DIMACS Technical Report 95-36
August 1995

Nonoverlapping Local Alignments (Weighted Independent Sets of Axis Parallel Rectangles)

by

Vineet Bafna[1]	Babu Narayanan [2]	R. Ravi[3]
DIMACS Center	DIMACS Center	DIMACS
Rutgers University	Rutgers University	Princeton University
Piscataway, NJ08855	Piscataway, NJ08854	NJ08544.

[1](*bafna@dimacs.rutgers.edu*) Supported by a DIMACS postdoctoral fellowship under Special Year National Science Foundation grant BIR-9412594
[2](*bon@dimacs.rutgers.edu*)Supported by a DIMACS postdoctoral fellowship under grants XX STC-88-09648 and 91-19999
[3](*ravi@cs.princeton.edu*) Supported by DIMACS psotdoctoral fellowship under Special Year National Science Foundation grant BIR-9412594

DIMACS is a cooperative project of Rutgers University, Princeton University, AT&T Bell Laboratories and Bellcore.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

We consider the following problem motivated by an application in computational molecular biology. We are given a set of weighted axis-parallel rectangles such that for any pair of rectangles and either axis, the projection of one rectangle does not enclose that of the other. Define a pair to be independent if their projections in both axes are disjoint. The problem is to find a maximum-weight independent subset of rectangles.

We show that the problem is NP-hard even in the uniform case when all the weights are the same. We analyze the performance of a natural local-improvement heuristic for the general problem and prove a performance ratio of 3.25. We extend the heuristic to the problem of finding a maximum-weight independent set in $(d+1)$-claw free graphs, and show a tight performance ratio of $d-1+\frac{1}{d}$. A performance ratio of $\frac{d}{2}$ was known for the heuristic when applied to the uniform case. Our contributions are proving the hardness of the problem and providing a tight analysis of the local-improvement algorithm for the general weighted case.

1 Introduction

Let S be a set of axis-parallel rectangles, such that for any pair $a, b \in S$ of rectangles, the interval defined by projecting a on an axis does not include the interval defined by projecting b on the same axis. If the two intervals intersect, then we say that a and b are *conflicting.* A set S of rectangles is *independent* if no pair of rectangles in S is conflicting.

Definition 1 Independent subset of rectangles (IR): *Given a set S of axis-parallel rectangles and an integer k, does there exist a subset $S' \subseteq S$, such that S' is independent and $|S'| > k$.*

The extension of the problem when the rectangles are weighted is immediate. This problem is motivated by an application in molecular biology in which rectangles correspond to regions of high local similarity, and the problem is to find a large number of such regions that are independent.

Theorem 1 *IR is NP-complete.*

We construct a *conflict graph* from the given set of rectangles. Each node in the graph corresponds to a rectangle in the set and every two conflicting rectangles have an edge between them in the conflict graph. The IR problem can be phrased as the maximum independent set problem for the conflict graph. While the maximum independent set problem in arbitrary graphs is well known to be notoriously hard to approximate [ALM$^+$92], we use the structure of the graphs arising from our problem to provide good approximation algorithms. Define a d-claw as the graph $K_{1,d}$, i.e., a star with d leaves. A graph is d-clawfree if has no induced d-claw. A key property that we use in devising our approximation algorithms and analyzing them is that a conflict graph of non-overlapping axis-parallel rectangles is 5-clawfree. A simple consequence of 5-clawfree property of the conflict graph is that a greedy algorithm that picks a node of maximum weight to add to the solution and continues by deleting the picked node and its neighborhood has a performance ratio of 4.

We consider a simple local improvement heuristic, t-opt, for the problem parameterized by the size, t, of the improvement. We shall describe it informally here for the unweighted problem. Begin with an arbitrary maximal independent set I in the graph. If there is an improvement that involves swapping at most t nodes into I, then we perform such an improvement. In other words, if there is an independent set A of at most t nodes in $V - I$ whose neighborhood in I has size less than that of A, then this set may be added and its neighborhood deleted from I. This results in a net increase in the size of I. The local improvement algorithm performs such t-improvements as long as they are available. It is not hard to argue that this algorithm runs in polynomial time for any fixed t. Halldórsson [Hal95] has shown that the t-opt heuristic when applied to a $d+1$-clawfree graph achieves a performance ratio of $\frac{d}{2} + \epsilon$ for any fixed $\epsilon > 0$ and in fact ϵ decreases exponentially in t. In Section 4, we provide a simple construction that shows that the performance ratio of $\frac{d}{2}$ is the best possible for the heuristic.

