On the Efficiency of Polynomial Time Approximation Schemes

Marco Cesati* Luca Trevisan†

January 8, 1997

Abstract

A polynomial time approximation scheme (PTAS) for an optimization problem A is an algorithm that on input an instance of A and $\epsilon > 0$ finds a $(1+\epsilon)$-approximate solution in time that is polynomial for each fixed ϵ. Typical running times are $n^{O(1/\epsilon)}$ or $2^{1/\epsilon^{O(1)}}n$.

While algorithms of the former kind tend to be impractical, the latter ones are more interesting. In several cases, the development of algorithms of the second type required considerably new (and sometimes harder) techniques. For some interesting problems (including Euclidean TSP) only an $n^{O(1/\epsilon)}$ approximation scheme is known.

Under likely assumptions, we prove that for some problems (including natural ones) there cannot be approximation schemes running in time $f(1/\epsilon)n^{O(1)}$, no matter how fast function f grows.

Our result relies on a connection with Parameterized Complexity Theory. We show that this connection is necessary.

Keywords: Computational Complexity, Approximation Algorithms, Parameterized Complexity.

1 Introduction

Approximation algorithms are a natural and effective way to deal with the intractability of optimization problems. For $r > 1$, we say that an algorithm is r-approximate if it computes solutions whose cost is within the multiplicative factor r from the optimum. For stronlgy NP-hard optimization problems, the best possible approximation algorithm can find $(1+\epsilon)$-approximate solutions in time polynomial in the size of the input but (at least) exponential in ϵ. Such an algorithm is called a polynomial-time approximation scheme (in short PTAS). We call PTAS the class of optimization problems that admit a PTAS.

Typical running times of a PTAS are $n^{O(1/\epsilon)}$ or $2^{O(1/\epsilon)}n$. While a PTAS of the former type becomes useless even for moderate values of ϵ and n, a PTAS of the latter type can return in a reasonable amount of time a good approximation for an enormous instance. This observation motivates our definition of *efficient* PTAS as an approximation scheme running in time $f(\epsilon)n^c$, where f is an arbitrary function and c is a constant independent of ϵ. We call EPTAS the class of problems adimitting an efficient PTAS.

*Dipartimento di Scienze dell'Informazione, Università di Roma "La Sapienza". Via Salaria 113, I-00198 Roma, Italy. Email `cesati@dsi.uniroma1.it`.

†Centre Universitaire d'Informatique, Université de Genève, Rue Général-Dufour 24, CH-1211, Genève, Switzerland. Email `trevisan@cui.unige.ch`.

In general, bounding the running time by a polynomial whose *degree* does not depend on ϵ is an implicit but widely followed rule in the design of approximation schemes. We know of several cases where a EPTAS has been designed for a problem for which a PTAS was already known; the more efficient algorithm was often quite different from the previous one.

A PTAS running in time $n^{O(1/\epsilon)}$ for the problem of scheduling processes on identical machines was given by Hochbaum and Shmoys in [15]. Very recently, the same authors gave a $O(f(\epsilon)+n)$ algorithm [16], where f is doubly exponential. Arora, Karger, and Karpinski [3] present PTAS's for dense constraint satisfaction problems (including dense MAX CUT and dense MAX 3SAT) running in time $n^{O(1/\epsilon)}$. For dense MAX CUT, a randomized algorithm by de la Vega [9] runs in time $O(2^{\mathrm{poly}(1/\epsilon)}n^c)$. Similar performances are obtained by Frieze and Kannan [13] using an algorithmic version of Szemeredy's regularity lemma. A more efficient algorithm is given by Goldreich et al. [14]. It is still open whether there exists an EPTAS for dense MAX 3SAT. Khanna and Motwani [17] develop PTAS's for a large variety of planar constraint satisfcation problems; the running time is $n^{O(1/\epsilon)}$. For some of those problems, Crescenzi and Trevisan [8] improve the running time to $2^{\mathrm{poly}(1/\epsilon)}n$. Perhaps the most interesting problem for which a PTAS exists is the Euclidean TSP. Arora's PTAS [2] runs in time roughly $n^{30/\epsilon}$. Arora formulates an algorithmic problem that, if solved, would imply an EPTAS for Euclidean TSP.

