IDA PAPER P-2347

RELATI 'E APPROXIMATION BOUNDS FOR NP-COMPLETE OPTIMIZATION PROBLEMS

Kevin J. Rappoport

March 1990

Prepared for
Computer and Software Engineering Division (CSED)

IDA
INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311-1772

IDA Log No. HQ 89-035063

DEFINITIONS

IDA publishes the following documents to report the results of its work.

Reports

Reports are the most authoritative and most carefully considered products IDA publishes. They normally embody results of major projects which (a) have a direct bearing on decisions affecting major programs, (b) address issues of significant concern to the Executive Branch, the Congress and/or the public, or (c) address issues that have significant economic implications. IDA Reports are reviewed by outside panels of experts to ensure their high quality and relevance to the problems studied, and they are released by the President of IDA.

Group Reports

Group Reports record the findings and results of IDA established working groups and panels composed of senior individuals addressing major issues which otherwise would be the subject of an IDA Report. IDA Group Reports are reviewed by the senior individuals responsible for the project and others as selected by IDA to ensure their high quality and relevance to the problems studied, and are released by the President of IDA.

Papers

Papers, also authoritative and carefully considered products of IDA, address studies that are narrower in scope than those covered in Reports. IDA Papers are reviewed to ensure that they meet the high standards expected of refereed papers in professional journals or formal Agency reports.

Documents

IDA Documents are used for the convenience of the sponsors or the analysts (a) to record substantive work done in quick reaction studies, (b) to record the proceedings of conferences and meetings, (c) to make available preliminary and tentative results of analyses, (d) to record data developed in the course of an investigation, or (e) to forward information that is essentially unanalyzed and unevaluated. The review of IDA Documents is suited to their content and intended use.

The work reported in this document was conducted under IDA's Central Research Program. Its publication does not imply endorsement by the Department of Defense or any other Government Agency, nor should the contents be construed as reflecting the official position of any Government Agency.

This Paper has been reviewed by IDA to assure that it meets high standards of thoroughness, objectivity, and appropriate analytical methodology and that the results, conclusions and recommendations are properly supported by the material presented.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1990	Final

4. TITLE AND SUBTITLE
Relative Approximation Bounds for NP-Complete Optimization Problems

5. FUNDING NUMBERS
CRP 9000-506

6. AUTHOR(S)
Kevin J. Rappoport

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
1801 N. Beauregard St.
Alexandria, VA 22311-1772

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Paper P-2347

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
1801 N. Beauregard St.
Alexandria, VA 22311-1772

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, unlimited distribution: 26 September 1990.

12b. DISTRIBUTION CODE
2A

13. ABSTRACT *(Maximum 200 words)*

IDA Paper P-2347, Relative Approximation Bounds for NP-Complete Optimizatiion Problems, documents a new method for determining lower bounds on the relative accuracy attainable for polynomial-time approximations to certain restricted types of NP-Complete optimization problems. These problems are the bounded optimizations, NP constrained optimizations, and Δ-optimizations. The initial tasking was to investigate up to three promising areas in Complexity Theory. The first area involved developing new separation results for Krentel's OptP classification of Np optimization problems. The second area was to examine the possible extension of Krentel's absolute error bounds for NP approximations to relative error bounds, while the third are involved examination of the EP theory of parallel algorithms by Kruskal, Randolph, and Snir. However, time did not permit investigation of the third area.

14. SUBJECT TERMS
Complexity Theory; NP Optimization Problems; Parallel Algorithms; EP Theory; OptP CLasses; Query Bounds; Approximation Bounds.

15. NUMBER OF PAGES
58

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

IDA PAPER P-2347

RELATIVE APPROXIMATION BOUNDS FOR NP-COMPLETE OPTIMIZATION PROBLEMS

Kevin J. Rappoport

March 1990

INSTITUTE FOR DEFENSE ANALYSES

IDA Central Research Project
CRP 9000-506

PREFACE

The Complexity Theory CRP (9000-506) was initiated to study new developments in Complexity Theory and report on any techniques that may be applicable to IDA tasking. The initial tasking was to investigate up to three promising areas in Complexity Theory. The first area involved developing new separations results for Krentel's OptP classification [Kr87] of NP optimization problems. The second area was to examine the possible extension of Krentel's absolute error bounds for NP approximations to relative error bounds, while the third area involved examination of the EP theory of parallel algorithms by Kruskal, Randolph, and Snir.

The development of new separation results would allow Krentel's absolute error bounds on polynomial-time approximation to be extended to cover more types of NP optimization problems, while the relative approximation results represented new work. The importance of any error bounds was deemed important in algorithm design. By providing a method to determine theoretical limits on the approximability, algorithm designers could spend less time working on problem variations not likely to yield good approximations. EP theory was deemed important since it addressed parallel efficiency and represented an alternative to NC theory in parallel algorithm design. Of these three areas, two were addressed and one was reported on under this CRP.

The first subtask was to seek new proofs of separation for certain OptP classes that were not yet known to be separable, to apply these separations to show absolute approximation bounds for classes where they did not previously apply. Several simple proofs were discovered that showed new separations between certain classes based upon the assumption that NP$\not\subseteq$ DTIME($n^{\log n}$) and other weaker results. As it turned out, these proofs were simplifications of proofs previously discovered by Krentel and Gasarch. Beigel [Be88] recently proved an exhaustive series of separation results that extended Krentel's work. Because of the completeness of this recent work no further refinement of this area was attempted under this CRP.

The second subtask was to investigate the possibility of using Query bounds in the spirit of Gasarch and Krentel to prove *relative* bounds on approximations. Krentel's work provided absolute accuracy bounds. Relative results yield more information, and are generally more useful in practice. Under this CRP a new method was discovered to bound the relative error of approximations for certain limited classes of NP-Complete optimization problems. These results were then extended to yield $\frac{1}{n^k}$-approximation results for several optimization problems.

The third subtask was to investigate the recently developed EP theory that characterizes parallel algorithms by their efficiency. Time did not permit investigation of this area.

This paper was reviewed extensively throughout its life cycle by members of the Institute for Defense Analyses review team: Dr. Eric Roskos, Dr. Cy Ardoin, Dr. Michael Kappel, Mr. Steve Edwards, Mr. Jim Baldo, and Mr. Terry Mayfield.

TABLE OF CONTENTS

1 INTRODUCTION

This paper presents a new method for determining lower bounds on the relative accuracy attainable for polynomial-time approximations to certain restricted types of NP-Complete optimization problems herein referred to as 1) bounded optimizations, 2) NP constrained optimizations, and 3) Δ-optimizations. The method makes use of query analyses with Oracle Turing machines, with the goal to develop a consistent approach for determining lower bounds for accuracy in poly-time approximations. The methods presently allow k-approximation results to be shown for these restricted NP optimization problems and allows $\frac{1}{n^k}$-approximation results to be extended to unrestricted problems. The goal is to find methods that allow k-approximation and ε-approximation results to be extended to unrestricted problems.

The ability to show lower bounds on error has ramifications in algorithm design. Since no exact polynomial-time solutions are known for any NP-Complete optimization problems, algorithm designers must often resort to polynomial-time approximation algorithms for these problems. This process often involves inventing variations of the underlying problem by simplification, substitutions, or other methods in an attempt to find a problem that can easily be approximated. Having theoretical tools to show lower bounds on error for some of these problem variations would allow the algorithm designer to quickly discard approaches that cannot be approximated to the required degree of accuracy.

The first class of problem studied in this paper is the bounded optimization problem. This is a NP optimization problem where the number of elements of the witness that may participate in the optimization is bounded. As an example, consider BOUNDED CLIQUE. This problem consists of a graph G, and a set of distinguished vertices U. We ask how many of the distinguished vertices can be found in any one clique of size k. Since the optimal solution is never more than $|U|$, the optimization is bounded by how large U may be as a function of the vertex set V. Although this problem class is contrived to aid in the query analyses, it yields some problems that are interesting in their own right.

The second type of restricted optimization is the NP constrained optimization. In this problem we are given two instances of an NP-Complete problem I_1, and I_2. I_1 is used to constrain the legal witnesses of I_2, and I_2 is the optimization problem. An example of this type of problem is C MAX SAT. This problem consists of two boolean formulas φ_1, and φ_2 over variables X. The goal is to output the maximum number of clauses in φ_2 that can be simultaneously satisfied by a witness that completely satisfies φ_1.

The third type of optimization is the Δ-optimization. In this problem we are given two instances of a problem I_1, and I_2. The goal is then to compute the optimization on both instances and output their difference. As an example, consider Δ-CLIQUE. In this problem we are given two graphs G_1, and G_2. Let k_1 be the maximal clique in G_1, and k_2 be the maximal clique in G_2. We wish to know how much larger k_2 is than k_1.

