bin packing problem :-

Let: m be the number of items to be loaded.
n " " " " bins.

Let: p_i $i = 1, \dots m$ be the weight of item i.
q_j $j = 1, \dots n$ " " capacity " " j.

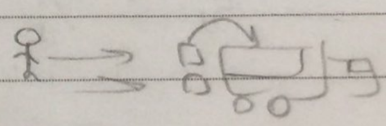
items cannot be split.

objective
$\Rightarrow$ is to (assign each item to exactly one bin)

- total weight of items in bin $j \leq q_j$
- smallest # of bins is used. $\Rightarrow$ criteria

* Two types of BPP :-

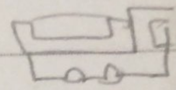
- Offline : data is all known in advance. (دعوا)
- on line : input data is given over time.



23 - 7 - 2019.

mathmatical formulation of BPP :

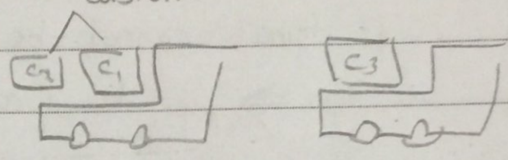
Bin



let: $y_j = \begin{cases} 1 & \text{, if bin } j \text{ is used} \\ 0 & \text{, o.w.} \end{cases}$

let: $X_{ij} = \begin{cases} 1 & \text{; if item } i \text{ is assigned to bin } j \\ 0 & \text{, o.w} \end{cases}$

customer



$$\min \sum_{j=1}^{n} y_j \quad \ldots (1)$$

$X_{11} = 1$ $\quad i = 1, 2, 3$

$x_{12} = 0$ $\quad j = 1, 2.$

$x_{21} = 0$

$x_{22} = 0$.

S.t :

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i = 1, \ldots m \quad \ldots (2)$$

$$\sum_{i=1}^{m} p_i \, x_{ij} \leq q_j \quad \forall j = 1, \ldots, n \quad \ldots (3)$$

$$x_{ij} \leq y_j \quad \forall \; i = 1, \ldots m \quad j = 1, \ldots n \quad \ldots (4)$$

$$x_{ij}, y_j \in \{0, 1\} \quad \forall$$

obj (1) : min # of bins

constraint set (2) : Each item is assigned to exactly one bin

Software.

AMPL

constraints set (3) : capacity violation

constraints set (4) : you cannot assign an item to an unopenned bin.

$x_{ij} \leq y_j$

$x_{11} \leq y_1$

$x_{21} \leq y_1$

$x_{31} \leq y_1$

Example:

ultimate objective ⇐ Decision
⇓
criterion

according to constraints ⇐

$j = 1, \ldots, 4$

$i = 1, \ldots, 100$

$p = \{3, 4, 1, 2, 3, \ldots\}$

$q = 30.$

bin اعرف كم اسطح بلل

$Y_1 = \{0, 1\}$

$Y_2 = \{0, 1\}$

$Y_3 = \{0, 1\}$

$Y_4 = \{0, 1\}$

Y (j) in (1, ... 4) : binary.

28-7-2019.

$q_r = Q$ كل الـ bins الهم نفس الـ capacity

* $\left\lceil \frac{\sum p_i}{Q} \right\rceil$ ⇒ lower bound ⇒ min # of bins

* $q_{rj} \neq Q$ ⇒ let $r = \max q_{rj}$

↳ $\left\lceil \frac{\sum p_i}{r} \right\rceil$ ⇒ lower bound

Heuristics :

⇒ On line :

- Next fit (NF) , 2-optimal
- First fit (FF) , 1.7- optimal
- Best fit (BF) , 1.7 optimal

⇒ off line :

- Net fit decreasing (NFD)
- First fit decreasing (FFD) , 1.5 - optimal
- Best fit decreasing (BFD) , 1.5 - optimal

* Let L be a list of items weights :

$L = \{p_1, p_2, \dots p_m\}$

$L = \{16, 10, 14, 12, 4, 8, 3, 12, 7, 14, 9, 6, 3\}$

⇒ $q_{rj} = Q = 20$

- aquick lower bound $= \left\lceil \frac{\sum p_i}{Q} \right\rceil = \left\lceil \frac{118}{20} \right\rceil =$

$= 6$ bins

⇒ online:

1- next fit: $\frac{16}{1}$ $\frac{10}{2}$ $\frac{14}{3}$ $\frac{12,4}{4}$ $\frac{8,3}{5}$ $\frac{12,7}{6}$ $\frac{14}{7}$ $\frac{9,6,3}{8}$

↳ allow immediate dispatch of the bin once the next bin is open.

