Available online at www.sciencedirect.com
ScienceDirect
Discrete Applied Mathematics 156 (2008) 3046–3056
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Note

FPTAS for half-products minimization with scheduling applications

Erdal Erel[a,*], Jay B. Ghosh[b]

[a] Faculty of Business Administration, Bilkent University, 06800 Bilkent, Ankara, Turkey
[b] Korea University Business School, Korea University, Seoul, Korea

Received 14 January 2007; received in revised form 14 January 2008; accepted 17 January 2008
Available online 7 March 2008

Abstract

A special class of quadratic pseudo-boolean functions called "half-products" (HP) has recently been introduced. It has been shown that HP minimization, while NP-hard, admits a fully polynomial time approximation scheme (FPTAS). In this note, we provide a more efficient FPTAS. We further show how an FPTAS can also be derived for the general case where the HP function is augmented by a problem-dependent constant and can justifiably be assumed to be nonnegative. This leads to an FPTAS for certain partitioning type problems, including many from the field of scheduling.

Keywords: Quadratic pseudo-boolean functions; Dynamic programming; Approximation scheme

1. Introduction

Given n binary variables x_i, $1 \leq i \leq n$, and $3n$ parameters p_i, q_i and r_i such that $p_i, q_i, r_i \geq 0$ and integer, $1 \leq i \leq n$, a pseudo-boolean function $f(\mathbf{x})$ of the n-variable vector $\mathbf{x}$, $\mathbf{x} = (x_1, \ldots, x_n)$, $f(\mathbf{x}) = -\sum_{1\leq i\leq n} p_i x_i + \sum_{1\leq i<j\leq n} q_i r_j x_i x_j$, has been called the "half-products" (HP) function by Badics and Boros [1]. They have shown that $f(\mathbf{x})$ minimization is NP-hard and given a fully polynomial time approximation scheme (FPTAS) for the problem; this delivers an ε-approximate solution $\mathbf{x}^{f\varepsilon}$ such that $f(\mathbf{x}^{f\varepsilon}) - f(\mathbf{x}^*) \leq \varepsilon|f(\mathbf{x}^*)|$, where $\mathbf{x}^*$ is an optimal solution, in $O(n^2 \log Q/\varepsilon)$ time, where Q equals $\sum_{1\leq i\leq n} q_i$, for any ε, $0 < \varepsilon \leq 1$. They have also applied this FPTAS to the NP-hard completion time variance problem (CTV) to obtain an ε-approximate solution to CTV in $O(n^3 \log Q/\varepsilon)$ time.

At this point, a few preliminaries about HP are in order. First, note that, in a non-trivial instance of the HP minimization problem, some parameters are necessarily positive such that $\sum_{1\leq i\leq n} p_i > 0$ and $\sum_{1\leq i<j\leq n} q_i r_j > 0$. (Otherwise: if the first inequality is not true, set all $x_i = 0$ to get an optimal solution; if the second is not true, set all $x_i = 1$.) Also note that, in such an instance, if $\mathbf{x}^*$ is optimal, then $-P < f(\mathbf{x}^*) < 0$ (where $P = \sum_{1\leq i\leq n} p_i$). (The left inequality follows since $f(\mathbf{x}^*) \leq \min_x\{-\sum_{1\leq i\leq n} p_i x_i\} + \min_x\{\sum_{1\leq i<j\leq n} q_i r_j x_i x_j\}$ and equality is impossible; the right inequality follows since $f(\mathbf{x}) = -p_i < 0$, where $\mathbf{x}$ is given by $(x_1 = 0, \ldots, x_{i-1} = 0, x_i = 1, x_{i+1} = 0, \ldots, x_n = 0)$ with $p_i > 0$ and $f(\mathbf{x}^*) \leq f(\mathbf{x})$.) Finally, note that it is not necessary to assume that $p_i \geq 0$

* Corresponding author. Tel.: +90 312 266 4164; fax: +90 312 266 4958.
E-mail address: erel@bilkent.edu.tr (E. Erel).

0166-218X/$ - see front matter
doi:10.1016/j.dam.2008.01.018

for HP minimization, since x_i can be set to 0 and all HP terms involving p_i, q_i and r_i removed if $p_i < 0$ [1]. (In fact, our algorithms do not assume that $p_i \geq 0$ and set $x_i = 0$ whenever $p_i < 0$.)

We now briefly review the past works on HP. Kubiak [11] has independently developed what is an HP representation of CTV and exploited it to develop a couple of pseudo-polynomial time dynamic programs to solve the problem. Following this, Jurisch et al. [8] have developed HP representations for four scheduling problems (without explicitly calling them so). In addition to CTV, their list includes: 2-machine make-span problem (MAKS); 2-machine weighted completion time problem (WCT); weighted earliness tardiness problem (WET). Focus has now shifted to HP as formally exposited by Badics and Boros [1]. Janiak et al. [7] has reformulated a sub-class of HP as what they call positive HP to prove that the scheduling problem with controllable processing times (CONT) is NP-hard and to give an FPTAS for the problem. Cheng and Kubiak [2] have used HP again to give an FPTAS for the agreeably weighted completion time variance problem (AWCTV). Kubiak [12] has considered another interesting sub-class of HP, called ordered-symmetric HP, to which MAKS and CTV belong, and has given an efficient FPTAS for this sub-class. Finally, it should be mentioned that an FPTAS outside of the HP framework has been in existence for each of the above problems except CONT (see, for example, [15] for MAKS and WCT, [17] for AWCTV, [10] for WET, and [13] for CTV). We will briefly state the problems referred above as and when the need arises. Meanwhile, note that Kubiak [12] and Jurisch et al. [8] provide formal statements of these problems and their complete HP representations. Finally, some recent works on scheduling, either as extensions of the six problems mentioned above or otherwise, address HP optimization to a certain extent (for example, [16]) or present an opportunity for HP application (for example, [6]).

Only rarely does a problem at hand map directly to a pure HP representation. If it does, an FPTAS for f-minimization is also an FPTAS for the problem. One example is MAKS. Typically though, a problem leads to a representation where HP is augmented with a problem-dependent constant. This calls for h-minimization, where $h(\mathbf{x}) = f(\mathbf{x}) + K$ and K is independent of $\mathbf{x}$. Note that $h(\mathbf{x}) = K$ for $\mathbf{x} = (0, \ldots, 0)$ and that $K = h(\mathbf{0}) \geq h(\mathbf{x}^*)$, where $\mathbf{x}^*$ minimizes both $f(\mathbf{x})$ and $h(\mathbf{x})$. In a non-trivial instance of the problem, using the bounds on $f(\mathbf{x}^*)$ introduced earlier, we have: $K - P < h(\mathbf{x}^*) < K$. The issue of the sign of $h(\mathbf{x}^*)$ is important and is addressed at length in Section 3. For now, simply note that K is a "nuisance" parameter, which often prevents an FPTAS for f-minimization from yielding an FPTAS for h-minimization. (See [7,12] for a discussion of this difficulty.) However, this is not always the case. The FPTAS for f-minimization yields an FPTAS for h-minimization, as long as the ratio $|f(x*)|/|h(x*)|$ remains bounded by a polynomial function of the problem size. This is indeed the case for CTV, WCT and AWCTV [12]. Unfortunately, no such bound may exist for some problems such as WET and CONT [12,7]. A different approach is needed. This is the background in which Janiak et al. [7] have introduced their positive HP-formulation for CONT; to the best of our knowledge, WET (which is not a positive HP) has not been addressed from an HP perspective as yet. In this note, we provide a general alternative to h-minimization, which is also efficient. Our approach, which assumes that $f(\mathbf{x}) \geq 0$ based on contextual grounds, allows one to attack a large class of problems, that admits an augmented HP representation, within a unified framework (rather than on a piece-meal basis).

