# PCP and Hardness of Approximation

January 30, 2009

Our goal herein is to define and prove basic concepts regarding hardness of approximation. We will state —but obviously not prove— a PCP theorem as a starting point.

Let us first revisit our perspective on optimization problems.

## Optimization, Approximation and Gap-problems

A decision problem or a language $L$ is one where all strings $x$ are partitioned into those that are *good inputs* and the *bad* ones. In general, we are really interested in *optimization* problems, where for a problem $A$ we consider all potential *solutions* $w$, and every input $x$ determines a subset of all possible solutions that are good for $x$. Furthermore, every input $x$ imposes a ranking among all good solutions according to the *optimized parameter* $\rho_A(x, w)$; and our problem $A$ calls for the best (either minimal or maximal) solution according to the optimized parameter $\rho_A$. An *optimization* algorithm $M_A$ returns a best solution.

Denote by $\rho_A(x)$ the optimal solution, namely the value $\rho_A(x, w)$ for the best $w$.

An *approximation algorithm* $M_A$ for a given optimization problem $A$ comes up with a solution $w$ that is not necessarily best for $x$, nevertheless, deviates from the optimal by a known *approximation ratio* $c$. Namely, say $A$ is a maximization problem, then the outcome of $M_A$ is a good solution for $x$ satisfying:

$$\frac{\rho_A(x, M_A(x))}{\rho_A(x)} > c$$

Note that here $c$ is a real number between 0 and 1. For a minimization problem, we can either replace the left-hand-side by its inverse, or alternatively, define $c$ to be at least 1 and require the left-hand-side to be smaller than $c$.

A note regarding the approximation ratio: It is sometimes defined as a number between 0 and 1 and sometimes as a number larger than 1. These two are easily interchanging (taking the inverse) as according to both definitions 1 is naturally the optimal ratio and an approximation is better the closer it is to 1.

Now, let us define a gap version of an optimization problem, which is the closest we can make an optimization problem look like a decision problem, for the purpose for the consequent analysis. For a maximization problem $A$ define the *gap problem* $\textsf{gap}\text{-}A[\alpha, \beta]$ to partition all inputs into three parts:

- good: all $x$'s so that, $\rho_A(x) \geq \beta$.

- bad: all $x$'s so that $\rho_A(x) \leq \alpha$.
- don't care: all other $x$'s.

An algorithm for $\textsf{gap-}A$ must accept all good inputs, reject all bad inputs, however, may either accept or reject all the "don't care" inputs.

**Claim 0.1.** *Given a polynomial time c-approximation algorithm $M_A$ for optimization problem $A$, there is a polynomial-time algorithm that solves $\textsf{gap-}A[\alpha, \beta]$ for all $\alpha$ and $\beta$ so that $\frac{\alpha}{\beta} \leq c$.*

*Proof.* For maximization — apply $M_A$ and accept if $\rho_A(x, M_A(x)) > \alpha$. (For minimization accept if $\rho_A(x, M_A(x)) < \beta$). In case $x$ is a "good" input thus $\rho_A(x) \geq \beta$, the solution $w_x$ obtained by $M_A$ satisfies $\rho_A(x, w_x) > c \cdot \rho_A(x) \geq \alpha$ and therefore $x$ is accepted. In case $\rho_A(x) \leq \alpha$, $M_A$ cannot return a better solution, hence $x$ is rejected. (A similar argument applies for a minimization problem). □

**Corollary 0.2.** *If $\textsf{gap-}A[\alpha, \beta]$ is NP-hard, then so is the $\frac{\alpha}{\beta}$-approximation of $A$.*

## gap-3SAT and PCP

Next, let us consider a basic Maximization, Approximation and gap-version for 3SAT formulas.

