Variant of *KP01*: *Multiple Choice KP* (*MCKP*)

In addition to the input data for KP01:

the set of the n items is *partitioned* into k disjoint subsets $N_1, N_2, \ldots, N_k$.

- **determine a subset of the n items, with at most one item for each subset N_h ($h = 1, \ldots, k$), so as to maximize the global profit, and such that the global weight is not larger than the knapsack capacity C.**

BLP Model for *MCKP* (2)

- **determine a subset of the n items, with at most one item for each subset N_h ($h = 1, \ldots, k$), so as to maximize the global profit, and such that the global weight is not larger than the knapsack capacity C.**

$$\max \quad \sum_{j=1,n} P_j \, x_j$$

$$\sum_{j=1,n} W_j \, x_j \leq C$$

$$\sum_{j \in N_h} x_j \leq 1 \qquad (h = 1, \ldots, k)$$

$$x_j \in \{0, 1\} \qquad (j = 1, \ldots, n)$$

BLP Model **_MCKP_ is NP-Hard**

BLP Model for *MCKP* (3)

* Define the *Binary Matrix* A_{hj} ($h = 1, \ldots, k$; $j = 1, \ldots, n$), with:

- $A_{hj} = 1$ if $j \in N_h$
- $A_{hj} = 0$ otherwise.
- *Matrix* A_{hj} belongs to the input data of the instance

$$\max \quad \sum_{j=1,n} P_j \, x_j$$

$$\sum_{j=1,n} W_j \, x_j \leq C$$

$$\sum_{j=1,n} A_{hj} \, x_j \leq 1 \qquad (h = 1, \ldots, k)$$

$$x_j \in \{0, 1\} \qquad (j = 1, \ldots, n)$$

Multiple Choice KP (*MCKP*) is NP-Hard

***MCKP*: in addition to the input data for KP01:**

the set of the n items is *partitioned* into k disjoint subsets N_1, N_2 ,…, N_k .

- **determine a subset of the n items, with at most one item for each subset N_h (h = 1, …, k), so as to maximize the global profit, and such that the global weight is not larger than the knapsack capacity C.**

- **Input: m, C, k, (P_j), (W_j) (j = 1, …, n), N_h (h = 1, …, k)**
- **Size: 3 + 2 n + k * n (matrix A_{hj}), with $k \le n$: n * n**
- **Size: 3 + 2 n + n (partition of the set {1, 2, …, n}) : n .**
- ***Binary Decision Tree*: similar to the decision tree of *KP-01*: *n levels, 2 descendent nodes and constant time for each node*:**
- ***MCKP* ∈ *Class NP* ;**
- ***MCKP* is a "generalization" of *KP-01* : *KP-01* ∝ *MCKP***

BLP Model for *MCKP*

* *Binary Matrix* A_{hj} ($h = 1, \ldots, k$; $j = 1, \ldots, n$), with:

- $A_{hj} = 1$ if $j \in N_h$; $A_{hj} = 0$ otherwise.

$$\max \quad \sum_{j=1,n} P_j \, x_j$$

$$\sum_{j=1,n} W_j \, x_j \leq C$$

$$\sum_{j=1,n} A_{hj} \, x_j \leq 1 \qquad (h = 1, \ldots, k)$$

$$x_j \in \{0, 1\} \qquad (j = 1, \ldots, n)$$

The *BLP Model* has a number of binary variables x_j **polynomial** in the size of *MCKP*:

$$MCKP \in \textit{Class NP}$$

Multiple Knapsack Problem (MKP01)

Given: ***n*** **items, *m* containers (knapsacks)**

P_j **profit of item *j*** **($j = 1, \ldots, n$)**

W_j **weight of item *j*** **($j = 1, \ldots, n$)**

C_i **capacity of container *i*** **($i = 1, \ldots, m$)**

insert a subset of the n items in each container in order to maximize the global profit of the items inserted in the containers, and in such a way that the sum of the weights of the items inserted in each container i (i = 1, …, m) is not greater than the corresponding capacity C_i

Each item can be inserted in at most one container.

