European Journal of Operational Research 228 (2013) 24–32

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Fast approximation schemes for Boolean programming and scheduling problems related to positive convex Half-Product

Hans Kellerer [a], Vitaly Strusevich [b,*]

[a] Institut für Statistik und Operations Research, Universität Graz, Universitätsstraße 15, A-8010 Graz, Austria
[b] School of Computing and Mathematical Sciences, University of Greenwich, Old Royal Naval College, Park Row, Greenwich, London SE10 9LS, UK

ARTICLE INFO

Article history:
Received 8 August 2012
Accepted 29 December 2012
Available online 8 January 2013

Keywords:
Scheduling
Half-Product
Quadratic knapsack
Scheduling with rejection
Scheduling with controllable processing times
FPTAS

ABSTRACT

We address a version of the Half-Product Problem and its restricted variant with a linear knapsack constraint. For these minimization problems of Boolean programming, we focus on the development of fully polynomial-time approximation schemes with running times that depend quadratically on the number of variables. Applications to various single machine scheduling problems are reported: minimizing the total weighted flow time with controllable processing times, minimizing the makespan with controllable release dates, minimizing the total weighted flow time for two models of scheduling with rejection.

1. Introduction

The topic of designing approximation schemes for scheduling problems with min-sum objective functions has recently drawn considerable attention. While for many problems of this range purpose-built approximation schemes have been developed, a general framework has been identified based on reformulation of the original scheduling problems in terms of minimization problems of quadratic Boolean programming. The Half-Product Problem and the closely related Symmetric Quadratic Knapsack Problem appear to be among the most suitable models, see recent reviews by Kacem et al. (2011) and Kellerer and Strusevich (2012).

This paper studies a version of the Half-Product Problem and its modification with the knapsack constraint, establishes conditions under which the problems admit fast fully polynomial-time approximation schemes, describes the relevant algorithms and discusses their scheduling applications.

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

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

where for each j, $1 \leqslant j \leqslant n$, the coefficients α_j and β_j are non-negative integers, while γ_j is an integer that can be either negative or positive; in fact, without loss of generality, we may assume that all γ_j are non-negative, since otherwise for a negative γ_j we may set $x_j = 0$ without increasing the value of $H(\mathbf{x})$. Problems of quadratic Boolean programming similar to (1) were introduced in 1990s as mathematical models for various scheduling problems by Kubiak (1995) and Jurisch et al. (1997). This function and the term "Half-Product" were introduced by Badics and Boros (1998), who considered the problem of minimizing the function $H(\mathbf{x})$ with respect to Boolean decision variables with no additional constraints. The function $H(\mathbf{x})$ is called a *Half-Product* since its quadratic part consists of roughly half of the terms of the product $\left(\sum_{j=1}^{n} \alpha_j x_j\right)\left(\sum_{j=1}^{n} \beta_j x_j\right)$. 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.

In this paper, we refer to the problem of minimizing function $H(\mathbf{x})$ of the form (1), as *Problem HP*. This problem is NP-hard in the ordinary sense, even if $a_j = b_j$ for all $j = 1, 2, \ldots, n$, as proved by Badics and Boros (1998). It has numerous applications, mainly to machine scheduling; see Erel and Ghosh (2008) and Kellerer and Strusevich (2012) for reviews. Notice that in those applications a scheduling objective function usually is written in the form

$$F(\mathbf{x}) = H(\mathbf{x}) + K, \tag{2}$$

where K is a given additive constant. We refer to the problem of minimizing function $F(\mathbf{x})$ of the form (2), as *Problem HPAdd*.

* Corresponding author. Tel.: +44 20 83318662; fax: +44 20 83318665.
E-mail addresses: hans.kellerer@uni-graz.at (H. Kellerer), V.Strusevich@greenwich.ac.uk (V. Strusevich).

