A Fast FPTAS for Single Machine Scheduling Problem of Minimizing Total Weighted Earliness and Tardiness about a Large Common Due Date

Hans Kellerer
Institute für Statistik und Operations Research,
Universität Graz, Universitätsstraße 15, A-8010, Graz, Austria
Email: hans.kellerer@uni-graz.at

Kabir Rustogi and Vitaly A. Strusevich
Department of Mathematical Sciences, University of Greenwich,
Old Royal Naval College, Park Row, Greenwich, London SE10 9LS, U.K.
E-mail: {K.Rustogi,V.Strusevich}@greenwich.ac.uk

Abstract

We address the single machine scheduling problem to minimize the total weighted earliness and tardiness about a nonrestrictive common due date. This is a basic problem with applications to the just-in-time manufacturing. The problem is linked to a Boolean programming problem with a quadratic objective function, known as the half-product. An approach to developing a fast fully polynomial-time approximation scheme (FPTAS) for the problem is identified and implemented. The running time matches the best known running time for an FPTAS for minimizing a half-product with no additive constant.

Keywords: Single machine scheduling; Earliness-Tardiness; Half-Product Problem; FPTAS

1 Introduction

In just-in-time manufacturing, the management is looking for a production plan such that all orders are completed as close to the agreed due dates as possible. It is viewed equally undesirable to achieve either an early completion or a late completion of an order. Earliness normally generates penalties related to holding the completed orders before their delivery. On the other hand, lateness implies a late delivery which not only results into possible fees but also might affect the image of the manufacturer as a reliable partner.

Problems related to just-in-time scheduling have been an object of extensive research since the early 1990s. In scheduling terms, an order is seen as a job and the manufacturing facilities as machines. Various machine environments have been studied, from a single machine to parallel machines and to various shop models.

In this paper, we address one of the basic and most known problem of just-in-time scheduling. The jobs of set $N = \{1, 2, \dots, n\}$ have to be processed without preemption on a single machine. The processing of job $j \in N$ takes p_j time units. There is a positive weight w_j associated with job j. All values p_j and w_j are positive integers. The machine processes at most one job at a time. The total processing time of all jobs is denoted by $p(N) = \sum_{j=1}^{n} p_j$.

For a schedule S, let $C_j(S)$ denote the completion time of job $j \in N$. The jobs have a common due date d. A job j is said to be *early* if $C_j(S) - d \leq 0$, and its *earliness* is defined as $E_j(S) = d - C_j(S)$. On the other hand, a job is said to be *late* if $C_j(S) - d > 0$, and its *tardiness* is defined as $T_j(S) = C_j(S) - d$. The aim is to find a schedule that minimizes total weighted earliness and tardiness, i.e., the function $\sum_{j \in N} w_j (E_j + T_j)$. Here w_j is understood as the cost of having job j early or late for one time unit. Notice that for job j the same weight w_j is applied, no matter the job is late or early.

There is a considerable interest in problems on just-in-time scheduling with alternative objective functions; see, e.g., [16, 21] where the objective is to minimize the weighted number of early and late jobs. Still, the problems with the earliness-tardiness criterion are seen as most important for just-in-time manufacturing, since where the weighted earliness is directly related to holding costs and the weighted tardiness represents a penalty for a late delivery.

If $p(N) \leq d$, the due date is called *large* or *nonrestrictive*; otherwise, for $p(N) > d$, the due date is called *small* or *restrictive.* Denote the problem to minimize the total weighted earliness-tardiness by $1|d_j = d, p(N) \leq d| \sum w_j(E_j + T_j)$ if the due date is large and by $1|d_j = d, p(N) > d| \sum w_j(E_j + T_j)$ if the due date is small.

Problem $1|d_j = d, p(N) \leq d| \sum w_j(E_j + T_j)$ is known to be solvable in $O(n \log n)$ time, provided that the weights are equal; otherwise, it is NP-hard in the ordinary sense and admits a pseudopolynomial dynamic programming algorithm. We may assume that

$$n \leq \max_{j \in N} \{p_j, w_j\}, \tag{1}$$

since otherwise the dynamic programming algorithm takes polynomial time. If for job $j \in N$ the weight for earliness is w_j^E and the weight for tardiness is w_j^T, then it is still open whether the resulting problem with the objective $\sum \left(w_j^E E_j + w_j^T T_j \right)$ is NP-hard in the strong sense or is solvable in pseudopolynomial time. See the paper by Hall and Posner [3] for a detailed study of the problems with symmetric and asymmetric weights.