We first present an $O(n^2/\varepsilon)$ FPTAS for f-minimization and discuss the conditions under which this also yields an FPTAS for h-minimization (either directly or after minor manipulation). We use as examples MAKS and CTV, for which the most competitive FPTAS have been given by Sahni [15] and Kubiak et al. [13], respectively. The time complexity obtained by us for all HP is an improvement over that of Badics and Boros [1]. However, Kubiak [12] reports the same time complexity as ours for his sub-class of ordered-symmetric HP.

Next, we turn to h-minimization. We begin by discussing how to obtain lower and upper bounds on $h(\mathbf{x}^*)$ in polynomial time and how these bounds can be used to determine the sign of $h(\mathbf{x}^*)$ when possible. For different values of K, we briefly outline when and how an FPTAS can be obtained. We then confine ourselves to a context such as scheduling, where in most cases it can be assumed *a priori* that $h(\mathbf{x}^*) \geq 0$. We provide an approximation scheme, which becomes an FPTAS if polynomially-related lower and upper bounds on $h(\mathbf{x}^*)$ are available. As an example, we use WCT, for which the most competitive FPTAS has been given by Sahni [15]. When the upper-to-lower-bound ratio is large, one may be able to use the approach of Gens and Levner [4] or that of Kovalyov [9] to tighten it. We adopt an alternate approach, where we utilize our basic approximation scheme within a binary search framework to obtain an FPTAS that is $O(n^2 \log(\varepsilon K)/\varepsilon)$ and valid when $h(\mathbf{x}) \geq 0$. It is relevant in situations where there are no available bounds on $h(\mathbf{x}^*)$ except the trivial, as well as where the upper-to-lower-bound ratio is polynomially bounded but unacceptably large. We use as examples WET, CONT and AWCTV, for which the most competitive FPTAS have been given by Kovalyov and Kubiak [10], Janiak et al. [7] and Cheng and Kubiak [2], respectively. (Note that, though polynomially-related lower and upper bounds are available for AWCTV, a better time complexity is realized when we use binary search to tighten their ratio.) In summary, our approach along with that of Janiak et al. [7] provides closure to h-minimization over a large range of K values. For the six specific problems introduced at the outset, our results compare remarkably well with what exists in the literature at present (this is particularly true for AWCTV). The only other general approach available, the FPTAS for positive HP [7], exhibits a time complexity of similar order but is more restrictive scope-wise in a given context.

2. f-minimization

We obtain our FPTAS for f-minimization through minor modification of a pseudo-polynomial time dynamic program (DP) for minimizing $f(\mathbf{x})$. The technique we use is often called "state-space thinning". Different variants of this approach (originally due to Ibarra and Kim [5]) have been used by others as well (e.g., [13]).

2.1. DP

The DP is described in an enumerative form. We assign the value of 0 or 1 to the n variables successively, starting with x_1 at stage 1 and ending with x_n at stage n. At stage k, $1 \le k \le n$, we consider assignment to x_k. Let $Q_k = \sum_{1\le i\le k} q_i x_i$, and $F_k = -\sum_{1\le i\le k} p_i x_i + \sum_{1\le i<j\le k} q_i r_j x_i x_j$ for a partial assignment $\mathbf{x}_k = (x_1, \dots, x_k)$ at this stage. The pair $\langle Q_k, F_k\rangle$, often called the "state" of the assignment, carries all information about $\mathbf{x}_k$ needed to facilitate the forward movement of the DP. For convenience, let $(\mathbf{x}_k, \langle Q_k, F_k\rangle)$ represent the partial assignment $\mathbf{x}_k$ and the $\langle Q_k, F_k\rangle$ pair associated with it. We now state a result that will help us identify and retain a set of partial assignments at each stage such that one of them will provably lead to an optimal full assignment $\mathbf{x}_n^*$ at the end of stage n. (Badics and Boros [1] also present this result. Note further that results like this exist for structurally similar problems that are not necessarily HP; see [15].) In what follows, let $\mathbf{x}_n$ be synonymous with $\mathbf{x}$, with or without any superscript on $\mathbf{x}$.

Lemma 1. *Given $(\mathbf{x}_k, \langle Q_k, F_k\rangle)$ and $(\mathbf{x}'_k, \langle Q'_k, F'_k\rangle)$ at stage k of the DP such that $Q_k \le Q'_k$ and $F_k \le F'_k$, it is sufficient to retain only $(\mathbf{x}_k, \langle Q_k, F_k\rangle)$ for further enumeration.*

Proof. Note first that we can write $f(\mathbf{x})$ as follows:

$$f(\mathbf{x}) = \left[-\sum_{1\le i\le k} p_i x_i + \sum_{1\le i<j\le k} q_i r_j x_i x_j\right] + \left[-\sum_{k+1\le i\le n} p_i x_i + \sum_{k+1\le i<j\le n} q_i r_j x_i x_j\right] + \left[\left(\sum_{1\le i\le k} q_i x_i\right)\left(\sum_{k+1\le i\le n} r_i x_i\right)\right].$$

Suppose next that the partial assignment $\mathbf{x}'_k = (x'_1, \dots, x'_k)$, given by $\langle Q'_k, F'_k\rangle$, is optimally completed with the assignment $(x'_{k+1}, \dots, x'_n)$ for the remaining $n-k$ variables, resulting in $\mathbf{x}'_n = (x'_1, \dots, x'_k, x'_{k+1}, \dots, x'_n)$, given by $\langle Q'_n, F'_n\rangle$. We have:

$$f(\mathbf{x}'_n) = F'_n = F'_k + \left[-\sum_{k+1\le i\le n} p_i x'_i + \sum_{k+1\le i<j\le n} q_i r_j x'_i x'_j\right] + Q'_k\left(\sum_{k+1\le i\le n} r_i x'_i\right).$$

Now complete $\mathbf{x}_k = (x_1, \dots, x_k)$, given by $\langle Q_k, F_k\rangle$, with the same assignment for the remaining $n-k$ variables as above to obtain $\mathbf{x}_n = (x_1, \dots, x_k, x'_{k+1}, \dots, x'_n)$, given by $\langle Q_n, F_n\rangle$. We now have:

$$f(\mathbf{x}_n) = F_n = F_k + \left[-\sum_{k+1\le i\le n} p_i x'_i + \sum_{k+1\le i<j\le n} q_i r_j x'_i x'_j\right] + Q_k\left(\sum_{k+1\le i\le n} r_i x'_i\right).$$

Upon subtraction, we get:

$$\begin{aligned} f(\mathbf{x}_n) - f(\mathbf{x}'_n) &= F_n - F'_n = (F_k - F'_k) + (Q_k - Q'_k)\left(\sum_{k+1\le i\le n} r_i x'_i\right) \\ &\le 0 \quad \text{(as per the condition stated in the lemma).} \end{aligned}$$

We see that the partial assignment $\mathbf{x}'_k$ does not lead to a full assignment that is better than what we can get by completing $\mathbf{x}_k$. We can thus discard $(\mathbf{x}'_k, \langle Q'_k, F'_k\rangle)$. □

As before, let $(\mathbf{x}_k, \langle Q_k, F_k\rangle)$ be a partial assignment $\mathbf{x}_k$ and its $\langle Q_k, F_k\rangle$ pair. Let Ω_k be the set of all $(\mathbf{x}_k, \langle Q_k, F_k\rangle)$ considered during stage k of the DP. Assume that "$\oplus$" stands for a concatenation and "$\emptyset$" for a null assignment. We can now state the following procedure.