The main technique utilized in this paper is oracle query analysis. It is used to establish the number of queries required to solve a problem on a P^{NP} Oracle Turing machine. The query requirements are based on some complexity theoretic assumption such as $P \neq NP$, or $NP \neq R$. That is, the problem will require at least the number of queries specified as long as the complexity theoretic assumption holds. The initial query requirements are established by proofs that directly show the assumption is violated if a base problem can be solved in fewer than the required number of queries. Metric reductions are then used to propagate query requirements to other problems. The reductions are constructed to allow the new query requirements to be determined from the query requirements of the source problem.

Once the query requirements have been established, they can be used to develop lower bounds on the accuracy of polynomial-time approximations assuming the complexity theoretic assumption holds. In this paper k-approximation results are established for several restricted problems. The k-approximation results can be extended to $\frac{1}{n^k}$-approximation results for several unrestricted optimizations problems including CLIQUE, and MAXIMUM INDEPENDENT SET. Extensions to k-approximation and ε-approximations have met with limited success since all unrestricted k-approximation results so far found can be proven by other means.

The remainder of the paper is organized as follows. Section 2 presents the definitions required to develop query requirements and presents two approximation bound proofs. Section 3 presents query requirements for a base problem LEX and uses them to show examples of query requirement proofs for problems in each of the three restricted optimization classes. Section 4 utilizes these results to show approximation bounds for restricted and unrestricted NP optimization problems.

2 APPROXIMATION BOUNDS FROM QUERY BOUNDS

This section establishes the tools required to show k-approximation bounds for NP optimization problems based upon the number of queries required to solve them on a P^{NP} Oracle Turing Machine (OTM). Two theorems are then presented that use query bounds to show lower bounds on relative and absolute error. The first theorem is a modified version of Krentel's theorem for absolute approximation bounds and allows absolute approximation bounds to be expressed in terms of natural problem size measures. A new theorem is then introduced that uses precise query bounds and the number of bits required to express the solution to get relative approximation bounds.

2.1 DEFINITIONS

In this section we introduce the concept of a "natural" measure for bounding queries in an NPC optimization problem. These concepts are used to show theorems for bounding absolute and relative errors of polynomial time approximations. The natural measures will be 1) within a polynomial of any reasonable encoding, 2) natural for bounding the queries required under the intuitive binary search using NP queries, and 3) natural for bounding the size of the solution in bits. Examples include the number of variables for LEX-SAT and other SAT witness problems, or the number of vertices for COLORING, CLIQUE, and other vertex counting graph problems, etc. The concept of a "natural" query measure and notations are formally introduced below.

DEFINITION 2.1.1 Natural measure, A natural measure, $m_\Pi:\Pi \rightarrow N$, is function that relates a "size" to some instance of problem Π in a natural and standard way. For example, $m_\Pi(G) = |V|$ is a natural measure for CHROMATIC NUMBER({V,E})). This will be also be abbreviated as "n-measure".

NOTATION 2.1.2 When dealing with known problem n-measures, the actual function will be explicitly written. For example, for graph problems where the queries are measured in terms of the size of the vertex set we will use |V| instead of $m_{\Pi}(G)$. When discussing unspecified problems with unspecified n-measures we will use the symbol m. Sometimes this will also be written as $m(x)$ to emphasize that the n-measure value is a function of the input instance.

DEFINITION 2.1.3 Query requirement, $Q_{\Pi}: m_{\Pi}(I) \rightarrow N$, is function that describes the number of queries required to solve an instance I of problem Π on P^{NP} machine. The query requirements are relative to some complexity theoretic premise, such as P≠NP, or NP≠RP, etc. Sometimes the query requirements for a specific problem will be written with the problem name as the subscript. For example, Q_{CLIQ} is the query requirement for the problem CLIQUE.

NOTATION 2.1.4 When dealing with problems with known query requirements as a function of their n-measure, we will write the function explicitly. For example, a logarithmic query requirement for Vertex Cover would be written as log|V|. When dealing with unspecified query bounds, the symbol "Q(m)" will be used.

DEFINITION 2.1.5 k-approximation, Problem Π has a polynomial-time k-approximation if there exists a polynomial-time algorithm that guarantees

$$\left\| \frac{\mathrm{Approx}_{\Pi}(x) - \mathrm{opt}_{\Pi}(x)}{\mathrm{opt}_{\Pi}(x)} \right\| \leq k(\mathrm{opt}_{\Pi}(x)).$$

If k(z) is a constant function, the approximation result can also be expressed in terms of overapproximations and underapproximations. An overapproximation approaches a solution from above, while an underapproximation approaches a solution from below. A k-approximation is one that can guarantee $(\forall x)$ Approx(x) $\leq$ (k+1)opt(x) for overapproximations, and $(\forall x)$ Approx(x) $\geq$ (k-1)opt(x) for underapproximations. Both notations will be used in this paper.

2.2 ERROR BOUND THEOREMS

The theorems presented in this section establish lower bounds on the absolute and relative accuracy attainable for polynomial time approximations to NP-Complete optimization problems assuming query requirements $Q(m(x))$. Since the query requirement is conditional on some complexity theoretic assumption such as P≠NP, NP≠RP, etc; this allows lower bounds to be established for various complexity theoretic assumptions. These theorems are stated without reference to this condition to allow them to be more generally applicable. In Section 3 these theorems will be applied assuming P≠NP.

Theorem 2.1 below is an adaptation of Krentel's OptP approximation results showing that the absolute accuracy attainable by a polynomial time approximation is bounded by $2^{Q(m(x))-1}$. Theorems 2.2 and 2.3 impose a relation between query requirements and output bits required to bound the *relative* accuracy of polynomial-time approximations to bounded as well. Corollaries 2.4 and 2.5 give k-approximation bounds when the difference between query requirements and bits required to express the solution is a constant.

THEOREM 2.2.1 If Π requires $Q(m)$ queries then no polynomial-time approximation can guarantee $||\text{Approx}(x)-\text{opt}(x)|| \leq 2^{Q(m(x))-1}$. (NOTE: $||x||$ is absolute value)

PROOF If Π requires $Q(m)$ queries then by definition it cannot be calculated in polynomial time with $Q(m)$-1 queries unless the associated premise is false. If Π can be approximated to within $2^{Q(m)-1}$ in polynomial time, then an OTM algorithm can be built that violates the query requirement as follows. Assume Π can be approximated to within $2^{Q(m)-1}$ in polynomial time. Run the polynomial time approximation so opt(x) is known to within $2^{Q(m(x))-1}$. Then run a deterministic binary search with an NPC oracle to resolve the last $Q(m)$-1 bits and solve Π in $Q(m)$-1 queries. Therefore, any polynomial-time approximation must have

$$||\text{Approx}(x)-\text{opt}(x)|| \leq 2^{Q(m)-1} \text{ infinitely often.}\Diamond$$

We now use the absolute accuracy bound of Theorem 2.1 to show a new result for relative approximation bounds as well. This result can be used to show that for certain problems, no poly-time approximations can guarantee $(\forall x)[\text{Approx}(x) \leq k \bullet \text{opt}(x)]$ below certain cut-off values of k. Since Theorem 2.1 only gives us information about the absolute value of the error, $||\text{Approx}(x)-\text{opt}(x)||$, we need to specify if the approximation approaches the optimum from above or below. Most relative error assessments deal with approximations approaching from above, so Theorem 2.2 deals with this case. A similar result for approximations approaching from below is shown in Theorem 2.3.

It should be noted that although these theorems can bound the accuracy attainable by such approximations, they say nothing about the *existence* of such approximations. Existence is proven by exhibiting an algorithm.

THEOREM 2.2.2 If problem Π requries $Q(m)$ queries and opt(x) never requires more than $Q'(m)$ bits, then no poly-time approximation for Π approaching from above can guarantee $(\forall x)$ Approx(x) $< \left(1+2^{Q(m) - Q'(m) - 1}\right)$opt(x).

PROOF From Theorem 2.1 we know that $\|\text{Approx(x)-opt(x)}\| = \Omega(2^{Q(m)-1})$. In what follows $|x|$ is the length of x in bits. Let opt(x) be such that $(\forall x)[|\text{opt}(x)| \leq Q'(m(x))]$, so opt(x) $\leq 2^{Q(m(x))}$ for all inputs of n-measure m. We can then construct the ratio

$$\frac{\|\text{Approx(x) - opt(x)}\|}{\text{opt(x)}} \geq \frac{2^{Q(m) - 1}}{2^{Q'(m)}} = 2^{Q(m) - Q'(m) - 1} \quad \text{infinitely often}$$

Since the numerator is a lower bound and the denominator is an upper bound, the inequality holds. Recalling that Approx(x) $\geq$ opt(x) and solving to get Approx(x) and opt(x) on different sides, we get

$$\text{Approx(x)} \geq \left(1+2^{Q(m) - Q'(m) - 1}\right)\text{opt(x)} \quad \text{infinitely often.} \Diamond$$

THEOREM 2.2.3 If problem Π requries $Q(m)$ queries and opt(x) never requires more than $Q'(m)$ bits, then no poly-time approximation for Π approaching from below can guarantee $(\forall x)$ Approx(x) $> \left(1-2^{Q(m) - Q'(m) - 1}\right)$opt(x).