A major direction of research on scheduling problems with a common due date is design of fully polynomial-time approximation schemes (FPTAS); see the review paper by Kacem et al. [6]. Approximation schemes that require strongly polynomial time are of special interest.

For a collection of decision variables $\mathbf{x}$, consider a problem of minimizing a function $\Phi(\mathbf{x})$, with a positive optimal value $\Phi(\mathbf{x}^*)$. Let $\mathbf{x}^H$ denote a feasible solution found by a heuristic algorithm. A polynomial-time algorithm that finds such a solution $\mathbf{x}^H$ that the inequality $\Phi(\mathbf{x}^H)/\Phi(\mathbf{x}^*) \leq \rho$ holds for all instances of the problem is called a *$\rho-$approximation* algorithm and $\rho \geq 1$ is called the *worst-case ratio bound.* A family of $\rho-$approximation algorithms is called a *fully polynomial-time approximation scheme (FPTAS)* if $\rho = 1 + \varepsilon$ for any $\varepsilon > 0$ and the running time is polynomial with respect to both the length of the problem input and $1/\varepsilon$. A special attention is paid to the design of FPTASs that require *strongly polynomial* running time, i.e., time bounded by a polynomial that depends on n and $1/\varepsilon$ only.

The running time of the first FPTAS for problem $1|d_j = d, p(N) \leq d| \sum w_j(E_j + T_j)$ developed by Kovalyov and Kubiak [13] is $O(n^2 \log^3 \left(\max\{p_j, w_j, n, 1/\varepsilon\} / \varepsilon^2 \right)$, or under the assumption (1), is $O(n^2 \log^3 \left(\max\{p_j, w_j, 1/\varepsilon\} / \varepsilon^2 \right)$. The best FPTAS known prior to this paper requires $O(n^2 \log (\max\{p_j, w_j\} / \varepsilon)$ as proved by Erel and Ghosh [2]. Notice that for a less traditional measure of quality of an approximate solution, it is shown in [17] that the problem admits a so-called differential FPTAS that requires $O\left(n^2/\varepsilon\right)$ time.

The main result of this paper is an FPTAS for problem $1|d_j = d, p(N) \leq d|\sum w_j(E_j+T_j)$ which requires $O\left(n^2/\varepsilon\right)$ time, which is strongly polynomial unlike the running times of all approximation schemes previously known for the problem. Such a scheme allows solving large-size problems with a chosen accuracy in a reasonable computation time. This improved running time is achieved by reformulating the problem in terms of a Boolean programming problem with a quadratic objective function known as the half-product. Although the FPTAS by Erel and Ghosh [2] is also based on the reduction of problem $1|d_j = d, p(N) \leq d|\sum w_j(E_j+T_j)$ to minimizing a half-product function with an additive constant, we achieve the running time that matches the best time known for an FPTAS for a simpler problem of minimizing a half-product function with no additive constant. The improvement is due to applying a combination of methods, most of which have been known but not applied together to handle the scheduling problem under consideration.

The remainder of this paper is organized as follows. In Section 2, we reformulate problem $1|d_j = d, p(N) \leq d|\sum w_j(E_j + T_j)$ as a minimization problem of a special form of a convex half-product with an additive constant and outline an approach to designing an FPTAS that requires $O\left(n^2/\varepsilon\right)$ time, which matches the best known running time of an FPTAS for minimizing the half-product with no additive constant. The implementation details of this approach (which are surprisingly simple) are given in Section 3. Section 4 contains concluding remarks.

2 Half-Product Reformulation

In this section, we establish relations between problem $1|d_j = d, p(N) \leq d|\sum w_j(E_j + T_j)$ and the problem of minimizing a special form of the half-product.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be a vector with n Boolean components. Consider the function

$$H(\mathbf{x}) = \sum_{1\leq i<j\leq n} \alpha_i \beta_j x_i x_j - \sum_{j=1}^{n} \gamma_j x_j, \tag{2}$$

where for each j, $1 \leq j \leq n$, the coefficients α_j and β_j are non-negative integers, while γ_j is an integer that can be either negative or positive.

