On the Approximability of the Selection Problem

Herbert Baier*
Lehrstuhl für Theoretische Informatik
Universität Würzburg, Germany
baier@informatik.uni-wuerzburg.de

December 1996

Abstract

Under the assumption *EXPTIME* $\neq$ *NEXPTIME*, we show the nonexistence of polynomial-time bounded-error approximation algorithms solving the MAX-Selection problem. In addition, a linear reduction from the MAX2SAT problem to the MAX-Selection problem is given. Therefore, if there exists a polynomial-time approximation scheme for the MAX-Selection problem, then *P* = *NP*.

Keywords: Computational complexity, reducibilities, approximation algorithms

1 Introduction

Although we do not know in present days how to solve efficiently (i.e. in polynomial time) an *NP*-complete optimization problem, an approximate solution for such a problem with a bounded error may be sufficient. However, we can be more ambitious and for any bounded error we would want to solve efficiently this problem (polynomial-time approximation scheme – PTAS). Extending Fagin's [4] syntactic definition of *NP*, Papadimitriou and Yannakakis [10] defined a suitable complexity class of *NP*-optimization problems called *MAXSNP* which contains complete problems with respect to a particular reduction, namely the linear reduction. An interesting property of this reduction is: If an *NP*-complete optimization problem S linear-reduces to an *NP*-complete optimization problem T and there exists a PTAS for T, then there exists also a PTAS for S. Hence, there exists a PTAS for a *MAXSNP*-complete problem if and only if there exist PTAS for every problem in *MAXSNP*. However, Arora *et al.* [1] proved: If there exists a PTAS for a *MAXSNP*-complete problem, then *P* = *NP*. Thus, we can prove the nonexistence of PTAS for an *NP*-complete optimization problem under assumption *P* $\neq$ *NP* showing a linear reduction from a *MAXSNP*-complete problem to it.

Under the assumption *EXPTIME* $\neq$ *NEXPTIME*, Feige *et al.* [5] developed a method to show the nonexistence of polynomial-time bounded-error approximation algorithms solving some *NP*-complete optimization problems. The basic idea is to find an exponential-time many-one reduction from the problems that can be accepted by two-prover one-round interactive proof systems (*MIP*$(2,1)$) to the problem in question such that the following holds: If there exists a polynomial-time bounded-error approximation algorithm solving this problem, then we have exactly solutions for the problems in *MIP*$(2,1)$, i.e. *MIP*$(2,1) \subseteq$ *EXPTIME*. However, Feige and Lovász [7] proved *MIP*$(2,1) =$ *NEXPTIME*, and we are done.

For the MAX-Selection problem we are given disjoint sets $A_1, \ldots, A_k, B_1, \ldots, B_l$ and a function $f : (A_1 \cup \cdots \cup A_k) \times (B_1 \cup \cdots \cup B_l) \to \mathbb{N}$, and we seek selection sets A and B, i.e. A (B, respectively) contains

*Supported by **CNPq** grant 290020/92.1 (Brazil) and in part by **DAAD** (Germany).

at most one element of each A_i (B_i, respectively), such that $\sum_{a \in A, b \in B} f(a,b)$ is as large as possible. We apply the two methods described above on the MAX-Selection problem. Thus, if there exists a PTAS for this problem then *P* = *NP*, and if there exists a polynomial-time bounded-error approximation algorithm solving it then *EXPTIME* = *NEXPTIME*.

2 Preliminaries and Previous Results

Since it does not make sense to study the polynomial-time approximate solutions to *NP*-complete optimization problems when *P* = *NP*, in what follows we assume *P* ≠ *NP*. In the present paper, we consider only maximization problems. Thus, in the following definitions and results we restrict our self to the maximization case (for minimization case and details see [9, 8]).

An approximation algorithm can be defined in many ways. We will adopt the following definition. Let S be an *NP*-complete optimization problem, $S(x)$ be the set of feasible solutions of S on input x and, for each solution $s \in S(x)$, let $c(s)$ be the positive integer cost of solution s. The *optimum solution* of the optimization problem S on input x is defined as follows (by above remark S is a maximization problem): $\mathrm{OPT}_S(x) =_{\mathrm{df}} \max_{s \in S(x)} c(s)$. Let M be a polynomial-time algorithm which on input x returns a feasible solution $M(x) \in S(x)$. If for every input x we have $(1-\epsilon) \cdot \mathrm{OPT}_S(x) \leq c(M(x)) \leq \mathrm{OPT}_S(x)$ then we say that M is a *polynomial-time ϵ-approximation algorithm solving S*. Note that this definition makes sense for constant ϵ such that $0 < \epsilon \leq 1$ and that a smaller ϵ means a better approximation. Now, define