MKP01 (2)

Given: n items, m containers (knapsacks)

P_j profit of item j $\quad (j = 1, \ldots, n)$

W_j weight of item j $\quad (j = 1, \ldots, n)$

C_i capacity of container i $\quad (i = 1, \ldots, m)$

insert a subset of the n items in each container in order to maximize the global profit of the items inserted in the containers, and in such a way that the sum of the weights of the items inserted in each container i (i = 1, …, m) is not greater than the corresponding capacity C_i

$P_j > 0 \quad (j = 1, \ldots, n)$

$W_j > 0 \quad (j = 1, \ldots, n)$

MKP01 (3)

Given: n items, m containers (knapsacks)

P_j profit of item j $\quad (j = 1, \ldots, n)$

W_j weight of item j $\quad (j = 1, \ldots, n)$

C_i capacity of container i $\quad (i = 1, \ldots, m)$

$P_j > 0 \;\; (j = 1, \ldots, n); \;\; W_j > 0 \;\; (j = 1, \ldots, n)$

$\sum_{j=1,n} W_j > \max\{ C_i : i = 1, \ldots, m\}$

$W_j \leq \max\{ C_i : i = 1, \ldots, m\} \quad (j = 1, \ldots, n)$

$\min\{ C_i : i = 1, \ldots, m\} \geq \min\{ W_j : j = 1, \ldots, n\}$

Mathematical Model of MKP01

$$x_{ij} = \begin{cases} 1 & \text{if item } j \text{ is inserted in container } i \\ 0 & \text{otherwise} \end{cases} \qquad (i = 1, \ldots, m;\ j = 1, \ldots, n)$$

$$\max \ \Sigma_{j=1,n}\ P_j\ (\Sigma_{i=1,m}\ x_{ij})$$

$$\Sigma_{j=1,n}\ W_j\ x_{ij} \le C_i \qquad (i = 1, \ldots, m)$$

$$x_{ij} \in \{0,1\} \qquad (i = 1, \ldots, m;\ j = 1, \ldots, n)$$

???

MKP01 is NP-Hard

MKP01: given: n items, m containers (knapsacks),
P_j profit of item j, W_j weight of item j ($j = 1, \ldots, n$),
C_i capacity of container i ($i = 1, \ldots, m$):
insert a subset of the n items in each of the m containers in order to maximize the global profit of the inserted items, and in such a way that the global weight of the items inserted in each container i (i = 1, …, m) is not greater than the corresponding capacity C_i

Input: n, m, (P_j), (W_j) ($j = 1, \ldots, n$), (C_i) ($i = 1, \ldots, m$)

- Size: $2 + 2n + m$: $n + m$, ($m \leq n$: Size n)
- *Decision Tree*: *n levels* (one for each item j);

  (m +1) *descendent nodes* (*insert item j in knapsack 1, or 2, …, or m, or in no knapsack*) *and constant time for each node*:

  MKP01 ∈ *Class NP* ;

  (*BLP model* with ($m * n$) binary variables x_{ij})

- *MKP01* is a "generalization" of *KP-01* :

  KP-01 ∝ *MCKP*

Mathematical Model of MKP01

$$x_{ij} = \begin{cases} 1 & \text{if item } j \text{ is inserted in container } i \\ 0 & \text{otherwise} \end{cases} \qquad (i = 1, \ldots, m;\; j = 1, \ldots, n)$$

$$\max \sum_{j=1,n} P_j \left(\sum_{i=1,m} x_{ij}\right)$$

$$\sum_{j=1,n} W_j\, x_{ij} \leq C_i \qquad (i = 1, \ldots, m)$$

$$\sum_{i=1,m} x_{ij} \leq 1 \qquad (j = 1, \ldots, n)$$

$$x_{ij} \in \{0,1\} \qquad (i = 1, \ldots, m;\; j = 1, \ldots, n)$$

BLP Model

MKP01 is NP-Hard

Generalized Assignment Problem (GAP)

Given: ***m machines*** **(persons) and** ***n jobs*** **(tasks):**

c_{ij} **cost for assigning job j to machine i ($i = 1, \ldots, m$; $j = 1, \ldots, n$);**

r_{ij} ***amount of resource utilized for*** **assigning job j to machine i ($i = 1, \ldots, m$; $j = 1, \ldots, n$); $r_{ij} \geq 0$;**

b_i **amount of resource available for machine i ($i = 1, \ldots, m$), $b_i > 0$.**

Assign each job to a machine so as to minimize the global cost, and in such a way that the global resource utilized by each machine i is not greater than the corresponding available resource b_i

Generalized Assignment Problem (GAP)

Assign each job to a machine so as to minimize the global cost, and in such a way that the global resource utilized by each machine i is not greater than the corresponding available resource b_i.

