Optimization Problems and Approximation

We are unable to solve NP-complete problems efficiently, i.e., there is no known way to solve them in polynomial time.

Most of them are decision versions of optimization problems…

- with a set of feasible solutions for each instance
- with an associated quality measure

Why not looking for an approximate solution?

Is there a difference in complexity?

Optimization Problems and Approximation
Example Knapsack revisited

$KNAPSACK = < I, f >$

$I = \{ < S, w, W, v > | \; S = \{1,\ldots,n\}, \; w, v : S \to N, \;\; W \in N \}$

$f(i) = \left\{ T \subseteq S \mid \sum_{i \in T} w(i) \le W \; , \;\; \sum_{i \in T} v(i) \to \max \right\}$

All set $T \subseteq S : \sum_{i \in T} w(i) \le W$ are feasible solutions.

$\sum_{i \in T} v(i)$ is the quality of the solution T wrt. to the instance i.

Optimization Problems and Approximation
Example Knapsack revisited

All set $T \subseteq S : \sum_{i \in T} w(i) \le W$ are feasible solutions.

$\sum_{i \in T} v(i)$ is the quality of the solution T wrt. to the instance i.

$KNAPSACK = < I, \mathrm{sol}, \mathrm{m}, \max >$

$I = \{ < S, w, W, v > | \; S = \{1,\ldots,n\}, \; w, v : S \to N, \; W \in N, \; V \in N \}$

$\mathrm{sol}(i) = \left\{ T \subseteq S : \sum_{i \in T} w(i) \le W \right\}$

$\mathrm{m}(i, s) = \sum_{i \in T} v(i)$

Optimization Problems and Approximation
Definition of Optimization Problems

$OPTPROB = < I, \mathrm{sol}, \mathrm{m}, \mathrm{type} >$

- I the instance set
- $\mathrm{sol}(i)$ the set of feasible solutions for instance i ($\mathrm{sol}(i)$ nonempty for $i \in I$)
- $\mathrm{m}(i, s)$ the measure of solution s wrt. instance i (positive integer for $i \in I$ and $s \in \mathrm{sol}(i)$)

$$\mathrm{opt}(i) = \underset{s \in \mathrm{sol(i)}}{\mathrm{type}} \; \mathrm{m}(i, s)$$

Optimization Problems and Approximation
Example Problem: MaxkSat

$MaxkSat = < I, \mathrm{sol}, \mathrm{m}, \max >$

- $I = CNF-Formulas$ with at most k literals per clause
- $\mathrm{sol}(\varphi) =$ set of assignments to the vars. of φ
- $\mathrm{m}(\varphi, A) =$ the number of clauses which are satisfied by A

$MaxSat$ has all $CNF-Expressions$ as instances.

There is also a weighted version : Each clause has a weight - - the measure is the sum of the weights of the satisfied clauses.

Example Problem: MaxkSat
NP-hardness

$MaxkSat = < I, \mathrm{sol}, \mathrm{m}, \max >$

- $I = CNF-Formulas$ with at most k literals per clause
- $\mathrm{sol}(\varphi) =$ set of assignments to the vars. of φ
- $\mathrm{m}(\varphi, A) =$ the number of clauses which are satisfied by A

$Max3Sat(D)$ is certainly $NP-complete$
(thus $Max3Sat$ is $NP-hard$) :
$3SAT$ is a special case

But also $Max2Sat(D)$ is $NP-complete$....

Optimization Problems and Approximation
Performance Ratio

Approximation algorithms deliver solutions of guaranteed quality – they are not heuristics.

But how to measure the quality of a solution?

Let $O = < I, \mathrm{sol}, \mathrm{m}, \mathrm{type} >$ be an optimization problem.
given $i \in I$ and a $s \in \mathrm{sol}(i)$ we define

$$R(i,s) = \max\left\{\frac{\mathrm{opt(i)}}{\mathrm{m(i,s)}}, \frac{\mathrm{m(i,s)}}{\mathrm{opt(i)}}\right\}$$

as the performance ratio.
$s \in sol(i)$ is a an $r-approximate$ solution if $R(i,s) \le r$.

