definition

Let n points $X_1,\ldots,X_n$ be given in the plane. Select k of these points to serve as centers. Do the selection in such a way that the sum of the distances from the n points to their closest center is minimized.

This means: Minimize $Z_n(S)$ where

$$\#(S) = k$$
$$S \subset \{1,...,n\} \text{ and } Z_n(S) = \sum_{i=1}^{n} \min_{j \in S} d_{ij}$$

and where d_{ij} is the distance from point X_i to point X_j .

For that problem we do not know a polynomial algorithm. [Fisher and Hochbaum] made a probabilistic analysis under the following stochastic model:
The n points $X_1,..,X_n$ are distributed uniformly and independently over a square region R in the plane.
In general, we assume that $k = o(n)$.
The authors derive bounds on the optimal value $Z(n,k)$

Theorem 13 [Fischer and Hochbaum]

1) For any $\beta < \frac{\sqrt{2}}{e\sqrt{\pi}}$ almost everywhere

$Z(n,k) \geq \beta(n,k) \sqrt{\frac{A(R)}{k}}$, where $A(R)$ is the area of R .

2) If R is a square then for $\beta < \frac{\sqrt{2}}{e\sqrt{\pi}}$ almost everywhere

$$\beta\left[\frac{(n-k)\sqrt{A(R)}}{\sqrt{k}}\right] \leq Z(n,k) \leq \sqrt{2}\left[\frac{(n-k)\sqrt{A(R)}}{\lfloor\sqrt{k}\rfloor}\right] .$$

The heuristic analyzed by Fisher and Hochbaum for the square works as follows:
R is partitioned into t^2 identical subsquares $(R_1,...,R_{t^2})$.
Let $d(R_i,R_j)$ be the distance between the sets R_i and R_j .
Let N_i be the subset of $\{X_1,...X_n\}$ belonging to R_i . And let $H = \{1 \leq i \leq t^2 | N_i \neq \emptyset\}$ be an indexset.

Now we calculate - by enumeration - the value

$$Z_n^A = \min_{\substack{J \subset H \\ \#(J) \leq k}} \left\{ \sum_{i \in H} \min_{j \in J} \#(N_i) \cdot d(R_i, R_j) \right\}$$

This is a non-Euclidean problem with at most t^2 points. Let the solution set be $\bar{J}$. Select from every N_j with $j \in \bar{J}$ one of the points to obtain a set S_A . (If necessary, augment it with arbitrary additional points to achieve $\#(S_A) = k$) .

It is clear that $Z_n^A \leq Z(n,k) \leq Z_n(S_A)$. And the error can be bounded as follows

$$Z_n(S_A) - Z(n,k) \leq 2\sqrt{2A(R)}\,((n-k)/t) \ .$$

Also, the running time for achieving a relative accuracy of at most ε is studied:

Theorem 14

ε-optimality can be achieved almost everywhere when ε is fixed .

i) if $k \leq \ln n$ in polynomial running time

ii) if $k \leq \ln \ln n$ in linear time.

A precise estimation leads to running time of

$$O\left[\left(\frac{1}{\varepsilon^2}\right)(\ln n)^{\frac{3}{2}}\, n^{1+2\ln\frac{1}{\varepsilon}}\right] \text{ if } k \leq \ln n \ .$$

Another extremely difficult problem arises when n locations $X_1,\ldots,X_n$ for n plants are available and when the task is to minimize the transportation costs of material or products between the plants. Suppose that the management knows the amount c_{ij} of material which has to be transported from plant i to plant j on a normal working day and that the transportation cost is $c_{ij}\,d_{ij}$, where d_{ij} is the "distance" between X_i and X_j . Now the question is: How shall the plants be assigned to the locations in order to minimize the sum of transportation costs. This gives the Quadratic Assignment Problem.

Definition

Given two real $n \times n$ - matrices $C = (c_{ij})$ and $D = (d_{ij})$, let $\sigma \in S_n$ be a permutation of $\{1,\ldots,n\}$.
The quadratic assignment problem is the task to

$$\text{Minimize } Q(\sigma,n) = \sum_{\sigma\in S_n} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, d_{\sigma(i)\sigma(j)} \ .$$

We introduce the additional notation:

σ_* is the permutation leading to $\min_{\sigma \in S_n} \{Q(\sigma,n)\}$

σ^* " " $\max_{\sigma \in S_n} \{Q(\sigma,n)\}$.

[Burkard and Fincke (1983)] derived surprising relations between $Q(\sigma^*,n)$ and $Q(\sigma_*,n)$ under certain stochastic models. In a first variant, the values d_{ij} are defined as the distances between points X_i and X_j, e.g.: $||X_i - X_j||_2$. So we obtain the "planar quadratic assignment problem". The stochastic assumptions are the following:
Let the $X_i (i=1,...,n)$ be independently and uniformly distributed over the square $[0,1] \times [0,1]$ and let the c_{ij} be identically and independently distributed over $[0,1]$ with mean value $c > 0$.

The authors found out that the relative distance between worst and best solution can (with probability tending to 1) be bounded from above by a function itself tending to 0.