***GAP* is NP-Hard**

The *Feasibility Problem* of *GAP* is NP-Hard

Decisional binary variables:

x_{ij} = 1 if job j is assigned to machine i;

x_{ij} = 0 otherwise; $(i = 1, \ldots, m;\ j = 1, \ldots, n)$

Mathematical Model of *GAP*

- **Objective function (minimum cost)**

  $$\min \quad \Sigma_{i=1,m} \Sigma_{j=1,n} \, c_{ij} \, x_{ij}$$

- **One machine assigned to each job:**

  $$\Sigma_{i=1,m} \; x_{ij} \;=\; 1 \qquad (\,j = 1, \ldots, n)$$

- **Resource utilized for each machine:**

  $$\Sigma_{j=1,n} \; r_{ij} \, x_{ij} \;\leq\; b_i \qquad (\,i = 1, \ldots, m)$$

$$x_{ij} \in \{0, 1\} \qquad (i = 1, \ldots, m, \; j = 1, \ldots, n)$$

BLP Model

Maximization Version of *GAP* (*Max-GAP*)

- **Objective function (maximum "cost")**

$$\max \quad \Sigma_{i=1,m} \Sigma_{j=1,n} c_{ij} x_{ij}$$

- **One machine assigned to each job:**

$$\Sigma_{i=1,m} x_{ij} = 1 \qquad (j = 1, \ldots, n)$$

- **Resource utilized for each machine:**

$$\Sigma_{j=1,n} r_{ij} x_{ij} \leq b_i \qquad (i = 1, \ldots, m)$$

$$x_{ij} \in \{0, 1\} \qquad (i = 1, \ldots, m,\ j = 1, \ldots, n)$$

GAP is NP-Hard

Given: ***m* machines** and ***n* jobs**:

c_{ij} **cost** (**r_{ij} *amount of resource utilized***) **for assigning job j to machine i** ($i = 1, \ldots, m$; $j = 1, \ldots, n$);

b_i **amount of resource available for machine i** ($i = 1, \ldots, m$):

Assign each job to a machine so as to minimize the global cost, and in such a way that the global resource utilized by each machine i is not greater than the corresponding available resource b_i.

Input: $m, n, (c_{ij}), (r_{ij})$ $(i = 1, \ldots, m; j = 1, \ldots, n)$;

(b_i) $(i = 1, \ldots, m)$

Size: $2 + 2\, m * n + m : \; m * n$

The *Feasibility Problem of GAP (F-GAP)* is NP-Hard.

Feasibility Problem of GAP (F-GAP)

Given: ***m* machines and *n* jobs:**

r_{ij} *amount of resource utilized* for assigning job j to machine i (i = 1, …, m; j = 1, …, n);

b_i amount of resource available for machine i (i = 1, …, m):

Assign each job to a machine in such a way that the global resource utilized by each machine i is not greater than the corresponding available resource b_i.

Input: m, n, (r_{ij}) (i = 1, …, m; j = 1, …, n); (b_i) (i = 1, …, m):

Size: $m * n$

- ***Decision Tree*: *n levels* (one for each job *j*);**

*** *m descendent nodes* (*insert job j in machine 1, or 2, …, or m*) *and constant time for each node*:**

F-GAP* ∈ *Class NP

Also *GAP* ∈ *Class NP* (same *Size* and *Decision Tree* as *F-GAP*); (*BLP model* with ($m * n$) binary variables x_{ij})

Feasibility Problem of GAP (F-GAP)

Given: *m machines* **and** *n jobs*:

r_{ij} *amount of resource utilized* **for assigning job** j **to machine** i ($i = 1, \ldots, m$; $j = 1, \ldots, n$);

b_i **amount of resource available for machine** i ($i = 1, \ldots, m$):

PP $\propto$ ***F-GAP*** **:**

- **Given any instance of *PP*:** t, (a_j), b **(Size:** t**)**

1) Define (in time $O(t)$**) an instance (**$m, n, (r_{ij}), (b_i)$ **) of *F-GAP*:**

  * $n := t$

  * $m := 2$; $b_1 := b$; $b_2 := \Sigma_{j=1,t}\, a_j - b$

  * $r_{1j} := a_j$; $r_{2j} := a_j$ ($j = 1, \ldots, n$).