We address the issue of whether all PTAS problems admit an efficient PTAS. We show that this problem is connected to parameterized complexity. This research area is a new, powerful framework with which to address the different parameterized behaviour of many computational problems. Almost all natural problems have instances consisting of at least two logical items; many NP-complete problems admit "efficient" algorithms for small values of one item (the *parameter*). Downey and Fellows have introduced [10, 11, 12] the notion of "fixed parameter tractable problem": a parameterized problem admitting a solving algorithm whose running time, for any fixed value of the parameter, is bounded by a polynomial of constant degree. The class of all fixed parameter tractable problems is called FPT. Since a direct proof that some NP-complete problem is not fixed parameter tractable would imply that $\mathsf{P} \neq \mathsf{NP}$, Downey and Fellows have then introduced a "parameterized reduction" and a hierarchy of classes containing likely fixed parameter intractable problems (the so-called *W hierarchy*): $\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \ldots \subseteq \mathsf{W}[\mathsf{P}] \subseteq \mathsf{SP}$. Moreover, many very natural parameterized problems have been shown to be complete for several levels of the W hierarchy, thus giving evidence that this framework is a useful tool for proving (likely) parameterized intractability.

Our main results are:

1. A two-way relation between parameterized complexity and the $\mathsf{EPTAS} \stackrel{?}{=} \mathsf{PTAS}$ question. Namely, if $\mathsf{FPT} = \mathsf{SP}$, then $\mathsf{EPTAS} = \mathsf{PTAS}$, and if $\mathsf{FPT} \neq \mathsf{W}[\mathsf{P}]$ then $\mathsf{EPTAS} \neq \mathsf{PTAS}$. We prove the former result with the usual argument of *prefix languages*. The weak reverse connection makes use of a mildly involved scaling argument.

2. A natural problem mentioned in [17] is not in EPTAS unless $\mathsf{FPT} = \mathsf{W}[1]$. This result requires a proof of $\mathsf{W}[1]$-completeness of the induced language. The main new ingredient of this proof is a *cross-over gadget* for multivalued satisfiability problems.

2 Definitions

Definition 1 (Parameterized problem) *A* parameterized problem *is a set $L \subset \Sigma^* \times \Sigma^*$, where Σ is a fixed alphabet. In an instance $\langle x, y\rangle$ of a parameterized problem, y encodes the* parameters *and is usually an integer (or a set of integers).*

Definition 2 (SP, FPT) *A parameterized problem L is in* SP *if there is a function g and an algorithm T such that T determines if $\langle x, k\rangle \in L$ in time bounded by $|x|^{g(k)}$. L is in* FPT *if there is a constant c, a function f and an algorithm T such that T determines if $\langle x, k\rangle \in L$ in time bounded by $f(k)\,|x|^c$.*

There exists an hierarchy $\mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \ldots \subseteq \mathsf{W}[t] \subseteq \ldots$ of classes that are intermediate between FPT and SP. On top of it, lies the class W[P], still contained in SP (see [10] for definitions). There is some evidence that these classes do not collapse to FPT. In particular, if FPT = W[1] then 3SAT can be solved in $2^{o(n)}$ time [1] (where n is the number of variables). Natural complete problems are known for most of these classes under FPT-preserving reductions.

An *optimization problem* A consists of three objects: (1) the set I of instances, (2) for any instance $x \in I$, a set $\mathsf{sol}(x)$ of solutions, and (3) for any instance $x \in I$ and for any solution $y \in \mathsf{sol}(x)$, a measure $\mathsf{m}(x, y)$. The goal of an optimization problem is, given an instance x, to find an optimum solution y, that is, a solution whose measure is maximum or minimum depending on whether the problem is a maximization or a minimization one. For a formal definition of NP optimization (NPO) problems see [4]. In the following opt will denote the function that maps an instance x into the measure of an optimum solution.

Let A be an NPO problem. For any instance x and for any solution $y \in \mathsf{sol}(x)$, the *performance ratio* of y with respect to x is defined as

$$R(x, y) = \max\left\{\frac{\mathsf{opt}(x)}{\mathsf{m}(x, y)}, \frac{\mathsf{m}(x, y)}{\mathsf{opt}(x)}\right\}.$$

Observe that the performance ratio is always a number greater than or equal to 1 and is as close to 1 as the solution is close to an optimum solution.

Let $r > 1$ be a fixed constant; we say that an algorithm T for an optimization problem A is *r-approximate* if, for any instance x of size n, the performance ratio of the feasible solution $T(x)$ with respect to x is at most r.

