Typical Properties of Winners and Losers in Discrete Optimization

Rene Beier*
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Berthold Vöcking†
Fachbereich Informatik
Universität Dortmund, Germany

ABSTRACT

We present a probabilistic analysis for a large class of combinatorial optimization problems containing, e.g., all *binary optimization problems* defined by linear constraints and a linear objective function over $\{0,1\}^n$. By parameterizing which constraints are of stochastic and which are of adversarial nature, we obtain a semi-random input model that enables us to do a general average-case analysis for a large class of optimization problems while at the same time taking care for the combinatorial structure of individual problems. Our analysis covers various probability distributions for the choice of the stochastic numbers and includes *smoothed analysis* with Gaussian and other kinds of perturbation models as a special case. In fact, we can exactly characterize the smoothed complexity of optimization problems in terms of their random worst-case complexity.

> A binary optimization problem has a *polynomial smoothed complexity* if and only if it has a pseudopolynomial complexity.

Our analysis is centered around structural properties of binary optimization problems, called *winner*, *loser*, and *feasibility gaps*. We show, when the coefficients of the objective function and/or some of the constraints are stochastic, then there usually exist a polynomial $n^{-\Omega(1)}$ gap between the best and the second best solution as well as a polynomial slack to the boundary of the constraints. Similar to the condition number for linear programming, these gaps describe the sensitivity of the optimal solution to slight perturbations of the input and can be used to bound the necessary accuracy as well as the complexity for solving an instance. We exploit the gaps in form of an adaptive rounding scheme increasing the accuracy of calculation until the optimal solution is found. The strength of our techniques is illustrated by applications to various NP-hard optimization problems from mathematical programming, network design, and scheduling for which we obtain the the first algorithms with polynomial average-case/smoothed complexity.

*Email: rbeier@mpi-sb.mpg.de. Supported by the DFG studies program GRK 623.
†Email: berthold.voecking@uni-dortmund.de. Supported in part by DFG grant Vo889/1-2.

STOC'04, June 13–15, 2004, Chicago, Illinois, USA.

Categories and Subject Descriptors

F.2.0 [**Analysis of Algorithms and Problem Complexity**]: Miscellaneous

General Terms

Algorithms

Keywords

optimization problems, average-case analysis, smoothed analysis

1. INTRODUCTION

Many combinatorial optimization problems have an objective function or constraints specified in terms of real numbers representing natural quantities like time, weight, distance, or utility. This includes some well-studied optimization problems like, e.g., traveling salesperson, shortest path, minimum spanning tree as well as various scheduling and packing problems. When analyzing the complexity of algorithms for such problems, we usually assume that these numbers are integers or rational numbers with a finite length representation. The hope is that it suffices to measure and compute with some bounded precision in order to identify an optimal or close to optimal solution. In fact, if real numbers occur only in the objective function and if this objective function is well-behaved (e.g., a linear function) then calculating with reasonable approximations of the input numbers yields a feasible solution whose objective value is at least close to the optimal objective value. More problematically, however, if the constraints are defined by real numbers, then calculating with rounded input numbers might miss all interesting solutions or might even produce infeasible solutions.

How can one solve optimization problems (efficiently) on a computer when not even the input numbers can be specified exactly? – In practice, optimization problems in which real numbers occur in the input are solved by simply rounding the real numbers more or less carefully. Fortunately, this approach seems to yield reasonable results. We seek for a theoretically founded explanation why this rounding approach usually works. Studying this issue under worst case assumptions does not make very much sense as, in the worst case, the smallest inaccuracy might lead to an infeasible or utterly sub-optimal solution. This question needs to be studied in a stochastic model. In the following probabilistic analysis, we will show that, under some reasonable and quite general stochastic assumptions, one can usually round real-valued input numbers after only a logarithmic number of bits without changing the optimal solution. In fact, our probabilistic analysis goes far beyond the point of explaining phenomena occurring in practice. We are

able to provide algorithms with polynomial average-case complexity (more precisely, polynomial smoothed complexity) for a quite general class of discrete optimization problems. Our analysis covers various well-studied NP-hard discrete optimization problems from mathematical programming, network design, and scheduling like, e.g., multi-dimensional knapsack, constrained spanning tree, or scheduling to minimize the weighted number of tardy jobs.

1.1 A semi-random input model for discrete optimization problems

A discrete optimization problem is specified in terms of an objective function and a feasible region over a set of discrete variables. Usually the variables are binary, the objective function is linear, and the feasible region is defined by a set of linear constraints. Suppose an optimization problem Π is defined by a set of n binary variables $x_1,\ldots,x_n$, an objective function of the form *minimize* (or *maximize*) $c^T x$, and a finite set of constraints $w_j^T x \leq t_j$ or $w_j^T x \geq t_j$. In the following, we use the phrase *expression* as a generic term for the linear expressions $c^T x$ and $w_j^T x$ occurring in the objective function and the constraints, respectively. We explicitly distinguish between those expressions that shall be of stochastic nature and those that shall be of adversarial nature. In particular, we assume that there is a set of *stochastic expressions* whose coefficients shall be random or randomly perturbed real numbers and a set of *adversarial expressions* in the sense that we treat the numbers in these expressions like rational or integer numbers in a usual worst-case analysis. The reason for distinguishing stochastic and adversarial expressions is that we do not want that the randomization destroys the combinatorial structure of the underlying optimization problems. At this point, let us remark that we can slightly relax the assumptions made above. In fact, the linearity assumption for adversarial expressions can be dropped, that is, objective function and constraints only need to be specified by linear expressions if they are stochastic.

Let us make the above explanations more concise. The number of stochastic expressions is denoted by $k \geq 1$ and the number of stochastic constraints by $k' \in \{k-1,k\}$. A *stochastic instance* of an optimization problem Π is described in terms of a possibly stochastic (and in this case linear) objective function and a feasible region that is defined by the intersection of an arbitrary subset $\mathcal{S} \subseteq \{0,1\}^n$ with k' subsets $\mathcal{B}_1,\ldots,\mathcal{B}_{k'} \subseteq \{0,1\}^n$ each of which is defined by a stochastic, linear constraint. The coefficients in the stochastic expressions are specified by independent continuous probability distributions with domain $\mathbb{R}$. Different coefficients might be drawn according to different distributions. The only restriction on these distributions is that their density function is piecewise continuous and bounded. Assuming bounded densities is necessary as otherwise worst-case instances could be approximated arbitrarily well by specifying distributions with very high density. For a given distribution, the supremum of its density function is called its *density parameter*. We will see that the maximum density parameter over all distributions plays an important role in our analysis. This parameter is denoted by ϕ. Intuitively, ϕ can be seen as a measure specifying of how close the instances might be to the worst case. A worst-case instance can be interpreted as a stochastic instance in which the probability measure for each stochastic number is mapped to a single point. Thus, the larger ϕ, the closer we are to a worst-case analysis.

In our probabilistic analysis, we assume that the objective function defines a unique ranking among all solutions in $\{0,1\}^n$. Observe, if the objective function is stochastic then the coefficients are continuous random variables so that the probability that there exist solutions with same objective value is 0. In other words, a unique ranking is given with probability 1. Recall that the objective function does not have to be linear if it is adversarial, but if it is linear, i.e., of the form $c^T x$, $c \in \mathbb{Q}^n$, then a unique ranking can always be enforced by encoding the lexicographical order among the solutions into the less significant bits of the objective function without changing the computational complexity of the underlying optimization problem by more than a polynomial factor. In fact, most of the algorithmic problems that we will study have algorithms that implicitly realize a unique ranking. In this case, one does not even need an explicit encoding. Given a unique ranking, we seek to find the *winner*, i.e., the highest ranked solution in $\mathcal{S} \cap \mathcal{B}_1 \cap \ldots \cap \mathcal{B}_{k'}$. In the following, optimization problems satisfying all the conditions above are called *binary optimization problems with stochastic expressions* or, for short, *binary optimization problems*.

Smoothed Analysis. The framework of smoothed analysis was introduced by Spielman and Teng in [24]. Essentially, they show that the Simplex algorithm has polynomial smoothed complexity when input numbers are slightly perturbed using a Gaussian distribution with small standard deviation. It is assumed that first an adversary specifies all coefficients in the constraint matrix such that $||w|| \leq 1$, for every coefficient w, and then these adversarial numbers are slightly perturbed by adding to each of them a random numbers drawn according to a Gaussian distribution with mean 0 and a specified standard deviation $\sigma > 0$. Spielman and Teng prove a running time for the Simplex algorithm under the shadow vertex pivot rule that is polynomial in the number of variables and constraints as well as in $\frac{1}{\sigma}$. Similar results have been obtained for other variants of the Simplex algorithm as well as for a few other problems [1, 3, 5, 8, 25]. Our probabilistic analysis is not restricted to the model of smoothed analysis, but we use this nice framework to illustrate our results.