Optimization Problems and Approximation
Performance Ratio

Let $O = < I, \mathrm{sol}, \mathrm{m}, \mathrm{type} >$ be an optimization problem.
given $i \in I$ and a $s \in \mathrm{sol}(i)$ we define

$$R(i,s) = \max\left\{\frac{\mathrm{opt(i)}}{\mathrm{m(i,s)}}, \frac{\mathrm{m(i,s)}}{\mathrm{opt(i)}}\right\}$$

as the performance ratio.
$s \in sol(i)$ is a an $r-approximate$ solution if $R(i,s) \le r$.

$R(i,s) = 1$ implies that s is optimal.

$R(i,s) \ge 1$ in general, the closer to 1, the better.

Example Problem MaxkSat
Performance Ratio

$MaxkSat = < I, \mathrm{sol}, \mathrm{m}, \max >$
$I = CNF-Formulas$ with at most k literals per clause
$\mathrm{sol}(\varphi) =$ set of assignments to the vars. of φ
$\mathrm{m}(\varphi, A) =$ the number of clauses which are satisfied by A

$R(\varphi, A) = \frac{\mathrm{opt}(\varphi)}{\mathrm{m}(\varphi, A)}$ If we have an A with $R(\varphi, A) \le \frac{3}{2}$ then

no A' can satisfy more than $\frac{3}{2}\mathrm{m}(\varphi, A)$ clauses.

Optimization Problems and Approximation
The Class NPO

NPO is the class of optimization problems whose decision versions are in NP.

$OPTPROB = < I, \mathrm{sol}, \mathrm{m}, \mathrm{type} > \in NPO$ iff

$\exists$polynomial $p : \forall i \in I, s \in \mathrm{sol(i)} : |s| \le p(|i|)$

deciding $s \in \mathrm{sol(i)}$ is in P

computing $\mathrm{m}(s,i)$ is in FP

Optimization Problems and Approximation
Approximation Problem

Let $O = < I, \mathrm{sol}, \mathrm{m}, \mathrm{type} >$ be an optimization problem and r a function $N \to [1, \infty)$.

Then the approximation problem $< O, r >$ is to find for all instances $i \in I$ an $r(|i|)$-$approximate$ solution $s \in \mathrm{sol}(i)$.

The question is which approximation problems $< O, r >$ are located in FP.

And *how to prove* that they are not (under some assumption such as $P \ne NP$)..

Approximation Algorithm
Example Problem: MaxSat

approxMaxSat(φ)
1. for $i = 1$ to n
2. val := E($\mathrm{m}(\varphi, A \cup \{x_i = \mathrm{true}\})) >$ E($\mathrm{m}(\varphi, A \cup \{x_i = \mathrm{false}\})$);
3. A := A $\cup \{x_i = \mathrm{val}\}$; $\varphi := \varphi[x_i = \mathrm{val}]$;
4. return A;

$$\mathrm{E}(\varphi, \{\}) = \sum_{C \in \varphi} 1 - 2^{-|C|} \ge \sum_{C \in \varphi} 1 - 2^{-1} = \frac{1}{2}|\varphi|$$

Thus, this algorithm is a 2-approximate algortithm or better.

Approximation Algorithm
Example Problem: VertexCover

approxVertexCover(V, E)
1. C := ∅;
2. while $E \neq \emptyset$ do
3. pick a $<u,v> \in E$
4. $C := C \cup \{u,v\}$;
5. remove $\{u,v\}$ from V, E;
6. return C;

C is indeed a valid cover.

Every cover must cover all the edges picked in line 3.