The local improvement heuristic can be extended in a natural way to weighted graphs. An independent set A of size at most t provides a t-improvement if the total weight of its neighborhood in I is less than the weight of A. When all the weights are polynomially bounded, the local improvement algorithm runs in polynomial time. We prove the following theorem improving the trivial performance ratio of d for the local improvement heuristic.

Theorem 2 *Let I be a locally optimal independent set for d-opt, and I^* be the optimal independent set in a node-weighted $d+1$-clawfree graph. Then $w(I^*) \leq (d - 1 + \frac{1}{d})w(I)$, where $w(S)$ denotes the sum of the weights of the nodes in S.*

Note that in the biological example that motivated this research, $d = 4$, and the above theorem shows a performance bound of 3.25 implying an 18% improvement in the worst-case quality of the output solution. Though the improvement is modest, we also demonstrate that the bound is almost best possible for the local improvement heuristic that we analyze.

Theorem 3 *For all positive integers d, t and for all $\epsilon > 0$, there exist node-weighted $d+1$-claw free graphs with a maximum weight independent set I^*, an independent set I that is locally optimal with respect to t-opt but $w(I^*) \geq (d - 1 - \epsilon) \cdot w(I)$.*

The class of d-clawfree graphs includes two other important classes of graphs: graphs with degree at most d and unit disk graphs. The latter is the family of intersection graphs of unit disks in the plane and can be shown to be 6-clawfree by a simple geometric argument. Thus our results provide a tight analysis of the local heuristic for the weighted independent set problem in these classes of graphs. Note that there has also been work on obtaining better ratios for the unweighted independent set problem in bounded degree graphs [BF94].

In Section 2, we describe in more detail how the IR problem arises in the application to molecular biology. In Section 3, we present the NP-hardness proof of Theorem 1. In Section 4, we sketch the basic local improvement algorithm for the unweighted (uniform) case. We then extend the heuristic to the weighted case and present an analysis of the same. We generalize the analysis to arbitrary clawfree graphs and show its tightness. Finally, in Section 5, we conclude with open issues.

2 Motivation

A fundamental problem that arises in the analysis of genetic sequences is to assess the similarity between two such sequences. Traditional notions of similarity have suggested aligning the sequences to reflect globally [NW70] or locally [SW81] similar regions in the string. A global alignment arranges the two strings with spaces inserted within them, so that the characters are organized in columns and most columns contain identical or similar characters in both strings. Such alignments tend to reflect similar regions between the two strings that have remained conserved over the evolutionary process of point mutations that has led to the divergence between the two sequences.

Recent studies on genome rearrangements [BP93, BP95, HP95, KS93, KS94, KR95] have addressed the notion of distances between sequences under more large-scale mutational operations. An example is a "reversal" that works on a large contiguous block of a genomic sequence and reverses the order of certain "markers" in the fragment. Another macro-mutational operation is a transposition that transfers a block of sequence to another position. These rearrangements have been postulated and confirmed to occur in the evolutionary history between several existing species [HCKP94]. The body of work mentioned above addresses the computation of a minimal set of such rearrangement operations to transform an initial sequence A to a final sequence B. The input to such a procedure is a set of disjoint fragments that occur in both the strings, their relative order and orientation in the two strings. When these fragments code for some genetic information, they are termed *genes* and what is being supplied in this case is the gene order and orientation in the two strings for a set of common genes. Thus what is required is a set of fragments which remain highly conserved in both strings (the orientation may be reversed in the two strings), such that the similarity between the two copies of a fragment is appreciable and a large number of such fragments are available for investigation of genome rearrangements. Moreover, no two fragments selected for comparison must overlap in either string, since rearrangements work on segments of the string and therefore cannot separate overlapping fragments.

