RUTCOR
RESEARCH
REPORT

On the complexity of local search in unconstrained quadratic binary optimization

Dávid Papp[a]

RRR 1-2007, JANUARY, 2007

RUTCOR
Rutgers Center for
Operations Research
Rutgers University
640 Bartholomew Road
Piscataway, New Jersey
08854-8003
Telephone: 732-445-3804
Telefax: 732-445-5472
Email: rrr@rutcor.rutgers.edu
http://rutcor.rutgers.edu/∼rrr

[a]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003. Email: dpapp@rutcor.rutgers.edu

Rutcor Research Report
RRR 1-2007, JANUARY, 2007

On the complexity of local search in unconstrained quadratic binary optimization

Dávid Papp

Abstract. We show by construction that the procedure of local search for the problem of finding a local optimum of a quadratic pseudo-boolean function takes exponential time in the worst case, regardless of the neighbor selection and tie-breaking rules.

1 Introduction

We consider the problem of finding a local maximum of the *quadratic pseudo-boolean function*

$$f(\mathbf{x}) = f(x_1, \dots, x_n) = \sum_{i=1}^{n} c_i x_i + \sum_{1 \le i < j \le n}^{n} c_{ij} x_i x_j, \quad \mathbf{x} \in \{0,1\}^n,\ c_i \in \mathbb{R},\ c_{ij} \in \mathbb{R}.$$

A straightforward approach is to use *(increasing) local search*, that is, start from an arbitrary vertex of the boolean hypercube, and if it is not a local maximum, move to one of its *feasible neighbors*, i.e. to a neighbor where the function attains a larger value than at the current vertex. This method clearly works (terminates in a finite number of steps) for an arbitrary pseudo-boolean function, but its complexity is not clear in general.

In this paper we show that the time complexity of the above local search algorithm is exponential *regardless of the neighbor selection and tie-breaking rules.* Note that we do not consider the problem of finding a good initial point, because with the smartest possible strategy we could choose a local optimum to be the initial point. Hence we assume that the local search always starts from an "arbitrary" vertex, which in this paper will be the origin.

2 Previous Results

In the survey paper [1] a few problems, including the MAX-CUT problem, are listed which cannot be solved in (worst-case) subexponential time by local search, but it does not contain any proofs. The two references given in [1] are [2] and [3], both of which consider the MAX-CUT problem. This is clearly a special case of maximizing a quadratic pseudo-boolean function, and hence such a result would be stronger than the one in this paper. Unfortunately, [2] is an unavailable manuscript. The paper [3] contains a proof of the claim that MAX-CUT can be reduced with a tight PLS reduction to the FLIP problem, which is known to be PLS-complete [1]. This in turn proves that local search is indeed exponential for MAX-CUT. However, this proof is not constructive, as it does not allow us to construct a sequence of MAX-CUT instances, increasing in size, for which local search takes exponentially increasing number of steps. The construction given in this paper, together with its proof, is considerably simpler, and it does not use the theory related to PLS-completeness.

Another type of result is the one in [4]. It is proven that the k-OPT TRAVELING SALESMAN problem is PLS-complete for some fixed k, and it is stated as a corollary (but without a separate proof) that there are instances of this problem for which the k-opt local search takes exponential time. (Here $k \geq 10000$.) Let us note, however, that PLS-completeness does not imply the existence of exponentially hard instances. And even though the corollary is indeed true, the proof, again, does not lead to a construction.

The aim of this paper is to provide a simple construction, which shows that for some instances of quadratic pseudo-boolean optimization every increasing local search algorithm takes exponential time to find a local maximum.

3 The Construction

Let us call a path (defined as a sequence of adjacent vertices) *increasing* if the objective function is monotone and nondecreasing along its vertices. In order to prove our claim it suffices to show that for a sequence of polynomials with increasing number of variables any increasing path that connects the origin to a local optimum has length exponential in the size of the polynomials.

Let us define the sequence of n-variate polynomials $\{f_n\}$ ($n = 2, 6, 10, \dots$) recursively as follows.

$$\begin{aligned} f_2(x_1, x_2) &:= x_1 + x_2, \\ f_{n+4}(\mathbf{x}, x_{n+1}, x_{n+2}, x_{n+3}, x_{n+4}) &:= f_n(\mathbf{x}) - M_n n^2 x_{n+1} + M_n(n+1)\left(\sum_{i=1}^{n} x_i\right) x_{n+1} - \\ &- x_{n+2} - 2M_n n\left(\sum_{i=1}^{n} x_i\right) x_{n+2} + 2M_n n(n+2) x_{n+1} x_{n+2} - \\ &- 4\left(\sum_{i=1}^{n} x_i\right) x_{n+3} + 2x_{n+1}x_{n+3} + 2x_{n+2}x_{n+3} - 3x_{n+3} + \\ &+ (M_n(n-1)+4)\left(\sum_{i=1}^{n} x_i\right) x_{n+4} + 6M_n n^2 x_{n+3}x_{n+4} - 5M_n n^2 x_{n+4}, \end{aligned}$$

where $M_n := \max f_n - \min f_n + 1$. Let p_n denote the length of the shortest path local search may follow to a local maximum starting from the zero vector. (Eg. $p_2 = 2$.)

