Revisiting subgradient techniques applied to 3d linear assignment problem

Alexandre Tusera, Dominique Fortin

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Revisiting Subgradient Techniques Applied to 3D Linear Assignment Problem

Alexandre Tusera et Dominique Fortin

N° 2266

Mai 1994

PROGRAMME 1

Architectures parallèles,

bases de données,

réseaux et systèmes distribués

ISSN 0249-6399

INRIA
ROCQUENCOURT

Revisiting Subgradient Techniques Applied to 3D Linear Assignment Problem

Alexandre Tusera* et Dominique Fortin**

Programme 1 — Architectures parallèles, bases de données, réseaux
et systèmes distribués
Projet Archi

Rapport de recherche n° 2266 — Mai 1994 — 36 pages

Abstract: A subgradient technique which relies on a local approximation of AP3 polytope is proposed. On the one hand, it improves over standard subgradient enhanced with a whole class of facets, since it requires far less facets; furthermore, the selected facets are more accurate with respect to the gap between a relaxed solution and a feasible one. On the other hand, it compares favorably to Bundle-Trust method which is based on a subgradient approximation. Apart from this method, a Monge sequence is devised on a substructure local to relaxed/feasible gap, that affords us to recover from zig-zagging in some pathological cases.

Key-words: subgradient, 3 dimensional assignment, Bundle-Trust, Monge sequence, facets

(Résumé : tsvp)

*Alexandre.Tusera@inria.fr
**Dominique.Fortin@inria.fr

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33 1) 39 63 55 11 – Télécopie : (33 1) 39 63 53 30

Réexamen des techniques de subgradient appliquées à l'affectation linéaire 3D

Résumé : Une variante de subgradient basée sur une approximation locale du polytope AP3 est proposée. D'un côté, elle améliore la technique standard augmentée d'une classe complète de facettes puisqu' elle réduit considérablement le nombre de facettes utilisées ; de plus ces dernières sont plus précises vis à vis du voisinage défini par les solutions relaxée et faisable. D'un autre côté, elle se révèle compétitive par rapport à la méthode Bundle-Trust basée, elle, sur une approximation du subgradient. En outre, une séquence de Monge locale à une structure dépendant uniquement du couple de solutions relaxée/faisable, permet de sortir du zig-zag dans un certain nombre de cas pathologiques.

Mots-clé : subgradient, affectation linéaire 3D, Bundle-Trust, séquence de Monge, facettes

Contents

1 Introduction

The 3-index Linear Assignment Problem (AP3) can be stated as 0-1 integer programming problem

$$\max_x z = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} c_{ijk}x_{ijk} \quad (1)$$

$$\sum_{j=1}^{n}\sum_{k=1}^{n} x_{ijk} = 1\ ,\ \forall i \in [1,n] \quad (2)$$

$$\sum_{i=1}^{n}\sum_{k=1}^{n} x_{ijk} = 1\ ,\ \forall j \in [1,n] \quad (3)$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ijk} = 1\ ,\ \forall k \in [1,n] \quad (4)$$

$$x_{ijk} = \begin{cases} 0 \\ 1 \end{cases} \quad (5)$$

It is known that this problem is NP-hard (see Balas and Saltzman [5]). In the past, various heuristic algorithms were proposed : the branch & bound method in Pierskala [13], primal-dual method combined with branch & bound in Hansen and Kaufman [12], the branch & bound method improved by the subgradient optimization in Burkard and Rudolf [7], Balas and Saltzman [5], Lagrangian relaxation with subgradient optimization in Frieze and Yadegar [1] and the linear relaxation improved by the branch & bound method in Qi, Balas and Gwan [14].

The branch & bound method, linear relaxation and their combinations are efficient if the size of the problem is small. But for n $\geq$ 26 they are not applicable because space and computation time are very huge, see Qi, Balas and Gwan [14]. Thus for the size n = 100, only Lagrangian relaxation with subgradient optimization seems tractable. In this paper we describe an algorithm which is based on Lagrangian relaxation where a restricted number of facets of some subclasses are incorporated.

We were guided by an upper bound of $O(n^2)$ Lagrange multipliers, associated to selected facets, together with a limited $O(n^3)$ time complexity for each iteration step; variants unable to select facets among a given class require at least $O(n^3)$ Lagrange multipliers, see Balas and Saltzman [5].

Though we are aware of extracting a good feasible solution from a Monge sequence of AP3 problem, the $O(n^9)$ time complexity forces usage of previous mentioned heuristics. However, this complexity speeds up to $O(n^3)$ whenever Monge sequence is applied to a sparse structure built around duality gap, i.e., around the topology of relaxed and feasible solution of AP3 problem.

Our computational results show that this algorithm is more efficient than the incorporation of whole subclasses of facets and than the Bundle Trust method.

The paper is organized as follows : Section 2 describes a Lagrangian relaxation of AP3 and three facet classes of the AP3 polytope. The new algorithm, named as LA-algorithm is presented in Section 3. Section 4 describes a variant of LA-algorithm, where a partial Monge sequence is computed. Section 5 describes an alternate method, the Bundle Trust method, to calculate the Lagrange multipliers. Finally, Section 6 presents the results of computational experiments.

2 Lagrangian Relaxation

Consider an integer program (IP)

$$\max_x z = cx \tag{6}$$

$$Ax \leq b \tag{7}$$

$$x \in Z_+^m \tag{8}$$

(7) can be splitted in m_1 complicated and m_2 nice constraints as

$$A^1 x \leq b^1 \tag{9}$$

$$A^2 x \leq b^2 \tag{10}$$

$$\tag{11}$$

Now for any $u \in I\!R_+^{m_1}$ consider the problem (LR)

$$\Phi = \min_u \{\max_x = cx + u(b_1 - A^1 x)\} \tag{12}$$

with (8) and (10). The problem LR is called the *Lagrangian dual* of IP. By dropping the complicated constraints set we obtain a relaxation that is easier to solve then the original problem.

It is well-known that the optimum of Φ named as $\hat{\Phi}$ is equal to the optimum of original problem; so one tries to find $\hat{\Phi}$ via Lagrange multipliers.

Techniques for solving the Lagrangian relaxation are known as :

- standard subgradient optimization used in Held, Wolfe and Crowder [11], Frieze and Yadegar [1]
- Bundle Trust method used in Schramm and Zowe [9], [10]

The drawbacks of both techniques are the lack of guarantee to reach the optimum and the tendency of subgradient to converge to local optimum, so-called as "zig-zagging".

2.1 Lagrangian Relaxation of AP3

Among the "natural" relaxation of AP3 (quoted in Balas and Saltzman [5]), we choose to relax the ground set K into the objective function

$$\Phi = \min_{u} \ \{ \ \max_{x} \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}(c_{ijk} - u_k)x_{ijk} + \sum_{k=1}^{n} u_k \ \} \tag{13}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{n} x_{ijk} = 1 \ , \ \forall j \in [1,n] \tag{14}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ijk} = 1 \ , \ \forall k \in [1,n] \tag{15}$$

An optimal solution to this relaxation is achieved through solving an assignment problem over the ground sets I and J, see Frieze and Yadegar [1], which only requires $O(n^3)$ polynomial time.

2.2 The Facets of AP3 Polytope

First let us recall standard notation associated with AP3 problem.

- The matrix A_n of constraints of AP3 problem has very regular structure shown for the case $n = 3$ in Fig. 1.

  The row and column index set of A_n will be denoted by R_n and S_n respectively. From regularity of A_n follow : $|R_n| = |I_n| + |J_n| + |K_n| = 3n$

```
[ 111111111                 ]
[         111111111         ]
[                 111111111 ]
[ 111     111     111       ]
[   111     111     111     ]
[     111     111     111   ]
[ 1  1  1  1  1  1  1  1  1   ]
[  1  1  1  1  1  1  1  1  1  ]
[   1  1  1  1  1  1  1  1  1 ]
```

Figure 1: The matrix A of constraints of AP3 for $n = 3$

and $|S_n| = |I_n| \times |J_n| \times |K_n| = n^3$. For a set $Q \subseteq R_n$, we use Q_I, Q_J and Q_K to denote the part of Q in I, J and K respectively.

