Electronic Colloquium on Computational Complexity - Reports Series 1996 - available via:

ECCC
TR96-043

FTP: ftp.eccc.uni-trier.de:/pub/eccc/
WWW: http://www.eccc.uni-trier.de/eccc/
Email: ftpmail@ftp.eccc.uni-trier.de with subject 'help eccc'

On polynomial time approximation schemes and approximation preserving reductions

Edmund Ihler*

Abstract

We show that a fully polynomial time approximation scheme given for an optimization problem can always be simply modified to a polynomial time algorithm solving the problem optimally if the computation model is the deterministic Turing Machine or the logarithmic cost RAM and if the range of the error bound is the rational numbers or a subset $(0, b)$.

Moreover, we prove that a P-reduction is not necessarily an A-reduction for some suitable error bound transformation but we give a sufficient criteria.

keywords: approximation, combinatorial problems, computational complexity, reductions

1 Introduction

Combinatorial optimization problems differ considerably with respect to their approximability. In the past years progress was made toward a unified theory of approximation complexity. The successful theory of NP-completeness for decision problems (see e.g. [5]) was a basis for this development. But reductions which compare optimization problems with respect to their degree of approximability are more complex than a Turing reduction. Much work has been done to find suitable types of reductions to classify NP-hard optimization problems by their degree of approximability (for a detailled overview, see e.g. [2], [3], [6], [7]). Alternativley, a purely syntactical way to partition the class of combinatorial optimization problems with respect to these degrees has been partially successful ([8], [9], [11], [12]). A breakthrough came with a new technique to prove that several famous problems are NP-hard to approximate (see overview [1]).

For an NP-hard optimization problem the best we can get is a fully polynomial time approximation scheme (*fptas*). Such a scheme A computes for each extended input (I, ϵ) a solution which differs at most by an error of ϵ from the optimum. The runtime of A is polynomially bounded in both the encoding length $|I|$ and the magnitude of $\frac{1}{\epsilon}$. It is convenient to formulate the whole framwork using the notion of *algorithm* without any specific computation model. However, caution is necessary in this approach because the range $\mathbb{E}$ of the desired error bound ϵ is

*Edmund Ihler, GMD, SET.FIT, Schloß Birlinghoven, D-53754 Sankt Augustin, Germany; Edmund.Ihler@gmd.de

critical. If the algorithm is not able to read each allowed ϵ within the required time bounds, the notion of fptas degenerates. In this case each fptas is able to compute an optimal solution for each input. In Section 3, we prove this fact for the deterministic Turing Machine and the logarithmic cost RAM computation model if the allowed error bound range is the set of all rational numbers or any subset $(0, b)$. We also propose a reasonable error range for these computation models. Unit cost models are interesting for approximation algorithms. But for the theory of the approximation complexity of optimization problems this seems not to be the suitable computation model.

Unfortunately, an fptas for an optimization problem is usually not achievable. Sometimes a polynomial time approximation scheme (*ptas*) is available which is not full, i.e. the runtime of the algorithm is only polynomially bounded in $|I|$ for each fixed error bound ϵ. But in practice it is even hard to find a polynomial time approximation algorithm (*apx*) which computes solutions which differ only by a constant bounded relative error from the optimum. Obviously each fptas is a ptas and each ptas is an apx for some arbitrary fixed ϵ. These three degrees of approximabiliy are well established today. The corresponding approximation preserving reductions are the F-, P- and A-reduction. This means, e.g. an A-reduction between two optimization problems Π and Π' carries over each apx for Π' to an apx for Π. One might conjecture that each P-reduction is automatically an A-reduction (see ([7], Proposition 3.5)). But we will prove in Section 4 that this is not the case in general. Nevertheless, we give a simple criteria when this conjecture is true.