Thus every cover must contain at least | C | /2 vertexes.

$$R(G,C) = \frac{\mathrm{m}(G,C)}{\mathrm{opt}(G)} \leq 2$$

Approximation Classes
APX

We have two approximation problems, which can be solved
within a constant performance ratio
within polynomial time.

So it's time to define a corresponding class: APX.

Let O be an NPO problem.
$O \in APX$ iff there exists an $r-approximation$ algorithm for O which run in polynomial time for some constant $r \geq 1$.

Approximation Classes
Example Problem: TSP (I)

We will show that $TSP \in APX \Leftrightarrow P = NP$.

We use another $NP-complete$ problem to reduce from: $HAMILTONIANCYCLE$

$HAMILTONIANCYCLE$: Given a graph $G = <V, E>$, is there a cycle, which visits any node exactly once?

We construct a distance matrix M as follows (for $r \geq 1$):

$$M(u,v) = \begin{cases} 1: <u,v> \in E \\ \lceil r\,|V|\,\rceil: \text{otherwise} \end{cases}$$

Approximation Classes
Example Problem: TSP (II)

We construct a distance matrix M as follows ($r \geq 1$):

$$M(u,v) = \begin{cases} 1: <u,v> \in E \\ \lceil r\,|V|\,\rceil: \text{otherwise} \end{cases}$$

If G is a positive instance, then $\mathrm{opt}(M) = |V|$.
Otherwise $\mathrm{opt}(M) \geq \lceil r|V| \rceil + |V| - 1$.

Now assume that there is an $r-approximate$ algorithm for TSP.

Approximation Classes
Example Problem: TSP (III)

If G is a positive instance, then $\mathrm{opt}(M) = |V|$.
Otherwise $\mathrm{opt}(M) \geq \lceil r|V| \rceil + |V| - 1$.

Now assume that there is an $r-approximate$ algorithm apporx for TSP and let $s = \text{approx}(M)$.

If $G \in HAMILTONIANCYCLE$, we find

$r \geq R(M,s) = \frac{\mathrm{m}(M,s)}{\mathrm{opt(M)}} = \frac{\mathrm{m}(M,s)}{|V|}$ and so $|V|\, r \geq \mathrm{m}(M,s)$.

But otherwise we have

$$\mathrm{m}(M,s) \geq \mathrm{opt(M)} \geq \lceil r\,|V|\,\rceil + |V| - 1 > \lceil r\,|V|\,\rceil$$

Approximation Classes
Example Problem: TSP (IV)

So we could prove that $TSP \notin APX$ (assuming $P \neq NP$) by giving a reduction from an $NP-hard$ problem, which established a gap between positive and negative instances.

The gap was large enough to distinguish whether we reduced from a positive or a negative instance.

Wanted: A generic reduction from $NP-hard$ problems, to approximation problems which produces gaps.

Approximation Classes
Relationships

$APX \subseteq NPO$

$TSP \in APX \Leftrightarrow P = NP$

$APX \subset NPO \Leftrightarrow P \neq NP$

Max3Sat and *VertexCover* are in *APX*.

Approximation Classes
Approximation Schemes

An algorithm which can be parametrized with the performance ration to achieve is called *approximation – scheme.*

Let $O = \langle I, \mathrm{sol}, \mathrm{m}, \mathrm{type} \rangle$ be an optimization problem.
Then an algorithm A is an approximation scheme for O iff
for all $i \in I,\ r > 1$ and $s = A(i, r)$
$s \in \mathrm{sol}(I)$ and $R(i, s) \leq r$.

Approximation Schemes
The classes PTAS and FPTAS

$O \in FPTAS$ if there is an approximation scheme A such that $A(i, r)$ runs in $DTIME(\mathrm{poly}(|i|, 1/(r-1)))$ for all $i \in I$ and $r > 1$.

$O \in PTAS$ if there is an approximation scheme A such that $A(i, r)$ runs in $DTIME(\mathrm{poly}(|i|))$ for all $i \in I$ and any fixed $r > 1$.