$$R(S) =_{\mathrm{df}} \{\epsilon : 0 < \epsilon \leq 1 \wedge \text{ there is a polynomial-time } \epsilon\text{-approximation algorithm solving } S\}$$

as the *error ratios of S*. Note that $\max R(S)$ is always defined and it is 1. If $R(S) = \{1\}$ then no polynomial-time algorithm can guarantee solutions for instances of S which differs from optimal solutions by a fraction. Thus, we say that S can be solved by a *polynomial-time bounded-error approximation algorithm* if and only if there exists an ϵ such that $\epsilon < 1$ and $\epsilon \in R(S)$.

Normally, for an *NP*-complete optimization problem S we are interested in the minimal value of $R(S)$. However, it is possible that no such minimum exists, i.e. there exists a sequence of approximation algorithms for S whose error ratios have limit 0. A *polynomial-time approximation scheme* (PTAS) for an *NP*-complete optimization problem S is a family of algorithms containing for each $0 < \epsilon \leq 1$ a polynomial-time ϵ-approximation algorithm solving S. This leave to a natural classification of the *NP*-complete optimization problems into three categories

1. Those which approximability has no limits in the sense of existing PTAS (e.g. the MAXKNAPSACK problem [8]);
2. Those which can be approximated to a fixed error ratio less than 1 (e.g. the MAXCUT problem [9]);
3. Those which can only be "approximated" to an error ratio 1 (e.g. the MAXCLIQUE problem [1]).

Extending Fagin's [4] syntactic definition of *NP*, Papadimitriou and Yannakakis [10] defined a suitable complexity class of *NP*-optimization problems called *MAXSNP* which contains interesting *NP*-complete optimization problems. Completeness in *MAXSNP* is showed using *L*-reductions (for linear reductions).

Definition (*L*-reduction). *Given two NP-complete optimization problems S and T, (f, g) is an L-reduction from S to T if and only if f and g are computable in polynomial-time and there are constants $\alpha, \beta > 0$ such that*

1. *if x is an instance of S then $f(x)$ is an instance of T with*
$$\mathrm{OPT}_T(f(x)) \leq \alpha \cdot \mathrm{OPT}_S(x)$$
2. *if s is a feasible solution of $T(f(x))$ then $g(s)$ is a feasible solution of $S(x)$ such that*
$$|\mathrm{OPT}_S(x) - c(g(s))| \leq \beta \cdot |\mathrm{OPT}_T(f(x)) - c(s)|$$

Thus, a problem in *MAXSNP* is *MAXSNP*-complete, if all problems in *MAXSNP* are *L*-reducible to it. The MAX2SAT problem (i.e. the problem of determining the maximum number of clauses in a 2SAT expression that can be satisfied by a truth assignment) is an example of a *MAXSNP*-complete problem. Two important remarks that are a direct consequence of *L*-reduction definition follow [10]:

- It is transitive, i.e. if (f, g) is an *L*-reduction from S to T with constants α and β, and (f', g') is an *L*-reduction from T to X with constants α' and β', then $(f \circ f', g \circ g')$ is an *L*-reduction from S to X with constants $\alpha \cdot \alpha'$ and $\beta \cdot \beta'$, where $\circ$ is the traditional composition of functions;
- If S *L*-reduces to T with constants α and β and $\epsilon \in R(T)$, then $\min\{\alpha\beta\epsilon, 1\} \in R(S)$.

Hence, if S *L*-reduces to T and vice-versa with $\alpha = \beta = 1$, we say that S and T are *L-equivalent.* The following result is a direct consequence of above remarks.

Proposition. *Let S and T be NP-complete optimization problems such that there exists an L-reduction from S to T. If there exists a PTAS for T then there exists a PTAS for S.* ❑

Therefore, there exists a PTAS for a *MAXSNP*-complete problem if and only if there exist PTAS for every problem in *MAXSNP*. The MAXINDEPENDENTSET and MAXCLIQUE problems are *L*-equivalent (graph complement). Berman and Schnitger [3] showed the nonexistence of polynomial-time bounded-error approximation algorithms solving the MAXINDEPENDENTSET problem (and therefore MAXCLIQUE problem) unless there exists a PTAS for MAX2SAT problem. Recently, Arora *et al.* answered these questions showing:

Theorem 1 (Arora *et al.* [1], 1992). *If there exists a PTAS for a MAXSNP-complete problem then P = NP.* ❑

Summarizing the discussion in this section: if *P* $\neq$ *NP* then the problems MAXINDEPENDENTSET and MAXCLIQUE do not allow any polynomial-time bounded-error approximation algorithm.

