Combinatorial Dominance Guarantees for Heuristic Algorithms (Extended Abstract)

Daniel Berend* Steven S. Skiena†

1 Introduction

The design of approximation algorithms for NP-hard problems is perhaps the most active research area in the theory of combinatorial algorithms. However, provably good approximation algorithms often perform poorly in practice compared to local search heuristics without the theoretical pedigree. Further, many fundamental optimization problems are provably hard to approximate even to within a polynomial factor [2].

In this paper, we study the notion of a *combinatorial dominance guarantee* as an alternate performance guarantee for assessing the quality of a given heuristic or approximation algorithm. We establish novel and interesting dominance guarantees even for certain inapproximable problems and heuristic search algorithms.

The intuition behind this performance measure (which is made formal in Section 2) rests on the letter of recommendation one could write on behalf of a given person, or heuristic solution. A recommendation like *"She is half as good as I am, but I am the best in the world"* captures the traditional notion of a provable approximation ratio, but does not tell us how the candidate ranks with respect to a larger field. She might be anywhere from the second best in the world to the worst around.

A recommendation like *"She is the best of the 75 students in my class this year"* is more common, and analogous to a combinatorial dominance guarantee. It certifies the candidate as provably superior to a certain number of members of a given pool, with the implied assumption that this says something meaningful about the candidate's global ranking as well. The larger the number of competitors dominated by the candidate, the stronger the recommendation.

Note that there is no a priori way to distinguish which of the letters represents the stronger recommendation. Likewise, it is not clear how combinatorial dominance compares to approximation ratio as a measure of the quality of a heuristic. We make no particular claims, but show that this measure can be easily applied to *tightly* analyze a wide variety of heuristics, including those for problems for which efficient approximation algorithms cannot exist. An outline of our main results appears in Table 1.

The notion of combinatorial dominance guarantees is by no means original to us. See Section 1.1 for a discussion of previous results. We have found, however, that this work is relatively unknown within the algorithmic research community. Our primary contribution is to significantly extend the range of problems for which dominance analysis can be applied to, and to introduce new general techniques which apply to a wide range of problems and heuristics. In this paper:

*Depts. of Math. and Computer Science, Ben-Gurion Univ., Beer-Sheva 84105 Israel, berend@cs.bgu.ac.il

†Corresponding author. Department of Computer Science, SUNY Stony Brook, NY 11794-4400, skiena@cs.sunysb.edu

Problem	Heuristic	Combinatorial Dominance Bounds $f(n)$ "good"	$b(n)$ "bad"
Monotonic Subset	incremental insertion	$2^{\lceil \frac{n}{2} \rceil} + 2^{\lfloor \frac{n}{2} \rfloor} - 1$	$2^n - f(n)$
Clique/Ind. Set	incremental insertion	$2^{n-1} + n$	$2^{n-1} - n$
Vertex Cover	vertex insertion	$2^{n-1} + n$	$2^{n-1} - n$
Vertex Cover	edge insertion	$2^n - (1.839 + o(1))^n$	$(1.839 + o(1))^n$
Set Cover	incremental insertion	$\Theta\left(n^{-0.1807} 1.7087^n\right)$	$2^n - f(n)$
Set Cover	greedy insertion	$\Theta\left(n^{-0.1807} 1.7087^n\right)$	$\Omega\left(\frac{1}{\sqrt{n}}\left(\frac{4}{\sqrt[3]{n}}\right)^n\right)$
Knapsack	incremental insertion	$2^{n-1} + 1$	$2^{n-1} - 1$
Knapsack	decreasing insertion	$2^{n-1} + 1$	$3 \cdot 2^{n-4}$
Knapsack	local exchange	$2^{n-1} + n$	$2^{n-1} - n$
Knapsack	$(1+\varepsilon)$-PTAS	$2^{n/2}$	$2^{n-1} - 1$
Knapsack	$(l+\varepsilon)$-PTAS	$2^{n/2}$	$2^n - 2^{n-l} - l$
Max-2Sat	majority vote	1	$2^n - 1$
Max-2Sat	step-by-step vote	2	$2^n - n - 1$
Max-2Sat	local exchange	$n + 1$	$2^n - n - 1$
Binary Optimization	local exchange	$n + 1$	$2^n - n - 1$

Table 1: Summary of our results on combinatorial dominance bounds for common heuristics.