Procedure DP_Min F:
Step 0: Set $\mathbf{\Omega}_0 = \{(\emptyset, \langle 0, 0\rangle)\}$.
Step 1: For $k = 1$ through n:

(a) For each $(\mathbf{x}_{k-1}, \langle Q_{k-1}, F_{k-1}\rangle) \in \mathbf{\Omega}_{k-1}$, add the following to $\mathbf{\Omega}_k$: $(\mathbf{x}_{k-1} \oplus 0, \langle Q_{k-1}, F_{k-1}\rangle)$ always, and $(\mathbf{x}_{k-1} \oplus 1, \langle Q_{k-1} + q_k, F_{k-1} - p_k + r_k Q_{k-1}\rangle)$ only if - $p_k + r_k Q_{k-1} < 0$.

(b) For all $(\mathbf{x}_k, \langle Q_k, F_k\rangle) \in \mathbf{\Omega}_k$ with the same Q_k, retain one with the smallest F_k.

Step 2: From $\mathbf{\Omega}_n$, find a member $(\mathbf{x}_n, \langle Q_n, F_n\rangle)$ with the minimum F_n.

Theorem 1. DP_Min F *solves the f-minimization problem in $O(nQ)$ time.*

Proof. The above procedure is correct as: (1) all possible extensions of $\mathbf{x}_{k-1}$ to $\mathbf{x}_k$ at any stage k are considered; (2) all $(\mathbf{x}_k, \langle Q_k, F_k\rangle)$ added to $\mathbf{\Omega}_k$ at any stage k are computed properly in Step 1(a); (3) $(\mathbf{x}_{k-1} \oplus 1, \langle Q_{k-1} + q_k, F_{k-1} - p_k + r_k Q_{k-1}\rangle)$ is not added to $\mathbf{\Omega}_k$ in Step 1(a) at any stage k only if it is strictly worse than $(\mathbf{x}_{k-1} \oplus 0, \langle Q_{k-1}, F_{k-1}\rangle)$ in the 2nd coordinate of the $\langle\cdot, \cdot\rangle$ pair (note that it is no better in the 1st) and can be discarded as per Lemma 1; (4) $(\mathbf{x}_k, \langle Q_k, F_k\rangle)$ is removed from $\mathbf{\Omega}_k$ at any stage k only if, compared to another member of $\mathbf{\Omega}_k$, it is no better in the 2nd coordinate of the $\langle\cdot, \cdot\rangle$ pair (note that it is equal in the 1st) and can be discarded as per Lemma 1.

Next, note that the procedure retains exactly one member for each distinct value Q_k and further that Q_k is an integer bounded above by Q (recall that $Q = \sum_{1\le i\le n} q_i$). The cardinality of $\mathbf{\Omega}_k$ is thus $O(Q)$. This translates to an $O(nQ)$ total execution time over n stages of the DP. Therefore, DP_Min F minimizes $f(\mathbf{x})$ in $O(nQ)$ time. □

2.2. FPTAS

We now describe our FPTAS for f-minimization. The procedure is identical to DP_Min F except for the thinning mechanism used in Step 1(b). The new mechanism keeps the size of $\mathbf{\Omega}_k$ polynomially bounded and guarantees that $f(\mathbf{x}^{f\varepsilon}) - f(\mathbf{x}^*) \le \varepsilon|f(\mathbf{x}^*)|$, for $0 < \varepsilon \le 1$ (note that an ε outside of this range is meaningless for f-minimization), where $\mathbf{x}^{f\varepsilon}$ is the solution delivered and $\mathbf{x}^*$ is an optimal solution. Before moving on to the procedure, let $\mathrm{UBF}_k = \min\{F_k\}$, where the minimum is taken over all $(\mathbf{x}_k, \langle Q_k, F_k\rangle) \in \mathbf{\Omega}_k$ at the end of Step 1(a) in εAPX_Min F which follows (this step is the same as in DP_Min F), and $\Delta_k = (-\varepsilon \mathrm{UBF}_k)/n$. Notice here that $|\mathrm{UBF}_k| \le |f(\mathbf{x}^*)|$ and thus that $\Delta_k \le (\varepsilon|f(\mathbf{x}^*)|)/n$.

Procedure ε **APX_Min** F:

[All steps are identical to Procedure DP_Min F except Step 1(b). Replace as follows.]

Step$1'$(b):

(i) Divide the interval $[\mathrm{UBF}_k, 0]$ into subintervals of width Δ_k.
(ii) From all $(\mathbf{x}_k, \langle Q_k, F_k\rangle) \in \mathbf{\Omega}_k$ with F_k in the same subinterval, retain one with the smallest Q_k.

Theorem 2. εAPX_Min F *delivers an ε-approximate solution to the f-minimization problem in $O(n^2/\varepsilon)$ time and is thus an FPTAS.*

Proof. At stage k of εAPX_Min F, the number of subintervals considered is bounded above by $\lceil -\mathrm{UBF}_k/\Delta_k\rceil$. Substituting for Δ_k, we have: $\lceil -\mathrm{UBF}_k/\Delta_k\rceil \le n/\varepsilon + 1$. At most one member is retained in each subinterval, and thus the cardinality of $\mathbf{\Omega}_k$ at the end of Step $1'$(b) is $O(n/\varepsilon)$. Over n stages, εAPX_Min F thus has a time complexity of $O(n^2/\varepsilon)$.

To see the correctness of the approximation, consider first any subinterval at the end of Step $1'$(b)(i) in stage k of εAPX_Min F. Let $(\mathbf{x}^a_k, \langle Q^a_k, F^a_k\rangle)$ be a member of $\mathbf{\Omega}_k$ prior to the start of Step $1'$(b)(ii), which has F_k in that subinterval and the smallest Q_k among all members of $\mathbf{\Omega}_k$ at that point with F_k in the same subinterval, and which is retained by εAPX_Min F at the end of Step $1'$(b)(ii). Let $(\mathbf{x}^b_k, \langle Q^b_k, F^b_k\rangle)$ be another member of $\mathbf{\Omega}_k$ prior to the start of Step $1'$(b)(ii), which is associated with the same subinterval, and which is discarded at the end of Step $1'$(b)(ii). We have: $F_k^a - F_k^b \le \Delta_k$ and $Q_k^a - Q_k^b \le 0$. Let the partial assignment $\mathbf{x}_k^b = (x_1^b, \ldots, x_k^b)$ be optimally completed to obtain the full assignment $\mathbf{x}_n^b = (x_1^b, \ldots, x_n^b)$ and let $\mathbf{x}_k^a = (x_1^a, \ldots, x_k^a)$ be identically completed to obtain the full assignment $\mathbf{x}_n^{ab} = (x_1^a, \ldots, x_k^a, x_{k+1}^b, \ldots, x_n^b)$. Using the expression for $f(\mathbf{x}_n)$ given earlier in the proof of Lemma 1, we get: $f(\mathbf{x}_n^{ab}) - f(\mathbf{x}_n^b) \le \Delta_k$. Let $\mathbf{x}_n^a = (x_1^a, \ldots, x_n^a)$ be the full assignment obtained upon an optimal completion of $\mathbf{x}_k^a$. By definition, $f(\mathbf{x}_n^a) \le f(\mathbf{x}_n^{ab})$. It follows then that $f(\mathbf{x}_n^a) - f(\mathbf{x}_n^b) \le \Delta_k \le (\varepsilon|f(\mathbf{x}^*)|)/n$. This is the maximum error due to the approximation at stage k.