We note a single column of A_n as $a^s \in S_n$ where $s = (i, j, k)$ means that a^s has 1's in position $i \in I, j \in J, k \in K$ and $a^s \cap Q$ refers to the number of 1's of a^s in Q.

For $x \in I\!R^{n^3}$ and $S' \subseteq S$, let us denote :

$$x(S') = \sum (x_{a^s} : a^s \in S')$$

- The *intersection graph* $G_{A_n} = (V, E)$ of the matrix A_n has a node s for every column a^s of A_n and an edge (s, t) for every pair of columns a^s and a^t such that $a^s \cdot a^t \neq 0$. The intersection graph G_{A_n} for $n = 2$ is shown in Fig. 2.

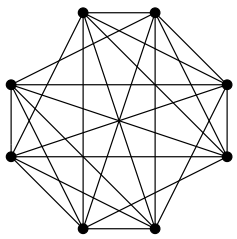

Figure 2: The intersection graph G_{A_n} for $n = 2$

- We note P_I, the convex hull of feasible solutions to AP3 problem :

$$P_I = \text{conv}\{x \in \{0,1\}^{n^3} : A_n x = e\}$$

where e is the 1's vector. Currently the following three facet-defining inequalities classes for AP3 polytope are known :

2.2.1 Facets Associated with the Odd Holes

Let $Q \subseteq R_n$, $|Q| = 2p+1$, $1 \leq p \leq n-1$ and $1 \leq |Q_L| \leq p$ for $L = I, J, K$. Let

$$S(Q) = \{a^s \in S_n : |a^s \cap Q| \geq 2\}$$

Then

$$x(S(Q)) \leq p \tag{16}$$

defines a facet of P_I for $n \geq 3$.

Facets (16) are known as facets associated with the odd holes of G_{A_n} since they can be looked as lifted from odd holes inequalities. The value p corresponds to the length of odd hole H where $|H| = 2p+1$, see Balas and Saltzman [4].

- In the special case $p = 1$ the facets (16) are reduced to (3n + 2)-node cliques of G_{A_n} (noted as c2 for short). They are defined in Balas and Saltzman [4] as the node set $\{a^s\} \cup T(a^s)$ where

  $$T(a^s) = \{a^t \in S_n \backslash \{a^s\} \ : \ a^s \cdot a^t = 2\} \tag{17}$$

  Every class-2 clique is identified by its centre-node $a^s = (i, j, k)$. One example of class-2 clique for $n = 5$ is in Fig. 3.

- In the special case $p = 2$ the facets (16) can be seen as union of four node sets (17). Every facet (noted as oh2 for short) is identified by 4-tuple of nodes $\{a^{s_1}, a^{s_2}, a^{s_3}, a^{s_4}\}$. One example of this facet for $n = 5$ is in Fig. 4.

2.2.2 Comb Facets

Let $D, Q \subseteq R_n$, $D \cap Q = \emptyset$, $|D_L| + |Q_L| = p+1$ and $1 \leq |Q_L| \leq r$ for $L = I, J, K$, $|Q| = 2r+1$, $1 \leq r \leq p \leq n-3$. Let

$$C_1(D) = \{a^s \in S_n : a^s \subseteq D\},$$

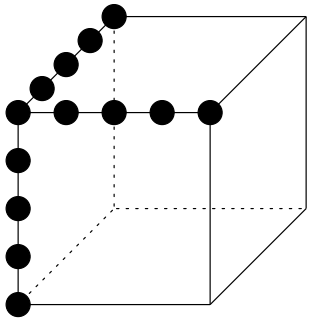

Figure 3: Class-2 clique for $n = 5$

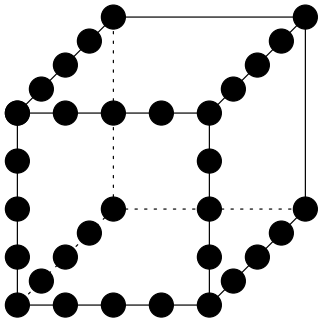

Figure 4: Facet associated with the odd hole for $p = 2$ and $n = 5$

$$C_2(D,Q) = \{a^s \in S_n : |a^s \cap D| = 1, |a^s \cap Q| = 2\},$$
$$C(D,Q) = C_1(D) \cup C_2(D,Q)$$

Then

$$x(C(D,Q)) \leq p \tag{18}$$

defines a facet of P_I for $n \geq 4$. We call such a facet, a *comb facet* (See Gwan and Qi [8]).

In the special case $r = 1$ and $p = 1$ the comb facets are 4-node cliques of G_{A_n}, known as class-3 cliques (noted as c3 for short). Every facet c3 is identified by 4-tuple of nodes $\{a^{s_1}, a^{s_2}, a^{s_3}, a^{s_4}\}$. One example of class-3 clique for $n = 5$ is in Fig. 5.

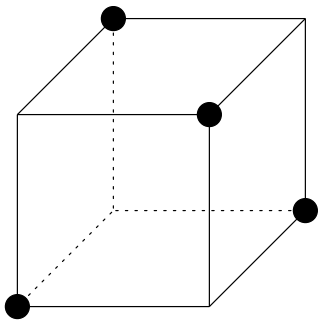

Figure 5: Class-3 clique for $n = 5$

2.2.3 Bull Facets

Let $D \subset R_n$ have the same cardinalities in I, J and K, i.e., $|D_I| = |D_J| = |D_K| = r$, where $1 \leq r \leq n-4$. Let $H \subseteq R_n$, $H \cap D = \emptyset$, $|H| = 2$, $|H_L| \leq 1$ for $L = I, J, K$. Let $Q = D \cup H$. Let

$$B(D) = \{a^s \in S_n : a^s \subseteq D\},$$

$$F(Q, D) = \{a^s \in S_n : |a^s \cap Q| \geq 2, 2 \geq |a^s \cap D| \geq 1\}$$

Then

$$2x(B(D)) + x(F(Q, D)) \leq 2r \tag{19}$$

defines a facet of P_I for $n \geq r + 4$. We call such a facet, a *bull facet* (See Gwan and Qi [8]). One example of this facet for $r = 1$ and $n = 5$ is shown in Fig. 6.

2.3 The Use of Facets

- linear relaxation :
  For an arbitrary noninteger solution of the linear relaxation, it is always possible to identify a facet inequality that cuts off the current fractional solution and add it to the constraints set; however, the price to pay can be quite heavy,

- Lagrangian relaxation :
  In this case, we do not have a feasible linear relaxation solution at hand,

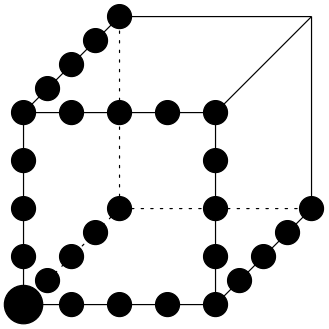

Figure 6: Bull facet for $r = 1$ and $n = 5$

so we cannot identify violated facet inequalities directly. But we can add the facet inequalities into the objective function Φ along with Lagrange multipliers. From (13) we get

$$\Phi = \min_{u} \{ \max_{x} \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}(c_{ijk} - u_k)x_{ijk} + \sum_{k=1}^{n} u_k + \sum_{c\in C} \lambda_c(1 - \sum_{t\in S(c)} x_t)\}$$

where $S(c)$ stands for the node set of one facet c from a given class of facets C. The solution $\hat{\Phi}$ violates some facet inequalities which may be added to the objective function, in turn.

In general, for a NP-hard problem the number of facets and the direct identification of violated facets is also NP-hard ; which is known as the separation problem in Balas and Qi [6]. Fortunately, it happens that for some subsets of facets, those facets which are violated are easy to enumerate.

3 LA Algorithm

In the linear relaxation approach, the more facet inequalities are introduced, the better (or not worse) result. So, it is natural to try to introduce a whole class of facets in Lagrangian relaxation.

Let us incorporate subclass c2 of facet (16) into the function Φ, for instance. Despite of the fact that this subclass has the smallest cardinality among all subclasses of facets of AP3 polytope (n^3), the size of required memory grows

considerably (n^3 new Lagrange multipliers are introduced). Surprisingly the result (in average) is not good; even, in several cases, it is worse than the method without any facets.