We refer to the problem of minimizing function $H(\mathbf{x})$ of the form (2), as *Problem HP*. Let a Boolean vector $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*)$ such that inequality $H(\mathbf{x}^*) \leq H(\mathbf{x})$ holds for all Boolean vectors $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be called an optimal solution of Problem HP. Notice that we only are interested in the instances of the problem for which the optimal value of the function is strictly negative; otherwise, setting all decision variables to zero solves the problem.

Problems of minimizing quadratic functions similar to (2) were introduced in the 1990s in [5, 14] as mathematical models for various scheduling problems. Function (2) and the term "half-product" were introduced in [1], where the problem of minimizing function $H(\mathbf{x})$ with respect to Boolean decision variables with no additional constraints was considered. The function $H(\mathbf{x})$ is called a *half-product* since its quadratic part consists of roughly half of the terms of the product $\left(\sum_{j=1}^{n} \alpha_j x_j\right)\left(\sum_{j=1}^{n} \beta_j x_j\right)$. It is proved in [1] that minimizing $H(\mathbf{x})$ is NP-hard in the ordinary sense even if $\alpha_j = \beta_j$ for all $j = 1, 2, \ldots, n$. By contrast, maximizing function $H(\mathbf{x})$ of the form (2) with respect to Boolean decision variables with no additional constraints requires $O\left(n^3\right)$ time, as demonstrated in [7].

Using the fact that for Boolean variables $x_j^2 = x_j$, $j \in N$, we can rewrite (2) as

$$H(\mathbf{x}) = \sum_{1 \le i \le j \le n} \alpha_i \beta_j x_i x_j - \sum_{j=1}^{n} (\gamma_j + \alpha_j \beta_j) x_j,$$

and the quadratic term in the above expression can be represented as

$$\sum_{1 \le i \le j \le n} \alpha_i \beta_j x_i x_j = \frac{1}{2}\mathbf{x}^T \mathbf{G}\mathbf{x} + \frac{1}{2}\sum_{j=1}^{n} \alpha_j \beta_j x_j,$$

where

$$\mathbf{G} = \begin{bmatrix} \alpha_1\beta_1 & \alpha_1\beta_2 & \cdots & \alpha_1\beta_n \\ \alpha_1\beta_2 & \alpha_2\beta_2 & \cdots & \alpha_2\beta_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1\beta_n & \alpha_2\beta_n & \cdots & \alpha_n\beta_n \end{bmatrix}. \tag{3}$$

The following statement follows from [18].

Theorem 1 *Matrix* $\mathbf{G}$ *of the form (3) is positive semidefinite and function* $H(\mathbf{x})$ *is convex, provided that the items are numbered so that*

$$\frac{\alpha_1}{\beta_1} \le \frac{\alpha_2}{\beta_2} \le \ldots \le \frac{\alpha_n}{\beta_n}. \tag{4}$$

Problem HP and its variants have numerous applications, mainly to machine scheduling. Notice that in those applications a scheduling objective function usually is written in the form

$$F(\mathbf{x}) = H(\mathbf{x}) + K, \tag{5}$$

where K is a given additive constant. We refer to the problem of minimizing function $F(\mathbf{x})$ of the form (5), as *Problem HPAdd.* Notice that normally K is a fairly large positive constant such that $F(\mathbf{x}^*) > 0$. Moreover, in all known scheduling applications function $F(\mathbf{x})$ is convex; see surveys by Kellerer and Strusevich [10, 12] for comprehensive discussions of relevant scheduling applications.

For problem $1|d_j = d, p(N) \le d| \sum w_j(E_j + T_j)$, assume that the jobs are numbered in such a way that

$$\frac{p_1}{w_1} \le \frac{p_2}{w_2} \le \ldots \le \frac{p_n}{w_n}. \tag{6}$$

The sequence of jobs numbered in accordance with (6) is called a *Smith* sequence or a *WSPT* sequence (Weighted Shortest Processing Time). In an optimal schedule for the classical single machine problem of minimizing the sum of the weighted completion times $\sum_{j \in N} w_j C_j$, the jobs are processed according to the WSPT sequence.