Definition 3 (PTAS, EPTAS) *A polynomial time approximation scheme (PTAS) for an* NPO *problem A is an algorithm $T(\cdot, \cdot)$ that, on input a rational $r > 1$ and an instance x of A, returns an r-approximate feasible solution $T(x, r)$; furthermore, the running time of $T(x, r)$ is polynomial in $|x|$ for each fixed r. A PTAS is said to be an efficient PTAS (EPTAS) if its running time is bounded by $f(r)|x|^c$ where f is an arbitrary function and c is a fixed constant. We call* PTAS *(resp.* EPTAS*) the class of optimization problems admitting a PTAS (resp. EPTAS).*

Definition 4 (Induced language) *For a maximization (resp. minimization) problem A, the induced language is the parameterized problem consisting of all the pairs $\langle x, k\rangle$ where x is an instance of A and $\mathsf{opt}(x) \geq k$ (resp. $\mathsf{opt}(x) \leq k$).*

3 Necessity of Parameterized Complexity

Theorem 5 *If* FPT = SP *then* EPTAS = PTAS.

Proof: Let A be a PTAS problem, then an approximation scheme $T(x, r)$ exists whose running time is polynmomial for every fixed $r > 1$; let us consider the following paramterized language $L_T = \{\langle (x, y), k\rangle : \exists z. yz = T(x, 1 + 1/k)\}$. Clearly, $L_T \in$ SP, and thus $L_T \in$ FPT. From an FPT algorithm for L_T it is easy to construct an algorithm T' that simulates T and whose running time is bounded by $f(\lceil 1/(r-1)\rceil)|x|^\alpha$ for certain function f and constant α. □

The previous result means that our question can be explored only in the realm of fixed parameter complexity, since $\mathsf{FPT} = \mathsf{SP}$ is not known to imply any collapse in standard complexity theory, say, the collapse of the polynomial hierarchy

4 A Natural Hard Problems

Lemma 6 *If* $A \in \mathsf{EPTAS}$ *then* $L_A \in \mathsf{FPT}$.

Proof: In order to decide whether $\langle x, k \rangle \in L_A$ it is sufficient to compute a $(1+1/2k)$-approximate solution for x. □

In the following we will consider the multivalued (planar) Max KCSP and Max GSAT problems.

For a fixed $K \geq 1$, the multivalued Max KCSP problem is defined as follows: the instance contains a number n and a set of K-ary constraints over variables $x_1, \ldots, x_N$. A constraint is specified by listing its satisfying assignments. A solution is an assigment of values of $\{0, \ldots, n-1\}$ to the variables. The measure of a solution is the number of constraints it satisfies, and the goal is to maximize such number.

The multivalued Max GSAT problem is similar, but this time a constraint is a DNF expression (a disjunction of conjunctions of literals) where a literal is an equation of the form $[x_i = a]$. A standard restriction of the Max GSAT problem is, for fixed $B \geq 1$, the Max GSATB problem, where a constraint is a DNF expression such that the lenght of the disjunctions is at most B. Note that, for any fixed K, multivalued Max KCSP is a special case of multivalued Max GSATK.

Given an instance of multivalued Max KCSP or Max GSAT, its *incidence graph* is defined as follows: there is a node for any variable and one for any constraints; for any constraint, there are edges between the node representing the constraint and all the nodes representing the variables occurring in it. An instance is said to be planar if its incidence graph is planar. Khanna and Motwani [17] prove that planar multivalued Max GSAT admits a PTAS.

Multivalued KCSP (resp. GSATB, GSAT) is the parameterized problem where the instance is a multivalued Max KCSP (resp. Max GSATB, Max GSAT) instance, the parameters are the number of variables and the number of constraints, and the question is whether the instance is satisfiable.

Lemma 7 *Multivalued* 2CSP *is* $\mathsf{W}[1]$-*hard.*

Proof: We give parameterized reduction from the CLIQUE problem, that is $\mathsf{W}[1]$-complete [12]. Given a graph $G = (V, E)$ with m edges and n nodes, and an integer k, we show how to construct, in time $O((m+n)k^2)$, an instance ϕ of multivalued 2CSP with k variables and $k(k-1)/2$ constraints such that the formula is satisfiable iff the graph has a k-clique. The hardness of the problem will follow.