- We prove *tight* or *almost tight* bounds for heuristics on several fundamental optimization problems over subsets, thus significantly extending the range of the theory. This enables us to analyze and compare greedy heuristics for problems such as clique and independent set, which are doomed to being inapproximable to a polynomial factor [2], and requires us to extend notions of combinatorial dominance to constrained problems where (unlike the previously studied TSP) not all permutations, subsets, or set partitions form legal solutions.

- We prove an exponential dominance bound for the incremental insertion heuristic for the general class of *monotonically-constrained* subset problems, which includes clique, independent set, vertex cover, and knapsack. Further, we prove that this is the best possible result over problems in this class.

- Incremental insertion can perform much better for specific monotonically-constrained subset problems, however. We prove *tight* bounds on incremental insertion for clique/independent set, set cover, and knapsack. Our analysis yields bounds which vary substantially according to problem.

- We establish certain general results relating approximation ratio and combinatorial dominance guarantees, particularly for optimization problems over subsets. We also prove certain limitations on the combinatorial dominance guarantees of polynomial-time approximation schemes (PTAS).

- We demonstrate a general technique to award combinatorial dominance "certificates" for arbitrary solutions of typical optimization problems, and apply this technique to the traveling salesman and maximum satisfiability problems in the full paper. Similar approximation ratio certificates are not forthcoming for ad hoc solutions, short of explicit comparison against solutions from previously analyzed heuristics.

This certificate technique provides some initial ideas on how to meaningfully extend dominance theory to randomized algorithms. Randomized heuristics offer high *expected* dominance guarantees, since picking a solution uniformly at random from a search space of size $|S_P(n)|$ yields an expected $|S_P(n)|/2$ combinatorial dominance bound. However, its worst-case dominance bound is 1. Such expected dominance bounds are completely incomparible to the *worst-case* guarantees provided in this paper.

This paper is organized as follows. In Section 1.1, we briefly survey previous work. In Section 2, we formalize our notions of combinatorial dominance guarantees. In Section 3, we prove tight bounds on incremental insertion heuristics over the class of monotonically constrained subset problems. In Section 4, we analyze several heuristics for vertex/set cover, clique, knapsack, and maximum satifiability.

Proofs and descriptions of several results have been omitted for space reasons, but will appear in the full paper. A host of open problems remain.

1.1 Previous Work

Combinatorial dominance guarantees have been studied primarily within the operations research community. The basic notion appears to have been independently discovered several times, most recently by us. The primary focus has been on algorithms for TSP, specifically designing polynomial-time algorithms which dominate exponentially large neighborhoods. The first TSP heuristics with exponential domination number are due to Sarvanov and Doroshko [12, 13].

The question of whether a polynomial-time algorithm can dominate $(n-1)!/p(n)$ tours, where $p(n)$ is polynomial, appears to have first been raised by Glover and Punnen [4]. Dominance bounds for TSP have been most aggressively pursued by Gutin, Yeo, and Zverovich in a series of papers (including [5, 6]) culminating in a polynomial-time algorithm which dominates $\Theta((n-1)!)$ tours. These bounds follow by applying certain Hamiltonian cycle decomposition theorems to the complete graph. Interested readers should consult their excellent survey [7].

Deineko and Woeginger [3] survey the complexity of optimizing TSPs over several well-defined but exponentially-large neighborhoods. Such optima by definition have large domination numbers. Balas and Simonetti [1] perform an experimental study of certain linear-time dynamic programming algorithms for TSP which dominate exponentially many solutions.

Previous work on dominance analysis of conventional heuristics includes [8, 11], which proves that the nearest neighbor, minimum spanning tree, and greedy heuristics perform extremely poorly for symmetric and asymmetric TSPs. We [10] have recently developed a model for analyzing heuristic search algorithms (such as simulated annealing and backtracking) based on the ideas of combinatorial dominance.

2 Definitions

Consider a given combinatorial optimization problem P, represented by a solution space $S_P(n)$ and objective function $C_P(I,x)$. The *solution space* $S_P(n)$ is the set of all combinatorial objects representing possible solutions x to P, where $|x| = n$. Each optimization problem is characterized by an *objective function* $C_P(I,x)$ defined on all elements $x \in S_P(n)$ for a given instance I, where $n = |x|$. If P is a maximization (minimization, resp.) problem, we seek the $x_0 \in S_P(n)$ such that $C_P(I,x_0) \geq C_P(I,x)$ ($C_P(I,x_0) \leq C_P(I,x)$, resp.) for all $x \in S_P(n)$.