From this result, we subsume that exists a restricted number of facets (of any subclass) which is "reasonable and efficient", i.e., which leads to a better result than the method with a whole subclass as well as the methods without any facets.

3.1 Local Facet Trust Region

In this approach we consider the facets of some subset which are violated by $\hat{\Phi}$ and simultaneously satisfied by the feasible solution (named as $\hat{\phi}$).

We prefer these notations $\hat{\Phi}$ and $\hat{\phi}$ for relaxed and feasible solutions respectively, over upper and lower bounds since they do not rely on min or max versions of objective function.

We note this subset of facets as LA (local approximation) since these facets approximate locally the AP3 polytope (locally with respect to the $\hat{\Phi}$ and $\hat{\phi}$).

Hereafter, we give the upper bound of cardinality of LA for some subclasses of facets :

subclass of facets	number of facets in whole subclass	max. number of facets in the subset LA
$c2$	n^3	$2(n-1)$
$c3$	$n^3(n-1)^3/4$	$(n-1)^2$
$oh2$	$3n^3(n-1)^2/4$	$6(n-1)(n-2)$

We can see that the number of such facets decreases drastically with respect to the number of facets in a whole subclass.

3.2 Determination of Upper Bound of Cardinality of LA Set

First we introduce the following notation :

- We note $\hat{\Phi}_d^K$ as the subset of nodes of $\hat{\Phi}$ which share common index K_d of the ground set K. Thus, we can write

$$\hat{\Phi}_d^K = \{(I_1, J_1, K_d), (I_2, J_2, K_d), \cdots, (I_s, J_s, K_d)\} \tag{20}$$

  $\hat{\Phi}_d^K$ can be seen as the node set which violates one trivial facet of AP3 corresponding to the index $K_d \in K$. Decomposition of $\hat{\Phi}$ along all the r violated trivial facets, in relaxed ground set K, leads to :

$$\hat{\Phi} = \bigcup_{d=1}^{r} \hat{\Phi}_d^K \tag{21}$$

- There is one to one correspondence between the nodes of $\hat{\Phi}$ and $\hat{\phi}$. We note the corresponding subset $\hat{\phi}_d^K \subseteq \hat{\phi}$ to $\hat{\Phi}_d^K$ as

$$\hat{\phi}_d^K = \{(I_1, J_1, K_1), (I_2, J_2, K_2), \cdots, (I_s, J_s, K_s)\} \tag{22}$$

  Whenever exists LA set, then exists one node $(I_i, J_i, K_i) \in \hat{\phi}_d^K$ such that $K_i = K_d$.

3.2.1 Class-2 cliques c2

We define a subset of class-2 cliques

$$LA_{c2_d} = \{(I_t, J_u, K_d) \quad : \quad \forall t \in [1, s] \backslash i, \quad \forall u \in [1, s] \backslash i, \quad t \neq u\}$$

We can see that every class-2 clique of LA_{c2_d} contains two nodes of $\hat{\Phi}_d^K$ and one node of $\hat{\phi}_d^K$, thus the conditions of LA are satisfied. The cardinality of LA_{c2_d} is

$$|LA_{c2_d}| = 2(|\hat{\Phi}_d^K| - 1) = 2(s - 1)$$

From (21), we derive :

$$|LA_{c2}| = \sum_{d=1}^{r} 2(|\hat{\Phi}_d^K| - 1)$$

$|LA_{c2}|$ is maximal for $r = 1$ and for $|\hat{\Phi}_1^K| = n$. Hence the upper bound

$$2(n - 1)$$

3.2.2 Class-3 cliques c3

We define a subset of 4-nodes of class-3 cliques

$$LA_{c3_d} = \{((I_i, J_i, K_d), (I_t, J_t, K_d), (I_i, J_t, v), (I_t, J_i, v)) \ : \\ \forall t \in [1, s] \backslash i, \ \ \forall v \in [1, n] \backslash K_d\}$$

We can see that every class-3 clique of LA_{c3_d} contains two nodes of $\hat{\Phi}_d^K$ and one node of $\hat{\phi}_d^K$, thus the conditions of LA are satisfied. The cardinality of LA_{c3_d} is

$$|LA_{c3_d}| = (n-1)(|\hat{\Phi}_d^K| - 1) = (n-1)(s-1)$$

From (21), we derive :

$$|LA_{c3}| = (n-1) \sum_{d=1}^{r} (|\hat{\Phi}_d^K| - 1)$$

$|LA_{c3}|$ is maximal for $r = 1$ and for $|\hat{\Phi}_1^K| = n$. Hence the upper bound

$$(n-1)^2$$

3.2.3 Odd holes p = 2

For every triplet

$$\{(I_t, J_t, K_t), (I_u, J_u, K_u), (I_i, J_i, K_i)\}$$

such that $t, u \in [1, s] \backslash i$, $t \neq u$ and $v \in \{t, u\}$ we define a set LA_{oh2_d} of facets oh2 :

$$(I_i, J_t, K_d), (I_i, J_u, K_d), (I_i, J_t, K_v), (I_i, J_u, K_v)$$
$$(I_i, J_u, K_d), (I_t, J_u, K_d), (I_i, J_u, K_v), (I_t, J_u, K_v)$$
$$(I_i, J_t, K_d), (I_u, J_t, K_d), (I_i, J_t, K_v), (I_u, J_t, K_v)$$
$$(I_t, J_i, K_d), (I_u, J_i, K_d), (I_t, J_i, K_v), (I_u, J_i, K_v)$$
$$(I_t, J_i, K_d), (I_t, J_u, K_d), (I_t, J_i, K_v), (I_t, J_u, K_v)$$
$$(I_u, J_i, K_d), (I_u, J_t, K_d), (I_u, J_i, K_v), (I_u, J_t, K_v)$$

We can see that every facet of LA_{oh2_d} contains three nodes of $\hat{\Phi}_d^K$ and two nodes of $\hat{\phi}_d^K$, thus the conditions of LA are satisfied. The cardinality of LA_{oh2_d} is

$$|LA_{oh2_d}| = 12 \begin{pmatrix} |\hat{\Phi}_d^K|-1 \\ 2 \end{pmatrix} = 6(|\hat{\Phi}_d^K| - 1)(|\hat{\Phi}_d^K| - 2)$$

Using (21) we get

$$|LA_{oh2}| = 6 \sum_{d=1}^{r} (|\hat{\Phi}_d^K| - 1)(|\hat{\Phi}_d^K| - 2)$$

$|LA_{oh2}|$ is maximal for $r = 1$ and for $|\hat{\Phi}_1^K| = n$. Hence the upper bound

$$6(n-1)(n-2)$$

3.3 LA Algorithm

The main idea of the algorithm which incorporates the LA facets (named as LA-algorithm) is :

- to find the LA set and incorporate it into Φ function,
- to keep LA set in next iterations while at least one facet is violated,
- to update the LA set whenever all facets are satisfied and restart with new Lagrange multipliers associated with new LA set.

The outline of LA-algorithm in pseudocode form follows :

iteration i :
calculate $\hat{\Phi}$ with incorporated LA
calculate $\hat{\phi}$
if (LA = $\emptyset$ or no facet of LA is violated)
then
    LA := find_LA(G_{A_n},$\hat{\Phi}$,$\hat{\phi}$,facet_class)
    remove Lagrange multipliers of old LA from Φ
    incorporate Lagrange multipliers of new LA into Φ
fi
update Lagrange multipliers of Φ

The LA-algorithm takes care only of local facets with respect to $\hat{\Phi}$ and $\hat{\phi}$. The purpose of this approach is to prevent "zig-zagging" of subgradient. Another approach which try to prevent this "zig-zag" behavior is to use the Bundle Trust method where the subgradient is replaced by an approximation. Our experiments show that LA-algorithm is more efficient than Bundle Trust method.

However these two methods may be seen as extreme, with respect to subgradient applied to a given polytope, since one approximates the subgradient of the actual polytope and the other one approximates the polytope while keeping the true subgradient calculation.

An outline of the procedures find_LA in pseudocode for AP3 problem for three sets LA_{c2}, LA_{c3} and LA_{oh2} follows.