The problem of selecting fragments of high local similarity between two strings can be tackled by applying one of several known methods for local alignment [SW81] in the literature. The output of such a method is a set of pairs of substrings from A and B that have high local similarity. However, the projection of these pairs in the two strings may not be disjoint as required. It is useful to picture these regions of local similarity as axis-parallel rectangles in the plane where the axes are the two strings A and B being compared. A pair of substrings of high local similarity identifies the rectangle formed by the intersection of the horizontal and vertical slabs corresponding to these substrings in A and B. The rectangle may be weighted with the strength of the local similarity. The resulting problem is to find a maximum-weight set of rectangles whose projections are disjoint in both the axes. This leads to the IR problem introduced earlier. The non-enclosing condition on the projections of the rectangles translates to disallowing similarity pairs in which a substring in one pair is completely contained in that of the other pair. This is a reasonable assumption for data from sequences because the input data can be pruned to eliminate similarities that disobey this requirement.

3 NP-completeness of IR: Proof of Theorem 1

IR is trivially in the class NP. We shall show NP-hardness by transforming from 3SAT.

Let U be an instance of $3SAT$ with m clauses $c_1, \ldots, c_m$ and n variables. For each variable x, define a *cycle gadget* as follows (see Fig. 1(a)). The cycle gadget has exactly $2m$ rectangles arranged in a cycle so that only conflicting pairs are the ones that appear consecutively in the cycle. Label the rectangles in the cycle gadget for x as $x_j, \bar{x}_j$, for $1 \leq j \leq m$. The following lemma is immediate:

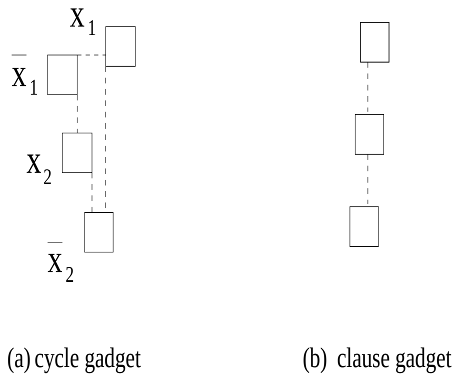

Figure 1: A cycle gadget and a clause gadget for $m = 2$.

Proposition 1 *A cycle gadget with $2m$ rectangles has a maximum independent subset of size m. Further, there are only two such subsets of maximum size, either the set of all $x_j's$ or the set of all $\bar{x}_j's$.*

For each clause c_j, $1 \leq j \leq m$, we define a clause gadget as set of three rectangles (see Fig. 1(b)), one for each literal in the clause, that are pairwise conflicting. If literal x appears in clause c_j, label the corresponding rectangle in the clause gadget as $c_{x,j}$. Finally, place all the rectangles on the plane as follows (see Fig. 2): A pair (a, b) of rectangles conflicts if and only if

- a, b belong to the same clause.
- a, b are adjacent rectangles in a cycle gadget, i.e. $a = x_j$ and $b = \bar{x}_j$ or $b = \bar{x}_{j-1}$.
- $a = c_{x,j}$ and $b = \bar{x}_j$.

Figure 2 gives a layout for the case $n = 3, m = 2$ and $U = (\bar{x} + y + \bar{z})(\bar{x} + \bar{y} + z)$.

Therefore, we have transformed an instance U of $3SAT$ to an instance S of IR, such that $|S| = 2mn + 3m$.

Proposition 2 *$U \in SAT$ if and only if there exists a independent subset $S' \subseteq S$ such that $|S'| \geq mn + m$.*

Proof : Let $U \in 3SAT$ be satisfiable. For any variable x that is TRUE in a valid truth assignment, pick all the rectangles x_j for $1 \leq j \leq m$, otherwise pick $\bar{x}_j$ and for all $1 \leq j \leq m$. Clearly, m rectangles are picked from each of the n cycle gadgets, and they are independent. For each clause c_j, there is at least one literal $x \in c_j$ that is TRUE. By construction, rectangle $c_{x,j}$ only conflicts with other rectangles in the same clause gadget and with $\bar{x}_j$, none of which has been selected. Therefore, one rectangle from each of the clause gadgets can be picked for a total of $mn + m$ rectangles.