A *heuristic* $H_P(I)$ for problem P is a procedure which selects an element $x \in S_P(n)$ for every instance I, where n is a function of $|I|$. A heuristic $H_P(I)$ offers an $F(n)$ *combinatorial dominance* *guarantee* for problem P if for all instances I the solution $H_P(I)$ is provably at least as good as $F(n)$ elements of $S_P(n)$. In this paper, we let $f(n)$ denote our best lower bound on the existential function $F(n)$.

The converse of a combinatorial dominance guarantee for a heuristic is its *blackball bound*[1] (or negative domination number), a worst-case measure of the number of superior solutions which may be missed by the heuristic. A specific instance I' yields a blackball bound of $b(n)$ to a heuristic $H_P(I)$ for problem P if the solution $H_P(I')$ is provably worse than at least $b(n)$ elements of $S_P(n)$. By definition, there must exist an instance I' such that the number of solutions superior to $H_P(I')$ is $|S_P(n)| - F(n)$. Our bounds for a given heuristic are *tight* if $f(n) + b(n) = |S_P(n)|$ and *near tight* if $f(n) = \Omega(|S_P(n)| - b(n))$ and $b(n) = \Omega(|S_P(n)| - f(n))$.

3 Guarantees for General Subset Problems

Many important NP-complete optimization problems are defined over subsets. The hardness of such problems results from imposing constraints which render certain elements of the search space to be infeasible solutions. These optimization problems then seek the maximum (or minimum) feasible solution under a given cost function.

A solution x *positively dominates* solution y for a maximization (minimization, resp.) problem f if x and y are both feasible solutions and $C_p(I, x) \geq C_p(I, y)$ ($C_p(I, x) \leq C_p(I, y)$, resp.). Throughout this paper, we adhere to the convention that any feasible solution dominates *every* infeasible solution with respect to combinatorial dominance guarantees, and that infeasible solutions are ranked in an arbitrary order (say lexicographic). Thus the dominance number x of any feasible solution is the number of positively dominated solutions plus the number of infeasible solutions. This convention has advantages with respect to ease of analysis over positive dominance, and leads to several interesting results.

A maximization (minimization, resp.) problem P is *monotonically-constrained* if (1) for any feasible solution X of P and $X' \subset X$ ($X \subset X'$, resp.), the solution X' is a feasible solution inferior to X, and (2) for any infeasible solution X of P and $X \subset X'$ ($X' \subset X$, resp.), the solution X' is also infeasible.

Incremental insertion provides a significant combinatorial dominance guarantee for any monotonically constrained problem. The incremental insertion heuristic starts with an arbitrary permutation p of the elements of the solutions space U (i.e. the vertices of a graph for independent set or numerical element weights for knapsack). The first element of p is selected to be in the solution subset S. We then repeatedly consider the ith element of p, adding it to S iff the resulting subset remains a feasible solution.

Theorem 1 *Let P be a monotonically-constrained optimization problem. Then the incremental insertion heuristic yields a combinatorial dominance guarantee of $f(n) = 2^{\lceil n/2 \rceil} + 2^{\lfloor n/2 \rfloor} - 1$ for any instance of P. Further, there exist monotonically-constrained problems for which this is the best possible result.*

Proof: Let S be the solution provided by the algorithm. Every proper subset of S must represent an inferior feasible solution to P, since P is a monotonically-constrained problem. Further, for every non-empty subset of $X = U - S$, the set $S \cup X$ must be infeasible, or else the elements of X would have also been selected by the incremental insertion heuristic. Thus S dominates its 2^k

[1] A blackball letter is designed to hurt instead of help: *"I personally know at least 100 people who are better candidates for our club than that bozo"*. The larger the bound, the more damning the implied message.

subsets and and all its 2^{n-k} supersets, where $k = |S|$. Combining the two collections, and noting that S itself belongs to both, we find that S dominates at least $2^k + 2^{n-k} - 1$ solutions. This expression is minimal for $k = \lceil n/2 \rceil$ (as well as $k = \lfloor n/2 \rfloor$), yielding the result.