We generalize smoothed analysis as follows. At first, we do not necessarily perturb all coefficients in the constraint matrix, but only the coefficients in the stochastic expressions. Initially, an adversary chooses all input numbers. The domain of the coefficients in those constraints that shall be stochastic is restricted to $[0,1]$ or $[-1,1]$, depending on whether the domain should be non-negative or also include negative numbers. Then a random perturbation slightly changes the coefficients in the stochastic constraints by adding an independent random number to each of them. These random numbers are drawn according to a specified family of probability distributions satisfying the following conditions. Let $f : \mathbb{R} \to \mathbb{R}_{\geq 0}$ be any piecewise continuous density function such that $\sup_s(f(s)) = 1$ and $\int |s| f(s) ds$ is finite, that is, the random variable described by f has a finite expected absolute value. Function f is called the *perturbation model*. For $\phi \geq 1$, we define f_ϕ by scaling f, that is, $f_\phi(s) = \phi f(s\phi)$, for every $s \in \mathbb{R}$. This way, the density parameter of f_ϕ is ϕ. We obtain ϕ-*perturbations* according to the perturbation model f by adding an independent random variable with density function f_ϕ to each stochastic input number. For example, one obtains the Gaussian perturbation model from [24] by choosing f to be the Gaussian density with standard deviation $(2\pi)^{-1/2}$. A non-negative domain can be obtained, e.g., by choosing f to be the density of the uniform distribution over $[0,1]$. In [24] the running time is described in terms of the standard deviation σ. In contrast, we describe the running time in terms of the density parameter ϕ. For the Gaussian and the uniform distribution these two parameters are closely related; in both cases, ϕ is proportional to $\frac{1}{\sigma}$.

Let us illustrate our semi-random input model by an example. In the single-pair shortest path problem one seeks for the shortest path in a graph $G = (V,E)$ between a given source $s \in V$ and a given target $t \in V$. In the binary program formulation of this problem there is a variable x_e for each edge $e \in E$. Thus, n corresponds to the number of edges. A solution x is feasible if the edges in the set

$\{e \in E \mid x_e = 1\}$ form a path from s to t. Let $\mathcal{S}$ denote the set of all solutions satisfying this condition. The combinatorial structure described by $\mathcal{S}$ should not be touched by our randomization. It makes sense, however, to assume that the objective function is stochastic as its coefficients describe measured quantities. So we may assume that these coefficients are perturbed with uniform ϕ-perturbations, that is, each of these coefficients corresponds to the sum of an adversarial number from $[0,1]$ and an independent random number drawn uniformly from $[0,\phi^{-1}]$. In the constrained shortest path problem (see, e.g., [11]), edges do not only have lengths but additionally each edge comes with a latency parameter ℓ_e. Now one seeks for the shortest path satisfying an additionally specified linear time constraint $\sum_e \ell_e x \leq T$. This additional constraint corresponds to a subset $\mathcal{B}_1 \subseteq \{0,1\}^{|E|}$ so that now $\mathcal{B}_1 \cap \mathcal{S}$ is the set of feasible solutions. Due to the additional constraint, the problem becomes NP-hard. We will see, however, that there is an algorithm with "polynomial smoothed complexity" if either the objective function or the additional latency constraint is stochastic.

1.2 How accurately do we need to calculate?

More precisely, we ask how many bits of each stochastic input number do we need to reveal in order to determine the winner? – We say that the winner is *determined* after revealing some number of the bits, when there is only one possible candidate for the winner, regardless of the outcomes of the unrevealed bits.

THEOREM 1. *Consider any instance of a binary optimization problem Π. Let $n \geq 1$ denote the number of binary variables and $k \geq 1$ the number of stochastic expressions.*

a) Suppose the expected absolute value $\mathbf{E}[|w|]$ of every stochastic coefficient w is bounded from above by $\mu > 0$. Then the number of bits in front of the floating point of any stochastic number is bounded by $O(\log(1+\mu nk))$, **whp**[1].

b) Let $\phi > 0$ denote the maximum density parameter, that is, all density functions are upper-bounded by ϕ. Then the winner is uniquely determined when revealing $O(\log(1+\phi nk))$ bits after the binary point of each stochastic coefficient, **whp**.

One can always scale the input of a linear optimization problem by multiplying all input numbers with a factor $\gamma > 0$. Obviously, the parameters μ and ϕ must be adapted to this scaling, that is, μ needs to be multiplied with γ and ϕ with $\frac{1}{\gamma}$. Observe that this kind of scaling does not change the overall number of bits that need to be revealed as $\log(\mu nk) + \log(\phi nk) = \log(\gamma\mu nk) + \log(\frac{1}{\gamma}\phi nk)$. In case of a smoothed analysis, the right way to scale the input numbers is already build into the model. According to our definitions, the density function f specifying the perturbation model has to have a finite expected absolute value. For any fixed model of perturbation, $\int |s| f(s) ds = O(1)$. In particular, the expected absolute value of the density function f_ϕ is $O(\phi^{-1})$. Taking into account that the domain of the initial adversarial choices for the stochastic coefficients is $[-1,1]$ or $[0,1]$, we observe that ϕ-perturbations yield coefficients with an expected absolute value of at most $\mu = O(1+\frac{1}{\phi})$. In order to simplify the notation, our model of smoothed analysis is restricted to density parameters $\phi \geq 1$. This leads to the following result on the overall number of bits that need to be revealed per stochastic input number.

COROLLARY 2. *For any fixed perturbation model f, the winner is uniquely determined when revealing $O(\log(\phi nk))$ bits of each stochastic coefficient,* **whp**.

[1] with high probability, with probability $1-(nk)^{-\alpha}$, for every fixed $\alpha > 0$

Let us explain the concepts and ideas behind the analysis for Theorem 1. Part a) of the theorem follows simply by applying the Markov inequality to the expected absolute values of the individual coefficients. The interesting part of the theorem is stated in b). In order to identify the winner one needs to *isolate* the winner from other feasible solutions having a worse objective value. Furthermore, one needs to *separate* the winner from those infeasible solutions that have a better objective value than the winner. Our analysis is based on a *generalized Isolating Lemma* – i.e., a generalization of the well-known Isolating Lemma by Mulmuley, Vazirani and Vazirani [18] – and a novel *Separating Lemma*.

The Isolating Lemma was originally presented in an article about RNC algorithms for perfect matchings [18]. It is known, however, that the lemma does not only apply to the matching problem but to to general binary optimization problems with a linear objective function. The lemma states that the optimal solution of a binary optimization problem is unique with probability at least $\frac{1}{2}$ when choosing the coefficients of the objective function independently, uniformly at random from the set $\{1,2,\ldots,2n\}$. This is a very surprising and counterintuitive result as there might be an exponential number of feasible solutions whose objective values fall all into a polynomially large set, namely the set $\{1,2,\ldots,2n^2\}$, so that one can expect that an exponential number of solutions are mapped to the same objective value. The reason why the winner nevertheless is isolated is that the objective values of different solutions are not independent but the solutions represent subsets over a ground set of only n random numbers. We adapt the Isolating Lemma towards our continuous setting and generalize it towards piecewise continuous probability distributions as described in Section 1.1. In particular, different coefficients may follow different continuous probability distributions. Suppose only the objective function is stochastic, and the feasible region is fixed arbitrarily. Let ϕ denote the maximum density parameter over all coefficients in the objective functions. Define the *winner gap* to be the difference between the objective value of the winner and the second-best feasible solution, provided there are at least two feasible solutions. The generalized Isolating Lemma states that the winner gap is a continuous random variable whose density function is bounded from above by $2\phi n$, and this bound is tight. From this result one can immediately derive the following lower bound on the size of the winner gap. For every $\varepsilon \in [0,1]$, the winner gap is lower-bounded by $\frac{\varepsilon}{2\phi n}$ with probability at least $1-\varepsilon$. As a consequence, it suffices to reveal only $O(\log(\phi n))$ bits of each coefficient of the objective function in order to identify the winner, **whp**.

We accompany the Isolating Lemma with a novel *Separating Lemma*, enabling us to separate the winner from infeasible solutions with better objective value than the winner. For the time being, consider any binary optimization problem in which a single constraint is stochastic. The difficulty in checking the feasibility with respect to this constraint is that it might be likely that there are many solutions that are exponentially close to the constraint hyperplane. Nevertheless, we will see that the optimal solution can be identified by inspecting only a logarithmic number of bits per input number, **whp**. The reason is that we do not need to check the feasibility of all solutions but only of some particular solutions. The *losers* are those solutions that have a rank higher than the winner but they are infeasible because of the considered constraint. The *loser gap* is defined to be the minimal amount by which a loser (except for the solution 0^n) exceeds the constraint threshold. The Separating Lemma shows that the supremum of the density function of the loser gap is at most ϕn^2. Hence, for every $\varepsilon > 0$, the loser gap is at least $\frac{\varepsilon}{\phi n^2}$ with probability at least $1-\varepsilon$. Let us try to give some intuition about this result. If there are only a few

losers then one can imagine that neither of them comes very close to a random or randomly perturbed hyperplane. However, there might be an exponential number of losers. In this case, however, the winner has a relatively low rank as there is an exponential number of solutions better than the winner; but this is very unlikely if the constraint hyperplane is likely to come very close to the good solutions which correspond to the losers. Seeing it the other way around, if there are many losers then the hyperplane is likely to be relatively far away from the losers, which might intuitively explain the phenomenon described by the Separating Lemma. Besides the loser gap, we study the so-called *feasibility gap* corresponding to the slack of the optimal solution with respect to the stochastic constraint. Essentially, we prove that the density functions of loser and feasibility gaps have the same maximum supremum so that the density of the feasibility gap is lower-bounded by $\frac{\varepsilon}{\phi n^2}$ as well. In fact, our analysis for loser and feasibility gaps is heavily based on symmetry properties between them.