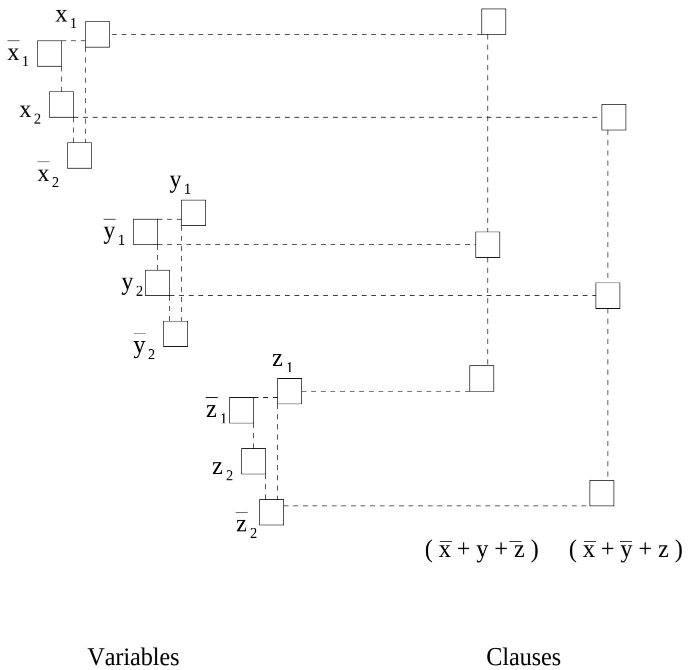

Figure 2: An instance of 3SAT with $n = 3, m = 2$ and $U = (\bar{x}+y+\bar{z})(\bar{x}+\bar{y}+z)$, transformed to an instance of IR

Correspondingly, let $S' \subset S$ be conflicting and $|S'| \geq mn + m$. Now, each cycle gadget can contribute at most m rectangles and each clause can contribute at most 1 rectangle to an independent set. Therefore, in order to get $mn+m$ rectangles each cycle gadget must supply m and each clause must supply 1 rectangle. We consider the following truth assignment. For each clause gadget, if the rectangle chosen is $c_{x,j}$, then set x to be TRUE. Clearly, each clause has at least one TRUE literal, and we only need to ensure that both x and $\bar{x}$ are not set to TRUE. Suppose that was the case, implying that for some $1 \leq j, j' \leq m$, $c_{x,j}$ and $c_{\bar{x},j'}$ were selected. Then, in the cycle gadget of x, neither $\bar{x}_j$ nor $x_{j'}$ can be selected. By proposition 1, this cycle gadget does not supply m independent rectangles.

Theorem 1 follows. □

4 Approximating independent sets in clawfree graphs

We begin with some formal definitions.

Definition 2

1. *Consider a set S of axis-parallel rectangles. Each rectangle may be identified by a pair of intervals (I_x, I_y) defining its projections on the two axes. Rectangle b* overlaps *rectangle a if one of its intervals contains an interval of a. S is* non-overlapping *if no rectangle*

overlaps another. Two rectangles b and a conflict *if at least one of their intervals intersect.*

2. *Define the conflict graph $G_S(V, E)$ of a set S of axis-parallel rectangles as follows: each rectangle corresponds to a vertex $v \in V$, and $(v, w) \in E$ iff a and b are conflicting. In the following, we will drop the subscript S when the context is clear. For $X \subseteq V$, let $G(X)$ be the graph induced by the vertices in X.*

3. *Let $w : V \to \mathcal{R}^+$ be the weight function on rectangles. For $X \subseteq V$, $w(X) = \sum_{x \in X} w(x)$.*

4. *Define the neighborhood of a vertex $v \in V$ as $N(v) = \{x \in V | (v, x) \in E\}$. For $X \subseteq V$, $N(X) = \cup_{x \in X} N(x)$. Also, define $N^i(x) = N(N^{i-1}(x))$ for $i > 0$ and $N^0(x) = \{x\}$.*

As we observed earlier, the problem of finding an independent set of rectangles is that of finding a maximum weighted independent set in the corresponding conflict graph. In order to provide good approximate solutions, we make the following observation.