Theorem 15

Under the above-mentioned stochastic model we have for any $\alpha > 0$

$$P\left\{\frac{Q(\sigma^*,n) - Q(\sigma_*,n)}{Q(\sigma_*,n)} \leq \frac{3+\alpha}{c}\, n^{-0,18}\right\} \underset{n\to\infty}{\to} 1 \quad .$$

[Frenk, Houweninge and Rinnooy Kan] weaken the assumption on the range of c_{ij} (which is now distributed over $(0,\infty)$) and demand that there exists a neighbourhood of 0 such that for all λ belonging to that neighbourhood

$$E(e^{-\lambda c_{ij}}) \quad \text{is finite.}$$

Theorem 16 [Frenk, Houweninge and Rinnooy Kan]

There are constants f and M such that for $\sigma = \sigma_*$ and $\sigma = \sigma^*$

$$\limsup_{n\to\infty} \frac{n^{\frac{1}{4}}}{\ln n} \left| \frac{Q(\sigma,n)}{f \cdot c \cdot n^2} - 1 \right| < M \qquad \text{almost everywhere}$$

If X and Y are idependently and uniformly distributed points in the unit square, then $f = E\,||X-Y||$.
Note that we have a sharper version of convergence here.

In both papers the approach is similar. The square is partioned into a certain number m(n) of congruent subsquares $S_1,...,S_{m^2}$. Fundamental theorems of probability theory (Chebyshev) show that such a subsquare contains about $\frac{n}{m^2}$ points X_i.

The probabilities for the bad cases (where there are much less or much more points in a subsquare) can be aggregated and estimated. They dissappear for $n \to \infty$. In the

same way we treat the sums of C-columns, which are in the regular cases about nc . In the good cases we replace the point X_i by the center of its subsquare, calculate the QAP-value for the simplified situation and estimate the error we have made.
Both papers contain also an analysis of the general QAP. Frenk et al. work under the assumption:

c_{ij} and d_{ij} are identically and independently distributed on $(0,\infty)$ with $E(c_{ij}\, d_{ij}) = d > 0$ and in a neighbourhood of 0 $E(e^{-\lambda c_{ij}\, d_{ij}}) < \infty$.

Theorem 17 [Frenk, Houweninge and Rinnooy Kan]

There is a constant M such that almost everywhere for $\sigma = \sigma_*$ and $\sigma = \sigma^*$

$$\limsup_{n\to\infty} \frac{n^{\frac{1}{2}}}{\sqrt{\ln n}} \left| \frac{Q(\sigma,n)}{d\, n^2} - 1 \right| \leq M \ .$$

Under a bit weaker assumptions but with $c_{ij}, d_{ij} \in [0,1]$ Burkhard and Fincke obtain again stochastic convergence.

Theorem 18 [Burkhard and Fincke 1983]

$$P\left\{ \frac{Q(\sigma^*,n) - Q(\sigma_*,n)}{Q(\sigma_*,n)} \leq \frac{2+\alpha}{d \cdot c}\, n^{-0,225} \right\} \underset{n\to\infty}{\to} 1 \quad \forall \alpha > 0 \ .$$

Here the asymptotic is rather weak. It is much better in the case of a closely related problem, the quadratic bottleneck assignment problem. [Burkhard and Fincke 1982] show a similar result for that problem.
The only difference to the problem above is the objective

$$\max_{1\leq i,j\leq n} c_{ij}\, d_{\sigma(i)}\, d_{\sigma(j)}$$

which is to be minimized.
Such a problem arises in practice when we concentrate on the single transportation costs between pairs of branches and not on the sum.

The property in Theorems 15 and 18 turned out to be a special case of the following general result for combinatorial optimization problems.
Consider a combinatorial optimization problem with ground set E_m . Let T_m consist of all feasible subsets of E_m and let S be an arbitrary element of T_m . In addition $c_m : E_m \to \mathbb{R}^+$ is a weight function and $|S|, |T_m|$ denote cardinalities.

[Burkhard and Fincke 1985 show]:

Theorem 19:

If $c_m(e)$, $e \in S$, $S \in T_m$, $n \in IN$ are identically distributed random variables in [0,1] with expected value $E := E(c_m(e))$ and variance $\sigma^2 := \sigma^2(c_m(e)) > 0$,
If for given $\varepsilon > 0$, ε_o fullfills $0 < \varepsilon_o \leq \sigma^2$ and

$c_m(e)$, $e \in S$ are independently distributed for every fixed $S \in T_m$, $m \in IN$,
if all $S \in T_m$ have the same cardinality,
if for $\lambda_o = 2(\varepsilon_o \sigma / (\varepsilon_o + 2\sigma^2))^2$
$\lambda_o |S| - \log |T_m| \to \infty$ as $m \to \infty$

Then

$$P\left\{ \frac{\max\limits_{S \in T_m} \sum\limits_{e \in S} c_m(e)}{\min\limits_{S \in T_m} \sum\limits_{e \in S} c_m(e)} < 1 + \varepsilon \right\} \geq 1 - 2|T_m| \exp(-|S|\lambda_o) \to 1 \quad \text{as} \quad m \to \infty$$

Loosely spoken, this says the following:
If the number of coefficients in the objective (multiplied by a factor λ_o) minus logarithm of the number of feasible solutions tends to infinity for $m \to \infty$, then the relative error tends to 0 in probability:

Whereas the QAP is INIP-hard, the Linear Assignment Problem is rather easy to solve (e.g. as a linear programing problem). For that purpose an $O(n^3)$-running-time algorithm exists.
But here we do not find that type of convergence towards the same limit (in the relative sense).