This bound is the tightest possible over the general class of problems. Consider the problem of maximizing a function $g(A)$ defined for each subset A of some fixed set of n elements. For the problem to be monotonically-constrained it suffices to require that $g(A) \leq g(B)$ if A and B are feasible solutions with $A \subseteq B$. Now consider the following example. Sets strictly containing $\{1, 2, \ldots, \lfloor n/2 \rfloor\}$ are infeasible, and for other subsets of $\{1, 2, \ldots, n\}$:

$$g(A) = \begin{cases} |A|, & A \subseteq \{1, 2, \ldots, \lfloor n/2 \rfloor\}, \\ n, & \text{otherwise.} \end{cases}$$

Incremental insertion will yield the solution $\{1, 2, \ldots, \lfloor n/2 \rfloor\}$, which dominates exactly $2^{\lceil n/2 \rceil} + 2^{\lfloor n/2 \rfloor} - 1$ solutions. ∎

Corollary 1 *The incremental insertion heuristic yields a $\Omega(2^{n/2})$ combinatorial dominance guarantee for clique, independent set, set and vertex cover, and knapsack.*

In subsequent sections, we will give improved dominance bounds for all of these problems, using alternate heuristics and/or problem-specific analysis.

We can also give general bounds on dominance number as a function of the approximation ratio of certain heuristics:

Theorem 2 *Let $H_P(n)$ be a factor k-approximation ($1/k$-approximation) algorithm for unweighted minimization (maximization) problem P. Then any solution generated by $H_P(n)$ has a combinatorial dominance guarantee of $\binom{n}{n/(k+1)} = \Omega\left(\frac{1}{\sqrt{n}}\left(\frac{k+1}{k^{k/(k+1)}}\right)^n\right)$. Further, this is the best such result possible.*

Proof: We prove this for minimization problems, but the proof is analogous for maximization. Let x be the size of the solution returned by $H_P(n)$. Because of the approximation ratio, we are guaranteed that the optimal solution is of size at least x/k and at most x. Thus any subset of size $< x/k$ is infeasible, while anything larger than x is inferior.

Thus the heuristic solution dominates any subset outside this range, and $f(n) \geq \binom{n}{x} + \binom{n}{x/k}$. This quantity is minimized when both are (approximaely) equal, i.e. when $x \approx kn/(k+1)$. Then, by Stirling's formula:

$$f(n) \geq \binom{n}{n/(k+1)} \asymp \frac{1}{\sqrt{n}}\left(\frac{k+1}{k^{k/(k+1)}}\right)^n .$$

To prove that this is the best result possible, we define an objective function assigning each set its size, except that all sets of size $> kn/(k+1)$ are infeasible. A solution guaranteed to have an approximation ratio of k may well choose a set of size $n/(k+1)$, which will dominate (up to a constant factor) exactly the number of solutions asserted by the theorem. ∎

4 Problem-Specific Analysis

In this section, we analyze several heuristics for vertex/set cover, clique, knapsack, and maximum satifiability. Due to lack of space, we omit the proofs except for the cover problems. The breadth of the problems and heuristics studied suggests the potential range and power of dominance analysis.

4.1 Vertex and Set Cover

The vertex cover problem seeks the smallest subset $S \subset V$ in a given graph $G = (V, E)$ such that each $e \in E$ has at least one edgepoint in S. The incremental vertex insertion heuristic for vertex cover yields a combinatorial dominance guarantee of $2^{n/2}$ by Corollary 1. However, there are a variety of vertex cover heuristics known which offer 2-approximation factors, including incremental edge insertion and matching. By Theorem 2, these heuristics lead to improved combinatorial dominance guarantees:

Corollary 2 *All 2-factor approximations for vertex cover yield a combinatorial dominance guarantee* $f(n) \approx (3/2^{2/3})^n/\sqrt{n} \approx 1.88988^n/\sqrt{n}$

Still, tighter bounds can follow from more specified analysis. A well-known factor-2 approximation for vertex cover is the maximum matching heuristic, which takes both endpoints of all edges in any maximum matching. This heuristic offers an excellent dominance bound:

Theorem 3 *For the maximum matching (edge insertion) heuristic for vertex-cover,* $f(n) = 2^n - b(n)$ *and* $b(n) = (1.839 + o(1))^n$.

Proof: Suppose we are given a maximal matching consisting, say, of the edges of the edges v_{2i-1}, v_{2i} for $1 \le i \le l$. The cover provided by the algorithm is $\{v_1, v_2, \ldots, v_{2l}\}$. Deleting all edges except for those of the selected matching, we may only increase the number of feasible solutions, thereby increasing the number of solutions better than the given solution. Hence we may assume to begin with that the set of edges of the graph consists only of the edges $(v_1, v_2), (v_3, v_4), \ldots, (v_{2l-1}, v_{2l})$.