2) Determine (if it exists) a feasible solution (x_{1j} , x_{2j}**) of *F-GAP*.**

3) If a feasible solution of *F-GAP* exists, then *PP* has a feasible solution (x_{1j} , x_{2j} **)**

Otherwise: *PP* has no feasible solution.

Computing time $O(n)$ **(hence** $O(t)$**, polynomial in the size of *PP*).**

*** *F-GAP* is NP-Hard**

Bin Packing Problem (BPP)

Given:

n items;

W_j weight of item j ($j = 1, \dots, n$) ($W_j > 0$);

m containers (bins), each with capacity C:

insert *all the n items in the containers in order to minimize the number of used containers, and in such a way that the sum of the weights of the items inserted in a container is not greater than the capacity C.*

$W_j < C \quad j = 1, \dots, n$

$\sum_{j=1,n} W_j > C$

Bin Packing Problem (BPP)

Given:

n **items;**

W_j **weight of item *j* (*j* = 1, …, *n*)** ($W_j > 0$);

m **containers (bins), each with capacity *C*:**

insert *all the n items in the containers in order to minimize the number of used containers, and in such a way that the sum of the weights of the items inserted in a container is not greater than the capacity C.*

BPP is NP-Hard

The *Feasibility Problem* of *BPP* is NP-Hard

Mathematical Model of BPP

$$x_{ij} = \begin{cases} 1 & \text{if item } j \text{ is inserted in container } i \\ 0 & \text{otherwise} \end{cases} \qquad (i = 1, \ldots, m;\ j = 1, \ldots, n)$$

(**)

$$y_i = \begin{cases} 1 & \text{if container } i \text{ is used} \\ 0 & \text{otherwise} \end{cases} \qquad (i = 1, \ldots, m)$$

Mathematical Model of BPP (2)

$$\min \ \sum_{i=1,m} y_i$$

$$\sum_{j=1,n} W_j x_{ij} \le C \qquad (i = 1, \ldots, m)$$

$$\sum_{i=1,m} x_{ij} = 1 \qquad (j = 1, \ldots, n)$$

$$y_i \in \{0, 1\} \qquad (i = 1, \ldots, m)$$

$$x_{ij} \in \{0, 1\} \qquad (i = 1, \ldots, m; \ j = 1, \ldots, n)$$

???

Bin Packing Problem (*BPP*) is NP-Hard

Given: n **items;** m **bins (each with capacity** C**);**

W_j **weight of item** j **(** $j = 1, \ldots, n$**):**

insert *all the n items in the bins in order to minimize the number of used bins, and in such a way that the global weight of the items inserted in a bin is not greater than the capacity C.*

- **Input:** $n, m, C, (W_j)$ $(j = 1, \ldots, n)$; **Size:** $3 + n$: n
- $m \leq n$

Feasibility Problem of BPP (F-BPP)

Given: n **items;** m **bins (each with capacity** C**);**

W_j **weight of item** j **(**$j = 1, \ldots, n$**):**

insert ***all the n items in the m bins in such a way that the global weight of the items inserted in a bin is not greater than the capacity C.***

F-BPP is NP-Hard

- **Input:** $n, m, C, (W_j)$ $(j = 1, \ldots, n)$; **Size:** $3 + n : n$
- ***Decision Tree*: *n levels* (one for each item *j*);**

*** *m descendent nodes* (*insert item j in bin 1, or 2, …, or m*) *and constant time for each node* (**$m \leq n$ **):**

F-BPP $\in$ ***Class NP*** **;**

Also ***BPP*** $\in$ ***Class NP*** **(same** ***Size*** **and** ***Decision Tree*** **as** ***F-BPP*****);**