The following lemma proved by Hall and Posner [3] describes a possible structure of an optimal schedule for problem $1|d_j = d, p(N) \le d| \sum w_j(E_j + T_j)$.

Lemma 1 *For problem* $1|d_j = d, p(N) \le d| \sum w_j(E_j + T_j)$ *there exists an optimal schedule in which some job completes exactly at time d, i.e., it has neither earliness nor tardiness. There is no intermediate idle time in job processing, but some idle time may occur before the first early job. The jobs that start either at or after the due date are processed in the order of their numbering by the WSPT rule. The early jobs are processed in the order opposite to their numbering.*

Based on Lemma 1, Kubiak [15] and Kellerer and Strusevich [9] show how to reformulate problem $1|d_j = d, p(N) \leq d| \sum w_j(E_j + T_j)$ as Problem HPAdd. Introduce Boolean decision variables

$$x_j = \begin{cases} 1, & \text{if job } j \text{ completes by the due date } d \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Considering the jobs in the order of their numbering, it follows that the earliness of job j that completes by time d and the tardiness of job j that starts after the due date can be written as

$$E_j = \sum_{i=1}^{j-1} p_i x_i \text{ and } T_j = \sum_{i=1}^{j} p_i(1 - x_i),$$

respectively, so that the objective function can be written as

$$\sum_{j=1}^{n} w_j(E_j + T_j) = \sum_{1 \leq i < j \leq n} p_i w_j x_i x_j + \sum_{1 \leq i < j \leq n} p_i w_j (1 - x_i)(1 - x_j) + \sum_{j=1}^{n} p_j w_j (1 - x_j). \tag{8}$$

In turn, (8) can be rewritten as

$$\sum_{j=1}^{n} w_j(E_j + T_j) = 2 \sum_{1 \leq i < j \leq n} p_i w_j x_i x_j - \sum_{j=1}^{n} \left(w_j \left(\sum_{i=1}^{j-1} p_i \right) + p_j \left(\sum_{i=j}^{n} w_i \right) \right) x_j + \sum_{1 \leq i \leq j \leq n} p_i w_j.$$

Thus, problem $1|d_j = d, p(N) \leq d| \sum w_j(E_j + T_j)$ reduces to Problem HPAdd with

$$\alpha_j = 2p_j,\ \beta_j = w_j,\ \gamma_j = w_j \left(\sum_{i=1}^{j-1} p_i \right) + p_j \left(\sum_{i=j}^{n} w_i \right),\ j \in N;\ K = \sum_{1 \leq i \leq j \leq n} p_i w_j.$$

The objective function is convex, which follows from Theorem 1 and the adopted numbering of jobs (6).

We now pass to discussing the issues related to the design of approximation schemes for problems of minimizing the half-product and its variants. Problem HP admits an FPTAS that requires $O\left(n^2/\varepsilon\right)$ time, as proved by Erel and Ghosh [2]. Notice that algorithms that behave as an FPTAS for Problem HP (without an additive constant) do not necessarily deliver an $(1+\varepsilon)$-approximate solution for Problem HPAdd of minimizing a function (5) which includes an additive constant. This fact has been pointed out by many researchers, see, e.g., [4] by Janiak et al. and [2] by Erel and Ghosh. For Problem HPAdd and its special cases it is especially challenging to develop an FPTAS of the running time $O\left(n^2/\varepsilon\right)$, which matches that the best known for the problem with no additive constant in the objective. The time $O\left(n^2/\varepsilon\right)$ can be achieved for various variants of the problem, but still not for the general case. Notice that according to the most recent survey in the area by Kellerer and Strusevich [12], problem $1|d_j = d, p(N) \leq d| \sum w_j(E_j + T_j)$ is the only scheduling problem that reduces to Problem HPAdd but is not known to admit an FPTAS that requires $O\left(n^2/\varepsilon\right)$ time.

Our approach to developing a fast FPTAS for problem $1|d_j = d, p(N) \leq d| \sum w_j(E_j + T_j)$ is based on the reduction of the problem to minimizing a special form of the half-product function. The objective function (8) is a special case of a so-called symmetric quadratic function given below

$$Z(\mathbf{x}) = \sum_{1 \leq i < j \leq n} \alpha_i \beta_j x_i x_j + \sum_{1 \leq i < j \leq n} \alpha_i \beta_j (1 - x_i)(1 - x_j) + \sum_{j=1}^{n} \mu_j x_j + \sum_{j=1}^{n} \nu_j (1 - x_j) + K, \tag{9}$$

where all coefficients are non-negative. To see the relation between (9) and (8), define $K = 0$ and $\alpha_j = p_j$, $\beta_j = w_j$, $\mu_j = 0$, $\nu_j = p_j w_j$, $j \in N$.