Lemma 1 *A conflict graph of non-overlapping axis-parallel rectangles is* 5*-clawfree.*

Proof (by contradiction): Assume the statement is not true. There is an independent set of 5 rectangles, all conflicting with one rectangle s. Let s be defined by the interval pair $((x_1, x_2), (y_1, y_2))$. Each rectangle that conflicts with but is not overlapped by s must contain at least one of the 4 points x_1, x_2, y_1, y_2. Assuming 5 such rectangles, one of these points must be contained in two of these rectangles. These two rectangles conflict, a contradiction. □

Consider the problem of finding a maximum weight independent set in a $d + 1$-clawfree graph. One simple heuristic is the greedy one: Add a vertex of maximum weight to the current independent set I, discard all its neighbors and continue. This greedy heuristic performs quite well.

Lemma 2 *Let I^* be a maximum weighted independent set in a $d + 1$-clawfree graph G, and I be an independent set selected by the greedy heuristic. Then $w(I^*) \leq d \cdot w(I)$.*

Proof : The proof is straightforward and hence omitted. □

In the following discussion, we shall attempt to find better algorithms for finding maximum weighted independent sets in $d+1$ clawfree graphs. Even constant factor improvements are desirable, especially when d is small (Note that it is 4 in our application). Specifically, we will focus on a natural heuristic, which is based on iteratively improving the solution through some local changes. This heuristic for computing maximum weight independent sets in $d + 1$-clawfree graphs is described in figure 3.

Note that this algorithm runs in polynomial time if the weights are uniform or if they are polynomial functions of n.

Let us assume for the moment that all rectangles have the same weight. By Theorem 1, the problem remains NP-hard. Halldórsson [Hal95] has shown that t-opt, when applied

Procedure t-opt($\mathcal{I}$)
begin
  $I \leftarrow$ maximal-independent-set ($\mathcal{I}$)
  while $\exists$ independent set $A \subset V - I$, $|A| \leq t$
    and $w(A) > w(N(A) \cap I)$
      $I \leftarrow I \bigoplus A$
  endwhile
  return I
end;

Figure 3: A local improvement algorithm for node weighted graphs

to a $d+1$-clawfree graph, achieves a performance ratio of $\frac{d}{2} + \epsilon$ for any fixed $\epsilon > 0$. It is interesting to note that his analysis uses only a restricted form of improvements that he calls t-ear-improvements. We present below a simple construction that shows that the performance ratio of $\frac{d}{2}$ is the best possible for the local improvement heuristic. To this end, we use the following result of Erdös and Sachs, which can be found in [Bol78].

Lemma 3 *Given positive integers d and g, for all n sufficiently large, there exist d-regular graphs on 2n vertices with girth at least g.*

Theorem 4 *For all positive integers d and t, there exist d+1-claw free graphs with an independent set I, where I is locally optimal with respect to t-opt but* $|I^*| \geq \frac{d}{2}|I|$.

Proof : By Lemma 3, we have a d-regular graph $G = (V, E)$ on n vertices with girth t (for all sufficiently large even n). Construct a new graph G' on vertex set $V \cup E$, and connect vertices x, y in G' if $x \in V$, $y \in E$ and y is incident on x in G. Intuitively, this corresponds to subdividing every edge in G by addition of a new vertex of degree 2. Clearly, G' is bipartite and $d+1$-clawfree. Also, the girth of G' is at least $2t$. Let $I = V$ and $I^* = E$. Since the minimum degree of a vertex in G' is 2, the girth condition implies that every subset of E of size at most t has a neighborhood of size at least $t+1$ in V. Hence, the independent set I is optimal with respect to t-opt. Noting that $|I^*| = |E| = \frac{d}{2}|V| = \frac{d}{2}|I|$ completes the proof. □

Weighted Independent Sets

We now turn to analyze the performance of t-opt for weighted $d+1$-clawfree graphs and show that its performance is provably inferior to the performance for the unweighted case, even when the weights are a polynomial function of n. We also provide matching upper bounds. The following lemma provides a simple upper bound of d and motivates the detailed analysis that follows.

