UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE
HOUARI BOUMEDIENE,
USTHB - ALGER

LABORATOIRE DE RECHERCHE OPERATIONNELLE,
COMBINATOIRE, INFORMATIQUE THEORIQUE
ET METHODES STOCHASTIQUES, RECITS

Les annales RECITS

http://www.lrecits.usthb.dz/annales.htm

Vol. 1, 2014

ISSN : 2392-5426

Objectifs :

Les annales RECITS édités par le Laboratoire sont des publications scientifiques de qualité qui ont un rôle important à jouer pour la diffusion des travaux de recherche du laboratoire. Ils permettent, aux membres du laboratoire, de disposer d'un moyen de faire connaître rapidement un résultat de recherche avant sa parution dans une conférence ou une revue. Typiquement, il s'agit de la première version d'un article soumis à une revue ou à une conférence internationale et qui ne paraîtra éventuellement que beaucoup plus tard.

Comité éditorial :

Procédure de soumission :

Il n'existe pas de procédure de revue pour ces rapports de recherche. Pour publier un rapport de recherche, il faut envoyer un courrier au responsable du comité éditorial contenant le texte (fichiers latex et pdf) accompagné du formulaire de publication signé par un des membres du comité éditorial. La première page (style à télécharger) doit contenir uniquement et obligatoirement le titre, les noms, prénoms, adresses et e-mails des auteurs, les résumés et les mots clés en français et en anglais et éventuellement le financement (sponsoring, bourse de recherche, aide financière, etc.). Le tirage papier est fait en deux exemplaires dont un est remis aux auteurs. Le rapport de recherche est publié sur la page web du laboratoire et répertorié au Centre de Recherche sur l'Information Scientifique et Technique (CERIST).

Les annales RECITS du laboratoire peuvent accueillir aussi des textes de chercheurs n'appartenant pas au laboratoire et qui travaillent sur des thèmes proches que ceux développés par le laboratoire. Ces chercheurs sont priés de contacter le responsable du comité éditorial.

Le contenu des textes publiés relève de la seule responsabilité de leurs auteurs.

Contacts :

Email : lrecits@usthb.dz
Tél :
Fax :

CONTENU

Les Annales RECITS (http://www.lrecits.usthb.dz)
Vol. 01, 2014, pages 1-10

On the expansion of Fibonacci and Lucas polynomials, revisited

Hacène Belbachir[a] **and Athmane Benmezai**[b,c]

[a]USTHB, Faculty of Mathematics, RECITS Laboratory, DG-RSDT
BP 32, El Alia 16111, Bab Ezzouar, Algiers, Algeria
[b] University of Dely Brahim, Fac. of Eco. & Manag. Sc., RECITS Lab., DG-RSDT
Rue Ahmed Ouaked, Dely Brahim, Algiers, Algeria
[c] University of Oran, Faculty of Sciences, ANGE Lab., DG-RSDT
BP 1524, ELM_Naouer, 31000, Oran, Algeria.

hacenebelbachir@gmail.com or hbelbachir@usthb.dz
athmanebenmezai@gmail.com

Abstract: As established by Prodinger in "On the Expansion of Fibonacci and Lucas Polynomials", we give q-analogue of identities established by Belbachir and Bencherif in "On some properties of bivariate Fibonacci and Lucas polynomials". This is doing according to the recent Cigler's definition for the q-analogue of Fibonacci polynomials, given in "Some beautiful q-analogues of Fibonacci and Lucas polynomials", and by the authors for the q-analogues of Lucas polynomials, given in "An Alternative approach to Cigler's q-Lucas polynomials".

Keywords: Fibonacci Polynomials; Lucas Polynomials; q-analogue.

Résumé : Comme établi par Prodinger dans "On the Expansion of Fibonacci and Lucas Polynomials", nous donnons le q-analogue des identités établies par Belbachir et Bencherif dans "On some properties of bivariate Fibonacci and Lucas polynomials". Ces identités sont basées sur l'approche de Cigler pour le q-analogue des polynômes de Fibonacci, donnée dans "Some beautiful q-analogues of Fibonacci and Lucas polynomials", et par les auteurs pour les q-analogues de polynômes de Lucas, donnée dans "An Alternative approach to Cigler's q-Lucas polynomials".

Mots clés : Polynômes de Fibonacci; Polynômes Lucas; q-analogue.

1 Introduction

The bivariate polynomials of Fibonacci and Lucas, denoted respectively by (U_n) and (V_n), are defined by

$$\begin{cases} U_0 = 0,\ U_1 = 1, \\ U_n = tU_{n-1} + zU_{n-2} \quad (n \geq 2), \end{cases} \quad \text{and} \quad \begin{cases} V_0 = 2,\ V_1 = t, \\ V_n = tV_{n-1} + zV_{n-2} \ \ (n \geq 2). \end{cases}$$

It is established, see for instance [1], that

$$U_{n+1} = \sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n-k}{k} t^{n-2k} z^k, \qquad V_n = \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{n}{n-k} \binom{n-k}{k} t^{n-2k} z^k \ \ (n \geq 1).$$

In [2], the first author and Bencherif proved that, for $n - 2\lfloor n/2 \rfloor \leq k \leq n - \lfloor n/2 \rfloor$, the families $\left(x^k U_{n+1-k}\right)_k$ and $\left(x^k V_{n-k}\right)_k$ constitute two basis of the $\mathbb{Q}$-vector space spanned by the free family $\left(x^{n-2k} y^k\right)_k$, and they found that the coordinates of the bivariate polynomials of Fibonacci and Lucas, over appropriate basis, satisfies remarkable recurrence relations. They established the following formulae

$$\begin{aligned}
V_{2n} &= 2U_{2n+1} - xU_{2n}, & (1)\\
2U_{2n+1} &= \sum_{k=0}^{n} a_{n,k} t^k V_{2n-k}, \ \text{ with } \ a_{n,k} = 2\sum_{j=0}^{n} (-1)^{j+k} \binom{j}{k} - (-1)^{n+k} \binom{n}{k}, & (2)\\
V_{2n} &= \sum_{k=1}^{n} b_{n,k} t^k V_{2n-k}, \text{ with } \ b_{n,k} = (-1)^{k+1} \binom{n}{k}, & (3)\\
V_{2n-1} &= \sum_{k=1}^{n} c_{n,k} t^k U_{2n-k}, \ \text{ with } \ c_{n,k} = 2(-1)^{k+1} \binom{n}{k} - [k=1], & (4)\\
2V_{2n-1} &= \sum_{k=1}^{n} d_{n,k} t^k V_{2n-1-k}, \ \text{ with } \ d_{n,k} = (-1)^{k+1} \frac{2n-k}{n} \binom{n}{k}, & (5)\\
2U_{2n} &= \sum_{k=1}^{n} e_{n,k} t^k V_{2n-k}, \ \text{ with} & (6)\\
e_{n,k} &= (-1)^{k+1} \frac{2n-k}{n} \binom{n}{k} + \sum_{j=0}^{n-1} (-1)^{j+k-1} \binom{j}{k-1} - \frac{1}{2} (-1)^{n+k} \binom{n-1}{k-1}.
\end{aligned}$$

A similar approach was done by the authors, see [3], for Chebyshev polynomials.

As q-analogue of Fibonacci and Lucas polynomials, J. Cigler [6], considers the following expressions

$$\mathbf{F}_{n+1}(x, y, m) = \sum_{k=0}^{\lfloor n/2 \rfloor} q^{\binom{k+1}{2} + m\binom{k}{2}} \begin{bmatrix} n-k \\ k \end{bmatrix}_q x^{2n-k} y^k, \qquad (7)$$

$$\mathbf{Luc}_n(x, y, m) = \sum_{k=0}^{\lfloor n/2 \rfloor} q^{(m+1)\binom{k}{2}} \begin{bmatrix} n-k \\ k \end{bmatrix}_q \frac{[n]_q}{[n-k]_q} x^{2n-k} y^k, \qquad (8)$$

with the q-notations

$$[n]_q = 1 + q + \cdots + q^{n-1},\ [n]_q! = [1]_q[2]_q \cdots [n]_q,\ \begin{bmatrix} n-k \\ k \end{bmatrix}_q = \frac{[n]_q!}{[k]_q![n-k]_q!}.$$

Without loss the generality, we can suppose that $t = 1$. We refer here to the modified polynomials given by H. Prodinger in the introduction of [8].

Using the q-analogues of Fibonacci and Lucas polynomials suggested by J. Cigler, H. Prodinger, see [8], give q-analogues for relations (3) and (5), and the authors, see [4], give q-analogues for relations (1), (2), (4), (6).

The q-identities associated to relations (2) and (6), given in [4], do not give for $q = 1$ the initial relations. This is the motivation which conclude to this paper: we propose an alternative q-analogue for all the former relations (1), (2), (3), (4), (5) and (6) based on Cigler's definition, see [6], for the Fibonacci polynomials, and the definitions givn by the authors, see [5], for the Lucas polynomials.

In [5], we have defined the q-Lucas polynomials of the first kind $\mathbf{L}(z)$ and the q-Lucas polynomials of the second kind $\mathbb{L}(z)$ respectively by

$$\mathbf{L}_n(z,m) \ :=\ \sum_{k=0}^{\lfloor n/2 \rfloor} q^{\binom{k}{2}+m\binom{k}{2}} \begin{bmatrix} n-k \\ k \end{bmatrix}_q \left(1 + \frac{[k]_q}{[n-k]_q}\right) z^k, \tag{9}$$

$$\mathbb{L}_n(z,m) \ :=\ \sum_{k=0}^{\lfloor n/2 \rfloor} q^{\binom{k+1}{2}+m\binom{k}{2}} \begin{bmatrix} n-k \\ k \end{bmatrix}_q \left(1 + q^{n-2k}\frac{[k]_q}{[n-k]_q}\right) z^k, \tag{10}$$

and we have showed that the polynomials $\mathbf{L}_n(z)$ and $\mathbb{L}_n(z)$ satisfy the recursions

$$\mathbf{L}_{n+1}(z,m) = \mathbf{L}_n(z,m) + q^{n-1} z \mathbf{L}_{n-1}\left(q^{m-1}z, m\right), \tag{11}$$

$$\mathbb{L}_{n+1}(z,m) = \mathbb{L}_n(qz,m) + qz\mathbb{L}_{n-1}\left(q^{m+1}z, m\right). \tag{12}$$

These two recursions are satisfied by the q-analogue of Fibonacci polynomials $\mathbf{F}_n(z,m)$, see [6].

2 Main results

In [5], the authors expressed the q-Lucas polynomials of the both kinds in terms of q-Fibonacci polynomials by the identities

$$\mathbf{L}_n(z,m) = 2\mathbf{F}_{n+1}\left(\frac{z}{q}, m\right) - \mathbf{F}_n(z,m),$$

$$\mathbb{L}_n(z,m) = 2\mathbf{F}_{n+1}(z,m) - \mathbf{F}_n(z,m),$$

which are considered as a q-analogue of the equality (1).

The following result gives two q-analogues of relation (2), the first one is related to the q-Lucas polynomials of the first kind and the second one is related to the q-Lucas polynomials of the second kind.

Theorem 1 *For every integer $n \geq 0$, one has*

$$2\mathbf{F}_{2n+1}\left(\frac{z}{q},m\right) = \sum_{k=0}^{n} q^{\binom{k}{2}} \begin{bmatrix} n \\ k \end{bmatrix}_q (-1)^{n+k} \mathbf{L}_{2n-k}(z,m) + 2\sum_{j=0}^{n-1}\sum_{k=0}^{j} q^{\binom{k}{2}-2nj} \begin{bmatrix} j \\ k \end{bmatrix}_q (-1)^{k+j} \mathbf{L}_{2n-k}\left(q^{2j}z,m\right), \quad (13)$$

$$2\mathbf{F}_{2n+1}(z,m) = 2\sum_{j=0}^{n-1}\sum_{k=0}^{j} q^{\binom{j-k}{2}-\binom{j}{2}+2nj} \begin{bmatrix} j \\ k \end{bmatrix}_q (-1)^{k+j} \mathbb{L}_{2n-k}\left(q^{k-2j}z,m\right) + \sum_{k=0}^{n} q^{n(n-k)+\binom{k+1}{2}} \begin{bmatrix} n \\ k \end{bmatrix}_q (-1)^{k} \mathbb{L}_{2n-k}\left(q^{k-n}z,m\right). \quad (14)$$

The q-analogue of relation (3), found by Prodinger in [8], is given by the following Theorem which gives also a second q-analogue identity.

Theorem 2 *For $n \geq 1$, we have*

$$\mathbf{F}_{2n}(z,m) = \sum_{j=1}^{n} q^{\binom{j}{2}} \begin{bmatrix} n \\ j \end{bmatrix}_q (-1)^{j+1} \mathbf{F}_{2n-j}(z,m), \quad (15)$$

$$\mathbf{F}_{2n}(z,m) = \sum_{j=1}^{n} q^{\binom{n-j}{2}} \begin{bmatrix} n \\ j \end{bmatrix}_q (-1)^{j+1} \mathbf{F}_{2n-j}\left(q^{j}z,m\right). \quad (16)$$

The relation (4), admits two q-analogues related to the q-Lucas polynomials of the first kind $\mathbf{L}(z)$, and two q-analogues related to the q-Lucas polynomials of the second kind $\mathbb{L}(z)$, given respectively by the following Theorem.

Theorem 3 *For $n \geq 1$, the q-Lucas polynomials of the first kind is developed by*

$$\mathbf{L}_{2n-1}(z,m) = 2\sum_{j=1}^{n} q^{\binom{j}{2}} \begin{bmatrix} n \\ j \end{bmatrix}_q (-1)^{j+1} \mathbf{F}_{2n-j}\left(\frac{z}{q},m\right) - \mathbf{F}_{2n-1}(z,m), \quad (17)$$

$$\mathbf{L}_{2n-1}(z,m) = 2\sum_{j=1}^{n} q^{\binom{n-j}{2}} \begin{bmatrix} n \\ j \end{bmatrix}_q (-1)^{j+1} \mathbf{F}_{2n-j}\left(q^{j-1}z,m\right) - \mathbf{F}_{2n-1}(z,m). \quad (18)$$

and the q-Lucas polynomials of the second kind is developed by

$$\mathbb{L}_{2n-1}(z,m) = 2\sum_{j=1}^{n} q^{\binom{j}{2}} \begin{bmatrix} n \\ j \end{bmatrix}_q (-1)^{j+1} \mathbf{F}_{2n-j}(z,m) - \mathbf{F}_{2n-1}(z,m), \quad (19)$$

$$\mathbb{L}_{2n-1}(z,m) = 2\sum_{j=1}^{n} q^{\binom{n-j}{2}} \begin{bmatrix} n \\ j \end{bmatrix}_q (-1)^{j+1} \mathbf{F}_{2n-j}\left(q^{j}z,m\right) - \mathbf{F}_{2n-1}(z,m). \quad (20)$$

Using the q-Lucas polynomials of the both kinds we find two q-analogues of relation (5).