We can obtain the following bound for the size of the polynomials.

Lemma 1. *The size of f_n is polynomial in n.*

Proof. It is enough to show that the largest coefficient has size polynomial in n. Notice, that from the recursive definition, and because $\max f_n - \min f_n$ can be bounded from above by the sum of the absolute values of the coefficients, $M_{n+4} = O(M_n n^2 + \text{poly}(n))$, consequently $M_n = O\left((n!)^2\right)$, which shows that every coefficient of the polynomial can be represented by $O(n \log n)$ bits. □

First we examine which local maximums may be accessible from the origin via an increasing path.

Lemma 2. *Every increasing path of f_n starting from the origin ends at the antipodal of the origin,* $(1, 1, \dots, 1)$.

Proof. The theorem holds if $n = 2$. We prove by induction that if it holds for f_n, then it holds for f_{n+4} as well.

One way to see this is to take the partial derivatives of f_{n+4} with respect to each variable, and check their signs at points which may be included in an increasing path. This solution can be summarized in a table, as shown in Table 1 on page 7. Notice that for boolean functions the partial derivatives are the same as discrete partial derivatives, which give the change in the function value upon the change of one variable. Changing a variable from zero to one results in an increase if and only if the corresponding partial derivative is positive. Similarly, changing a variable from one to zero results in an increase if and only if the corresponding partial derivative is negative. Each but the last column of the table contains precisely one cell which indicates an increasing direction, implying that there is essentially one increasing path. In the first column it can be either the first or second cell, but for any fixed $\curvearrowleft$ only one of them. In the last column there is no partial derivative that indicates a possible increase. The increasing paths can hence be characterized based on this table: each of them first reaches the point $(1, 1, \ldots, 1, 0, 0, 0, 0)$, then reaches $(1, 1, \ldots, 1, 1, 1, 0, 0)$ in two more steps, then in n steps goes to $(0, 0, \ldots, 0, 1, 1, 0, 0)$, after which it goes to $(0, 0, \ldots, 0, 1, 1, 1, 1)$, and finally it follows a path which is a translation of the first part of the path to arrive at $(1, 1, \ldots, 1, 1, 1, 1, 1)$, where the path ends.

Alternatively, we can determine the increasing paths by directly going along the possible paths to verify that increasing paths end at the antipodal of the origin. To simplify this process, we hereby state the following properties of increasing paths of f_{n+4}, which are all very easy to prove, and which we are going to use repeatedly.

1. *If $x_{n+2} = x_{n+3} = x_{n+4} = 0$, then we can change x_{n+1} from 0 to 1 with an increase in the objective function if and only if $x_i = 1$ for every $i \leq n$.* This is because then the coefficient of x_{n+1} is $M_n n > 0$, while otherwise it is at most $-M_n(n^2+(n-1)(n+1)) < 0$.

2. *If $x_{n+3} = x_{n+4} = 0$, then we can change x_{n+2} from 0 to 1 with an increase in the objective function if and only if $x_{n+1} = 1$.* This is trivial.

3. *If $x_{n+4} = 0$, then we can change x_{n+3} from 0 to 1 with an increase in the objective function if and only if $x_{n+1} = x_{n+2} = 1$, and $x_i = 0$ for every $i \leq n$.* The "if" direction is trivial, we can increase the objective function by one. To see that the "only if" part holds, observe that if $x_i = 1$ for some $i \leq n$, then the objective function will increase by at most $-4 + 2 + 2 - 3 < 0$, and if x_{n+1} or x_{n+2} is zero, then the increase is at most $2 - 3 < 0$.

4. *We can change x_{n+4} from 0 to 1 with an increase in the objective function if and only if $x_{n+3} = 1$.* In the "if" direction observe that the coefficient of $x_{n+3}x_{n+4}$ is larger than the sum of the absolute values of every negative coefficient of the terms involving x_{n+4}. On the other hand, if $x_{n+3} = 0$, then the term $-5M_n n x_{n+4}$ is dominating.

5. *An increasing path that enters the subcube defined by $x_{n+1} = x_{n+2} = 1$ never leaves that subcube.* Leaving the subcube would result in a decrease of the objective function, as the the coefficient of $x_{n+1}x_{n+2}$ dominates the sum of the absolute values of every negative coefficient of terms which contain these two variables.

6. *An increasing path that enters the subcube defined by $x_{n+3} = x_{n+4} = 1$ never leaves that subcube.* The proof is similar to the previous one.