Let us remark that, when analyzing the winner gap, it is assumed a random objective function and a fixed feasible region. In contrast, when analyzing loser and feasibility gaps, it is assumed a random constraint or a set of random constraints instead of a random objective function. In other words, the random expressions defining the objective function and the constraints are assumed to be stochastically independent. In fact, if the feasible region and the objective function are correlated, then winner, loser, and feasibility can not be lower-bounded by a polynomial. The optimization variant of the subset-sum problem (i.e., knapsack with profits equal to weights) is a simple counterexample. Lueker [16] proved exponentially small gaps for this problem.

1.3 Characterizing polynomial smoothed complexity

Based on the gap properties, we aim at characterizing which discrete optimization problems have polynomial time algorithms under random perturbations. We formalize this as follows. Fix any binary optimization problem Π and any perturbation model f. Let I_N denote the set of all unperturbed instances of length N that the adversary may specify. The definition of the input length N needs some clarification as the coefficients in the stochastic expressions are assumed to be real numbers. We define that each of these numbers has a virtual length of one. (This way, we ensure $N \geq kn$.) The bits of the stochastic numbers can be accessed by asking an oracle in time $O(1)$ per bit. The bits after the binary point of each coefficient are revealed one by one from left to right. The deterministic part of the input does not contain real numbers and can be encoded in an arbitrary fashion. For an instance $I \in I_N$, let $I + f_\phi$ denote the random instance that is obtained by a ϕ-perturbation of I. We say that Π has *smoothed polynomial complexity* if and only if it admits an algorithm $\mathcal{A}$ whose running time T satisfies

$$\exists \alpha, \beta > 0 : \forall \phi \geq 1 : \forall N \in \mathbb{N} : \max_{I \in I_N} \mathbf{E}\left[\left(T(I + f_\phi)\right)^\alpha\right] \leq \beta \phi N \ .$$

This definition of polynomial smoothed complexity follows more or less the way how polynomial complexity is defined in average-case complexity theory, adding the requirement that the running time should be polynomially bounded not only in N but also in ϕ. It is not difficult to show that the assumption on the running time of $\mathcal{A}$ is equivalent to requiring that there exists a polynomial $P(N, \phi, \frac{1}{\varepsilon})$ such that for every $N \in \mathbb{N}, \phi \geq 1, \varepsilon \in [0,1]$, the probability that the running time of $\mathcal{A}$ exceeds $P(N, \phi, \frac{1}{\varepsilon})$ is at most ε. Observe that this does not imply that the expected running time is polynomially bounded. To enforce expected polynomial running time, the exponent α in the definition of polynomial smoothed complexity should have been placed outside instead of inside the expectation. The reason for not defining polynomial complexity based on the expected running time is that this is not a sufficiently robust notion. For example, an algorithm with expected polynomial running time on one machine model might have expected exponential running time on another machine model. In contrast, the above definition yields a notion of polynomial smoothed complexity that does not vary among classes of machines admitting polynomial time simulations among each other. Although polynomial smoothed complexity does not always imply polynomial bounds on the expected running time, we will show that several of our algorithmic results yield expected polynomial running time on a RAM.

We show that the smoothed complexity of a binary optimization problem can be characterized in terms of its worst-case complexity. Theorem 1 shows that one usually only needs to reveal a logarithmic number of bits per real-valued input number. This suggests that there should be a connection between pseudopolynomial worst-case running time and polynomial average-case complexity. For a binary optimization problem Π, let Π_u denote the corresponding optimization problem in which all numbers in the stochastic expression are assumed to be integers in unary representation instead of randomly chosen real-valued numbers. The following theorem holds for any fixed perturbation model f.

THEOREM 3. *A binary optimization problem Π has polynomial smoothed complexity if and only if $\Pi_u \in \mathsf{ZPP}$.*

In other words, Π has polynomial smoothed complexity if it admits a (possibly randomized) algorithm with (expected) pseudopolynomial worst-case running time. This characterization immediately shows that strongly NP-hard optimization problems do not have polynomial smoothed complexity, unless $\mathsf{ZPP} = \mathsf{NP}$. This result might not sound very surprising as the hardness of strongly NP-hard problems does not rely on large or precisely specified input numbers. Observe, however, that the strong NP-hardness of a problem does not immediately rule out the possibility of a polynomial average-case complexity. For example, the TSP problem with edge lengths drawn uniformly at random from $[0,1]$ might have a polynomial average-case complexity. Our theorem, however, shows that it does not have a polynomial smoothed complexity, unless $\mathsf{P} = \mathsf{NP}$. The more sophisticated part of the theorem is the other direction stating that every binary problem admitting a pseudopolynomial time algorithm has a polynomial smoothed complexity. This result is based on the generalized Isolating Lemma and the Separating Lemma. The idea is as follows. We design efficient verifiers checking whether a solution computed with a certain precision is actually the optimal solution of Π. The success probability of these verifiers is analyzed with the help of the gap properties. In an adaptive rounding procedure we increase the precision until the optimal solution is found. The overall running time of this meta-algorithm is polynomial if the algorithm computing the solutions with bounded precision has pseudopolynomial running time.

Algorithmic Applications. Let us illustrate the strength of Theorem 3 by giving some algorithmic applications to some well-known optimization problems and comparing these results with previous work on the probabilistic analysis of optimization problems. There has been substantial effort to analyze random instances of the knapsack problem, see, e.g., [5, 6, 12, 14, 15]. The knapsack problem can be seen as the simplest non-trivial binary optimization problem as its feasible region is described by only one single linear constraint. The problem belongs to the class of packing problems, that is, the constraint is of the form $w^T x \leq t$ and the coefficients are assumed to be non-negative. To our knowledge, the knapsack

problem is the only NP-hard optimization problem that was previously known to have polynomial smoothed complexity [5]. The multi-dimensional knapsack problem is a natural generalization in which there are multiple packing constraints instead of only one. Dyer and Frieze [9] proved that, with constant probability, this problem can be solved in polynomial time if the number of constraints is constant and the coefficients in the constraints as well as in the objective function are chosen uniformly at random from $[0,1]$. Their result, however, does not yield polynomial average-case complexity as the dependence of the running time on the failure probability is not bounded by a polynomial. The multiple knapsack problem with a constant number of constraints admits a pseudopolynomial algorithm. Hence, Theorem 3 implies a polynomial smoothed and, hence, also a polynomial average-case complexity for this problem. Moreover, the pseudopolynomial algorithm also works for general 0/1 integer programming with any fixed number of constraints. Therefore, this class of problems has polynomial smoothed complexity when assuming that the objective function and all constraints are stochastic. Furthermore, the theorem shows that general 0/1 integer programming with an unbounded number of constraints has no polynomial smoothed complexity as it is strongly NP-hard.

The problem of scheduling to minimize the weighted number of tardy jobs is defined by n jobs each of which coming with a processing time p_i, a due date d_i, and a penalty c_i that has to be paid if job i is not finished in time. The jobs shall be scheduled on a single machine such that the sum of the penalties is minimized. In terms of n binary variables $x_1,\ldots,x_n$, the objective is to minimize $c^T x$ where $x_i = 1$ if job i cannot be finished in time. Observe that the problem is essentially solved once these binary variables are determined as we can assume w.l.o.g. that an optimal schedule executes the selected jobs in the order of non-decreasing deadlines. The exact formulation of the feasible region in terms of a binary program is not of interest to us. The input of the problem consists only of $3n$ numbers, the processing times, the due dates, and the penalties. As the scheduling problem admits an algorithm whose running time is pseudopolynomial with respect to the penalties, the problem has polynomial smoothed complexity for stochastic penalties.

Next we come to multicriteria optimization problems. If several criteria shall be optimized simultaneously then usually one of them is declared to be the objective function, and the others are formulated in form of a constraint with a given threshold. Often when a single-criteria optimization problem is polynomial, the problem becomes NP-hard when adding another criteria in form of a linear constraint. Examples for such problems are shortest path, spanning tree, or matching [17, 11, 19]. Theorem 3 enables us to prove polynomial smoothed complexity for such multicriteria problems as follows. The problems listed above have exact algorithms with pseudopolynomial running time [19, 4, 18], that is, given an integer k and an instance of these problems one can compute a solution with objective value exactly k in pseudopolynomial time. Using standard coding techniques (see, e.g., [23]) a pseudopolynomial algorithm for the exact single-criteria decision problem implies a pseudopolynomial algorithm for its multicriteria optimization variant. Combining this observation with Theorem 3 yields the following result.

COROLLARY 4. *Let Π be a (single objective) binary optimization problem. Suppose the exact version of Π admits an algorithm with pseudopolynomial running time. Then any multicriteria variant of Π with stochastic coefficients with respect to all criteria has a polynomial smoothed complexity.*

A similar approach was used in [19] to derive approximation schemes for multiobjective optimizations problems. The corollary implies polynomial smoothed complexity for the multicriteria variants of shortest path, spanning tree, and matching. One does not always need to assume that all criteria are of stochastic nature. For example, the bicriteria variant of the shortest path problem, i.e., the constrained shortest path problem, has an algorithm whose running time is pseudopolynomial with respect to the objective function and another algorithm that is pseudopolynomial with respect to the additional constraint. Applying Theorem 3 directly to these algorithms yields that the constrained shortest path problem has polynomial smoothed complexity even when either only the objective function or the additional constraint are stochastic.