Theorem 4 *For $n \geq 1$, we have*

$$2\mathbf{L}_{2n-1}(z,m) = \sum_{k=1}^{n} q^{\binom{k}{2}} \begin{bmatrix} n \\ k \end{bmatrix}_q (-1)^{k+1} q^{k+1} \left(1 + \frac{[n-k]_q}{[n]_q}\right) \mathbf{L}_{2n-2-k}(z,m), \qquad (21)$$

$$2\mathbb{L}_{2n-1}(z,m) = \sum_{k=1}^{n} q^{\binom{n-k}{2}} \begin{bmatrix} n \\ k \end{bmatrix}_q (-1)^{k+1} q^{1-n} \left(1 + q^k \frac{[n-k]_q}{[k]_q}\right) \mathbb{L}_{2n-1-k}\left(q^k z, m\right) \qquad (22)$$

In the following theorem, we give two q-analogues of (6).

Theorem 5 *For every integer $n \geq 0$, one has*

$$\begin{aligned} & 2\mathbf{F}_{2n}\left(\frac{z}{q}, m\right) \\ = & \frac{1}{2}\sum_{k=1}^{n} q^{\binom{k}{2}} \begin{bmatrix} n \\ k \end{bmatrix}_q (-1)^{k+1} q^{k+1} \left(1 + \frac{[n-k]_q}{[n]_q}\right) \mathbf{L}_{2n-2-k}(z,m) + \\ & \frac{1}{2}\sum_{k=0}^{n-1} q^{\binom{k}{2}} \begin{bmatrix} n-1 \\ k \end{bmatrix}_q (-1)^{n-1+k} \mathbf{L}_{2n-2-k}(qz,m) + \\ & \sum_{j=0}^{n-2}\sum_{k=0}^{j} q^{\binom{k}{2}-2nj-2j} \begin{bmatrix} j \\ k \end{bmatrix}_q (-1)^{k+j} \mathbf{L}_{2n-2-k}\left(q^{2j+1}z, m\right). \end{aligned} \qquad (23)$$

and

$$\begin{aligned} & 2\mathbf{F}_{2n}(z) \\ = & \frac{1}{2}\sum_{k=1}^{n} q^{\binom{n-k}{2}} \begin{bmatrix} n \\ k \end{bmatrix}_q (-1)^{k+1} q^{1-n} \left(1 + q^k \frac{[n-k]_q}{[k]_q}\right) \mathbb{L}_{2n-1-k}\left(q^k z, m\right) + \\ & \frac{1}{2}\sum_{k=0}^{n} q^{(n-1)(n-k)+\binom{k+1}{2}} \begin{bmatrix} n-1 \\ k \end{bmatrix}_q (-1)^{n-1+k} \mathbb{L}_{2n-2-k}\left(q^{k+1-n}z, m\right) + \\ & \sum_{j=0}^{n-1}\sum_{k=0}^{j} q^{\binom{j-k}{2}-\binom{j}{2}+2nj-2j} \begin{bmatrix} j \\ k \end{bmatrix}_q (-1)^{k+j} \mathbb{L}_{2n-2-k}\left(q^{k-2j}z, m\right). \end{aligned} \qquad (24)$$

Remark 1 *Notice that the coefficients appeared in the sums given in the different results do not depends on m.*

3 Proof of the results

We need the following Lemmas.

Lemma 6 *For $U_n(z)$ satisfying (11) and (12) respectively, we have for $k \geq 1$*

$$\begin{aligned}
\sum_{j=0}^{k} q^{\binom{j}{2}} \begin{bmatrix} k \\ j \end{bmatrix}_q (-1)^j U_{n+k-j}(z,m) &= q^{m\binom{k}{2}+\binom{n}{2}-\binom{n-k}{2}} z^k U_{n-k}\left(q^{mk-k}z,m\right), \\
\sum_{j=0}^{k} q^{\binom{k-j}{2}} \begin{bmatrix} k \\ j \end{bmatrix}_q (-1)^j U_{n+k-j}\left(q^{j-k}z,m\right) &= q^{m\binom{k}{2}} z^k U_{n-k}\left(q^{mk}z,m\right).
\end{aligned}$$

Proof. We use induction over k, the case $k=1$ is given by the recursions (11) and (12). We suppose the relation true for k,

$$\begin{aligned}
& q^{m\binom{k+1}{2}+\binom{n}{2}-\binom{n-1-k}{2}} z^{k+1} U_{n-k-1}\left(q^{mk+m-1-k}z,m\right) \\
= \; & q^{m\binom{k}{2}+\binom{n}{2}-\binom{n-k}{2}} q^k z^k \left(U_{n+1-k}\left(q^{mk-k}z,m\right) - U_{n-k}\left(q^{mk-k}z,m\right)\right) \\
= \; & q^{m\binom{k}{2}+\binom{n-1}{2}-\binom{n-1-k}{2}} z^k U_{n+1-k}\left(q^{mk-k}z,m\right) \\
& - q^{m\binom{k}{2}+\binom{n}{2}-\binom{n-k}{2}} q^k z^k U_{n-k}\left(q^{mk-k}z,m\right) \\
= \; & \sum_{j=0}^{k} q^{\binom{j}{2}} \begin{bmatrix} k \\ j \end{bmatrix}_q (-1)^j U_{n+1-k-j}(z,m) - q^k \sum_{j=0}^{k} q^{\binom{j}{2}} \begin{bmatrix} k \\ j \end{bmatrix}_q (-1)^j U_{n+k-j}(z,m) \\
= \; & \sum_{j=0}^{k} q^{\binom{j}{2}} \left(q^{\binom{j}{2}} \begin{bmatrix} k \\ j \end{bmatrix}_q + q^k q^{\binom{j-1}{2}} \begin{bmatrix} k \\ j-1 \end{bmatrix}_q \right) (-1)^j U_{n+k+1-j}(z,m) \\
= \; & \sum_{j=0}^{k} q^{\binom{j}{2}} \begin{bmatrix} k \\ j \end{bmatrix}_q (-1)^j U_{n+k+1-j}(z,m)
\end{aligned}$$

and

$$\begin{aligned}
& q^{m\binom{k+1}{2}} z^{k+1} U_{n-1-k}\left(q^{mk+m}z,m\right) \\
= \; & q^{m\binom{k}{2}} z^k \left(U_{n+1-k}\left(q^{mk-1}z,m\right) - U_{n-k}\left(q^{mk}z,m\right)\right) \\
= \; & q^{m\binom{k}{2}} q^k \sum_{j=0}^{k} q^{\binom{k-j}{2}} \begin{bmatrix} k \\ j \end{bmatrix}_q (-1)^j U_{n+k-j}\left(q^{j-1-k}z,m\right) - \\
& \sum_{j=0}^{k} q^{\binom{k-j}{2}} \begin{bmatrix} k \\ j \end{bmatrix}_q (-1)^j U_{n+k-j}\left(q^{j-k}z,m\right) \\
= \; & \sum_{j=0}^{k} \left(q^k q^{\binom{k-j}{2}} \begin{bmatrix} k \\ k-j \end{bmatrix}_q + q^{\binom{k-j+1}{2}} \begin{bmatrix} k \\ k-j+1 \end{bmatrix}_q \right) (-1)^j U_{n+k-j}\left(q^{j-1-k}z,m\right) \\
= \; & \sum_{j=0}^{k} q^{\binom{k-j}{2}} \begin{bmatrix} k \\ j \end{bmatrix}_q (-1)^j U_{n+k+1-j}\left(q^{j-1-k}z,m\right)
\end{aligned}$$

■

Lemma 7 *For every integer $n \geq 1$, one has*

$$\begin{aligned}
&\mathbf{F}_{2n+1}\left(\frac{z}{q},m\right) \\
= \ & (-z)^n q^{(m+1)\binom{n}{2}} + \sum_{j=0}^{n-1} q^{(m+1)\binom{j}{2}} (-z)^j \mathbf{L}_{2(n-j)}\left(q^{mj+j}z,m\right), \\
&\mathbf{F}_{2n+1}(z,m) \\
= \ & (-z)^n q^{\binom{n+1}{2}+m\binom{n}{2}} + \sum_{j=0}^{n-1} q^{j(2n-1)+(m-3)\binom{j}{2}} (-z)^j \mathbb{L}_{2(n-j)}\left(q^{mj-j}z,m\right).
\end{aligned}$$

Proof. We use induction over n, the case $n=1$ is given by the following relations, see [5]

$$\begin{aligned}
\mathbf{L}_n(z,m) &= \mathbf{F}_{n+1}\left(\frac{z}{q},m\right) + z\mathbf{F}_{n-1}(q^m z,m), \\
\mathbb{L}_n(z,m) &= \mathbf{F}_{n+1}(z,m) + q^{n-1}z\mathbf{F}_{n-1}\left(q^{m-1}z,m\right).
\end{aligned}$$

We suppose the identities true for n, then

$$\begin{aligned}
&\mathbf{F}_{2n+3}\left(\frac{z}{q},m\right) \\
= \ & \mathbf{L}_{2n+2}(z,m) - z\mathbf{F}_{2n+1}(q^m z,m), \\
= \ & \mathbf{L}_{2n+2}(z,m) + (-z)^{n+1} q^{(m+1)\binom{n+1}{2}} - \\
& z\sum_{j=0}^{n-1} q^{\binom{j}{2}} \left(-q^{m+1}z\right)^j \mathbf{L}_{2(n-j)}\left(q^{mj+j+m+1}z,m\right), \\
= \ & (-z)^{n+1} q^{(m+1)\binom{n+1}{2}} + \sum_{j=0}^{n} q^{\binom{j}{2}} (-z)^j \mathbf{L}_{2(n+1-j)}\left(q^{mj+j}z,m\right).
\end{aligned}$$

and

$$\begin{aligned}
&\mathbf{F}_{2n+3}(z,m) \\
= \ & \mathbb{L}_{2n+2}(z,m) - q^{2n+1}z\mathbf{F}_{2n+1}\left(q^{m-1}z,m\right), \\
= \ & \mathbb{L}_{2n+2}(z,m) - q^{2n+1}z(-z)^n q^{\binom{n}{2}+m\binom{n+1}{2}} + \\
& q^{2n+1}z\sum_{j=0}^{n-1} q^{j(2n+1)+(m-3)\binom{j+1}{2}} (-z)^j \mathbb{L}_{2(n-j)}\left(q^{mj+m-1-j}z,m\right), \\
= \ & \mathbb{L}_{2n+2}(z,m) + (-z)^{n+1} q^{\binom{n+2}{2}m\binom{n+1}{2}} - \\
& q^{2n+1}\sum_{j=0}^{n-1} q^{j(2n+1)+(m-3)\binom{j+1}{2}} (-z)^j \mathbb{L}_{2(n-j)}\left(q^{mj+m-1-j}z,m\right), \\
= \ & (-z)^{n+1} q^{\binom{n+2}{2}m\binom{n+1}{2}} + \sum_{j=0}^{n} q^{j(2n+1)-(m-3)\binom{j}{2}} (-z)^j \mathbb{L}_{2(n+1-j)}\left(q^{mj-j}z,m\right).
\end{aligned}$$

■

Proof of relations (13) and (14).. Replacing $U_n(z)$ by $\mathbf{L}_n(z)$ and $\mathbb{L}_n(z)$ respectively, in Lemma 6, we get

$$\begin{aligned}\sum_{k=0}^{n} q^{\binom{k}{2}} \begin{bmatrix} n \\ k \end{bmatrix}_q (-1)^k \mathbf{L}_{2n-k}(z,m) &= q^{(m+1)\binom{n}{2}} z^n 2, \\ \sum_{k=0}^{j} q^{\binom{k}{2}} \begin{bmatrix} j \\ k \end{bmatrix}_q (-1)^k \mathbf{L}_{2n-k}\left(q^{2j}z,m\right) &= q^{(m+1)\binom{j}{2}+2nj} z^j \mathbf{L}_{2(n-j)}\left(q^{mj+j}z,m\right).\end{aligned}$$

and

$$\begin{aligned}\sum_{k=0}^{n} q^{\binom{n-k}{2}} \begin{bmatrix} n \\ k \end{bmatrix}_q (-1)^k \mathbb{L}_{2n-k}\left(q^{k-n}z,m\right) &= q^{m\binom{n}{2}} z^n 2, \\ \sum_{k=0}^{j} q^{\binom{j-k}{2}} \begin{bmatrix} j \\ k \end{bmatrix}_q (-1)^k \mathbb{L}_{2n-k}\left(q^{k-2j}z,m\right) &= q^{(m-2)\binom{j}{2}-j} z^j \mathbb{L}_{2(n-j)}\left(q^{mj-j}z,m\right).\end{aligned}$$

using these relations in Lemma 7, we draw

$$\begin{aligned}2\mathbf{F}_{2n+1}\left(\frac{z}{q},m\right) &= \sum_{k=0}^{n} q^{\binom{k}{2}} \begin{bmatrix} n \\ k \end{bmatrix}_q (-1)^{n+k} \mathbf{L}_{2n-k}(z,m) + \\ &\quad 2\sum_{j=0}^{n-1}\sum_{k=0}^{j} \left(-q^{-2n}\right)^j q^{\binom{k}{2}} \begin{bmatrix} j \\ k \end{bmatrix}_q (-1)^k \mathbf{L}_{2n-k}\left(q^{2j}z,m\right),\end{aligned}$$

and

$$\begin{aligned}2\mathbf{F}_{2n+1}(z,m) &= \sum_{k=0}^{n} q^{\binom{n-k}{2}+\binom{n+1}{2}} \begin{bmatrix} n \\ k \end{bmatrix}_q (-1)^{n+k} \mathbb{L}_{2n-k}\left(q^{k-n}z,m\right) + \\ &\quad \sum_{j=0}^{n-1}\sum_{k=0}^{j} q^{2nj} q^{\binom{j-k}{2}-\binom{j}{2}} \begin{bmatrix} j \\ k \end{bmatrix}_q (-1)^{k+j} \mathbb{L}_{2n-k}\left(q^{k-2j}z,m\right)\end{aligned}$$