**Claim 0.3.** *Given a 3CNF formula with exactly 3 independent literals in each clause, there exists an assignment that satisfies 7/8 of the clauses.*

*Proof.* A random assignment expectedly satisfies 7/8 fraction of the clauses. This follows as each clause is satisfied with that probability and one can then apply linearity of expectation for the overall average. If the average is 7/8, there must be at least one assignment satisfying that fraction. □

Hence, one can expect in general a 3CNF formula to have 7/8 of its clauses satisfied by some assignment. We've already seen that the computational problem of whether such a formula is satisfied is NP-hard. What then about the most extreme gap-problem imaginable hard, which calls for distinguishing between the case where the formula is completely satisfied and the case where the fraction of clauses of the formula satisfied is only slightly larger than 7/8?

**Theorem 1** (PCP). *For any $\epsilon > 0$, $\textsf{gap-}3SAT[7/8 + \epsilon, 1]$ is NP-hard.*

*Proof.* Attached to the margin... □

The theorem implies that for any $L \in NP$ there is a poly-time Karp-reduction so that for any input $x$ the resulting 3CNF formula is satisfiable iff $x \in L$, furthermore, in case $x \notin L$, one can satisfy only $\leq 7/8 + \epsilon$ of the clauses.

Consider a verifier that needs to verify whether a given formula is satisfiable, however, while reading only a constant number of bits of an assignment to the formula's variables. The verifier may decide on which variables to read, once the assignment is set and using some random bits; and is allowed some small error probability thus may accept an unsatisfied formula, however with probability close to 0. The verifier can choose randomly a small number of clauses and read the variables these clauses include, then verify the assignment to these variables satisfy the chosen clauses. The probability this process fails to discover the formula is only $(7/8+\epsilon)$-satisfiable is $(7/8+\epsilon)^l$ where $l$ is the number of clauses chosen randomly. (The verifier fails to detect that case only if all chosen clauses happen to be satisfied by the assignment).

Since any NP-language $L$ can be reduced to gap-3SAT, one can thus verify probabilistically membership in any $L \in NP$.

## Gap-Preserving Reductions

A *gap-preserving* Karp-reduction from a gap problem $\mathsf{gap}\text{-}A[\alpha_A, \beta_A]$ to another gap-problem $\mathsf{gap}\text{-}B[\alpha_B, \beta_B]$ is a Karp-reduction $f$ from $A$ to $B$ so that for a good input $x$ for $A$, $f(x)$ is a good input for $B$, while for a bad input $x$ for $A$, $f(x)$ is bad for $B$. The don't-care inputs of $A$ can be mapped to any input.

Now, let us utilize the above PCP theorem to show approximating some optimization problems NP-hard, via gap-preserving reductions.

Let us start with the CLIQUE/Independent-set problem, and concentrate on a special type of a CLIQUE problem, where the graph $G = (V, E)$ comprises $m$ pairwise disjoint independent-sets, each of which has 3 vertexes. Such graph is referred to as $(m, 3)$*-partite.*

Showing NP-hardness for that special case certainly implies hardness for the general case, furthermore, one may use the special structure of the graph for subsequent reductions.

**Lemma 0.4.** *$\mathsf{gap}\text{-}SAT[7/8+\epsilon, 1] \preceq_L \mathsf{gap}\text{-}CLIQUE[(7/8+\epsilon)m, m]$ where the resulting graph is $(m, 3)$-partite.*

This —combined with the PCP theorem above— implies the problem is NP-hard.

*Proof.* Apply the same reduction, used earlier to show CLIQUE is NP-hard, from 3SAT to CLIQUE. This reduction results in a $(m, 3)$-partite-graph, where $m$ is the number of clauses in the formula. The exact same reduction turns out to be gap-preserving. Completeness is as before. For soundness, note that a clique of size $l$ in the resulting graph implies an assignment to the formula that satisfies at least $l$ of the clauses. □