3 The Selection Problem

In this section, the selection problem and other related problems are defined. We prove the *NP*-completeness of these problems, *L*-reductions between them and, under the assumption *P* $\neq$ *NP*, the nonexistence of PTAS for them is shown. We begin with the definition of the selection problem.

Problem (SEL). *The selection problem.*

Given: *Disjoint sets $A_1, \ldots, A_k, B_1, \ldots, B_l$, a function $f : (A_1 \cup \cdots \cup A_k) \times (B_1 \cup \cdots \cup B_l) \to \mathbb{N}$ computable in polynomial-time and $q \in \mathbb{N}$.*

Question: *Are there selection sets A and B, i.e. $A \subseteq A_1 \cup \cdots \cup A_k$, $|A \cap A_i| \leq 1$ ($i = 1, \ldots, k$), $B \subseteq B_1 \cup \cdots \cup B_l$ and $|B \cap B_j| \leq 1$ ($j = 1, \ldots, l$), such that $\sum_{a \in A, b \in B} f(a, b) \geq q$?*

Next, an example for the SEL problem is presented. Consider the Figure 1(a), where the edges represent the values of the function f that are greater than 0. The sets $A = \{a_{1,3}, a_{2,1}, a_{3,1}\}$ and $B = \{b_{1,2}, b_{2,2}\}$ are a feasible solution for $q \leq 3$, i.e. $\sum_{a \in A, b \in B} f(a, b) = 3$.

Let us give an equivalent graph problem. A *k-dimensional graph* $G = (V_1, \ldots, V_k, E)$ is a graph with k disjoint vertex sets $V_1, \ldots, V_k$, where an edge in E is a k-tuple $(v_1, \ldots, v_k)$ with $v_i \in V_i$, e.g. the corresponding two-dimensional graph is a regular bipartite graph. If G is a weighted graph with edges set E, then let $c(e)$ be the cost of the edge $e \in E$. In what follows we assume that the graphs are undirected without loops.

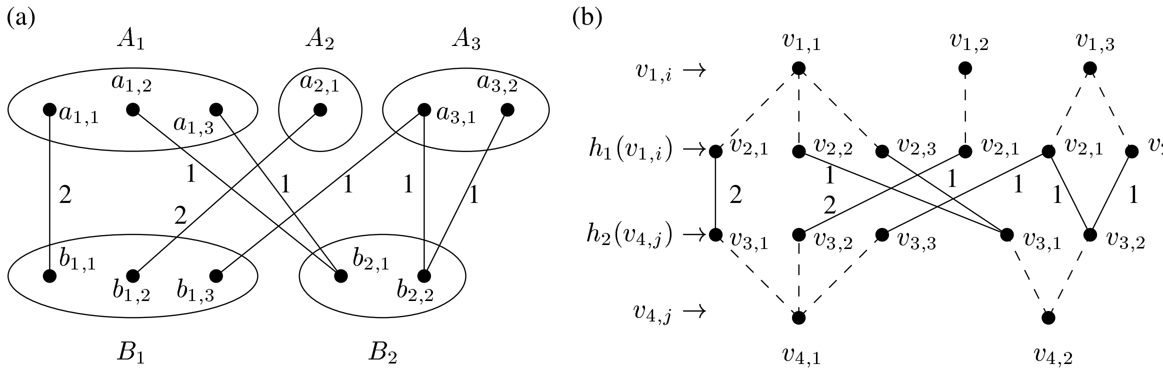

Figure 1: Instances for (a) SEL problem and (b) SWF problem

Problem (SWF)**.** *The switched flow problem.*

Given: *A 4-dimensional weighted graph $G = (V_1, \dots, V_4, E)$, where the edge costs are positive integer (i.e. $c(e) \in \mathbb{N}$ for $e \in E$), and $q \in \mathbb{N}$.*

Question: *Are there functions $h_1 : V_1 \to V_2$ and $h_2 : V_4 \to V_3$ satisfying the following inequality: $\sum_{v_1 \in V_1, v_4 \in V_4} c\left((v_1, h_1(v_1), h_2(v_4), v_4)\right) \geq q$?*

The graph of Figure 1(b) can be considered as an instance for SWF problem, where $v_{i,j} \in V_i$. To represent the graph correctly, we draw some vertices of partitions V_2 and V_3 more than one time, e.g. the vertex $v_{2,1}$ in the partition V_2. Hence, each path from a vertex in V_1 to a vertex in V_4 represents exactly one edge in the graph. Observe that each solid line is a part of exactly one edge but a dashed line can be a part for many edges. If the values $h_1(v_{1,2}) = v_{2,1}$, $h_1(v_{1,3}) = v_{2,1}$, $h_2(v_{4,1}) = v_{3,2}$, and $h_2(v_{4,2}) = v_{3,2}$ are chosen for functions h_1 and h_2 ($h_1(v_{1,2})$ anyone), then we have a solution for $q \leq 3$, i.e. $\sum_{v_1 \in V_1, v_4 \in V_4} c\left((v_1, h_1(v_1), h_2(v_4), v_4)\right) = 3$. Next, we show that the optimization versions of SEL and SWF problems are L-equivalent.