■

Proof of relations (15) and (16).. For $k = n$ and $U_n(z) = \mathbf{F}_n(z)$ in Lemma 6, we have

$$\begin{aligned}\sum_{j=0}^{n} q^{\binom{j}{2}} \begin{bmatrix} n \\ j \end{bmatrix}_q (-1)^j \mathbf{F}_{2n-j}(z,m) &= q^{m\binom{k}{2}+\binom{n}{2}} z^n \mathbf{F}_0\left(q^{mn-n}z,m\right) = 0, \\ \sum_{j=0}^{n} q^{\binom{n-j}{2}} \begin{bmatrix} n \\ j \end{bmatrix}_q (-1)^j \mathbf{F}_{2n-j}\left(q^{j-n}z,m\right) &= q^{m\binom{n}{2}} z^n \mathbf{F}_0\left(q^{mn}z,m\right) = 0.\end{aligned}$$

■

Proof of relations (17), (18), (19) and (20).. It suffices to replace relations (15) and (16) in

$$\begin{aligned}\mathbf{L}_{2n-1}(z,m) &= 2\mathbf{F}_{2n}\left(\frac{z}{q},m\right) - \mathbf{F}_{2n-1}(z,m), \\ \mathbb{L}_{2n-1}(z,m) &= 2\mathbf{F}_{2n}(z,m) - \mathbf{F}_{2n-1}(z,m).\end{aligned}$$

■

Proof of relations (21) and (22).. According to Lemma 6, we have

$$\begin{aligned}\sum_{k=0}^{n-1} q^{\binom{k}{2}} \begin{bmatrix} n-1 \\ k \end{bmatrix}_q (-1)^k \mathbf{L}_{2n-2-k}(z,m) &= 2(z)^{n-1} q^{m\binom{n-1}{2}+\binom{n-1}{2}}, \\ \sum_{k=0}^{n-1} q^{\binom{k}{2}} \begin{bmatrix} n-1 \\ k \end{bmatrix}_q (-1)^k \mathbf{L}_{2n-1-k}(z,m) &= (z)^{n-1} q^{m\binom{n-1}{2}+\binom{n}{2}},\end{aligned}$$

and

$$\begin{aligned}\sum_{k=0}^{n-1} q^{\binom{n-1-k}{2}} \begin{bmatrix} n-1 \\ k \end{bmatrix}_q (-1)^k \mathbb{L}_{2n-2-k}\left(q^{k+2-n}z,m\right) &= 2q^{m\binom{n-1}{2}} (qz)^{n-1} \\ \sum_{k=0}^{n-1} q^{\binom{n-1-k}{2}} \begin{bmatrix} n-1 \\ k \end{bmatrix}_q (-1)^k \mathbb{L}_{2n-1-k}\left(q^{k+1-n}z,m\right) &= q^{m\binom{n-1}{2}} z^{n-1}\end{aligned}$$

■

Then

$$\begin{aligned}&2\mathbf{L}_{2n-1}(z,m) \\ &= q^{n-1}\sum_{k=1}^{n} q^{\binom{k-1}{2}} \begin{bmatrix} n-1 \\ k-1 \end{bmatrix}_q (-1)^{k+1} \mathbf{L}_{2n-1-k}(z,m) - \\ &\quad 2\sum_{k=1}^{n-1} q^{\binom{k}{2}} \begin{bmatrix} n-1 \\ k \end{bmatrix}_q (-1)^k \mathbf{L}_{2n-1-k}(z,m), \\ &= \sum_{k=1}^{n} q^{\binom{k}{2}} \begin{bmatrix} n \\ k \end{bmatrix}_q (-q)^{k+1} \left(q^{n-k}\frac{[k]_q}{[n]_q} + 2\frac{[n-k]_q}{[n]_q}\right) \mathbf{L}_{2n-1-k}(z,m), \\ &= \sum_{k=1}^{n} q^{\binom{k}{2}} \begin{bmatrix} n \\ k \end{bmatrix}_q (-q)^{k+1} \left(1+\frac{[n-k]_q}{[n]_q}\right) \mathbf{L}_{2n-2-k}(z,m).\end{aligned}$$

and

$$\begin{aligned}&2\mathbb{L}_{2n-1}\left(q^{1-n}z,m\right) \\ &= q^{1-n}\sum_{k=1}^{n} q^{\binom{n-k}{2}} \begin{bmatrix} n-1 \\ k-1 \end{bmatrix}_q (-1)^{k+1} \mathbb{L}_{2n-1-k}\left(q^{k+1-n}z,m\right) - \\ &\quad 2\sum_{k=1}^{n-1} q^{\binom{n-1-k}{2}} \begin{bmatrix} n-1 \\ k \end{bmatrix}_q (-1)^k \mathbb{L}_{2n-1-k}\left(q^{k+1-n}z,m\right), \\ &= \sum_{k=1}^{n} q^{\binom{n-k}{2}} \begin{bmatrix} n \\ k \end{bmatrix}_q (-1)^{k+1} q^{1-n} \left(\frac{[k]_q}{[n]_q} + 2q^k\frac{[n-k]_q}{[n]_q}\right) \mathbb{L}_{2n-1-k}\left(q^{k+1-n}z,m\right), \\ &= \sum_{k=1}^{n} q^{\binom{n-k}{2}} \begin{bmatrix} n \\ k \end{bmatrix}_q (-1)^{k+1} q^{1-n} \left(1+q^k\frac{[n-k]_q}{[k]_q}\right) \mathbb{L}_{2n-1-k}\left(q^{k+1-n}z,m\right).\end{aligned}$$

Proof of the relations (23), (24).. Using relations (13), (14), (21), (22) in

$$\begin{aligned}2\mathbf{F}_{2n}\left(\frac{z}{q},m\right) &= \mathbf{L}_{2n-1}(z,m) + \mathbf{F}_{2n-1}(z,m), \\ 2\mathbf{F}_{2n}(z,m) &= \mathbb{L}_{2n-1}(z,m) + \mathbf{F}_{2n-1}(z,m).\end{aligned}$$

we draw the results. ■

Remark 2 *Considering these results, we obtain a duality between the q-Lucas polynomials of the first kind and the q-Lucas polynomials of the second kind.*

References

[1] H. Belbachir, F. Bencherif, Linear recurrent sequences and powers of a square matrix. *Integers* **6**, 2006, A12.

[2] H. Belbachir, F. Bencherif, On some properties of bivariate Fibonacci and Lucas polynomials, *J. Integer Sequences* **11**, Article 08.2.6, 2008.

[3] H. Belbachir, F. Bencherif, On some properties of Chebyshev polynomials, Discuss. Math. Gen. Algebra Appl., **28** (1), (2008), 121–133.

[4] H. Belbachir, A. Benmezai, Expansion of Fibonacci and Lucas polynomials: An answer to Prodinger's question, J. Integer Sequences, Vol. **15** (2012),.Article12.7.6

[5] H. Belbachir, A. Benmezai, An Alternative approach to Cigler's q-Lucas polynomials, Applied Mathematics and Computation, **226** (2014), 691–698.

[6] J. Cigler, Some beautiful q-analogues of Fibonacci and Lucas polynomials, arXiv:11042699.

[7] J. Cigler, A new class of q-Fibonacci polynomials, *Electronic J. Combinatorics* **10**, Article R19 2003.

[8] H. Prodinger, On the Expansion of Fibonacci and Lucas Polynomials, *J. Integer Sequences* **12**, Article 09.1.6, 2009.

Les Annales RECITS (http://www.lrecits.usthb.dz)
Vol. 01, 2014, pages 11-26

NP-hardness of the two-machine flowshop problem with coupled-operations

Meziani Nadjat[1,3], Oulamara Ammar[2] and Boudhar Mourad[3]

[1] Abderrahmane Mira University, Bejaia, Algeria
[2] LORIA laboratory, University of Lorraine, France
[3] RECITS laboratory, Faculty of Mathematics, University USTHB, BP 32, Bab-Ezzouar, El-Alia 16111, Algiers, Algeria

ro_nadjet07@yahoo.fr , ammar.oulamara@loria.fr , mboudhar@usthb.dz

Abstract: We consider the problem of the coupled operations on two machines with the objective to minimizing the makespan such as each job consists of two operations on the first machine separated by a time interval and only one operation on the second machine. We study the complexity of two special cases of the general problem and we show that it's NP-hard.

Keywords: coupled tasks, time lag, flow shop, makespan.

Résumé : Le problème étudié est flowshop à deux machines avec des opérations couplées sur la première machine séparées par un délai exact et d'une seule opération sur la deuxième machine. Notre objectif est de minimiser la date de fin de traitement. Nous montrons la NP-difficulté de deux de ses sous problèmes.

Mots clés : tâches couplées, temps de latence, atelier à cheminement unique, makespan.

1 Introduction

Scheduling of coupled-task problem was introduced by Shapiro in [16]. The problem consists of a set of n jobs, to be scheduled on a single machine. Each job j consists of two operations, these operations have to be executed in specified order with an intermediate exact delay L_j after the completion of the first operation. Each job j is described by a triplet (a_j, L_j, b_j) where a_j and b_j represents the processing times of the first and the second operation of jobs j, respectively. During the delay time L_j the machine is inactive and another job can be processed during the time rely L_j, as described in figure 5. The objective is determine the sequencing of operations of each job that minimize makespan or other regular objective function. According to the notation introduced by Graham et al [11], the problem of coupled-task with one machine with the objective of minimizing C_{max} noted by $1/Coup - Task, a_j, l_j, b_j/C_{max}$.

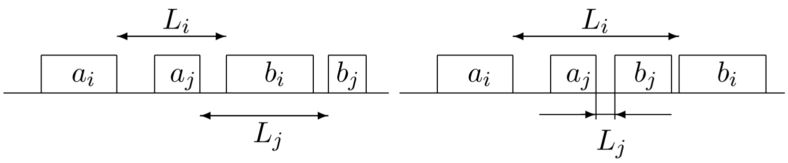

Figure 1: Examples of jobs interleaving

The motivation of a coupled-task problem stems from a scheduling problem of radar tasks which consists in the emission of the pulses and the reception of answers after the time interval, [16]. Shapiro [16], discussed the practical cases of the problem coupled-task and gave heuristics with numerical experiments. Orman and Potts [15] studied the single machine problem, with the objective of minimizing the C_{max}. They counted several problems and they have arranged them hierarchically according to their complexity. The problem $1/Coup - Task, a_j = a, L_j = l, b_j = b/C_{max}$ was left open by Orman and Potts [15] and for which others authors were interested. Ahr et al [3] proposed an exact algorithm using the dynamic programming which allows to resolve the problem for small instances where L is fixed. This algorithm was adapted by Brauner et al in [5] to resolve a coupled-task problem motivated by the time management problems of cyclic production with robots. Other researchers headed to the approximability of these problems. Thus, Ageev and Baburin [1] proposes an 7/4-approximation to solve the problem $1/Coup-Task, a_j = b_j = 1, L_j/C_{max}$. For the problems $1/Coup - Task, a_j, L_j, b_j/C_{max}$, Ageev and Kononov [2] gives several results of approximability and the limits of non-approximability according to the values of a_j and b_j. Few works were considered by adding constraints to the coupled tasks problem. Blazewicz et al [4] proved that the polynomial problem $1/Coup-task, a_j = b_j = 1, L_j = l/C_{max}$ is NP-hard by adding precedence constraints between the coupled tasks. Yu et al [18] proved that the problem on one machine $1/Coup - Task, a_j = b_j = 1, L_j/C_{max}$ is NP-hard. Simonin et al. [17] studied the problem of coupled-task with precedence constraints. They proposed a polynomial algorithm for this special problem.

Scheduling problem with exact delay is also considered in the context of flowshop environment, mainly in the two machines flowshop. The two machines flowshop scheduling problem with exact delay consists of n jobs, each job j consists of two operations $O_{A,j}$

and $O_{B,j}$ that will be processed on machines A and B, respectively. Operation $O_{B,j}$ is started on machine B after exact time delay L_j of completing of operation $O_{A,j}$. This problem is denoted $F2/L_j/C_{max}$ in the literature. Yu et al. [18] showed that problem $F2/a_j = b_j = 1, L_j/C_{max}$ is NP-hard, Ageev and Baburin [1] proposes an 3/2-approximation to solve the problem $F2/a_j = b_j = 1, L_j/C_{max}$. Ageev and Kononov [2] gives several results of approximability and the limits of non-approximability for problem $F2/L_j/C_{max}$ according to the values of a_j and b_j. Flowshop scheduling with exact time delay is special case of general problem in which time delay is bounded by minimal L_j^{min} and maximal L_j^{max} time delay. Mitten [14] provides a polynomial algorithm to minimize the makespan in two machines permutation flow shop scheduling problem with minimal time delay, i.e., $L_j^{max} = +\infty$. Lenstra et al. [13], studied the general two machines flow shop scheduling problem with minimal delay $F2/L_j^{min}/C_{max}$ and they proved that this problem is NP-hard in the strong senses. Dell Amico [6], focuses on the makespan minimization in two-machine flowshop with minimal time delay and presents a tabu search approach. Other research works related to flowshop scheduling with minimal and/or maximal time delay are studied in, [8], [9].

In this paper we consider the two-machine flowshop scheduling problem with coupled-operations. This problem consists of scheduling a set of jobs, each job is composed of two coupled-operations that should be processed on the first and on the second machine. The process of the second coupled-operation on the second machine starts only after the completion time of the first coupled-operation on the first machine. There are two other models, namely a model 1 whose the coupled-operations are carried out only on the first machine and model 2 whose couple-operations are processed on the second machine. Note that for the makespan minimization, model 1 and model 2 are equivalent. In this paper, our study is focused on model 1 in order to minimize the makespan. To our the best of knowledge this problem never considered in the literature. The motivation of flowshop scheduling problem with coupled-operations appears in workshops chemical productions where one machine must carry out several operations of the same job and an exact delay is imposed between the execution of each two consecutive operations due to the chemical reactions.