Approximation Schemes
Example Problem: KNAPSACK

$KNAPSACK = \langle I, \mathrm{sol}, \mathrm{m}, \max \rangle$

$I = \{\langle S, w, W, v \rangle \mid S = \{1, \ldots, n\},\ w, v : S \to N,\ W \in N\}$

$\mathrm{sol}(i) = \left\{ T \subseteq S : \sum_{x \in T} w(x) \leq W \right\}$

$\mathrm{m}(i, s) = \sum_{x \in T} v(x)$

Let $W(x, v)$ be the minimum weight attainable by selecting among the *first x items* such that that their total value is *exactly v.*

Example Problem: KNAPSACK
A Pseudo-Polynomial Algorithm

Let $W(x, v)$ be the minimum weight attainable by selecting among the *first x items* such that their total value is *exactly v.*

$W(0,0) = 0$

$W(0, v) = \infty \quad v > 0$

$W(x+1, v) = \min\{W(x, v), [W(x, v - v(x+1)) + w(x+1)]\}$

By building the table of the $W(x, v)$ for $0 \leq x \leq n$ and $0 \leq v \leq V = \sum_{x \in S} v(x)$ we can solve *KNAPSACK*.

This algorithm runs in $DTIME(\mathrm{poly}(n, V))$ (pseudo-poly.)

Example Problem: KNAPSACK
An FPTAS (I)

This algorithm runs in $DTIME(\mathrm{poly}(n, V))$ (pseudo-poly.)

Assume $\varepsilon > 0$ fixed.

Let $l = \lfloor \log \max_{x \in S} v(x) \rfloor$.

Choose k with $\frac{n}{n^k} < \varepsilon$.

Set $L = l - k \log n$.

Define i' with $v'(x) = \lfloor v(x) / 2^L \rfloor 2^L$

We keep the most significant $k \log n$ bits.

The rest, i.e., $L = l\text{-}k \log n$, gets zeroized.

Example Problem: KNAPSACK
An FPTAS (II)

This algorithm runs in $DTIME(\mathrm{poly}(n,V))$ (pseudo-poly.)

Assume $\varepsilon > 0$ fixed.

Let $l = \lfloor \log \max_{x\in S} v(x) \rfloor$.

Choose k with $\frac{n}{n^k} < \varepsilon$.

Set $L = l - k \log n$.

Define i' with

$v'(x) = \lfloor v(x)/2^L \rfloor 2^L$

$$\sum_{x\in T} v(x) \le \sum_{x\in T} v'(x) + |T|\, 2^L$$

$$\mathrm{opt}(i) \le \mathrm{opt}(i') + n2^L$$

$$\frac{\mathrm{opt}(i)}{\mathrm{opt}(i')} \le 1 + \frac{n2^L}{\mathrm{opt}(i')} \le 1 + \frac{n2^L}{\max_{x\in S} v'(x)} \le 1 + \frac{n2^L}{2^l} \le 1 + \varepsilon$$

Example Problem: KNAPSACK
An FPTAS (III)

This algorithm runs in $DTIME(\mathrm{poly}(n,V))$ (pseudo-poly.)

Assume $\varepsilon > 0$ fixed.

Let $l = \lfloor \log \max_{x\in S} v(x) \rfloor$.

Choose k with $\frac{n}{n^k} < \varepsilon$.

Set $L = l - k \log n$.

Define I' with

$v'(i) = \lfloor v(i)/2^L \rfloor 2^L$

$$\frac{\mathrm{opt}(i)}{\mathrm{opt}(i')} \le 1 + \varepsilon$$

$$\sum_{x\in T} v'(x) \le \sum_{x\in T} v(x)$$

$$\frac{\mathrm{opt}(i)}{\mathrm{m}(i, \mathrm{optsol}(i'))} \le \frac{\mathrm{opt}(i)}{\mathrm{opt}(i')} \le 1 + \varepsilon$$

Solving I' optimally yields an $1+\varepsilon$ approximate solution for I

Example Problem: KNAPSACK
An FPTAS (IV)

This algorithm runs in $DTIME(\mathrm{poly}(n,V))$ (pseudo-poly.)