**Corollary 0.5.** *Approximating MAX-CLIQUE to within a ratio $7/8+\epsilon$ is NP-hard.*

## The Generalized, Constraint-Graph problem

**Definition 1** (Constraint Graph)**.** *A* constaint graph *is a tuple $U = (V, E, \Sigma, \Phi)$ where $(V, E)$ is a graph, $\Sigma$ is a set of colors for the vertexes $V$, and $\Phi\colon E \to P[\Sigma^2]$ assigns one constraint to the colors at the end of each edge, specifying pairs of colors satisfies that constraint (these are distinct predicates, one for each edge).*

*Once assigning a color to each vertex $A\colon V \to \Sigma$, each constraint is either* satisfied *or* unsatisfied*. For a partial assignment $A\colon V \to \Sigma \cup \{\bot\}$, each constraint is either* satisfied*,* unsatisfied*, or* undefined *if the color at any of its ends is undefined.*

*For now, let us consider the following maximization problem for a given constraint-graph refer to as $MAX_V - CSG$: color the largest number of vertexes (in a partial assignment) so that no constraint is unsatisfied (all are either satisfied or undefined).*

*Later, we will consider a different maximization problem $MAX_E - CSG$, where all vertexes are colored (full assignment) and the optimized parameter is the number of constraints satisfied.*

We may now consider the $gap_V - CSG$ general problem and utilize it in the following sequence of gap-preserving reductions.

Let us denote by $kCSG$ a constraint-graph problem where the number of colors (possible assignments to the vertexes) is $k$.

What can we say about the complexity of $CSG$? Let us first observe that $\mathsf{gap}_V$-$3CSG[7/8+\epsilon, 1]$ is NP-hard. This follows directly from the PCP theorem above, and a simple gap-preserving reduction from $\mathsf{gap}$-$3SAT[7/8+\epsilon, 1]$ to this problem:

**Claim 0.6** (CSG: 3-and-7/8). *gap-$3SAT[7/8+\epsilon, 1] \preceq_L$ gap$_V$-$3CSG[7/8+\epsilon, 1]$*

*Proof.* Given a 3SAT formula $\varphi$ of $m$ clauses, construct a $CSG$ instance $U$, where $V$ —the set of vertexes— has one vertex for each clause. Assume some order between the 3 literals in each clause, the 3 colors a vertex can be assigned each correspond to one literal in the corresponding clause. $U$ has a constraint between any two vertexes that correspond to clauses that share the same variable, however one occurrence is negated while the other is not. In that case, the two colors that correspond to the two conflicting literals do not satisfy the constraint. If there are more than one shared variables, no conflicting colors can satisfy the constraint. All other color combinations do satisfy that constraint.

Completeness and soundness are straightforward. □

## Independent-Set

One of the benefits of using the more general $CSG$ problem is that no matter what specific constraints the problem imposes, it is still always naturally reducible to the Independent-set/CLIQUE problems. The only difference between the fraction of vertexes colored and the size of the Independent-set in the resulting graph, as a fraction of the size of the graph, is that it is divided by the number of colors $k$:

**Claim 0.7** (CSG-to-IS). *gap$_V$-$kCSG[\delta, 1] \preceq_L$ gap-$IS[\delta/k, 1/k]$*

*Proof.* Given a $CSG$ instance $U = (V, E, \Sigma, \Phi)$, construct $G = (V \times \Sigma, E')$, namely, there is one vertex in $G$ for each pair of a vertex in $U$ and a color for it. The vertexes in $G$ that correspond to a single vertex in $U$ form a clique in $G$, and each constraint of $U$ imposes a bipartite graph between the corresponding two sets of vertexes in $G$:

$$E' = \{((u,i),(u,j)) \mid i \neq j\} \;\cup\; \{((u,i),(v,j)) \mid (i,j) \notin \Phi(u,v)\}$$

where $\Phi(u,v)$ in case there is no constraint between $u$ and $v$ of $U$ is interpreted to always be satisfied (it is the complete set of pairs of colors).

Let us show completeness and soundness for the reduction.