Proposition 2. *The* MAXSEL *and* MAXSWF *problems are L-equivalent.*

Proof. We prove one direction, since the other follows in a similar way. Consider the following feasible instance x for MAXSEL problem: disjoint sets $A_1, \dots, A_k, B_1, \dots, B_l$, a function $f : (A_1 \cup \dots \cup A_k) \times (B_1 \cup \dots \cup B_l) \to \mathbb{N}$ computable in polynomial-time and $q \in \mathbb{N}$. Let $A_i = \{a_{i,1}, \dots, a_{i,r_i}\}$ for $i = 1, \dots, k$ and $r_i \in \mathbb{N}$, and let $B_j = \left\{b_{j,1}, \dots, b_{j,s_j}\right\}$ for $j = 1, \dots, l$ and $s_j \in \mathbb{N}$. Construct the following instance $G = (V_1, \dots, V_4, E)$ for MAXSWF problem (compare the Figures 1(a) and 1(b)):

1. $V_1 = \{v_{1,i} : 1 \leq i \leq k\}, \quad V_2 = \{v_{2,i} : 1 \leq i \leq \max_{1 \leq j \leq k} |A_j|\}$
   $V_4 = \{v_{4,i} : 1 \leq i \leq l\}, \quad V_3 = \{v_{3,i} : 1 \leq i \leq \max_{1 \leq j \leq l} |B_j|\}$
2. $E = \{(v_{1,i}, v_{2,j}, v_{3,p}, v_{4,q}) : 1 \leq i \leq k \wedge 1 \leq j \leq |A_i| \wedge 1 \leq q \leq l \wedge 1 \leq p \leq |B_q|\}$,
   where the cost of an edge $(v_{1,i}, v_{2,j}, v_{3,p}, v_{4,q}) \in E$ is $f(a_{i,j}, b_{q,p})$.

Thus, s is a feasible solution for MAXSWF with functions h_1 and h_2 if and only if s is a feasible solution for MAXSEL with selection sets $A = \{a_{i,j} : h_1(v_{1,i}) = v_{2,j}\}$ and $B = \{b_{i,j} : h_2(v_{4,i}) = v_{3,j}\}$, i.e. $\mathrm{OPT}_{\mathrm{MAXSEL}}(x) = \mathrm{OPT}_{\mathrm{MAXSWF}}(G)$. Since f is computable in polynomial-time, we have the desired result. ❑

Consider the following restrictions to the definitions of problems SEL and SWF, where the item (a) refers to the SEL problem and the item (b) refers to the SWF problem:

1. (a) the range of f is restricted to $\{0, 1\}$, i.e. $f : (A_1 \cup \dots \cup A_k) \times (B_1 \cup \dots \cup B_l) \to \{0, 1\}$

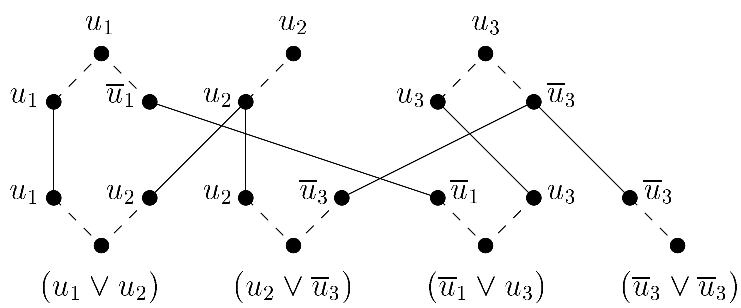

Figure 2: The MAX2SAT problem *L*-reduces to the $\text{MAXSWF}_{\text{p}}^{1}$ problem

  (b) the edge costs are restricted to $\{0,1\}$, i.e. $c(e) \in \{0,1\}$ for $e \in E$

2. there exists a premise with

  (a) $\max_{A,B} \sum_{a \in A, b \in B} f(a,b) \leq \max\Big\{k, l, \max_{1 \leq i \leq k} |A_i|, \max_{1 \leq j \leq l} |B_j|\Big\}$

  (b) $\max_{h_1,h_2} \sum_{v_1 \in V_1, v_4 \in V_4} c((v_1, h_1(v_1), h_2(v_4), v_4)) \leq \max_{1 \leq i \leq 4} |V_i|$

The examples of Figures 1(a) and 1(b) satisfy respectively the premises (a) and (b). Considering these restrictions, we can define following variations of SEL and SWF problems.