7. *If $x_i = 0$ for every $i \leq n$ and $x_{n+1} = x_{n+2} = 1$, then we can change some x_i with $i \leq n$ to 1 only if $x_{n+4} = 1$.* Because if $x_{n+4} = 0$, then changing x_i would result in an increase of less than $M_n(n+2)$ due to the terms f_n and $x_i x_{n+1}$, which is dominated by the decrease $2M_n n$ due to the term $x_i x_{n+2}$.

8. *For any $\mathbf{x} \in \{0,1\}^n$, $f_{n+4}(\mathbf{x}, 1, 1, 1, 1) = f_n(\mathbf{x}) + C_n$, where C_n depends on n, but not on $\mathbf{x}$.* Substitute the values of the last four variables into the definition of f_{n+4} and observe that every term in $\{x_i x_j \mid i \leq n < j\}$ cancels out.

Consider now an increasing path of f_{n+4}, starting at the origin. By the inductive assumption and the definition of f_{n+4} as long as the path remains in the subcube defined by $x_{n+1} = x_{n+2} = x_{n+3} = x_{n+4} = 0$, it coincides with an increasing path of f_n. Claims 1–4 imply that it cannot leave the subcube until $x_i = 1$ for each $i \leq n$, and that when this condition is satisfied, there is exactly one possible direction the path can proceed, which is changing x_{n+1} to 1.

Hence, the path must arrive at vertex $(\mathbf{1}, 1, 0, 0, 0)$. Now we cannot change back x_{n+1} to 0, because we never step back. Also, we cannot change any x_i with $i \leq n$ back to 0, because the possible increase in the term f_n is less than M_n, which does not compensate for the decrease in the term $M_n(n+1)x_i x_{n+1}$. This observation, together with Claims 2–4, yields that the only feasible neighbor of the current vertex is $(\mathbf{1}, 1, 1, 0, 0)$.

Claim 5 ensures that from now on the value of x_{n+1} and x_{n+2} never change. Claims 3 and 4 guarantee that the last two variables also cannot change until the first n variables are all set to zero. At the same time, due to the terms $-2M_n x_i x_{n+2}$, any x_i with $i \leq n$ can be changed from 1 to 0 with an increase of the objective function. Consequently in the next n steps the increasing path can only continue by changing every x_i, $i \leq n$, one by one, back to zero. (Clearly these steps cannot be interspersed by $0 \to 1$ changes, as such steps would decrease the objective function, due again to the terms $-2M_n x_i x_{n+2}$.)

Eventually we arrive at the vertex $(\mathbf{0}, 1, 1, 0, 0)$. According to Claim 5 the two components of value 1 will never change, and knowing this Claim 7 prevents the first n variables from changing their values until x_{n+4} is set to 1. Now Claims 3 and 4 imply that the only feasible neighbor of the current vertex is $(\mathbf{0}, 1, 1, 1, 0)$, and the only feasible neighbor of that one is $(\mathbf{0}, 1, 1, 1, 1)$.

Claims 5 and 6 now ensure that from now on only the first n variables may have their value changed, and Claim 8 implies that $(\mathbf{x}', 1, 1, 1, 1)$ is a feasible neighbor of $(\mathbf{x}, 1, 1, 1, 1)$ if and only if $(\mathbf{x}', 0, 0, 0, 0)$ is a feasible neighbor of $(\mathbf{x}, 0, 0, 0, 0)$. Consequently the remaining subpath of the path being considered is a translation of an increasing path of f_n. Using our inductive assumption the increasing path of f_{n+4} also ends at $(1, 1, \ldots, 1)$. □

Reviewing the previous proof one can observe that we have also proved our main theorem.

Theorem 1. *Any local search algorithm starting from the origin takes exponentially many steps (both in n and in the size of f_n) to find a local maximum of the polynomial f_n.*

Proof. It is enough to show that the shortest increasing path, corresponding to f_n, connecting the origin to its antipodal has length exponential in n. Then from Lemma 1 it also follows that the path length is exponential in the size of f_n, too.

In the previous proof we have seen that every increasing path corresponding to f_{n+4} contains two disjoint subpaths which are isomorphic to some increasing path corresponding to f_n. Consequently $p_{n+4} \geq 2p_n$, which yields (since $p_2 = 2$) the inequality $p_n \geq 2^{n/4}$. □

Even simpler constructions can be given for special cases of the problem (in terms of the heuristics used). An interesting special case is the when the *greedy local search* procedure is applied: in this case the neighbor selection rule is to choose the one with the highest objective function value. It is left to the reader to apply any of the techniques used in the proof of Theorem 1 to prove the following.