Recently, a slightly extended version of the P-reduction was introduced in literature by [2]. This PTAS-reduction is also an A-reduction. Unfortunately, the successful L-reduction introduced in [12] for MAX SNP-completeness is not a PTAS-reduction. Nevertheless, an L-reduction is both a P-reduction and an A-reduction. The problem with PTAS-reduction is that the error bound transformation implied by L-reduction is not surjective relative to the error bound range $(0, 1)$ used for PTAS-reductions. Moreover, the range $(0, 1)$ is not suitable for apx as it does not cover an apx with an error bound ≥ 1.

2 Basic Definitions

In general an *optimization problem* is a tuple $\Pi = (\mathcal{I}, \mathcal{C}, cost, \lesseqgtr)$: $\mathcal{I} \subseteq \Sigma^*$ is the set of *inputs* over some alphabet Σ. For each input there is a set of *(feasible) solutions* $\mathcal{S}_\Pi(I)$. Each pair of input and solution (I, S) is called *configuration*. $\mathcal{C}$ is the set of configurations. Function $cost : \mathcal{C} \rightarrow \mathbb{R}$ is called *cost function* (often called objective function). The *problem type* $\lesseqgtr$ defines whether Π is a *minimization problem*, i.e. $\lesseqgtr = \leq$, or a *maximization problem*, i.e. $\lesseqgtr = \geq$. For each input I a solution S is called *optimal solution*, if for each configuration (I, S') we have $cost(I, S) \lesseqgtr cost(I, S')$. We denote the cost for an optimal solution of input I by $opt_\Pi(I)$. This definition of an optimization problem is equivalent to the standard definition, e.g. in [5] or [13]. W.l.o.g. we assume that $\mathcal{S}_\Pi(I)$ is always non empty and that there exists an optimal solution for each input I. This simplifies the following definitions but is not really necessary (see [6]).

From the theoretical point of view so called NP-*optimization problems* are of interest. These are combinatorial problems $\Pi = (\mathcal{I}, \mathcal{C}, cost, \lessgtr)$, where $\mathcal{I}$ and $\mathcal{C}$ are recognizable in polynomial time, for each configuration (I, S) the encoding length $|S|$ is polynomially bounded by the encoding length $|I|$, $cost$ is computable in polynomial time, and each $cost(I, S)$ is in $\mathbb{N}$, the set of positive natural numbers. This definition was originated in [14] and is equivalent to the definitions in e.g. [2], [4], [7], [8], [11].

The error of a solution is used to describe how far this solution is from the optimal solution. The most general definition of an *error function* $\mathcal{E}_\Pi$ requires only that for a configuration (I, S) the error $\mathcal{E}_\Pi(I, S)$ is zero if and only if S is an optimal solution of I. We omit the subscript Π in all of our notations if there is no danger of confusion. The most common error function is the *relative error* $\mathcal{E}^{rel}$. For minimization problems this error function is defined by $\mathcal{E}^{rel}(I, S) = \frac{cost(I,S)}{opt(I)} - 1$.

Definition 1 : A *fully polynomial time approximation scheme (fptas)* A for an optimization problem $\Pi = (\mathcal{I}, \mathcal{C}, cost, \lessgtr)$ with error function $\mathcal{E}$ and error bound range $\mathbb{E}$ computes for each extended input $(I, \epsilon) \in \mathcal{I} \times \mathbb{E}$ a solution $S \in \mathcal{S}(I)$, such that $\mathcal{E}(I, S) \leq \epsilon$. There exists a polynomial p bounding for each extended input $(I, \epsilon) \in \mathcal{I} \times \mathbb{E}$ the runtime $t_A(I, \epsilon)$ of A by $t_A(I, \epsilon) \leq p(|I|, 1/\epsilon)$.

The explicit definition of an error bound range $\mathbb{E}$ is not common in the approximation framework. But there is one important problem with respect to the underlying computation model and the notion of fptas that we want to discuss in the following section.

3 Error bound range and fptas

Consider the standard Turing machine model TM, where each natural number n is encoded by its binary representation $bin(n)$, or the logarithmic cost random access machine model log-RAM, where reading a natural number n costs at least $\log(n)$ time. For these computation models the range of error bound ϵ for an *fptas* is critical: If $\mathbb{E}$ is the set of positive rational numbers $\mathbb{Q}^+$ or even a subset $(0, b) \subset \mathbb{Q}^+$, then the notion of fptas degenerates. But this kind of error bound range typically occurs in the literature. Let us have a closer look.