Let k_1 through k_m, $1 \le k_1 < \cdots < k_m \le n$, be m, $1 \le m \le n$, critical stages of εAPX_Min F such that, for $i = 1, \ldots, m-1$, $(\mathbf{x}_{k_i}^i, \langle Q_{k_i}^i, F_{k_i}^i \rangle)$ with $\mathbf{x}_{k_i}^i = (x_1^i, \ldots, x_{k_i}^i)$ remains a member of $\mathbf{\Omega}_{k_i}$ at stage k_i prior to the start of Step $1'$(b)(ii), but is discarded in favor of another member $(\mathbf{x}_{k_i}^{i+1}, \langle Q_{k_i}^{i+1}, F_{k_i}^{i+1} \rangle)$ with $\mathbf{x}_{k_i}^{i+1} = (x_1^{i+1}, \ldots, x_{k_i}^{i+1})$ at the end of that step. Also, let $\mathbf{x}_n^i = (x_1^i, \ldots, x_n^i)$ be the full assignment obtained upon an optimal completion of $\mathbf{x}_{k_i}^i$, for $i = 1, \ldots, m$. Note that only $\mathbf{x}_n^m$ survives until the end of Step $1'$(b)(ii) of εAPX_Min F in stage n.

From the foregoing, it is clear that $f(\mathbf{x}_n^{i+1}) - f(\mathbf{x}_n^i) \le (\varepsilon|f(\mathbf{x}^*)|)/n$, for $i = 1, \ldots, m-1$. Summing the inequalities over all i, we get: $f(\mathbf{x}_n^m) - f(\mathbf{x}_n^1) \le (m-1)(\varepsilon|f(\mathbf{x}^*)|)/n \le \varepsilon|f(\mathbf{x}^*)|$, since $m \le n$. Now, let $\mathbf{x}_n^{f\varepsilon}$ be the solution delivered by εAPX_Min F at the end of Step 2. We have: $f(\mathbf{x}_n^{f\varepsilon}) \le f(\mathbf{x}_n^m)$. Finally, let $\mathbf{x}_n^*$ be an optimal solution and set $\mathbf{x}_n^1 = \mathbf{x}_n^*$. It immediately follows that $f(\mathbf{x}_n^{f\varepsilon}) - f(\mathbf{x}_n^*) \le \varepsilon|f(\mathbf{x}^*)|$. (Note that $\varepsilon|f(\mathbf{x}^*)|$ is the maximum total error over all n stages due to the approximation.) This completes our proof that εAPX_Min F is an $O(n^2/\varepsilon)$ FPTAS for f-minimization. □

As an example of where the above FPTAS directly yields an FPTAS for the original problem, consider MAKS. Here, there are n jobs, with processing times $t_1, \ldots, t_n$, that are to be scheduled on 2 identical machines such that $\max_i\{C_i\}$, where C_i is the completion time of job i, is minimized. This is equivalent to finding a job assignment to the machines which will maximize the product of the make-spans on the individual machines. Kubiak [12] provides the HP representation and other relevant details. For now, it suffices to note that the HP representation does not contain the usual constant term and, in that sense, is a pure HP. Thus, our FPTAS is also an $O(n^2/\varepsilon)$ FPTAS for MAKS. This is similar to what one gets from [15].

2.3. *h-minimization via f-minimization*

We now state a result, due to Kubiaket al. [13], that tells us when the above FPTAS can be manipulated easily to yield an FPTAS for h-minimization as well. For ease of notation, let $F^* = f(\mathbf{x}^*)$ and $H^* = h(\mathbf{x}^*) = K + f(\mathbf{x}^*)$, where $\mathbf{x}^*$ is a solution that minimizes both $f(\mathbf{x})$ and $h(\mathbf{x})$. Similarly, let $F^{f\varepsilon} = f(\mathbf{x}^{f\varepsilon})$ and $H^{f\varepsilon} = h(\mathbf{x}^{f\varepsilon}) = K + f(\mathbf{x}^{f\varepsilon})$, where $\mathbf{x}^{f\varepsilon}$ is the solution delivered by εAPX_Min F. (Note that no FPTAS exists for $H^* = 0$. We thus assume that $H^* \ne 0$. We will see in Section 3 how this issue can be addressed.)

Theorem 3. *If $|F^*| \le \alpha|H^*|$ for some $\alpha > 0$, then the solution $\mathbf{x}^{f\varepsilon}$ delivered by εAPX_Min F is an $\alpha\varepsilon$-approximate solution to the h-minimization problem.*

Proof. We have: $F^{f\varepsilon} - F^* \le \varepsilon|F^*| \Rightarrow (K + F^{f\varepsilon}) - (K + F^*) \le \varepsilon|F^*| \Rightarrow H^{f\varepsilon} - H^* \le \varepsilon|F^*|$. Using the stated condition, this leads to: $H^{f\varepsilon} - H^* \le \alpha\varepsilon|H^*|$. □

If the above condition holds for an α such that $\alpha \sim \text{poly}(S)$, where S is the size of a problem instance, we get an $O(n^2\text{poly}(S)/\varepsilon)$ FPTAS for h-minimization if we use ε/α instead of ε in εAPX_Min F. We show below how this applies to CTV.

CTV involves n jobs, with processing times $t_1, \ldots, t_n$, that are to be scheduled on a single machine such that $[\sum_i (C_i - C_{\text{avg}})^2]/n$, where C_i is the completion time of job i and $C_{\text{avg}} = [\sum_i C_i]/n$, is minimized. As always, [12] provides the HP representation and other relevant details. Only note that the HP representation in this case includes a constant term. Recently, Kubiak et al. [13] have arrived at a new lower bound for CTV and have realized an α of 3 through it; their best FPTAS, based on a "rounding" technique applied to the maximization of - f(x), is $O(n^2/\varepsilon)$. Clearly, if we use an α of 3, εAPX_Min F also yields an $O(n^2/\varepsilon)$ FPTAS for CTV minimization.

2.4. Remarks

1. Notice that it is the particular structure of the HP function which permits the kind of separation between the assigned and the unassigned variables that we get in the expression for $f(\mathbf{x}_n)$, as given in the proof of Lemma 1. This makes the optimization of the HP function and the approximation of its optimal value relatively easy (compared to general pseudo-boolean quadratic functions). (Similar separation can also be seen in non-HP contexts; see [15,14].) One disadvantage of the HP-formulation over a conventional non-HP one is that the development of an HP function can sometimes be quite involved; for example, consider AWCTV [2]. However, the HP-formulation has certain advantages; see Remark 4.

2. An alternate DP is also possible for f-minimization. (Kubiak [11], Jurisch et al. [8] and Janiak et al. [7] make similar observations in their respective contexts.) The alternate DP builds the solution in reverse, starting with x_n and finishing with x_1. The time complexity in this case is $O(nR)$, where $R = \sum_{1\le i\le n} r_i$. Finally, an FPTAS can also be developed in this case, much the same way as we have done here; we omit the details. See also [7].

3. Obviously, there is a pair of DP and FPTAS for the direct maximization of $-f(\mathbf{x})$ similar to DP_Min F and εAPX_Min F (obtained through a trivial modification). In fact, it may be preferable to work with $-f(\mathbf{x})$, as it avoids the awkwardness of minimizing over negative numbers. Kubiak et al. [13], among others, actually do this.