Theorem 2. *The greedy local search algorithm starting from the origin takes exponentially many steps (both in n and in the size of g_n) to find a local maximum of the polynomial g_n ($n \in \{2, 6, 10, \dots\}$) defined by:*

$$\begin{aligned} g_2(x_1, x_2) :=& 2x_1 + x_2, \\ g_{n+4}(\mathbf{x}, x_{n+1}, x_{n+2}, x_{n+3}, x_{n+4}) :=& 4g_n(\mathbf{x}) + 3x_{n+1} - \sum_{i=1}^{n} (2M_n + i + 2) x_i x_{n+2} \\ &+ 3M_n n x_{n+1} x_{n+2} + 2x_{n+3} + M_n x_{n+1} x_{n+3} \\ &- 4M_n n x_{n+2} x_{n+4} + 5M_n n x_{n+3} x_{n+4}, \end{aligned}$$

where $M_n := 4(\max g_n - \min g_n)$.

Starting from the zero vector, the path the greedy local search procedure might follow is unique (i.e., there are no ties), the path ends at the unique local maximum (which is the complement of the characteristic vector of the set $\{4, 8, \cdots, n-2\}$), and it has length greater than $2^{n/4}$.

4 Acknowledgements

I thank Endre Boros for helpful discussions.

References

[1] M. Yannakakis. Computational Complexity, in E. Aarts, JK. Lenstra (eds.) *Local search in combinatorial optimization*, pp. 19–55., Princeton University Press, Princeton, 2003. ISBN 0-691-11522-2.

[2] A. Haken. Connectionist networks that take exponential time to stabilize, Manuscript.

[3] AA. Schäffer, M. Yannakakis. Simple local search problems that are hard to solve, *SIAM Journal on Computing* **20**:56–87, 1991.

[4] MW. Krentel. On finding and verifying locally optimal solutions, *SIAM Journal on Computing* **19**:742–749, 1990.

Partials	at $(\mathbf{x},0,0,0,0)$	at $(\mathbf{x},1,0,0,0)$	at $(\mathbf{x},1,1,0,0)$	at $(\mathbf{0},1,1,0,0)$	at $(\mathbf{0},1,1,1,0)$	at $(\mathbf{x},1,1,1,1)$
$\frac{\mathrm{d}}{\mathrm{d}x_i} f_{n+4}$ $(1 \leq i \leq n)$	$\frac{\mathrm{d}}{\mathrm{d}x_i} f_n$	$M_n(n+1)+$ $+\frac{\mathrm{d}}{\mathrm{d}x_i} f_n > 0$	$M_n(1-n)+$ $+\frac{\mathrm{d}}{\mathrm{d}x_i} f_n < 0$	$M_n(1-n)+$ $+\frac{\mathrm{d}}{\mathrm{d}x_i} f_n < 0$	$M_n(1-n)+$ $+\frac{\mathrm{d}}{\mathrm{d}x_i} f_n - 4$ < 0	$\frac{\mathrm{d}}{\mathrm{d}x_i} f_n$
$\frac{\mathrm{d}}{\mathrm{d}x_{n+1}} f_{n+4}$	$-M_n(n^2-(n+1)S)$ $> 0 \text{ iff } S = n$	$-M_n n^2+$ $+M_n(n+1)S)$ $> 0 \text{ iff } S = n$	$M_n(n+1)S+$ $+M_n n(n+4)$ > 0	$M_n n(n+4)$ > 0	$M_n n(n+4)+$ $+2 > 0$	$M_n(n+4)+$ $+M_n(n+1)S+$ $+2 > 0$
$\frac{\mathrm{d}}{\mathrm{d}x_{n+2}} f_{n+4}$	$-1-2M_n nS < 0$	$-1-2M_n nS+$ $+2M_n n(n+2)$ > 0	$-1-2M_n nS+$ $+2M_n n(n+2)$ > 0	$-1+4M_n+$ $+2M_n n > 0$	$2M_n n(n+2)+$ $+1 > 0$	$2M_n n(n+2)-$ $-2M_n nS$ $+1 > 0$
$\frac{\mathrm{d}}{\mathrm{d}x_{n+3}} f_{n+4}$	$-3-4S < 0$	$-1-4S < 0$	$1-4S$ $< 0 \text{ iff } S \neq 0$	1	1	$6M_n n^2+$ $+1-4S > 0$
$\frac{\mathrm{d}}{\mathrm{d}x_{n+4}} f_{n+4}$	$-5M_n n^2+4S+$ $+M_n(n-1)S < 0$	$-5M_n n^2+4S+$ $+M_n(n-1)S$ < 0	$-5M_n n^2+4S+$ $+M_n(n-1)S$ < 0	$-5M_n n^2 < 0$	$M_n n^2 > 0$	$M_n n^2+4S+$ $+M_n(n-1)S$ > 0

Table 1: Signs of partial derivatives at specific points. $S := \sum_{i=1}^{n} x_i$