The problem of minimizing function (9), with an additional linear knapsack constraint $\sum_{j=1}^{n} \alpha_j x_j \leq A$, is studied in [8, 9, 20]. However, the resulting approximation schemes require more time than $O\left(n^2/\varepsilon\right)$ and cannot be adapted for our purposes.

The approach that is employed in this paper is based on the following theorem that is due to [2] by Erel and Ghosh.

Theorem 2 *Suppose that for Problem HPAdd of minimizing a function $F(\mathbf{x})$ of the form (5), the lower and upper bounds on the value of $F(\mathbf{x}^*)$, i.e., the values F_{LB} and F^{UB} such that $F_{LB} \leq F(\mathbf{x}^*) \leq F^{UB}$, can be found in $T(n)$ time. If the ratio F^{UB}/F_{LB} is bounded from above by some ρ, then there exists an algorithm that delivers a solution $\mathbf{x}^0$ such that $F(\mathbf{x}^0) - F_{LB} \leq \varepsilon F_{LB}$ in $O(T(n) + \rho n^2/\varepsilon)$ time.*

If the required lower and upper bounds F_{LB} and F^{UB} are provided, the algorithm presented in [2] by Erel and Ghosh delivers a $(1+\varepsilon)$-solution. Moreover, if both $T(n)$ and ρ are bounded from above by a polynomial of the length of the input of the problem, then the algorithm guaranteed by that theorem behaves as an FPTAS. Theorem 2 has been used for deriving FPTASs for several scheduling problems that can be reformulated as Problem HPAdd; see [2] by Erel and Ghosh and [10, 12] by Kellerer and Strusevich. In particular, an FPTAS for problem $1|d_j = d, p(N) \leq d| \sum w_j(E_j + T_j)$ with the best running time of $O(n^2 \log(\max\{p_j, w_j\}/\varepsilon)$ known so far is due to Erel and Ghosh [2] and is based on Theorem 2 and its refinement.

To apply Theorem 2 to designing an FPTAS for problem $1|d_j = d, p(N) \leq d| \sum w_j(E_j + T_j)$ that runs in $O\left(n^2/\varepsilon\right)$ time, we have to find the required upper and lower bounds on the optimal value of the function in at most $T(n) = O\left(n^2\right)$ time and to demonstrate that the ratio of these bounds is bounded by a constant. Details of a possible implementation of the actions that lead to a required FPTAS are given in the following section.

3 Obtaining an FPTAS

Instead of dealing with an objective function of problem $1|d_j = d, p(N) \leq d| \sum w_j(E_j + T_j)$ directly, we describe how to adapt Theorem 2 to the problem of minimizing a more general symmetric quadratic function $Z(\mathbf{x})$ of the form (9), provided that it is convex. As we know, for problem $1|d_j = d, p(N) \leq d| \sum w_j(E_j + T_j)$ the function (8) is convex and is a special case of $Z(\mathbf{x})$. Despite a considerable volume of research on optimizing the half-product and its variants, the problem of minimizing a symmetric quadratic function $Z(\mathbf{x})$ without a knapsack constraint has not been studied before.

In order to find the upper and lower bounds on the value of the objective function, we solve the continuous relaxation of the problem to get a lower bound followed by an appropriate rounding of the obtained fractional solution to get an upper bound. A similar technique has been used in [9, 11] by Kellerer and Strusevich, where several scheduling problems have been reduced to a Problem HPAdd with a convex objective function $F(\mathbf{x})$, the lower bound F_{LB} has been found by solving the continuous relaxation and the upper bound F^{UB} has been derived by an appropriate rounding of the fractional solution.