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2012.12.028

Consider the function

$$P(\mathbf{x}) = \sum_{1\leqslant i<j\leqslant n} \alpha_i\beta_j x_i x_j + \sum_{j=1}^{n}\mu_j x_j + \sum_{j=1}^{n} v_j(1-x_j) + K, \quad (3)$$

where all coefficients α_j are positive integers, and β_j, μ_j, v_j and K are non-negative integers. Following Janiak et al. (2005), we call the problem of minimizing the function $P(\mathbf{x})$ of the form (3) the *Positive Half-Product Problem* or *Problem PosHP*. Notice that Problem PosHP is a special case of Problem HPAdd.

In all Half-Product Problems introduced above, the minimum is sought for over all n-dimensional Boolean vectors, i.e., they are quadratic Boolean programming problems with no additional constraints. In this paper, we also study a more restricted version of Problem PosHP, in which an additional knapsack constraint is introduced, i.e., the problem

$$\begin{aligned} &\text{Minimize} \quad P(\mathbf{x}) = \sum_{1\leqslant i<j\leqslant n} \alpha_i\beta_j x_i x_j + \sum_{j=1}^{n}\mu_j x_j + \sum_{j=1}^{n} v_j(1-x_j) + K \\ &\text{Subject to} \quad \sum_{j=1}^{n}\alpha_j x_j \leqslant A \\ &\qquad\qquad\quad x_j \in \{0,1\}, \quad j=1,2,\ldots,n, \end{aligned} \quad (4)$$

which we call the *Positive Half-Product Knapsack Problem* and denote by *Problem PosHPK*.

Similarly to the classical Linear Knapsack Problem (see the comprehensive monographs Martello and Toth (1990) and Kellerer et al. (2004) on this most studied problem of Combinatorial Optimization), Problem PosHPK contains a linear *knapsack* constraint. We can view the value α_j as the weight of item j, $1 \leqslant j \leqslant n$, i.e., $x_j = 1$ means that item j is placed into a knapsack with capacity A, while $x_j = 0$ means that the corresponding item is not placed into the knapsack. An important feature of our problem is that the coefficients α_j in the knapsack constraint are the same as in the quadratic terms of the objective function. The latter feature makes Problem PosHPK to be a special case of another quadratic knapsack problem, namely the problem

$$\begin{aligned} &\text{Minimize} \quad S(\mathbf{x}) = \sum_{1\leqslant i<j\leqslant n} \alpha_i\beta_j x_i x_j + \sum_{1\leqslant i<j\leqslant n} \alpha_i\beta_j (1-x_i)(1-x_j) \\ &\qquad\qquad + \sum_{j=1}^{n}\mu_j x_j + \sum_{j=1}^{n} v_j(1-x_j) + K \\ &\text{Subject to} \quad \sum_{j=1}^{n}\alpha_j x_j \leqslant A \\ &\qquad\qquad\quad x_j \in \{0,1\}, \quad j=1,2,\ldots,n. \end{aligned} \quad (5)$$

Following Kellerer and Strusevich (2010a,b), we call the latter problem the *Symmetric Quadratic Knapsack Problem*, or *Problem SQK*. We use the term "*symmetric*" because both the quadratic and the linear parts of the objective function are separated into two terms, one depending on the variables x_j, and the other depending on the variables $(1 - x_j)$. A comprehensive review of the results on Problem SQK and its scheduling applications is given by Kellerer and Strusevich (2012).

Table 1 summarizes the notation introduced above for all Boolean programming problems under consideration.

Since this paper focuses on the development of approximation algorithms and schemes, below we recall the definitions of the relevant notions. For a problem of minimizing a function $Z(\mathbf{x})$, where $\mathbf{x}$ is a collection of decision variables, let $\mathbf{x}^*$ denote the vector that delivers the minimum to the function $Z(\mathbf{x})$; we call $\mathbf{x}^*$ an *optimal solution* of the corresponding problem. A polynomial-time algorithm that finds a feasible solution $\mathbf{x}'$ such that $Z(\mathbf{x}')$ is at most $\rho \geqslant 1$ times the optimal value $Z(\mathbf{x}^*)$ is called a *ρ-approximation* algorithm; the value of ρ is called a *worst-case ratio bound*. For a problem of Boolean programming of minimizing a function $Z(\mathbf{x})$, which may take negative and positive values, a vector $\mathbf{x}'$ is called an *ε-approximate solution* if for a given positive ε the inequality $Z(\mathbf{x}') - Z(\mathbf{x}^*) \leqslant \varepsilon|Z(\mathbf{x}^*)|$ holds. A family of algorithms that for any given positive ε find an ε-approximate solution is called a *Fully Polynomial-Time Approximation Scheme* (*FPTAS*), provided that the running time depends polynomially on both the length of the input and $1/\varepsilon$.