Assume $\varepsilon > 0$ fixed.

Let $l = \lfloor \log \max_{x\in S} v(x) \rfloor$.

Choose k with $\frac{n}{n^k} < \varepsilon$.

Set $L = l - k \log n$.

Define I' with

$v'(x) = \lfloor v(x)/2^L \rfloor 2^L$

We can solve I' in

$$DTIME(\mathrm{poly}(n, V'/2^L)) = DTIME(\mathrm{poly}(n, n2^{k \log n})) = DTIME(\mathrm{poly}(n, 1/\varepsilon))$$

Solving I' optimally yields an $1+\varepsilon$ approximate solution for I within $DTIME(\mathrm{poly}(|I|, 1/\varepsilon))$. $KNAPSACK \in FPTAS$.

Approximation Schemes
Polynomially Bound Problems

Let $O = < I, \mathrm{sol}, \mathrm{m}, \mathrm{type} >$ be a problem in NPO.

If there is polynomial p such that $\forall i \in I, s \in \mathrm{sol}(i): \ \mathrm{m}(i,s) \le p(|i|)$ then O is polynomially bound, i.e., $O \in NPO - PB$.

If there is an $NP-hard$ problem in $NPO-PB$ which admits an $FPTAS$, then $P = NP$.

Polynomially Bound Problems
Permit no FPTAS (I)

If there is an $NP-hard$ problem in $NPO-PB$ which admits an $FPTAS$, then $P = NP$.

Let O be a maximation problem in $NPO-PB$.

Set $r(i) = 1 + \frac{1}{p(|i|)}$, where p is the poly.-bound.

An $r(i)$-approximate solution s for i is optimal since,

$\frac{p(|i|)+1}{p(|i|)} = r(i) \ge \frac{\mathrm{opt}(i)}{\mathrm{m}(i,s)}$ gives

$$\mathrm{m}(i,s) \ge \mathrm{opt}(i)\frac{p(|i|)}{p(|i|)+1} = \mathrm{opt(i)} - \frac{\mathrm{opt(i)}}{p(|i|)+1} > \mathrm{opt}(i) - 1$$

Polynomially Bound Problems
Permit no FPTAS (II)

Set $r(i) = 1 + \frac{1}{p(|i|)}$, where p is the poly.-bound.

An $r(i)$-approximate solution s for i is optimal since,

$\frac{p(|i|)+1}{p(|i|)} = r(i) \ge \frac{\mathrm{opt}(i)}{\mathrm{m}(i,s)}$ gives

$$\mathrm{m}(i,s) \ge \mathrm{opt}(i)\frac{p(|i|)}{p(|i|)+1} = \mathrm{opt(i)} - \frac{\mathrm{opt(i)}}{p(|i|)+1} > \mathrm{opt}(i) - 1$$

If O would be in $FPTAS$ then we can solve O optimally in $DTIME(\mathrm{poly}(|i|, 1/(r(|i|)-1)) = DTIME(\mathrm{poly}(|i|)$.

Approximation Classes
Relationships

$FPTAS \subseteq PTAS \subseteq APX \subseteq NPO$

$\subsetneq$

$TSP \in APX \Leftrightarrow P = NP$

$Max3Sat \in FPTAS \Leftrightarrow P = NP$

Two questions : Are there problems in $PTAS\text{-}FPTAS$?
Are there problems in $APX - PTAS$?
(as usual, based on $P \neq NP$)

Approximation Classes
Problems in PTAS-FPTAS

$PLANAR\ INDEPENDENTSET$ is in $NPO - PB$ and is $NP - hard$.

$PLANAR\ INDEPENDENTSET \in FPTAS \Rightarrow P = NP$.