1.4 Other aspects

In order to obtain **expected polynomial running time** under random perturbations one needs an algorithm with "pseudolinear" instead of pseudopolynomial running time. Such pseudolinear algorithms exist on a uniform RAM, e.g., for the knapsack problem, the problem of scheduling to minimize weighted tardiness or the constraint shortest path problem. Hence, assuming the uniform RAM model, all these problems admit algorithms with expected polynomial running time under random perturbations. With some more effort the same result can also be obtained on a log-RAM. More details are given in a full version of this paper.

On a first view, the **Euclidean variants of TSP and Steiner tree** might look like interesting candidates for problems with polynomial smoothed complexity. Using the same techniques as in the proof of Theorem 3 one can easily prove, however, that polynomial smoothed complexity for these problem would imply an randomized fully polynomial time approximation scheme. Thus, aiming at smoothed analysis for Euclidean TSP or Steiner tree only makes sense if one believes that these problems might admit an FPAS.

One criticism of the smoothed analysis of the Simplex algorithm is that the additive perturbations destroy the zero-structure of an optimization problem as it replaces zeros with small values. See also the discussion in [24]. The same criticism applies to the zero-structure in binary programs. It turns out, however, that our probabilistic analysis in Section 2 is robust enough to deal with **zero-preserving perturbations**. In particular, we can extend our input model by allowing the adversary to declare some of the coefficients in the stochastic constraints to be fixed to zero. This way, we can extend our results to further algorithmic applications, e.g., we obtain polynomial smoothed complexity for the general assignment problem (GAP) with any fixed number of bins.

2. ANALYSIS OF THE GAP PROPERTIES

In this section, we will formally define winner, loser, and feasibility gaps and prove upper bounds on the density functions of these random variables. Before going into the details of the analysis, the term "upper bound on the density" needs some clarification as the density of a continuous variable is not uniquely defined. A continuous random variable X is defined by its *distribution* $F_X(t) = \mathbf{Pr}[X \leq t]$. In general, the *density* f_X is any non-negative function satisfying $F_X(t) = \int_{-\infty}^{t} f_X(s)ds$. Observe that the integrand is not uniquely determined. It might be redefined on any set of points of measure 0 without affecting the integral. We say that a continuous random variable X is *well-behaved* if its distribution function F_X is piecewise differentiable. In this case, X admits a piecewise continuous density function f_X which at all of its continuous points corresponds to the derivative of F_X. As usual, we ignore the trifling indeterminacy in the definition of f_X and refer to f_X as *the* density of X. In particular, the *supremum of the density of X* refers solely to the supremum over the points at which f_X is

continuous, and we say that the density is *bounded* if there exists $b \in \mathbb{R}$ such that $f_X(s) \leq b$, for every point $s \in \mathbb{R}$ at which f_X is continuous. Throughout the analysis $[n]$ denotes $\{1,\ldots,n\}$.

The winner gap. We consider an instance of a discrete optimization problem whose solutions are described by n binary variables $x_1,\ldots,x_n$. The set of feasible solution is denoted by $\mathcal{S} \subseteq \{0,1\}^n$. We assume that there are at least two feasible solutions, but otherwise $\mathcal{S}$ can be specified arbitrarily. The objective function is denoted by $c^T x$. The numbers $c_i \in \mathbb{R}$, $i = 1,\ldots,n$, are assumed to be stochastic, that is, they are treated as independent random variables following possibly different, well-behaved continuous probability distributions with bounded density. W.l.o.g., we consider a maximization problem. Let $x^* = \text{argmax}\{c^T x \,|\, x \in S\}$ denote the winner and $x^{**} = \text{argmax}\{c^T x \,|\, x \in S \setminus \{x^*\}\}$ the second best solution. The *winner gap* Δ is defined to be the difference between the objective values of a best and a second best solution, that is,

$$\Delta \;=\; c^T x^* - c^T x^{**} \quad .$$

The random variable Δ is well-behaved, i.e., Δ admits a piecewise continuous density function. This can be seen as follows. The probability space of Δ is $(c_1 \times \ldots \times c_n) \subseteq \mathbb{R}^n$. Each pair of solutions defines a hyperplane in $\mathbb{R}^n$, consisting of all points where the two solutions have equal objective function. These hyperplanes partition $\mathbb{R}^n$ into a finite number of polyhedral cells. Fix any cell $C \subseteq \mathbb{R}^n$. In this cell, x^* and x^{**} are uniquely determined. In particular, $(\Delta|C) = c^T x^* - c^T x^{**}$. Thus, the random variable $\Delta|C$ is a linear functional of the well-behaved continuous variables $c_1,\ldots,c_n$. Thus the density of $\Delta|C$ corresponds to the convolution of piecewise-continuous variables, and hence it is piecewise continuous, too. Consequently, $f_\Delta = \sum_C \mathbf{Pr}[C]\, f_{\Delta|C}$ is piecewise continuous as well, so that Δ is well-behaved. The same kind of argument applies to other gap variables that we will define in the following.

LEMMA 5 (GENERALIZED ISOLATING LEMMA). *Let ϕ_i denote the density parameter of c_i, $1 \leq i \leq n$, and $\phi = \max_i \phi_i$. For every choice of the feasible region $\mathcal{S}$ and every choice of the probability distributions of $c_1,\ldots,c_n$, the density function of Δ is bounded from above by $2\sum_{i\in[n]} \phi_i \leq 2\phi n$.*

PROOF. At first, we observe, if there is a variable x_i that takes the same value in all feasible solutions, then this variable does not affect the winner gap and it can be ignored. Thus, w.l.o.g., for every $i \in [n]$, there are at least two feasible solutions whose vectors differ in the i-th bit, i.e., with respect to the i-th variable. Under this assumption, we can define the winner gap with respect to bit position $i \in [n]$ by

$$\Delta_i \quad = \quad c^T x^* - c^T y \qquad (1)$$

with $x^* = \text{argmax}\{c^T x \,|\, x \in \mathcal{S}\}$ and $y = \text{argmax}\{c^T x \,|\, x \in \mathcal{S}, x_i \neq x_i^*\}$. In words, Δ_i is the difference between the objective value of the winner x^* and the value of a solution y that is best among those solutions that differ in the i-th bit from x^*, i.e., the best solution in $\{x \in \mathcal{S} \,|\, x_i \neq x_i^*\}$.

Clearly, the best solution, $x^* = (x_1^*,\ldots,x_n^*)$, and the second best solution, $x^{**} = (x_1^{**},\ldots,x_n^{**})$, differ in at least one bit, that is, there exists $i \in [n]$ such that $x_i^* \neq x_i^{**}$. If the best and the second best solution differ in the i-th bit then $\Delta = \Delta_i$. Thus, Δ is guaranteed to take a value also taken by at least one of the variables $\Delta_1,\ldots,\Delta_n$. We will prove upper bounds on the density functions of the variables $\Delta_1,\ldots,\Delta_n$ and use these bounds and the following lemma to obtain an upper bound on the density of the random variable Δ. Observe that the random variables $\Delta_1,\ldots,\Delta_n$ are well-behaved continuous, but there might be various kinds of dependencies among these variables.

LEMMA 6. *Let $X_1,\ldots,X_n$ and X denote well-behaved continuous random variables. Suppose X always takes a value equal to one of the values of the variables $X_1,\ldots,X_n$. Then for all $t \in \mathbb{R}$, $f_X(t) \leq \sum_{i\in[n]} f_{X_i}(t)$.*

The lemma follows directly from elementary probability theory. Therefore, we skip the proof. In the following, we will prove an upper bound of $2\phi_i$ on the density function for the random variable Δ_i, for every $i \in [n]$. Combining this bound with Lemma 6 immediately yields that the density function of Δ is bounded from above by $2\sum_{i\in[n]} \phi_i$, so that the theorem is shown.