2 Problem formulation and classification

We consider the flowshop scheduling problem with two machines M_1 and M_2. A set $J = \{1, \ldots, n\}$ of jobs need to be scheduled on both machines. Each job j comprises a coupled-operation $O_{1,j}$ and a operation $O_{2,j}$ to be processed on machines M_1 and M_2, respectively, in this order for all jobs. Each coupled-operation $O_{1,j}$ of job j is described by a triplet (a_j, L_j, b_j) where a_j and b_j represents the processing times of the first and the second operation of coupled-operation $O_{1,j}$, respectively, whereas the operation $O_{2,j}$ is described by it processing time c_j. For each job j, operation $O_{2,j}$ can starts only if the coupled-operation $O_{1,j}$ is completed. The objective is to determine the sequencing of jobs that minimize the makespan.

In [15], Orman and potts studied the problem of the coupled-tasks on a single machine and they derive several results of problems depending of values of a_j, L_j and b_j, furthermore, they provide a classification of these problems depending on their complexity. Clearly, all NP-hard problems on a signal machine, remain NP-Hard in case of flowshop environment. Thus, in this paper, we focus our studies on problems that already proved polynomial in case of single machine. Figure 2 provide the list of all polynomial problems proposed by Orman and Potts [15] and the relation between these problems.

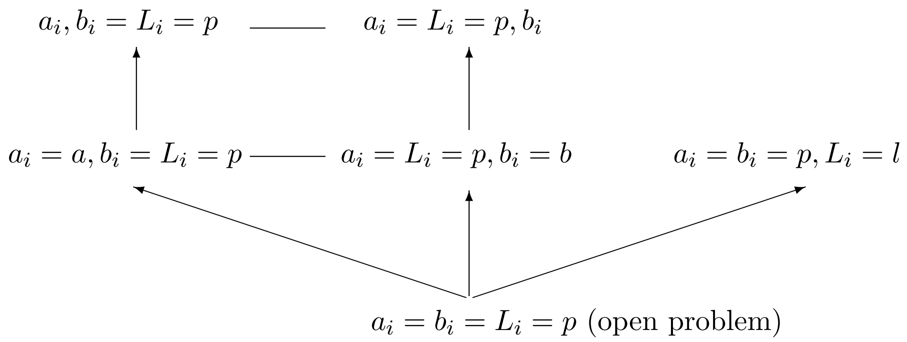

Figure 2: Polynomial problems of Orman and Potts

The problem of two-machine flow shop with coupled-operations on the first machine has not been tackled in the literature, then we are interested in this type of problem based on the polynomial problems class to define problems to be studied. However, by adding one machine to get two machines flow shop problem with coupled tasks on the first machine and one operation with processing time c_j on the second machine, we obtain the various problems presented in the following scheme.

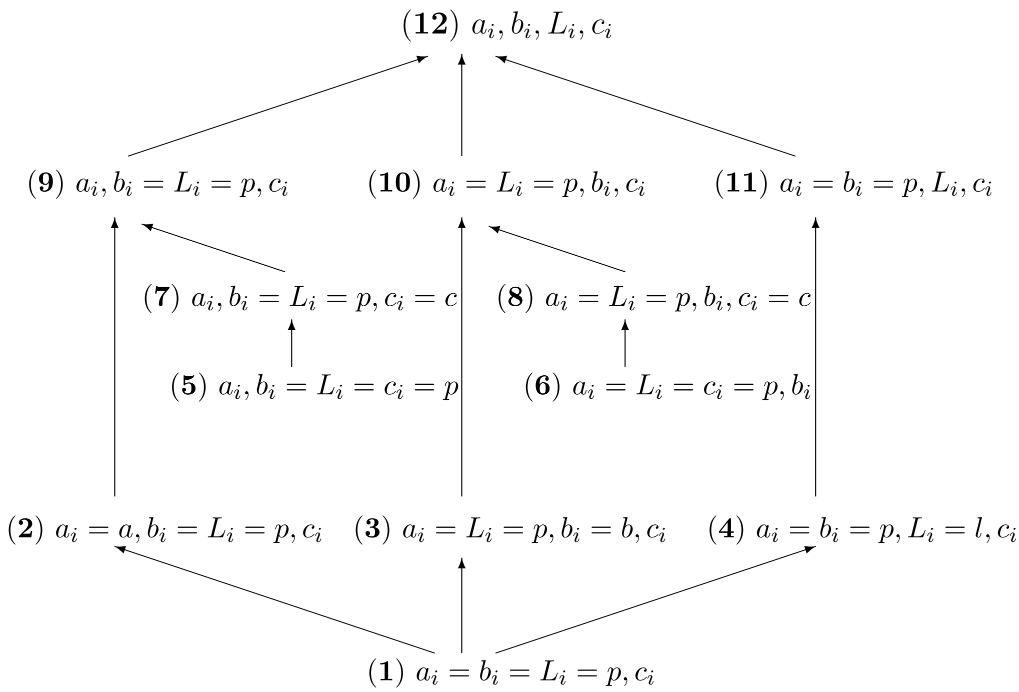

Figure 3: Problems classification

In this work, we present the complexity of the sub-problems (9) and (10) of the problem of the flow shop on two machines with coupled tasks on the first machine with the objective to minimize the makespan.

3 The first subproblem

In this section, we Consider the problem $F2/Coup-Opr(1), a_i, b_i = L_i = p, c_i/C_{max}$, abbreviated in the following as problem $F2C(a_i, b_i = L_i = p, c_i)$. We show that the problem $F2C(a_i, b_i = L_i = p, c_i)$ is NP-hard using a reduction of the partition problem with equal size, which is known to NP-Hard, [10].

The Partition with Equal Size problem (PES) is stated as follows : Given a set $E = \{e_1, e_2, ..., e_{2n}\}$ of $2n$ positives integers, where $\sum_{i=1}^{2n} e_i = 2B$ for some integer B. Does there exist a partition of E into two disjoint subsets E_1 and E_2 such as $\sum\limits_{i \in E_1} e_i = \sum\limits_{i \in E_2} e_i = B$ and $|E_1| = |E_2| = n$? This problem remains NP-hard even each $e_i > 1$. In our proof we assume that all $e_i > 1$.

Given an arbitrary instance of PES, we build an instance $\mathcal{I}$ of problem $F2C(a_i, b_i = L_i = p, c_i)$ with a set of $4n + 2$ jobs as follows:

- Jobs of type U, denoted U_i, $i = 1, \ldots, 2n$;
- n identical jobs denoted V;
- $n + 1$ identical jobs denoted W;
- One job denoted T;

For all the jobs, we set $L_i = b_i = p$, $i = 1, \ldots, 4n + 2$ where $p > B$. Processing times of jobs on machine M_1 and M_2 are given in the Table 3.

Jobs	a_i	c_i
U_i, $i = 1, \ldots, 2n$	$p - e_i$	e_i
V	$p + 1$	$4p$
W	p	0
T	$B + p + 1 - n$	$(4n + 1)p - 2B$

Table 1: Jobs Processing times

Let the threshold for the makespan be y, where $y = 4(2n + 1)p + 1$.

In the following, the notation $(VW)-job$ refers to the composite job (VW) in which jobs V and W are interleaved and the first operation of V is at the first position. Furthermore, the composite (VW) can be seen as compact job with processing times $4p + 1$ and $4p$ on machines M_1 and M_2, respectively, and has a time-lag $l = -p$, and can be scheduled as shown in Figure 4.

In an instance $\mathcal{I}$ of the scheduling problem $F2C(a_i, b_i = L_i = p, c_i)$, a schedule is said to be feasible if the makespan is lower or equal to y. The decision problem asks: is there a feasible schedule to the problem $F2C(a_i, b_i = L_i = p, c_i)$ with makespan less than or equal to y? We have the following results.

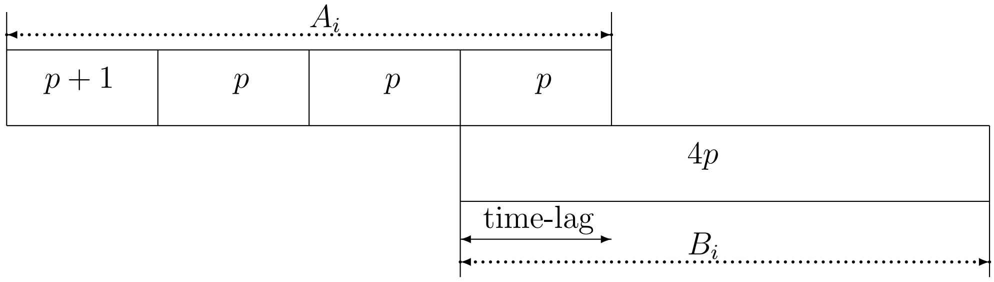

Figure 4: Example of composite job (VW)

Lemma 1 *Given an instance of the The Partition with Equal Size problem, if there is a solution to this instance, there exists a solution to the corresponding instance of the problem $F2C(a_i, b_i = L_i = p, c_i)$ with makespan less or equal to y.*

Proof. If there is a solution to an instance of The Partition with Equal Size problem, i.e., there is a partition E_1 and E_2 of E such that $\sum_{i\in E_1} e_i = \sum_{i\in E_2} e_i = B$ where E_1 and E_2 each contains exactly n elements, then we can schedule the jobs of the corresponding instance $\mathcal{I}$ as follows.

Let J_1 and J_2 be the subset of the $U-$Jobs corresponding to the subsets E_1 and E_2, respectively. Then the desired schedule S exists where the completion time $C_{max}(S)$ of schedule S is equal to $4(2n+1)p+1$. In this schedule, the $U-$Jobs of J_1 (J_2) are interleaved with $V-$Jobs ($W-$Jobs), and one job W is interleaved with job T. Let $(VU)-$Jobs $((VU)_1, \ldots, (VU)_n)$, $(WU)-$Jobs $((WU)_1, \ldots, (WU)_n)$ and $(WT)-$job be the composite jobs obtained by the interleaving operation. Schedule S is constructed as follows (see Figure 6): start by scheduling $(VU)-$jobs, followed by $(TW)-$job then finish with $(WU)-$jobs. The order of composite jobs of set $(VU)-$Jobs and $(WU)-$Jobs in schedule S can be chosen in any order. Figure 6 pictures the resulting schedule. In schedule S there is no idle-time between composite jobs on machine M_1 and M_2, then,

$$C_{max}(S) = p+1+2p+4np+\sum_{i\in J_1} e_i+(4n+1)p-2B+\sum_{i\in J_2} e_i = 4p+8np+1 = 4(2n+1)p+1.$$

■

In the following, we show that if there exists a solution to an instance of the scheduling problem with makespan less or equal to y, then there exists a solution to the corresponding instance of the Partition with Equal Size problem. We show that for any feasible schedule S, the minimum value of makespan is y. Furthermore, to obtain a schedule with makespan equal to y, there must be a solution to the corresponding instance Partition with Equal Size problem. In order to show this result we need to establish the following.

1. All jobs are interleaved.
2. There is no composite (UU)-Jobs.
3. The $T-$job is interleaved with a $W-$Job.

Let X_1, X_2, Y_1, Y_2 be the processing times of interleaved jobs in relation to the position of the job T in schedule S as showed in Figure 5. Clearly, $LB = X_1 + \max\{X_2, Y_2\}$ is a lower bound for the makespan of schedule S.

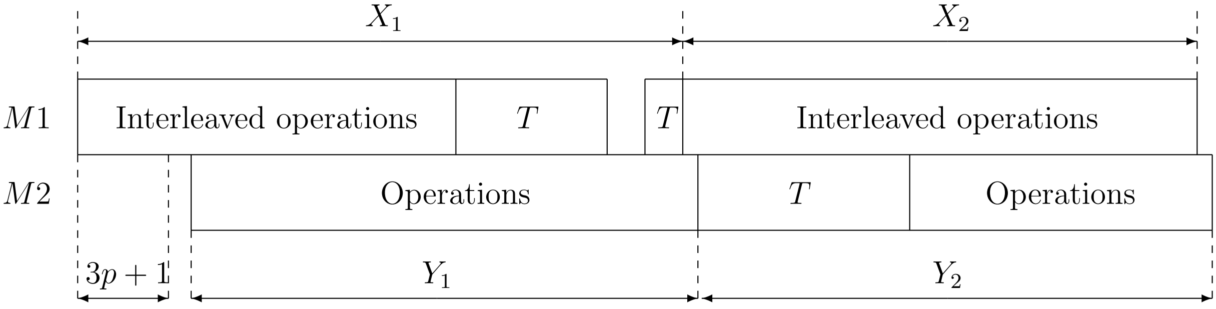

Figure 5: Some notations

Let A_i and B_i be the processing times of composite jobs on machines M_1 and M_2, respectively, and let l_i the time-lag of the composite jobs as showed in Figure 4. Table 2 summarizes the values of A_i, B_i and l_i of all possible composite jobs related our instance of scheduling problem. Furthermore, depending on the processing times of composite jobs, we have,

Jobs	A_i	B_i	l_i
(V, U_i)	$4p+1$	$4p+e_i$	$-p$
(V, W)	$4p+1$	$4p$	$-p$
(T, U_i)	$B+4p+1-n$	$(4n+1)p-2B+e_i$	$-p$
(T, W)	$B+4p+1-n$	$(4n+1)p-2B$	$-p$
(U_i, U_j)	$4p-e_i$	e_i+e_j	$-e_i$
(U_i, W)	$4p-e_i$	e_i	$-e_i$
(W, W)	$4p$	0	0

Table 2: Processing times of composite jobs.

- For any composite job, the jobs V and T can only be in first position and
- The composite job (U_iW) has small processing time than (WU_i) on the first machine, than it is easy to show that in schedule S, all composite job (WU_i) can be transformed into (U_iW) without increasing the value of makespan.