Unproven : $PLANAR\ INDEPENDENTSET \in PTAS$.

Approximation Classes
Relationships

$FPTAS \subseteq PTAS \subseteq APX \subseteq NPO$

$\subsetneq$

$PLANAR\ INDEPSET \in FPTAS$
$\Leftrightarrow P = NP$

$TSP \in APX \Leftrightarrow P = NP$

$Max3Sat \in FPTAS \Leftrightarrow P = NP$

One question : Are there problems in $APX - PTAS$?
(as usual, based on $P \neq NP$)

Hardness in Approximation

Wanted : A generic reduction from $NP - hard$ problems, to approximation problems which produces gaps.

Remember the reduction to TSP...

Relies on the so-called PCP-Theorem – an alternative formulation of NP.

It allows to reduce $NP - complete$ languages to approximation problems.

Hardness in Approximation
PCP-Verification

1. reads the input x

4. reads $O(q(|x|))$ proof bits

Input x

Proof Π

$PCP(r(n), q(n))$
Verifier

3. computes proof positions to read

5. decides

Random-Source

2. reads $O(r(|x|))$ random bits

Hardness in Approximation
PCP-Theorem (I)

A language L is in $PCP(r(n), q(n))$
if there is a polynomial time $PCP(r(n), q(n))$ - $Verifier\ V$
such that

$$\forall x \in L \ \ \exists \Pi : \Pr_{\bar{R}}\left[V(x, \Pi, \bar{R}) = \text{accept}\right] = 1$$

$$\forall x \notin L \ \forall \Pi : \Pr_{\bar{R}}[V(x, \Pi, \bar{R}) = \text{aceppt}] \leq 1/2$$

with $|\bar{R}| = O(r(|x|))$, and V reading $O(q(n))$ bits non - adaptively from Π.

Easy : $NP \supseteq PCP(\log n, 1)$ Hard : $NP \subseteq PCP(\log n, 1)$

Hardness in Approximation
PCP-Theorem (I)

A language L is in $PCP(r(n), q(n))$ if there is a polynomial time $PCP(r(n), q(n))$-*Verifier V* such that

$\forall x \in L \;\; \exists \Pi : \Pr_{\bar{R}}\left[V(x, \Pi, \bar{R}) = \text{accept}\right] = 1$

$\forall x \notin L \; \forall \Pi : \Pr_{\bar{R}}[V(x, \Pi, \bar{R}) = \text{accept}] \leq 1/2$

with $|\bar{R}| = O(r(|x|))$, and V reading $O(q(n))$ bits non-adaptively from Π.

PCP-Theorem: $NP = PCP(\log n, 1)$

Hardness in Approximation
PCP-Theorem (II)

PCP-Theorem: $NP = PCP(\log n, 1)$

How to use?

Reduce the verification process to an approximation problem such that the gap of the PCP-Verifier translates into a gap in the measure of the optimal solution(s).

Hardness in Approximation
Example Problem: Max3Sat (I)

Observe that once the $O(q(n))$ bits have been read from the proof Π, the decision of V is only depending on them.

Thus we can define a set of Boolean Expressions $\varphi[x, \bar{R}](\bar{p})$ where

- x is the input,
- $\bar{R}$ is the random string of length $O(\log n)$,
- $\bar{p}$ are the bits read in Π,
- $\varphi[x, \bar{R}](\bar{p}) = 1 \Leftrightarrow V(x, \Pi, \bar{R}) = \text{accept}$.

Hardness in Approximation
Example Problem: Max3Sat (II)

Each $\varphi[x, \bar{R}](\bar{p})$ can be expressed by d clauses, where d is constant (since $|\bar{p}|$ is constant).

Let φ be the conjunction of the expressions $\varphi[x, \bar{R}](\bar{p})$ for all $\bar{R}$ ($|\bar{R}| = c \log n$).