First, we want to ensure the lowest possible read complexity for the rationals in the considered computation models: All non-negative rational numbers can be represented by pairs (n, d) of natural numbers. We denote all these pairs where n and d have no common divisor by $\mathcal{Q}^+$. For the TM we assume that for an extended input (I, ϵ) where $\epsilon = \frac{n}{d}$, the string $bin(n)bin(d)enc(I)$ is on the tape and that the head is located over the leftmost symbol, where $enc(I)$ is the encoding of I and $(n, d) \in \mathcal{Q}^+$. For the log-RAM we assume a sequence $n, d, a_1, \ldots, a_{l_I}$ of natural numbers in the RAM-registers, where $a_1, \ldots, a_{l_I}$ encodes I and $(n, d) \in \mathcal{Q}^+$.

Theorem 1 *Let A be an fptas for an optimization problem $(\mathcal{I}, \mathcal{C}, cost, \lessgtr)$ with error function $\mathcal{E}$ and error bound range $\mathbb{E} = \mathbb{Q}^+$ or any interval $(0, b) \subset \mathbb{Q}^+$. Let the underlying computation model be the TM or log-RAM model. Then A can be modified to an algorithm with a time complexity polynomially bounded in $|I|$ that computes for each $I \in \mathcal{I}$ an optimal solution.*

Proof: We will show first that for each b there are rationals $\epsilon = \frac{n}{d}$ with $\epsilon \in (0, b)$ and $(n, d) \in \mathcal{Q}^+$, where the time complexity of reading n does not allow to read n completely.

To see this, fix an arbitrary $I \in \mathcal{I}$. Let 1^i be the string $11 \cdots 1$ consisting of i times 1. Consider the series $\epsilon_{m,i} = \frac{n_i}{m(n_i+1)}$, where $bin(n_i) = 1^i$. There exist arbitray large prime numbers m. This means that for each fixed $i \in \mathbb{N}$ there exists an m such that $\epsilon_{m,i} \in (0, b)$ and $(n_i, m(n_i + 1)) \in \mathcal{Q}^+$. On the other hand the fully polynomial time complexity guarantees constants $c, k \in \mathbb{N}$ such that the runtime $t_A(I, \epsilon)$ for each extended input is bounded by $t_A(I, \epsilon) \leq c + (\frac{1}{\epsilon})^k$ for all $\epsilon \in \mathbb{E}$. The time complexity of reading n_i is $\log n_i = i$. Thus $i \leq c + (\frac{1}{\epsilon})^k$ is a necessary condition for reading n_i completely. But for each fixed m, $\lim_{i\to\infty} i = \infty$ whereas $\lim_{i\to\infty}(c + (\frac{1}{\epsilon})^k)$ $= c + m \cdot \lim_{i\to\infty} \frac{n_i+1}{n_i} = c + m = const.$ Altogether this means that for each I there exists a prime number m_I and a natural number i_I such that $\epsilon_{m_I,i_I} \in (0, b)$ and $(n_{i_I}, m_I(n_{i_I} + 1)) \in \mathcal{Q}^+$ where n_{i_I} cannot be read completely by A. To construct this ϵ_{m_I,i_I}, first choose any prime number $m_I > \frac{1}{b}$ and then choose i_I sufficiently big.

For the log-RAM model this means that A cannot read n_{i_I} at all. This means that it is not possible for A to read any ϵ, because it does not know about the read complexity in advance. Thus A computes for each extended input (I, ϵ) an optimal solution S completely independent of ϵ. The modification of A for the log-RAM model is to extend each input I by an arbitrary fixed error bound ϵ.