Let us assume that $V = \{0, \ldots n-1\}$. ϕ has k variables $x_1, \ldots, x_k$ that take value over $0, \ldots, n-1$. x_i is supposed to be the i-th node of a k-clique in G. There is a constraint C_{ij} for any unordered pair $i, j \in \{1, \ldots, k\}$. The constraint enforces that x_i and x_j are endpoints of some edge of G (and thus, in particular, they are different). The constraint is

$$\bigvee_{(u,v) \in E} ([x_i = u] \wedge [x_j = v]) \vee ([x_i = v] \wedge [x_j = u]).$$

It is clear that ϕ is satisfiable iff G has a k-clique. □

The next lemma shows that the previous hardness result cannot be pushed higher in the W hierarchy, because even a much more general problem is in W[1].

Lemma 8 *Multivalued* GSAT *belongs to* W[1].

Proof: We will show a parameterized reduction from Multivalued GSAT (with parameters the number of variables k_1 and the number of constraints k_2) to the Short Nondeterministic Turing Machine Computation problem. Its instance is a nondeterministic Turing machine T with some fixed number of work tapes; the parameter is an integral number k; the question is whether there is a deterministic computation path of T (starting from empty tape) which accepts in at most k steps. This problem is W[1]-complete ([6, 7]).

Let us consider an instance of Multivalued Max GSAT:

$$\bigwedge_{i=1\ldots k_2} \bigvee_{j=1\ldots d_i} \bigwedge_{h=1\ldots c_{i,j}} [x_{ijh} = v_{ijh}]$$

where x_{ijh} is a variable in the set $\{x_1 \ldots x_k\}$ and v_{ijh} is a number in $\{0 \ldots n-1\}$. Note that $c_{i,j} \leq k_1$.

We should derive a nondeterministic Turing machine that executes the following pseudo-code:

nondeterministically guess k_1 values for $x_1 \ldots x_{k_1}$
for $i = 1$ **to** k_2 **do**
  nondeterministically guess j in $\{1 \ldots d_i\}$
  for $h = 1$ **to** $c_{i,j}$ **do**
    if $x_{ijh} \neq v_{ijh}$ **then reject**
  enddo
enddo
accept

Let us suppose that the original formula is satisfiable; then there is a deterministic computation path of T which guesses the correct values for the k_1 variables and then, for each index i, guesses the index j of a satisfied conjunction. Thus the computation never rejects, and eventually it accepts. Conversely, if a deterministic computation path of T accepts, the formula must be satisfiable.

The number of steps of each deterministic computation path is at most $O(k_1 + k_1 \cdot k_2)$ (observe that the alphabet of T should contain at least one symbol for each value in $\{0 \ldots n-1\}$). $\square$

Lemma 9 *Multivalued Planar* 3CSP *ia* W[1]*-hard.*

Proof: Let ϕ be an instance of multivalued 2CSP with k_1 variables and k_2 constraints. An embedding of the incidence graph of ϕ can have at most $(k_1k_2)^2$ crossings. Each crossing has the form shown in Figure 1, left. We remove such crossing by creating new variables x'_i and x'_j and a planar gadget enforcing $x_i = x'_i$ and $x_j = x'_j$; the cross-over gadget is depicted in Figure 1, right. The constraints of the gadget are as follows:

$$C_1 : x_i + x_j \equiv z \pmod n \qquad C_2 : x_j + x'_i \equiv z \pmod n$$
$$C_3 : x_i + x'_j \equiv z \pmod n \qquad C_4 : x'_j + x'_i \equiv z \pmod n$$

The gadget only requires 3CSP constraints. If $x_i = x'_i$ and $x_j = x'_j$ then all the constraints of the gagdet can be satisfied by setting $z \equiv x_i + x_j \pmod n$. Conversely, if all the constraints of the gadget are satisfied, then

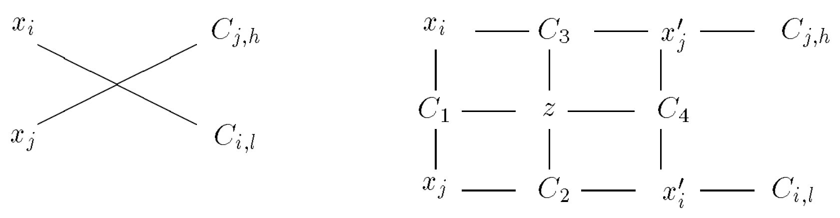

Figure 1: A multivalued cross-over gadget.