The continuous relaxation is obtained from the original Boolean programming formulation by relaxing the integrality constraints and replacing the condition $x_j \in \{0,1\}$ by $0 \le x_j \le 1$, $j = 1, 2, \ldots, n$. If $\mathbf{x}^C = (x_1^C, \ldots, x_n^C)$, $0 \le x_j^C \le 1$, is the corresponding solution vector and $Z(\mathbf{x}^C)$ denotes the optimal value of the function for the continuous relaxation, then clearly $Z(\mathbf{x}^C) \le Z(\mathbf{x}^*)$. The following statement holds for the continuous relaxation of the problem of minimizing a convex function (5).

Theorem 3 *For a convex objective function, the continuous relaxation of Problem HPAdd without and with the linear knapsack constraint $\sum_{j \in N} \alpha_j \le A$ can be solved in $O(n^2)$ time.*

Theorem 3 is proved by Kellerer and Strusevich [9] by reducing the continuous relaxation to finding the minimum cost flow with a convex quadratic cost function in a special network. The latter problem is a special case of the problem studied by Tamir [19] who develops a polynomial-time algorithm for its solution. In the case under consideration, Tamir's algorithm requires $O\left(n^2\right)$ time. Thus, for the problem of minimizing a convex function $Z(\mathbf{x})$ of the form (9) Theorem 3 is applicable, a solution vector $\mathbf{x}^C$ to the continuous relaxation can be found in $O(n^2)$ time, and the value $Z(\mathbf{x}^C)$ can be taken as a lower bound Z_{LB}.

To obtain an upper bound Z^{UB}, we perform an appropriate rounding of the fractional components of vector $\mathbf{x}^C$. Notice that we are not after a particularly small or tight bound ρ on the ratio Z^{UB}/Z_{LB}, as long as ρ is a constant. Thus, it appears that for our purposes a very simple (in fact, the simplest) rounding algorithm is sufficient.

Algorithm SQRound

Step 1. Given a vector $\mathbf{x}^C = (x_1^C, \ldots, x_n^C)$, $0 \le x_j^C \le 1$, a solution to the continuous relaxation of the problem of minimizing function (9), determine the sets $I_1 = \left\{ j \in N,\ x_j^C \le \frac{1}{2} \right\}$ and $I_2 = \left\{ j \in N,\ x_j^C > \frac{1}{2} \right\}$ and find vector $\mathbf{x}^H = (x_1^H, \ldots, x_n^H)$ with components

$$x_j^H = \begin{cases} 0 & \text{if } \ j \in I_1 \\ 1 & \text{if } \ j \in I_2 \end{cases}.$$

Step 2. Output vector $\mathbf{x}^H = (x_1^H, \ldots, x_n^H)$ as heuristic solution for the problem of minimizing function (9).

Algorithm SQRound delivers a solution to the problem of minimizing a symmetric quadratic function (9) by rounding a fractional solution, which is captured in its name. Its running time is $O(n)$. Clearly, the inequalities $Z(\mathbf{x}^C) \le Z(\mathbf{x}^*) \le Z(\mathbf{x}^H)$ hold, i.e., we may take $Z(\mathbf{x}^H)$ as an upper bound Z^{UB} on the optimal value $Z(\mathbf{x}^*)$. We now estimate the ratio $Z^{UB}/Z_{LB} = Z(\mathbf{x}^H)/Z(\mathbf{x}^C)$.

Theorem 4 *Let $\mathbf{x}^C$ be an optimal solution of the continuous relaxation of the problem of minimizing function $Z(\mathbf{x})$ of the form (9), and $\mathbf{x}^H$ be a vector found by Algorithm SQRound. Then*

$$\rho = \frac{Z(\mathbf{x}^H)}{Z(\mathbf{x}^C)} \le 4.$$

Proof: For a vector $\mathbf{x}^C$, let I_1 and I_2 be the index sets found in Step 2 of Algorithm SQRound. For a vector $\mathbf{x} = (x_1, \ldots, x_n)$, where $0 \le x_j \le 1$, define