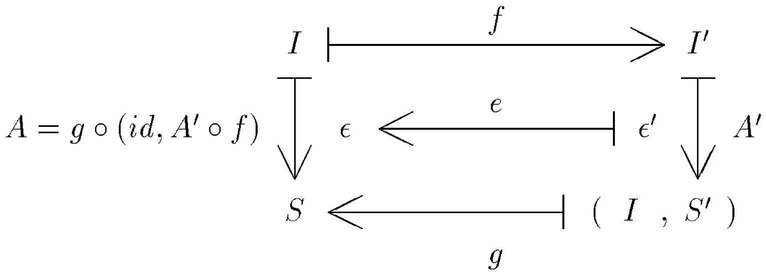
For the TM model the existence of such an i_I for each $I \in \mathcal{I}$ means that A can only read a prefix of n_{i_I}. But A does not know the magnitude of ϵ_{m_I,i_I} which can be arbitrary close to 0. So in this case A has to compute the optimal solution. The desired modification of A is as follows: Instead of reading the input bits of n for some $\epsilon = \frac{n}{d}$, the modified A reads 1's until it terminates. The termination is ensured, because A terminates for n_{i_I}, too. Thus A acts as if there were an ϵ_{m_I,i_I} in the input and computes an optimal solution. $\square$

The solution for this problem is an error bound range like $\mathbb{E} = \mathbb{N} \cup 1/\mathbb{N}$ where $1/\mathbb{N} = \{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots\}$. The simple restriction to rationals $\epsilon < 1$ (e.g. in [2], [4], [7], [11]) does not prevent this degeneration of the fptas notion.

4 Approximation preserving reductions

The notion of A-reduction and P-reduction is common in the field of approximation complexity of combinatorial optimization problems (see e.g. [4], [7], [10], [11]). An A-reduction is able to transform an apx, a P-reduction is able to transform a ptas from one optimization problem to another. By this means the approximation complexity can be compared. It is an obvious consequence of the definition of apx and ptas that each ptas for an optimization problem Π immediately implies an apx for Π: Extend each input I by some arbitrary but constant $\epsilon \in \mathbb{E}$ and use the ptas. The resulting apx complies with the error bound ϵ. It might be suggestive to think that a similar result is true for A- and P-reduction ([7], Proposition 3.5). However, it is not true that each P-reduction is also an A-reduction. We prove this negative result by a counterexample but we also show which additional conditions are sufficient for a positive answer. First we define both A- and P-reduction in their standard way (see Figure 1 and Figure 2).

Figure 1: An A-reduction (f, g) with error bound transformation e.

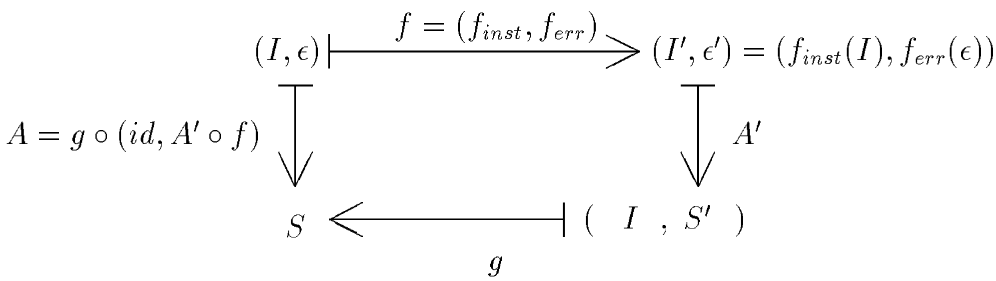


Definition 2 : An A-reduction from an optimization problem $\Pi = (\mathcal{I}, \mathcal{C}, cost, \underline{\diamond})$ to an optimization problem $\Pi' = (\mathcal{I}', \mathcal{C}', cost', \underline{\diamond}')$ with error function $\mathcal{E}$ and error bound range $\mathbb{E}$ is a pair of transformations (f, g) and an *error bound transformation* e, where

1. $f : \mathcal{I} \to \mathcal{I}'$
2. for all $\epsilon' \in \mathbb{E}$, for all $I \in \mathcal{I}$ and all $S' \in \mathcal{S}_{\Pi'}(f(I))$:
   (a) $g(I, S') \in \mathcal{S}_{\Pi}(I)$
   (b) $\mathcal{E}_{\Pi'}(f(I), S') \leq \epsilon' \Rightarrow \mathcal{E}_{\Pi}(I, g(I, S')) \leq e(\epsilon')$
3. the runtimes of both f and g are polynomially bounded in their input lengths.