PROOF Using reasoning similar to Theorem 2.2 and recalling that Approx(x) $\leq$ opt(x), we get

$$\text{Approx(x)} \leq \left(1-2^{Q(m) - Q'(m) - 1}\right)\text{opt(x)} \quad \text{infinitely often.} \Diamond$$

COROLLARY 2.2.4 If the query requirements and optimization constraint for problem Π differ by an additive constant κ, then no poly-time approximation for Π approaching from above can guarantee $(\forall x)[\text{Approx}(x) < k \cdot \text{opt}(x)]$ for

$$k \leq \left(\frac{1 + 2^{\kappa+1}}{2^{\kappa+1}} \right)$$

PROOF By Theorem 2.2. ◊

COROLLARY 2.2.5 If the query requirements and optimization constraint for problem Π differ by an additive constant κ, then no poly-time approximation for Π approaching from below can guarantee $(\forall x)[\text{Approx}(x) > k \cdot \text{opt}(x)]$ for

$$k \geq \left(\frac{2^{\kappa+1} - 1}{2^{\kappa+1}} \right)$$

PROOF By Theorem 2.3. ◊

3 QUERY BOUNDS FOR RESTRICTED NP OPTIMIZATION PROBLEMS

This section presents precise query bounds for several restricted NP optimization problems in terms of "natural" problem size measures,. The query bounds Q(m) describe the number of NP queries required to solve the problem on a P^{NP} oracle machine, and are constructed assuming P≠NP. That is, no P^{NP} oracle machine can guarantee a solution with fewer than Q(m) queries unless P=NP. Additionally, Q(m) will be described as a function of some natural measure of the input problem size such as variables in a boolean formula, vertices in a graph, etc. The proofs follow in two steps. First, query requirements are established for a base problem. Second, standard metric reductions are then used to propagate the query bounds to other problems.

The query bound proofs for the base problem make use of the work of Krentel [Kr87], and improvements made by Beigel [Be88], for establishing query bounds for the base problem C^2-$LEX_{(1-\varepsilon)\log n}$. The query requirements for the base problem C^2-$LEX_{(1-\varepsilon)\log n}$ is established in two steps. First, the problem is shown to be NP-Hard by showing a reduction to the NP version C^2-SAT. C^2-$LEX_{(1-\varepsilon)\log n}$ is then shown to require $(1-\varepsilon)\log n$ queries unless P=NP.

Once the base problem is established, metric reductions are exhibited to establish query requirements for several restricted NP optimization problems. Reductions are based partly on the OptP proofs of Gasarch and Pearlman [GP88], and appropriate modifications of NP proofs in the literature. The query bound proofs work by carefully examining reductions that preserve the number of queries required for solution on a P^{NP} oracle machine. These examinations allow the optimization constraints to be carefully measured in terms of the natural problem metrics. Care is also taken to ensure that the optimization constraints match the query requirements so that the theorems of Section 2 can be used to obtain bounds on relative error. Since most of these reductions are adaptations of well-known reductions in the literature, only a brief sketch is presented for several reductions.

3.1 NAMING CONVENTION

The naming convention for problems in this section will be to place the optimization bound as a subscript to the problem. Thus $\text{CLIQUE}_{f(n)}$ is the CLIQUE problem with an optimization bound of $f(n)$.

3.2 PROVING QUERY REQUIREMENTS FOR $C^2\text{-LEX}_{(1-\varepsilon)\log n}$

$C^k\text{-LEX}_{(1-\varepsilon)\log n}$ is a variation of the FP problem $\text{LEX}_{f(n)}$ introduced by Krentel. The only additional restriction is that the number of clauses is at most $|X|^k$ where $|X|$ is the number of variables. $C^k\text{-LEX}_{f(n)}$ can be proven to require $f(n)$ queries for certain $f(n)$ by the methods of Krentel and Beigel. The additional restriction on the number of clauses is required in the reduction to VAR SAT. In this paper we will use $C^2\text{-LEX}_{(1-\varepsilon)\log n}$ and prove it requires $(1-\varepsilon)\log n$ queries by Beigel's approach. A formal definition of $\text{LEX}_{g(n)}$ and $C^k\text{-LEX}_{g(n)}$ follows.

DEFINITION 3.2.1 $\text{LEX}_{g(n)}$ Let φ be a boolean formula in CNF format with an ordered set of distinguished variables $X = \{u_0,\ldots,u_{f(n)}\}$ where n is the total number of variables, and let ω be a witness that satisfies φ. Output the maximum value of $f(\omega)$ over any witness ω, where $f(\omega)$ is defined as.

$$f(\omega) = \max\left(\sum_{\substack{x_i \text{ true in } \omega \\ x_i \in X}} 2^i \right)$$

DEFINITION 3.2.2 C^2-$LEX_{g(n)}$, Let φ be a boolean formula in CNF format with with at most $|V|^2$ clauses where $|V| = n$ is the number of variables in φ, and a set $X = \{u_0, \ldots, u_{f(n)}\}$ of distinguished variables. Let ω be a witness that satisfies φ. Output the maximum of $f(\omega)$ over any witness ω, where $f(\omega)$ is defined as above.

The proof of query requirements proceeds by first showing that a formula in C^k-format is NP-Complete for any k>1. It then follows that C2-$LEX_{(1-\varepsilon)\log n}$ will be an NP-Complete optimization problem. Beigel's proof will be restated with the appropriate modifications to prove that C2-$LEX_{(1-\varepsilon)\log n}$ requires $(1-\varepsilon)\log n$ queries for $\varepsilon>0$ unless P=NP.

3.2.1 C^k-SAT is NP-Complete

C^k-SAT is a variation of SAT where the number of clauses is at most $|X|^k$ and $|X|$ is the number of variables. C^k-SAT is easily seen to be in NP by guessing the satisfying assignment. C^k-SAT can also be shown to be NP-Complete for all k>1 by examination of Cooke's proof for SAT$\in$NP-Complete.

THEOREM3.2.1.1 SAT $\leq^P$ C^k-SAT

PROOF Cooke's proof produces a boolean formula in C^p-SAT for some p since the number of clauses is polynomial in the input tape. Therefore we only need to show that $\forall k < p, p > 1$, C^p-SAT $\leq^p$ C^k-SAT. Let φ be an instance of C^p-SAT with variables X, $|X| = n$, and $|\varphi| = n^p$. Create a new formula ψ by padding φ with a single disjunction of variables $\{u_1, u_2, \ldots, u_j\}$. The number of variables will be adjusted to bring C^k-format into C^p-format.

We get $\psi = \varphi \wedge (u_1, u_2, \ldots, u_j)$. We now want $|\psi| < N^k$, where N=j+n is the number of variables in ψ. Since $|\psi| = |\varphi| + 1 = n^p + 1$, we want $n^p + 1 \leq (n+j)^k$. Solving for j we get $j \geq \sqrt[k]{n^p + 1} - n$. So let $j = n^{\frac{p}{k}}$. This is a polynomial reduction since we are adding only a polynomial number of variables.◊

3.2.2 C^2-LEX$_{(1-\varepsilon)\log n}$ Requires $(1-\varepsilon)\log n$ Queries

From Section 3.1 we saw that C^2-LEX$_{(1-\varepsilon)\log n}$ is an NP-Complete optimization problem since C^2-SAT is NP-Complete. We now show that C^2-LEX$_{(1-\varepsilon)\log n}$ requires $(1-\varepsilon)\log n$ queries. The following sketch is a modification of the proof that appears in Krentel [Kr87] and improved by Beigel [Be88]. It shows that if $(\varphi, \kappa) \in C^2$-LEX$_{(1-\varepsilon)\log n}$ can be solved with less than $(1-\varepsilon)\log n$ queries, then $\varphi \in$ SAT can be solved in polynomial time and P=NP.

The proof works by assuming an OTM M_g exists to solve $(\varphi, \kappa) \in C^2$-LEX$_{(1-\varepsilon)\log n}$ in $((1-\varepsilon)\log n) - 1$ queries. The machine M_g is then used to build a new algorithm to solve $\varphi \in$ SAT in polynomial time. The new algorithm works by simulating M_g on variations of φ until enough information is extracted to construct a witness to φ. The witness is then tested to see if $\varphi \in$ SAT. Beigel shows that the information extracting rounds can be performed in polynomial time so SAT$\in$P. The only modification to the proof is to ensure that it will work for formulas in C^2 format. This is established by the following lemma.

LEMMA 3.2.2.1 Let φ be a boolean formula in C^2-format with n variables. Let $\varphi'=\varphi\wedge\psi$, be a new formula formed by augmenting the original formula φ by ψ where ψ has $\alpha\log(n)2^{\alpha\log(n)}$ clauses and $\alpha\log(n)$ *new* variables. If $\alpha\leq 1$ then φ' is also in C^2 format.