The better solutions are all sets of vertices of size not exceeding $2l - 1$, containing at least one out of each pair of vertices v_{2k-1} and v_{2k}. In other words, a better solution is obtained by selecting first j vertices out of $v_{2l+1}, \ldots, v_n$, then exactly one out of each of $k > j$ of the l pairs v_{2i-1} and v_{2i}, $1 \le i \le l$ and both elements of all other $l - k$ pairs. Hence:

$$b(n) = \max_{l \le n/2} \sum_{j=0}^{n-2l} \left(\binom{n-2l}{j} \sum_{k=j+1}^{l} \binom{l}{k} 2^k \right) .$$

We need to find the l for which the right hand side is maximal. Since the total number of terms in the double sum is $O(n^2)$, the main thing is to find for which l the largest term in the sum is as large as possible. At this stage, we care only to find the largest c such the sum in question may be approximately c^n (and thus ignore factors of polynomial size). Note first that the first factor $\binom{n-2l}{j}$ is maximal for $j = n/2 - l$. The factor $\binom{l}{k} 2^k$ is maximal for $k = \left\lceil \frac{2l-1}{3} \right\rceil$. For that value of k this factor is approximately (up to an $O(l)$ factor) 3^l. We distinguish between two cases.

Case I: $\frac{n}{2} - l + 1 \le \left\lceil \frac{2l-1}{3} \right\rceil$.

In this case, the maximal term is is obtained for $j = n/2 - l$ and $k = \left\lceil \frac{2l-1}{3} \right\rceil$. Its value, up to a polynomial factor, is

$$\binom{n-2l}{n/2-l} \binom{l}{\lceil (2l-1)/3 \rceil} 2^{\lceil (2l-1)/3 \rceil} \approx 2^{n-2l} 3^l .$$

The right hand side decreases with l, so that the expression is maximal for the smallest possible l, namely (up to an additive constant) $l \approx 3n/10$. The maximal term is then approximately

$$2^{n-2 \cdot 3n/10} 3^{3n/10} = 1.835n .$$

Case II: $\frac{n}{2} - l + 1 > \left\lceil \frac{2l-1}{3} \right\rceil$.

As j grows from 0 up to $\left\lceil \frac{2l-1}{3} \right\rceil$, the factor $\binom{n-2l}{j}$ grows, and for $\binom{l}{k}2^k$ the maximum is approximately 3^l. As j continues to grow up to $n/2 - l$, the first term continues to grow, but the second, which is just $\binom{l}{j+1}2^{j+1}$, decreases. From that point on, both terms decrease. Thus, the maximal term is obtained for some j between $\left\lceil \frac{2l-1}{3} \right\rceil$ and $n/2 - l$. Suppose the optimal choices are $l \approx \alpha n$ and $j \approx \beta n$. Employing Stirling's formula we obtain

$$\binom{n-2l}{j}\binom{l}{j+1}2^{j+1} \approx \left(\frac{(1-2\alpha)^{1-2\alpha}}{\beta^\beta(1-2\alpha-\beta)^{1-2\alpha-\beta}} \cdot \frac{\alpha^\alpha}{\beta^\beta(\alpha-\beta)^{\alpha-\beta}} 2^\beta \right)^n . \tag{1}$$

Thus, we have to find the value $(\alpha, \beta) = (\alpha_0, \beta_0)$ maximizing the function of α and β on the right hand side of (1). Differentiating with respect to each variable we arrive at the equations

$$\beta = 1 - \alpha - \sqrt{1 - 4\alpha + 5\alpha^2}$$

and

$$\beta = 6 - 8\alpha - \frac{1}{\alpha} .$$

Routine calculations show that α_0 is a root of the quartic $44\alpha^4 - 66\alpha^3 + 38\alpha^2 - 10\alpha + 1$, so that $\alpha_0 \approx 0.2822$, and the corresponding value of β is $\beta_0 = 0.1988$. The value of the maximum of the function of α and β on the right hand side of (1) is therefore 1.839. Hence

$$b(n) = (1.839 + o(1))^n$$

and

$$f(n) = 2^n - (1.839 + o(1))^n .$$

∎

The previous proof uses the following general technique. To determine a lower bound for $f(n)$ we assume a general problem instance. We then tranform this instance into an instance of a special subclass, for which the heuristic provably works no better than for the original problem. Hence any lower bound for $f(n)$ which holds for the restricted family must hold for general instances. We will apply this technique to other problems and heuristics.