For a P-reduction the situation is a little bit more complicated. Transformation f has to transform each extended input (I, ϵ). There are additional restrictions on f and on the time complexity of f and g (see Figure 2).

Figure 2: A P-reduction (f, g).



Definition 3 : A P-reduction from an optimization problem $\Pi = (\mathcal{I}, \mathcal{C}, cost, \underline{\diamond})$ to an optimization problem $\Pi' = (\mathcal{I}', \mathcal{C}', cost', \underline{\diamond}')$ with error function $\mathcal{E}$ and error bound range $\mathbb{E}$ is a pair of transformations (f, g), where

1. $f = (f_{inst}, f_{err})$ with $f_{inst} : \mathcal{I} \to \mathcal{I}'$, $f_{err} : \mathbb{E} \to \mathbb{E}$
2. for all $(I, \epsilon) \in \mathcal{I} \times \mathbb{E}$ and all $S' \in \mathcal{S}_{\Pi'}(f_{inst}(I))$:
   (a) $g(I, S') \in \mathcal{S}_{\Pi}(I)$
   (b) $\mathcal{E}_{\Pi'}(f_{inst}(I), S') \leq f_{err}(\epsilon) \Rightarrow \mathcal{E}_{\Pi}(I, g(I, S')) \leq \epsilon$

3. for each fixed $\epsilon \in \mathbb{E}$ the runtimes of both f and g are polynomially bounded in their input lengths.

It is easy to prove that A- and P-reduction compose and that they really transform a whole apx resp. ptas (see e.g. [6], [10]). Note that our following results also hold for an additional dependency of g on ϵ as introduced in [2] and [6]. But this is not essential for our results.

For the proof of Theorem 2, we use the following two minimization problems.

Definition 4 Let MinN be the optimization problem $(\mathbb{N}, \mathcal{C}, cost, \leq)$, where $\mathcal{C} := \{(n,i) \mid n,i \in \mathbb{N} \wedge i \leq n\}$ and $cost(n,i) = i$. Let MinN′ be the version where $cost(n,i) = \begin{cases} 2, & \text{if } i = 1 \\ 3, & \text{else} \end{cases}$.

These problems are trivial, clearly NP optimization problems, and not NP-hard. The optimal solution is always the natural number 1. Theorem 2 shows the core of our result. A corollary for two NP-hard NP minimization problems follows. Note that for MinN the relative error for a non-optimal solution i, $1 < i \leq n$, of an input $n \in \mathbb{N}$ is $\frac{i}{1} - 1 = i - 1$. But for MinN′ it is $\frac{3}{2} - 1 = \frac{1}{2}$. Now we are able to state the announced theorem.

Theorem 2 *There exist NP-optimization problems* $\Pi = (\mathcal{I}, \mathcal{C}, cost, \lesseqgtr)$ *and* $\Pi' = (\mathcal{I}', \mathcal{C}', cost', \lesseqgtr')$ *and a P-reduction* $((f_{inst}, f_{err}), g)$ *from* $(\mathcal{I}, \mathcal{C}, cost, \lesseqgtr)$ *to* $(\mathcal{I}', \mathcal{C}', cost', \lesseqgtr')$ *for the relative error* $\mathcal{E}^{rel}$ *and an error bound range* $\mathbb{E}$*, such that there exists* **no** *function* e *for which* (f_{inst}, g) *is an A-reduction with error bound transformation* e.