We will denote :

- one clique of G_{A_n} by CL,
- the cardinality of CL by $|CL|$,
- the number of common nodes CL and $\hat{\Phi}$ by $|\text{CL} \cap \hat{\Phi}|$,
- one odd hole oh2 of G_{A_n} by OH2 and
- the facet associated with the odd hole OH2 by FACET_OH2.

procedure find_LA(G_{A_n},$\hat{\Phi}$,$\hat{\phi}$,c2)
forall CL in G_{A_n} do
    CL := find_clique(G_{A_n})
    if $|CL| = 7$ and $|\text{CL} \cap \hat{\Phi}| = 2$ and $|\text{CL} \cap \hat{\phi}| = 1$ then
      put CL into the list LA_{c2}
    fi
od

procedure find_LA(G_{A_n},$\hat{\Phi}$,$\hat{\phi}$,c3)
forall CL in G_{A_n} do
    CL := find_clique(G_{A_n})
    if $|CL| = 4$ and $|\text{CL} \cap \hat{\Phi}| = 2$ and $|\text{CL} \cap \hat{\phi}| = 1$ then

```
        put CL into the list LA_c3
    fi
od
```

procedure find_LA(G_{A_n},$\hat{\Phi}$,$\hat{\phi}$,oh2)
forall OH2 in G_{A_n} do
    OH2 := find_oh2(G_{A_n})
    FACET_OH2 := lift_oh2(OH2)
    sieve(LA_{oh2},FACET_OH2)
od

The procedure lift_oh2 generates the facet associated with the odd hole OH2, i.e., all nodes of G_{A_n} which share an edge with at least two nodes of OH2, are found. The procedure sieve checks whether facet associated with the odd hole OH2 is already in the list LA_{oh2} since several odd holes may lead to the same facet.

These generic procedures are quite inappropriate since they require the knowledge of every facet in a given class. However, it is only a matter of skill to compile them, to achieve both space and time complexities mentioned before.

The aim of outlines of these generic procedures is only to show the main idea and the principle of the LA algorithm. The optimal LA algorithm is based on the regular structure of G_{A_n}.

Since G_{A_n} of AP3 problem is regular, we can give an outline of the optimal procedure find_LA in pseudocode form for AP3 problem for the sets LA_{c2}, LA_{c3} and LA_{oh2}.

We need the array RLX[n $\times$ 3] which contains all nodes of the relaxed solution $\hat{\Phi}$, hence

$$a^s = (i, j, k) = (RLX[p, 1], RLX[p, 2], RLX[p, 3]) \quad \forall p \in [1, n]$$

We suppose that the nodes of $\hat{\Phi}$ are sorted by K ground set index in increasing order, i.e.,

$$RLX[p_1, 3] \leq RLX[p_2, 3], \quad p_1 \leq p_2 \quad \text{and} \quad \forall p_1, p_2 \in [1, n]$$

The array FSB[n] contains the K ground set index of nodes of feasible solution $\hat{\phi}$ so the node (RLX[p,1], RLX[p,2], RLX[p,3]) corresponds to FSB[p] $\forall p \in [1, n]$.

We need as well the array C[n] which contains the cardinalities $|\hat{\Phi}_k|$ for every index $k \in K$.

```
procedure find_LA(RLX,FSB,C,c2)
pt_rlx := 0
for k to n do
    if C[k] = 0 then next fi
    if C[k] = 1 then pt_rlx := pt_rlx+1 next fi
    k_fsb := 0
    test if LA exists
    for k1 to C[k] do
        if RLX[pt_rlx+k1,3] = FSB[pt_rlx+k1] then
          i_fsb := RLX[pt_rlx+k1,1]
          j_fsb := RLX[pt_rlx+k1,2]
          k_fsb := RLX[pt_rlx+k1,3]
          ind_pt_rlx := k1
          break
        fi
    od
    if k_fsb > 0 then
      for k1 to C[k] do
          if k1 = ind_pt_rlx then next fi
          i_rlx1 := RLX[pt_rlx+k1,1]
          j_rlx1 := RLX[pt_rlx+k1,2]
          couple of class-2 cliques of LA_c2
          (i_rlx1,j_fsb,k_fsb)
          (i_fsb,j_rlx1,k_fsb)
      od
    fi
    pt_rlx := pt_rlx+C[k]
od
```

```
procedure find_LA(RLX,FSB,C,c3)
pt_rlx := 0
for k to n do
    if C[k] = 0 then next fi
    if C[k] = 1 then pt_rlx := pt_rlx+1 next fi
    k_fsb := 0
    test if LA exists
    for k1 to C[k] do
        if RLX[pt_rlx+k1,3] = FSB[pt_rlx+k1] then
          i_fsb := RLX[pt_rlx+k1,1]
          j_fsb := RLX[pt_rlx+k1,2]
          k_fsb := RLX[pt_rlx+k1,3]
          ind_pt_rlx := k1
          break
        fi
    od
    if k_fsb > 0 then
      for k1 to C[k] do
          if k1 = ind_pt_rlx then next fi
          i_rlx1 := RLX[pt_rlx+k1,1]
          j_rlx1 := RLX[pt_rlx+k1,2]
          for k2 to n do
              if k2 = k_fsb then next fi
              class-3 cliques of LA_c3
              (i_fsb,j_fsb,k_fsb)
              (i_rlx1,j_rlx1,k_fsb)
              (i_fsb,j_rlx1,k2)
              (i_rlx1,j_fsb,k2)
          od
      od
    fi
    pt_rlx := pt_rlx+C[k]
od

procedure find_LA(RLX,FSB,C,oh2)
```

```
pt_rlx := 0
for k to n do
    if C[k] = 0 then next fi
    if C[k] = 1 then pt_rlx := pt_rlx+1 next fi
    if C[k] = 2 then pt_rlx := pt_rlx+2 next fi
    k_fsb := 0
    test if LA exists
    for k1 to C[k] do
        if RLX[pt_rlx+k1,3] = FSB[pt_rlx+k1] then
          i_fsb := RLX[pt_rlx+k1,1]
          j_fsb := RLX[pt_rlx+k1,2]
          k_fsb := RLX[pt_rlx+k1,3]
          ind_pt_rlx := k1
          break
        fi
    od
    if k_fsb > 0 then
      for k1 to C[k] do
          if k1 = ind_pt_rlx then next fi
          i_rlx1 := RLX[pt_rlx+k1,1]
          j_rlx1 := RLX[pt_rlx+k1,2]
          for k2 from k1+1 to C[k] do
              if k2 = ind_pt_rlx then next fi
              i_rlx2 := RLX[pt_rlx+k2,1]
              j_rlx2 := RLX[pt_rlx+k2,2]
              k_l1 := FSB[pt_rlx+k1]
              k_l2 := FSB[pt_rlx+k2]
              set_i := [i_fsb,i_rlx1,i_rlx2]
              set_j := [j_fsb,j_rlx1,j_rlx2]
              set_k := [k_l1,k_l2]
              for i1 to 3 do
                  for i2 from i1 to 3 do
                      for j1 to 3 do
                          if j1 = i1 or j1 = i2 then next fi
                          for j2 from j1 to 3 do
```

```
                                    if j2 = i1 or j2 = i2 or (i1 = i2 and j1 = j2) then next fi
                                    for k3 to 2 do
                                        one facet of LA_oh2
                                        (set_i[i1],set_j[j1],k_fsb)
                                        (set_i[i2],set_j[j2],k_fsb)
                                        (set_i[i1],set_j[j1],set_k[k3])
                                        (set_i[i2],set_j[j2],set_k[k3])
                                    od
                                od
                            od
                        od
                    od
                od
            od
        fi
        pt_rlx := pt_rlx+C[k]
od
```

4 Skeleton

LA-algorithm described in previous section, incorporates the facets without respect to values of coefficients c_{ijk} of AP3 problem. But it is possible to take into account c_{ijk} values as follows :

1. define a restricted set RS of nodes of G_{A_n} which is local to $\hat{\Phi}$ and $\hat{\phi}$,
2. try to find $\hat{\phi}'$ in RS which is better than $\hat{\phi}$,
3. find the set LA for $\hat{\Phi}$ and $\hat{\phi}'$.

Consider the node sets in Fig. 7 a) and b).

