How Easy Is Local Search?

DAVID S. JOHNSON, CHRISTOS H. PAPADIMITRIOU

& MIHALIS YANNAKAKIS

Balabhaskar Balasundaram

baski@tamu.edu

Dept. of Industrial Engineering

TAMU

Overview

- Search Problems in Combinatorial Optimization.
- Local Optima.
- Motivation.
- The Class PLS.
- PLS Reducibility.
- First PLS-Complete Problem.
- Other PLS-Complete Problems.
- References.

Search Problems

- Each instance is associated with a finite set of feasible solutions.
- Each feasible solution has a cost.
- Objective is to find a solution of minimum (maximum) cost.

Search Problems

- Each instance is associated with a finite set of feasible solutions.
- Each feasible solution has a cost.
- Objective is to find a solution of minimum (maximum) cost.
- Define a “Neighborhood” for each solution.
- A solution is said to be locally optimal if there is no solution in its neighborhood with “better” cost.

Motivation

- Local search algorithms have been observed to be efficient in practice.
- The assumption that local optima are easy to obtain has never been challenged.

Motivation

- Local search algorithms have been observed to be efficient in practice.
- The assumption that local optima are easy to obtain has never been challenged.
- How easy is it to find a local optimum ?

The Class PLS - Polynomial-time Local Search

A PLS problem L can be a maximization or minimization problem:

- L has a set D_L of instances.
- For each instance $x \in D_L$ we have a finite set $F_L(x)$ of solutions, all with the same polynomially bounded length.
- For each solution $s \in F_L(x)$, we have a non-negative integer *cost* $c_L(s, x)$ and also a subset $N(s, x) \subseteq F_L(x)$ called the *neighborhood* of s.
- And the following algorithms must exist.

The Class PLS -Contd.

Polynomial-time algorithms A_L, B_L & C_L such that:

- Given an instance $x \in D_L$, A_L produces a particular standard solution $A_L(x) \in F_L(x)$.
- Given an instance x and a string s, B_L checks if $s \in F_L(x)$ and if so, computes that cost $c_L(s, x)$.
- Given an instance x and a solution s, C_L identifies an solution $s' \in N(s, x)$ with better cost if it exists OR reports that s is locally optimal.

Standard Local Search Algorithm

Inherent in the definition of a PLS problem, is the following algorithm:

1. Given x, use A_L to produce a starting solution $s = A_L(x)$.
2. Repeat until locally optimal:
   Apply algorithm C_L to x and s.
   If C_L yields a better cost neighbor s' of s, set $s = s'$.

Standard Algorithm Problem

Given x, find the local optimum s that would be output by the *standard local search algorithm* for L on input x.

Standard Algorithm Problem

Given x, find the local optimum s that would be output by the *standard local search algorithm* for L on input x.

LEMMA 1. There is a PLS problem L whose standard algorithm problem is NP-hard.

Hence, general polynomial time algorithms for the standard algorithm problems, seems unlikely.

What About Some Local Optimum ?

LEMMA 4. If a PLS problem is NP-Hard, then NP=co-NP.

Finding *some* local optimum for a PLS problem L is an "easier" task than finding the local optimum that is output by the standard algorithm for L.

PLS Reducibility

A problem L in PLS is PLS-reducible to another, K, if there are polynomial-time computable functions f and g such that

1. f maps instances x of L to instances $f(x)$ of K.
2. g maps (solution of $f(x), x$) pairs to solutions of x.
3. For all instances x of L, if s is a local optimum for instance $f(x)$ of K, then $g(s, x)$ is a local optimum for x.

$$L \quad \leq_{pls} \quad K$$

$$x \quad \stackrel{f}{\longrightarrow} \quad f(x)$$

$$(g(s), x) \quad \stackrel{g}{\longleftarrow} \quad (s, f(x))$$

PLS Reducibility -Contd.

LEMMA 5. If L, K, J are problems in PLS such that $L \leq_{pls} K$ and $K \leq_{pls} J$, then $L \leq_{pls} J$.

LEMMA 6. If L, K are problems in PLS such that $L \leq_{pls} K$ and if there is polynomial-time algorithm for finding local optima for K, then there is also a polynomial-time algorithm for finding local optima for L.

A problem L in PLS is said to be PLS-Complete if every problem in PLS is PLS-reducible to L.

Problem Definition: FLIP

FLIP: Given a circuit x with m inputs and n outputs, a solution in $F_{FLIP}(x)$ is any bit vector s with m components. It has m bit vectors of length m with hamming distance one from s as neighbors. The cost of solution s is defined as $\sum_{j=1}^{n} 2^j y_j$, where y_j is the j^{th} output of the circuit x with input s.

Algorithm A_{FLIP} returns the all-1 vector, B_{FLIP} (cost-computation) is straight-forward from above and C_{FLIP} returns the best of the m neighbors of s (ties broken lexicographically) if s is not locally optimal.

FLIP is a minimization problem.

First PLS-Complete Problem

THEOREM 1. FLIP is PLS-Complete.

So is MAXFLIP.

COROLLARY 1.1.
(a) The standard algorithm problem for FLIP is NP-hard.
(b) There are instances of FLIP for which the standard algorithm requires exponential time.

Other PLS-Complete Problems

- GRAPH PARTITIONING under Kernighan-Lin neighborhood. Johnson, Papadimitriou & Yannakakis 1988.
- GRAPH PARTITIONING under the Swap neighborhood. Schäffer & Yannakakis 1991.
- MAX CUT under the Flip neighborhood. Schäffer & Yannakakis 1991.
- MAX 2SAT under the Flip neighborhood. Schäffer & Yannakakis 1991.
- TSP under the k-opt neighborhood. Krentel 1989.
- TSP under the Lin-Kernighan neighborhood. Papadimitriou 1990.

References

1. D. S. Johnson, C. H. Papadimitriou and M. Yannakakis, How Easy is Local Search?. Journal of Computer and System Sciences 37 (1988), 79-100.
2. Mark W. Krentel. Structure in locally optimal solutions. Proc. 30th Annual Symposium on Foundations of Computer Science (1989) 216-221, Research Triangle Park, North Carolina.
3. M. W. Krentel, On Finding and Verifying Locally Optimal Solutions. SIAM Journal on Computing 19(4) (1990) 742-749.
4. C. H. Papadimitriou, A. A. Schäffer, and M. Yannakakis, On the complexity of local search (extended abstract). Proc. 22nd Annual ACM Symposium on Theory of Computing (1990) 438-445, Baltimore, Maryland.
5. A. A. Schäffer and M. Yannakakis, Simple local search problems that are hard to solve. SIAM Journal on Computing 20(1) (1991) 56-87.