$$\begin{aligned}
Z_1(\mathbf{x}) &= \sum_{\substack{1\le i<j\le n\\ i,j\in I_1}} \alpha_i\beta_j x_i x_j + \sum_{\substack{1\le i<j\le n\\ i,j\in I_1}} \alpha_i\beta_j (1-x_i)(1-x_j);\\
Z_2(\mathbf{x}) &= \sum_{\substack{1\le i<j\le n\\ i\in I_1, j\in I_2}} \alpha_i\beta_j x_i x_j + \sum_{\substack{1\le i<j\le n\\ i\in I_1, j\in I_2}} \alpha_i\beta_j (1-x_i)(1-x_j);\\
Z_3(\mathbf{x}) &= \sum_{\substack{1\le i<j\le n\\ i\in I_2, j\in I_1}} \alpha_i\beta_j x_i x_j + \sum_{\substack{1\le i<j\le n\\ i\in I_2, j\in I_1}} \alpha_i\beta_j (1-x_i)(1-x_j);\\
Z_4(\mathbf{x}) &= \sum_{\substack{1\le i<j\le n\\ i,j\in I_2}} \alpha_i\beta_j x_i x_j + \sum_{\substack{1\le i<j\le n\\ i,j\in I_2}} \alpha_i\beta_j (1-x_i)(1-x_j);\\
Z_5(\mathbf{x}) &= \sum_{j\in I_1} \mu_j x_j + \sum_{j\in I_1} \nu_j (1-x_j);\\
Z_6(\mathbf{x}) &= \sum_{j\in I_2} \mu_j x_j + \sum_{j\in I_2} \nu_j (1-x_j).
\end{aligned}$$

By the rounding conditions in Step 2 of Algorithm SQRound, we derive

$$Z_2(\mathbf{x}^H) = Z_3(\mathbf{x}^H) = 0,$$

while

$$\begin{aligned}
Z_1(\mathbf{x}^H) &= \sum_{\substack{1\le i<j\le n\\ i,j\in I_1}} \alpha_i\beta_j; & Z_1(\mathbf{x}^C) &\ge \frac{1}{4}\sum_{\substack{1\le i<j\le n\\ i,j\in I_1}} \alpha_i\beta_j;\\
Z_4(\mathbf{x}^H) &= \sum_{\substack{1\le i<j\le n\\ i,j\in I_2}} \alpha_i\beta_j; & Z_4(x^C) &\ge \frac{1}{4}\sum_{\substack{1\le i<j\le n\\ i,j\in I_2}} \alpha_i\beta_j;\\
Z_5(\mathbf{x}^H) &= \sum_{j\in I_1} \nu_j; & Z_5(\mathbf{x}^C) &\ge \frac{1}{2}\sum_{j\in I_1} \nu_j;\\
Z_6(\mathbf{x}^H) &= \sum_{j\in I_2} \mu_j; & Z_5(\mathbf{x}^C) &\ge \frac{1}{2}\sum_{j\in I_2} \mu_j;
\end{aligned}$$

Thus, we have that

$$\begin{aligned}
Z(\mathbf{x}^H) &= \sum_{k=1}^{6} Z_k(\mathbf{x}^H) + K = Z_1(\mathbf{x}^H) + Z_4(\mathbf{x}^H) + Z_5(\mathbf{x}^H) + Z_6(\mathbf{x}^H) + K\\
&\le 4Z_1(\mathbf{x}^C) + 4Z_4(\mathbf{x}^C) + 2Z_5(\mathbf{x}^C) + 2Z_6(\mathbf{x}^C) + K\\
&\le 4\sum_{k=1}^{6} Z_k(\mathbf{x}^C) + 4K = 4Z(\mathbf{x}^C),
\end{aligned}$$

as required. ■

It follows immediately from Theorem 4 that for the problem of minimizing a convex function (9) Theorem 2 is applicable with $T(n) = O\left(n^2\right)$ and $\rho = 4$. Hence, we obtain the following statement.

Theorem 5 *The problem of minimizing a convex symmetric quadratic function (9) admits an FPTAS that requires $O\left(n^2/\varepsilon\right)$ time.*

A direct application of Theorem 5 to problem $1|d_j = d, p(N) \leq d|\sum w_j(E_j + T_j)$ yields the following result.

Theorem 6 *Problem $1|d_j = d, p(N) \leq d|\sum w_j(E_j + T_j)$ of minimizing total weighted earliness and tardiness about a large common due date admits an FPTAS that requires $O\left(n^2/\varepsilon\right)$ time.*

4 Conclusion

The main result of the paper is a fast FPTAS for problem $1|d_j = d, p(N) \leq d|\sum w_j(E_j+T_j)$, i.e., the single machine problem to minimize total weighted earliness and tardiness about a large common due date.