$x \in L \Rightarrow \exists \Pi : \Pr_{\bar{R}}\left[V(x, \Pi, \bar{R}) = \text{accept}\right] = 1$

$\Rightarrow$ all $\varphi[x, \bar{R}]$ can be satisified satisfied simultansously

$\Rightarrow \varphi$ satisfiable.

Hardness in Approximation
Example Problem: Max3Sat (III)

Each $\varphi[x, \bar{R}](\bar{p})$ can be expressed by d clauses, where d is constant (since $|\bar{p}|$ is constant).

Let φ be the conjunction of the expressions $\varphi[x, \bar{R}](\bar{p})$ for all $\bar{R}$ ($|\bar{R}| = c \log n$).

$x \notin L \Rightarrow \forall \Pi : \Pr_{\bar{R}}\left[V(x, \Pi, \bar{R}) = \text{accept}\right] \leq 1/2$

$\Rightarrow$ each assignment must leave $1/2$ of the expressions $\varphi[x, \bar{R}]$ unsatisified.

$$\Rightarrow \frac{\text{opt}(\varphi)}{|\varphi|} \leq f = \frac{1}{2} + \frac{1}{2}\frac{d-1}{d} < 1$$

Hardness in Approximation
Example Problem: Max3Sat (IV)

$$x \in L \Rightarrow \frac{\text{opt}(\varphi)}{|\varphi|} = 1$$

$$x \notin L \Rightarrow \frac{\text{opt}(\varphi)}{|\varphi|} \leq f = \frac{1}{2} + \frac{1}{2}\frac{d-1}{d} < 1$$

Let A be an $1 < r < \frac{1}{f}$ approximate solution for φ.

$\frac{\text{m}(\varphi, A)}{\text{opt}(\varphi)} \geq \frac{1}{r} > f$

$x \in L \Rightarrow \text{m}(\varphi, A) > f \,\text{opt}(\varphi) = f\,|\varphi|$

$x \notin L \Rightarrow \text{m}(\varphi, A) \leq \text{opt}(\varphi) \leq f\,|\varphi|$ (for all A)

r-approximating $Max3Sat$ is $NP-hard$ (constant $r > 1$). ●

Hardness in Approximation

Remark: Decoding of PCP-Proofs

$$\forall x \in L \;\; \exists \Pi : \Pr_{\bar{R}}\left[V(x,\Pi,\bar{R}) = \text{accept}\right] = 1$$

$$\forall x \notin L \;\; \forall \Pi : \Pr_{\bar{R}}[V(x,\Pi,\bar{R}) = \text{acceppt}] \le 1/2$$

Given a proof Π with $\Pr_{\bar{R}}\left[V(x,\Pi,\bar{R}) = \text{accept}\right] > 1/2$ a proof Π' with $\Pr_{\bar{R}}\left[V(x,\Pi',\bar{R}) = \text{accept}\right] = 1$ can be reconstructed efficiently (in FP).

Π is basically encoded for error - correction - - thus it possible to find the corresponding "usually encoded" proof efficiently.

Approximation Classes

Relationships

$FPTAS \subseteq PTAS \subseteq APX \subseteq NPO$

$PLANAR\ INDEPSET \in FPTAS \Leftrightarrow P = NP$

$TSP \in APX \Leftrightarrow P = NP$

$Max3Sat \in PTAS \Leftrightarrow P = NP$

Approximation Classes

Relationships

$FPTAS \subseteq PTAS \subseteq APX \subseteq NPO$

$FPTAS \subset PTAS \subset APX \subset NPO \quad \Leftrightarrow \quad P \neq NP$

Approximation Classes

More Classes

Let O be an NPO problem.
$O \in F - APX$ iff there exists an $r - approximation$ algorithm for O which run in polynomial time for some $function\ f \in F$.

$FPTAS \subseteq PTAS \subseteq APX \subseteq \log - APX \subseteq \text{poly} - APX \subseteq \exp - APX \subseteq NPO$