The problem is:

Given an $n \times n$ - matrix $A = \{a_{ij}\}$. Find a permutation $\sigma_* \in S_n$ such that

$$L(\sigma_*, n) = \sum_{i=1}^{n} a_{i\sigma(i)} \text{ is minimized.}$$

[Karp 1976] has shown that

if the a_{ij} are independent uniformly distributed variables on [0,1] then
$E(L(\sigma_*, n)) < 2$.

But here the sequences

$\frac{E(L(\sigma^*,n))}{n}$ may have different limits. (σ^* = worst solution) .

So we have the ridiculous situation that the easier solvable problem must be handled more carefully when $n \to \infty$.

4. The Traveling Salesman Problem

Perhaps the most famous problem of combinatorial optimization is the Traveling Salesman Problem:
A person has to visit n cities and to return to the starting point. Which order of the visited cities provides the least traveling distance for that tour?

Formal: Given a complete graph (V,E) with vertices $v_1,\ldots,v_n$ and weights c_{ij} attributed to the edges $e_{ij} = (v_i,v_j)$. Consider all cycles $v_{\sigma(1)}\ldots..v_{\sigma(n)}\, v_{\sigma(1)}$ where $\sigma \in S_n$.

Find - among these cycles - that of minimal weight

$$\sum_{i=1}^{n-1} c_{\sigma(i)\sigma(i+1)} + c_{\sigma(n)\sigma(1)} \cdot$$

The problem is INIP - hard. Heuristics are used to obtain approximate solutions. Some of these heuristics show a remarkable good behaviour.
A realistic version of that problem arises when the weights c_{ij} are defined as the distances between points X_i and X_j in the plane. Then we speak of the Euclidean Traveling Salesman Problem (sometimes defined also in higher dimensions).
To study the average behaviour of heuristics one uses stochastic models as follows:

> Let $v_1,\ldots,v_n$ be random points distributed uniformly and independently over the d-dimensional unit cube.

The following questions arise:

> Is it possible to get information on the optimal tour length? Is it possible to analyze reliable heuristics?
> How fast are such heuristics?

The mostly analyzed algorithm is closely related to the divide -and conquer- methods used in section 3.
[Karp, Steele] [Halton and Terada] analyze such a fixed dissection algorithm.
Let s(n) a function of n which is appropriate for the following application.
First stage: Partition the cube into s(n) congruent subcubes Q_i . Consider V_i ,

the subset of V , which is located in Q_i and denote $n_i = \#(V_i)$ for $i=1,\ldots,s(n)$. Calculate an internal optimal tour for every subset V_i by an exact enumerative method (derived from dynamical programming).

Second stage: Choose a representative out of every nonempty subcube and put them into a certain order according to the location of their subcube. To derive such a rule for that order is not very difficult, because we have a set of simply describable points.

Third stage: Connect the subcube-cycles (first stage) according to the order obtained in the second stage, delete excess edges and convert by this way the set of subcube tours into a cycle for the whole cube.

The rule mentioned in stage 2 is based on dividing the unit cube by parallel hyperplanes (orthogonal to the n-th coordinate) into strips of equal width . Then the points X_i of strip k are projected on the hyperplane basic to step k .These points have to be visited according to a rule for dimension $d-1$.

The recursion is initialized with dimension $d = 2$, where we run through the points from left to right in the bottom strip, after that run from right to left in the second strip, from left to right in the third strip etc.

It is well-known that the length of the optimal tours L_n and the length of the constructed tour L^F_n satisfy

$$\lim_{n\to\infty} L_n \, n^{-(d-1)/d} = \lim_{n\to\infty} L^F_n \, n^{-(d-1)/d} = c_d \text{ with probability } 1 \; .$$

Theorem 20 [Karp and Steele]

If $s(n) = \frac{n}{\sigma(n)}$, where $\sigma(n)$ is an unbounded increasing function of n , then

$$\sum_{n-1}^{\infty} P\left(\frac{L_n^F}{L_n} \geq 1 + \varepsilon\right) < \infty \quad \forall \varepsilon > 0 \; .$$

The method for tour construction in the subcubes requires exponential time (bounded by $C(n_i)^2 \, 2^{n_i}$) .

But our distribution-assumptions assure that the n_i do not exceed $\frac{n}{s(n)}$ dramaticallly (with high probability). So the "exponential" time for touring subcubes does not deteriorate the behaviour too much. Stage 2 and Stage 3 can be carried out in moderate time.

Theorem 21 [Karp and Steele]

For the specific choice $\sigma = \frac{n}{s(n)} = \ln n$, the expected execution time of the given

dissection method is $0(n^2 \ln n)$.

[Halton and Terada] use a closely related algorithm and a similar, slightly more general stochastic model. They prove that their algorithm runs in probability in time o[n s(n)] for an arbitrarily chosen function s satisfying $s(n) \to \infty$ and $\frac{s(n)}{n} \to 0$.

In addition, they show that with probability one (i.e. almost surely) the length of the generated cycle is asymptotically equivalent to the optimal length.

A second version of the problem, the Asymmetric Traveling Salesman problem, requires different algorithms and different analization methods. Here $d_{ij} = d_{ji}$ is not necessary.

Now, we want to find the minimal closed directed-tour through the points $v_1,...,v_n$.