Lemma 4 *Let I be a locally optimal solution for* 1*-opt in a* $d+1$*-clawfree graph. Then if* I^* *is the optimal solution,*

$$w(I^*) \leq d \cdot w(I)$$

Proof : Consider the bipartite graph $G(I \cup I^*)$. By local optimality, we know that for all $v \in V - I$ (in particular, for all $v \in I^*$), $w(v) \leq w(N(v))$, where $N(v)$ refers to the neighborhood of v in $G(I \cup I^*)$. Therefore,

$$\begin{aligned} w(I^*) &= \textstyle\sum_{v\in I^*} w(v) &\leq& \textstyle\sum_{v\in I^*}\sum_{u\in N(v)} w(u) \\ &= \textstyle\sum_{u\in I}\sum_{v\in N(u)} w(u) &\leq& d\cdot w(I) \end{aligned}$$

□

Next, we show that the performance of t-opt improves somewhat as we increase t. While the improvement is somewhat modest, it might still be useful for small values of d.

Let I be a locally optimal independent set for d-opt implying that for all $X \subseteq V - I$, $|X| \leq d$, $w(X) \leq w(N(X) \cap I)$. Let $d(v)$ be the degree of v in $G(I \cup I^*)$. Note that we can without loss of generality, assume that I and I^* are disjoint sets. Otherwise, we work with the graph $G((I \cup I^*) - J)$, where $J = I \cap I^*$. Define $I_i^* = \{v \in I^* | d(v) = i\}$. Clearly, I^* is partitioned into exactly d sets. For $v \in I$, let $d_i(v)$ be the degree of v in $G(I \cup I_i^*)$.

Lemma 5 *Let I be a locally optimal independent set for d-opt. For $1 \leq i \leq d$, let $f_i(u) = 1$ if $d_i(u) > 0$ and* 0 *otherwise. Then,*

1. $i \cdot w(I_i^*) \leq \sum_{u\in I}[d_i(u)\cdot(i-1) + f_i(u)]\cdot w(u)$, *for all* $i \leq d$
2. $\sum_{i=1}^{d} i \cdot w(I_i^*) \leq \sum_{u\in I}[(\sum_{i=1}^{d} d_i(u)\cdot(i-1)) + 1]\cdot w(u)$

Proof : Observe that $G(I \cup I_i^*)$ has exactly $i|I_i^*|$ edges. Therefore,

$$i \cdot w(I_i^*) \;=\; \textstyle\sum_{v\in I_i^*}\sum_{u\in N(v)} w(v) \;=\; \sum_{u\in I} w(N(u))$$

Further, any element $u \in I$ has at most d neighbors, therefore by local optimality for d-opt, $w(N(u)) \leq w(N^2(u))$ for all $u \in I$. Using this and rearranging terms, we get

$$\begin{aligned} i \cdot w(I_i^*) &\leq \sum_{u\in I} w(N^2(u)) \\ &= \sum_{u\in I, d_i(u)>0} (d_i(u)(i-1)+1)w(u) \end{aligned}$$

The last equality follows from the fact that for an $u \in I$ such that $d_i(u) > 0$, the number of times this u is counted in the sum is exactly $d_i(u)(i-1)+1$. This proves the first proposition. The second follows by a similar argument on the graph $G(I \cup I^*)$. □

Next, we prove a technical lemma that we will use to bound the value of $w(I^*)$.

Lemma 6 *For arbitrary integer $d > 0$, consider the following integer program*

$$\begin{array}{l} IP(d) = \max \sum_{i=1}^{d} \frac{(i-1)\cdot d \cdot d_i + (d-i)\cdot f_i}{i} \\ s.t. \\ \sum_{i=1}^{d} d_i \leq d \\ \forall i, f_i \leq d_i \\ \forall i, 0 \leq f_i \leq 1 \\ \forall i, d_i \in \{0, 1, 2, \ldots, d\} \end{array}$$

Then, $IP(d) = d(d-1)$.

Proof : We will prove that the integer program is maximized when $d_i = 1, f_i = 1$ for all i. Clearly, this solution is feasible. Also, observe that any optimal solution will have the property that $\sum_{i=1}^{d} d_i = d$ and $f_i = 1$ for all i such that $d_i > 0$. If this was not true, there would exist some d_i or f_i that could be incremented to increase the value of the objective function. Therefore, it is sufficient to prove that there exists an optimal solution in which $d_i \leq 1$ for all i.