4. In terms of an FPTAS for HP, one can thin the state-space in one of two ways: either on F or on Q. We choose to thin on F, whereas [1] for HP and [7] for positive HP choose to thin on Q. Thinning on F makes sense when a strong dominance condition such as the one in Lemma 1 (with inequalities on both state variables as opposed to just one) exists. It also works well when the objective is maximization (see [15].) Lemma 1 and the observation that f-minimization is essentially is a maximization of $-f(\mathbf{x})$ have motivated us to thin on F and helped us realize the $O(n^2/\varepsilon)$ time bound for our FPTAS.

3. h-minimization

A few things need clarification before we proceed to our main results. First, it is easily recognized that the solution $\mathbf{x}^*$, which minimizes $f(\mathbf{x})$, minimizes $h(\mathbf{x})$ as well. The sign of $h(\mathbf{x}^*)$ is immaterial there. However, the sign becomes a major issue when we try to develop an FPTAS for h-minimization. It follows from [1] that the problem of determining if there is an $\mathbf{x}^*$ such that $f(\mathbf{x}^*) \le -K$ (or, equivalently, determining if $h(\mathbf{x}^*) = K + f(\mathbf{x}^*) \le 0$), for an arbitrary K, is NP-hard. It is thus unlikely that we will find the sign of $f(\mathbf{x}^*)$ in polynomial time. We can resort to one of the two approaches at this juncture. In the first, we treat K free of any contextual underpinning. In the second, we exploit the context in which an HP is formulated to get an answer to the sign question. We will cover both approaches, but will focus primarily on the second (using scheduling as our context).

3.1. The context-free approach

When nothing can be assumed about the sign of $h(\mathbf{x})$, we can still obtain a valid pair of lower and upper bounds on $h(\mathbf{x}^*)$ in polynomial time. We have given one pair in Section 1: $K - P < h(\mathbf{x}^*) < K$ (recall that $P = \sum_{1\le i\le n} p_i$). Letting $\mathbf{x}^{f\varepsilon}$ be the solution delivered by εAPX_Min F in $O(n^2/\varepsilon)$ time, we get another: $h(\mathbf{x}^{f\varepsilon}) = K + f(\mathbf{x}^{f\varepsilon})/(1-\varepsilon) \le h(\mathbf{x}^*) \le K + f(\mathbf{x}^{f\varepsilon})$. Let UBF $= \min\{0, f(\mathbf{x}^{f\varepsilon})\}$, LBF $= \max\{-P, f(\mathbf{x}^{f\varepsilon})/(1-\varepsilon)\}$. Similarly, let UBH $= K +$ UBF and LBH $= K +$ LBF. Clearly: if UBH < 0, then $h(\mathbf{x}^*) < 0$; if LBH > 0, then $h(\mathbf{x}^*) > 0$, and, in fact, $h(\mathbf{x}) > 0$ for all $\mathbf{x}$, as $h(\mathbf{x}) \le h(\mathbf{x}^*)$; otherwise, $h(\mathbf{x}^*)$ is of indeterminate sign. (We use these particular LBH and UBH mostly for illustration; any legitimate pair will suffice.)

Suppose UBH < 0 (i.e., $K < -$UBF). As before, let $F^* = f(\mathbf{x}^*)$, $H^* = h(\mathbf{x}^*)$, $F^{f\varepsilon} = f(\mathbf{x}^{f\varepsilon})$ and $H^{f\varepsilon} = h(\mathbf{x}^{f\varepsilon})$, $\mathbf{x}^{f\varepsilon}$ being an ε-approximate solution for f-minimization. Consider the sub-cases: $K \le 0$ and $0 < K < -$UBF. For $K \le 0$: it is seen that $|F^*| \le |H^*|$; it follows from Theorem 3 that εAPX_Min F is an FPTAS for h-minimization as well. For $0 < K < -$UBF (where $|F^*| > |H^*|$): it follows again from Theorem 3 that εAPX_Min F yields an FPTAS for h-minimization, as long as $K \le (\alpha - 1)|\text{UBF}|/\alpha$, where α, $\alpha > 1$, is a constant or a polynomially bounded function of the problem size; the FPTAS question remains open otherwise.

Suppose LBH > 0 (i.e., $K > -$LBF). For $K > -$LBF: Theorem 3 tells us once again that εAPX_Min F yields an FPTAS for h-minimization if $K \ge (\alpha+1)|\text{LBF}|/\alpha$ for α, $\alpha > 0$. It is known that this condition holds for all of our six problems except CONT and WET. This brings us to the question: what if the above condition does not hold? Janiak et al. [7] answers this question partially by imposing the restriction that $K - P \geq 0$ be true in an HP that is augmented with a constant. (An augmented HP under this restriction is called a positive HP.) They give an $O(n^2 \log(Q)/\varepsilon)$ and an $O(n^2 \log(R)/\varepsilon)$ FPTAS for h-minimization for this sub-class of HP, and show further that CONT belongs to this sub-class. Among our six problems, it appears that WET is the only one that neither admits an FPTAS via Theorem 3 nor is a positive HP.

3.2. The context-dependent approach

Our approach lets us side-step the difficulty faced in the case of WET by assuming that $f(\mathbf{x}) \geq 0$ for all $\mathbf{x}$. We do this in the context of the problem. Our rationale is that an HP typically appears in a given context. In such a context, it may be justified to assume that $h(\mathbf{x})$ is of a certain sign for all $\mathbf{x}$ and be possible further to exploit certain properties of $h(\mathbf{x})$ to develop a general FPTAS for h-minimization. One context that is of particular interest to us is scheduling. It is in this context that we have independently developed our FPTAS [3], under the assumptions that $h(\mathbf{x}) \geq 0$, that K has a contextual meaning, and that both K and $h(\mathbf{x})$ are nondecreasing in n. Most scheduling cost functions represented by $h(\mathbf{x})$ will be nonnegative (with rare exceptions such as those involving job lateness), i.e., it will be fair to assume that $h(\mathbf{x}) \geq 0$. Also, in a scheduling problem that admits an HP-formulation and involves partitioning the job set, K will be the cost of an actual schedule where all the jobs are placed in one partition or the other (to one machine as in MAKS and WCT, before the shortest job as in CTV and AWCTV, in the early set as in WET, or at the lowest processing time limit as in CONT) and will thus satisfy $\mathrm{K} \geq 0$. Finally, it will generally be true that the cost of a k-job schedule derived from a $(k-1)$-job schedule will be no less than that of its parent, i.e., $h(\mathbf{x}_k) \geq h(\mathbf{x}_{k-1})$ for all k, $0 < k \leq n$; this implies that, if we let $h(\mathbf{x}_k) = K_k + h(\mathbf{x}_k)$, $K_k \geq K_{k-1}$. These assumptions are implicit in our approach.

We start with a DP for h-minimization that is rooted in contexts where the above assumptions apply and is critical to the development of our FPTAS. As in the last section, this DP is slightly modified to obtain an approximation scheme. This in turn becomes an FPTAS when we have access to lower and upper bounds, call them LBH and UBH as before, on H^* such that UBH/LBH is bounded above by poly(S), a polynomial function of the problem size S. Even when this is not the case, we can use the approximation scheme within a binary search scheme to obtain an FPTAS.

Before we go on to describe our DP and FPTAS for h-minimization, we highlight that we focus only on situations where $h(\mathbf{x}) \geq h(\mathbf{x}^*) \geq 0$. We also note that, given that $h(\mathbf{x}) \geq 0$, the inapproximable case (where $h(\mathbf{x}^*) = 0$) can be recognized in $O(n)$ time. (Recall from our earlier comments that doing so in general is NP-hard.) To see how this is done, refer to DP_Min H given in the next sub-section and simply set UBH $= 0$. If DP_Min H returns a solution, the answer is in the affirmative; otherwise, not.