PROOF Since φ is in C^2 format,.we know that $|\varphi| \leq |X|^2$. Let $|X| = n$. We will add new variables so $|X'| = n+\alpha\log(n)$. The C^2-format then requires

$|\varphi'| \leq |X'|^2 = n^2 + 2n\alpha\log n + (\alpha\log n)^2.$

To get φ' we are adding $\alpha\log(n)2^{\alpha\log(n)}$ new clauses so we get

$|\varphi'| = |\varphi| + (\alpha \log n)2^{\alpha\log n} \leq n^2 + \alpha n^{\alpha}\log n.$

So for φ' to be in C^2-format we must have

$n^2 + \alpha n^{\alpha}\log n \ \leq\ n^2 + 2n\alpha\log n + (\alpha \log n)^2$

which surely holds for $\alpha \leq 1$. $\Diamond$

We now present a sketch of Beigel's proof. The only modification is to ensure that Beigel's proof can handle the C^2 formatting restriction.

THEOREM3.2.2.2 C2-$LEX_{(1-\varepsilon)\log n}$ requires $(1-\varepsilon)\log n$ queries unless P=NP.

PROOF Assume that M_g is an OTM that solves an instance of C2-$LEX_{(1-\varepsilon)\log n}$ in $((1-\varepsilon)\log n) - 1$ queries, as long as the input formula is in C^2 format. Simulate the action of M_g on φ for all queries. We can do this since there are no more than $O(\log(n))$ queries. Call the queries variables y_1,

$y_2,\ldots,y_{(1-\varepsilon)\log n-1}$. We now have the lowest $(1-\varepsilon)\log n$ variables, $\{x_1,x_2,\ldots,x_{(1-\varepsilon)\log n}\}$, of a satisfying assignment as a function of $\{y_1,y_2,\ldots,y_{(1-\varepsilon)\log n-1}\}$. Write the functional dependency between the x and y variables as a truth table with $2^{(1-\varepsilon)\log n-1}$ columns and $(1-\varepsilon)\log n$ rows. The truth table can now be written as a boolean formula ψ in CNF format with $((1-\varepsilon)\log n)2^{(1-\varepsilon)\log n-1} < n\log(n)$ clauses.

We now make a new formula $\varphi' = \varphi\wedge\psi$, with variables $\{y_1,\ldots,y_{(1-\varepsilon)\log n-1}, x_1,\ldots,x_n\}$. Note that by lemma 3.2.2.1 above φ' is guaranteed to be in C^2-format. Rename the variables; $\{y_1,\ldots,y_{(1-\varepsilon)\log n-1}, x_{(1-\varepsilon)\log n+1},\ldots,x_n, x_1,\ldots,x_{(1-\varepsilon)\log n}\}$. The variables $\{x_1,\ldots,x_{(1-\varepsilon)\log n}\}$ at the end of the list are now the dependent variables, and all others are the independent variables. The new formula is now processed repeatedly using the same simulation of M_g until all of the x_i variables are known in terms of y variables. We now try all assignments to the y variables to get the values of the x variables. If any of them is a satisfying assignment then φ is satisfyable, otherwise it is not.

Krentel and Beigel show that the above process can be accomplished in polynomial time if M_g exists. Thus if we can solve C^2-$\text{LEX}_{(1-\varepsilon)\log n}$ in fewer than $(1-\varepsilon)\log n$ queries then P=NP.◊

3.3 QUERY REQUIREMENTS FOR BOUNDED OPTIMIZATIONS

This section establishes query bounds for bounded optimizations. These are problems in which the number of elements in the problem that are allowed to participate in the optimization is bounded as some function of the input size. The problems in this section are proven to have specific query bounds $Q(m)$ unless P=NP. That is, no P^{NP} oracle machine can guarantee a solution with fewer than $Q(m)$ queries unless P=NP.

All problems are reduced to a base problem called C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$. C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$. is an instance of SAT where a set of $(N/2)^{(1-\varepsilon)}$ of the variables are "marked", and the number of clauses is at most quadratic in the number of variables. The problem is to output the maximum number of marked variables that are true in any satisfying assignment. C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$ is proved to require $(1-\varepsilon)\log(N/2)$ queries unless P=NP by reduction to C^2-$LEX_{(1-\varepsilon)\log n}$ from Section 2. Several other problems are then reduced to this base problem to obtain query bounds.

3.3.1 Bounded Optimization Problems

A bounded optimization problem is stated in three parts; π, f, and M. The first part, π, is an NP problem definition including the requirements for a witness, ω. The second part is the optimization function $f(\omega)$ over the witnesses. The third part, M, is the optimization constraint and bounds the maximum value that the optimized problem may take as a function of the input problem size. This constraint may restrict the number of elements in a problem instance that can participate in the optimization.

As an example consider a whimsical version of the Travelling Salesman Problem. This problem is constructed by requiring the TSP route to be a Hamiltonian cycle that maximizes the weights from some preferred edge set. That is, we make a TSP route between all cities, but wish the route to use certain preferred city to city flights when possible. This could be called the Frequent Flyer TSP since it seeks to maximize mileage on certain selected legs of the excursion, presumably those that have been designated as "bonus flights" by the airline. Let the preferred edge set be P, this problem is then expressed as

$f(\omega)_{FF\ TSP}$: $\max_{\omega} \sum_{\substack{e_i \in \omega \\ e_i \in P}} w(e_i)$

$M_{FF\ TSP}$: $\sum_{\substack{e_i \in \omega \\ e_i \in P}} w(e_i) \leq \sqrt[3]{\frac{|V|}{16}}$ where $|V|$ is the total number of vertices.

Notice that in $M_{FF\ TSP}$ above, the bound limits the number of edges that participate in the optimization since the sum of preferred edge weights must be less than the total number of edges in any Hamiltonian cycle through the graph. It is also a bound on the magnitude of any optimal solution.

3.3.2 Base Problem: C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$

The base problem for all CNPO reductions needs to be able to "count" constraints that are violated. This is accomplished in a variation of SAT by counting true variables from a prespecified set of "marked" variables. As with all NPCO problems, the size of the "marked" set is a function of the problem size. A formal definition of C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$ follows.

DEFINITION 3.3.2.1 C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$

$\pi_{C\text{-}VAR\ SAT}$: A boolean formula φ in CNF format with at most $|V|^2$ clauses where $|V|$ is the number of variables in φ. $X = \{u_0, u_{f(n)}\}$ is an ordered set of distinguished variables where n is the total number of variables. A witness ω is a satisfying assignment to φ.

$f(\omega)_{\text{C-VAR SAT}}$: $\max_{\omega} \sum_{\substack{x_i \text{ true in } \omega \\ x_i \in X}} 1(x_i)$ where $1(x)$ is the constant function outputting 1.

$M_{\text{C-VAR SAT}}$: $opt \leq (N/2)^{(1-\varepsilon)}$ where N is the total number of variables.

We can now show that C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$ requires $(1-\varepsilon)\log(N/2)$ queries unless P=NP by a reduction to C2-$LEX_{(1-\varepsilon)\log n}$. Notice that in the LEX problem, the subscript refers to the length of the output bit vector in terms of the number of variables in the LEX instance. In the VAR SAT problem, the subscript is the size of the "marked" variable set in terms of the number of variables in the VAR SAT instance. The order difference in these functions is due to the fact that the VAR SAT instance only counts elements from the set, while the LEX instance effectively weights the elements by 2^i before counting them.

3.3.2.1 Construction

The proof proceeds by exhibiting a construction of a C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$ instance ψ from an instance φ of C^2-$LEX_{(1-\varepsilon)\log n}$. The construction will be a collection of boolean formulas over new variables from the sets Z, U, C, and the original LEX instance. The Z variables are used to simulate the 2^i weighted X variables from LEX. The U and M variables are unmarked and marked dummy variables that are included only to make the mathematical analysis easier. The Z variables will be used to construct the formula σ, which ensures that a subset of Z will all be true for each distinguished variable x_i from LEX. The dummy variables will be collected into a single clause δ.

Using the procedure below a new formula ψ is created consisting of a concatenation of the formulas φ, σ, and δ. After the construction is complete, the new VAR SAT instance is shown to be in C^2-format if the original LEX instance was. The optimal solution to LEX is then shown to be obtinable from the optimal solution to VAR SAT, completing the reduction.