If $R_i \in \hat{\Phi}$ (relaxed) and $F_i \in \hat{\phi}$ (feasible) for $i \in [1,4]$ then the node sets $S_1 = \{A_1, A_2, B_1, B_2\}$ and $S_2 = \{A_3, A_4, B_3, B_4\}$ are local with regard to $\{R_1, R_2, F_1, F_2\}$ and $\{R_3, R_4, F_3, F_4\}$ respectively. We can define the RS set as the union of all sets S_1 and S_2 in current $\hat{\Phi}$. The RS set for n = 4 is shown

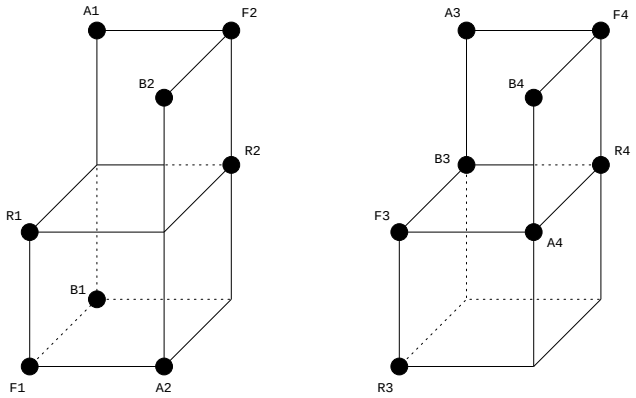

Figure 7: Skeleton node sets

in Fig. 8. The nodes of the feasible solution are colored as white, they can be seen as a backbone. All others points are colored as black, they can be seen as coasts. It is possible that in RS exists $\hat{\phi}'$ which is better than $\hat{\phi}$.

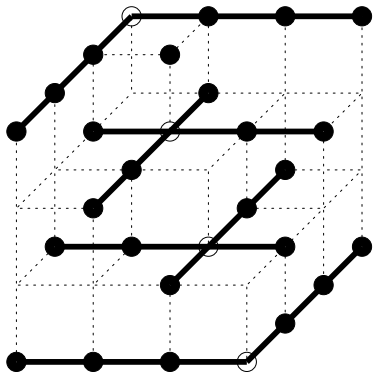

Figure 8: Skeleton node set RS for n = 4

One possibility to retrieve $\hat{\phi}'$ from RS, is to use Monge sequence as in Rudolf [15]. Since the structure of RS is sparse, the number of inequalities involved in Monge sequence decreases to $O(n^2)$ and the whole complexity from $O(n^9)$ to $O(n^3)$. The assignment of elements of Monge sequence leads to the greedy solution $\hat{\phi}'$.

The outline of LA-algorithm with the skeleton improvement in pseudocode form follows :

iteration i :
calculate $\hat{\Phi}$ with incorporated LA
calculate $\hat{\phi}$
if (LA = $\emptyset$ or no facet of LA is violated)
then
    LA := find_LA_skeleton(G_{A_n},$\hat{\Phi}$,$\hat{\phi}$,facet_class)
    remove Lagrange multipliers of old LA from Φ
    incorporate Lagrange multipliers of new LA into Φ
fi
update Lagrange multipliers of Φ

The outline of the procedure find_LA_skeleton follows, where MS stands for the Monge sequence on RS, the procedure find_RS returns the RS set of $\hat{\Phi}$ and $\hat{\phi}$, the procedure find_monge_sequence finds the Monge sequence on RS and the procedure find_greedy extracts the greedy solution from MS.

This improvement intends to balance the brute force relaxation of the whole ground set K by reintroducting the third dimension as a local skeleton built from a local backbone (feasible solution) along with coasts (relaxed solution).

procedure find_LA_skeleton(G_{A_n},$\hat{\Phi}$,$\hat{\phi}$,facet_class)
RS := find_RS($\hat{\Phi}$,$\hat{\phi}$)
MS := find_monge_sequence(RS)
$\hat{\phi}'$:= find_greedy(MS)
if $\hat{\phi}'$ is better than $\hat{\phi}$
then
    $\hat{\phi}$:= $\hat{\phi}'$
fi
LA := find_LA(G_{A_n},$\hat{\Phi}$,$\hat{\phi}$,facet_class)

5 Bundle Trust Method

Another approach to compute the new Lagrange multipliers is to minimize Lagrangian function via *Bundle Trust Method* :

Let f be a convex (not necessarily differentiable) function from $\mathbb{R}^n$ into $\mathbb{R}$ and let us know $f(x)$ and one (arbitrary) subgradient $g(x) \in \partial f(x)$ at every x.

Two closely related methods have been proposed during recent years how to minimize such f (Schramm and Zowe [9], [10]) ; an *ϵ-descent-like method (L)* and *modified cutting plane idea (M)*, both are closely related : they are dual to each other.

If the gradient $\nabla f(x)$ of the convex f exists almost everywhere and, where $\nabla f(x)$ does not exist, we substitute them by subdifferential $\partial f(x)$ whose elements are characterized by an inequality

$$g \in \partial f(x) \;\Leftrightarrow\; (g, y - x) \leq f(y) - f(x) \;\; \forall y \in \mathbb{R}^n \tag{23}$$

where $(g, y - x)$ denotes the dot product of g and $y - x$. It is well-known that $\partial f(x)$ is a nonempty convex compact set which shrinks to $\nabla f(x)$ whenever gradient exists. Let $proj(0|\partial f(x_k))$ denote the unique element in $\partial f(x)$ which is nearest to the origin in the Euclidean norm $\|.\|$, thus we are led to the *iteration scheme* :

1. Compute $d_k := -proj(0|\partial f(x_k))$
2. Compute s_k with $f(x_k + s_k d_k) = \max_{s \geq 0} f(x_k + s_k d_k)$
3. Put $x_{k+1} := x_k + s_k d_k$

This scheme can provide a sequence $\{x_k\}$ which converges to a nonstationary (local optimum) point $\bar{x}$. This is caused by the loss of information in $\partial f(x)$ when the x_k approach some $\bar{x}$ where f is not differentiable.

The solution consists in replacing the subdifferential in the iteration scheme by the larger *ϵ-subdifferential* $\partial_\epsilon f(x)$ whose elements are characterized by

$$g \in \partial_\epsilon f(x) \;\Leftrightarrow\; (g, y - x) \leq f(y) - f(x) + \epsilon \;\; \forall y \in \mathbb{R}^n,\; \epsilon \geq 0 \tag{24}$$

This set collects the subgradient informations from neighbourhood of x, thus when x_k is close to $\bar{x}$, then set $\partial_\epsilon f(x)$ will detect the nondifferentiability at $\bar{x}$ and one can avoid convergence to a nonoptimal $\bar{x}$.

The main idea of (L) and (M) methods is the *Bundle concept* : At the iteration x_k we have at our disposal the sequence $x_1, x_2, \cdots x_k$ and a collection of auxiliary points y_i together with subgradients $g_i \in \partial f(y_i)$ for $i \in [1, k]$.

We restrict ourselves to the variant (M). This method uses the informations contained in the bundle to build up a *cutting plane approximation model* of f in x_k

$$\max_{1 \leq i \leq k} \{f(y_i) + (g_i, x - y_i)\} \tag{25}$$

to determine a direction of descent $d \in I\!R^n$ for f. This means to minimize term

$$\min\{f(x_k + d) - f(x_k)\} \tag{26}$$

Using model (25) this term is approximated by

$$\max_{1 \leq i \leq k} \{f(y_i) + (g_i, x_k + d - y_i)\} - f(x_k) \tag{27}$$

With the linearization errors

$$\alpha_i^k = f(x_k) - (f(y_i) + (g_i, x_k - y_i)) \tag{28}$$

we can write this in a form

$$\max_{1 \leq i \leq k} \{(g_i, d) - \alpha_i^k\} - f(x_k) \tag{29}$$

Thanks to convexity, all α_i^k are nonnegative (a consequence of (23)). For convenience, let us skip the constant $f(x_k)$ and put

$$\hat{f}(x_k, d) = \max_{1 \leq i \leq k} \{(g_i, d) - \alpha_i^k\} \quad \text{for} \quad d \in I\!R^n \tag{30}$$

Minimization of $\hat{f}(x_k, d)$ may not have a solution at all or one which may lead to some d so large in norm that $\hat{f}(x_k, d)$ cannot be considered any more as an approximation of $f(x_k + d) - f(x_k)$.