3.3. DP

We have noted above that in our approach $\mathbf{x}$, $h(\mathbf{x})$ and $\mathbf{K}$ correspond to specific contextual entities or attributes. For example, in the scheduling context, $\mathbf{x}$ represents the partition that yields a schedule, $h(\mathbf{x})$ the cost of that schedule, and K the cost of an extreme schedule where all jobs are placed in a single partition. This holds for a partial assignment $\mathbf{x}_k$ as much as it does for a full assignment $\mathbf{x}$. This also necessitates us to define K_k for a k-variable assignment such that $h(\mathbf{x}_k) = K_k + h(\mathbf{x}_k)$ and $K_k = h(\mathbf{0}_k)$ for all k, $0 \leq k \leq n$. Returning to the scheduling context again, $\mathbf{x}_k$ represents a partial schedule involving jobs indexed 1 through k, $h(\mathbf{x}_k)$ the cost of that schedule, and K_k the cost of scheduling all these k jobs similarly. Finally, let K_n be synonymous with K and let $K_0 = 0$.

We are now ready to describe the DP. First, suppose that we know an upper bound on H^* and call it UBH as before; we have already discussed the computation of UBH. (Recognize that K is always a choice for UBH, at least initially.) The DP in this case works much like the one in Section 2. Only this time, we use $(\mathbf{x}_k, \langle Q_k, H_k \rangle)$ at stage k to represent a partial assignment $\mathbf{x}_k$ and its $\langle Q_k, H_k \rangle$ pair, where $H_k = K_k + F_k$, and use an alternate thinning criterion. We state a result, almost identical to Lemma 1, that will help us identify and retain a set of partial assignments at each stage such that one of them will provably lead to an optimal full assignment $\mathbf{x}_n^*$.

Lemma 2. *Given* $(\mathbf{x}_k, \langle Q_k, H_k \rangle)$ *and* $(\mathbf{x}'_k, \langle Q'_k, H'_k \rangle)$ *at stage* k *of the DP such that* $Q_k \leq Q'_k$ *and* $H_k \leq H'_k$*, it is sufficient to retain only* $(\mathbf{x}_k, \langle Q_k, H_k \rangle)$ *for further enumeration.*

Proof. As in the proof of Lemma 1, consider the optimal completion of $\mathbf{x}'_k$ to $\mathbf{x}'_n$ and the identical completion of $\mathbf{x}_k$ to $\mathbf{x}_n$. We have:

$$H'_n = H'_k + \left[(K_n - K_k) - \sum_{k+1\le i\le n} p_i x'_i + \sum_{k+1\le i<j\le n} q_i r_j x'_i x'_j\right] + Q'_k\left(\sum_{k+1\le i\le n} r_i x'_i\right), \quad \text{and}$$

$$H_n = H_k + \left[(K_n - K_k) - \sum_{k+1\le i\le n} p_i x'_i + \sum_{k+1\le i<j\le n} q_i r_j x'_i x'_j\right] + Q_k\left(\sum_{k+1\le i\le n} r_i x'_i\right).$$

This leads to:

$$\begin{aligned} H_n - H'_n &= (H_k - H'_k) + (Q_k - Q'_k)\left(\sum_{k+1\le i\le n} r_i x'_i\right) \\ &\le 0 \quad \text{(as per the condition of the lemma).} \end{aligned}$$

We see that the partial assignment $\mathbf{x}'_k$ does not lead to a full assignment that is better than what we can get by completing $\mathbf{x}_k$. We can thus discard $(\mathbf{x}'_k, \langle Q'_k, H'_k\rangle)$. □

Letting $\mathbf{\Omega}_k$ be the set of $(\mathbf{x}_k, \langle Q_k, H_k\rangle)$ considered during stage k of the DP, we can state the following procedure.

Procedure DP_Min H:
Step 0: Set $\mathbf{\Omega}_0 = \{(\emptyset, \langle 0, 0\rangle)\}$.
Step 1: For $k = 1$ through n:

(a) For each $(\mathbf{x}_{k-1}, \langle Q_{k-1}, H_{k-1}\rangle) \in \mathbf{\Omega}_{k-1}$, add to $\mathbf{\Omega}_k$: $(\mathbf{x}_{k-1} \oplus 0, \langle Q_{k-1}, K_k - K_{k-1} + H_{k-1}\rangle)$ always, and $(\mathbf{x}_{k-1} \oplus 1, \langle Q_{k-1} + q_k, K_k - K_{k-1} + H_{k-1} - p_k + r_k Q_{k-1}\rangle)$ only if $-p_k + r_k Q_{k-1} < 0$.
(b) Delete from $\mathbf{\Omega}_k$ all $(\mathbf{x}_k, \langle Q_k, H_k\rangle)$ with $H_k >$ UBH.
(c) For all $(\mathbf{x}_k, \langle Q_k, H_k\rangle) \in \mathbf{\Omega}_k$ with the same H_k, retain one with the smallest Q_k.

Step 2: From $\mathbf{\Omega}_n$, find a member $(\mathbf{x}_n, \langle Q_n, H_n\rangle)$ with the minimum H_n.

Theorem 4. DP_Min H *solves the h-minimization problem in O(nUBH) time.*

Proof. The correctness of DP_Min H is easily established using arguments almost identical to those used in the proof of Theorem 1, basing them this time on Lemma 2 rather than Lemma 1. The only additional thing to note here is that $(\mathbf{x}_k, \langle Q_k, H_k\rangle)$ is discarded if $H_k = h(\mathbf{x}_k) >$ UBH. Since by assumption $h(\mathbf{x}_n) \ge h(\mathbf{x}_k)$ for $k \le n$, we will have in this case $h(\mathbf{x}_n) >$ UBH for any full assignment $\mathbf{x}_n$ obtained from $\mathbf{x}_k$. The completion of $\mathbf{x}_k$ will not thus lead to an optimal solution and $\mathbf{x}_k$ can thus be discarded.

The time complexity of DP_Min H this time around is seen to be O(nUBH), because of the change in the thinning criterion w.r.t. DP_Min F. □

3.4. FPTAS

We can now describe an approximation scheme for h-minimization. Let LBH be a lower bound on H^*; clearly, LBH < UBH. Also, let $\text{UBH}_k = \max\{H_k\}$, where the maximum is taken over all $(\mathbf{x}_k, \langle Q_k, H_k\rangle) \in \mathbf{\Omega}_k$ at the end of Step 1(b) in εAPX_Min H which follows (this step is the same as in DP_Min H). With $\Delta = (\varepsilon\text{LBH})/n$, the basic procedure (which delivers a solution value within εLBH of the optimal solution value H^*) is as follows.

Procedure ε **APX_Min** H:
[All steps are identical to Procedure DP_Min H except Step 1(c). Replace as follows.]
Step1′(c):

(i) Divide the interval $[0, \text{UBH}_k]$ into subintervals of width Δ.
(ii) From all $(\mathbf{x}_k, \langle Q_k, F_k\rangle) \in \mathbf{\Omega}_k$ with H_k in the same subinterval, retain one with the smallest Q_k.

Theorem 5. εAPX_Min H *produces in* $O(\beta n^2/\varepsilon)$ *time a solution for the h-minimization problem with value within* ε *LBH of the optimal solution value* H^**, where* $\beta \ge$ UBH/LBH.

Proof. The number of subintervals at stage k of εAPX_Min H is bounded by $\lceil \text{UBH}_k/\Delta \rceil$. Substituting for Δ, we have:

$$\lceil \text{UBH}_k/\Delta \rceil \leq (n/\varepsilon)(\text{UBH}_k/\text{LBH}) + 1 \leq (n/\varepsilon)(\text{UBH}/\text{LBH}) + 1$$

(the latter inequality follows from Step 1(b) in εAPX_Min H or DP_Min H).