The stochastic model of Karp and Steele demands that the weights d_{ij} are drawn independently from the uniform distribution over [0,1]. So we have n^2-n random variables ($d_{ii} = 0$ is assumed).

We cannot rely on the triangle inequality as in the Euclidean case. So it becomes impossible to work with the geometrical methods used above. The heuristic used here is based on first solving the corresponding linear assignment problem. It can be stated as (LAP).

$$\text{Minimize}_{\tau \in S_n} \sum_i d_{i\tau(i)} \quad .$$

Note that τ ranges over all permutations, whereas the feasible permutations for the TSP are required to be acyclic. So the above mentioned LAP is a relaxation of the TSP.

First the assignment problem is solved. This is possible within $0(n^3)$ steps.
Then we consider the graph corresponding to the LAP-solution. A patching algorithm is used to convert that graph into a cycle.
Since the assignment solution consists of the arcs $\{(i,\tau(i)|| \; i=1,2,...,n\}$ and the permutation may be cyclic , the resulting solution may consist of more then one cycle. In that case one connects the two cycles with highest length in the following manner.
Let i_1 and i_2 be the indices of the two cycles C_1 and C_2 . Let $\tau(i_1)$ and $\tau(i_2)$ be the corresponding neighbours. Delete the arcs $i_1\tau(i_1)$ and $i_2\tau(i_2)$ and connect (cross-wise), i.e. i_1 with $\tau(i_2)$ and i_2 with $\tau(i_1)$. So we obtain one cycle out of two, i_1 and i_2 are chosen in such a way that the extension of the tour length becomes minimal. Repeated application on the two longest cycles leads to a tour with length T .

Karp and Steele prove the

Theorem 22 [Karp and Steele]

Let T be the heuristical tour length and T^* be the optimal tour length. Then the above-mentioned stochastic model induces that

$$E[\frac{T - T^*}{T^*}] = 0(n^{-\frac{1}{2}}) \quad .$$

[Frieze] deals with the task to solve the symmetric problem (with $\frac{(n-1)(n-2)}{2}$ free variables) exactly.

He demands that

> the edge-weights are independent random variables, where each weight is chosen uniformly from a set $\{0,1,2,\ldots,B(n)-1\}$ of integers.

Theorem 23 [Frieze]

> If $B(n) = o(\frac{n}{\ln\ln n})$ then there is an algorithm ALGT with polynomial running
>
> time $0(n^3 \ln n)$
>
> such that
>
> $\lim_{n\to\infty} P(\text{ALGT solves the symmetric TSP exactly}) = 1$.

5. Probabilistic analysis of graphs

Many combinatorial optimization problems (compare sections 3 and 4) can be formulated as problems of graph theory. For the analysis of algorithms for such problems it is necessary to study the stochastic behaviour of graphs. Instead of discussing further special problems we want to deal with some properties of randomly generated graphs. They are basic for understanding the performance of algorithms on such graphs. For that purpose we follow the book of [Palmer].

Let again $G = (V,E)$ be a graph with vertex-set V and edge-set E, where $V = \{v_1,\ldots,v_n\}$.
For generating random graphs two stochastic models are very popular.

Model A: Consider the $\binom{n}{2}$ ordered pairs (v_i,v_j) of $V \times V$ with $1 \le i < j \le n$ and the random 0-1-variables $I(i,j)$. Regard them as independently and identically distributed variables with $P(I(i,j) = 1) = p \in [0,1]$.

Now consider the empty graph $G_0 = (V,\varnothing)$ and connect v_i and v_j iff $I(i,j) = 1$.

We can obtain $2^{\binom{n}{2}}$ different (labeled) graphs. The probability of a fixed graph is $p^q(1-p)^{\binom{n}{2}-q}$. And the probability that the resulting graph has q edges is

$$2^{-\binom{n}{2}}\binom{\binom{n}{2}}{q} p^q (1-p)^{\binom{n}{2}-q}$$

Model B: For integers n and q , $(0 \le q \le \binom{n}{2})$ let the sample space consist of all graphs with n vertices and q edges. Such a graph results from choosing - without repetition - q out of $\binom{n}{2}$ ordered pairs (v_i,v_j) and introducing the corresponding edges $e_{ij} = v_iv_j$. Here the probability of a certain (labeled) graph with q edges is $\binom{\binom{n}{2}}{q}^{-1}$.

Asymptotically, model B can often be replaced by model A, which admits easier computations. This is possible by choosing $q = \lfloor p\binom{n}{2} \rfloor$. So we concentrate on model A.

We say that "almost all graphs have a property Q", if
$\lim_{n\to\infty}$ P(Q is satisfied in the sample space of n-vertex-graphs) = 1 .
(This notion corresponds to convergence in probability.

Fixing p and a natural number k yields astonishing results under Model A:

Almost all graphs have diameter 1.
Almost all graphs are k - connected.
Almost all graphs are nonplanar.
Almost all graphs contain a given subgraph of order k as an induced subgraph.

When variations of p (in dependence of n) are allowed, one may ask for certain borders or threshold functions (telling the necessary size of $p(n)$) separating the regions where a property can be guaranteed almost surely or not. That means that we search for a function $q(n)$ such that Q is satisfied almost by all graphs if $p(n) = o(q(n))$.
One of the most interesting results is

Theorem 23

If $p(n) = \frac{\ln n}{n} + \frac{x}{n}$

then we know that

P(number of isolated vertices = k) $\underset{n\to\infty}{\to} \frac{e^{-\mu}\mu^k}{k!}$ for $\mu = e^{-x}$,i.e.

this number is Poisson-distributed.