Table 1
Notation for Boolean programming problems under consideration.

Problem acronym	Objective/formulation	Additional constraints
HP	$H(\mathbf{x})$ (1)	None
HPAdd	$F(\mathbf{x})$ (2)	None
PosHP	$P(\mathbf{x})$ (3)	None
PosHPK	$P(\mathbf{x})$ (4)	$\sum_{j=1}^{n}\alpha_j x_j \leqslant A$
SQK	$S(\mathbf{x})$ (5)	$\sum_{j=1}^{n}\alpha_j x_j \leqslant A$

A detailed review on the design of FPTASs for problems relevant to this study is given by Kellerer and Strusevich (2012). Badics and Boros (1998) give the first FPTAS for Problem HP but its running time is $O(n^2 \log \sum \alpha_j/\varepsilon)$ is not strongly polynomial. The first FPTAS for Problem HP that requires strongly polynomial time $O(n^2/\varepsilon)$ is due to Erel and Ghosh (2008).

The algorithms that behave as an FPTAS for Problem HP to minimize a Half-Product $H(\mathbf{x})$ of the form (1) do not necessarily deliver an ε-approximate solution for the problem of minimizing the function $F(\mathbf{x})$ of the form (2), although both problems have the same optimal solution $\mathbf{x}^*$ and for any vector $\mathbf{x}$ the equality $F(\mathbf{x}) - F(\mathbf{x}^*) = H(\mathbf{x}) - H(\mathbf{x}^*)$ holds. This is due to the fact that $H(\mathbf{x}^*) < 0$ and it is possible that $|F(\mathbf{x}^*)| = |H(\mathbf{x}^*) + K| < |H(\mathbf{x}^*)|$. Starting from the pioneering work by Badics and Boros (1998), the matter of designing an FPTAS for the Half-Product problem with an additive constants, including adapting an FPTAS for Problem HP, has initiated many publications, see, e.g., Janiak et al. (2005), Kubiak (2005), Erel and Ghosh (2008) and Kellerer and Strusevich (2012). In particular, addressing this issue Janiak et al. (2005) introduce the Positive Half-Product, i.e., Problem PosHP. However, the FPTAS that they develop in their paper for Problem PosHP still requires $O(n^2 \log \sum \alpha_j/\varepsilon)$ time. Erel and Ghosh (2008) present several conditions under which an FPTAS for Problem HP behaves as an FPTAS for Problem HPAdd; see, e.g., Lemma 2 in Section 2 of this paper.

As follows from the survey by Kellerer and Strusevich (2012), in all known applications of quadratic Boolean programming problems related to the Half-Product the objective function is convex. That is why we study Problem PosHPK and its relaxed version, Problem PosHP without a knapsack constraint, provided that the objective function $P(\mathbf{x})$ is convex. For each of these problems with a convex objective function, this paper delivers an FPTAS that requires $O(n^2/\varepsilon)$ time. This running time is much smaller than $O(n^4/\varepsilon^2)$ established by Kellerer and Strusevich (2010b) for Problem SQK (under additional assumptions that include the convexity of an objective function) or $O(n^4 \log\log n + n^4/\varepsilon^2)$ provided by Xu (2012) for an arbitrary Problem SQK. The developed FPTAS can be applied to several scheduling problems, resulting in improved approximation schemes for their solution.

2. What is needed to design an FPTAS

In this section, we describe the ingredients that are needed in order to design an FPTAS for Problem PosHP and Problem PosHPK.

We start our consideration with Problem PosHPK of the form (4). As is often the case, the resulting FPTAS is obtained by modifying a dynamic programming (DP) algorithm for the problem. Such an algorithm is given below. Notice that the algorithm is rather straightforward and manipulates with the objective function (3)