Let us fix an index $i \in [n]$. It only remains to be shown that the density of Δ_i is bounded from above by $2\phi_i$. We partition $\mathcal{S}$, the set of feasible solutions, into two disjoint subsets $\mathcal{S}_0 = \{x \in \mathcal{S} \,|\, x_i = 0\}$ and $\mathcal{S}_1 = \{x \in \mathcal{S} \,|\, x_i = 1\}$. Now suppose all random variables $c_k, k \neq i$ are fixed arbitrarily. Obviously, under this assumption, the winner among the solutions in $\mathcal{S}_0$ and its objective value are uniquely determined as the objective values of the solutions in $\mathcal{S}_0$ do not depend on c_i. Although the objective values of the solutions in $\mathcal{S}_1$ are not fixed, the winner of $\mathcal{S}_1$ is uniquely determined as well because the unknown outcome of the random variable c_i does not affect the order among the solutions in $\mathcal{S}_1$. For $j \in \{0,1\}$, let $x^{(j)}$ denote the winner among the solutions in $\mathcal{S}_j$. We observe $\Delta_i = |c^T x^{(1)} - c^T x^{(0)}|$ because the solutions x^* and y as defined in Equation (1) cannot be contained in the same set $\mathcal{S}_j$, $j \in \{0,1\}$. Hence, Δ_i takes either the value of $c^T x^{(1)} - c^T x^{(0)}$ or the value of $c^T x^{(0)} - c^T x^{(1)}$. Observe that the random variable c_i appears as a simple additive term in both of these expressions, and the density of c_i is at most ϕ_i. Therefore, both expressions describe random variables with density at most ϕ_i as well. (Observe that this holds, regardless of whether we assume that the other variables are fixed or random numbers.) Consequently, Lemma 6 yields that the density of Δ_i is at most $2\phi_i$. This completes the proof of Lemma 5. □

Loser and feasibility gaps for a single constraint. We consider an instance of an optimization problem over n binary variables. The objective function can be fixed arbitrarily; we rank all solutions (feasible and infeasible) according to their objective value in non-increasing order. Solutions with the same objective values are ranked in an arbitrary but fixed fashion. The feasible region is described by a subset $\mathcal{S} \subseteq \{0,1\}^n$ intersected with the half-space $\mathcal{B}$ described by an additional linear constraint. W.l.o.g. the constraint is of the form $w^T x \leq t$. The set $\mathcal{S}$ and the threshold t are assumed to be fixed. The coefficients $w_1,\ldots,w_n$ correspond to independent random variables following possibly different, well-behaved continuous probability distributions with bounded density. The *winner*, denoted by x^*, is the solution with highest rank in $\mathcal{S} \cap \mathcal{B}$. The *feasibility gap* is defined by

$$\Gamma \;=\; \begin{cases} t - w^T x^* & \text{if } \mathcal{S} \cap \mathcal{B} \neq \emptyset, \text{ and} \\ \perp & \text{otherwise.} \end{cases}$$

In words, Γ corresponds to the slack of the winner with respect to the threshold t. Observe that x^* might be undefined as there is no feasible solution. In this case, the random variable Γ takes the value $\perp$ (*undefined*). The domain of Γ is $\mathbb{R}_{\geq 0} \cup \{\perp\}$. The density function f_Γ over $\mathbb{R}_{\geq 0}$ is well-behaved continuous. The function f_Γ does not necessarily integrate to 1 but only to $1 - \mathbf{Pr}[\Gamma = \perp]$. In the following, when talking about the density of Γ, we solely refer to the function f_Γ over $\mathbb{R}_{\geq 0}$, that is, we ignore the probability of the event $\{\Gamma = \perp\}$ as it is of no relevance to us.

A solution is called a *loser* if it is contained in $\mathcal{S}$ and has a higher rank than x^*, that is, the losers are those solutions from $\mathcal{S}$ that are

better than the winner (w.r.t. the ranking), but they are cut off by the constraint $w^T x \leq t$. The set of losers is denoted by $\mathcal{L}$. If there is no winner, as there is no feasible solution, then we define $\mathcal{L} = \mathcal{S}$. The *loser gap* is defined by

$$\Lambda = \begin{cases} \min\{w^T x - t \mid x \in \mathcal{L}\} & \text{if } \mathcal{L} \neq \emptyset, \text{ and} \\ \perp & \text{otherwise.} \end{cases}$$

As in the case of the feasibility gap, when talking about the density of the loser gap, we solely refer to the function f_Λ over $\mathbb{R}_{\geq 0}$ and ignore the probability of the event $\{\Lambda = \perp\}$.

Our goal is to upper-bound the densities of Γ and Λ. Observe that the solution 0^n is different from all other solutions in $\mathcal{S}$ as its feasibility does not depend on the outcome of the random coefficients $w_1, \ldots, w_n$. Suppose $0^n \in \mathcal{S}$ and 0^n has the highest rank among all solutions in $\mathcal{S}$. Then one can enforce $\Gamma = 0$ by setting $t = 0$. Similarly, one can enforce $\Lambda \to 0$ for $t \to 0$. For this reason, we need to exclude the solution 0^n from our analysis. Assuming $0^n \notin \mathcal{S}$, the following theorem shows that both the loser and the feasibility gap are likely to have polynomial size.

LEMMA 7 (SEPARATING LEMMA). *Let ϕ_i denote the density parameter of w_i, and $\phi = \max_{i \in [n]} \phi_i$. Suppose $0^n \notin \mathcal{S}$. Then the densities of Γ and Λ are bounded from above by $n \sum_{i=1}^{n} \phi_i \leq \phi n^2$.*

PROOF. We will heavily use symmetry properties between the two gaps. At first, we will prove an upper bound of ϕn on the density of the loser gap under the assumption that the ranking satisfies a certain monotonicity property. Next, we will show that the supremum of the density functions for loser and feasibility gaps are identical for worst-case choices of the threshold t. This way, the upper bound on the density of the loser gap holds for the feasibility gap as well. Then we will show that monotonicity assumption for the feasibility gap can be dropped at the cost of an extra factor n, thereby achieving an upper bound of ϕn^2 on the density of the feasibility gap. Finally, by applying the symmetry between loser and feasibility gap again, we obtain the same result for the loser gap.

The ranking among the solutions is called *monotone* if all pairs of solutions $x, y \in \mathcal{S}$, x having a higher rank than y, satisfy that there exists $i \in [n]$ with $x_i \geq y_i$. When considering the binary solution vector as subsets of $[n]$, a ranking is *monotone* if no solution that is a proper subset of another solution S is ranked lower than S. This property is naturally satisfied for maximization problems having a linear objective function with positive coefficients, but also if all solutions in $\mathcal{S}$ have the same number of ones.

LEMMA 8. *Suppose $0^n \notin \mathcal{S}$ and the ranking is monotone. Then f_Λ is bounded from above by $\sum_{i \in [n]} \phi_i$.*

PROOF. Fix $t \in \mathbb{R}$ arbitrarily. As in the proof for the winner gap, we define n random variables $\Lambda_1, \ldots, \Lambda_n$ with maximum densities $\phi_1, \ldots, \phi_n$ such that at least one of them takes the value of Λ. For $i \in [n]$, define $\mathcal{S}_i = \{x \in \mathcal{S} \mid x_i = 1\}$ and $\bar{\mathcal{S}}_i = \mathcal{S} \setminus \mathcal{S}_i$. Let $\vec{x}^{(i)}$ denote the winner from $\mathcal{S}_i$, i.e., the solution with highest rank in $\mathcal{S}_i \cap \mathcal{B}$. Now let $\mathcal{L}_i$ denote the set of losers from $\mathcal{S}_i$ with respect to $\vec{x}^{(i)}$, that is, $\mathcal{L}_i = \{x \in \mathcal{S}_i \mid x \text{ has a higher rank than } \vec{x}^{(i)}\}$. If $\vec{x}^{i}$ does not exist then we set $\mathcal{L}_i = \mathcal{S}_i$. Now define $x_{\min}^{(i)} = \operatorname{argmin}\{w^T x \mid x \in \mathcal{L}_i\}$, and

$$\Lambda_i = \begin{cases} w^T x_{\min}^{(i)} - t & \text{if } \mathcal{L}_i \neq \emptyset, \text{ and} \\ \perp & \text{otherwise.} \end{cases}$$

Observe that $\mathcal{L}_i$ is not necessarily a subset of $\mathcal{L}$ as $\vec{x}^{(i)}$ might have a lower rank than x^*. In fact, $x_{\min}^{i}$ might be feasible so that Λ_i can take negative values. The reason for this "wasteful" definition is that it yields some kind of independence that we will exploit in the following arguments.

Claim A: The density of Λ_i is at most ϕ_i. This claim can be seen as follows. The definitions above ensure $\mathcal{L}_i \subseteq \mathcal{S}_i$ while $\vec{x}^{(i)} \in \bar{\mathcal{S}}_i$. Suppose all variables $w_j, j \neq i$ are fixed arbitrarily. We prove that the density of Λ_i is bounded by ϕ_i under this assumption, and hence the same bound holds for randomly chosen $w_j, j \neq i$ as well. The winner $\vec{x}^{(i)}$ can be determined without knowing the outcome of w_i as $\vec{x}^{(i)} \in \bar{\mathcal{S}}_i$ and for all solutions in $\mathcal{S}_i$ the i-th entry is zero. Observe that $\mathcal{L}_i$ is fixed as soon as $\vec{x}^{(i)}$ is fixed, and so is $x_{\min}^{(i)}$. As a consequence, w_i is not affected by the determination of $x_{\min}^{(i)}$. As the i-th bit of $x_{\min}^{(i)}$ is set to one, the random variable Λ_i can be rewritten as $\Lambda_i = w^T x_{\min}^{(i)} - t = \kappa + w_i$, where κ denotes a fixed quantity and w_i is a random variable with density at most ϕ_i. Consequently, the density of Λ_i is bounded from above by ϕ_i.