Clearly, at most one pair is retained in each subinterval. The cardinality of $\mathbf{\Omega}_k$ at the end of Step 1′(c) is thus $O(\beta n/\varepsilon)$ for some β such that UBH/LBH $\leq \beta$. We can thus say that εAPX_Min H has a time complexity of $O(\beta n^2/\varepsilon)$ over n stages.

Using arguments almost identical to those used in the proof of Theorem 2 but basing them this time on the decomposition of $h(\mathbf{x}_n)$ (see the proof of Lemma 2), it can be shown that the maximum error at stage k is bounded above by Δ or $(\varepsilon\text{LBH})/n$. This implies that the maximum cumulative error over n stages is bounded above by εLBH.

As in the proof of Theorem 4, note that $(\mathbf{x}_k, \langle Q_k, H_k \rangle)$ is discarded here also when $H_k = h(\mathbf{x}_k) > \text{UBH}$. It has been shown that, if this happens, we will have $h(\mathbf{x}_n) > \text{UBH}$ for any full assignment $\mathbf{x}_n$ obtained from $\mathbf{x}_k$. Supposing that $h(\mathbf{x}_n) - h(\mathbf{x}_n^*) \leq \varepsilon\text{LBH}$ and letting $\mathbf{x}_n^{hu}$ be the assignment such that $h(\mathbf{x}_n^{hu}) = \text{UBH}$, we see that $h(\mathbf{x}_n^{hu}) - h(\mathbf{x}_n^*) \leq \varepsilon\text{LBH}$ as well. In other words, the maximum total error remains bounded above by εLBH. □

εAPX_Min H is not an FPTAS in general. It becomes as such if we know a β such that $\beta \sim \text{poly}(S)$. We use WCT as an example.

In WCT, there are n jobs with processing times $t_1, \ldots, t_n$ and weights $w_1, \ldots, w_n$ that are to be scheduled on 2 identical machines such that $\sum_i w_i C_i$, where C_i is the completion time of job i is minimized. Sahni [15] has given an $O(n^2/\varepsilon)$ FPTAS for this NP-hard problem. The key here is that we have UBH $= h(\mathbf{0}) = K$ and LBH $= h(\mathbf{0})/2$, which are both valid bounds on H^*. Since $\beta = 2$ in this case, εAPX_Min H yields an $O(n^2/\varepsilon)$ FPTAS as well.

When we do not know a lower bound on H^* that is polynomially related to a known upper bound as above, we can search for it by progressively halving the upper bound, seeding the process with an initial UBH (for which K is always a choice) and an initial LBH (for which $1/\varepsilon$ is always a choice). (Note that we can check if $H^* > 1/\varepsilon$ in $O(n/\varepsilon)$ time by using DP_Min H with UBH $= 1/\varepsilon$. If this is true, DP_Min H will not return a solution; otherwise, the solution it will return will be optimal.) With this set-up, a pair of proper upper and lower bounds (both of which are valid and whose ratio is a constant) can be found in at most $\log(\varepsilon K)$ steps. In what follows, we assume w.l.o.g. that $0 < \varepsilon \leq 1$ (note here that, an ε-approximate solution is also an ε'-approximate solution for $\varepsilon < \varepsilon'$). The search procedure is given below, with UBH_{init} and LBH_{init} being the initial values assigned to UBH and LBH.

Procedure SEARCH:
Step 0: Set LBH $= \text{UBH}_{\text{init}}/4$, UBH $= \text{UBH}_{\text{init}}$, BESTH $= \text{UBH}_{\text{init}}$ and VALID $=$ *false*.
Step 1: Do While (VALID $=$ *false*):

(a) Invoke Procedure εAPX_Min H; let APXH be the solution value delivered.
(b) Set BESTH $= \min\{\text{BESTH}, \text{APXH}\}$.
(c) If BESTH $\geq (1+\varepsilon)$ LBH or LBH $\leq \text{LBH}_{\text{init}}$, set VALID $=$ *true*.
Else: LBH $\leftarrow$ LBH/2 and UBH $\leftarrow$ UBH/2.

Step 2: Deliver BESTH and the associated assignment $\mathbf{x}^{h\varepsilon}$ as the solution.

Theorem 6. *SEARCH produces in $O(n^2 \log(\beta)/\varepsilon)$ time an ε-approximate solution to the h-minimization problem and is thus an FPTAS, where* $\beta \geq \text{UBH}_{\text{init}}/\text{LBH}_{\text{init}}$.

Proof. Note that UBH/LBH $= 4$ always. εAPX_Min H, when invoked, thus runs in $O(n^2/\varepsilon)$ time. We halve UBH_{init} at most $\log(\text{UBH}_{\text{init}}/\text{LBH}_{\text{init}})$ times, since

$$\log(\text{UBH}_{\text{init}}/\text{LBH}_{\text{init}}) \geq \max\{i : \text{UBH}_{\text{init}}/2^{i+1} \geq \text{LBH}_{\text{init}}\}.$$

Thus, in the worst case, Step 1 of SEARCH executes $O(\log(\beta))$ times. The overall time complexity of SEARCH can therefore be stated as $O(n^2 \log(\beta)/\varepsilon)$.

To see that SEARCH yields an ε-approximate solution, note that, in Step 0, UBH is a valid upper bound but LBH has not been verified as such. Assume that this is still the case until the start of the ith pass of SEARCH. Since UBH is valid, in Step 1(a), εAPX_Min H will always deliver a solution and that solution will have a value such that APXH $\leq H^* + \varepsilon$LBH. Thus, in Step 1(b), we will have: BESTH $\leq$ APXH $\leq H^* + \varepsilon$LBH. In Step

1(c), if BESTH $\geq (1+\varepsilon)$LBH, then we will have: LBH $+ \varepsilon$LBH $\leq$ BESTH $\leq H^* + \varepsilon$LBH; it immediately follows that LBH $\leq H^*$ and thus that LBH is valid. Similarly, LBH is also valid if LBH $\leq$ LBH$_{\text{init}}$. The procedure can now terminate and deliver $\mathbf{x}^{h\varepsilon}$ as the solution and BESTH as the solution value. However, in Step 1(c), if BESTH $< (1+\varepsilon)$LBH, LBH will remain unvalidated and we will thus need to go through at least the $(i+1)$th pass. Notice that at this point BESTH < 2 LBH (as $\varepsilon \leq 1$). The current UBH $= 4$ LBH and it will remain valid when we halve it for the $(i+1)$th pass, where LBH will also be halved to maintain UBH/LBH $= 4$. The process continues in this manner until LBH is validated (remember that UBH is always valid) and that happens in a finite number of passes (bounded by $\log(\beta)$ as shown above). □

We have seen that SEARCH provides an $O(n^2 \log(\beta)/\varepsilon)$ FPTAS for h-minimization in general (for the context-dependent case). If we let UBF$_{\text{init}} = K$ and LBF$_{\text{init}} = 1/\varepsilon$, the FPTAS is $O(n^2 \log(\varepsilon K)/\varepsilon)$. We now see how it applies to three recently studied scheduling problems — WET, CONT and AWCTV.