Consider an optimal solution in which this is not true, so that $d_i > 1$ for some i. Then, as $\sum_i d_i \leq d$, there exists j such that $d_j = 0, f_j = 0$. Then, if we decrement d_i by 1, and set $d_j = 1, f_j = 1$, it is easy to see that the solution remains feasible.

Now, the objective function is the sum of d terms, where the contribution of the i^{th} term is

$$\left(d - \frac{d}{i}\right) d_i + \left(\frac{d}{i} - 1\right) f_i$$

Furthermore, the new solution affects only the i^{th} and j^{th} terms of this function. The net change is

$$-\left(d - \frac{d}{i}\right) + \left(d - \frac{d}{j}\right) + \left(\frac{d}{j} - 1\right) = \left(\frac{d}{i} - 1\right)$$

which is non-negative, so the new solution remains optimal. Continuing in this fashion, we eventually get an optimal solution in which all $d_i \leq 1$. □

We now restate and prove Theorem 2.

Theorem 5 *Let I be a locally optimal independent set for d-opt, and I^* be the optimal independent set. Then $w(I^*) \leq (d - 1 + \frac{1}{d})w(I)$.*

Proof : As the sets I_i^* partition I^*, we have the identity

$$d \cdot w(I^*) = \sum_{i=1}^{d} i \cdot w(I_i^*) + \sum_{i=1}^{d-1} (d-i) \cdot w(I_i^*)$$

Applying the bounds obtained from lemma 5, we get

$$d \cdot w(I^*) \;\; \leq \;\; \sum_{u \in I} \left[\sum_{i=1}^{d} \left(d_i(u) \cdot (i-1) + (d-i) \frac{d_i(u)(i-1) + f_i(u)}{i} \right) + 1 \right] \cdot w(u)$$

$$= \sum_{u \in I} \left[\sum_{i=1}^{d} \left(\frac{(i-1) \cdot d \cdot d_i(u) + (d-i) \cdot f_i(u)}{i} \right) + 1 \right] \cdot w(u)$$
$$\leq (IP(d) + 1) \cdot w(I)$$

where $IP(d)$ is the optimum value of the integer program described in lemma 6. Therefore, $w(I^*) \leq (d - 1 + \frac{1}{d})w(I)$. □

Next, we show that our analysis is tight, by demonstrating the existence of claw-free graphs for which the heuristic cannot achieve a performance better than $d-1$. First, we present a technical lemma describing the existence of bipartite graphs with an expansion property. Its proof is implicit in proofs for existence of expander graphs (see for example, Chung[Chu78]).

Lemma 7 *For all positive integers d, t, and for all $\epsilon > 0$ there exists an integer n, and bipartite graphs with bipartition (I, O) with the following properties.*

- $|I| = |O| = n$.
- *For all vertices $v \in I \cup O$, $deg(v) \leq d$.*
- *For all $X \subseteq O$, $|X| \leq t$, $|N(X)| \geq (d - 1 - \epsilon) \cdot |X|$.*

Note that these graphs are different from expander graphs in that the expansion is large (close to the maximum degree) but is required only for subsets of some constant size t. As a consequence of the existence of such graphs, we can derive Theorem 3. We restate and prove it below.

Theorem 6 *For all positive integers d, t and for all $\epsilon > 0$, there exist $d+1$-claw free graphs with an independent set I, such that I is locally optimal with respect to t-opt but $w(I^*) \geq (d - 1 - \epsilon) \cdot w(I)$.*

Proof :

Let (I, O) be a bipartite graph with the expansion property described in lemma 7. Further, for all elements $v \in I$, let $w(v) = 1$, and for all elements $u \in O$, let $w(u) = d - 1 - \epsilon$. By the third condition in Lemma 7, in the graph (I, O), I is a locally optimal solution with respect to t-opt, and $w(O) = (d - 1 - \epsilon) \cdot w(I)$. □

5 Concluding Remarks

We conclude by describing many problems that arise naturally from this work. The problem we study is geometric, and we suspect that it might have applications to problems in computational geometry. However, the only related work that we found was a study of intersecting rectangles (for hidden surface removal) which corresponds to the case when *both* projections intersect. On the other side, can geometric techniques be applied to improve the quality of our solution?