1) Construction of σ. For each of the distinguished variables $\{x_2,\ldots,x_{(1-\varepsilon)\log n}\}$ in φ create a clause using i new variables $Z = \{z_{i,1},\ldots,z_{i,i}\}$ where i is the subscript of the distinguished variable x_i. The clauses will have the form

$$c_i=\left[(x_i \vee {\sim}z_{i,1})\wedge(x_i \vee {\sim}z_{i,2})\wedge\ldots\wedge(x_i \vee {\sim}z_{i,2^{i-1}})\right]\wedge\left[({\sim}x_i \vee {\sim}z_{i,1})\wedge\ldots\wedge({\sim}x_i \vee {\sim}z_{i,2^{i-1}})\right]$$

The new formula σ will then be

$$\sigma = \left[\bigwedge_{i=2}^{(1-\varepsilon)\log n} c_i\right]$$

2) Construction of δ. δ consists of a single disjunction of all of the dummy variables; $M=\{m_1,m_2\}$, and $U = \{ u_1,\ldots,u_{n-n^{(1-\varepsilon)}}\}$. The new formula will then be $\delta = (m_1 \vee m_2 \vee u_1 \vee \ldots \vee u_{n-n^{(1-\varepsilon)}})$.

3) Let $\psi = \varphi \wedge \sigma \wedge \delta$, where φ is the original instance of C^2-$LEX_{(1-\varepsilon)\log n}$. Mark only the variables in $Z \cup M \cup \{x_1\}$.◊

We now need to show that ψ is indeed in C^2 format, assuming that φ was.

CLAIM 3.3.1.1.1 The instance ψ of C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$ is in C^2-format if the input instance φ is in C^2-format.

PROOF We first need to count up the total number of clauses and variables in ψ, so we first calculate $|Z|$. For each x_i we create 2^{i-1} new z variables which gives us an expression for $|Z|$ as

$$|Z| = \sum_{i=2}^{(1-\varepsilon)\log n} 2^{i-1} = \sum_{j=1}^{(1-\varepsilon)\log n - 1} 2^{j} = 2^{(1-\varepsilon)\log n} - 2 = n^{(1-\varepsilon)} - 2 \text{ new z variables}$$

The total number of variables can now be calculated as

$$|X'| = |Z|+|U|+|M|+|X| = 2|X| = 2n$$

For each x_i we create 2^i new clauses.which gives us an expression for the total number of new clauses in σ as

$$|\sigma| = \sum_{i=2}^{(1-\varepsilon)\log n} 2^{i} = 2^{(1-\varepsilon)\log n + 1} - 4 = n^{2(1-\varepsilon)} - 4$$

Since $|\delta| = 1$ we have the total number of clauses as

$$|\psi| = |\varphi| + n^{2(1-\varepsilon)} - 3$$

The C^2 formatting restriction on φ ensures that $|\varphi| \leq n^2$, and the C^2 formatting restriction on ψ requires that $|\psi| < |X'|^2 = 4n^2$. Recalling that $(1-\varepsilon) < 1$ we get

$$|\psi| < n^2 + n^{2(1-\varepsilon)} - 3 < 2n^2 < 4n^2 = |X'|$$

So ψ is in C^2 format.◊

From the analysis above we can also calculate the number of marked variables in terms of the total number of variables in ψ. Let X' be the complete variable set, and $|X'| = N$. The number of "marked" variables is then $(N/2)^{(1-\varepsilon)}$. This expression is placed in the subscript of C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$. The only task remaining is to show that the solution to C2-$LEX_{(1-\varepsilon)\log n}$ can be obtained from the solution to C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$ in polynomial time.

CLAIM 3.3.1.1.2 The optimal solution to C^2-$LEX_{(1-\varepsilon)\log n}$ can be obtained from the optimal solution to C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$ as, $opt_{LEX} = opt_{VAR\ SAT} - 2$.

PROOF In ψ, the only variables marked will be in $Z \cup M \cup \{x_1\}$. The construction of σ requires all of the z_i variables to be true iff the corresponding x_i is true. The number of z_i variables in ψ for each x_i variable in φ is the exact weight of x_i in LEX, so the number of z_i and x_1 variables true will be exactly the output from LEX. The construction of δ requires that all variables in M be true in the optimal solution, so they are subtracted off. ◊

3.3.2.2 C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$ Requires $(1-\varepsilon)\log(N/2)$ Queries

THEOREM3.3.1.2.1 C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$ requires $(1-\varepsilon)\log(N/2)$ queries unless P=NP.

PROOF From the reduction we see that the number of variables in ψ is $|X'| = N = 2n$, where n is the number of variables in φ, the instance of C^2-$LEX_{(1-\varepsilon)\log n}$. If we can solve ψ in less than $(1-\varepsilon)\log(N/2)$ queries then we could solve φ in less than $(1-\varepsilon)\log n$ queries by converting it to ψ

using the reduction procedure. This violates Theorem 3.2.2.2 so C^2-VAR $SAT_{(N/2)^{(1-\varepsilon)}}$ requires $(1-\varepsilon)\log(N/2)$ queries unless P=NP.◊

3.3.3 WEIGHTED VAR SAT: WV $SAT_{\left(\frac{N}{2}\right)^{1-\varepsilon}}$

WV sat is a generalization of C^2-VAR SAT where the distinguished variables may carry weights ≥ 1. The only restriction is that the total sum of the weights not exceed $(N/2)^{(1-\varepsilon)}$.

DEFINITION 3.3.2.1 WV $SAT_{(N/2)^{(1-\varepsilon)}}$

$\pi_{WV\,SAT}$: A boolean formula φ in CNF format with at most $|V|^2$ clauses where $|V|$ is the number of variables in φ. $X = \{x_0, x_{f(N)}\}$ is a set of distinguished variables with weights $w(x_i)$ and where N is the total number of variables. A witness ω is a satisfying assignment to φ.

$f(\omega)_{WV\,SAT}$: $$\max_{\omega} \sum_{\substack{x_i \text{ true in } \omega \\ x_i \in X}} w(x_i)$$

$M_{WV\,SAT}$: $$\sum_{x_i \in X} w(x_i) \leq \frac{N}{2}^{(1-\varepsilon)}$$

C^2-VAR SAT is a restriction of WV SAT where the formula is required to be in C^2-format and all weights for the distinguished variables are 1. It then follows that WV SAT requires $(1-\varepsilon)\log(N/2)$ queries unless P=NP.

3.3.4 WEIGHTED VERTEX COVER: $\text{WVC}_{\left(\frac{V}{16}\right)^{\frac{1-\epsilon}{3}}}$

WVC is a weakening of the standard minimal vertex cover problem MIN VC. In WVC we are given a graph G and permitted to weight some subset of the vertices. We are seeking the minimum weighted vertex cover for G. That is we seek a vertex cover ofr G that includes the minimum total weight from the weighted elements. It is a weakening of MIN VC since the total weight of all distinguished vertices is not sufficient to allow the MIN VC problem to be calculated by weighting all vertices by 1.

π_{WVC}: A graph G=(V,E), a set of distinguished vertices U with weighting function $w : U \rightarrow \mathbb{N}$, and maximum vertex cover size k. A witness ω is a vertex cover on G that is of size $\leq$ k.

$f(\omega)_{WVC}$: $$\min_{\omega} \sum_{\substack{u_i \in \omega \\ u_i \in U}} w(u_i)$$

M_{WVC}: $$\sum_{u_i \in U} w(u_i) \leq \left(\frac{|V|}{16}\right)^{\frac{1-\varepsilon}{3}}$$

The reduction follows in two stages. The first is a reduction from C^2-VAR SAT to a restricted version of WVC where all weights on distinguished vertices are 1. The query requirement from this restriction then follows.

The reduction from C^2-VAR SAT follows the standard reduction to 3-SAT in [GJ79] with two modifications. The first is that the vertices in each clause cluster are allowed to number up to |X|, one for each of |X| possible vars in the clause. So the cluster of vertices for each clause is then a complete graph on |C| vertices where |C| is the number of variables in the clause. The second modification is that the marked variables from C^2-VAR SAT now allow a var vertex to be similarly marked. The construction follows.

CONSTRUCTION

1) For each variable x_i in φ create two vertices v_i and $\sim v_i$. Place an edge between them.

2) For each clause c_j in φ create a clique of size $|c_j|$.

3) For every variable in clause c_j connect an edge to the appropriate v_i and $\sim v_i$.vertex dependening upon which form of the literal appears in c_j.

4) For each distinguished variable x_i in φ mark the appropriate $\sim v_i$ vertex.

5) Let $k = |X| + \sum_{c_i \in \varphi} |c_i| - |\varphi|$ where |X| is the number of variables in φ and $|\varphi|$ is the number of clauses.

In the construction above, the edges radiating out from the clause cluster are connected to other variable representing vertices. These are referred to as spokes. The edges between elements of a single cluster are referred to as internal edges. The edges between $(v_i, \sim v_i)$ pairs are also internal edges.

Since the marked variables are mapped directly to marked vertices it is easy to see that the maximum number of marked variables in a witness to φ will be |U| - the minimum number of marked vertices in a witness to G. We now only need to show that the construction ensures that the associated formula φ is satisfied iff there is a vertex cover of the specified size. This is established in the following lemma.