(*BLP model* with ($m * n + m$**) binary variables** x_{ij}, y_i **)**

F-BBP is NP-Hard

Given: n items; m bins (each with capacity C);

W_j weight of item j ($j = 1, \ldots, n$):

insert *all the n items in the m bins in such a way that the global weight of the items inserted in a bin is not greater than the capacity C.*

- $PP \propto$ *F-BPP* :
- Given any instance of *PP*: t, (a_j), b (Size: t)

1) Define (in time $O(t)$) an instance $(n, (W_j), m, C)$ of *F-BPP*:
   - $n := t$
   - $C := b$
   - $m := 2$
   - $W_j := a_j$ ($j = 1, \ldots, n$).
2) Determine (if it exists) a feasible solution (x) of *F-BPP.*
3) If a feasible solution (x_{1j} , x_{2j}) of *F-BPP* exists, then *PP* has a feasible solution (x_{1j} , x_{2j})

   Otherwise: *PP* has no feasible solution.

   Computing time $O(n)$ (hence $O(t)$, polynomial in the size of *PP*)

F-BPP is a particular case of F-GAP

F-GAP: **given:** ***m machines*** **and** ***n jobs*****:**

r_{ij} ***amount of resource utilized*** **for assigning job** j **to machine** i **(**$i = 1, \ldots, m$**;** $j = 1, \ldots, n$**);**

b_i **amount of resource available for machine** i **(**$i = 1, \ldots, m$**):**

assign each job to a machine so that the global resource utilized by each machine i is not greater than the available resource b_i**.**

F-BPP: **given:** n **items;** m **bins (each with capacity** C**);**

W_j **weight of item** j **(**$j = 1, \ldots, n$**):**

insert ***all the n items in the m bins so that the global weight of the items inserted in a bin is not greater than the capacity C.***

Arising when:

$r_{ij} := W_j \quad (i = 1, \ldots, m;\ j = 1, \ldots, n);$

$b_i := b \quad (i = 1, \ldots, m)$

Mathematical Model of BPP (2)

(M1) $\min \sum_{i=1,m} y_i$

$$\sum_{j=1,n} w_j x_{ij} \le C \qquad (i = 1, \ldots, m)$$

$$\sum_{i=1,m} x_{ij} = 1 \qquad (j = 1, \ldots, n)$$

$$x_{ij} \le y_i \qquad (i = 1, \ldots, m;\ j = 1, \ldots, n)$$

$$y_i \in \{0, 1\} \qquad (i = 1, \ldots, m)$$

$$x_{ij} \in \{0, 1\} \qquad (i = 1, \ldots, m;\ j = 1, \ldots, n)$$

BLP Model

Mathematical Model of BPP (2)

(M1) min $\sum_{i=1,m} y_i$

$\sum_{j=1,n} w_j \, x_{ij} \leq C$ $(i = 1, \ldots, m)$

$\sum_{i=1,m} x_{ij} = 1$ $(j = 1, \ldots, n)$

$x_{ij} \leq y_i$ $(i = 1, \ldots, m; j = 1, \ldots, n)$

$y_i \in \{0, 1\}$ $(i = 1, \ldots, m)$

$x_{ij} \in \{0, 1\}$ $(i = 1, \ldots, m; \; j = 1, \ldots, n)$

$(m + n + m\,n)$ constraints

Alternative Models of BPP

(M2) $\min \sum_{i=1,m} y_i$

$\sum_{j=1,n} w_j x_{ij} \le C$ $\quad (i = 1, \ldots, m)$

$\sum_{i=1,m} x_{ij} = 1$ $\quad (j = 1, \ldots, n)$

$\sum_{j=1,n} x_{ij} \le M y_i$ $\quad (i = 1, \ldots, m) \quad M \ge n$

$y_i \in \{0, 1\}$ $\quad (i = 1, \ldots, m)$

$x_{ij} \in \{0, 1\}$ $\quad (i = 1, \ldots, m; \; j = 1, \ldots, n)$

$(2m + n)$ constraints

Alternative Models of BPP (2)