Lemma 2 *In feasible schedule S, All jobs are interleaved.*

Proof. Assume that in schedule S at least two jobs are not interleaved. Let consider that the $U-$jobs are indexed in non increasing order of e_j values of their processing time a_j. Since the total processing time of composite jobs on the first machine is a lower bound of the makespan, then the best interleaving operation of jobs that minimize the total processing time on the first machine is a lower bound of schedule S. Let consider the following interleaving operation of jobs: n composite jobs $((VW))$, one composite job (TW), and $n-1$ composite jobs (U_iU_k) where $i = n, \ldots, 2n-2$; $k = 1, \ldots n-1$, and jobs U_{2n-1} and U_{2n} are scheduled alone, respectively. It is easy to see that this interleaving operation of jobs provides the minimal total processing time of jobs on the first machine when exactly two jobs are not interleaved. Thus,

$$\begin{aligned} C_{max}(S) &\geq (B+p+1-n+3p)+(4np+n)+n.4p-\sum_{i=n-1}^{2n-2} e_i - e_{2n-1} - e_{2n} \\ &\geq 4(2n+1)p+1+2p+B-\sum_{i=n-1}^{2n} e_i = y+2p+B-\sum_{i=n-1}^{2n} e_i \end{aligned}$$

Since $p > B$ and $\sum_{i=n-1}^{2n} e_i < 2B$ then $2p + B - \sum_{i=n-1}^{2n} e_i > 0$, hence, $C_{max}(S) > y$. then in feasible solution all jobs are interleaved. ■

Lemma 3 *In feasible schedule S, there is no composite $(UU)-$Jobs.*

Proof. Let assume that there exists in schedule S' one composite $(UU)-$Jobs. Assume that this $(UU)-$Job is composed of a pair (U_i, U_j) of jobs in which U_i is interleaved with U_j and the other $U-$Jobs are interleaved with the other type, $V-$Jobs, $W-$Jobs and $T-$Job. We distinguish the following cases:

a. The job T is interleaved with a $W-$job, then it remains n $W-$Jobs, n $V-$Jobs and $(2n-2)$ $U-$Jobs. Depending on these remaining jobs, two ways of their interleaving are possible, namely,

   1. (TW), (WW), $(VU_k)_{k\in I_1}$, $(U_kW)_{k\in I_2}$, where $|I_1| = n$, $|I_2| = n-2$ and $I_1 \cup I_2 \cup \{i, j\} = \{1, \ldots, 2n\}$.
   2. (TW), (VW), $(VU_k)_{k\in I_1}$, $(U_kW)_{k\in I_2}$, where $|I_1| = n-1$, $|I_2| = n-1$ and $I_1 \cup I_2 \cup \{i, j\} = \{1, \ldots, 2n\}$.

b. The job T is interleaved with a $U-$job, then it remains n $V-$Jobs, $(n+1)$ $W-$Jobs and $(2n-3)$ $U-$Jobs. Again, depending on these remaining jobs, three ways of their interleaving are possible, namely,

   1. (TU_r), (WW), (WW), $(VU_k)_{k\in I_1}$, $(U_kW)_{k\in I_2}$, where $|I_1| = n$, $|I_2| = n-3$ and $I_1 \cup I_2 \cup \{i, j, r\} = \{1, \ldots, 2n\}$.
   2. (TU_r), (WW), (VW), $(VU_k)_{k\in I_1}$, $(U_kW)_{k\in I_2}$, where $|I_1| = n-1$, $|I_2| = n-2$ and $I_1 \cup I_2 \cup \{i, j, r\} = \{1, \ldots, 2n\}$.
   3. (T, U_r), (VW), (VW), $(VU_k)_{k\in I_1}$, $(U_kW)_{k\in I_2}$, where $|I_1| = n-2$, $|I_2| = n-1$ and $I_1 \cup I_2 \cup \{i, j, r\} = \{1, \ldots, 2n\}$.

In the following we examine the value of makespan of each sequence of interleaving jobs.

Case a.1: According to Mitten's algorithm [8], the optimal schedule of the composite jobs of the case 1.a is given by the sequence $S = \langle (VU_k)_{k\in I_1}, (TW), (U_iU_j), (U_kW)_{k\in I_2}, (WW)\rangle$. According to the processing times of these composite jobs given in Table 2, the parameters X_1, X_2, Y_1 and Y_2 (see figure 5) of the above order are as follow,

$$X_1 = 4np + 3p + B + 1, \quad X_2 = 4pn + p - (\sum_{k\in I_2} e_k + e_i).$$
$$Y_1 = 4np + (\sum_{k\in I_1} e_k), \quad Y_2 = 4np + p - 2B + (\sum_{k\in I_2} e_k + e_i + e_j).$$

Then, the lower bound of $C_{max}(S)$ is

$$\begin{aligned} LB &= X_1 + \max\{X_2, Y_2\} = 8np + 4p + 1 + \max\{B - (\sum_{k\in I_2} e_k + e_i), (\sum_{k\in I_2} e_k + e_i + e_j) - B\} \\ &= y + \max\{B - (\sum_{k\in I_2} e_k + e_i), (\sum_{k\in I_2} e_k + e_i + e_j) - B\} \end{aligned}$$

Since $\max\{B - (\sum_{k\in I_2} e_k + e_i), (\sum_{k\in I_2} e_k + e_i + e_j) - B\} > 0$ then $LB > y$. Thus S is not a feasible solution.

Case a.2: The optimal schedule in this case is $S = \langle (VU_k)_{k\in I_1}, (TW), (VW), (U_iU_j), (U_kW)_{k\in I_2}\rangle$. The parameters X_1, X_2, Y_1 and Y_2 are as follow.

$$X_1 = 4np - p + B, \quad X_2 = 4np + 5p + 1 - (\sum_{k\in I_2} e_k + e_i)$$
$$Y_1 = 4np - 4p + (\sum_{k\in I_1} e_k), \quad Y_2 = 4np + 5p - 2B + (\sum_{k\in I_2} e_k + e_i + e_j).$$

The value of lower bound is

$$\begin{aligned} LB &= X_1 + \max\{X_2, Y_2\} = 8np + 4p + 1 + \max\{B - (\sum_{k\in I_2} e_k + e_i), (\sum_{k\in I_2} e_k + e_i + e_j) - B - 1\}. \\ &= y + \max\{B - (\sum_{k\in I_2} e_k + e_i), (\sum_{k\in I_2} e_k + e_i + e_j) - B - 1\} \end{aligned}$$

Since $\forall k, e_k > 1$, we have $\max\{B - (\sum_{k\in I_2} e_k + e_i), (\sum_{k\in I_2} e_k + e_i + e_j) - B - 1\} > 0$, then $LB > y$. Thus S is not a feasible solution.

Case b.1: The optimal sequence of the composite jobs in this case is $S = \langle (VU_k)_{k\in I_1}, (TU_r), (U_iU_j), (U_kW)_{k\in I_2}, (WW), (WW)\rangle$. The parameters X_1, X_2, Y_1 and Y_2 are,

$$X_1 = 4np + B + 3p + 1, \quad X_2 = 4np + p - (\sum_{k\in I_2} e_k + e_i)$$
$$Y_1 = 4np + \sum_{l\in k_1} e_k, \quad Y_2 = 4np + p - 2B + (\sum_{k\in I_2} e_k + e_i + e_j + e_r).$$

Then the lowed bound of $C_{max}(S)$ is $LB = X_1 + \max\{X_2, Y_2\} = y + \max\{B - (\sum_{l\in I_2} e_l + e_i), (\sum_{l\in I_2} e_l + e_i + e_j + e_k) - B\}$. Since $\max\{B - (\sum_{l\in I_2} e_l + e_i), (\sum_{l\in I_2} e_l + e_i + e_j + e_k) - B\} > 0$ then $LB > y$. Thus, S is not a feasible solution.

Case b.2: The optimal sequence of the composite jobs in this case is $S = \langle (VU_k)_{k\in I_1}, (TU_r), (VW), (U_iU_j), (U_kW)_{k\in I_2}, (WW)\rangle$ and the parameters X_1, X_2, Y_1 and Y_2 are

$$X_1 = 4np - p + B, \qquad X_2 = 4np + 5p - (\sum_{k\in I_2} e_k + e_i)$$
$$Y_1 = 4np - 4p + (\sum_{k\in I_1} e_k), \quad Y_2 = 4np + 5p - 2B + (\sum_{k\in I_2} e_k + e_i + e_r).$$

Then the lowed bound of $C_{max}(S)$ is $LB = X_1 + \max\{X_2, Y_2\} = y + \max\{B - (\sum_{k\in I_2} e_k + e_i + 1), (\sum_{k\in I_2} e_k + e_i + e_j + e_r) - B - 1\}$. Since $\forall i, e_i > 1$, then $\max\{B - (\sum_{k\in I_2} e_k + e_i + 1), (\sum_{k\in I_2} e_k + e_i + e_j + e_r) - B - 1\} > 0$, then $LB > y$. Thus, S is not a feasible solution.

Case b.3: The optimal sequence of the composite jobs of here is $S = \langle (VU_k)_{k\in I_1}, (TU_r), (VW), (VW), (U_iU_j), (U_kW)_{k\in I_2}$ and the parameters X_1, X_2, Y_1 and Y_2 are

$$X_1 = 4np - 5p - 1 + B, \qquad X_2 = 4np + 9p + 2 - (\sum_{k\in I_2} e_k + e_i)$$
$$Y_1 = 4np - 8p + (\sum_{k\in I_1} e_k), \quad Y_2 = 4np + 9p - 2B + (\sum_{k\in I_2} e_k + e_i + e_j + e_r).$$

Then the lowed bound of $C_{max}(S)$ is $LB = X_1 + \max\{X_2, Y_2\} = y + \max\{B - (\sum_{k\in I_2} e_k + e_i), (\sum_{k\in I_2} e_k + e_i + e_j + e_r) - B - 1\}$. Since $\forall i, e_i > 1$, we have $\max\{B - (\sum_{k\in I_2} e_k + e_i), (\sum_{k\in I_2} e_k + e_i + e_j + e_r) - B - 1\} > 0$, then $LB > y$. Thus S is not a feasible solution. Note that if we have more than one interleaving $UU-$jobs then X_2 increases and $LB > y$. ■

Lemma 4 *In a feasible schedule S, the $T-$Job is interleaved with $W-$Job.*

Proof. Let assume that there exists a schedule S' in which T is interleaved with $U-$job. Let U_r the $U-$job interleaved with T. From lemmas 2 and 3, the only possible composite jobs are

1. $(VU_k)_{k\in I_1}$, $(U_kW)_{k\in I_2}$ where $|I_1| = n-2$, $|I_2| = n+1$ and $I_1 \cup I_2 \cup \{r\} = \{1, \ldots, 2n\}$.
2. $(VU_k)_{k\in I_1}$, $(U_kW)_{k\in I_2}$, (VW) where $|I_1| = n-1$, $|I_2| = n$ and $I_1 \cup I_2 \cup \{r\} = \{1, \ldots, 2n\}$.
3. $(VU_k)_{k\in I_1}$, $(U_kW)_{k\in I_2}$, (WW) where $|I_1| = n$, $|I_2| = n-1$ and $I_1 \cup I_2 \cup \{r\} = \{1, \ldots, 2n\}$.

For above cases the optimal sequences are

- Case 1. $S' = \langle (VU_k)_{k\in I_1}, (TU_r), (U_kW)_{k\in I_2} \rangle$
- Case 2. $S' = \langle (VU_k)_{k\in I_1}, (TU_r), (VW), (U_kW)_{k\in I_2} \rangle$
- Case 3. $S' = \langle (VU_k)_{k\in I_1}, (TU_r), (U_kW)_{k\in I_2}, (WW) \rangle$

Similarly to the proof of lemma 3, it is easy to show that for each above case $C_{max}(S') > y$, then in schedule S job R is interleaved with job W. ■

Lemma 5 *If there is a solution to an instance of problem $F2C(a_i, b_i = L_i = p, c_i)$, then there exists a solution to the corresponding instance of the Partition with Equal Size problem.*

Proof. From lemmas 2, 3 and 4, the unique interleaved operations of jobs in schedule S is $(VU_k)_{k\in I_1}$, (TW) and $(U_kW)_{k\in I_2}$ where where $|I_1| = n$, $|I_2| = n$ and $I_1 \cup I_2 = \{1, \ldots, 2n\}$. The optimal sequence of these composite jobs is $S' = \langle (VU_k)_{k\in I_1}, (TW), (U_kW)_{k\in I_2}\rangle$. The parameters X_1, Y_1, X_2 and Y_2 of this sequence are,

$$X_1 = 4np + p + 1 + B, \quad X_2 = 4np + p - \sum_{k\in I_2} e_k$$
$$Y_1 = 4np + \sum_{k\in I_1} e_k, \qquad Y_2 = 4np + p - 2B + \sum_{k\in I_2} e_k.$$

The lowed bound of $C_{max}(S)$ is $LB = X_1 + \max\{X_2, Y_2\} = y + \max\{B - \sum_{k\in I_2} e_k, \sum_{k\in I_2} e_k - B\}$. since $C_{max}(S) \le y$, then $\max\{B - \sum_{k\in I_2} e_k, \sum_{k\in I_2} e_k - B\} = 0$. Thus $\sum_{k\in I_2} e_k = B$ and $\sum_{k\in I_1} e_k = B$. since $|I_1| = |I_2| = n$, then we obtain a solution for the PES problem. ■

From Lemmas 1 and 5, we know that there is a solution to the instance of the Partition with Equal Size problem if and only if there is a solution to the corresponding instance $\mathcal{I}$ of the scheduling problem. Therefore, the following result.

Theorem 6 *The problem $F2C(a_i, b_i = L_i = p, c_i)$ is binary NP-hard.*

4 The second subproblem

In this section, we show that the problem $F2/Coup - Opr(1), a_i = L_i = p, b_i, c_i/C_{max}$, abbreviated in the following as problem $F2C(a_i = L_i = p, b_i, c_i)$ is NP-hard using a reduction of the Partition problem with Equal Size used in section **??**.