Problems (SEL_{p} and SWF_{p}, $\text{SEL}_{\text{p}}^{1}$ and $\text{SWF}_{\text{p}}^{1}$). *Variations of* SEL *and* SWF *problems.*

- *Define* SEL_p *(*SWF_p*, respectively) as the* SEL *problem (*SWF *problem, respectively) but the premise (a) (the premise (b), respectively) being satisfied;*
- *Define* SEL_p^1 *(*SWF_p^1*, respectively) as the* SEL_p *problem (*SWF_p *problem, respectively) but the range of f (the edge costs, respectively) being restricted to* $\{0,1\}$.

The abbreviation ALLSEL is used for the problems SEL, SEL_{p} and $\text{SEL}_{\text{p}}^{1}$, and the abbreviation ALLSWF is used for the problems SWF, SWF_{p} and $\text{SWF}_{\text{p}}^{1}$.

In the same way we proved the Proposition 2, we can show some *L*-equivalences between the problems defined above.

Proposition 3. *The following problems are L-equivalent:*

i) MAXSEL_p *and* MAXSWF_p*;*

ii) MAXSEL_p^1 *and* MAXSWF_p^1. ❑

Obviously, every problem containing a restriction can be *L*-reduced to its respectively problem without this restriction.

Proposition 4. *The following L-reductions with $\alpha = \beta = 1$ are valid:*

i) from MAXSEL_p^1 *to* MAXSEL_p*, and from* MAXSWF_p^1 *to* MAXSWF_p*;*

ii) from MAXSEL_p *to* MAXSEL*, and from* MAXSWF_p *to* MAXSWF. ❑

By the chain of *L*-reductions proved above we can show that the problems in *MAXSNP* are *L*-reducible to the ALLMAXSEL and ALLMAXSWF problems by giving an *L*-reduction from a *MAXSNP*-complete problem to the $\text{MAXSWF}_{\text{p}}^{1}$ problem, since *MAXSNP* is closed under *L*-reductions and this reduction is transitive. This will be done in Lemma 5, where we show an *L*-reduction from MAX2SAT problem to the $\text{MAXSWF}_{\text{p}}^{1}$ problem. The idea this reduction follows: The function h_1 is used to assign the boolean variables with values "true" or "false" and the function h_2 is used to verify which clauses are satisfied with this assignment, e.g. the graph of Figure 2 (the solid lines have cost 1) is an instance for $\text{MAXSWF}_{\text{p}}^{1}$ problem corresponding to the 2SAT boolean expression $(u_1 \vee u_2) \wedge (u_2 \vee \overline{u}_3) \wedge (\overline{u}_1 \vee u_3) \wedge (\overline{u}_3 \vee \overline{u}_3)$.

Lemma 5. MAX2SAT *L-reduces to* MAXSWF_p^1 *with* $\alpha = \beta = 1$.

Proof. Let Q be a 2SAT boolean expression with clauses $K = \{K_1, \ldots, K_n\}$ and boolean variables $U = \{u_1, \ldots, u_m\}$, and define $\varphi(u) =_{\mathrm{df}} \varphi(\overline{u}) =_{\mathrm{df}} u$ for $u \in U$. The reduction function maps such instance Q to the following graph $G = (V_1, V_2, V_3, V_4, E)$

- $V_1 = U$, $V_2 = V_3 = (U \cup \{\overline{u} : u \in U\})$ and $V_4 = K$.
  By definition, the sets V_i are disjoint. This is done by introducing indexes.
- $E = \{(v_1, v_2, v_3, v_4) : 1 \leq i \leq 4 \wedge v_i \in V_i \wedge v_4 = (l_1 \vee l_2) \wedge \bigvee_{j=1,2}(v_2 = v_3 = l_j \wedge \varphi(l_j) = v_1)\}$ with each edge having cost 1.