$$x_i \equiv z - x_j \equiv z - (z - x'_i) \equiv x'_i \pmod{n}$$

and

$$x_j \equiv z - x_i \equiv z - (z - x'_j) \equiv x'_j \pmod{n} ;$$

since $x_i, x_j, x'_i, x'_j \in \{0, \ldots, n-1\}$, it follows that $x_i = x'_i$ and $x_j = x'_j$. After replacing each crossing with a gadget, we have that the new (planar) instance is satisfiable if and only if the former instance was satisfiable. □

From Lemmas 6 and 9 the main result of this section follows.

Theorem 10 *Multivalued Planar* MAX 3CSP $\notin$ EPTAS *unless* W[1] = FPT.

5 An Artificial Hard Problem

In this section we will give a weak converse of Theorem 5, namely, we will prove the following theorem.

Theorem 11 *If* W[P] $\neq$ FPT *then* EPTAS $\neq$ PTAS.

We will prove the theorem by defining an artificial optimization problem that admits a PTAS but does not admit an EPTAS unless W[P] = FPT.

Let us define the parameterized Linear Inequalities problem in the following way:

INSTANCE: A system of n linear inequalities S.

PARAMETER: an integer $k \leq n$.

QUESTION: can we delete k inequalities and get a system that is consistent over the rationals?

Linear Inequalities is complete in W[P] and thus cannot belong to FPT unless FPT = W[P] [1]. Let us define an optimization version MAX SUBSYSTEM such that the measure in defined in a very tricky way.

INSTANCE: A set of linear inequalities S.

SOLUTION: A subset $S' \subset S$ such that $S - S'$ is consistent.

MEASURE: Let $k = |S'|$ and $n = |S|$: $m(S, S') = \left\lfloor 2(n+1)^{2\frac{k+1}{k}} \right\rfloor$.

Theorem 11 is implied by the following two lemmas.

Lemma 12 *Max Subsystem is in* PTAS*.*

Proof: Let us consider the following algorithm $T_k(S)$: first considers all $O(n^k)$ subsets $S' \subset S$ of cardinality at most k. If feasible solutions are found, then T_k returns one with maximum measure, otherwise returns a trivial solution S_{trivial} ($n-1$ equations removed). In the first case, algorithm T_k returns an optimum solution, otherwise, the minimum number of equations that has to be removed is $k^* \geq k+1$ and the achieved performance ratio is

$$\begin{aligned} R(S, S_{\text{trivial}}) &= \frac{\left\lfloor 2(n+1)^2 \frac{k^*+1}{k^*} \right\rfloor}{\left\lfloor 2(n+1)^2 \frac{n}{n-1} \right\rfloor} \\ &\leq \frac{\left\lfloor 2(n+1)^2 \frac{k+2}{k+1} \right\rfloor}{\left\lfloor 2(n+1)^2 \frac{n}{n-1} \right\rfloor} \\ &\leq \frac{2(n+1)^2 \frac{k+2}{k+1}}{2(n+1)^2} \\ &= \frac{k+2}{k+1} < 1 + 1/k \end{aligned}$$

□

Lemma 13 *If* Max Subsystem *is in* EPTAS*, then* Linear Inequalities *is in* FPT*.*

Proof: Let T be a EPTAS for the problem Max Subsystem and let f be a function such that the running time of $T(x,r)$ is bounded by $f(\lceil 1/(r-1) \rceil)|x|^\alpha$. Let us consider the following FPT algorithm for Linear Inequalities.

1. Execute $T(S,r)$ with $r = 1 + 1/(3(k^2+2k)$ and let S' be the returned solution.
2. If $|S'| \leq k$ then accept else reject.