(M3) $\min \sum_{i=1,m} y_i$

$\sum_{j=1,n} W_j x_{ij} \le C\, y_i$ $\quad (i = 1, \ldots, m)$

$\sum_{i=1,m} x_{ij} = 1$ $\quad (j = 1, \ldots, n)$

$y_i \in \{0, 1\}$ $\quad (i = 1, \ldots, m)$

$x_{ij} \in \{0, 1\}$ $\quad (i = 1, \ldots, m;\ j = 1, \ldots, n)$

$(m + n)$ constraints

Alternative Models of BPP (3)

(M1) $\Sigma_{j=1,n} W_j x_{ij} \leq C$ $(i = 1, \ldots, m)$

$x_{ij} \leq y_i$ $(i = 1, \ldots, m; j = 1, \ldots, n)$

(M2) $\Sigma_{j=1,n} W_j x_{ij} \leq C$ $(i = 1, \ldots, m)$

$\Sigma_{j=1,n} x_{ij} \leq M y_i$ $(i = 1, \ldots, m)$ $M \geq n$

(M3) $\Sigma_{j=1,n} W_j x_{ij} \leq C y_i$ $(i = 1, \ldots, m)$

- ***EXAMPLE:*** $C = 100$, $W_1 = 50$, $n = 1000, \ldots$
- ***"Linear Relaxation"*** **of the variables** y_i $(0 \leq y_i \leq 1)$,
- $x_{11} = 1$, $y_1 = 0.5$ $(x_{1j} = 0,\ j = 2, \ldots, n)$:

(M2) **and** *(M3)*: **all constraints are satisfied**

(M1) $i = 1, j = 1$: **constraint** $x_{ij} \leq y_i$ $(1 \leq 0.5)$ **is not satisfied**

Linear Relaxation of Model (M1)

- ***Lower Bound LB*** **on the value of the optimal solution of *BPP*:**

  $LB = \Sigma_{j=1,n} W_j / C$ ($LB > 1$); $k = \lceil LB \rceil$

* ***"Linear Relaxation"*** **of the variables** x_{ij} **and** y_i**:**

$0 \le x_{ij} \le 1, \quad 0 \le y_i \le 1 \quad (i = 1, \ldots, m;\ j = 1, \ldots, n).$

- **Optimal solution of the *Linear Relaxation of BPP* (*Model M1*):**
- $y_i = 1 / LB = C / \Sigma_{j=1,n} W_j \ (< 1) \quad i = 1, \ldots, k - 1$
- $y_k = 1 - \Sigma_{i=1, k-1}\, y_j \quad (0 \le y_k < y_1 < 1)$
- $y_h = 0 \quad h = k + 1, \ldots, m$
- $x_{ij} = y_i \quad (0 \le x_{ij} < 1) \quad i = 1, \ldots, m;\ j = 1, \ldots, n$

Linear Relaxation of Model (M1)

- **Optimal solution of the *Linear Relaxation of BPP* (*Model M1*):**
- $y_i = 1 / LB = C / \Sigma_{j=1,n} W_j \quad (< 1) \qquad i = 1, \ldots, k - 1$
- $y_k = 1 - \Sigma_{i=1, k-1}\, y_j \qquad (0 \leq y_k < y_1 < 1)$
- $y_h = 0 \qquad h = k + 1, \ldots, m$
- $x_{ij} = y_i \qquad (0 \leq y_k < 1) \qquad i = 1, \ldots, m;\ j = 1, \ldots, n$
- ***Constraints:***

$\Sigma_{j=1,n} W_j\, x_{ij} \leq C \qquad (i = 1, \ldots, m)$

$\Sigma_{j=1,n} W_j\, y_j = \Sigma_{j=1,n} W_j / LB = C \qquad (i = 1, \ldots, k - 1);$

$\Sigma_{j=1,n} W_j\, y_k < \Sigma_{j=1,n} W_j y_1 = C;$

$\Sigma_{j=1,n} W_j\, y_j = 0 < C \qquad (i = k + 1, \ldots, m)$

Linear Relaxation of Model (M1)

- **Optimal solution of the *Linear Relaxation of BPP* (*Model M1*):**
- $y_i = 1 / LB = C / \Sigma_{j=1,n} W_j \quad (< 1) \qquad i = 1, \ldots, k-1$
- $y_k = 1 - \Sigma_{i=1, k-1} \, y_j \qquad (0 \leq y_k < y_1 < 1)$
- $y_h = 0 \qquad h = k + 1, \ldots, m$
- $x_{ij} = y_i \qquad (0 \leq y_k < 1) \qquad i = 1, \ldots, m; \; j = 1, \ldots, n$
- ***Constraints:***