**Completeness:** If there is a coloring $A\colon V \to \Sigma$ for $U$ satisfying all constraints, the set of vertexes in $G$

$$I_A = \{(v, A(v)) \mid v \in V\}$$

constitutes an independent-set in $G$ of size

$$|I_A| = |V| = \frac{|V \times \Sigma|}{|\Sigma|}$$

(one in each clique that correspond to a vertex in $U$). Two distinct vertexes in $I_A$ cannot belong to a clique that corresponds to a vertex $v \in V$, and if there were an edge between $(v, i)$ and $(u, j)$ and both belonged to $I_A$, $A$ would not have satisfied $\Phi(u, v)$.

**Soundness:** Given an Independent-set $I$ in $G$, since every clique in $G$ that correspond to a vertex $v$ of $U$ can have at most one representative in $I$, the following partial-coloring is well defined

$$A_I(v) = i \mid (v, i) \in I$$

namely, each vertex $v$ in $U$ is colored by the color $i$ whose corresponding vertex $(v, i)$ in $G$ belongs to $I$ if such exists. $A_I$ colors $|I|$ vertexes in $U$ and satisfies all constraints between those. Note that $|V \times \Sigma| = |V| \cdot |\Sigma|$ thus $A_I$ colors in $U$ a fractional size larger by a factor $|\Sigma|$ than the fractional size of $I$ in $G$. □

## Amplifying the Gap

Now that we have a basic hardness of gap-CSG and know that any hardness result for CSG translates to Independent-set, let us "amplify" the gap in our CSG hardness, thereby obtain a stronger hardness result for Independent-set. The following claim asserts that one can amplify the fraction of vertexes colored by a power of $l$, which would only cost in a blowup of the number of colors (and the size of the instance) by a power of $l$ as well:

**Claim 0.8** (Amplification)**.** $\mathsf{gap}_V\text{-}kCSG[\delta, 1] \preceq_L \mathsf{gap}_V\text{-}k^l CSG[\delta^l, 1]$

*Proof.* Given a $CSG$ instance $U = (V, E, \Sigma, \Phi)$, construct a $CSG$ instance $U' = (V^l, E', \Sigma^l, \Phi')$. Let us use the notation $\vec{v}$ to denote a vector of $l$ vertexes in $V$, namely a vertex in $U'$, $\vec{v} \in V^l$. Similarly, use $\vec{a} \in \Sigma^l$ to denote a color of $U'$, which we interpret as one color $\vec{a}_i$ for each vertex $\vec{v}_i \in V$. $U'$ has a constraint for every two vertexes $\vec{v}$ and $\vec{u}$ that either share a common vertex of $U$ or so that there is a constraint in $U$ between a vertex of $\vec{v}$ and a vertex of $\vec{u}$:

$$E' = \{(\vec{v}, \vec{u}) \mid \exists i, j \text{ s.t. } \vec{v}_i = \vec{u}_j\} \;\cup\; \{(\vec{v}, \vec{u}) \mid \exists i, j \text{ s.t. } (\vec{v}_i, \vec{u}_j)) \in E\}$$

The constraint $\Phi'(\vec{v}, \vec{u})$ cannot be satisfied if the same vertex of $U$ is assigned different colors, or if the coloring violates a constraints of $U$:

$$\vec{v}_i = \vec{u}_j \text{ and } \vec{a}_i \neq \vec{b}_j \quad \Rightarrow \quad (\vec{a}, \vec{b}) \notin \Phi'(\vec{v}, \vec{u})$$

$$(\vec{v}_i, \vec{u}_j) \in E \text{ and } (\vec{a}_i, \vec{b}_j) \notin \Phi(\vec{v}_i, \vec{u}_j) \quad \Rightarrow \quad (\vec{a}, \vec{b}) \notin \Phi'(\vec{v}, \vec{u})$$

All other pairs of colors in $\Sigma^l$ do satisfy the constraint $\Phi'(\vec{u}, \vec{v})$.