LEMMA3.3.2.1 In the above construction.there is a vertex cover of size $\leq k \Leftrightarrow \varphi$ is satisfiable.

PROOF⇐ Assume φ is satisfiable. For every variable that is true place the v vertex into the cover, and for every variable that is false place the ~v vertex into the cover. This uses a total of |X| vertices. Since φ is satisfied by the assignment each clause has at least one variable that is true in that clause. Let u_i be a literal that supports each clause c_i. For each clause cluster c_i, accept all vertices *except* u_i into the vertex cover. Each clause cluster will then have all internal edges covered and the cover is now of size k.

Now, for each clause cluster c_i all spokes are covered except those connected to the supporting literal u_i. But since the supporting literals are true by definition they will also be in the cover and the remaining spokes are covered.

PROOF⇒ All internal edges in each v_i ,~v_i.pair must be covered so any vertex cover must contain at least one vertex in each pair. Thus there are at least |X| (v_i ,~v_i) vertices in the cover. Also, each clause cluster must have all internal edges covered. It can be proven by induction that a k-clique has a minimum vertex cover of k-1 so at least $|c_i|$ -1 vertices from each cluster must be included. Thus there are at least k vertices in the vertex cover of G. Since k is the size limit there are exactly k vertices in the vertex cover and they are distributed as above.

Since each cluster has $|c_i|$ -1 vertices, each cluster has exactly one spoke it cannot cover. But since the vertex cover exists this spoke must have been picked up by the $(v_i, \sim v_i)$ vertex at the other end of the spoke. Let these $(v_i, \sim v_i)$ vertices be the true variables. Since exactly one of each $(v_i, \sim v_i)$ pair is in the cover, they form a proper boolean assignment. Thus there is a boolean assignment where all clauses have at least one supporting true literal.

In the construction the total number of vertices N will satisfy $N \le 2n+n^3 \le 2n^3$, where n is the number of vars in C^2-VAR SAT. The total number of marked vertices is $t(n/2)^{(1-\varepsilon)}$. We must rescale the optimization constraint and query requirement from $(n/2)^{(1-\varepsilon)}$ to $\left(\frac{N}{16}\right)^{\frac{1-\varepsilon}{3}}$ where N is the total number of vertices in WVC.

3.3.5 WEIGHTED MAX INDEPENDENT SET: WMIS$_{\left(\frac{V}{16}\right)^{\frac{1-\varepsilon}{3}}}$

WMIS is a weakening of MAX INDEPENDENT SET because the optimization constraint is insufficient to allow all vertices to be counted on an independent set. The problem consists of, a graph G=(V,E), a minimum independent set size m, and a set of preferred vertices U. It asks for a maximally weighted subset of U that lies on an independent set over V. A formal definition follows.

π_{WMIS}: A graph G=(V,E), a set of distinguished vertices U with weighting function $w : U \rightarrow \mathbb{N}$, and a minimum independent set size m. A witness ω is an independent set on G of size $\geq$ m.

$f(\omega)_{WMIS}$: $$\max_{\omega} \sum_{\substack{u_i \in \omega \\ u_i \in U}} w(u_i)$$

M_{WMIS}: $$\sum_{u_i \in U} w(u_i) \leq \left(\frac{|V|}{16}\right)^{\frac{1-\varepsilon}{3}}$$

The reduction follows simply from WVC. For any graph G let $W \subseteq V$ be a vertex cover of G. Then V-W will be an independent set of G. If W is the vertex cover that minimizes WVC then V-W will be the independent set that maximizes WMIS for the same weighting function and set of distinguished vertices U. The minimum independent set size is set as m = |V| - k, where k is the maximum vertex cover size in WVC, and |V| is the total number of vertices in G.

3.3.6 WEIGHTED CLIQUE: $WCLIQ_{\left(\frac{V}{16}\right)^{\frac{1-\epsilon}{3}}}$

WCLIQ is a weakened generalization of the MAX CLIQ problem since the optimization constraint is insufficient to allow the maximum clique to be found by weighting all vertices by 1. It asks for the maximally weighted subset of the distinguished vertices that lie on any clique in the graph G.

$\pi_{W\,CLIQ}$: A graph G=(V,E), a set of distinguished vertices U with weighting function $w : U \rightarrow \mathbb{N}$, and a minimum clique size m. A witness ω is a clique on G of size m or greater.

$f(\omega)_{W\,CLIQ}$: $$\max_{\omega} \sum_{\substack{u_i \in \omega \\ u_i \in U}} w(u_i)$$

$M_{W\,CLIQ}$: $$\sum_{u_i \in U} w(u_i) \leq \left(\frac{|V|}{16}\right)^{\frac{1-\epsilon}{3}}$$

The reduction follows exactly as for WVC using the complement of the graph G^c.

3.3.7 FREQUENT FLYER: $FF_{\left(\frac{V}{16}\right)^{\frac{1-\epsilon}{3}}}$

The FF problem consists of a weighted directed graph and subset P of the edges that are designated as preferred. The problem is to find the directed Hamiltonian cycle through the graph that includes the maximum weight from the set of preferred vertices. The total path length is of no concern. This problem is actually a Hamiltonian cycle problem, instead of a more complicated Travelling Salesperson problem since the total path length is not bounded. A formal definition follows.

π_{FF}: A weighted graph G=(V,E), and a set of preferred edges P. A witness ω is a Hamiltionian cycle through G.

$f(\omega)_{FF}$: $$\max_{\omega} \sum_{\substack{e_i \in \omega \\ e_i \in P}} w(e_i)$$

M_{FF}: $$\sum_{e_i \in P} w(e_i) \le \left(\frac{|V|}{16}\right)^{\frac{1-\varepsilon}{3}}$$

The reduction follows in two parts. The first part is a reduction from C^2-VAR SAT to a restricted FF problem where all preferred edges get weight 1. The reduction ftom the restricted problem then follows.

The reduction from C^2-VAR SAT follows the construction for reduction from 3-SAT in [AHU79] with the obvious generalization to allow clauses of any length. The preferred edges are chosen as the first cross edge in each H-ladder in the Aho, Hopcroft, and Ullman's construction for any variable that is marked. The total number of vertices N will then satisfy $N \le 2n+n^3 \le 2n^3$, where n is the number of variables in C^2-VAR SAT. The total number of preferred edges is then $(n/2)^{(1-\varepsilon)}$. We then rescale the optimization constraint and query requirement from $(n/2)^{(1-\varepsilon)}$ to $\left(\frac{N}{16}\right)^{\frac{1-\varepsilon}{3}}$ where N is the total number of vertices in the graph.

3.4 CONSTRAINED OPTIMIZATIONS

Constrained optimizations are actually two NP problems Π_1 and Π_2 bound together. The first problem Π_1 is a NP-Complete set problem and constrains the variables in the second problem. The second problem Π_2 is an NP-Complete optimization problem. There is generally a relation between Π_1 and Π_2 that limits the size of Π_2 as a function of the size of Π_1.

This section presents one constrained optimization problem since it is not difficult to convert the proofs from other sections into constrained optimization proofs. It should be noted that these types of problems are not particularly interesting as realistic optimization problems that might be encountered in practice. They are presented because they have the best sizing ratios of all of the restricted problems studied, and may offer the best hope to extend approximation results using adversary arguments or other information theoretic techniques. The problem presented in this section is called CMAX $SAT_{f(n)}$. A formal definition appears below.

DEFINITION CMAX $SAT_{f(n)}$. Let φ_1 and φ_2 be boolean formulas in CNF format where $|\varphi_2| \leq f(|\varphi_1|)$. Output the maximum number of clauses in φ_2 that can be simultaneously satisfied by any witness ω that satisfies all clauses in φ_1.

CLAIM 3.4.1 CMAX $SAT_{n^{(1-\varepsilon)}}$ requires $\frac{1-\varepsilon}{2}$ log n queries unless P=NP.

PROOF Reduce CMAX $SAT_{n^{(1-\varepsilon)}}$ to $C^2\text{-}LEX_{(1-\varepsilon)\log n}$ as follows. Let φ be an instance of $C^2\text{-}LEX_{(1-\varepsilon)\log n}$. For each distinguished variable x_i in $\{x_1,\ldots,x_{(1-\varepsilon)\log n}\}$ of φ create a group of 2^{i-1} singleton clauses consisting of a conjunction of i copies of the variable x_i. Let φ_2 be this collection of clauses, and let φ_1 be φ. We now have

$$|\varphi_2| = \sum_{i=0}^{(1-\varepsilon)\log n - 1} 2^i = n^{(1-\varepsilon)} - 1$$

Since $|\varphi_2| \leq |\varphi_1|^{1-\varepsilon}$, $|\varphi_2| \leq f(|\varphi_1|)$ is satisfied.◊

3.5 Δ-OPTIMIZATIONS

Δ-optimizations consist of two NP optimization problems Π_1 and Π_2 where Π_2 is an extension of Π_1. The objective is to compute the optimization on both problems separately, and output their difference. This has the effect of determining the degree to which the extension was able to affect the optimal value. As with constrained optimizations there is generally a relation between Π_1 and Π_2 that limits the size of Π_2 as a function of the size of Π_1.