The following case studies provide such bounds for important properties.

Theorem 24. The following properties hold for almost all graphs

a) If $p(n) = o(\frac{1}{n})$ Then there are no cycles. All components are trees.

b) If $p(n) = o(n^{-\frac{k}{k-1}})$ Then there are no components of order k or greater.

c) If $p(n)\, n^{\frac{k}{k-1}} \underset{n\to\infty}{\to} \infty$ Then every tree of order k is a component.

arbitrarily slowly

d) If $p(n)\, n = c < 1$ Then all components are trees or unicyclic. The largest component is a tree of order about log n .

e) If $p(n)\, n = c = 1$ Then certain components are cycles. The largest has order about $n^{\frac{2}{3}}$.

f) If $p(n) = c > 1$ Then there is a unique giant component. All but $o(n)$ vertices belong to the giant or to trees of order at most ln n .

g) If $p(n) = c\frac{\ln n}{n}$ with $c < 1$ then the graph is disconnected

h) If $p(n) = c\frac{\ln n}{n}$ with $c > 1$ then the graph is connected and Hamiltonian.

These results are based on work of Erdös and Renyi.

The jumps in connectivity and the quality of components have a dramatic effect on certain algorithms which require connectivity for solving the problem. For instance, think of Dijkstra's algorithm for finding the shortest connection between two points in the simple case where all the existing edges have weight 1 and all the nonexisting ∞ . Dijkstra's method runs in $0(n^3)$ steps in the worst case. But for average case analysis we empirically observe the following behaviour:

i) For small p most of the paths constructed from the origin end very early. There is little chance to reach the desired vertex.

ii) For medium p some paths are interrupted (but not early). We get along to the solution on a few paths.

iii) For high p most of the paths reach the goal. The shortest one is not long but a lot of overfluous work has to be done.

So the change in the quality of the graph changes the computational effort to solve the problem dramatically.

Remark

An improved version of Dijkstra's algorithm, developed and analyzed by [Spira] has an average case behaviour of $0(n^2(\log n)^2)$ when the lengths of the arcs (in the digraph) are identically and independently distributed nonnegative random variables.

IV. Open questions and drawbacks

On the first glance, the demonstrated results give a somehow confusing impression. The first general observation may be that most results are of an asymptotic nature. They hold under the assumption that the dimension of the given problem-instances tends to infinity. This has many reasons. Some of those are:

1) The analysis is often done by studying a simplified, regular situation (compare III.3 and III. 4) combined with estimations of the error caused by that simplification. These estimations rely heavily on the fact that an irregular situation does not occur with high probability. But this is true and can only be shown in asymptotic cases, where laws of large numbers or the central limit theorem or similar asymptotic theorems of probability theory apply.
Even in the case of linear programming analysis under the rotational-symmetry-model there is some "regular" behaviour in the asymptotic case. When $m >> n$ then the spherical measure of a cone spanned by $a_1,...,a_n$ will be small (with high probability) under the condition that the convex hull of $a_1,...,a_n$ is on the boundary of $CH(a_1,...a_m,0)$. This results from the need of a high distance between origin and hyperplane in order to provide the boundary condition. But if that distance is high in comparison to the spherical measure, then the latter can be approximated very well by linearization. That means: One replaces the volume of the intersection between unit ball and the cone of $a_1,...,a_n$ by the

volume of the pyramid generated by the points $0, \frac{1}{\|a_1\|} a_1, ..., \frac{1}{\|a_n\|} a_n$. The latter is

much easier to calculate. So also the evaluation of the integral formula is simplified in the asymptotic case.

2) Another reason lies in the formulas which are first derived for the desired moments or probabilities resulting from some characterization of the algorithm. Often we find very complicated sums, series (sometimes alternating) or definite integrals. Usually much more is known about their asymptotic limit then about their value in finite cases (consider integrals of the Beta-Function type with integration area ending between 0 and 1).
For instance, the expected number of components in a randomly generated graph (Model A,III.5, [Palmer]) with edge probability p that are trees of order at least m is

$$\sum_{k=m}^{n} \binom{n}{k} k^{k-2} p^{k-1} (1-p)^{\binom{k}{2} - (k-1) + k(n-k)}$$

For the evaluation of that formula one needs the Theorem of Moivre-Laplace, estimations of tails of binomial sums and Stirling's formula.

Of course, such asymptotical results are of restricted utility for practice. This is even worse, when - as often observed - the requirement of asymptotic is twofold: With probability tending to 1 , the variable is less than a function tending to 0 for $n \to \infty$.
In fact - this double use of asymptotics is a very strong tool. And very much can be proven under that continuous formulation.
In practical problems one has often to deal with rather moderate dimensions.