**Completeness:** Assuming a satisfying coloring for $U$, $A\colon V \to \Sigma$, the coloring $A'\colon V^l \to \Sigma^l$ for $U'$ defined by

$$A'(\vec{v}) = \vec{a} \text{ where } \vec{a}_i = A(\vec{v}_i)$$

satisfies all constraints of $U'$ (all vertexes $v \in V$ are assigned the same color $A(v)$ everywhere, and these colors satisfy all the constraints of $U$).

**Soundness:** A consistent partial-coloring of $U'$

$$A'\colon V^l - \to \Sigma^l$$

imposes the following partial-coloring $A\colon V - \to \Sigma$ for $U$, which colors each vertex by the unique color it is assigned in one of the vertexes of $U'$ if such exists:

$$A(v) = a \text{ where } \exists \vec{v} \in V^l \text{ s.t. } \vec{v}_i = v \text{ and } A'(\vec{v})_i = a$$

This partial-coloring is well defined as there could not be more than one color thus assigned to a vertex $v \in V$ (this would violate the first type of constraints of $U'$, demanding all colors for a vertex $v \in V$ to be consistent in $A'$). The coloring $A$ is also consistent as the second type of constraints in $U'$ ensures all colors are consistent according to $\Phi$.

Let us then compare the fractional size of the set of vertexes colored by $A'$ and by $A$: if a vertex $\vec{v} \in V^l$ is colored by $A'$ it must be that for all $i$, $\vec{v}_i$ is colored by $A$, hence

$$\frac{|A'^{-1}(\Sigma^l)|}{|V^l|} \leq \left(\frac{|A^{-1}(\Sigma)|}{|V|}\right)^l$$

This is the case simply as any $l$-sequences colored by $A'$ must have all its items colored by $A$.

This completes the soundness proof as if the fraction of vertexes of $U$ colored by $A$ is bounded from above by $\delta$ (this is the case where $U$ is a "bad" instance), the fraction of vertexes of $U'$ colored by $A'$ is bounded from above above by $\delta^l$. □

One can now conclude the Independent-set/CLIQUE problems are hard to approximate to within any constant:

**Corollary 0.9.** *For any constant $\delta > 0$ there exists some constant $k$ so that $\mathsf{gap}\text{-}IS[\delta/k, 1/k]$ is NP-hard.*

*Proof.* By the PCP Theorem 1 and claim 0.6, $\mathsf{gap}_V\text{-}3CSG[7/8+\epsilon, 1]$ is NP-hard. By the amplification claim 0.8 just proved it follows $\mathsf{gap}_V\text{-}CSG[\delta, 1]$ is NP-hard for any constant $\delta > 0$ while the number of colors $k$ is a function of $\delta$ however still constant. Combined with the gap-preserving reduction from $CSG$ to $IS$ of claim 0.7 the corollary follows. □

## Chromatic Number

Next, let us consider the *Chromatic Number* problem. A *coloring* of a graph $G$ is a partitioning of $G$'s vertexes to sets, each of which is an independent-set. The Chromatic Number of a graph, denoted $\chi(G)$, is the smallest number of independent-sets in a coloring of $G$.

This problem is a minimization problem unlike the problems we have tackled so far. Nevertheless, if one looks at the independent-set graph that results from the reduction above, one sees that, in the good case, it has an independent-set of fractional size $1/|\Sigma|$. If one can change the construction so that this independent-set would have $|\Sigma|$ copies that are pairwise disjoint, they would altogether cover the entire graph. If that transformation can be achieved while not changing the size of the largest independent-set in $G$, in the bad case the chromatic number of the graph would be considerably higher.

This is our next goal. For that purpose let us define some special case of the $CSG$ problem, one that when the CSG-to-IS reduction is applied to, results in a graph satisfying these requirements.