* $\Sigma_{i=1,m} \; x_{ij} = 1 \qquad (j = 1, \ldots, n)$

$\Sigma_{i=1,m} \, y_j = 1 \qquad (j = 1, \ldots, n)$

* $x_{ij} \leq y_j \qquad (i = 1, \ldots, m; \; j = 1, \ldots, n)$

$x_{ij} = y_j \qquad (i = 1, \ldots, m; \; j = 1, \ldots, n)$

Linear Relaxation of Model (M1)

- **Optimal solution of the *Linear Relaxation of BPP* (*Model M1*):**
- $y_i = 1 / LB = C / \Sigma_{j=1,n} W_j \quad (< 1) \qquad i = 1, \ldots, k - 1$
- $y_k = 1 - \Sigma_{i=1,\, k-1}\, y_j \qquad (0 \leq y_k < y_1 < 1)$
- $y_h = 0 \qquad h = k + 1, \ldots, m$
- $x_{ij} = y_i \qquad (0 \leq y_k < 1) \qquad i = 1, \ldots, m;\ j = 1, \ldots, n$

All the constraints are satisfied: feasible solution!

- ***Objective Function:***

$$(M1) \qquad z = \Sigma_{i=1,m}\, y_i$$

Linear Relaxation of Model (M1)

- **Optimal solution of the *Linear Relaxation of BPP* (*Model M1*):**
- $y_i = 1 / LB = C / \Sigma_{j=1,n} W_j \quad (< 1) \qquad i = 1, \ldots, k - 1$
- $y_k = 1 - \Sigma_{i=1, k-1}\, y_j \qquad (0 \le y_k < y_1 < 1)$
- $y_h = 0 \qquad h = k + 1, \ldots, m$
- $x_{ij} = y_i \qquad (0 \le y_k < 1) \qquad i = 1, \ldots, m;\; j = 1, \ldots, n$

All the constraints are satisfied: feasible solution!

- ***Objective Function:***

(M1) $\quad z = \Sigma_{i=1,m}\, y_i = 1$ **(useless Lower Bound!)**

Assignment Problem (AP)

Particular case of GAP:

***m* = *n*: *n* *machines* (persons) and *n* *jobs* (tasks):**

c_{ij} cost for assigning job *j* to machine *i* (*i* = 1, …, *n*; *j* = 1, …, *n*);

r_{ij} = 1 *amount of resource utilized for* assigning job *j* to machine *i* (*i* = 1, …, *n*; *j* = 1, …, *n*);

b_i = 1 amount of resource available for machine *i* (*i* = 1, …, *n*).