Claim B: If $\Lambda \neq \perp$, then there exists $i \in [n]$ such that Λ takes the value of Λ_i. To show this claim, let us first assume that x^* exists and $\mathcal{L} \neq \emptyset$. Let $x_{\min} \in \mathcal{L}$ denote the *minimal loser*, i.e., $x_{\min} = \operatorname{argmin}\{w^T x \mid x \in \mathcal{L}\}$. By definition, $x_{\min}$ has a higher rank than x^*. Because of the monotonicity of the ranking, there exists $i \in [n]$ such that $x^* \in \bar{\mathcal{S}}_i$ and $x_{\min} \in \mathcal{S}_i$. From $x^* \in \bar{\mathcal{S}}_i$, we conclude $x^* = \vec{x}^{(i)}$. Consequently, $x_{\min} \in \mathcal{L} \cap \mathcal{S}_i = \mathcal{L}_i$ so that $x_{\min} = x_{\min}^{(i)}$. Hence, $\Lambda = \Lambda_i$. Now suppose x^* does not exist. Then $\mathcal{L} = \mathcal{S}$ and $\mathcal{L}_i = \mathcal{S}_i$, for all $i \in [n]$. Thus, there exists $i \in [n]$ with $x_{\min} = x_{\min}^{i}$ because $\mathcal{S} = \bigcup_{i \in [n]} \mathcal{S}_i$ as $0^n \notin \mathcal{S}$. Finally, if $\mathcal{L} = \emptyset$ then the claim follows immediately as $\Lambda = \perp$.

Now applying Lemma 6 to the Claims A and B immediately yields the lemma. □

The following lemma shows that upper bounds on the density function of the loser gap also hold for the feasibility gap and vice versa. For a given threshold t, let $R(t) \subseteq \mathbb{R}_{\geq 0}$ denote the set of points at which the distribution functions of $\Lambda(t)$ and $\Gamma(t)$ are differentiable. As $\Lambda(t)$ and $\Gamma(t)$ are well-behaved continuous, the points in $\mathbb{R} \setminus R(t)$ have measure 0 and, hence, can be neglected.

LEMMA 9. *Suppose $0^n \notin \mathcal{S}$. Then $\sup_{t \in \mathbb{R}} \sup_{s \in R(t)} f_{\Gamma(t)}(s) = \sup_{t \in \mathbb{R}} \sup_{s \in R(t)} f_{\Lambda(t)}(s)$.*

PROOF. (Sketch) We take an alternative view on the given optimization problem. We interpret the problem as a bicriteria problem. The feasible region is defined by the set $\mathcal{S}$. On one hand, we seek for a solution from $\mathcal{S}$ whose rank is as high as possible. On the other hand, we seek for a solution with small weight, where the *weight* of a solution $x \in \mathcal{S}$ is defined by the linear function $w^T x$. A solution $x \in \mathcal{S}$ is called *Pareto-optimal* if there is no other solution $y \in \mathcal{S}$ so that y improves on x in rank and weight simultaneously.

We show that winners and minimal losers of the original optimization problem correspond to Pareto-optimal solutions of the bicriteria problem. Intuitively, we can imagine that all Pareto-optimal solutions are mapped onto a horizontal line such that a Pareto-optimal solution x is mapped to the point $w^T x$. Then $\Gamma(t)$ is the distance from the point t on this line to the closest Pareto point *left* of t (i.e., less than or equal to t), and $\Lambda(t)$ is the distance from t to the closest Pareto point *strictly right* of t (i.e., larger than t). Now let f be a measure over $\mathbb{R}$ describing the density of the Pareto points on the line. Let $t' \in \mathbb{R}$ denote the point maximizing f. Then $f(t')$ is the joint supremum of the densities of Γ and Λ when setting $t = t'$, and this choice of t maximizes the supremum of the density functions, so that the lemma follows. □

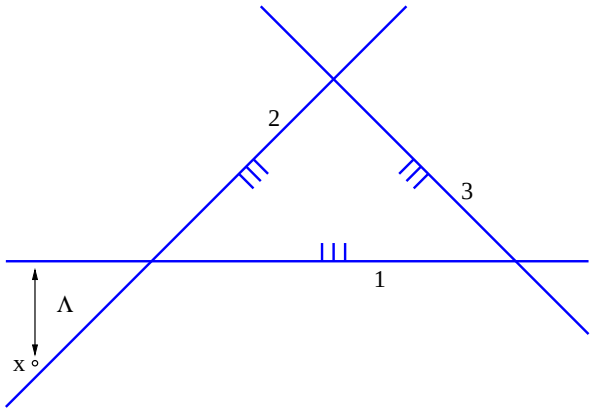

Figure 1: The picture shows three constraints of the form $-x_1 \leq t_1$, $x_1 - x_2 \leq t_2$, and $x_1 + x_2 \geq t_3$. Suppose x is the only loser. The loser gap of x is $\max\{t_j - w_j^T x \mid w_j^T x \leq t_j\} = t_1 - (-1,0)^T x$.

Combining the two lemmas, we observe that the density of the feasibility gap $\Gamma(t)$ is at most $\sum_{i\in[n]} \phi_i$, provided that the ranking is monotone and $0^n \notin \mathcal{S}$. Next we extend this result towards general kinds of rankings by breaking the original problem in subproblems. We partition $\mathcal{S}$ into the sets $\mathcal{S}^{(k)} = \{x \in \mathcal{S} \mid \sum_i x_i = k\}$, for $1 \leq k \leq n$. Observe that each of these sets contains only solutions with the same number of ones, and hence satisfies the monotonicity condition. Let $\Gamma^{(k)}(t)$ denote the feasibility gap over the set $\mathcal{S}^{(k)}$. By Lemma 8, the density of $\Gamma^{(k)}(t)$ is at most $\sum_{i\in[n]} \phi_i$, for every $t \in \mathbb{R}$. Furthermore, $\Gamma(t)$ takes the value of one of the variables $\Gamma^{(k)}(t)$, $1 \leq k \leq n$ because the winner of one of the subproblems is the winner of the original problem. As a consequence of Lemma 6, the density of $\Gamma(t)$ is at most $n\sum_{i\in[n]} \phi_i$, for every continuous point $t \in \mathbb{R}$. Let us remark that such a kind of argument cannot directly be applied to the loser gap. By applying Lemma 9, however, the bound for the feasibility gap holds for the loser gap as well. Hence, Lemma 7 is shown. □

Loser and Feasibility gap for multiple constraints. Assume there are $k \geq 2$ stochastic constraints. W.l.o.g., these constraints are of the form $w_j^T x \leq t_j$, for $j \in [k]$, and the sets of points satisfying these constraints are $\mathcal{B}_1, \ldots, \mathcal{B}_k$, respectively. We generalize the definition of feasibility and loser gap as follows. Given a set of solutions $\mathcal{S} \subseteq \{0,1\}^n$ and a ranking, the *winner* x^* is the highest ranked solution in $\mathcal{S} \cap \mathcal{B}_1 \cap \ldots \cap \mathcal{B}_k$. The *feasibility gap for multiple constraints* is the minimal slack of x^* over all stochastic constraints, that is, $\Gamma = \min_{j\in[k]}\{t_j - w_j^T x^*\}$, if x^* exists, and $\Delta = \perp$, otherwise. The set of *losers* $\mathcal{L}$ consists of all solutions from $\mathcal{S}$ that have a rank higher than x^*. Observe that a loser only needs to be infeasible with respect to one out of the k constraints. In particular, it is not true that the values of all losers are likely to be far away from all thresholds t_j, $j \in [k]$; not even if we consider only those constraints for which the respective losers are infeasible. Fortunately, however, we do not need such a property in the application of the loser gap. For every loser, one only needs a single constraint that renders the loser infeasible. Therefore, we define the *loser gap for multiple constraints* by $\Lambda = \min_{x\in\mathcal{L}} \max_{j\in[k]}\{w_j^T x - t_j\}$, if $\mathcal{L} \neq \emptyset$, and $\Lambda = \perp$, otherwise. Figure 1 illustrates this definition.

LEMMA 10. *Let ϕ denote the maximum density parameter of all coefficients in the stochastic constraints. Suppose $0^n \notin \mathcal{S}$. Then $\mathbf{Pr}[\Gamma < \varepsilon] \leq \varepsilon k\phi n^2$ and $\mathbf{Pr}[\Lambda < \varepsilon] \leq \varepsilon k\phi n^2$, for all $\varepsilon \in \mathbb{R}_{\geq 0}$.*

PROOF. First we show the bound for the feasibility gap. Suppose $\Gamma \leq \varepsilon$, for some $\varepsilon \in \mathbb{R}_{\geq 0}$. Then there exists $j \in [k]$ with $t_j - w_j^T x^* \leq \varepsilon$. Thus,

$$\mathbf{Pr}[\Gamma \leq \varepsilon] \leq \sum_{j\in[k]} \mathbf{Pr}\left[t_j - w_j^T x^* \leq \varepsilon\right] .$$

Now, for every individual $j \in [k]$, we can apply the Separating Lemma assuming that the set of feasible solutions with respect to all other constraints is fixed as the coefficients in this constraint are stochastically independent from the other constraints. This way, we obtain $\mathbf{Pr}[\Gamma \leq \varepsilon] \leq k \cdot \varepsilon\phi n^2$.