Proof: The idea of the proof is as follows: Suppose an apx A' for Π' computes for each $f_{inst}(I)$ a solution S' where the error bound ϵ' holds. If there exists an ϵ with $f_{err}(\epsilon) = \epsilon'$ then (f_{inst}, g) yields an apx A for Π which complies with the error bound ϵ. If there is no such ϵ for ϵ', the original $\bar{\epsilon}$ of any $\bar{\epsilon'} = f_{err}(\bar{\epsilon}) \geq \epsilon'$ serves the same purpose. This means we have to construct a P-reduction (f, g) where both are not possible.

Let Π =MinN and Π' =MinN′. Consider the reduction $((id, f_{err}), g)$ where id is the identity and

$$f_{err} : \epsilon \mapsto \frac{1}{4} \text{ and } g : (n, i') \mapsto i'.$$

This reduction is really a P-reduction because $\forall \epsilon \in \mathbb{E}\ \forall n, i' \in \mathbb{N}$ with $i' \leq id(n)$:

$$\mathcal{E}^{rel}_{MinN'}(id(n), i') \leq f_{err}(\epsilon) \leq \frac{1}{4} \Rightarrow \mathcal{E}^{rel}_{MinN}(n, g(n, i')) = \mathcal{E}^{rel}_{MinN}(n, i') = 0 \leq \epsilon.$$

This holds because for each input $id(n)$ of MinN′ only the optimal solution $i' = 1$ is able to comply with the error bound $f_{err}(\epsilon) \leq \frac{1}{4}$.

Now we want to construct a contradiction: Assume (id, g) is an A-reduction with some error bound transformation e. Then this guarantees $\forall \epsilon' \in \mathbb{E}\ \forall n, i' \in \mathbb{N}$ with $i' \leq id(n)$:

$$\mathcal{E}^{rel}_{MinN'}(id(n), i') \leq \epsilon' \Rightarrow \mathcal{E}^{rel}_{MinN}(n, g(n, i')) \leq e(\epsilon') \qquad (1)$$

We show that especially for the error bound $\epsilon' = \frac{1}{2}$ the existence of an error bound transformation e for (id, g) leads to a contradiction. We define $e_{1/2} = \lceil e(\frac{1}{2}) \rceil + 2$.

This means $e_{1/2}$ is a natural number. Therefore $e_{1/2}$ is an input for MinN and a non-minimal solution of $id(e_{1/2})$ for MinN$'$ and leads to a relative error $\mathcal{E}^{rel}_{MinN'}(id(e_{1/2}), e_{1/2}) = \frac{1}{2}$. We can apply Implication 1 for $n = e_{1/2}$ and $\epsilon' = \frac{1}{2}$, and have

$$\mathcal{E}^{rel}_{MinN}(e_{1/2}, g(e_{1/2}, e_{1/2})) = \mathcal{E}^{rel}_{MinN}(e_{1/2}, e_{1/2}) = e_{1/2} - 1 \leq e(\frac{1}{2}) \tag{2}$$

But the definition for $e_{1/2}$ yields

$$e_{1/2} - 1 = \lceil e(\frac{1}{2}) \rceil + 1 > e(\frac{1}{2}) \tag{3}$$

Inequality 3 is a contradiction to Inequality 2. □

Note that our proof only requires that the values $\frac{1}{4}$ and $\frac{1}{2}$ are in $\mathbb{E}$. Moreover, it can be modified to use other values. One might criticize that the involved minimization problems and the given ptas are trivial and that it is no problem to exclude these degenerative cases by slightly modified definitions for P-reduction and ptas. But the following corollary will show that we can give NP-hard minimization problems and non trivial P-reductions, too.

To do this, we need the notion of *product* for minimization problems. Let λ be a new symbol with respect to the encoding alphabet Σ.