To prevent these two defects one adds to (30) a term

$$\frac{1}{2t_k}\|d\|^2$$

with some fixed $t_k > 0$. This leads to the model

$$m(t_k; g_k, \cdots, g_1; x_k; d) := \hat{f}(x_k, d) + \frac{1}{2t_k}\|d\|^2 \quad \text{for} \quad d \in I\!R^n \tag{31}$$

Minimization of (31) yields a direction d. For small t_k a direction d will be small in norm. This ensures that we stay in a region where we can still trust the model m. (in some sense t_k is "dual" to ϵ introduced in (24)).

The solution of (31) is most easily computed via the quadratic programming model in $(v, d) \in I\!R^{n+1}$

$$\min\{v + \frac{1}{2t_k}\|d\|^2 \;\mid\; v \geq (g_i, d) - \alpha_i^k \quad \text{for} \quad i \in [1, k]\} \tag{32}$$

The algorithm for one iteration $x_k \to x_{k+1}$ follows :

Iteration $x_k \to x_{k+1}$:
Chosse $t_k^1 \in [t_1, t_2]$
Compute α_i^k for $i = 1 \cdots k$
$j := 1$
do
  Compute the solution v^j, d^j of (32)
  $y_{k+1}^j := x_k + d_k^j$
  Compute $g_{k+1}^j \in \partial f(y_{k+1}^j)$
  if $f(y_{k+1}^j) - f(x_k) < m_1 v_k^j$ then
    if $(g_{k+1}^j, d_k^j) \geq m_2 v_k^j$ or $t_k^j > t_2$ then
      Serious Step
      $x_{k+1} := y_{k+1}^j$
      $y_{k+1} := y_{k+1}^j$
      $g_{k+1} := g_{k+1}^j$
      Break
    else

$t_k^j := 2t_k^j$
$t_1 := t_k^j$
fi
else
if $f(d_k^j) + (g_{k+1}^j, d_k^j) \leq \beta$ or $t_k^j < t_1$ then
Null Step
$x_{k+1} := x_k$
$y_{k+1} := y_{k+1}^j$
$g_{k+1} := g_{k+1}^j$
Break
else
$t_k^j := \frac{1}{2} t_k^j$
$t_2 := t_k^j$
fi
j := j +1
od

For fixed $g_k, \cdots, g_1, x_k$ the trajectory $d_k := d(t_k)$ is continuous. Algorithm will terminate with $t_k := t_k^*$. This t_k^* is not necessary a global optimum of trajectory $d(t_k)$ but this is the first t_k found which fulfills the conditions of *Serious Step* or *Null Step*. In next subsections we detail branching and setting in the algorithm.

5.1 Initialization

Set $x_1 := 0$, $0 < m_1 < m_2 < 1$ and $\beta > 0$.

5.2 Serious Step

Condition $f(y_{k+1}^j) - f(x_k) < m_1 v_k^j$ ensures a decrease of at least m_1 times $v_k = \hat{f}(x_k, d)$ and since this is a main goal of algorithm we hope that the good new Lagrange multipliers x_k are found.

A good additive condition for convergence of the method is to obtain in next step some d_{k+1} which differs from the current d_k, thus we want

$$(g_{k+1}, d_k^j) - \alpha_{k+1}^{k+1} > (g_i, d_k^j) - \alpha_i^k \quad \text{for} \quad i = 1, \cdots, k$$

This means that the new constraint in (32) will be active. Using rules for Serious Step, we put $x_{k+1} = y_{k+1}$ in next step. This leads to $\alpha_{k+1}^{k+1} = 0$. The new constraint in (32) is

$$(g_{k+1}, d_k) - \alpha_{k+1}^{k+1} = (g_{k+1}, d_k)$$

Thus we have as desired

$$(g_{k+1}, d_k) \geq m_2 v_k > v_k > (g_i, d_k^j) - \alpha_i^k \text{ for } i = 1, \cdots, k$$

since $v_k < 0$ and $m_2 < 1$.

The purpose of the test $t_k^j > t_2$ is to prevent the t_k^j from becoming too large since then $\hat{f}(x_k, d)$ cannot be considered any more as an approximation of $f(x_k + d) - f(x_k)$.

5.3 Null Step

In this case the model m is bad, so we cannot achieve a decrease of f. We set $x_{k+1} := x_k$ and $y_{k+1} := x_k + d_k$.

Thus

$$\alpha_{k+1}^{k+1} = f(x_{k+1}) - f(x_{k+1} + d_k) - (g_{k+1}, x_{k+1} - x_{k+1} - d_k) = f(d_k) + (g_{k+1}, d_k)$$

If β is small, then

$$f(d_k) + (g_{k+1}, d_k) \leq \beta$$

ensures a small α_{k+1}^{k+1}. We conclude that g_{k+1} is "close" to $\partial f(x_{k+1})$ and thus it makes sense to add g_{k+1} to the bundle at x_k. (α_k^i "measures" how much $g_i \in \partial f(x_k)$ satisfies the subgradient inequality at the point x_k).

The test $t_k^j < t_1$ allows to prevent the t_k^j from becoming too small (we were too optimistic with respect to t_k^j, i.e., we cannot achieve a decrease of f despite of small t_k^j, so model m is bad).

If neither a Serious Step nor a Null Step is possible, then we improve the model by better t_k^j in next inner iteration.

5.4 Choice of Constants m_1, m_2

If algorithm does not terminate neither with Serious Step nor with Null Step this implies $t_1 \overset{\text{below}}{\longrightarrow} t_k^*$ and $t_2 \overset{\text{above}}{\longrightarrow} t_k^*$ for some $t_k^* \in [t_1, t_2]$, the interval $[t_1, t_2]$ will be very small. Since trajectory of $d(t)$ is continuous, for $d^* := d(t_k^*)$ and $v^* := v(t_k^*)$ together with $f(y_{k+1}^j) - f(x_k) < m_1 v_k^j$ and $f(y_{k+1}^j) - f(x_k) \geq m_1 v_k^j$ we have at t_k^*

$$f(x_k + d^*) - f(x_k) = m_1 v^* \tag{33}$$

Hence

$$g^* \in \partial f(x_k + d^*) \Rightarrow$$
$$(g^*, x_k - x_k - d^*) \leq f(x_k) - f(x_k + d^*)$$

and using (33)

$$(g^*, d^*) \geq m_1 v^* \tag{34}$$

thus in this iteration, algorithm will terminate with Serious Step. It is sufficient to use only the constant m_1 but in (34) we replace m_1 by $m_2 > m_1$ to relax this condition. The purpose of this relaxation is to speed up termination with this "critical" case.

The constants m_1, m_2 are scaling constants with the following behavioral limits :

- $1 > m_2 > m_1 \to 1$:
  Serious Step is more difficult to fulfill, since we ask relatively large decrease of $v_k = \hat{f}(x_k, d)$. Therefore many iterations will conclude in Null Step. The new constraint in the model (32), in next iteration $k+1$, will be less active (see Fig. 9).

  Our experiments show that easy problems converge fast, difficult problems converge fast at the beginning but after several iterations we observe "zig-zagging" behavior.

- $0 < m_1 < m_2 \to 0$:
  Serious Step is easy to fulfill. The new constraint in the model (32), in next iteration $k+1$, will be more active. The speed of decreasing duality gap is relatively slow, but the tendency to "zig-zag" of difficult problems is reduced.

For both cases, the average speed of decreasing the duality gap is shown in Fig. 10. Our measures show that the choice $m_1 := 0.1$ and $m_2 := 0.2$ is a good compromise between the speed of decreasing the duality gap and the tendency to "zig-zagging".

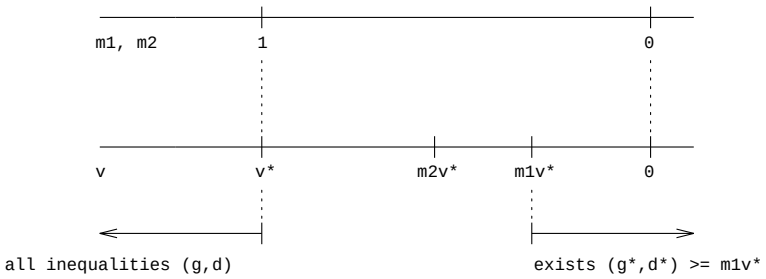

Figure 9: Relations between m_1, m_2 and v

5.5 Choice of Constant β

This constant is set to a small error ratio, typically $\beta := 0.1$.