A $qCSG_\Delta$ instance is a $CSG$ instance with its $q$ colors being $[0..q-1]$, and where each constraint $\Phi(u,v)$ is specified by a *set of allowed differences* $\Delta(u,v) \in P[[0..q-1]]$ modulo $q$, between the colors of $u$ and $v$. Specifically

$$\Phi(u,v) = \{(i,j) \mid i-j( \mod q) \in \Delta(u,v)\}$$

The hardness for this special type of $CSG$ can be utilized to show hardness for the chromatic number:

**Claim 0.10.** $\mathsf{gap}_V$-$qCSG_\Delta[\delta,1] \preceq_L \mathsf{gap}$-$\chi[q,q/\delta]$

*Proof.* Given a $qCSG_\Delta$ instance $U=(V,E,q,\Delta)$ apply the same reduction as in the proof of claim 0.7, resulting in a partite graph $G=(V\times[0..q-1],E')$, with each clique of size $q$.

**Completeness:** Given a satisfying assignment $A\colon V\to[q]$, the set $\{(v,A(v))\}$ is an independent set of $G$. Furthermore, the assignment $A^d$, defined as

$$A^d(v) = A(v)+d \mod q$$

is a satisfying assignment as well. The independent-sets that result from all such satisfying assignments $A^d$ are pairwise disjoint, therefore, altogether cover all of $G$ with $q$ independent-sets.

**Soundness:** Assuming one can color at most $\delta$ of the vertexes of $U$ without dissatisfying a constraint, the largest independent-set of $G$ is of size $\delta|V|$. The ratio between the number of vertexes in $G$ and the size of the largest independent-set in $G$ is clearly a lower-bound for the number of independent-set in a cover of $G$, hence

$$\chi(G) \geq \frac{|V|\cdot q}{\delta|V|} = \frac{q}{\delta}$$

□

Next, let us show a reduction, from a general $CSG$ problem to $qCSG_\Delta$ where the number of colors $q$ grows to be polynomial in the original size of the graph:

**Lemma 0.11.** *$\mathsf{gap}_V$-$kCSG[\delta, 1] \preceq_L \mathsf{gap}_V$-$qCSG_\Delta[\delta, 1]$ where $q$ is polynomial in the size of the $CSG$ instance and in $k$.*

*Proof.* Let us first assume the constraints form a complete graph in our $CSG$ problems: if there is no constraint between two vertexes, just add a trivial constraint, which is always satisfied (since we try to maximize the fraction of vertexes colored, this does not cause any change in our parameters). We would therefore omit the set of edges $E$ in the following analysis.

Starting with a $CSG$ instance $U = (V, \Sigma, \Phi)$ construct a $qCSG_\Delta$ instance $U' = (V, [0..q-1], \Delta)$; for that purpose let us use a special mapping

$$T \colon V \times \Sigma \to [0..q-1]$$

with the property that for every triplet of elements in its domain, the sum of maps is unique:

**Claim 0.12.** *For any set of elements $X$ and for $q \geq |X|^5$, one can efficiently construct a mapping $T \colon X \to [0..q-1]$ so that for any $x_1, x_2, x_3, x_4, x_5, x_6 \in X$, if*

$$T(x_1) + T(x_2) + T(x_3) \equiv T(x_4) + T(x_5) + T(x_6) \mod q$$

*then*

$$\{x_1, x_2, x_3\} = \{x_4, x_5, x_6\}$$

*Proof.* Assume without loss of generality that $X = [1..|X|]$, and assign values to $x \in X$ one by one in order. In each step, once $[1..l]$ have already been assigned without contradicting the condition, make sure the next value assigned, $T(l + 1)$, avoids, for any $u, v, w, y, z \in [1..l]$ the value

$$T(u) + T(v) + T(w) - T(y) - T(z) \mod q$$

This ensures that altogether the values assigned to $[1..l + 1]$ do not dissatisfy the requirement. The number of elements in $[0..q - 1]$ disqualified at the $l + 1$st stage is therefore $\leq l^5$, and if $q \geq |X|^5$ this process can fully specify $T$ with the requirement satisfied. □

Observe that as $T$ is unique for triplets, it is also unique for pair and for single elements as well.