Given an arbitrary instance of the Partition problem with Equal Size used in section **??**, we build an instance $(\mathcal{I})$ of problem $F2C(a_i = L_i = p, b_i, c_i)$ with a set of $4n + 4$ jobs as follows:

- Jobs of type U, denoted U_i, $i = 1, \ldots, 2n$;
- n identical jobs denoted V;
- $n + 2$ identical jobs denoted W;
- One job denoted T;
- One job denoted R;

For all the jobs, we set $a_i = L_i = p$, $i = 1, \ldots, 4n + 4$ where $p > B$. Processing times of jobs on machine M_1 and M_2 are given in the table 3.

Let the threshold for the makespan be y, where $y = 9p(n + 1)p + n + 2$. We have the following result.

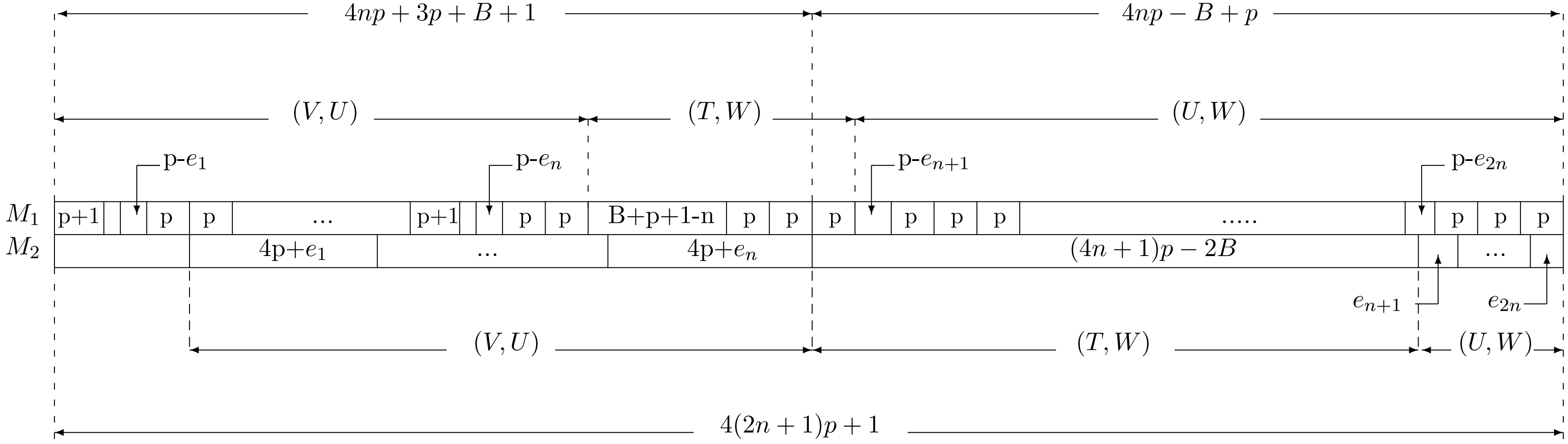

Figure 6: Schedule S

Jobs	b_i	c_i
U_i, $i = 1, \dots, 2n$	$p - e_i$	e_i
V	$p + 1$	$5p + 1$
W	p	0
T	$(n + 2)p + B + 1$	$4p(n + 1) - 2B + 1$
R	$p + 1$	0

Table 3: Jobs Processing times

Lemma 7 *Given an instance of the the Partition with Equal Size problem, if there is a solution to this instance, there exists a solution to the corresponding instance of the problem $F2C(a_i = L_i = p, b_i, c_i)$ with makespan less or equal to y.*

Proof. Assume that the Partition with Equal Size problem problem has a solution, and let E_1 and E_2 be the required subset of E such that $\sum_{i \in E_1} e_i = \sum_{i \in E_2} e_i = B$ and $|E_1| = |E_2| = n$. Let J_1 and J_2 be the subset of the $U-$Jobs corresponding to the subsets E_1 and E_2, respectively. Then the desired schedule S exists where the completion time $C_{max}(S)$ of schedule S is equal to $9p(n + 1)p + n + 2$. The composite jobs and their schedule is given by sequence $S = \langle (U_k V)_{k \in J_1}, (WT), (WU_k)_{k \in J_2}, (WR) \rangle$. ■

Assume now that the there exists a solution to an instance of the scheduling problem with makespan less or equal to y, then we show that there exists a solution to the corresponding instance of the Partition with Equal Size problem. In order to show this result we need to establish the following.

1. All jobs are interleaved.
2. There is no composite (UU)-Jobs.
3. The $T-$job is interleaved with a $W-$Job.
4. The $R-$job is interleaved with a $W-$Job

As presented in section **??** Table 4 summarizes the values of A_i, B_i and l_i of all possible composite jobs related to the instance of scheduling problem.

The following results establish the statement (1)-(4).

Lemma 8 *In a feasible schedule S, All jobs are interleaved.*

Proof. The proof is similar to the proof of lemma 2 ■

Lemma 9 *In a feasible schedule S, there is no composite $(UU)-$Jobs.*

Jobs	A_i	B_i	l_i
(U_i, V)	$4p + 1$	$5p + 1 + e_i$	$-e_i$
(W, V)	$4p + 1$	$5p + 1$	0
(W, U_i)	$4p - e_i$	e_i	0
(U_i, R)	$4p + 1$	e_i	$-e_i$
(W, W)	$4p$	0	0
(W, R)	$4p + 1$	0	0
(U_i, U_j)	$4p - e_i$	$e_i + e_j$	$-e_i$
(U_i, T)	$B + (n + 5)p + 1$	$4p(n + 1) - 2B + e_i$	$-e_i$
(W, T)	$B + (n + 5)p + 1$	$4p(n + 1) - 2B$	0

Table 4: Processing times of composite jobs.

Proof. The proof is similar to the proof of lemma 3 ■

Lemma 10 *In a feasible schedule S, the $T-$Job is interleaved with $W-$Job.*

Proof. The proof is similar to the proof of lemma 4 ■

Lemma 11 *In a feasible schedule S, the $R-$Job is interleaved with $W-$Job.*

Proof. Let assume that there exists a schedule S' in which R is interleaved with $U-$job. Let U_r be the $U-$job interleaved with R. From lemmas 8, 9 and 10, the only possible composite jobs are

1. (WT), (U_rR), $(U_kV)_{k\in I_1}$, $(WU_k)_{k\in I_2}$, and (WW) where $|I_1| = n$, $|I_2| = n - 1$ and $I_1 \cup I_2 \cup \{r\} = \{1, \ldots, 2n\}$.
2. (WT), (U_rR), (WV), $(VU_k)_{k\in I_1}$, and $(U_kW)_{k\in I_2}$, where $|I_1| = n - 1$, $|I_2| = n$ and $I_1 \cup I_2 \cup \{r\} = \{1, \ldots, 2n\}$.

For above cases the optimal sequences are

- Case 1. $S' = \langle (VU_k)_{k\in I_1}, (WT), (U_kW)_{k\in I_2}, (U_rR), (WW)\rangle$
- Case 2. $S' = \langle (VU_k)_{k\in I_1}, (WV), (WT), (U_kW)_{k\in I_2}, (U_rR)\rangle$

Similarly to the proof of lemma 3, it is easy to show that for each above case $C_{max}(S') > y$, then in schedule S job R is interleaved with job W. ■

Lemma 12 *If there is a solution to an instance of problem $F2C(a_i = L_i = p, b_i, c_i)$, then there exists a solution to the corresponding instance of the Partition with Equal Size problem.*

Proof. From lemmas 8, 9, 10 and 11, there unique interleaved operations of jobs in schedule S is (WT), WR, $(U_kV)_{k\in I_1}$, $(WU_k)_{k\in I_2}$ where $|I_1| = n$, $|I_2| = n$ and $I_1 \cup I_2 = \{1, \ldots, 2n\}$ The optimal sequence of these composite jobs is $S' = \langle (U_kV)_{k\in I_1}, (WT), (WU_k)_{k\in I_2}(WR)\rangle$. The parameters X_1, X_2 and Y_2 of this sequence are,

$$X_1 = 5np + 5p + B + n + 1, \qquad X_2 = 4np + 4p + 1 - \sum_{k\in I_2} e_k$$
$$Y_2 = 4np + 4p + 1 - 2B + \sum_{k\in I_2} e_k.$$

The lowed bound of $C_{max}(S)$ is $LB = X_1 + \max\{X_2, Y_2\} = y + \max\{B - \sum_{k\in I_2} e_k, \sum_{k\in I_2} e_k - B\}$. since $C_{max}(S) \leq y$, then $\max\{B - \sum_{k\in I_2} e_k, \sum_{k\in I_2} e_k - B\} = 0$. Thus $\sum_{k\in I_2} e_k = B$ and $\sum_{k\in I_1} e_k = B$. since $|I_1| = |I_2| = n$, then we obtain a solution for the PES problem. ■

From Lemmas 7 and 12, we know that there is a solution to the instance of the Partition with Equal Size problem if and only if there is a solution to the corresponding instance $\mathcal{I}$ of the scheduling problem. Therefore, the following result.

Theorem 13 *The problem $F2C(a_i = L_i = p, b_i, c_i)$ is binary NP-hard.*

5 Conclusion

In this article, we studied the complexity of flow shop problem with coupled tasks. Our problem consists in a flow shop with two machines with coupled tasks on the first machine such as each job consists in two operations on the first machine separated by a time lag and one operation on the second machine in order to minimize the total completion time. The problem is NP-hard in its general form. We have shown that two of its sub-problems are NP-hard.

References

[1] Ageev A.A. and Baburin A.E. *Approximation algorithms for UET scheduling problems with exact delays*, Operations Research Letters, 35: 533-540, 2007.

[2] Ageev A.A. and Kononov A.V. *Approximation Algorithms for Scheduling Problems with Exact Delays*, In WAOA, pp.1-14,2006.

[3] Ahr D.,Békési J., Galambos G., Oswald M. and Reinelt G.,*An exact algorithm for scheduling identical coupled tasks*, Mathematical Methods of Operational Research, 59:193-203, 11,June 2004.

[4] Blazewicz J., Ecker K., Kis T., Potts C.N., Tanas M and Whitehead J., *Scheduling of coupled tasks with unit processing times*, Journal of Scheduling, 13: 453-461, 2010.

[5] Brauner N., Finke G., Lehoux-Lebacque V., Potts C. and Whitehead J., *Scheduling of coupled tasks and one-machine no-wait robotic cells*, Computers and Operational Research, 36(2): 301-307, 2009.

[6] Dell'Amico M., *Shop problems with two machines and time lags*, Operations Research, 44: 777-787, 1996.

[7] Dell'Amico M. and Vaessens R.J.M., *Flow and open shop scheduling on two machines with transportation times and machine-independent processing times is NP-hard*, Materiali di discussione 141, Dipartimento di Economia Politica, Universita di Modena, 1996.

[8] Fondrevelle J., Oulamara A. and Portmann M.C., *Permutation flowshop scheduling problem with maximal and minimal time lags*, Computers and Operations Research, 33: 1540-1556, 2006.

[9] Fondrevelle J., Oulamara A. and Portmann M.C., *Permutation flow shop scheduling problems with time lags to minimize the weighted sum of machine completion times*, International Journal of Production Economics, 112: 168-176, 2008.

[10] Garey, M. R. and Johnson, D. S. Computers and Intractability: A Guide to the Theory of NP-Completeness. Victor Klee, ed. A Series of Books in the Mathematical Sciences. San Francisco, Calif.: W. H. Freeman and Co. 1979.

[11] Graham R.L., Lawler E.L., Lenstra J.K. , and Rinnooy Kan A.H.G., *Optimization and approximation in deterministic sequencing and scheduling: a survey*, Annals of Discrete Mathematics 5, pp. 287-326, 1979.

[12] Johnson S.M., *Optimal two and three stage production schedules with setup time included*, Naval Research Logistics Quarterly 1, pp. 61-67, 1954.

[13] Lenstra J.K., *Private communication*, 1991.

[14] Mitten L.G., *Sequencing n Jobs on Two Jobs with Arbitrary Time Lags*, Management Science, 5(3), pp. 293-298, 1958.

[15] Orman A.J. and Potts C.N., *On the complexity of coupled-Task Scheduling*, Discrete Applied Mathematics 72:141-154, 1997.

[16] Shapiro R.D., *Scheduling Coupled Tasks*, Naval Research logistics quartely 20: pp. 489-498, 1980.

[17] Simonin G., Giroudeau R. and Konig J.C., *Polynomial-time algorithms for scheduling problem for coupled-tasks in presence of treatment tasks*, Electronic Notes in Discrete Mathematics, 36: pp. 647-654, 2010.

[18] Yu W., Hoogeveen H. and Lenstra J.K., *Minimizing makespan in a two-machine flow shop with delays and unit-time operations is NP-hard*, Journal of Scheduling, 7: pp. 333-348, 2004.

Les Annales RECITS (http://www.lrecits.usthb.dz)
Vol. 01, 2014, pages 27-32

A combinatorial contribution to the multinomial Chu-Vandermonde convolution

Hacène Belbachir

USTHB, Faculty of Mathematics, RECITS Laboratory, DG-RSDT
BP 32, El Alia, 16111 Bab Ezzouar, Algiers, Algeria.

`hbelbachir@usthb.dz` and `hacenebelbachir@gmail.com`

Abstract: A combinatorial proof to multinomial Chu-Vandermonde convolution is given with an extension to polynomial case. We deal also with some probabilistic contributions as a simple application to random matrices.

Keywords: Chu-Vandermonde convolution; Hypergeometric distribution probability; Random matrix

---

Résumé : Une preuve combinatoire pour la convolution multinomial de Chu-Vandermonde est donné avec une extension au cas polynomiale. Nous donnons aussi une contributions probabilistes comme une application simple aux matrices aléatoires.

Mots clés : Convolution de Chu-Vandermonde; distribution hypergéométrique; matrice aléatoire

---

1 Introduction

Chu-Vandermonde identity states that for all $m, n, r \in \mathbb{N}$, we have the following

$$\binom{n+m}{r} = \sum_k \binom{n}{k}\binom{m}{r-k}, \tag{1}$$

where $\binom{n}{k} = \begin{cases} \frac{n!}{k!(n-k)!} & \text{for } \ 0 \le k \le n, \\ 0 & \text{otherwise.} \end{cases}$

$\binom{n}{k}$ is the binomial coefficient, combinatorially it counts the number of ways to take k members from n candidates.