In WET, there are n jobs with processing times $t_1, \ldots, t_n$ and weights $w_1, \ldots, w_n$, that are to be scheduled on a single machine such that $\sum_i w_i |C_i - d|$, where C_i is the completion time of job i and d is an unrestrictively large due-date, is minimized. The most competitive FPTAS for WET is due to Kovalyov and Kubiak [10] and has been developed outside of the HP framework. Assuming that $n \leq \max_i\{t_i, w_i\}$ (otherwise the problem is solvable in polynomial time), this FPTAS runs in $O(n^2 \log^3(\max_i\{1/\varepsilon, t_i, w_i\})/\varepsilon^2)$ time. Noting that in this case $K = h(\mathbf{0}) = \sum_{1\leq i\leq n} w_i(\sum_{1\leq j\leq i} t_j)$ and because $\varepsilon \leq 1$, it can be stated that our general FPTAS runs in $O(n^2 \log(\max_i\{t_i, w_i\})/\varepsilon)$ time.

As for CONT: there are n jobs; the processing time t_i of job i can be varied within the interval [0, u_i] at a unit compression cost of v_i; there is a unit completion time cost of w_i; and, the objective is to determine what value to assign to each t_i and how to schedule the jobs on a single machine such that $\sum_i w_i C_i + \sum_i v_i(u_i - t_i)$ is minimized (C_i as usual is the completion time of job i). The only FPTAS, that we know of, are due to Janiak et al. [7] and run in $O(n^2 \log(\max_i\{u_i\})/\varepsilon)$ and $O(n^2 \log(\max_i\{w_i\})/\varepsilon)$, respectively. Note that, in this case, $K = h(\mathbf{0}) = \sum_{1\leq i\leq n} u_i v_i$; thus, our FPTAS runs in $O(n^2 \log(\max_i\{u_i, v_i\})/\varepsilon)$ time.

We now address AWCTV, which involves n jobs, with processing times $t_1, \ldots, t_n$ and weights $w_1, \ldots, w_n$ satisfying $t_i < t_j \Rightarrow w_i \geq w_j$ for any i and j, that are to be scheduled on a single machine such that $[\sum_i w_i(C_i - C^w_{\text{avg}})^2]/W$, where C_i is the completion time of job i, $C^w_{\text{avg}} = [\sum_i w_i C_i]/W$ and $W = \sum_i w_i$, is minimized. The most competitive FPTAS for AWCTV is due to Cheng and Kubiak [2] and runs in $O(n^4 \log(\max_i\{t_i, w_i\})/\varepsilon)$ time. The expression for K is rather complicated here (see Theorem 1 of [2] and the computations leading to it). Upon examination of this expression, however, it can be stated that our general FPTAS runs in $O(n^2 \log(\max_j\{t_j, w_j\})/\varepsilon)$ time in this case.

Finally, we show how we can obtain two progressively better FPTAS for AWCTV by using a lower bound on H^*, call it LB$_{CK}$, available from [2] for this problem. It has been shown there that $K \leq 4n^2 LB_{CK}$. First, we can set LBF $=$ LB$_{CK}$ and UBF $= K$ in εAPX_Min H so that $\beta = 4n^2$. We immediately get an $O(n^4/\varepsilon)$ FPTAS for AWCTV. Second, we can set LBF$_{\text{init}}$ $=$ LB$_{CK}$ and UBF$_{\text{init}} = K$ in SEARCH and get an $O(n^2 \log(n)/\varepsilon)$ FPTAS, which is a significant improvement over what exists for AWCTV or what we have proposed thus far.

3.5. Remarks

1. In DP_Min H, Lemma 2 allows us to thin the state-space based either on Q or on H. We thin on H, as it enables us to check if $H^* = 0$ or $H^* > 1/\varepsilon$ in a desired time order. Incidentally, with slight modifications to DP_Min H (maintaining Ω_k such that its members are distinct in both coordinates of the $\langle\cdot,\cdot\rangle$ pair), we get an $O(n \min\{Q, UBH\})$ time DP.

2. As with f-minimization, there is an alternate DP for h-minimization, which runs in $O(nR)$ time; this one can be used as well for developing an FPTAS [7].

3. We add that SEARCH can also be implemented such that LBF is always valid but UBF is not. In that case, LBH is repeatedly doubled until UBH becomes valid.

4. Finally, our approach is flexible enough so that it can be adapted to solve many scheduling problems, that admit a strong dominance condition like Lemma 2, within our framework, whether or not they are amenable to an explicit HP representation (see [15]).

4. Conclusion

To conclude, we reiterate that certain partitioning type NP-hard problems, including many from the field of scheduling, can be cast as half-products minimization (specifically, h-minimization). Several such examples can be found in [12]. In this note, we have supplemented the work of other researchers by showing that it is often possible to give an $O(n^2/\varepsilon)$ and always an $O(n^2 \log(\varepsilon K)/\varepsilon)$ FPTAS for this class of NP-hard problems in a given context (where certain mild assumptions can be made regarding the sign and the properties of $f(\mathbf{x})$). Our general approach has led to FPTAS for specific problems that are comparable to and remarkably competitive with what exists in the literature at present.

References

[1] T. Badics, E. Boros, Minimization of half-products, Mathematics of Operations Research 23 (1998) 649–660.
[2] J. Cheng, W. Kubiak, A half-product based approximation scheme for agreeably weighted completion time variance, European Journal of Operational Research 162 (2005) 45–54.
[3] E. Erel, J.B. Ghosh, FPAS for half-products minimization, Discussion Paper 2002–03, Faculty of Business Administration, Bilkent University, Turkey, 2002.
[4] G.V. Gens, E.V. Levner, Fast approximation algorithm for job sequencing with deadlines, Discrete Applied Mathematics 3 (1981) 313–318.
[5] O. Ibarra, C.E. Kim, Fast approximation algorithms for the knapsack and sum of subset problems, Journal of the ACM 22 (1975) 463–468.
[6] A. Janiak, M. Winczaszek, A single processor scheduling problem with a common due window assignment, in: H. Fleuren, D. den Hertog, P. Kort (Eds.), Operations Research Proceedings 2004, Springer Berlin Heidelberg, 2005, pp. 213–220.
[7] A. Janiak, M.Y. Kovalyov, W. Kubiak, F. Werner, Positive half-products and scheduling with controllable processing times, European Journal of Operational Research 165 (2005) 416–422.
[8] B. Jurisch, W. Kubiak, J. Jozefowska, Algorithms for minclique scheduling problems, Discrete Applied Mathematics 72 (1997) 115–139.
[9] M.Y. Kovalyov, Improving the complexities of approximation algorithms for optimization problems, Operations Research Letters 17 (1995) 85–87.
[10] M.Y. Kovalyov, W. Kubiak, A fully polynomial approximation scheme for the weighted earliness-tardiness problem, Operations Research 47 (1999) 757–761.
[11] W. Kubiak, New results on the completion time variance minimization, Discrete Applied Mathematics 58 (1995) 157–168.
[12] W. Kubiak, Minimization of ordered, symmetric half-products, Discrete Applied Mathematics 146 (2005) 287–300.
[13] W. Kubiak, J. Cheng, M.Y. Kovalyov, Fast fully polynomial approximation schemes for minimizing completion time variance, European Journal of Operational Research 137 (2002) 303–309.
[14] E.L. Lawler, J.M. Moore, A functional equation and its application to resource allocation and sequencing problems, Management Science 16 (1969) 77–84.
[15] S. Sahni, Algorithms for scheduling independent tasks, Journal of the ACM 23 (1976) 116–127.
[16] N.V. Shakhlevich, V.A. Strusevich, Single machine scheduling with controllable release and processing parameters, Discrete Applied Mathematics 154 (2006) 2178–2199.
[17] G.J. Woeginger, An approximation scheme for minimizing agreeably weighted variance on a single machine, INFORMS Journal on Computing 11 (1999) 211–216.