***AP* is a Polynomial Problem solvable in $O(n^3)$ time.**

Mathematical Model of *GAP*

- **Objective function (minimum cost)**

$$\min \quad \Sigma_{i=1,m} \Sigma_{j=1,n} c_{ij} x_{ij}$$

- **One machine assigned to each job:**

$$\Sigma_{i=1,m} x_{ij} = 1 \qquad (j = 1, \ldots, n)$$

- **Resource utilized for each machine:**

$$\Sigma_{j=1,n} r_{ij} x_{ij} \leq b_i \qquad (i = 1, \ldots, m)$$

$$x_{ij} \in \{0, 1\} \qquad (i = 1, \ldots, m, \; j = 1, \ldots, n)$$

BLP Model

Mathematical Model of *AP*

- **Objective function (minimum cost)**

  $$\min \quad \Sigma_{i=1,n} \Sigma_{j=1,n} c_{ij} \, x_{ij}$$

- **One machine assigned to each job:**

  $$\Sigma_{i=1,n} \; x_{ij} = 1 \qquad\qquad (j = 1, \ldots, n)$$

- **Resource utilized for each machine:**

  $$\Sigma_{j=1,n} \; x_{ij} \leq 1 \qquad\qquad (i = 1, \ldots, \mathrm{n})$$

$$x_{ij} \in \{0, 1\} \qquad\qquad (i = 1, \ldots, n, \; j = 1, \ldots, n)$$

BLP Model

Mathematical Model of *AP*

- **Objective function (minimum cost)**

  $$\min \quad \Sigma_{i=1,n} \Sigma_{j=1,n}\, c_{ij}\, x_{ij}$$

- **One machine assigned to each job:**

  $$\Sigma_{i=1,n}\; x_{ij} \;=\; 1 \qquad\qquad (j = 1, \ldots, n)$$

- **Resource utilized for each machine:**

$$\Sigma_{j=1,n}\; x_{ij} \;\leq\; 1\!: \;\; \Sigma_{j=1,n}\; x_{ij} \;=\; 1 \quad (i = 1, \ldots, \mathrm{n})$$

$$0 \leq \; x_{ij} \leq \; 1 \qquad\qquad (i = 1, \ldots, n,\; j = 1, \ldots, n)$$

Mathematical Model of *AP*

- **Objective function (minimum cost)**

  $$\min \quad \Sigma_{i=1,n} \Sigma_{j=1,n} c_{ij}\, x_{ij}$$

- **One machine assigned to each job:**

  $$\Sigma_{i=1,n}\; x_{ij} \;=\; 1 \qquad\qquad (j = 1, \ldots, n)$$

- **Resource utilized for each machine:**

  $$\Sigma_{j=1,n}\; x_{ij} \;=\; 1 \qquad\qquad (i = 1, \ldots, n)$$

  $$x_{ij} \;\geq\; 0 \qquad\qquad (i = 1, \ldots, n,\; j = 1, \ldots, n)$$

LP Model

(the Coefficient Matrix is “Totally Unimodular”)

Maximization Version of *AP* (*Max-AP*)

- **Objective function (maximum "cost")**

  $$\max \quad \Sigma_{i=1,n} \Sigma_{j=1,n} c_{ij} x_{ij}$$

- **One machine assigned to each job:**

  $$\Sigma_{i=1,n} x_{ij} = 1 \qquad (j = 1, \ldots, n)$$

- **Resource utilized for each machine:**

  $$\Sigma_{j=1,n} x_{ij} = 1 \quad (i = 1, \ldots, n)$$

$$x_{ij} \geq 0 \quad (i = 1, \ldots, n, \; j = 1, \ldots, n$$

Min-Max Version of *AP* (*Bottleneck AP*)

- Assume $c_{ij} \geq 0$ ($i = 1, \ldots, n$; $j = 1, \ldots, n$);
- Objective function (minimum cost of an assignment)

$$\min \quad z = \mathrm{Max}\{c_{ij}\, x_{ij} : i = 1, \ldots, n; \;\; j = 1, \ldots, n\}$$

$$\Sigma_{i=1,n} \, x_{ij} = 1 \qquad (j = 1, \ldots, n)$$

$$\Sigma_{j=1,n} \, x_{ij} = 1 \qquad (i = 1, \ldots, n)$$

$$x_{ij} \geq 0 \qquad (i = 1, \ldots, n, \; j = 1, \ldots, n)$$

$$\min \quad z$$

$$z \geq c_{ij}\, x_{ij} \qquad (i = 1, \ldots, n; \;\; j = 1, \ldots, n)$$

BLP Model

Min-Max Version of GAP (*Bottleneck GAP*)

- Assume $c_{ij} \geq 0$ $(i = 1, \ldots, m; \; j = 1, \ldots, n)$;
- Objective function (minimum cost of an assignment)

$$\min \quad z = \text{Max}\{ c_{ij}\, x_{ij} : i = 1, \ldots, m; \; j = 1, \ldots, n\}$$

$$\Sigma_{i=1,m}\; x_{ij} = 1 \qquad (j = 1, \ldots, n)$$

$$\Sigma_{j=1,n}\; r_{ij}\, x_{ij} \leq b_i \qquad (i = 1, \ldots, m)$$

$$x_{ij} \in \{0, 1\} \qquad (i = 1, \ldots, m, \; j = 1, \ldots, n)$$

$$\min \quad z$$

$$z \geq c_{ij}\, x_{ij} \qquad (i = 1, \ldots, m; \; j = 1, \ldots, n)$$

BLP Model