Back to the proof of the lemma and the construction of $U'$:

$U'$ has the same vertex-set as $U$, and we set $q$ by applying $T$ to $V \times \Sigma$ (thus $q = |V \times \Sigma|^5$); we have yet to specify $\Delta$:

$$\Delta(u, v) = \{T(u, i) - T(v, j) \mod q \mid (i, j) \in \Phi(u, v)\}$$

Namely, the set of differences allowed between vertex $u$ and vertex $v$ consists of all differences that result from applying $T$ to a pair of colors that satisfy the original constraint.

**Completeness:** If $A$ is a satisfying assignment to $U$, then

$$A'(v) = T(v, A(v))$$

is a satisfying assignment to $U'$, as all the differences originate from satisfying pairs of colors.

**Soundness:** Let us show that the number of vertexes that can be colored, satisfying all constraints between them, is the same in $U$ and in $U'$. In fact, we'll show a stronger assertions, namely that each coloring of $U'$ originates from a coloring of $U$:

**Claim 0.13.** *Given a partial coloring, which colors a subset $V' \subseteq V$*

$$A' \colon V' \to [0..q-1]$$

*and satisfies all constraint between $u, v \in V'$ in $U'$, there is* **a unique** *$d$ and a coloring $A \colon V' \to \Sigma$ so that for every $v \in V'$*

$$A'(v) \equiv T(v, A(v)) + d \mod q$$

*and furthermore, $A$ satisfies all constraints between $u, v \in V'$ in $U$.*

*(That is, subtracting $d$ from all colors in $A'$ and then applying the inverse of $T$ results in a satisfying coloring for $U$ of the same set $V'$.)*

*Proof.* If $V' = \{u, v\}$ that is $|V'| = 2$, the fact that $T$ is unique in pairs implies there is a unique shift $d$ such that

$$A'(u) \equiv T(u, i) + d \mod q$$

and

$$A'(v) \equiv T(v, j) + d \mod d$$

If there were also a different $(i', j') \in \Phi(u, v)$ and $d'$ so that

$$A'(u) \equiv T(u, i') + d' \mod q$$

and

$$A'(v) \equiv T(v, j') + d' \mod d$$

then $T$ would have violated the uniqueness in pairs property as

$$T(u, i) + T(v, j') \equiv A'(u) + A'(v) - d - d' \equiv T(u, i') + T(v, j) \mod q$$

In case $|V'| = 3$, say $V' = \{u, v, w\}$, we have just proved that for each of $uv, vw, wu$ there is a unique pair of colors satisfying

$$(i, j) \in \Phi(u, v) \text{ such that } T(u, i) - T(v, j) \equiv A'(u) - A'(v) \mod q$$

$$(j', k) \in \Phi(v, w) \text{ such that } T(v, j) - T(w, k) \equiv A'(v) - A'(w) \mod q$$

and

$$(k', i') \in \Phi(w, u) \text{ such that } T(w, k') - T(u, i') \equiv A'(u) - A'(v) \mod q$$

On the other hand, the sum of the following 3 differences is clearly

$$A'(u) - A'(v) \quad + \quad A'(v) - A'(w) \quad + \quad A'(w) - A'(u) \equiv 0 \mod q$$

as each item is once added and once subtracted. This can be rewritten as

$$T(u,i) - T(v,j) \quad + \quad T(v,j') - T(w,k) \quad + \quad T(w,k') - T(u,i') \equiv 0 \mod q$$

which by the property of $T$ (uniqueness of sum over triplets) implies that

$$i = i', j = j' \text{ and } k = k'$$

and we have proven that the shift is the same.

As for the cases where $|V'| > 3$, one only needs to consider the pairwise shift $d_{uv}$ for every $u, v \in V'$. If they are not all the same, there must be a triplet $u, v, w$ which constitute an inconsistent triplet. We just proved, however that there is no such triplet. $\square$

This now completes the proof of the lemma.

$\square$

Altogether, we may conclude that approximating the chromatic number of a graph is NP-hard to approximate to within any constant factor.