Relation (1) admits the following well known extension: given $s \in \mathbb{N}$ and $n_1, n_2, \ldots, n_s$, r nonnegative integers, the following identity holds

$$\binom{n_1 + n_2 + \cdots + n_s}{r} = \sum_{k_1+k_2+\cdots+k_s=r} \binom{n_1}{k_1}\binom{n_2}{k_2}\cdots\binom{n_s}{k_s}. \tag{2}$$

Divided by the left expression of both sides of identities (1) and (2) respectively, the summands terms are interpreted as the hypergeometric and the polyhypergeometric probability distributions.

Our aim is to give some extensions of Chu-Vandermonde identity to the multinomial case. This work completes those of Gould [2, 3].

2 Multinomial Chu-Vandermonde identity

We use the following notation for the multinomial coefficient, for all $n_1, n_2, \ldots, n_t$ and $n \in \mathbb{Z}$,

$$\binom{n}{n_1, n_2, \ldots, n_t} = \begin{cases} \frac{n!}{n_1!n_2!\cdots n_t!} & \text{if } n_1, n_2, \ldots, n_t \text{ ares integers with sum } n, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The main advantages of such an interpretation of multinomial coefficient is that one can omit the use of exact limit in sums like $\sum_{\substack{n_1,n_2,\ldots,n_t/ \\ n_1+n_2+\cdots+n_t=n}} \binom{n}{n_1,n_2,\ldots,n_t}$ by simply writing $\sum_{n_1,n_2,\ldots,n_t} \binom{n}{n_1,n_2,\ldots,n_t}$ instead. In the sequel, for the sake of convenience, we exploit this kind of allowance.

The multinomial coefficient $\binom{n}{n_1,n_2,\ldots,n_t}$ counts the number of ways to constitute t distinguishable committees from n candidates such that the first committee contains n_1 indistinguishable members, the second contains n_2 indistinguishable members, ... and the t^{th} contains n_t indistinguishable members.

Theorem 1 *Let $t \in \mathbb{N}$ and $r_1, r_2, \ldots, r_t, n, m$ be nonnegative integers, the following identity holds*

$$\binom{n+m}{r_1, r_2, \ldots, r_t} = \sum_{k_1, k_2, \ldots, k_t} \binom{n}{k_1, k_2, \ldots, k_t} \binom{m}{r_1 - k_1, r_2 - k_2, \ldots, r_t - k_t}. \tag{4}$$

Proof. For a combinatorial proof see Theorem 2.

We sketch an algebraic proof, it suffices to develop

$$(x_1 + x_2 + \cdots + x_t)^{n+m} = (x_1 + x_2 + \cdots + x_t)^n (x_1 + x_2 + \cdots + x_t)^m$$

and to identify the coefficient of $x_1^{r_1} x_2^{r_2} \cdots x_t^{r_t}$ for both sides. ■

Now, give the generalized multinomial Chu-Vandermonde identity.

Theorem 2 *Let $t, s \in \mathbb{N}$ and $r_1, r_2, \ldots, r_t, n_1, n_2, \ldots, n_s$ be nonnegative integers, the following identity holds*

$$\binom{n_1 + n_2 + \cdots + n_s}{r_1, r_2, \ldots, r_t} = \sum_{k_{ij}} \binom{n_1}{k_{11}, k_{12}, \ldots, k_{1t}} \cdots \binom{n_s}{k_{s1}, k_{s2}, \ldots, k_{st}} \tag{5}$$

where the summation is taken over all $k_{ij},\ i = 1, \ldots, s;\ j = 1, \ldots, t$ such that $k_{1l} + k_{2l} + \cdots + k_{sl} = r_l,\ l = 1, \ldots, t$.

Proof. We sketch an algebraic proof, it suffices to develop

$$(x_1 + x_2 + \cdots + x_t)^{n_1 + \cdots + n_s} = (x_1 + x_2 + \cdots + x_t)^{n_1} \cdots (x_1 + x_2 + \cdots + x_t)^{n_s}$$

and to identify the coefficient of $x_1^{r_1} x_2^{r_2} \cdots x_t^{r_t}$ for both sides.

For a combinatorial proof: consider s different nationalities of students in the university with t levels of learning: the first year, the second year, ..., and the t^{th} year. From students composed by n_1 of nationality 1,...,n_s of nationality s ; we want to choose r_1 students of the first year, r_2 students of the second year,..., and r_t students of the t^{th} year. We do it by summing over all possible values of $k_{i,1}$ of the 1^{st} year ; $k_{i,2}$ of the 2^{nd} year; ... ; $k_{i,t}$ of the t^{th} year of nationality i for $i = 1, \ldots, s$. such that the sum of student of the same year j correspond to r_j. ■

3 A poly-multi-hypergeometric distribution probability

Where both sides of (5) are divided by the expression on the left, the sum be 1, in this case also we can interpret the terms of the sum as probabilities. The resulting probability distribution can be "named" as "*poly-multi-hypergeometric distribution*". "poly" for $n_1, n_2, \ldots, n_s$ and "multi" for $r_1, r_2, \ldots, r_t$.

Here we formulate an other example, to express the probability distribution that is the probability distribution of r_1 balls with number 1, r_2 balls with number 2, ..., and r_t balls with number t from an urn containing N balls with proportion p_1 for the color 1, p_2 for the color 2, ..., p_s for the color s,

$$K = \begin{bmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,t} \\ k_{2,1} & k_{2,2} & & k_{2,t} \\ \vdots & \vdots & & \vdots \\ k_{s,1} & k_{s,2} & \cdots & k_{s,t} \end{bmatrix} \begin{matrix} n_1 = Np_1 \\ n_2 = Np_2 \\ \\ n_s = Np_s \end{matrix}$$

then the probability to get a matrix distribution as a contingence table brewing the lines sums and the column sums is given by

$$P(K) = \frac{\binom{Np_1}{k_{1,1}, k_{1,2}, \ldots, k_{1,t}} \cdots \binom{Np_s}{k_{s,1}, k_{s,2}, \ldots, k_{s,t}}}{\binom{N}{r_1, r_2, \ldots, r_s}}, \tag{6}$$

We can consider this example as a way to introduce a random matrix. Also, by normalizing, we notice that K can be viewed as a double stochastic matrix.

4 Complex variant of Chu-Vandermonde identity

The identity (4) generalizes to non-integer arguments. We have to specify the way of this extension. Let $x \in \mathbb{C}$ and $r \in \mathbb{Z}$,we define

$$\binom{x}{r} = \begin{cases} \frac{1}{r!} x(x-1)\cdots(x-r+1), & r \geq 1 \\ 1, & r = 0 \\ 0, & r < 0 \end{cases}.$$

For $r > 0$, it is a polynomial of degree r.

$$\begin{aligned} & \binom{x}{r_1, r_2, \ldots, r_{t-1}, x - \sum_j r_j} \\ = & \binom{x}{r_1}\binom{x-r_1}{r_2}\cdots\binom{x-r_1-\cdots-r_{t-1}}{r_{t-1}}, \\ = & \frac{x(x-1)\cdots(x-r_1+1)}{r_1!}\frac{(x-r_1)\cdots(x-r_1-r_2+1)}{r_2!}\cdots \\ & \qquad \cdots\frac{(x-r_1-\cdots-r_{t-2})\cdots(x-r_1-\cdots-r_{t-1}+1)}{r_{t-1}!}, \end{aligned}$$

it is a polynomial of degree $r_1 + r_2 + \cdots + r_{t-1}$.

Lemma 3 *We have*

$$\binom{x}{r_1, r_2, \ldots, r_{t-1}, x-r} = \binom{r}{r_1, \ldots, r_{t-1}} \binom{x}{r}. \tag{7}$$

It is implicit that $r = \sum_{j=1}^{t-1} r_j$.

It is well known that for general complex valued x and y, Chu-Vandermonde identity takes the following form

$$\binom{x+y}{r} = \sum_{k=0}^{\infty} \binom{x}{k} \binom{y}{r-k}$$

Theorem 4 *For all x and y complex numbers and all $r_1, r_2, \ldots, r_{t-1}$ nonnegative integers, we set $r = r_1 + \cdots + r_{t-1}$ we have the following*

$$\begin{aligned} &\binom{x+y}{r_1, \ldots, r_{t-1}, x+y-r} \\ = &\sum_{k_1, \ldots, k_{t-1}} \binom{x}{k_1, \ldots, k_{t-1}, x - \sum_j k_j} \binom{y}{r_1 - k_1, \ldots, r_{t-1} - k_{t-1}, y - r + \sum_j k_j} \end{aligned} \tag{8}$$

Proof. Set $\sum_j k_j = k$, by Lemma3 and Theorem1, we have

$$\begin{aligned} &\binom{x+y}{r_1, \ldots, r_{t-1}, x+y-r} \\ = &\binom{r}{r_1, \ldots, r_{t-1}} \binom{x+y}{r} \\ = &\binom{r}{r_1, \ldots, r_{t-1}} \sum_k \binom{x}{k} \binom{y}{r-k} \\ = &\sum_{k_1, \ldots, k_{t-1}} \binom{l}{k_1, \ldots, k_{t-1}} \binom{r-l}{r_1 - k_1, \ldots, r_{t-1} - k_{t-1}} \sum_k \binom{x}{k} \binom{y}{r-k} \end{aligned}$$

In particular, for $l = k$

$$\begin{aligned} &\binom{x+y}{r_1, \ldots, r_{t-1}, x+y-r} \\ = &\sum_{k \geq 0} \sum_{\substack{k_1, \ldots, k_{t-1} \\ \sum k_j = k}} \binom{k}{k_1, \ldots, k_{t-1}} \binom{x}{k} \binom{r-k}{r_1 - k_1, \ldots, r_{t-1} - k_{t-1}} \binom{y}{r-k}, \end{aligned}$$

we conclude by the Lemma. ■

Now, we give the complex version of the generalized multinomial Chu-Vandermonde identity.

Theorem 5 *For all $x_1, \ldots, x_s$ complex numbers and all $r_1, r_2, \ldots, r_{t-1}$ nonnegative integers, set $r = \sum_{j=1}^{t-1} r_j$ and $x = \sum_{j=1}^{s} x_j$, we have the following*

$$\begin{aligned} &\binom{x_1 + \cdots + x_s}{r_1, \ldots, r_{t-1}, x - r} \\ = &\sum_{k_{i,j}} \binom{x_1}{k_{1,1}, \ldots, k_{1,t-1}, x_1 - \sum_j k_{1,j}} \cdots \binom{x_s}{k_{s,1}, \ldots, k_{s,t-1}, x_s - \sum_j k_{s,j}}, \end{aligned}$$

where the summation is taken aver all $k_{i,j}, i = 1, \ldots, s,\ j = 1, \ldots, t-1$ such that $k_{1,l} + k_{2,l} + \cdots + k_{s,l} = r_l,\ l = 1, \ldots, t-1$.

Proof. We leave the proof to the reader, it suffices to consider the proof of Theorem4 for s arguments. ∎

References

[1] Comtet L. (1970). *Analyse combinatoire*. Puf, Coll. Sup. Paris, Vol. 1 and Vol. 2.

[2] Gould, H. W.; Srivastava, H. M. (1997). *Some combinatorial identities associated with the Vandermonde convolution*. Appl. Math. Comput. **84**, no. 2-3, 97–102.

[3] Gould, H. W. (1956). *Some generalizations of Vandermonde's convolution*. Amer. Math. Monthly 63, 84–91.

Les Annales RECITS (http://www.lrecits.usthb.dz)
Vol. 1, 2014, pages 33-38

Méthode Exacte Pour le Problème du Stable Multi-Objectif

Badjara Mohamed El Amine et Chergui Mohamed El-Amine

Laboratoire RECITS - USTHB - Bab Ezzouar - Alger

mbadjara@usthb.dz mchergui@usthb.dz

Abstract: An exact method per branch and bound is proposed for the generation of efficient vertex packing set in a graph that his vertices are valued with a weight's vector. She exploits the data structure of the problem to devid it, in an arborescence strucuture, to independent sub-problems of reduce size which can resolved with a general method for multi-objective integer linear problems.

Keywords: vertex packing in graph ; integer linear program ; branch and bound ; multi-objective program ; efficient solution.

Résumé : Une méthode exacte par séparation-évaluation est mise en œuvre pour la génération de l'ensemble des stables efficaces dans un graphe dont chaque sommet est valué par un vecteur poids. Elle exploite la structure des données du problème du stable pour le décomposer, dans une structure arborescente, en sous problèmes indépendants de tailles réduites, chacun pouvant être résolus par une méthode générale de résolution d'un problème multi-objectif linéaire discret.

Mots clés : Stable dans un graphe ; programme linéaire en nombres entiers ; séparation et évaluation ; programme multi-objectif ; solution efficace.

1 Introduction

Le problème du stable est un problème classique d'optimisation combinatoire connu pour être de type *NP-difficile* et il en est de même dans sa version multi-objectif, noté $MOISP$. Ce problème n'a pas reçu suffisamment l'attention des chercheurs à notre connaissance, et très peu de travaux existent seulement pour le problème général connu sous le nom "Set Packing Problem" dans le cas bi-objectif [2, 3]. Nous proposons une méthode exacte par séparation et évaluation pour trouver tous les stables efficaces de $MOISP$, qui exploite la structure de la matrice des contraintes du problème du stable avant de faire appel à une méthode générale dédiée à la résolution de problèmes d'optimisation multi-objectif linéaires en variables discrètes ($MOILP$). Nous montrons que notre méthode est plus performante que la méthode générale décrite par Chergui et al. dans [1] pour la résolution de $MOILP$.

2 Principe de la méthode

Commençant par donner le modèle général de $MOISP$. Soit $G = (V, E, W)$ un graphe dont V est l'ensemble sommets, E l'ensemble des arêtes et W la matrice de poids des sommets ou chaque colonne j correspond à un vecteur de poids du $j^{ème}$ sommet. Alors le modèle s'écrit :

$$P : \left\{ \begin{array}{rcl} Max \quad Z(x) & = & Wx \\ Ax & \leq & b \end{array} \right. ; x \in \{0,1\}^n$$

où : A est la matrice d'incidence transopsée du graphe G.
b le vecteur du second membre dont tous ses composantes valent 1.
n le nombre de sommets du graphe G.