If an edge connects the vertices $u \in V_1$ and $w \in V_2$ then $u = \varphi(w)$, i.e. any function h_1 can choose for a variable of U either "true" ($h_1(u) = u$) or "false" ($h_1(u) = \overline{u}$). On the other hand, there exists for each vertex in V_4 at most two incident edges (a vertex $(u \vee u)$ or $(\overline{u} \vee \overline{u})$ has exactly one incident edge) and the function h_2 can select exactly one of them. Therefore, $\max_{h_1,h_2} \sum_{v_1 \in V_1, v_4 \in V_4} c((v_1, h_1(v_1), h_2(v_4), v_4)) \leq |V_4|$ and the premise (b) is satisfied. Hence, s is a feasible solution for MAX2SAT(Q) with an assignment for the boolean variables in U if and only if s is a feasible solution for $\text{MAXSWF}_\mathrm{p}^1(G)$ with

$$h_1(u)_{u \in U} = \begin{cases} u, & \text{if } u \text{ is assigned with "true"} \\ \overline{u}, & \text{if } u \text{ is assigned with "false"} \end{cases}$$

and h_2 determining the number of clauses satisfying the assignment of h_1, i.e. for each clause $(l_1 \vee l_2)$ we verify if $h_1(\varphi(l_1)) = l_1$. In the affirmative case, $h_2((l_1 \vee l_2)) = l_1$ is chosen, otherwise we choose $h_2((l_1 \vee l_2)) = l_2$. Therefore, $\mathrm{OPT}_{\text{MAXSWF}_\mathrm{p}^1}(G) = \mathrm{OPT}_{\text{MAX2SAT}}(Q)$, which completes the proof. ❑

The following corollary summarizes our discussion on the ALLSEL and ALLSWF problems.

Corollary 6. *The* ALLSEL *and* ALLSWF *problems are NP-complete, and if there exists a PTAS for one these (optimization) problems then P = NP.*

Proof. To show that the ALLSEL and ALLSWF problems are *NP*-complete, it is enough to prove that they belong to *NP*, since MAX2SAT is an *NP*-complete problem (see Garey and Johnson [8]) and it *L*-reduces to $\text{MAXSWF}_\mathrm{p}^1$ (Lemma 5). Guessing the sets A and B for the ALLSEL problems and guessing the functions h_1 and h_2 for the ALLSWF problems we get that these problems are in *NP*. The other part of the corollary follows by Theorem 1 and Lemma 5, since the MAX2SAT problem is *MAXSNP*-complete. ❑

Next, we show that the MAXSWF_p problem can be *L*-reduced to the MAXINDEPENDENTSET problem.

Proposition 7. MAXSWF_p *L-reduces to* MAXINDEPENDENTSET *with* $\alpha = \beta = 1$.

Proof. Let $G = (V_1, V_2, V_3, V_4, E)$ be an instance for MAXSWF_p problem, $n =_{\mathrm{df}} \max_{1 \leq i \leq 4} |V_i|$ and $m =_{\mathrm{df}} |E|$. Due premise (b) of MAXSWF_p problem, it follows that $0 \leq c(e) \leq n$ for all $e \in E$. The reduction function maps such graph G to the following graph $G' = (V', E')$ (see Figure 3)

- $V' = \{\langle e, i \rangle : e \in E \wedge 1 \leq i \leq c(e)\}$
- $E' = \{(\langle v, i \rangle, \langle w, j \rangle) : \langle v, i \rangle, \langle w, j \rangle \in V' \wedge v = (v_1, v_2, v_3, v_4) \wedge w = (w_1, w_2, w_3, w_4)$
  $\wedge((v_1 = w_1 \wedge v_2 \neq w_2) \vee (v_4 = w_4 \wedge v_3 \neq w_3))\}$

If $e \in E$ and $c(e) = l \geq 2$, then for $1 \leq i < j \leq l$, the vertices $\langle e, i \rangle$ and $\langle e, j \rangle$ have the same neighbors in G' and $(\langle e, i \rangle, \langle e, j \rangle) \notin E'$. I.e. if I' is an INDEPENDENTSET in G' and $\langle e, i \rangle \in I'$, then $I' \cup \{\langle e, j \rangle\}$ is also an INDEPENDENTSET in G'. Let I' be a maximal INDEPENDENTSET in G'. Hence, if $\langle e, i \rangle \in I'$ then $\langle e, j \rangle \in I'$ for all $j \in \{1, \ldots, c(e)\}$. Therefore, I' is an INDEPENDENTSET in G' if and only if there are functions h_1 and h_2 such that for all $e = (v_1, v_2, v_3, v_4) \in E$

$$\forall i \, (1 \leq i \leq c(e) \rightarrow \langle e, i \rangle \in I') \iff h_1(v_1) = v_2 \text{ and } h_2(v_4) = v_3.$$

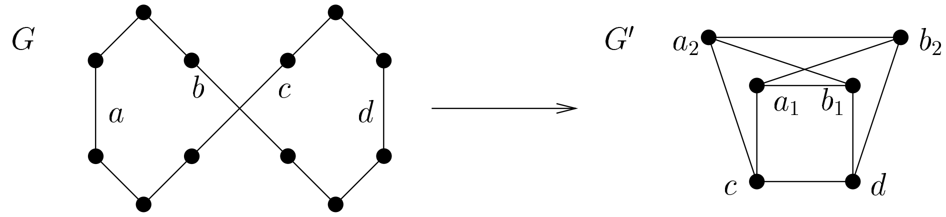

Figure 3: Graph Reductions from G to G'. Let $c(a) = c(b) = 2$ and $c(c) = c(d) = 1$.