* ما بصير تتجاوز ال capacity = 20

اذا خلّت الباية الثانية، الأولى بتطلع مباشرة وما بتستنى.

2- First Fit:

→ Initialization: place item 1 in bin 1 and remove from L

⇒ let:

$i = 2$, $j = 1$

→ iterations:

- place i in the lowest numbered bin for which it fits, if item i doesn't in any open bin, open a new bin $j+1$ & place i in it.

- Remove i from L, let i = i+1

- Stop when L is empty

16,4	10,8	14,3,3	12,7	12,6	14	9
1	2	3	4	5	6	7

7 bins

3- Best Fit :

→ Initialization: place item 1 in bin 1 and remove from L ⇒ j = 1 , i = 2

→ iteration :

- Find the bin j whose (total weight) are max, but not greater than $Q - w_j$ if item i doesn't fit any open bin ⇒ open a new bin j+1.
- i = i+1
- stop when L is empty

16,4	10,9	14,3,3	12,8	12,7	14,6
1	2	3	4	5	6

6 bins

$Q - w_j$

$20 - 16 = 4 \cancel{>} p_i = 14$

* NF → leaves once j+1 is open

* BF, FF → bin j leaves once it reaches full capacity.

29/7/2019

VRP:

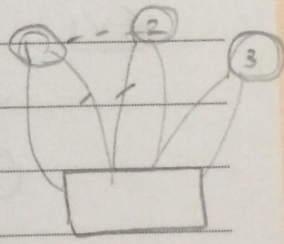
- Fleet of homogenuos vehicles.
  - ↳ Each with capacity Q
  - ↳ based at a depot

- n customers with demand q_i
- Find least cost set of vehicles tours.

Heuristics:

1- clarke-Wright saving Heuristic
2- Rout-first cluster second (RFCS) Heuristics
3- cluster-first Rout second (CFRS) Heuristics

multiple



* Clarcke-Wright Saving Heuristic:

Saving in costs of the new route
$= C_{01} + C_{20} - C_{12}$

Let T_i & T_j be two tours
end (i) is the end of tour i
begin (j) " " begining of tour j

Saving:
$S_{ij} = C_{end\,i,0} + C_{0,beg(j)} - C_{end(i)\,beg(j)}$

Scanned with CamScanner

Heuristic :

1. Generate a tour for each node i.

$S \Rightarrow$ set of tours.

2. calculate savings S_{ij} for each pair

$$S_{ij} = C_{0i} + C_{j0} - C_{ij}$$

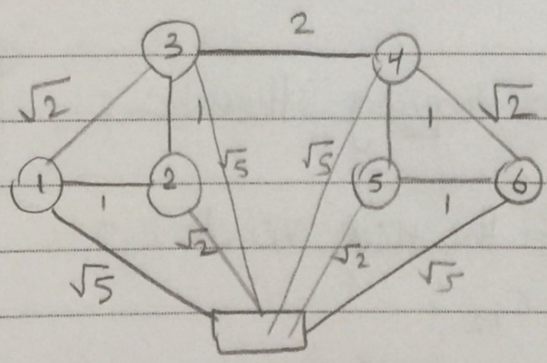
3. Sort saving in a non-increasing order.

4. Walk down the list of tours & continue as long as savings >0 & capacity allow

$$S_{ij} \geq 0$$

$$\sum q_{rk} \leq Q$$

Ex:



→ vehicle capacity 5

→ demand at customers 1, 2, 3 is 3

→ demand at customer 4, 5, 6 is 2.

أقل عدد سيارات علي احتاجها → $LB = \left\lceil \frac{15}{5} \right\rceil = 3.$

$T_1 = \{0-1-0\}$

$T_2 = \{0-2,-0\}$

$T_3 = \{0-3-0\}$

$T_4 = \{0-4-0\}$

$T_5 = \{0-5-0\}$

$T_6 = \{0-6-0\}$

$C^{6}_{2} = 15$ pairs

برتبهم وباخذ أعلى
saving
وبقارنها بال
capacity

$S_{12} = \sqrt{5} + \sqrt{2} - 1 = 2.65$

$S_{13} =$

$S_{14} =$

$S_{15} =$

$S_{16} = \sqrt{5} + \sqrt{5} - 4$

$S_{23} =$

$S_{24} =$

$S_{25} =$

$S_{26} =$

$S_{34} =$

$S_{35} =$

$S_{36} =$

$S_{45} =$

$S_{46} =$

$S_{56} =$