Knapsack Problem with Minimization Objective Function (*KP01-Min*)

Given:

n items,

P_j "profit" of item j, $j = 1, \ldots, n$ ($P_j > 0$),

W_j "weight" of item j, $j = 1, \ldots, n$ ($W_j > 0$),

one container ("knapsack") with "threshold" B:

determine a subset of the n items so as to minimize the global profit, and such that the global weight is not smaller than the knapsack threshold B.

***KP01-Min* is NP-Hard**

Mathematical Model of *KP01-Min*

$$y_j = \begin{cases} 1 & \text{if item } j \text{ is inserted in the knapsack} \\ 0 & \text{otherwise} \end{cases} \qquad (j = 1, \ldots, n)$$

$$\min \quad \sum_{j=1,n} P_j \, y_j$$

$$\sum_{j=1,n} W_j \, y_j \geq B$$

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

ILP Model (Binary Linear Programming Model)

KP01-Min **is "equivalent" to** ***KP01*****.**

KP01-Min is "equivalent" to *KP01*.

Set $y_j = 1 - x_j$ ($j = 1, \ldots, n$) and replace y_j with $1 - x_j$

1) min $T = \sum_{j=1,n} P_j\, y_j = \sum_{j=1,n} P_j\,(1 - x_j) =$

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

where $P = \sum_{j=1,n} P_j$

KP01-Min is "equivalent" to *KP01* (2).

2) $\sum_{j=1,n} W_j\, y_j = \sum_{j=1,n} W_j\,(1 - x_j) =$

$\sum_{j=1,n} W_j - \sum_{j=1,n} W_j\, x_j \geq B$

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

where $C' = \sum_{j=1,n} W_j - B$

KP01-Min is "equivalent" to *KP01* (3).

$$\text{Min } T = P - \max \sum_{j=1,n} P_j x_j$$

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

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

where: $P = \sum_{j=1,n} P_j$; $C' = \sum_{j=1,n} W_j - B$

* ***Problem KP01***

Variant of *KP01*:
Equality-KP01 (*E-KP01*)

- **Same input data as for the *KP01*.**

* **Determine a subset of the *n* items so as to maximize the global profit, and such that the global weight is equal to the knapsack capacity *C*.**

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

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

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

***E-KP01* is NP-Hard**

The *Feasibility Problem* of *E-KP01* is NP-Hard

Variant of *KP01*: *Subset Sum Problem* (*SSP*)

- **Item j has weight W_j and profit $P_j = W_j$ ($j = 1, \ldots, n$):**

  given a set of n positive numbers, select a subset of numbers so as to maximize the global sum, not exceeding a given value C.

- **Cut of metal planks with minimization of the waste.**

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

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

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

SSP* is NP-Hard. *SSP* is a special case of *KP01

Variant of *KP01*:
Change Making Problem (CMP)

- Given *n banknotes* and *a cheque (check)*,

* W_j is the *value* of banknote j ($j = 1, \ldots, n$), with $W_j > 0$,

- C is the value of the cheque:
- select a *minimum cardinality* subset of banknotes so that the global value is equal to C.

$$\min \sum_{j=1,n} x_j$$

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

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

CMP is NP-Hard (its *Feasibility Problem* is NP-Hard)

Variant of *KP01*:
Two-Constrained KP (*2C-KP*)

Given:

n **items,**

P_j **"profit" of item *j*, *j* = 1, …, *n*** **(P_j > 0),**

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

V_j **"volume" of item *j*, *j* = 1, …, *n*** **(V_j > 0),**

one **container ("knapsack") with:**

*** "weight capacity" *C*, and "volume capacity" *D*:**

determine a subset of the *n* items so as to maximize the global profit, and such that the global weight is not greater than the weight capacity *C* *and* the global volume is not greater than the volume capacity *D*.

2C-KP **is NP-Hard**

Mathematical Model for 2C-KP

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

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

$$\sum_{j=1,n} V_j \, x_j \le D$$

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

(BLP Model)

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.**

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

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

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

•

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

***MCKP* is NP-Hard**

BLP Model for *MCKP* (3)

* Define the *Binary Matrix* A_{hj} (h = 1, …, k; j = 1, …, 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

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

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

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

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

Variant of *KP01*: *Bounded-KP* (*BKP*)

In addition to the input data for KP01:

* d_j = number of available items of item-type j ($j = 1, \ldots, n$)

- x_j = number of items selected for item-type j ($j = 1, \ldots, n$)

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

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

$$0 \leq x_j \leq d_j \quad \text{INTEGER} \quad (j = 1, \ldots, n)$$

ILP Model; *BKP* is NP-Hard

Variant of *KP01*: *Unbounded-KP* (*UKP*)

No limit on the number of items available for each item-type

- x_j = number of items selected for item-type j ($j = 1, \ldots, n$)

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

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

$$x_j \geq 0 \qquad \text{INTEGER} \quad (j = 1, \ldots, n)$$

ILP Model

UKP **is NP-Hard**

Variant of *KP01*: *Unbounded-KP* (*UKP*)

No limit on the number of items available for each item-type ($d_j =$, $j = \infty 1, \ldots, n$)

- x_j = number of items selected for item-type j ($j = 1, \ldots, n$)

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

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

$$x_j \geq 0 \qquad \text{INTEGER} \quad (j = 1, \ldots, n)$$

UKP is a special case of *BKP*: $d_j = \mathit{int}\,(C / W_j)$ $j = 1, \ldots, n$

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 ($j = 1, \ldots, n$)

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

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

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

MKP01 (3)

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

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

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

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

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

Mathematical Model of MKP01

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

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

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

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

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

BLP Model

MKP01 is NP-Hard

Bin Packing Problem (BPP)

Given:

n items;

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

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.*

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

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

Bin Packing Problem (BPP)

Given:

n **items;**

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

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)

(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)

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

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

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

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

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

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

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

Alternative Models of BPP

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

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

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

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

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

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

$(2m + n)$ constraints

Alternative Models of BPP (2)

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

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

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

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

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

$(m + n)$ constraints

Alternative Models of BPP (3)

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

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

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

$\Sigma_{j=1,n} x_{ij} \le M y_i$ $(i = 1, \dots, m)$ $M \ge n$

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

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

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

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

Linear Relaxation of Model (M1)

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

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

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

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

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

Linear Relaxation of Model (M1)

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

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

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

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

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

Linear Relaxation of Model (M1)

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

- ***Objective Function:***

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

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 is not greater than the corresponding available resource.

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 is not greater than the corresponding available resource.

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; ($i = 1, \dots, m$; $j = 1, \dots, n$)

Mathematical Model of *GAP*

- **Objective function (minimum cost)**
  min $\Sigma_{i=1,m} \Sigma_{j=1,n} \, c_{ij} \, x_{ij}$

- **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

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 ($i = 1, \ldots, n$; $j = 1, \ldots, n$);

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

$b_i = 1$ amount of resource available for machine i ($i = 1, \ldots, n$).

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

Mathematical Model of *GAP*

- **Objective function (minimum cost)**
  min $\Sigma_{i=1,n} \Sigma_{j=1,n} c_{ij} x_{ij}$

- **One machine assigned to each job:**
  $\Sigma_{i=1,n} x_{ij} = 1 \quad (j = 1, \ldots, n)$

- **Resource utilized for each machine:**
  $\Sigma_{j=1,n} x_{ij} \leq 1 \quad (i = 1, \ldots, n)$

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

BLP Model

Mathematical Model of *GAP*

- **Objective function (minimum cost)**

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

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

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

- **Resource utilized for each machine:**

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

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

LP Model