Unit-4

Iterative Improvement & Branch & Bound

1 Stable Matching?

Blocking pair - a pair of men & women, who are currently unmatched prefer each other over their current partner.

→ Iterative improvement starts with some feasible solution, and proceeds to try to improve the feasible solution, with small localized changes

→ When no change improves the value of the objective fn. the algorithm returns the last feasible solution & stops

Algorithm

1. Start with all the men and women being free.

2. While there are free men, arbitrarily select one of them & do the following

Proposal : The selected free man propose to the woman who is highest on his preference list, who has not rejected him before

Response : If the woman is free, she accepts the proposal.

Otherwise, she checks if she prefers this man, over her current partner. If that is the case: she breaks her marriage, and accepts the proposal from the new man.

eg 1

Men's Preferences

	1st	2nd	3rd
Bob	Lea	Ann	Sue
Jim	Lea	Sue	Ann
Tom	Sue	Lea	Ann

Women's Preferences

	1st	2nd	3rd
Ann	Jim	Tom	Bob
Lea	Tom	Bob	Jim
Sue	Jim	Tom	Bob

	Ann	Lea	Sue
Bob	$\boxed{2,3}$	1,2	3,3
Jim	3,1	1,3	$\boxed{2,1}$
Tom	3,2	$\boxed{2,1}$	1,2

Iteration 1

Free Men

Bob → Lea ✓ ✗

Jim, Tom

Jim → Lea ✗

Jim → Sue ✓

Tom

Tom → Sue ✗

Tom → Lea ✓

Bob

Bob → Sue ✗

Bob → Ann ✓

Bob → Ann

Tom → Lea

Jim → Sue

eg2

A	O	M	N	L	P
B	P	N	M	L	O
C	M	P	L	O	N
D	P	M	O	N	L
E	O	L	M	N	P

L	D	B	E	C	A
M	B	A	D	C	E
N	A	C	E	D	B
O	D	A	C	B	E
P	B	E	A	C	D

	L	M	N	O	P
A	4,5	2,2	3,1	1,2	5,3
B	4,2	3,1	2,5	5,4	1,1
C	3,4	1,4	5,2	4,3	2,4
D	5,1	2,3	4,4	3,1	1,5
E	2,3	3,5	4,3	1,5	2,2

A → O ✓

B → P ✓

C → M ✓

D → P ✓

E → O ✗

E → L ✓

Free Men

B C D E

C D E

D E

E

eg3

	A	B	C
α	1,3	2,2	3,1
β	3,1	1,3	2,2
γ	2,2	3,1	1,3

$\alpha \to A$ ✓

$\beta \to B$ ✓

$\gamma \to C$ ✓

Free men

β, γ

eg4

	A	B	C	D
α	1,3	2,3	3,2	4,3
β	1,4	4,1	3,4	2,2
γ	2,2	1,4	3,3	4,1
δ	4,1	2,2	3,1	1,4

Free men

$\alpha \to A$ ✓ ✗

$\beta \to A$ ✗

$\boxed{\beta \to D}$ ✓

$\gamma \to B$ ✓ ✗

$\delta \to D$ ✗

$\boxed{\delta \to B}$ ✓

$\boxed{\gamma \to A}$ ✓

$\alpha \to B$ ✗

$\boxed{\alpha \to C}$ ✓

$\beta \gamma \delta$

$\gamma \delta$

δ

γ

α

$\boxed{\text{max time complexity} : O(n^2)}$

* Maximum Flow

Algorithm

Set $f_{total} = 0$

Repeat until there is no path from s to t

(i) Run DFS from s to find a flow path to t

(ii) Let f be the minimum capacity value on the path.