Theorem 4 *For the vertex insertion heuristic of the vertex-cover problem we have $f(n) = 2^{n-1} + n$ and $b(n) = 2^{n-1} - n$.*

Proof: Let $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$ and $E = \{(1, j) : 2 \le j \le n\}$. Going over the vertices from 1 to n, we see that the resulting cover will be $\{2, \ldots, n\}$. This solution dominates all 2^{n-1} subsets thereof, as well as all n sets of size at least $n - 1$ containing vertex 1, and is dominated by all other sets. Consequently, $b(n) \ge 2^{n-1} - n$.

Now let $G = (V, E)$ be any graph with $V = \{1, 2, \ldots, n\}$ and S any solution obtained by the heuristic. Since S is a cover, no two vertices in S^C can be neighbors. Each vertex in S is connected by an edge to at least one vertex in S^C. For each vertex in S, delete all edges connecting it to other vertices, except for one of the edges connecting it to a vertex in S^C. The set S can be obtained by our algorithm applied to the new graph as well. Since the change turns sets which are not feasible solutions in the old graph to feasible solutions for the new graph, the number of solutions dominated by S can only decrease by passing from the old graph to the new one. Hence, to prove that $f(n) \geq 2^{n-1} + n$, we may assume to begin with that each vertex in S has a single neighbor (which necessarily belongs to S^C).

Suppose, say, that $S = \{l+1, l+2, \ldots, n\}$, that each of the vertices $1 \leq i \leq k$ in S^C neighbors all the vertices in the (non-empty) set $M_i \subseteq S$, where $k \leq l$, and the vertices $k+1, \ldots, l$ are not connected by an edge to any other vertex. It will be convenient to put $M_i = \emptyset$ for $k+1 \leq i \leq l$. Clearly, S is a disjoint union of the sets M_i. A set S' is a feasible solution if and only if it is of the form $\cup_{i=1}^{l} A_i$, where $A_i \subseteq \{i\} \cup M_i$ for $1 \leq i \leq l$, and for each $1 \leq i \leq k$ we have either $i \in A_i$ or $A_i = M_i$. Such a solution S' is better than S if $|S'| < n - l = |S|$.

Let G_1 be the graph obtained from G by replacing each of the edges (k, v), $v \in M_i$, by the edge $(k-1, v)$. S is obtained by our algorithm applied to G_1 as well. To each feasible solution S' of the problem in G correspond a feasible solution S_1' of the problem in G_1 as follows: If $k-1 \in A_{k-1}$ or $A_k \supseteq M_k$, then $S_1' = S'$; otherwise $S_1' = S' \cup \{k-1\} - \{k\}$. It is easy to check that this yields a 1-1 correspondence from the set of all solutions of the problem for G which are better than S into the set of all solutions of the problem for G_1 which are better than S. Hence the number of solutions dominated by S in G_1 does not exceed the number of those dominated by S in G. Iterating this process, we may assume without loss of generality that $k = 1$.

Let us count the number of solutions dominated by S. First we have all infeasible solutions, numbering $2^{l-1}\left(2^{n-l} - 1\right)$. Next we have all feasible solutions not containing vertex 1, numbering 2^{l-1}. Finally, we have all solutions of size $n - l$ or more, containing vertex 1, whose number is $\sum_{j=n-l-1}^{n-1} \binom{n-1}{j}$. Altogether, the number of solutions dominated by S is

$$2^{l-1}\left(2^{n-l} - 1\right) + 2^{l-1} + \sum_{j=n-l-1}^{n-1} \binom{n-1}{j} \geq 2^{n-1} + \sum_{j=n-2}^{n-1} \binom{n-1}{j} = 2^{n-1} + n.$$

Thus $f(n) \geq 2^{n-1} + n$, which completes the proof. ▮

Different bounds result when applying incremental insertion to the more general set cover problem:

Theorem 5 *For the incremental insertion heuristic of the set cover problem we have*

$$f(n) = \max_{1 \leq l \leq n} \left(\sum_{j=0}^{l-1} \binom{n}{j} - 2^{l-1} \right) \approx n^{-0.1807} 1.7087^n$$

and $b(n) = 2^n - f(n)$.

A seemingly better algorithm for set-cover is obtained by taking each time a set with a maximal number of elements not belonging to any of the sets taken hitherto.

Theorem 6 *For the greedy insertion heuristic of the set-cover problem we have* $b(n) = \Omega\left(\frac{1}{\sqrt{n}}\left(\frac{4}{\sqrt[3]{n}}\right)^n\right)$.