Formally, a Δ-optimization requires that $\Delta m = \|m_1 - m_2\| = f(m_1)$. Where m_1 and m_2 measure the problem sizes. If the relation f between the problem sizes is polynomial the degree will be placed as an exponent on the Δ in the problem name. For example Δ^2-Π indicates a problem with a quadratic size disparity.

All problems in this section are reduced to a problem called C^2-$MLEX_{f(n)}$ defined below.

DEFINITION 3.5.1 C^2-$MLEX_{f(n)}$, Let φ be a boolean formula in CNF format where $|\varphi| \leq n^2$ where n is the number of variables. Output the LEX max lowest f(n) bits of any witness that satisfies c clauses where c is the most clauses that can be simultaneously satisfied by any witness.

CLAIM 3.5.2 C^2-$MLEX_{\log\sqrt{n}}$ requires $\log\sqrt{n}$ - 1 queries unless P=NP.

PROOF Recall that C^2-$LEX_{\log\sqrt{n}}$ requires $\log\sqrt{n}$ queries unless P=NP. Create OTM M_1 to solve C^2-$LEX_{\log\sqrt{n}}$ using OTM M_2 that solves C^2-$MLEX_{\log\sqrt{n}}$. M_1 will simply run M_2 and make one additional query to see of φ is satisfiable. If so it outputs the answer from M_2 , if not it outputs 0 to indicate an illegal φ. Thus M_2 was able to muster $\log\sqrt{n}$ - 1 queries.◊

Query bounds on C^2-$MLEX_{\log\sqrt{n}}$ can now be used to show query bounds for $\Delta^{1/6}$-MAX SAT defined below.

DEFINITION 3.5.3 $\Delta^{1/6}$-MAX SAT, Let φ_1 and φ_2 be boolean formulas in CNF format where φ_2 is an extension of φ_1 and, $|\varphi_2| - |\varphi_1| = \sqrt[6]{|\varphi_1|}$. Independently compute the maximum number of clauses simultaneously satisfiable in φ_1 and φ_2, and output the difference.

CLAIM 3.5.4 $\Delta^{1/6}$-MAX SAT requires $\log \sqrt[6]{|\varphi_1|} - 1$ queries unless P=NP.

PROOF Let φ' be an instance of C^2-MLEX$_{\log\sqrt{n}}$. Let φ be φ' padded with enough variables so that $|\varphi| = n^2$ exactly. Let φ_1 and φ_2 be defined as follows

$$\varphi_1 = (\varphi)^n, \quad \varphi_2 = (\varphi)^n (x_1)^1 (x_2)^2 (x_3)^4 \ldots (x_{\log\sqrt{n}})^{2^{\log\sqrt{n}-1}}$$

We now have $|\varphi_2| - |\varphi_1| = \sqrt[6]{|\varphi_1|}$, and the problem requires $\log(\sqrt[6]{|\varphi_1|})-1$ queries.◊

The methods presented above can also be applied against C^2-LEX$_{(1-\varepsilon)\log n}$ to prove $\Delta^{1/6}$-CLIQ, $\Delta^{1/6}$-MIS, $\Delta^{1/6}$-VC require $\log \sqrt[6]{|V|} - 1$ queries unless P=NP, where $|V|$ is the number of vertices in the graph.

4 APPROXIMATION BOUNDS

The error bound proofs of Section 2 can now be applied to the query bounds of Section 3 to get absolute and relative error bounds. Applying the theorems directly yields strong error bounds for 1/2-approximations on many restricted NP optimization problems. Ideally, one would like to be able to extend these error bounds to unrestricted problems. Efforts to extend these bounds under this CRP have met with mixed results. Strong 1/2-approximation error bounds have been extended for several problems, however closer inspection revealed that these bounds could be attained by other methods. Efforts to extend k-approximation bounds to problems for which there is no known bound (such as MAX CLIQUE, and MIS) have not been successful. It has, however, been possible to extend $\frac{1}{n^k}$-approximation results to these problems.

The remainder of this section is arranged as follows. First 1/2-approximation error bounds are presented by direct use of the theorems of Section 2. An example is then given to show how these error bounds might be extended to unrestricted problems. The section concludes with $\frac{1}{n^k}$-approximation results extended for MAX SAT.

4.1 1/2-APPROXIMATION RESULTS FOR RESTRICTED OPTIMIZATIONS

CLAIM 4.1.1 All bounded optimization problems Π presented in Section 3.3, have $Approx_\Pi > 3/2opt_\Pi$ for over-approximations and $Approx_\Pi < 1/2opt_\Pi$ for under-approximations unless P=NP.

PROOF All problems in Section 3.3 have query requirements that exactly match the maximum number of bits required for output. The results then follow from Corollaries 2.4 and 2.5.◊

CLAIM 4.1.2 All Δ-optimization problems Π presented in Section 3.4, have $\text{Approx}_{\Pi} > 5/4\text{opt}_{\Pi}$ for over-approximations and $\text{Approx}_{\Pi} < 3/4\text{opt}_{\Pi}$ for under-approximations unless P=NP.

PROOF All problems in Section 3.4 have query requirements that differ from the maximum number of bits required for output by an additive constant of 1. The results then follow from corollaries 2.4 and 2.5.◊

4.2 EXTENDING k-APPROXIMATION RESULTS TO UNRESTRICTED OPTIMIZATION PROBLEMS

This section presents a method for extending the error bounds from bounded optimization problems to unrestricted versions. 1/2-approximation bounds are presented for the unrestricted optimization problems KERNEL, and MINIMUM EXACT COVERING. The 1/2-approximation results on many bounded optimization problems can be extended by fairly simple constructions. Unfortunately, the extensions found to date can also be proven by other methods, and the query analysis technique has not been required. Problems for which there are no known error bound appear to be more difficult to analyze and may require more powerful techniques than are presented here. Formal definitions of the KERNEL and MINIMUM EXACT COVERING problems appear below.

DEFINITION 4.2.1 Kernel of a graph, Let G = (V,A) be a directed graph. The graph kernel is a subset of vertices V'of V such that for any vertex $u \in V-V'$

there is a vertex $v \in V$ such that the directed edge $(v,u) \in A$. Also, for all vertices $v_i, v_j \in V'$, the edges (v_i,v_j) or (v_j,v_i) do not appear in A.

DEFINITION 4.2.2 KERNEL, Let $G = (V,A)$ be a directed graph. Output the size of the largest kernel of G.

DEFINITION 4.2.3 Exact set covering, Let S be a set, and let $C=\{C_1,C_2,\ldots,C_n\}$ be a set of subsets over S. An exact covering of S is a subset $C'=\{C_a,C_b,\ldots,C_z\}$ of C such that all elements of S appear once and only once in some subset C_i contained in C'.

DEFINITION 4.2.4 MINIMUM EXACT COVER (MEC) , Let S be a set, $C=\{C_1,C_2,\ldots,C_n\}$ be a set of subsets over S. Output the size of the smallest exact covering of S.

The KERNEL, and MINIMUM EXACT COVER problems defined above have straightforward bounded versions defined below. Both of these problems can easily be proven to have no 1/2-approximations using the methods of Section 2, and 3.

DEFINITION 4.2.5 $\text{BKERNEL}_{f(|V|)}$, Let $G = (V,A)$ be a directed graph, and V' be a set of distinguished vertices such that $|V'| = f(|V|)$. Output the largest (or smallest) number of distinguished vertices in any kernel of G.

DEFINITION 4.2.6 BOUNDED OPTIMAL EXACT $\text{COVER}_{f(n)}$ ($\text{BOEC}_{f(n)}$) , Let S be a set, $C=\{C_1,C_2,\ldots,C_n\}$ be a set of subsets over S, and C' a subset of C such that $|C'| = f(|C|)$. Output the largest (or smallest) number of distinguished sets in C' in any exact covering of S.

1/2-approximation bounds can now be proven for KERNEL and MEC by extending the 1/2-approximation results for BKERNEL and BOEC. The proof works by assuming that there exists a 1/2-approximation algorithm for the unrestricted version, and then using this algorithm to build a polynomial-time 1/2-approximation algorithm for the bounded version. Since the bounded version has no such approximation unless P=NP, then the unrestricted version also has none unless P=NP. The formal proof for BKERNEL is presented below. A similar proof can be made for BOEC, and many other bounded optimization problems.

THEOREM 4.2.7 If KERNEL has a polynomial-time 1/2-approximation, then so does $\text{BKERNEL}_{f(|V|)}$ for any polynomial f.