And actually, the INIP-hard problems show that even those moderate sizes may cause great difficulties.
Grötschel (verbal-communication) characterizes the situation in the following sense:
For many INIP complete problems we frequently have a region of "small dimensional" instances which can be solved quickly, exactly and efficiently.
Then there is a set of moderate-sized instances, where the exploitation of the individual structure of the problem and the knowledge of the applicationfield often lead to practically efficient methods that solve these instances exactly and quickly.
There is a third category of instances with high dimensions, where only heuristics can help.
The quality of such heuristics can only be described in probabilistic terms. Here probabilistic analysis is necessary.
A fourth category of extremely high dimensions is nontractable by methods of discrete optimization - simply, for instance, because of storage-requirements. Here a reasonable line of attack is turning to continuous descriptions of the problems and trying with analytical methods.

One drawback of the present probabilistic analysis is that we do not know that the validity of the results is not restricted to the fourth category.
We need sharp estimations for the finite case - as for linear programming.
In my opinion the question whether IP = INIP should not be overestimated in its relevance. Whether an algorithm behaves like n^3 or like n is often more important.
Another risk in doing a probabilistic analysis lies in the choice of the stochastic model. The search for a realistic distribution remains a philosophical question. The impression

what this is may differ from individual to individual. Mathematicians tend to use the elegant and simple distributions: uniform distributions over [0,1] or over the unit square, constant probability for existence of an edge, independence of random variables etc. only these distributions seem to give a chance for evaluation.
But such simplifications may lead us far away from reality. If we try to modelize a realistic phenomenon we get into a problem-trivial at the first glance - but of greatest importance for probabilistic analysis.
For an example, consider the following situation. We want to check or test an algorithm for finding the shortest connection between two cities in a nationwide railway-network. To do a probabilistic analysis of that algorithm, we need a stochastic distribution of networks which could be representative or realistic. How can such a stochastic model be constructed?
Certainly the simple assumptions such as independence and constant probability for connections do not hold.
Given n points (the cities) and the (Euclidean) distances between pairs d_{ij}, we could associate the question for a connection with the distance (decreasing probability for growing d_{ij}) and with the importance of the cities (increasing probability with growing importance). Such an approach would be

$$P(\text{edge } v_i v_j \text{ exists}) = 1 - \frac{d_{ij}/c_{ij}}{d_{ij}/c_{ij} + 1}$$

where c_{ij} is increasing with the sum of inhabitants of v_i and v_j.
The single edges could be selected independently with that probability. But this still rather simple model ignores the fact that the first aim during the network-construction was to connect every city with the network. So every city should have at least two neighbour-stations. Those neighbour cities could be chosen out of $n - 1$ (experiment without repetition) where the probability for an edge from v_i to v_j is again proportional to the above given formula. Now we have a stochastic selection process of two stages:
First a selection of two out of $n - 1$, then independent Bernoulli experiments for the remaining edges.

Now recall that we did not care at all about evaluability during our attempt to obtain a realistic description.
The choice of the stochastic model would be less essential, if our analysis and our results were less sensitive. But they are sensitive, as we have learned from may examples.
In linear programming, the RSM and the SIM yield quite different behaviour. The results for Scheduling and Knapsack problems depend on independency of the entries. The Euclidean Problems of III 3. and III 4. are based on uniform distributions over the square. If we modify that assumption then it would still be possible to make a partition into equally probable subsets. But then one such subset could be large, the other small. The effect would be that we had much more trouble with the choice of representatives in the larger subsets, the error would be much higher.
There is another interesting question arising when we speak about sensitivity. While

A similar situation is with other problems:

> For Knapsack, Subset-Sum-Problems we analyze Greedy-algorithms. The patching algorithms for the traveling-salesman-problem are also rather simple. But already the improvement methods for the latter, cannot be treated yet.

I want to summarize these observations in the following way: We cannot rely on worst-case analysis alone. We must improve our ability to do average-case-analysis. We should know what we want, i.e. which stochastic model should be used.
We should not expect too much from probabilistic analysis but regard it as an additional aid for judging about algorithms. It may be that the real value of probabilistic analysis does not lie in the detailed formulas or results but in the process of understanding the manner how an algorithms works under normal circumstances.
I believe that this experience will be very helpful for finding practically efficient algorithms.

References

Adler, I., Karp, R. and Shamir, R. [1983a]: A Family of Simplex Variants Solving an $m \times d$ Linear Program in Expected Number of Pivot Steps Depending on d Only, University of California, Computer Science Division, Berkeley, December 1983.

Adler, I. Karp, R. and Shamir, R. [1983b]: A Simplex Variant Solving an $m \times d$ Linear Program in $0(\min^2,d^2))$ Expected Number of Pivot Steps. University of California, Computer Science Division, Berkeley.

Adler, I. and Megiddo, N. [1983]: A Simplex Algorithm where the Average Number of Steps is Bounded Between two Quadratic Functions of the Smaller Dimension, Dept. of Industrial Engineering and Operations Research, University of California, Berkeley, California, December 1983.

Adler, I. [1983]: The expected number of Pivots needed to solve parametric linear programs and the efficiency of the Self-Dual-Simplex-Method, University of California, Dept. of IE and OR.

AHO, A.V. Hopcroft, J.E., Ullmann J.D. [1974]: The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Massachusetts.

Artstein, Z. and Hart, S. [1981]: Law of Large Numbers for Random Sets and Allocation Processes, Mathematics of Operations Research, Vol. 6 No. 4 (1981), 485-492.

Bachem, A. and Euler, R. [1984]: Recent Trends in Combinatorial Optimization, OR-Spektrum, Band 6, Heft 1 (1984) 1-22.