Hence, $|I'| = \sum_{v_1 \in V_1, v_4 \in V_4} c((v_1, h_1(v_1), h_2(v_4), v_4))$. Since $|V'| \in O(nm)$ and $|E'| \in O((nm)^2)$, we get the desire result. ❑

4 Non-Approximability

In this section, the main result is presented. A *two-prover one-round interactive proof system* (*MIP*$(2,1)$) can be defined as follows (see [2, 6]):

Definition 8. $A \in \mathit{MIP}(2,1) \iff$ *there are functions f, g_1, g_2 computable in polynomial-time and polynomials p, q such that f is a 0-1-function and*

$$x \in A \rightarrow \exists h_1 \exists h_2 \left(\text{prob}\{z : |z| = p(|x|) \wedge f(x, z, h_1(g_1(x,z)), h_2(g_2(x,z))) = 1\} = 1\right)$$
$$x \notin A \rightarrow \forall h_1 \forall h_2 \left(\text{prob}\{z : |z| = p(|x|) \wedge f(x, z, h_1(g_1(x,z)), h_2(g_2(x,z))) = 1\} \le 2^{-|x|}\right)$$

where z is randomly chosen under uniform distribution, i.e. with probability $2^{-p(|x|)}$, and $|h_i(u)| \le q(|u|)$ for $i = 1, 2$.

The previous definition can be interpreted as follows: for $i = 1, 2$, $g_i(x, z)$ is the question of the verifier to the prover i, $h_i(g_i(x,z))$ is the answer of the prover i on question $g_i(x, z)$ and

$f(x, z, h_1(g_1(x,z)), h_2(g_2(x,z))) = 1 \iff$ the verifier on input x, random tape z and answers $h_1(g_1(x,z))$ and $h_2(g_2(x,z))$ of the provers accepts the input.

A function g is *computable in exponential-time* if and only if it is computable in time $2^{q(n)}$ for some polynomial q. Next, we use the MaxSwF_p problem to recognize the languages that can be accepted by *MIP*$(2,1)$ protocols.

Lemma 9. $A \in \mathit{MIP}(2,1) \implies$ *there exists a function g computable in exponential-time and a polynomial p such that*

$$x \in A \iff \text{OPT}_{\text{MaxSwF}_p}(g(x)) = 2^{p(|x|)}$$
$$x \notin A \iff \text{OPT}_{\text{MaxSwF}_p}(g(x)) \le 2^{p(|x|)-|x|}.$$

Proof. Consider the functions g_1, g_2, f and the polynomials p, q from Definition 8 and let $|h_i(u)| \le q(|u|)$ for $i = 1, 2$. Now, define a function s as follows

$$s(x, v_1, v_2, v_3, v_4) =_{\text{df}} |\{z : |z| = p(|x|) \wedge g_1(x,z) = v_1 \wedge g_2(x,z) = v_2 \wedge f(x, z, v_3, v_4) = 1\}|$$

Since g_1, g_2, f are computable in polynomial time and there are $2^{p(|x|)}$ different z's, then the function s can be computable in exponential time and the lengths of g_1 and g_2 in the definition of s are bounded by some polynomial r depending only on the length of x. Without loss of the generality, assume $p(n) \leq r(n)$. Furthermore, for any functions h_1, h_2

$$\sum_{|z|=p(|x|)} f(x, z, h_1(g_1(x, z)), h_2(g_2(x, z))) = \sum_{|v_1|,|v_2| \leq r(|x|)} s(x, v_1, v_2, h_1(v_1), h_2(v_2))$$

Therefore, by Definition 8

$$\begin{aligned} &\forall h_1 \forall h_2 \left(0 \leq \sum\nolimits_{|v_1|,|v_2| \leq r(|x|)} s(x, v_1, v_2, h_1(v_1), h_2(v_2)) \leq 2^{p(|x|)}\right) \\ x \in A \rightarrow &\exists h_1 \exists h_2 \left(\sum\nolimits_{|v_1|,|v_2| \leq r(|x|)} s(x, v_1, v_2, h_1(v_1), h_2(v_2)) = 2^{p(|x|)}\right) \\ x \notin A \rightarrow &\forall h_1 \forall h_2 \left(\sum\nolimits_{|v_1|,|v_2| \leq r(|x|)} s(x, v_1, v_2, h_1(v_1), h_2(v_2)) \leq 2^{p(|x|)-|x|}\right) \end{aligned} \tag{1}$$