PROOF Let G be an instance of $\text{BKERNEL}_{f(|V|)}$. Use the 1/2-approximation algorithm for KERNEL to build a polynomial-time approximation algorithm for $\text{BKERNEL}_{f(|V|)}$ as follows. Let f be of the form $\sqrt[k]{n}$ for some fixed k. Create a new graph G' from G by attaching a "plume" of $2n^{\frac{2}{k}}$ dummy vertices (where n is the number of vertices in G) to each distinguished vertex as shown in figure 4.2.1 below.

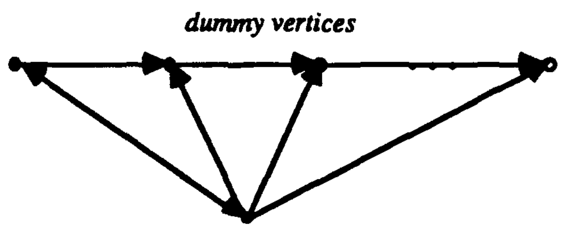

Figure 4.3.1

We now use the algorithm to find the minimal kernel of G'. Notice that if a distinguished vertex in G' is not included in the kernel, then $n^{\frac{2}{k}}$ of the dummy vertices in the plume are included. Since this is more than the total number of vertices in G, this has the effect of minimizing the number of distinguished vertices in the minimal kernel of G'. The total number of vertices γ in the minimal kernel of G' will then be $p(n^{\frac{2}{k}} + 1)+q$, where p is the minimal number of distinguished vertices from G, and q is the total number of non-distinguished vertices from G. Let ζ be a 1/2-approximation of γ, to get a 1/2-approximation of $\text{BKERNEL}_{f(|V|)}$ we divide ζ by $n^{\frac{2}{k}} + 1$. The q term drops out and we are left with a 1/2-approximation to $\text{BKERNEL}_{f(|V|)}$.◊

4.3 EXTENDING $(1/n^k)$-APPROXIMATION RESULTS TO UNRESTRICTED OPTIMIZATION PROBLEMS

The k-approximation results of Section 4.1 can be extended to yield $\frac{1}{n^k}$-approximation bounds (for some fixed k) unless P=NP. These results are weak since the error bound is a function of the problem size. This does, however, yield some useful information since there are several known algorithms that allow approximations of this form [Me84]. The proof follows lines similar to the k-approximation extensions of Section 4.2.

The $\frac{1}{n^k}$-approximation bounds can be established by one of several methods. The first method is to establish query requirements for the unrestricted problem without trying to match the solution bit requirements. Applying Theorem 2.2 or 2.3 directly can then yield the result. Using this method it is possible to show $\frac{1}{2\sqrt[3]{n^2}}$ - approximation results for the unrestricted MAX SAT problem.

The second method is to develop k-approximation bounds for the Δ-optimization version of the problem and show how to construct an approximation algorithm for the Δ-optimization from an approximation algorithm for the unrestricted optimization. The result is a proof along the lines of that presented in Theorem 4.2.6. An example of this second method will be shown below using the query requirements for $\Delta^{1/6}$-MAX SAT presented in Theorem 3.4.2. It should be noted that these weak bounds were established using query requirements for C^2-$LEX_{\log\sqrt{n}}$. Stronger results can be established when query requirements for C^2-$LEX_{(1-\varepsilon)\log n}$ are used.

THEOREM 4.3.1 MAX SAT has no polynomial-time $\frac{1}{8\sqrt{n}}$ - approximation algorithm

PROOF Recall that Theorem 4.1.2 established that $\Delta^{1/6}$-MAX SAT has no 1/4-approximation unless P=NP. We use a $\frac{1}{8\sqrt{n}}$ - approximation for unrestricted MAX SAT to construct a 1/4-approximation for $\Delta^{1/6}$-MAX SAT as follows. Approximate MAX SAT for both φ_1 and φ_2 independently. Since φ_2 is an extension of φ_1 we are assured that the difference in these two values will be within $1/4\mathrm{opt}_{\Delta^{1/6}\text{-MAX SAT}}$. Since $\Delta^{1/6}$-MAX SAT has no 1/4-approximation unless P=NP, we conclude that MAX SAT has no $\frac{1}{8\sqrt{n}}$ - approximation .◊

4.4 EXTENDING ε-APPROXIMATION RESULTS TO UNRESTRICTED OPTIMIZATION PROBLEMS

Ideally, one would like to extend true ε-approximation results to unrestricted problems. Several proofs have been constructed under this CRP similar to those for extending k-approximation results in Section 4.2. But unfortunately, all proofs constructed to date can be proven by other means. Problems such as CLIQUE, and MAXIMUM INDEPENDENT SET have no known approximation bounds, and continue to resist all but $\frac{1}{n^k}$-approximation bounds as shown in Section 4.4.

It may be possible to use some alternate technique such as adversary arguments, or some other information theoretic techniques to coaxe true k-approximate or ε-approximate bounds from these problems, but this is not known.

5 CONCLUSION

This paper presents a new method for bounding the accuracy attainable by a polynomial-time approximation to certain classes of NP-Complete optimization problems. $\frac{1}{2}$ - approximation bounds are presented for several restricted optimization problems, while $\frac{1}{n^k}$ -approximation bounds are presented for several unrestricted problems. The method is based on precisely counting the number of queries required to solve the problem on a P^{NP} oracle machine, and comparing it to the maximum number of bits that could be required to represent a solution. The approximation bounds rely on an underlying asumption that certain complexity classes do not coincide, such as P≠NP, NP≠RP, etc, so that the approximation bounds hold unless the assumption does not hold. The "strength" of the approximation results depends upon the strength of the assumption used in the construction.

In this paper all proofs are based on the assumption that P≠NP. Using this assumption several restricted NP optimization problems in graph theory and combinatorics are proven to have no polynomial-time approximations that guarantee accuracy to better than 1/2 of optimal. Due to the nature of the underlying assumption P≠NP the optimizations are generally restricted so that only some subset of the instance elements may participate in the optimization. For vertex and edge problems this subset was restricted to be at most $O(\sqrt[3]{n})$ of the total vertices in the graph. For the coloring problem this restraint was reduced to $O(\sqrt{n})$ of the total vertices in the graph. It appears that these constraints can be further loosened either by basing the proofs on a less stringent assumption such as NP≠RP, or by finding more "efficient" reductions.

Methods are also shown to get k-approximation and $\frac{1}{n^k}$-approximation results for unrestricted NP optimization problems. Unfortunately, all k-approximation results found under this CRP can be shown by other means, but the $\frac{1}{n^k}$-approximation results can be shown for problems that had no previously known error bounds. These results are weak since the error bound is a function of the problem size.

It may be possible to strengthen the error results for these unrestricted problems using adversary arguments or information theoretic techniques. This would be a significant result. Consider the Weighted Max Independent Set (WMIS) problem. It is currently known that the unconstrained optimization version of this problem either has an ε-approximation, or has no k-approximation at all, but it is an open question which one holds. Showing that uncontrained WMIS has no k-approximation below some k would establish that the related problem MAX MIS has no ε-approximation at all and hence no k-approximation. This would be a very strong result. A similar new result may be possible for the unconstrained CLIQUE problem.

BIBLIOGRAPHY

[Be88] R. Beigel. *NP-Hard Sets Are P-Superterse Unless R=NP.* Technical Report 88-04, The Johns Hopkins University, Dept. of Computer Science, 1988.

[Ga85] W. I. Gasarch, *The Complexity of Optimization Functions.* Technical Report TR-1652, The University of Maryland, Dept. of Computer Science, 1985.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* WH Freeman and Co. New York, 1979.

[GP88] W. I. Gasarch and S. R. Pearlman, *The Complexity of Optimization Problems Related to Partition.* Technical Report, The University of Maryland, Dept. of Computer Science, 1988.

[Kr86] M. W. Krentel. The Complexity of Optimization Problems. In *Proceedings of the 18^{th} annual ACM Symposium on Theory of Computing*, pages 69-76, 1986.

[Me84] K. Melhorn. *Graph Algorithms and NP-Completeness* EATCS Monographs in Theoretical Computer Science,Springer Verlag, Heidelberg, 1984.

Distribution List for IDA Paper P-2347

NAME AND ADDRESS	NUMBER OF COPIES
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
Mr. Karl H. Shingler Department of the Air Force Software Engineering Institute Joint Program Office (ESD) Carnegie Mellon University Pittsburgh, PA 15213-3890	1
IDA	
General Larry D. Welch, HQ	1
Ms. Ruth L. Greenstein, HQ	1
Mr. Philip L. Major, HQ	1
Dr. Robert E. Roberts, HQ	1
Mr. Kevin J. Rappoport, SRC	2
Ms. Anne Douville, CSED	1
Dr. Richard J. Ivanetich, CSED	1
Mr. Terry Mayfield, CSED	1
Ms. Katydean Price, CSED	2
Dr. Richard Wexelblat, CSED	1
IDA Control & Distribution Vault	2