Benveniste, R. [1982]: Evaluating Computational Efficiency: A Stochastic Approach, Mathematical Programming 22 (1982), 261-287.

Berenguer, S.E. and Smith, R.L. [1985]: The Expected Number of Extreme Points of a Random Linear Program, Mathematical Programing Study (1985), 1-6.

Blair, C. [1983]: Random Linear Programs with Many Variables and Few Constraints, College of Commerce and Business Administration. University of Illinois at Urbana, Champaign, April 1983.

Borgwardt, K.H. [1977]: Untersuchungen zur Asymptotik der mittleren Schrittzahl von Simplexverfahren in der linearen Optimierung, Dissertation Universität Kaiserslautern.

Borgwardt, K.H. [1982a]: Some Distribution-Independent Results About the Asymptotic Order of the Average Number of Pivot Steps of the Simplex-Method, Mathematics of Operations Research 7 (1982), 441-462.

Borgwardt, K.H. [1982b]: The Average Number of Pivot Steps Required by the Simplex-Method is Polynomial, Zeitschrift for Operations Research 26 (1982), 157-177.

Borgwardt, K.H. [1986]: The Simplex Method - A Probabilistic Analysis, Springer Verlag (1986).

Brown, G.G. and Schubert, B.O. [1984]: On Random Binary Trees, Mathematics of Operations Research (1984), No. 1, 43-65.

Burkard, R.E. and Fincke, U. [1982]: On Random Quadratic Bottleneck Assignment Problems, Mathematical Programming 23 (1982), 227-232.

Burkhard, R.E. and Fincke, U. [1983]: The Asymptotic Probabilistic Behaviour of Quadratic Sum Assignment Problems, Zeitschrift für Operations Research, Volume 27 (1983), 73-81.

Burkhard, R. and Fincke, U. [1985]: Probabilistic Asymptotic Properties of Some Combinatorial Optimization Problems, Discrete Applied Mathematics 12 (1985), 21-29.

Carson, J.S. and Law, A.M. [1977]: A Note on SPIRA'S Algorithm for the All-Pairs Shortest-Path Problem, SIAM Journal of Computation, Vol. 6 No. 4 (1977), 696-699.

D'Atri, G. and Puech, C. [1982]: Probabilistic Analysis of the Subset-Sum Problem, Discrete Applied Mathematics 4 (1982), 329-334.

Dempster, Fisher, Jansen, Lageweg, Lenstra and Rinnooy Kan [1983]: Analysis of Heuristics for Stochastic Programming: Results for Hierarchical Scheduling Problems, Vol. 8 No. 4 (1983), 525-537.

Dijkstra, E.W. [1959]: A Note on Two Problems in Connexion with Graphs. Numerische Mathematik 1 (1959), 269-271.

Erdös P. and Renyi, A. [1959]: On Random Graphs I, Publ. Math. Debrecan 6 (1959), 290-297.

Erdös, P. [1959]: Graph Theory and Probability I, Canadian Journal of Mathematics 11 (1959), 34-38.

Erdös, P. [1961]: Graph Theory and Probability II, Canadian Journal of Mathematics 13 (1961), 346-352.

Fisher, M.L. and Hochbaum, D.S. [1980]: Probabilistic Analysis of the Planar K-MEDIAN Problem, Mathematics of Operations Research, Vol.5 No. 1 (1980), 27-34.

Fisher, M.L. and Krieger, Abba M. [1984]: Analysis of a Linearization Heuristic for Single-Machine Scheduling to Maximize Profit, Mathematical Programming 28 (1984), 218-225.

Fox, B.L. , Lenstra, J.K. , Rinnooy Kan, A.H.G. and Schrage, L.E. [1977]: Branching from the Largest Upper Bound: Folklore and Facts, Report 7722, Erasmus University Rotterdam.

Frenk, J.B.G. Van Houweninge, M. and Rinnooy Kan, A.H.G. [1985]: Asymptotic Properties of the Quadratic Assignment Problem, Mathematics of Operations Research, Vol. 10 No. 1 (1982), 100-116.

Frieze, A.M. [1985]: On the Exact Solution of Random Travelling Salesman Problems With Medium Size Integer Coefficients, Dept. of Comp. Science & Statistics, Queen Mary College, London, England.

Gale, D. [1969]: How To Solve Linear Inequalities, American Mathematical Monthly 76 (1969), 589-599.

Garey, M.R. and Johnson, D.S. [1979]: Computers and Intractability: A Guide to the Theory of INIP-Completeness, Freeman, San Francisco 1979.

Gazmuri, P.G. [1985]: Probabilistic Analysis of a Machine Scheduling Problem, Mathematics of Operations Research, Vol. 10 No. 2 (1985), 328-339.

Graph Theory and Probability II, Canadian Journal of Mathematics 13 (1961), 346-352.

Grötschel, M. [1984] Developments in Combinatorial Optimization, Universität Augsburg, Preprint Nr. 26.

Grötschel, M. [1985]: Operations Research I, Skripten zur Vorlesung SS 1985, Universität Augsburg.

Haimovich, M. [1983]: The Simplex Algorithm is Very Good! - On The Expected Number of Pivot Steps and Related Properties of Random Linear Programs, Columbia University, New York, 1983.

Halton, J.H. and Terada, R. [1982]: A Fast Algorithm for the Euclidean Traveling Salesman Problem Optimal with Probability one, SIAM Journal of Computation, Vol. 11, No. 1 (1982), 28-46.