Definition 5 Let both $(\mathcal{I}, \mathcal{C}, cost, \leq)$ and $(\mathcal{I}', \mathcal{C}', cost', \leq)$ be minimization problems. We define the product $(\mathcal{I}, \mathcal{C}, cost, \leq) \times (\mathcal{I}', \mathcal{C}', cost', \leq) = (\mathcal{I}^\times, \mathcal{C}^\times, cost^\times, \leq)$ by

$$\begin{aligned}
\mathcal{I}^\times &= \{(I, \lambda) \mid I \in \mathcal{I}\} \cup \{(\lambda, I') \mid I' \in \mathcal{I}'\}, \\
\mathcal{C}^\times &= \{((I, \lambda), (S, \lambda)) \mid (I, S) \in \mathcal{C}\} \cup \{((\lambda, I'), (\lambda, S')) \mid (I', S') \in \mathcal{C}'\}, \\
cost^\times &: ((I, I'), (S, S')) \mapsto \begin{cases} cost(I, S), & \text{for } (I, S) \in \mathcal{C} \\ cost'(I', S'), & \text{for } (I', S') \in \mathcal{C}' \end{cases}
\end{aligned}$$

Corollary 1 *Let both $\Pi = (\mathcal{I}, \mathcal{C}, cost, \leq)$ and $\Pi' = (\mathcal{I}', \mathcal{C}', cost', \leq)$ be NP-hard NP-minimization problems, and let $((f_{inst}, f_{err}), g)$ be a P-reduction from Π to Π' for the relative error $\mathcal{E}^{rel}$ and an error bound range $\mathbb{E}$. Then there exists a P-reduction $((f^\times_{inst}, f^\times_{err}), g^\times)$ from $\Pi \times$MinN to $\Pi' \times$MinN$'$ such that there exists* **no** *function e for which $(f^\times_{inst}, g^\times)$ is an A-reduction with error bound transformation e.*

Proof: First we construct the desired P-reduction $((f^\times_{inst}, f^\times_{err}), g^\times)$:

$$\begin{aligned}
f^\times_{inst} &: (I, \lambda) \mapsto f_{inst}(I),\ (\lambda, n) \mapsto n \\
f^\times_{err} &: \epsilon \mapsto \min(f_{err}(\epsilon), \frac{1}{4}) \\
g^\times &: ((I, \lambda), (S', \lambda)) \mapsto g(I, S'),\ ((\lambda, n), (\lambda, i')) \mapsto i'.
\end{aligned}$$

To show that $((f^\times_{inst}, f^\times_{err}), g^\times)$ is really a P-reduction we can consider both components of the product problems independently. For the second component only the minimal solution 1 of MinN$'$ complies with an error bound $\leq \frac{1}{4}$. For the first component an error bound ϵ is mapped to $\min(f_{err}(\epsilon), \frac{1}{4})$. Reduction $((f^\times_{inst}, f^\times_{err}), g^\times)$

differs from $((f_{inst}, f_{err}), g)$ only for extended inputs (I, ϵ) with $f_{err}(\epsilon) > \frac{1}{4}$. But a related solution S' which complies with the error bound $\frac{1}{4}$ also complies with the error bound $f_{err}(\epsilon) > \frac{1}{4}$. Thus $((f_{inst}^{\times}, f_{err}^{\times}), g^{\times})$ is a P-reduction.

Completely analogous to the proof of Theorem 2, we assume an error bound transformation e and we define $e_{1/2} = \lceil e(\frac{1}{2}) \rceil + 2$ which leads again to the contradiction:

$$\mathcal{E}^{rel}_{\Pi \times MinN}((\lambda, e_{1/2}), g^{\times}((\lambda, e_{1/2}), (\lambda, e_{1/2})) = e_{1/2} - 1 \leq e(\frac{1}{2}). \tag{4}$$

□

Note that both $\Pi \times$MinN and $\Pi' \times$MinN$'$ of the previous proof are NP-hard NP-minimization problems. For inputs (λ, n) the optimal solution is always 1, but the inputs (I, λ) make the problem NP-hard. The contradiction results only from the second component of the product problem. The P-reduction $((f_{inst}^{\times}, f_{err}^{\times}), g^{\times})$ is non trivial for non trival P-reductions $((f_{inst}, f_{err}), g)$.