The paper presents yet another evidence that the link between Boolean programming problems with quadratic objective functions related to the half-product is useful for developing fast approximation schemes for scheduling problems.

Prior to this paper, problem $1|d_j = d, p(N) \leq d|\sum w_j(E_j + T_j)$ has remained the only known scheduling problem that can be formulated as Problem HPAdd but the existence of an FPTAS with the running time of $O\left(n^2/\varepsilon\right)$ has not been proved, despite the fact that all ingredients for such an FPTAS have already been in place. The final step of putting known results together is surprisingly simple, but delivers what is required.

The search for other problems, not necessarily from the problem area of scheduling, that admit a representation in terms of the half-product is of interest.

References

[1] Badics T, Boros E. Minimization of half-products. Math Oper Res 1998;33:649–60.

[2] Erel E, Ghosh JB. FPTAS for half-products minimization with scheduling applications. Discr Appl Math 2008;156:3046–56.

[3] Hall NG, Posner ME. Earliness-tardiness scheduling problems, I: weighted deviation of completion times about a common due date. Oper Res 1991;39:836–46.

[4] Janiak A, Kovalyov MY, Kubiak W, Werner F. Positive half-products and scheduling with controllable processing times. Eur J Oper Res 2005;165:416–22.

[5] Jurisch B, Kubiak W, Józefowska J. Algorithms for minclique scheduling problems. Discr Appl Math 1997;72:115–39.

[6] Kacem I, Kellerer H, Strusevich VA. Single machine scheduling with a common due date: Total weighted tardiness problems. In: Mahjoub AR, editor. Progress in combinatorial optimization, New York and London: Wiley-ISTE; 2011, Chapter 13, p. 391–421.

[7] Kellerer H, Sarto Basso R, Strusevich VA. Approximability issues for unconstrained and constrained maximization of half-product related functions. Theor Comp Sci 2015;659:64–71.

[8] Kellerer H, Strusevich VA. Fully polynomial approximation schemes for a symmetric quadratic knapsack problem and its scheduling applications. Algorithmica; 2010;57:769–95.

[9] Kellerer H, Strusevich VA. Minimizing total weighted earliness-tardiness on a single machine around a small common due date: An FPTAS using quadratic knapsack. Int J Found Comp Sci 2010;21:357–83.

[10] Kellerer H, Strusevich VA. The symmetric quadratic knapsack problem: Approximation and scheduling applications. 4OR – Quart J Oper Res 2012;10:111–61.

[11] Kellerer H, Strusevich VA. Fast approximation schemes for Boolean programming and scheduling problems related to positive convex half-product. Eur J Oper Res 2013;228:24–32.

[12] Kellerer H, Strusevich VA. Optimizing the Half-Product and related quadratic Boolean functions: Approximation and scheduling applications. Ann Oper Res 2016;240:39–94.

[13] Kovalyov MY, Kubiak W. A fully polynomial approximation scheme for weighted earliness-tardiness problem. Oper Res 1999;47: 757–61.

[14] Kubiak W. New results on the completion time variance minimization. Discr Appl Math 1999;58:157–68.

[15] Kubiak W. Minimization of ordered, symmetric half-products. Discr Appl Math 2005;146:287–300.

[16] Lann A, Mosheiov G. Single machine scheduling to minimize the number of early and tardy jobs. Comput Oper Res 1996;23:765–81.

[17] Sarto Basso R, Strusevich VA. Differential approximation schemes for half-product related functions and their scheduling applications. Discr Appl Math 2017;217:71–8.

[18] Skutella M. Convex quadratic and semidefinite programming relaxations in scheduling. J ACM 2001;48: 206–42.

[19] Tamir A. A strongly polynomial algorithm for minimum convex separable quadratic cost flow problems on two-terminal series-parallel networks. Math Progr 1993;59:117–32.

[20] Xu Z. A strongly polynomial FPTAS for the symmetric quadratic knapsack problem. Eur J Oper Res 2012;218: 377–81.

[21] Yin Y, Chen S-R, Cheng TCE, Wang D-J, Wue C-C. Just-in-time scheduling with two competing agents on unrelated parallel machines. Omega 2016;63: 41–47.