4.2 Clique / Independent Set

Despite the close relationship between vertex cover and clique, the proof of our tighter bounds for incremental insertion on vertex cover does not appear to carry over to clique. Through different logic, we obtain a surprisingly similar result. The proof will appear in the full paper.

Theorem 7 *For the incremental insertion heuristic of the clique problem we have* $f(n) = 2^{n-1} + n$ *and* $b(n) = 2^{n-1} - n$.

4.3 The Knapsack Problem

In the knapsack problem, we are given a set of integers $S = \{s_1, \ldots, s_n\}$ and a capacity T. We seek the subset $S' \subset S$ that maximizes $\sum_{s \in S'} s$ subject to the constraint $\sum_{s \in S'} s \leq T$. In this section, we analyze three classes of heuristics for the knapsack problem.

4.3.1 Incremental Insertion

The incremental insertion heuristic has been well studied for bin packing-type problems, and yields different approximation ratios for different insertion orders. We can analyze these insertion orders for dominance guarantees as well. For arbitrary insertion orders, we get:

Theorem 8 *For the incremental insertion heuristic of the knapsack problem we have* $f(n) = 2^{n-1} + 1$ *and* $b(n) = 2^{n-1} - 1$.

Now suppose we insert the elements in knapsack according to size. If the items are ordered from the smallest to the biggest, then there is no gain. In fact, the construction of an example yielding the blackball bound in Theorem 8 is where we start with the items with least weights. The next theorem addresses the greedy algorithm – taking at each stage the heaviest possible item.

Theorem 9 *For the decreasing insertion heuristic of the knapsack problem we have* $b(n) \geq 3 \cdot 2^{n-4}$.

4.3.2 Local Improvement Heuristics

Now consider the following local improvement heuristic. Both *insert* and *exchange* operations are permitted. Insert operations add a new element x to the solution S' , provided the capacity T of the knapsack is not exceeded. Exchange operations replace an element $x \in S'$ with an element $y \in S - S'$, provided the capacity is not exceeded and $y > x$. The later requirement ensures that we increase the value of S' with every operation. The algorithm terminates when the solution is locally optimal, i.e. none of the $|S - S'|$ insert operations or $|S - S'| \cdot |S'|$ exchange operations are both feasible and improving.

Adding exchange to incremental insertion does not significantly increase the dominance number:

Theorem 10 *For the local exchange heuristic of the knapsack problem we have* $f(n) = 2^{n-1} + n$ *and* $b(n) = 2^{n-1} - n$.

4.3.3 Scaling Heuristics

Knapsack was one of the first problems for which a PTAS was discovered [9], meaning that it is possible to approximate the optimal solution to within any factor $\epsilon \geq 1$ in time polynomial in n and $1/(\epsilon - 1)$. The PTAS bounds result from scaling the elements, grouping them into classes, and optimally packing the classes.

We show that any factor-c approximation yields a respectable combinatorial dominance guarantee. Interestingly, however, this bound does not approach 2^n even as $c \to 1$.

Theorem 11 *Let S be an instance of the knapsack problem, and $H(S)$ be a heuristic solution to S whose weight is at least $OPT(S)/c$, where $OPT(S)$ is the weight of the optimal solution to S. Then $H(S)$ achieves a combinatorial dominance guarantee of at least* $2^{n/(c+1)} - 1$.

Theorem 12 *For every $c > 1$ there exists a knapsack problem S and a solution thereof S_1 of total weight $W(S_1) > OPT(S)/c$ dominated by $2^{n-1} - 1$ solutions.*

Theorem 13 *For every positive integer l and $c > l$ there exists a knapsack problem S and a solution thereof S_1 of total weight $W(S_1) > OPT(S)/c$ dominated by $2^n - 2^{n-l} - l$ solutions.*

4.4 Maximum Satisfiability

Maximum satisfiability is particularly interesting as an example of a subset problem which is not monotonically constrained. Still, we can prove several interesting results about heuristics for it.

A natural class of heuristics for Max-Sat are based on analyzing the frequency of literals appearing in the input clauses. Namely, we give x_1 the value which will make the maximal number of clauses *true*. Suppose, say, we have chosen $x_1 = true$. In subsequent stages, we disregard all clauses containing x_1, which are already necessarily *true* by our choice, and omit $\overline{x}_1$ from all clauses containing it. We continue in the same manner.