Nevertheless, our idea for the proof of Theorem 2 leads to a positive version of Theorem 2.

Corollary 2 *Given two NP-optimization problems $\Pi = (\mathcal{I}, \mathcal{C}, cost, \lesseqgtr)$ and $\Pi' = (\mathcal{I}', \mathcal{C}', cost', \lesseqgtr')$ and a P-reduction $((f_{inst}, f_{err}), g)$ from Π to Π' for the relative error $\mathcal{E}^{rel}$ and an error bound range $\mathbb{E}$. If there exists a function $e : \mathbb{E} \to \mathbb{E}$ such that for each $\epsilon' \in \mathbb{E}$ we have $\epsilon' \leq f_{err}(e(\epsilon'))$, then (f_{inst}, g) is an A-reduction with error bound transformation e.*

Proof: All we have to show is that Inequality 2b of Definition 2 holds. For each $\epsilon' \in \mathbb{E}$ with $\mathcal{E}^{rel}_{\Pi'}(f_{inst}(I), S') \leq \epsilon'$ we have also $\epsilon' \leq f_{err}(e(\epsilon))$. Together with Inequality 2b of Definition 3 this completes the proof. □

Clearly, a P-reduction with invertible transformation f_{err} satisfies the conditions of Corollary 2 for $e = f_{err}^{-1}$. Also a surjective f_{err}, as required for the PTAS-reduction in [2], suffices because for each $\epsilon' \in \mathbb{E}$ there exists an ϵ where $f_{err}(\epsilon) = \epsilon'$, thus defining the desired function e. But the criteria of Corollary 2 is even more general. The L-reduction defined in [12] with $f_{err} : \epsilon \mapsto \frac{\epsilon}{\alpha\beta}$, $\alpha, \beta > 0$, is not surjective for an error bound range $\mathbb{E} = (0, 1)$. But an error bound transformation $e : \epsilon' \mapsto \min(\alpha\beta\epsilon', \alpha\beta)$ satisfies our criteria.

References

[1] S. Arora. *Probabilistic Checking of Proofs and Hardness of Approximation Problems.* PhD thesis, Departement of Computer Science, Princeton University, Princeton, NJ, USA, 1994. Report CS-TR-476-94.

[2] G. Ausiello, P. Crescenzi, and M. Protasi. Approximate solutions of NP optimization problems. *Theoretical Comput. Sci.*, 150:1–55, 1995.

[3] D. Brushi, D. Joseph, and P. Young. A structural overview of NP optimization problems. *Algorithms Review*, 2(1):1–26, 1991.

[4] P. Crescenzi and A. Panconesi. Completeness and approximation classes. *Information and Computation*, 93:241–262, 1991.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.

[6] E. Ihler. *Zur Übertragbarkeit von Approximationsalgorithmen für Optimierungsprobleme, insbesondere für das Problem Class Tree.* PhD thesis, Institut für Informatik, Universität Freiburg, Deutschland, 1993.

[7] V. Kann. *On the Approximability of NP-complete Optimization Problems.* PhD thesis, Departement of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden, 1992.

[8] P. G. Kolaitis and M. N. Thakur. Approximation properties of NP minimization classes. *Proc. 6th Annual Conf. on Structures in Computer Science*, pages 353–366, 1991.

[9] P. G. Kolaitis and M. N. Thakur. Logical definability of NP optimization problems. *Information and Computation*, 115:321–353, 1994.

[10] P. Orponen and H. Mannila. On approximation preserving reductions: Complete problems and robust measures. Technical Report C–1987–28, Department of Computer Science, University of Helsinki, 1987.

[11] A. Panconesi and D. Ranjan. Quantifiers and approximation. In *Proc. 22th Annual ACM Symp. on Theory of Computing*, pages 446–456, 1990.

[12] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.

[13] C. H. Papadimitrou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, 1982.

[14] A. Paz and S. Moran. Non deterministic polynomial optimization problems and their approximations. *Theoretical Comput. Sci.*, 15:251–277, 1981.