Hochbaum, D.S. and Shmoys, D.B. [1985]: A Best Possible Heuristic for the k-Center Problem, Mathematics of Operations Research, Vol. 10 No. 2 (1985), 180-184.

Karp, R.M. and Steele, J.M. [1986]: Probabilistic Analysis of Heuristics, in: The Traveling Salesman Problem, Lawler, Lenstra, Rinnooy Kan, Shmoys (eds.), John Wiley & Sons (1985), 181-205.

Karp, R.M. [1976]: The Probabilistic Analysis of Some Combinatorial Search Algorithms, in: Algorithms and Complexity, J.F. Traub (ed.), Academic Press, New York, 1976, 1-19.

Kemp, R. [1984]: Fundamentals of the Average Case Analysis of Particular Algorithms, Teubner, John Wiley & Sons, 1984.

Klee, V. [1980]: Combinatorial Optimization: What is the State of the Art? Mathematics of Operations Research, Vol. 5, No. 1 (1980), 1-26.

Klee, V. and Minty, G.J. [1972]: How Good is the Simplex Algorithm? In O. Shisha (ed) Inequalities III, Academic Press, New York, 159-175.

Korte, B. and Hausmann, D. [1978]: An Analysis of the Greedy Heuristic for Independence Systems, Annals of Discrete Mathematics 2 (1978), 65-74.

Lawler, E.L. [1979]: Fast Approximation Algorithms for Knapsack Problems, Mathematics of Operations Research, Vol. 4 No. 4 (1979), 339-356.

Lenstra, J.K. Rinnooy Kan A.H.G., Van Emde Boas, P. [1982]: An Appraisal of Computational Complexity for Operations Researchers, Erasmus University Rotterdam, Econometric Institute, Report 8211/0.

Lenstra, J.K. and Rinnooy Kan, A.H.G.U. [1977]: Computational Complexity of Discrete Optimization Problems, Erasmus University Rotterdam, Report 7727/0.

Lieberherr, K. [1981]: Probabilistic Combinatorial Optimization, Lecture Notes in Computer Science, 118, (1981), 423-432.

Lindberg, P.O. and Olafsson, S. [1980]: On the Lengths of Simplex Paths: The Assignment Case, Royal Institute of Technology, Stockholm, TRITA-MAI 1980-29.

Lindberg, P.O. and Olafsson, S. [1983]: On the Lengths of Simplex Paths: The Transportation Case, Royal Institute of Technology, Stockholm, TRITA-MAI 1983-6.

Lifschitz, V. [1980]: The Efficiency of an Algorithm of Integer Programming: A Probabilistic Analysis, Proceedings of the American Mathematical Society, Vol. 79 No. 1 (1980), 72-76.

Loulou, R. [1984]: Tight Bounds and Probabilistic Analysis of Two Heuristics for Parallel Processor Scheduling, Mathematics of Operations Research, Vol. 9 No. 1 (1984), 142-150.

May, J.H. and Smith, R.L. [1982]: Random Polytopes: Their Definition, Generation and Aggregate Properties Mathematical Programing 24 (1982), 39-54.

Meanti, Rinnooy Kan, Stougie, Vercellis [1984]: A Probabilistic Analysis of the Multiknapsack Value Function. Report 8435, Erasmus University Rotterdam, Econometric Institute.

Nicholson, T.A.J. [1966]: Finding the Shortest Route Between two Points in a Network, Computing Journal 9. (1966), 275-280.

Palmer, E.M. [1985]: Graphical Evolution : An Introduction to the Theory of Random Graphs, John Wiley & Sons, 1985.

Posa, L. [1976]: Hamiltonian Circuits in Random Graphs, Discrete Mathematics 14 (1976), 359-364.

Shamir, R. [1984]: The Efficiency of the Simplex-Method: A Survey, Dept. of Industrial Engineering and Operations Research, University of California, Berkeley, 1984.

Smale, S. [1982]: The Problem of the Average Speed of the Simplex Method, Proceedings of the 11th International Symposium on Mathematical Programming, Universität Bonn, August 1982, 530-539.

Smale, S. [1983]: On the Average Speed of the Simplex Method, Mathematical Programing 27 (1983), 241-262.

Spira, P.M. [1973]: A New Algorithm for Finding All Shortest Paths in a Graph of Positive Arcs in Average Time $0(n^2 \log^2 n)$, SIAM J. of Computation Vol. 2, No.1 (1973), 28-32.

Stein, D.M. [1978]: An Asymptotic, Probabilistic Analysis of a Routing Problem, Mathematics of Perations Research, Vol. 3 No. 2 (1978), 89-101.

Tinhofer, G. [1984]: Probabilistische Ansätze in der diskreten Optimierung, Berichte der Mathematisch-Statistischen Sektion in der Forschungsgesellschaft Joanneum, Bericht Nr. 230 (1984).

Todd, M.J. [1983]: Polynomial Expected Behaviour of a Pivoting Algorithm for Linear Complimentarity and Linear Programming Problems. Technical Report No. 595, School of Operations Research and Industrical Engineering. Cornell University, New York, 1983.

Walkup, D.W. [1979]: On the Expected Value of a Random Assignment Problem, SIAM Journal of Computation, Vol. 8 No. 3 (1979), 440 - 442.