Now, let $g(x)$ be the graph $G = (V_1, \ldots, V_4, E)$ such that

- $V_i = \{v : |v| \leq r(|x|)\}$ for $i = 1, \ldots, 4$;
- $E = \{(v_1, v_2, v_3, v_4) : 1 \leq i \leq 4 \wedge v_i \in V_i\}$
  where $c((v_1, v_2, v_3, v_4)) = s(x, v_1, v_4, v_2, v_3)$ for all edge $(v_1, v_2, v_3, v_4) \in E$

The function g is computable in exponential time, since s can be computable in exponential time and $|V_i| = 2^{r(|x|)+1} - 1$ ($i = 1, \ldots, 4$). Furthermore, for any functions h_1, h_2

$$\sum_{|v_1|,|v_4| \leq r(|x|)} c\left((v_1, h_1(v_1), h_2(v_4), v_4)\right) = \sum_{|v_1|,|v_4| \leq r(|x|)} s(x, v_1, v_4, h_1(v_1), h_2(v_4)).$$

Hence, the assumption with the desired premise (b) for $g(x)$ follows by (1). ❑

As a consequence of previous lemma, if there exists a polynomial-time bounded-error approximation algorithm solving the MAXSWF$_\text{p}$ problem then each language that can be accepted by an *MIP*$(2, 1)$ protocol can also be accepted by a deterministic exponential-time Turing machine.

Corollary 10. *If there is a polynomial-time bounded-error approximation algorithm solving the* MAXSWF$_p$ *problem, then MIP*$(2, 1) \subseteq$ *EXPTIME.*

Proof. Let M be a polynomial-time ϵ-approximation algorithm solving the MAXSWF$_\text{p}$ problem with $\epsilon < 1$ and $A \in$ *MIP*$(2, 1)$. Since ϵ is a constant between 0 and 1, we consider only the instances for A whose lengths are greater than $\log(1/(1 - \epsilon))$. Let x be an instance for A such that $|x| > \log(1/(1 - \epsilon))$. Consider the function g and the polynomial p from Lemma 9. Thus, for $x \in A$ we have $M(g(x)) \geq (1 - \epsilon)2^{p(|x|)} > 2^{p(|x|)-|x|}$, i.e.

$$x \in A \iff M(g(x)) > 2^{p(|x|)-|x|}$$

Since g is computable in exponential time, then $A \in$ *EXPTIME*. ❑

However, Feige and Lovász [7] proved *MIP*$(2, 1)$ = *NEXPTIME*. Our main result then follows in a trivial way (see Corollary 10 and the L-reductions from Section 3).

Theorem 11. *If there exists a polynomial-time bounded-error approximation algorithm solving one of the problems* MAXSWF$_p$, MAXSWF, MAXSEL$_p$ *or* MAXSEL, *then EXPTIME = NEXPTIME.* ❑

Acknowledgments. I would like to thank Prof. Dr. Klaus W. Wagner, Würzburg, for calling my attention to this problem and encouraging my pursuit of it, and Heribert Vollmer, Würzburg, for some helpful hints.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *33rd IEEE Symposium on Foundations of Computer Science*, pages 14–23, 1992.

[2] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *20th ACM Symposium on the Theory of Computing*, pages 113–131, 1988.

[3] P. Berman and G. Schnitger. On the complexity of approximating the independent set problem. In *6th Annual Symposium on Theoretical Aspects of Computer Science*, volume 349 of *Lecture Notes in Computer Science*, pages 256–267. Springer Verlag, 1989.

[4] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. Karp, editor, *Complexity of Computation*, volume 7, pages 43–73. SIAM-AMS Proceedings, 1974.

[5] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-compete. In *32nd IEEE Symposium on Foundations of Computer Science*, pages 2–12, 1991.

[6] U. Feige and J. Kilian. Two-prover protocols – low error at affordable rates. In *26th ACM Symposium on the Theory of Computing*, pages 172–183, 1994.

[7] U. Feige and L. Lovász. Two-prover one-round proof systems: Their power and their problems. In *24th ACM Symposium on the Theory of Computing*, pages 733–744, 1992.

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[9] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company Inc, 1994.

[10] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43:425–440, 1991.