Theorem 14 *For the step-by-step majority vote Max k-Sat heuristic, $f(n) \geq 2$ and $b(n) \geq 2^n - n - 1$ for all $n \geq 2$, even for $k = 2$.*

A general heuristic one may consider is that of local change. Namely, start from any initial solution $(x_{10}, x_{20}, \ldots, x_{n0})$. If it is possible to switch the value of one of the variables x_i and thereby (strictly) improve the value of g, we make this change. We continue in this way until no change in a single variable improves the current value of g. Such a point is a *local optimum*. Obviously, a local optimum dominates at least $n + 1$ solutions – the local optimum itself and all n solutions obtained from it by switching the value of a single variable. For a general optimization problem, defined in terms of an arbitrary function g of n binary variables, this is the best possible result. This lower bound for $f(n)$ holds even in the more restricted case of Max 2-Sat:

Theorem 15 *For the local exchange heuristic of Max k-Sat we have $f(n) = n + 1$ and $b(n) = 2^n - n - 1$, even for $k = 2$.*

5 Certified Dominance Bounds for Arbitrary Solutions

We demonstrate a general technique to award combinatorial dominance "certificates" for arbitrary solutions of typical optimization problems, and apply this technique to the traveling salesman and maximum satisfiability problems. Similar approximation ratio certificates are not forthcoming for ad hoc solutions, short of explicit comparison against solutions from previously analyzed heuristics.

Such a certificate is a proof that a given solution of some optimization problem is better than at least some prescribed number of solutions of that problem. The idea is as follows. Calculate the expected value E of the objective function for a random solution, and the variance V of the same quantity. Put $\sigma = \sqrt{V}$. Now suppose you have any feasible solution. Let v_0 be the value of our objective function for this solution. If $|v_0 - E| > \sigma$, then we can assert that there is some percentage of the solutions which are worse than our solution. Indeed, by Chebyshev's inequality:

$$P(v \text{ is better than } v_0) \leq P(|v - E| > |v_0 - E|) \leq \frac{V}{|v_0 - E|^2} < 1\,.$$

In the full paper, we demonstrate how to calculate the relevant quantites to certify TSP and Max-SAT solutions as examples of our technique.

Acknowledgments

We thank Michael Bender, Gregory Gutin, David Johnson, Matya Katz, and Saurabh Sethia for helpful discussions.

References

[1] E. Balas and N. Simonetti. Linear time dynamic programming algorithms for some new classes of restricted TSPs: a computational study. *INFORMS J. Computing*, 13:56–75, 2001.

[2] M Bellare, O. Goldreich, and M. Sudan. Free bits, PCPs, and non-approximability – towards tight results. In *Proc. IEEE 36th Symp. Foundations of Computer Science*, pages 422–431, 1995.

[3] V. Deineko and G. Woeginger. A study of exponential neighborhoods for the traveling salesman problem and the quadratic assignment problem. *Mathematical Programming, Ser. A*, 87:519–542, 2000.

[4] F. Glover and A. Punnen. The travelling salesman problem: new solvable cases and linkages with the development of new approximation algorithms. *J. Operations Research Society*, 48:502–510, 1997.

[5] G. Gutin and A. Yeo. Polynomial approximation algorithms for the TSP and the QAP with factorial domination number. *Discrete Applied Mathematics*, 2001.

[6] G. Gutin and A. Yeo. TSP tour dimination and hamilton cycle decomposition of regular digraphs. *Operation Research Letters*, 2001.

[7] G. Gutin, A. Yeo, and A. Zverivich. Polynominal restriction approach for the atsp and stsp. In *The Traveling Salesman Problem*, to appear.

[8] G. Gutin, A. Yeo, and A. Zverovich. Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the tsp.

[9] O. Ibarra and C. Kim. Fast approximation algorithms for knapsack and sum of subset problems. *J. ACM*, 22:463–468, 1975.

[10] V. Phan, S. Skiena, and P. Sumazin. A model for analyzing black-box optimization. submitted for publication, 2002.

[11] A. Punnen, F. Margot, and S. Kabadi. Tsp heuristics: Domination analysis and complexity. Research report 2001-06, Dept. of Mathematics, Univ. of Kentucky, March 2001.

[12] V. Saravanov and N. Doroshko. The approximate solution of the traveling salesman problem by a local algorithm that searches neighborhoods of exponential cardinality in quadratic time. *Software: Algorithms and Programs (in Russian)*, 31:8–11, 1981.

[13] V. Saravanov and N. Doroshko. The approximate solution of the traveling salesman problem by a local algorithm that searches neighborhoods of factorial cardinality in cubic time. *Software: Algorithms and Programs (in Russian)*, 31:11–13, 1981.