Next, we turn our attention to the loser gap. Unfortunately, we cannot generalize the bound on the loser gap from one to multiple constraints in the same way as we generalized the feasibility gap as the loser gap for multiple constraints does not correspond to the minimal loser gap over the individual constraints. Instead we will make use of the result for the feasibility gap established above. Assume $\Lambda \leq \varepsilon$, for some $\varepsilon \in \mathbb{R}_{\geq 0}$. Then there exists a loser x satisfying $\forall j \in [k] : w_j^T x - t_j \leq \varepsilon$. Let x_L denote the loser with this property that is ranked highest. Consider a relaxed variant Π' of the given optimization problem Π where the thresholds of all stochastic constraints are increased by ε, i.e., we have constraints $w_j^T x \leq t_j + \varepsilon$, $j \in [k]$. Observe that x_L is feasible in the relaxed problem Π' and, by the definition of x_L, no higher ranked solution is feasible. Thus, x_L is the winner of Π'. Since $t_j < w_j^T x_L \leq t_j + \varepsilon$ for some $j \in [k]$, the feasibility gap Γ' of the relaxed problem is smaller than ε. Hence, $\Lambda \leq \varepsilon$ implies $\Gamma' \leq \varepsilon$. Finally, applying the bound $\mathbf{Pr}[\Gamma' \leq \varepsilon] \leq \varepsilon k\phi n^2$ derived in the first part of the proof yields $\mathbf{Pr}[\Lambda \leq \varepsilon] \leq \varepsilon k\phi n^2$. □

3. PROOF OF THEOREM 1

First, suppose the only stochastic expression is the objective function. We reveal b bits after the binary point of each coefficient c_i ($1 \leq i \leq n$), and assume that all following bits are 0. This corresponds to rounding down all numbers to multiples of 2^{-b} causing an absolute rounding error of less than 2^{-b} for each number.

LEMMA 11. *Let ϕ denote the maximum density parameter over all coefficients in the objective function. When revealing b bits after the binary point of each coefficient in the objective function then the winner is uniquely determined with probability at least $1 - 2^{-b}n^2\phi$.*

PROOF. Let $\lfloor c \rfloor$ be the vector that is obtained by rounding each entry c_i of c down to the next multiple of 2^{-b}. Consider any two solutions $x, x' \in \mathcal{S}$. We have $|(c^T x - c^T x') - (\lfloor c \rfloor^T x - \lfloor c \rfloor^T x')| = |(c - \lfloor c \rfloor)^T (x - x')| < n2^{-b}$. Hence, if the winner gap Δ (with respect to the exact coefficients $c_1, \ldots, c_n$) is at least $n2^{-b}$ then the rounding can not affect the optimality of the winner. In this case the winner is uniquely determined after revealing only b bits of each coefficient c_i. Let ϕ_Δ denote the supremum of the density of Δ. Then $\mathbf{Pr}[\Delta < x] \leq x\phi_\Delta$ for all $x \in \mathbb{R}_{\geq 0}$. Using Lemma 5 and setting $x = n2^{-b}$ yields $\mathbf{Pr}\left[\Delta < n2^{-b}\right] \leq 2^{-b}n^2\phi$. □

Next suppose only some of the constraints are stochastic, and the objective function is adversarial. Let k' denote the number of stochastic constraints. W.l.o.g. the constraints are of the form $w_j^T x \leq t_j$, $j \in [k']$. We reveal b bits after the binary point of each coefficient and round down.

LEMMA 12. *Let ϕ denote the maximum density parameter over all coefficients in the stochastic constraints. When revealing b bits after the binary point of each coefficient, then the winner is uniquely determined with probability at least $1 - 2^{-b}k'n^3\phi$.*

PROOF. Observe that, due to rounding, infeasible solutions might become feasible whereas feasible solutions stay feasible. To ensure that the winner is uniquely determined it suffices to upper bound the maximum possible error in each constraint that is caused by the rounding. If this error is smaller than the loser gap then rounding cannot change the feasibility status of any loser, i.e., all infeasible solutions that have rank higher than the winner stay infeasible.

In order to apply the bound on the loser gap given in Lemma 10, let us first assume $0^n \notin \mathcal{S}$. The error for every solution with respect to any constraint is at most $n2^{-b}$. The definition of the loser gap for multiple stochastic constraints states that for every loser x there is a constraint $j \in [k']$ such that $w_j^T x - t_j \geq \Gamma$. Therefore, if $\Gamma \geq n2^{-b}$ then every loser stays infeasible after rounding. Applying Lemma 10, the probability for this event is at least $1 - k'\phi n^3/2^b$.

The solution 0^n can influence the loser gap in two ways. At first, 0^n can be a loser and thus decrease the loser gap. However, rounding the coefficients w_i does not change the objective value of this solution. Thus, 0^n stays infeasible under rounding and the loser gap with respect to all other solutions in unaffected. At second, 0^n might be the winner, which would result in a different loser set than under the assumption $0^n \notin \mathcal{S}$. In this case, however, 0^n has a higher rank than the previous winner so that the set of losers can only shrink. Therefore, the loser gap cannot decrease because of 0^n and, hence, the argument above about the feasibility of the other solutions remains valid. □

Now suppose some constraints and the objective function are stochastic. Let $k = k' + 1$ denote the number of stochastic expressions. The probability that winner and loser gap are sufficiently large, as described in the two lemmas above, is $1 - k'\phi n^3/2^b - \phi n^2/2^b \geq 1 - k\phi n^3/2^b$. This implies that the winner is uniquely determined when revealing $O(\log(k\phi n))$ bits, **whp**. This completes the proof of Theorem 1. □

4. PROOF OF THEOREM 3

At first, we prove that a randomized pseudopolynomial algorithm implies polynomial smoothed complexity. We design an algorithm with polynomial smoothed complexity calling the pseudopolynomial algorithm with higher and higher precision until the optimal solution is found. Due to space limitations, we only present the core of the algorithm and its analysis, namely we present how to compute a certified winner when only a bounded number of bits per input number is available. The algorithm has available b bits after the binary point of each random coefficient and either outputs the true winner or, if it cannot compute such a winner as it needs more bits, it reports a failure.

Certifying optimality. Suppose only the objective function is stochastic. W.l.o.g., consider a maximization problem with objective function $c^T x$. Let $\lfloor c_i \rfloor$ denote the coefficient c_i rounded down to the next smaller multiple of 2^{-b}. At first, we compute the optimal solution x' for the problem with the rounded coefficients $\lfloor c_1 \rfloor, \ldots, \lfloor c_n \rfloor$. To check, whether x' is optimal with respect to the original cost vector c, we generate another vector $\bar{c}$ of rounded coefficients. This time the rounding depends on x'. For all i with $x'_i = 1$, we set $\bar{c}_i := \lfloor c_i \rfloor$ and for all i with $x'_i = 0$, we set $\bar{c}_i := \lceil c_i \rceil = \lfloor c_i \rfloor + 2^{-b}$. Observe, the function $\delta(x) = c^T x - \bar{c}^T x$ is maximal for $x = x'$. Next, we compute the optimal solution x'' for the problem with the vector $\bar{c}$. If $x' = x''$ then x' simultaneously maximizes $\delta(x)$ and $\bar{c}^T x$. Consequently, it maximizes $\bar{c}^T x + \delta(x) = c^T x$ as well and, hence, x' corresponds to the true winner x^*. Thus, the algorithm outputs x' as a certified winner if $x' = x''$ and reports a failure, otherwise. Observe, if the winner gap is large enough so that the winner is uniquely determined in the sense of Lemma 11, then the algorithm will always compute a certified winner. Hence, the probability that the algorithm is successful corresponds to the bound given in Lemma 11.

Certifying feasibility. Now we show how to deal with stochastic constraints. W.l.o.g, we assume that all stochastic constraints are of the form $w_j^T x \leq t_j$, $1 \leq j \leq k'$. As described in Lemma 12 all coefficients are rounded down to the next multiple of 2^{-b}, and we compute a certified winner with respect to the rounded coefficients. Let x' denote the winner with respect to the rounded coefficients. Observe, that we rounded in such a way that feasible solutions stay feasible. However, we have to detect infeasible solutions that might become feasible due to the rounding and displace the true winner. Hence, we need to check whether x' is indeed feasible with respect to the original constraints. This would be trivial if the exact values of all constraint vectors $w_1, \ldots, w_k$ were available. However, we want to check the feasibility with only knowing b bits after the binary point of each coefficient. Let $\lfloor w_j \rfloor$ denote the vector that is obtained by rounding down each entry of w_j to the next multiple of 2^{-b}. Assume solution x is infeasible w.r.t. the j-th constraint and becomes feasible due to rounding. Then $\lfloor w_j \rfloor^T x \leq t_j < w_j^T x$ and hence $t_j - \lfloor w_j \rfloor^T x < w_j^T x - \lfloor w_j \rfloor^T x \leq n2^{-b}$, i.e. x has slack less than $n2^{-b}$ for constraint j. Our verifier will use this property. It classifies x' as possibly infeasible if it has slack less than $n2^{-b}$ for any of the k constraints. There are two reasons why the verifier may fail. At first, there might be a loser that appears to be feasible because of the rounding. As seen in the proof of Lemma 12, however, this can happen only if the loser gap is smaller than $n2^{-b}$. At second, by mistake the true winner might have been rejected as its slack is less than $n2^{-b}$. This can happen only if the feasibility gap is smaller than $n2^{-b}$. Applying Lemma 10 yields that the probability that one of these events happen is at most $2k'2^{-b}\phi n^3$.