Un vecteur $Z(x) \in \mathbb{R}^p$ domine un autre vecteur $Z(y) \in \mathbb{R}^p$ si $Z(x) \geq Z(y)$ et $Z(x) \neq Z(y)$. Dans ce cas, la solution y, qui est un stable dans le graphe G, n'est pas efficace pour (P). Le point idéal I de (P) est un point de l'espace des critères et a pour coordonnées $(Z_1^*, \cdots, Z_p^*)$, où Z_k^* est le poids maximum d'un stable de G relativement au critère k, $k = (1, p)$. Généralement, le point I n'est pas réalisable [4].

Le problème (P) a une structure particulière, la matrice des contraintes est creuse, chaque ligne contient exactement deux un (01), et le vecteur du second membre est égale à un (01) pour chacune de ses composantes.

Le principe de la méthode réside dans l'exploitation de cette structure et l'utilisation du principe de la séparation et évaluation. Initialement, on supprime tous les sommets ayant leurs poids négatifs ou nuls avec au moins une composante strictement négative car ce type de sommets ne peut appartenir à aucun stable efficace. Viennent ensuite, les étapes suivantes :

A. Opération de tri : L'ordre de traitement des sommets du graphe G dans l'arborescence de recherche ce fait sur la base d'une opération de tri de ces derniers préalablement établie. Pour cela, plusieurs opérations de tri sont envisageables et nous présentons une opération possible basée sur la combinaison du tri de la domination des poids des sommets et le tri décroissant des sommets.
A chaque sommet j de G est associé un ensemble $D(j)$ de sommets tels que $Z(j)$ domine $Z(t), \forall t \in D(j)$. L'ordre de traitement des sommets j de G dans l'arborescence se fait selon l'ordre décroissant de la quantité $\alpha(j)$ tel que :

$$\alpha(j) = |D(j)| + |\Gamma(j)|$$

où $\Gamma(j)$ est l'ensemble des sommets adjacents au sommet j. D'autres opérations de tri ont été aussi testées, mais les meilleurs résultats sont obtenus moyennant l'opération décrite ci-dessus.

B. Séparation : Une fois que le tri des sommets est réalisé, la séparation est faite sur la base de la règle "retenir" ou "ne pas retenir" un sommet j dans des solutions stables efficaces ($x_j = 1$ ou $x_j = 0$). En outre, le fait de retenir un sommet aura un effet de propagation de contraintes sur ses voisins dans ce sens que ; tous ses voisins seront automatiquement rejetés et beaucoup de contraintes du programme $MOISP$ deviennent saturées, ce qui a pour premier effet bénéfique la diminution de la taille de $MOISP$. Suite à cet "effet domino", le deuxième effet bénéfique réside dans l'apparition possible de sommets isolés dans le sous graphe obtenu.
Le cas correspondant à "ne pas retenir" un sommet entraine sa suppression du graphe courant et dans ce cas aussi, le sous graphe obtenu peut admettre des sommets isolés.
La règle générale de traitement d'un sommet j isolé est la suivante :
- Si tous les poids w_k^j du sommet isolé j sont positifs ou nuls alors, $x_j = 1$.
- S'il existe un poids w_k^j négatif, alors on aura deux cas à traiter : retenir le sommet j ($x_j = 1$) ou rejeter le sommet j ($x_j = 0$).

C. Evaluation : L'évaluation est faite en approximant par $\tilde{I}_l$ le point idéal local I_l de chaque sous problème correspondant à un nœud l de l'arborescence de recherche. Initialement, l'approximation ($\bar{Z}_k$) du maximum de chaque critère k est faite par défaut sur le vecteur $X = (x_j)_{j=1..n}$ avec $x_j = 1$ si le poids $w_k^j \geq 0$ et $x_j = 0$ si le poids $w_k^j < 0$, c'est-à-dire $\bar{Z}_k = \sum_{j \in \{1,\cdots,n\}, w_k^j} w_k^j$.
Lors de la séparation, si un sommet j est retenu dans une branche de l'arborescence, la mise à jour de ($\bar{Z}_j$) est faite par l'ajout des poids négatifs associés au sommet j à ($\bar{Z}_j$). Si le sommet j n'est pas retenu, la mise à jour de ($\bar{Z}_j$) est faite par la soustraction des poids positifs associés au sommet i à ($\bar{Z}_j$). De cette façon, on montre que $\tilde{I}_l$ domine I_l.

D. Principe général : Initialement, on supprime tous les sommets dont le vecteur poids est négatif ou nul, avec au moins une composante strictement négative, pour ensuite évaluer le premier nœud de l'arborescence. Le processus va se poursuivre par la séparation et l'évaluation de chaque nœud de l'arborescence, sachant qu'on a fixé au préalable le nombre de niveaux mesurant la profondeur de l'arborescence. L'effet domino a une incidence directe sur la réduction considérable de la taille du problème. Dans le pire des cas, pour réduire la matrice des contraintes à une matrice nulle, on aura 2^n feuilles dans l'arborescence quand la densité du graphe devient nulle (dans ce cas le graphe est un ensemble de sommets isolés). Donc, la fixation de la profondeur de l'arborescence a pour effet d'éviter l'exploration de 2^n feuilles. Ainsi, il suffit d'exécuter une des méthodes connues dans la littérature pour la résolution des programmes MOILP sur chaque feuille qui correspond à un sous graphe de G de densité non nulle, donc un programme de taille réduite par rapport au programme initiale.

D'autre part, on a dit que la séparation a pour conséquence un effet domino sur la réduction de la matrice courante A des contraintes, correspondant à un sous-graphe G, cet effet est détaillé dans ce qui suit :

- Si un sommet j est retenu ($x_j = 1$), sa colonne correspondante et toutes les lignes dans lesquelles le sommet j apparait, sont supprimées de la matrice A. De plus, tous ses sommets adjacents dans G seront automatiquement supprimés et donc, toutes les colonnes qui leurs correspondantes sont aussi supprimées de A.
- Si un sommet j est supprimé, on supprime sa colonne correspondante dans la matrice A. De plus, chaque ligne qui le contient dans A ne comportera plus qu'un seul élément non nul égal à un, correspondant à son sommet adjacent dans G.

On procédant ainsi, on peut avoir une colonne dans A avec un seul un (01), ce qui correspond à un sommet isolé j dans G obtenu après l'opération de réduction de la matrice A. Si tous ses poids sont positifs ou nuls avec au moins un poids strictement positif, alors on retient le sommet j. D'autre part, l'estimation $\tilde{I}_l$ du point idéal I_l d'un sous problème correspondant à un nœud l de l'arborescence de recherche, a pour effet de sonder ce nœud si $\tilde{I}_l$ est dominé par au moins une solution non dominée générée lors de la résolution de sous problèmes relatifs aux nœuds h, $h < l$.

E. Remarque : Un cas reste à vérifier, si tous les poids w_i^j des sommets sont négatifs alors, chacun des sommets j de G dont le vecteur critère est non dominé, constitue un stable efficace. D'autre part, si tous les poids w_i^j des sommets sont égaux alors, le problème (P) revient à déterminer tous les stables de cardinalité maximale et de fait, l'algorithme que nous proposons les génère tous.

3 Expérimentation numérique

La méthode des ensembles efficaces complets (EEC) décrite dans [1] et notre méthode par séparation et évaluation dédiée au problème du stable multi-objectif (BBMOISP) ont été mises en œuvre sous le langage de programmation MATLAB, en utilisant un PC Dual-Core, processeur 1.80 GHz et 1 Go de RAM. Les tests portent sur des instances générées aléatoirement avec n variables, $n \in \{20, 25, 30, 35, 40, 50, 60\}$, m contraintes (m arêtes de G), $m \in \{100, 120, 150, 250, 350, 500, 1000, 1200\}$ et p fonctions à optimiser, $p \in \{3, 7\}$. Les coefficients des fonctions objectifs sont des coefficients entiers non corrélés répartis uniformément dans l'intervalle[0,9]. Pour chaque instance (n, m, p) fixée, vingt jeux de données sont générées et l'ensemble des stables efficaces a été généré pour chaque jeu de données. La colonne ≪ MO ≫ indique le nombre moyen et la colonne ≪ MA ≫ le nombre maximum de stables efficaces trouvés pour l'instance considérée. Sous les colonnes≪ EEC ≫ et ≪ BBMOISP ≫, les colonnes ≪ MIN ≫, ≪ MAX ≫ et ≪ MOY ≫ concernent le temps CPU en secondes de chaque méthode. Notons d'abord, que les cases vides propres aux colonnes réservées à la méthode EEC indiquent que le temps CPU de cette dernière dépasse 2 heures et les calculs sont interrompus pour les instances correspondantes. D'autre part,

Instances	**MO**	**MA**	**EEC**			**BBMOISP**		
			MIN	MAX	MOY	MIN	MAX	MOY
(20,100,3)	6,9	11	2,85	6,89	5,03	0,29	1,45	0,65
(25,100,3)	6,8	15	11,5	78,7	47,7	0,71	11,6	4,09
(30,120,3)	8,5	22	163	829	453	1,91	101	38,8
(20,150,3)	6,6	11	3,82	6,96	5,29	0,33	0,72	0,58
(25,150,3)	9,6	15	22,8	65,1	42,2	1,28	7,38	2,86
(25,250,3)	7	12	29,7	62,4	43,9	1,37	2,1	1,74
(30,120,7)	64,8	100	523	1076	842	11	198	74,3
(35,150,3)	13,3	23	1188	4140	2196	69,4	795	273
(35,350,3)	9,3	15	772	1451	978	21,2	57,2	30,5
(35,500,3)	8,6	16	755	1143	869	10,35	20,7	14,9
(40,150,3)	22,7	38	-	-	-	8,81	2533	796
(40,350,3)	13,8	26	-	-	-	39,3	120	59,3
(50,250,3)	27,3	39	-	-	-	55,8	2618	989
(50,650,3)	18,5	28	-	-	-	36	65,8	47,1
(50,1000,3)	12,1	20	-	-	-	18,25	32,9	24,2
(60,1200,3)	17,6	31	-	-	-	72,1	151	100

Table 1 – Expériences numériques

les résultats obtenus montrent clairement une amélioration nette de la méthode EEC adaptée au problème MOISP. Un simple calcul permet d'affirmer que le temps CPU de la méthode BBMOISP est de l'ordre dix (10) fois plus petit, en moyenne, que celui de la méthode EEC et donc, la nouvelle méthode BBMOISP surclasse la méthode générale EEC.

4 Conclusion

Une méthode exacte, notée BBMOISP, basée sur le principe par séparation et évaluation est mise au point pour le problème du stable multi-objectif, MOISP. Elle permet la génération de tous les stables efficaces par décomposition du problème initial en sous problèmes disjoints de tailles réduites, chacun pouvant être résolu par une méthode générale relatée dans la littérature et dédiée au problème MOILP. Les résultats de l'expérimentation, même partiels, confirment l'adage "diviser pour régner" et montrent que la méthode est avantageuse par rapport à la méthode générale pour résoudre le problème MOILP décrite dans [1].
Dans le but d'accélérer la vitesse de convergence de la méthode à même de permettre le traitement d'instances de plus grandes dimensions, une investigation minutieuse à la recherche de la "meilleure" méthode pour la résolution du problème MOILP s'impose.

Références

[1] CHERGUI M.E-A., MOULAÏ M. and OUAÏL F.Z. : Solving the multiple objective integer linear pogramming problem. *Modelling, Computation and Optimization in Information Systems and Management Sciences Communications in Computer and Information Science* **vol.**(14), 69-76 (2008).

[2] Xavier Delorme, Xavier Gandibleux and Fabien Degoutin : *Evolutionary, constructive and hybrid procedures for the bi-objective set packing problem* , *European Journal of Operational Research*, **vol.** (204), **Issue** (2), 206-217 (2010).

[3] Xavier Delorme, Xavier Gandibleux and Fabien Degoutin : *GRASP for set packing problems* , *European Journal of Operational Research*, **vol.** (153), **Issue** (3), 564-580 (2004).

[4] Steuer Ralph E. *Multiple Criteria Optimization : Theory, Computation and Application* - New York : *John Wiley and Sons*, 1986.

Présentation :

RECITS est un laboratoire de recherche opérationnelle, de combinatoire, d'informatique théorique et de méthodes stochastiques agréé par l'arrêté ministériel n° 242 du 03 avril 2013. Ses domaines de compétence en recherche appliquée concerne principalement l'aide à la précise de décision par une méthodologie scientifique aboutissant à la conception de méthodes et d'algorithmes performants pour la résolution de problèmes fortement combinatoires. Les domaines d'application de nos activités de recherche sont orientés vers des entreprises et/ou des organismes publics ou privés. Nous citons à titre d'exemples : le secteur de la sécurité informatique, le secteur de la production industrielle, les hôpitaux, les banques, les compagnies d'assurance, la formation professionnelle, etc.

Le Laboratoire RECITS possède une compétence dans des domaines comme l'ordonnancement, le transport, le codage, la théorie des graphes, la combinatoire, la prévision et la prospective, les enquêtes et sondages, l'analyse des données, l'optimisation, la théorie de la décision, la gestion du risque, etc. Au-delà de ces domaines, ses compétences touchent à l'étude et à la résolution de problèmes fortement combinatoires, de par sa maîtrise des outils comme la recherche opérationnelle, les logiciels de résolution et de programmation avancée, les algorithmes évolutionnistes, etc.

Le laboratoire dispose d'équipes de recherche très actives dont la complémentarité des actions nous fournit tous les atouts pour la production de logiciels performants d'aide à la décision, utilisant des méthodes originales préalablement étudiées de manière théorique. Il est structuré en cinq équipes de recherche :

CATI Combinatoire, Arithmétique et Informatique Théorique

RO-TOP Recherche Opérationnelle pour le Transport et l'Ordonnancement en Productique

CEAPA Combinatoire Enumérative, Analytique, Probabiliste et Applications

STEP Séries Temporelles, Econométrie et Probabilités

O2M Optimisation à Objectifs Multiples

Contacts :

Web : **http://www.lrecits.usthb.dz**

Email : **lrecits@usthb.dz**

Tél :

Fax :