The running time of the algorithm respects the definition of FPT and is clearly correct when $\langle S, k \rangle \notin$ Linear Inequalities. Suppose now that $\langle S, k \rangle \in$ Linear Inequalities and that the algorithm rejects, we will derive a contraddiction. Let S_T be the solution returned by $T(S, 1+1/(3(k^2+2k))$.

$$\begin{aligned} R(S, S_T) &\geq \frac{\left\lfloor 2(n+1)^2 \frac{k^*+1}{k^*} \right\rfloor}{\left\lfloor 2(n+1)^2 \frac{k+2}{k+1} \right\rfloor} \\ &\geq \frac{\left\lfloor 2(n+1)^2 \frac{k+1}{k} \right\rfloor}{\left\lfloor 2(n+1)^2 \frac{k+2}{k+1} \right\rfloor} \\ &\geq \frac{2(n+1)^2 \dfrac{k+1-\dfrac{k}{2(n+1)^2}}{k}}{2(n+1)^2 \frac{k+2}{k+1}} \\ &\geq \frac{k+1-\frac{1}{2(k+1)}}{k} \cdot \frac{k+1}{k+2} \end{aligned}$$

$$
\begin{aligned}
&= \frac{k^2+2k+1-1/2}{k^2+2k} \\
&= \frac{k^2+2k+1/2}{k^2+2k} \\
&= 1+\frac{1}{2(k^2+2k)} > 1+\frac{1}{3(k^2+2k)}
\end{aligned}
$$

where fourth inequality is due to the fact that $k \leq n$, and so $k/2(n+1)^2 \leq 1/2(k+1)$. $\square$

6 Conclusions

The relations between fixed parameter complexity and approximability was already considered by Cai and Chen [5]. Their study was directed towards the approximation properties of optimization problems with a hard induced language. Our approach, instead, focuses on the relation between the approximation factor in a PTAS and the parameter in a parameterized problem. The difference between the two approaches is best noted by considering that the induced language of the problem Max Subsystem, defined in Section 5, is in FPT.

When restricted to boolean domains, the planar Max GSATB problem admits an EPTAS for any fixed B [8], while we have shown that multivalued planar Max GSAT3 is not in EPTAS unless FPT $=$ W[1]. It appears that the size of the domain changes quite dramatically the approximability properties of constraint satisfaction problems. This point seems to be worth further investigation.

Acknowledgements

We are grateful to Sanjeev Khanna for his contribution to this research.

References

[1] K. A. Abrahamson, R. G. Downey, and M. R. Fellows. Fixed-parameter tractability and completeness IV: on completeness for $W[P]$ and $PSPACE$ analogues. *Annals of Pure and Applied Logic*, 73(3):235–276, June 15, 1995.

[2] S. Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science*, 1996.

[3] S. Arora, D. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. In *Proceedings of the 27th ACM Symposium on Theory of Computing*, pages 284–293, 1995.

[4] D.P. Bovet and P. Crescenzi. *Introduction to the Theory of Complexity.* Prentice Hall, 1993.

[5] L. Cai and J. Chen. On fixed-parameter tractability and approximability of NP-hard optimization problems. In *Proc. of 2nd Israel Symposium on Theory of Computing and Systems*, pages 118–126, 1993. To appear also in the *Journal of Computer and System Science.*

[6] L. Cai, J. Chen, R. G. Downey, and M. R. Fellows. On the parameterized complexity of short computation and factorization. To appear in the Archive for Mathematical Logic, 1995.

[7] M. Cesati and M. Di Ianni. Computation models for parameterized complexity. *Mathematical Logic Quarterly*, 43:179–202, 1997.

[8] P. Crescenzi and L. Trevisan. MAXNP-completeness made easy. Manuscript, 1996.

[9] W.F. de la Vega. MAXCUT has a randomized approximation scheme in dense graphs. Manuscript, 1994.

[10] R. G. Downey and M. R. Fellows. Fixed-parameter intractability (extended abstract). In *Proceedings of the 7th IEEE Conference on Structure in Complexity Theory*, pages 36–49, Boston, June 1992. IEEE.

[11] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness I: Basic theory. *SIAM J. Comput.*, 24:873–921, 1995.

[12] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness II: On completeness for $W[1]$. *Theoretical Computer Science*, 141:109–131, 1995.

[13] A. Frieze and R. Kannan. The regularity lemma and approximation schemes for dense problems. In *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science*, 1996.

[14] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection with learning and approximation. In *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science*, 1996.

[15] D.S. Hochbaum and D.B. Shmoys. Using dual approximation algorithms for scheduling problems: practical and theoretical results. *Journal of the ACM*, 34(1):144–162, 1987.

[16] D.S. Hochbaum and D.B. Shmoys. A linear time approximation scheme for the makespan problem. Manuscript, 1996.

[17] S. Khanna and R. Motwani. Towards a syntactic characterization of PTAS. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 329–337, 1996.