Now let us briefly sketch the missing details of the algorithm and its analysis. Until now we did not specify how the optimal solution for the rounded coefficient is actually computed. For this purpose, we use the pseudopolynomial algorithm. The optimal solution is found when $b = O(\log(\phi nk))$, **whp**. Hence, the pseudopolynomial algorithm has to deal with numbers described by $O(\log(\phi nk))$ bits so that its running time is bounded by $2^{O(\log(\phi nk))} = \text{poly}(\phi nk) = \text{poly}(\phi N)$, **whp**.

Finally, we need to show that polynomial smoothed complexity implies the existence of a randomized pseudopolynomial algorithm. Polynomial smoothed complexity implies that there exists an algorithm $\mathcal{A}$ and a polynomial $P(n,\phi)$ such that the probability that the running time of $\mathcal{A}$ exceeds $P(n,\phi)$ is at most $\frac{1}{4}$. Since we are aiming for a pseudopolynomial time algorithm we can assume that all numbers in the stochastic expressions are integers. Let M denote the largest absolute value of those integers. The idea is to perturb all numbers only slightly such that the cumulative error in each expression is less than $\frac{1}{2}$. To adapt the problem to the smoothed analysis framework we first scale all input numbers in the stochastic constraints by M^{-1}. Then we generate a random perturbation and test if any number has changed by more than $(2nM)^{-1}$. In that case we output FAILURE. Otherwise we call the perturbation proper and run $\mathcal{A}$. If $\mathcal{A}$ has not completed after $P(n,\phi)$ time steps we stop $\mathcal{A}$ and output FAILURE. Let Q be the event that the perturbation is proper. There is a constant c, depending on perturbation model f, such that $\mathbf{Pr}[Q] \geq \frac{1}{2}$ when setting $\phi = 4cn^2kM$. Then the probability of success is

$$\mathbf{Pr}[Q \wedge (T \leq P(n,\phi))] \;\geq\; \mathbf{Pr}[Q] - \mathbf{Pr}[T > P(n,\phi)] \;\geq\; \frac{1}{4}\ .$$

The running time of this algorithm is pseudopolynomial because $\phi = O(Mn^2k)$. Hence, $\Pi_u \in \mathsf{ZPP}$. This completes the proof of Theorem 3. □

5. RELATIONSHIP TO CONDITION NUMBERS

To obtain a finer analysis of algorithms than that provided by worst-case complexity, one should find a way of distinguishing hard problem instances from easy ones. A natural approach is to find a quantity indicating the difficulty of solving a problem instance. In Numerical Analysis and Operations Research it is common to bound the running time of an algorithm in terms of a *condition number* of its input. The condition number is typically defined to be the sensitivity of the solution for a problem instance to slight perturbations of the input. For example, Renegar [20, 21, 22] presents a variant of the primal interior point method and describes its running time as a function of the condition number. Remarkably, his running time bound depends only logarithmically on the condition number. Dunagan, Spielman, and Teng [7] study this condition number in the smoothed analysis framework. Assuming Gaussian ϕ-perturbations, the condition number can be bounded by a function that is polynomial in ϕ. Thus, the running time of Renegar's algorithm depends only logarithmically on the density parameter ϕ. In contrast, the running time bound of the Simplex algorithm presented by Spielman and Teng in [24] is polynomial in ϕ.

In [25], Spielman and Teng propose to extend the condition number towards discrete optimization problems in order to assist the smoothed analysis of such problems. As a natural definition for the condition number of a discrete function they suggest *the reciprocal of the minimum distance of an input to one on which the function has a different value*. In fact, the minimum of winner, loser, and feasibility gap is a lower bound on the amount by which the coefficients of a binary optimization problem needs to be altered so that the winner, i.e., the solution taking the optimal value, changes. Let us define the reciprocal of this minimum to be the *condition number for binary optimization problems*. This allows us to summarize our analysis in an alternative way. Our probabilistic analysis in Section 2 shows that the condition number is bounded polynomially in the density parameter ϕ. Furthermore, in Section 4, we proved that a problem with pseudopolynomial worst-case complexity admits an algorithm whose running time is bounded polynomially in the condition number. Combining these results, we obtained algorithms whose smoothed complexity depends in a polynomial fashion on the density parameter ϕ. Let us remark that this kind of dependence is best possible for NP-hard optimization problems, unless there is a subexponential time algorithm for NP-complete problems. In particular a running time bound logarithmic in ϕ would yield a randomized algorithm with polynomial worst-case complexity.

6. ACKNOWLEDGMENTS

The authors wish to thank Uriel Feige who pointed out the connection of our work to the Isolating Lemma, and an anonymous referee for drawing the connection to the condition numbers.

7. REFERENCES

[1] C. Banderier, R. Beier and K. Mehlhorn. Smoothed Analysis of three combinatorial problems. *Proc. 28th Intern. Symp. on Math. Foundations of Computer Science, 198-207, 2003.*

[2] A. Blum and J. Dunagan. Smoothed Analysis of the Perceptron Algorithm. *Proc. of the 13th Annual ACM-SIAM Symp. on Discrete Algorithms*, 905-914, 2003.

[3] L. Becchetti, S. Leonardi, A. Marchetti-Spaccamela, G. Schäfer and T. Vredeveld. Average Case and Smoothed Competitive Analysis of the Multi-Level Feedback Algorithm. *Proc. of the 44th IEEE Symp. on Foundations of Computer Science, 462-471, 2003.*

[4] F. Barahona and W. R. Pulleyblank. Exact arborescences, matchings and cycles. *Discrete Applied Mathematics, 16, 617-630, 1998.*

[5] R. Beier and B. Vöcking. Random knapsack in expected polynomial time. *Proc. of the 35th ACM Symp. on Theory of Computing, 232-241, 2003.*

[6] R. Beier and B. Vöcking. Probabilistic analysis of knapsack core algorithms. *Proc. of the 14th Annual ACM-SIAM Symp. on Discrete Algorithms*, 2004.

[7] J. Dunagan, D. A. Spielman, and S.-H. Teng. Smoothed analysis of Renegar's condition number for linear programming. Available at http://arxiv.org/abs/cs.DS/0302011, 2002.

[8] V. Damerow, F. Meyer auf der Heide, H. Räcke, C. Scheideler and C. Sohler. Smoothed motion complexity. *Proc. of the 11th European Symp. on Algorithms, 2003.*

[9] M. E. Dyer and A. M. Frieze. Probabilistic analysis of the multidimensional knapsack problem. *Mathematics of Operations Research 14(1)*, 162-176, 1989.

[10] M. R. Garey and D. S Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* Freeman, 1979.

[11] M. X. Goemans and R. Ravi. The Constrained Minimum Spanning Tree Problem. *Fifth Scandinavian Workshop on Algorithm Theory, 66-75, 1996.*

[12] A. Goldberg and A. Marchetti-Spaccamela. On finding the exact solution to a Zero-One Knapsack Problem. *Proc. of the 16th ACM Symp. on Theory of Computing*, 359-368, 1984.

[13] L. A. Levin. Average Case Complete Problems. *SIAM J. Cornput.*, 15(1):285-286, 1986.

[14] G. S. Lueker. On the average difference between the solutions to linear and integer Knapsack Problems. *Applied Probability - Computer Science, the Interface*, 1, Birkhauser, 1982.

[15] G. S. Lueker. Average-case analysis of Off-Line and On-Line Knapsack Problems. *J. of Algorithms*, 19, 277-305, 1998.

[16] G. S. Lueker. Exponentially small bounds on the expected optimum of the Partition and Subset Sum Problems. *Random Structures and Algorithms*, 12, 51-62, 1998.

[17] K. Mehlhorn and M. Ziegelmann. Resource Constrained Shortest Paths. *Proc. 8th European Symp. on Algorithms, 326-337, 2000.*

[18] K. Mulmuley, U. Vazirani and V.V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica 7(1):105-113, 1987.*

[19] C. H. Papadimitriou and M. Yannakakis. The complexity of tradeoffs, and optimal access of web sources. *Proc. of the 41st IEEE Symp. on Foundations of Computer Science, 86-92 , 2000.*

[20] J. Renegar. Some perturbation theory for linear programming. Math. Programming, 65(1, Series A), 73-91, 1994.

[21] J. Renegar. Incorporating condition measures into the complexity of linear programming. *SIAM J. Optim., 5(3):506-524, 1995.*

[22] J. Renegar. Linear programming, complexity theory and elementary functional analysis. Math. Programming, 70(3, Series A), 279-351, 1995.

[23] H. M. Salkin. *Integer Programming.* Addison-Wesley, Reading, Mass., 1975.

[24] D. A. Spielman and S.-H. Teng. Smoothed Analysis of algorithms: why the Simplex Algorithm usually takes polynomial time. *Proc. of the 33rd ACM Symp. on Theory of Computing*, 296-305, 2001.

[25] D. A. Spielman and S.-H. Teng. Smoothed Analysis: Motivation and discrete models. *Proc. of WADS 2003*, 256-270, 2003.