Indeed, the only property we have exploited in finding approximate solutions is clawfreeness in the associated conflict graph. An interesting area of research is to investigate more properties of conflict graphs, and use these properties to find better algorithms or hardness of approximation results.

We have discussed the problem only in the context of pairwise alignments. It is often the case that $k > 2$ sequences are aligned, and biologists are interested in extracting meaningful blocks of locally aligned sequences, which correspond to hypercubes of dimension k. This natural extension to multiple alignment complicates the problem considerably, as the conflict graph of a set of k-dimensional cubes is only $2^k + 1$-clawfree. Different ideas are needed to provide meaningful approximations. It is also possible that general graphs are conflict graphs of some higher dimensional cubes, which might imply some hardness of approximation results for the problem.

Finally, local improvement algorithms have recently been studied extensively, and some interesting positive results have been obtained for related problems, such as independent sets and vertex covers in degree bounded graphs, 3-DM matching, k-set-packing etc. [BF94, Hal95]. Halldórsson [Hal95] shows reducibilities between these problems and uses these reductions to analyze local improvement heuristics for the unweighted case. We hope that the ideas in our analysis can be extended to analyzing heuristics for the weighted versions of these problems. This is particularly interesting for the case of independent sets in bounded degree graphs, where a slightly better local improvement can be applied to improve performance in the unweighted case [BF94], but nothing is known about the weighted version.

Acknowledgement

We would like to thank Dan Gusfield for explaining the biological motivation for the problem and Steve Skiena for useful discussions regarding the NP-completeness of the problem.

References

[ALM+92] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In *33rd IEEE Symp. on Foundations of Computer Science*, 1992.

[BF94] P. Berman and M. Fürer. Approximating maximum independent set in bounded degree graphs. In *Fifth ACM-SIAM Symp on Discrete Algorithms*, pages 365–371, 1994.

[Bol78] B. Bollobas. *Extremal Graph Theory.* Academic Press, 1978.

[BP93] V. Bafna and P. Pevzner. Genome rearrangements and sorting by reversals. In *34th IEEE Symp. on Foundations of Computer Science*, pages 148–157, 1993.

[BP95] V. Bafna and P. Pevzner. Sorting permutations by transpositions. In *The sixth annual ACM-SIAM symposium on discrete algorithms*, pages 614–623, 1995.

[Chu78] F. R. K. Chung. On Concentrators, Superconcentrators, Generalizers, and Non-blocking Networks. *The Bell Systems Technical Journal*, 58:1765–1777, 1978.

[Hal95] M. M. Halldórsson. Approximating discrete collections via local improvements. *The sixth annual ACM-SIAM symposium on discrete algorithms*, pages 160–169, 1995.

[HCKP94] S. Hannenhalli, C. Chappey, E. Koonin, and P. Pevzner. Scenarios for genome rearrangements: Herpesvirus evolution as a test case. In *Proc. of 3rd Intl. Conference on Bioinformatics and Complex Genome Analysis*, 1994.

[HP95] S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip. In *27th Annual ACM Symposium on Theory of Computing*, 1995.

[KR95] J. D. Kececioglu and R. Ravi. Of mice and men: Evolutionary distances between genomes under translocations. *The sixth annual ACM-SIAM symposium on discrete algorithms*, pages 604–613, 1995.

[KS93] J. Kececioglu and D. Sankoff. Exact and approximation algorithms for the inversion distance between two permutations. In *Proc. of 4th Ann. Symp. on Combinatorial Pattern Matching*, Lecture Notes in Computer Science 684, pages 87–105. Springer Verlag, 1993.

[KS94] J. Kececioglu and D. Sankoff. Efficient bounds for oriented chromosome inversion distance. In *Lecture notes in computer science*, volume 807, pages 307–325, 1994.

[NW70] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[SW81] T. F. Smith and M. S. Waterman. The identification of common molecular sequences. *Journal of Molecular Biology*, 147:195–197, 1981.