6 Computational Results

To evaluate LA-algorithm we used the same test-problems as in Balas and Saltzman [5], obtained thanks to M.J. Saltzman via ftp anonymous from the file : math.clemson.edu:/Saltzman/printap3.out.Z.

The integer cost coefficients are generated from 0 to 100 and the size n of these problems ranges from 4 up to 26.

We solve the AP3 problem as maximum in order to maintain coherence with the papers which describe the Bundle Trust method (Schramm and Zowe [9],[10]). Since in Balas and Saltzman [5], the AP3 problem is solved as minimum, we transformed the cost coefficients as follows : $c'_{ijk} = 100 - c_{ijk}$.

On these test-problems, we evaluated several methods :
Method 1 is the standard subgradient method, described in Frieze and Yadegar [1], next Method 2 is an improvement of Method 1 by incorporation of

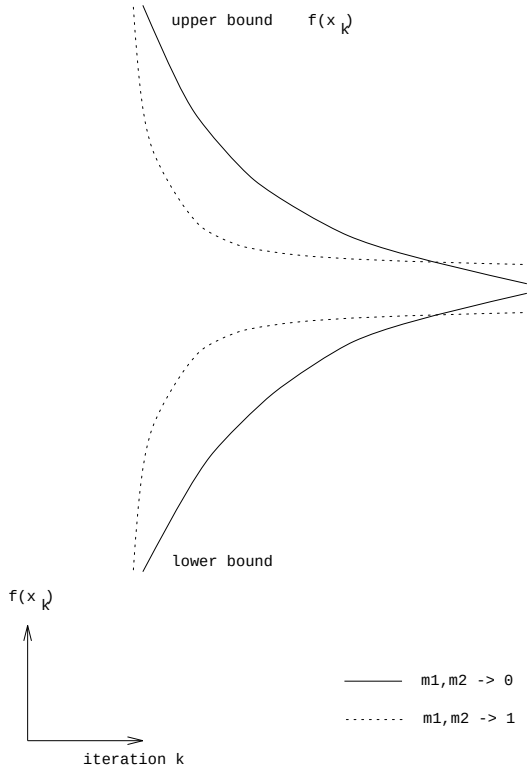

Figure 10: Average speed of decreasing the duality gap for $m_1, m_2 \to 0, 1$

all class-2 cliques. Method 3 is the subgradient method with modified subgradient that takes into account the recent history of direction. This strategy is described in Camerini, Fratta and Maffioli [3] : if the angle between the new and the previous direction is greater than some fixed value we don't trust in the new direction and carry on the old one; it must be thought as a second order refinement of standard subgradient and not to be confused either with an ϵ-subgradient or with a local approximation of the polytope. Method 4 is the Bundle Trust method described in Section 5. Method 5 and 6 are the LA-algorithms with incorporated class-2 cliques and odd-holes p = 2, respectively. Method 7 and 8 are the skeleton LA-algorithms with incorporated class-2 cliques and odd-holes p = 2, respectively. We give once more the result of

Method 4 to show that LA-algorithms lead to better result than the Bundle Trust method with increasing order of the size of problem.

The column "value" contains the best value of $\hat{\phi}$ and the column "loop" contains number of iterations in which $\hat{\phi}$ was reached. The subgradient procedure terminates if too many iterations occur without any improvement of $\hat{\phi}$. We fix this number to n, the problem size.

For every problem, the method which yields best result, is highlighted bold type style. According to best value of $\hat{\phi}$ and ordinal iteration in case of ties; whenever several methods achieve the same best value, no emphasis is done.

All code was written in Maple V programming language.

size	Method 1 Frieze [1]		Method 2 Balas n^3 [5]		Method 3 Maffioli [3]		Method 4 Bundle-Trust [9]	
	value	loop	value	loop	value	loop	value	loop
4	354	1	354	1	354	1	354	1
4	353	2	353	7	353	2	**353**	1
4	352	2	352	4	352	2	352	3
4	**347**	4	345	4	347	6	345	2
4	**383**	5	383	6	383	15	383	12
6	**552**	2	537	4	**552**	2	537	13
6	549	16	549	30	549	17	547	2
6	558	4	558	4	558	5	558	14
6	557	8	556	1	557	17	557	5
6	567	5	563	33	567	10	**567**	4
8	776	1	776	1	776	1	776	1
8	758	2	758	7	**768**	17	749	11
8	**777**	10	761	1	777	18	773	6
8	763	8	755	13	763	28	**763**	7
8	792	1	792	1	792	1	792	1
10	970	16	965	5	975	19	965	4
10	982	39	983	25	984	43	978	3
10	979	33	979	13	963	1	**979**	4
10	979	8	979	16	956	2	**979**	6
10	983	11	**983**	3	983	30	965	12
12	1159	14	**1178**	14	1159	43	1170	4
12	1160	1	1164	11	1160	1	**1169**	7
12	1175	28	1175	9	1175	61	**1189**	6
12	1180	36	1177	2	1175	3	1180	15
12	**1177**	17	1175	26	1171	1	1171	1
14	1374	81	1372	24	1368	6	1374	3
14	1377	20	**1382**	25	1368	6	1371	3
14	1382	11	1364	2	1371	19	1373	14
14	1370	45	1366	4	1371	8	**1381**	7
14	1375	7	1388	58	1375	17	1364	1

Table 1: Computational results

size	Method 1 Frieze [1]		Method 2 Balas n^3 [5]		Method 3 Maffioli [3]		Method 4 Bundle-Trust [9]	
	value	loop	value	loop	value	loop	value	loop
16	1564	51	**1583**	27	1555	8	1557	9
16	1583	61	1580	22	1580	47	1572	3
16	1569	89	1575	33	1578	24	1570	6
16	1559	15	1567	38	1562	31	1561	6
16	1572	48	1572	6	1570	51	**1584**	12
18	1771	11	**1789**	20	1752	12	1764	10
18	1778	20	1769	18	1776	14	**1787**	26
18	1767	68	1760	10	1761	21	**1784**	13
18	1774	78	1776	21	1762	2	1767	11
18	1781	10	**1786**	5	1758	1	1758	1
20	1965	15	1971	6	1960	8	1947	1
20	**1982**	40	1979	37	1954	29	1930	1
20	1961	26	**1984**	22	1969	19	1970	11
20	1973	79	1983	9	1962	27	**1987**	18
20	1969	30	1978	20	1941	2	1907	1
22	2167	58	2165	29	2159	2	2175	23
22	2165	5	2175	38	2158	1	2158	1
22	2167	81	**2183**	35	2156	42	2126	1
22	2156	4	2174	8	2158	15	2174	8
22	2168	37	2168	10	2145	1	2145	1
24	2368	15	2371	21	2367	20	2323	1
24	2371	53	2368	31	2352	1	2352	1
24	2374	115	2385	76	2375	32	2345	1
24	**2375**	30	2371	14	2367	1	2367	1
24	2370	44	**2381**	11	2365	42	2330	1
26	2567	62	**2578**	2	2558	38	2530	1
26	2577	46	2571	3	2572	9	2559	1
26	2562	92	2563	3	2547	1	2547	1
26	2565	23	**2580**	52	2560	39	2521	1
26	2568	139	**2577**	17	2548	13	2515	1

Table 2: Computational results

size	Method 5 LA_{c2}		Method 6 LA_{oh2}		Method 7 skeleton LA_{c2}		Method 8 skeleton LA_{oh2}		Method 4 Bundle-Trust [9]	
	value	loop	value	loop	value	loop	value	loop	value	loop
4	354	1	354	1	354	1	354	1	354	1
4	353	2	353	2	353	2	353	2	**353**	1
4	352	4	352	2	352	2	352	2	352	3
4	345	5	345	3	347	8	345	7	345	2
4	383	14	366	9	383	16	358	2	383	12
6	548	18	544	8	548	3	544	5	537	13
6	549	3	**558**	7	558	32	**558**	7	547	2
6	558	4	558	5	558	4	558	4	558	14
6	**557**	3	557	18	**557**	3	557	8	557	5
6	563	3	549	1	567	12	549	1	**567**	4
8	776	1	776	1	776	1	776	1	776	1
8	760	7	758	26	760	7	764	19	749	11
8	773	27	773	23	773	34	773	11	773	6
8	763	12	763	9	763	23	745	4	**763**	7
8	792	1	792	1	792	1	792	1	792	1
10	966	6	975	16	966	4	**975**	12	965	4
10	**984**	8	983	28	**984**	8	981	13	978	3
10	963	1	979	30	979	33	979	28	**979**	4
10	965	18	979	22	979	33	979	59	**979**	6
10	983	16	983	10	983	9	983	30	965	12
12	1166	10	1169	38	1159	13	1160	8	1170	4
12	1169	39	1160	1	1160	1	1160	1	**1169**	7
12	1164	13	1175	28	1166	13	1189	19	**1189**	6
12	1181	7	**1183**	16	1167	10	1181	47	1180	15
12	1171	1	1175	12	1175	18	1171	1	1171	1
14	1383	67	1372	57	1383	24	**1383**	18	1374	3
14	1381	23	1369	3	1369	27	1369	3	1371	3
14	1385	65	**1385**	7	1382	30	1385	12	1373	14
14	1374	54	1380	46	1373	7	1371	15	**1381**	7
14	1388	52	1375	9	**1388**	24	1388	41	1364	1

Table 3: Computational results

size	Method 5 LA_{c2}		Method 6 LA_{oh2}		Method 7 skeleton LA_{c2}		Method 8 skeleton LA_{oh2}		Method 4 Bundle-Trust [9]	
	value	loop	value	loop	value	loop	value	loop	value	loop
16	1573	34	1566	13	1577	38	1570	25	1557	9
16	**1587**	61	1580	87	1579	36	1585	27	1572	3
16	1578	21	1578	32	1578	21	**1578**	11	1570	6
16	**1575**	48	1573	53	1566	49	1560	7	1561	6
16	1579	28	1584	58	1571	10	1584	29	**1584**	12
18	1778	83	1773	22	1766	29	1772	70	1764	10
18	1787	42	1776	63	1768	9	1782	25	**1787**	26
18	1781	55	1782	27	1773	33	1782	37	**1784**	13
18	**1778**	31	1775	39	1764	28	1763	17	1767	11
18	1780	90	1781	36	1781	17	1781	20	1758	1
20	1968	61	**1979**	21	1977	81	1971	55	1947	1
20	1969	18	1968	86	1967	37	1982	46	1930	1
20	1976	62	1973	114	1963	13	1976	57	1970	11
20	1972	49	1977	26	1964	48	1979	19	**1987**	18
20	1961	5	1971	62	1971	54	**1982**	83	1907	1
22	2174	96	**2186**	19	2169	59	2175	42	2175	23
22	2165	13	**2183**	82	2172	27	2170	28	2158	1
22	2169	74	2169	99	2155	64	2168	20	2126	1
22	2166	26	2180	117	2163	29	**2189**	122	2174	8
22	2174	43	2170	58	2174	55	**2181**	64	2145	1
24	2366	44	2370	39	**2380**	62	2374	70	2323	1
24	2373	101	2375	117	2365	90	**2383**	63	2352	1
24	2380	126	2385	90	**2387**	40	2377	23	2345	1
24	2367	1	2379	45	2367	1	2367	1	2367	1
24	2375	130	2374	75	2359	10	2377	48	2330	1
26	2567	41	2572	58	2562	124	2570	57	2530	1
26	2568	43	**2581**	5	2571	73	**2581**	5	2559	1
26	2566	29	2564	10	2564	23	**2573**	55	2547	1
26	2564	87	2569	67	2563	101	2566	20	2521	1
26	2566	100	2576	126	2563	57	2570	52	2515	1

Table 4: Computational results

Moreover, to demonstrate efficiency of skeleton improvement, let us consider the AP3 pathological problem, with the cost coefficients set as follows :

$$c_{i_1,j_1,k_1} = c_{i_2,j_2,k_1} = c_{i_3,j_3,k_1} = v_1$$

$$c_{i_1,j_2,k_2} = c_{i_2,j_3,k_2} = c_{i_3,j_1,k_2} = v_2$$

$$c_{i_1,j_3,k_3} = c_{i_2,j_1,k_3} = c_{i_3,j_2,k_3} = v_3$$

where $i_1 \neq i_2 \neq i_3 \in [1,n]$, $j_1 \neq j_2 \neq j_3 \in [1,n]$, $k_1 \neq k_2 \neq k_3 \in [1,n]$, $v_1 \neq v_2 \neq v_3 > 0$ and $c_{ijk} = 0$ otherwise. If we consider only trivial facets, this regular structure yields many optimal solutions $\hat{\Phi}$ for every Lagrange multipliers. However, only a restricted number of them leads to global optimum. Since standard algorithms are deterministic this leads to "zig-zagging". Only LA-algorithms enhanced with skeleton avoids "zig-zagging" and finds out global optimum of this problem.

7 Conclusion

In this paper we described a heuristic based on a local approximation of the AP3 polytope w.r.t duality gap (relaxed/feasible solutions). Our computational experiments report that it gives better result than Bundle Trust method, based on an approximation of subgradient; moreover, computation remains far simpler than the Bundle Trust method since it does not require intermediate quadratic optimization. It could be worthwhile to study different tradeoffs between both approaches.

Then, it compares favorably to standard methods using facets, as it reduces their number within a tractable bound, $O(n^2)$, affording a more accurate choice among them. Finally, we introduce a restricted Monge sequence on a local substructure of AP3 w.r.t duality gap which prevents "zig-zagging" in some pathological cases.

Upto Monge sequence improvement, we guess that this heuristic could be applied on different problems (set packing problem, set covering problem), provided their underlying structure allows an efficient sieving of facets w.r.t duality gap.

References

[1] Frieze A.M. and Yadegar J. An algorithm for solving 3-dimensional assignment problem with application to scheduling a teaching. *J.Oper.Res.Soc*, 32:989–995, 1981.

[2] Lemarechal C. Basic theory in nondifferentiable optimization. *Institut National de Recherche en Informatique et en Automatique, rapport de recherche n.181*, 1982.

[3] Fratta L. Camerini P. and Maffioli F. On improving relaxation methods by modified gradient techniques. *Math. Prog. Study*, 3:26–34, 1975.

[4] Balas E. and Saltzman M. J. Facets of the three-index assignment polytope. *Discrete Applied Mathematics*, 23:201–229, 1989.

[5] Balas E. and Saltzman M. J. An algorithm for the three-index assignment problem. *Operations Research*, 39(1):150–161, 1991.

[6] Balas E. and Qi L. Linear-time separation algorithms for the three-index assignment polytope. *Discrete Applied Mathematics*, 43:1–12, 1993.

[7] Burkard R. E. and Rudolf R. Computational investigations on 3-dimensional axial assignment problem. *Belgian Journal of Operations Research*, 32(1–2):85–98, 1993.

[8] Gwan G. and Qi L. On facets of the three-index assignment polytope. *Australasian Journal of Combinatorics*, 6:67–87, 1992.

[9] Schramm H. and Zowe J. A combination of the bundle approach and the trust region concept. *Math. Research*, 45:196–209, 1988.

[10] Schramm H. and Zowe J. A version of the bundle idea for minimizing a nonsmooth function : Conceptual idea, convergence analysis, numerical results. *SIAM J. Optimization*, 2(1):121–152, February 1992.

[11] Wolfe P. Held M. and Crowder H. P. Validation of subgradient optimization. *Mathematical Programming*, 6:62–88, 1974.

[12] Hansen P. and Kaufman L. A primal-dual algorithm for the three-dimensional assignment problem. *Cahiers du CERO*, 15:327–336, 1973.

[13] Pierskala W. P. The multidimensional assignment problem. *Operations Research*, pages 422–431, 1968.

[14] Balas E. Qi L. and Gwan G. A new facet class and a polyhedral method for the three-index assignment problem. *Management Science Research Report MSRR-592, Carnegie Mellon University, Pittsburgh*, 1993.

[15] Rudolf R. On monge sequence in d-dimensional arrays. *Technische Universitaet Graz, Institut fur Mathematik Report 247*, 1993.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249- 6399