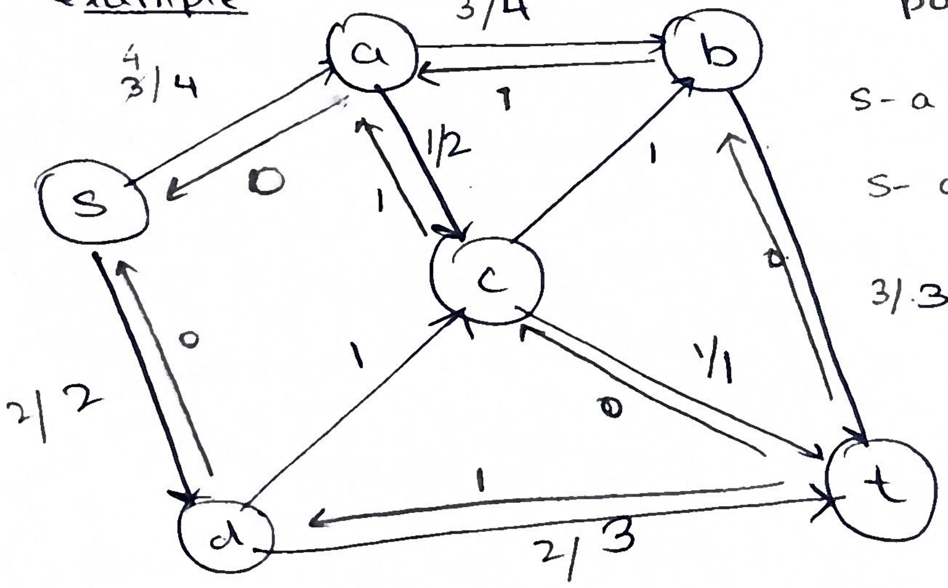
(iii) Add f to f_{total}

(iv) For each edge $u \to v$ on the path

Decrease $c(u \to v)$ by f

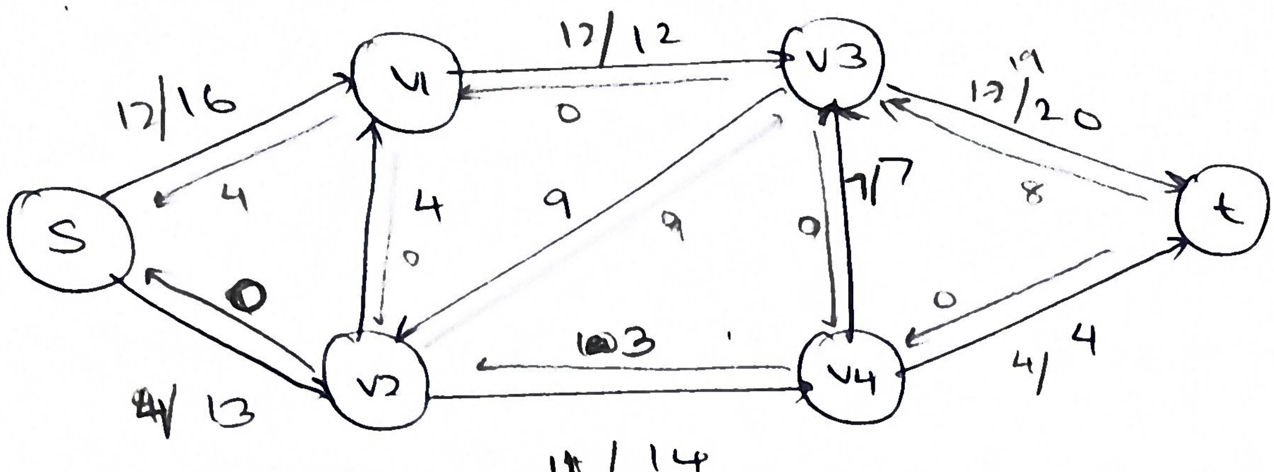
Increase $c(v \to u)$ by f

Example



	path	bottlenecks
	s-a-b-t	3
	s-d-t	2
3/3	s-a-c-t	1
		6

Example 2



13,

path	bottleneck
$S \to v_1 \to v_3 \to t$	12
$S \to v_2 - v_4 \to t$	4.
$S - v_2 - v_4 \to v_3 - t$	7

Time Complexity: $O(|E| * f)$

* Maximum matching in bipartite graphs

* Branch and bound using knapsack

Constructing a state space tree

Termination occurs when:

(i) The value of the node is not better than the best soln. seen so far

(ii) Constraints are violated

(iii) If no further choices can be made - compare value of objective fn for this feasible solution, with the best solution seen so far. update if solution is better.

Steps: sort by value by wt. ratio

a branch to the left $\Rightarrow$ inclusion

right $\Rightarrow$ exclusion

$O(2^n)$

computing upper bound

$$ub = \begin{cases} v + (W - w)(v_{i+1}/w_{i+1}) & 0 \le i < n \\ v & i = n \end{cases}$$

Example 1

item	weight	value	value/weight
1	4	40	10
2	7	42	6
3	5	25	5
4	3	12	4.

$W = 10$

(i) $0 + (10)(10) = 100$

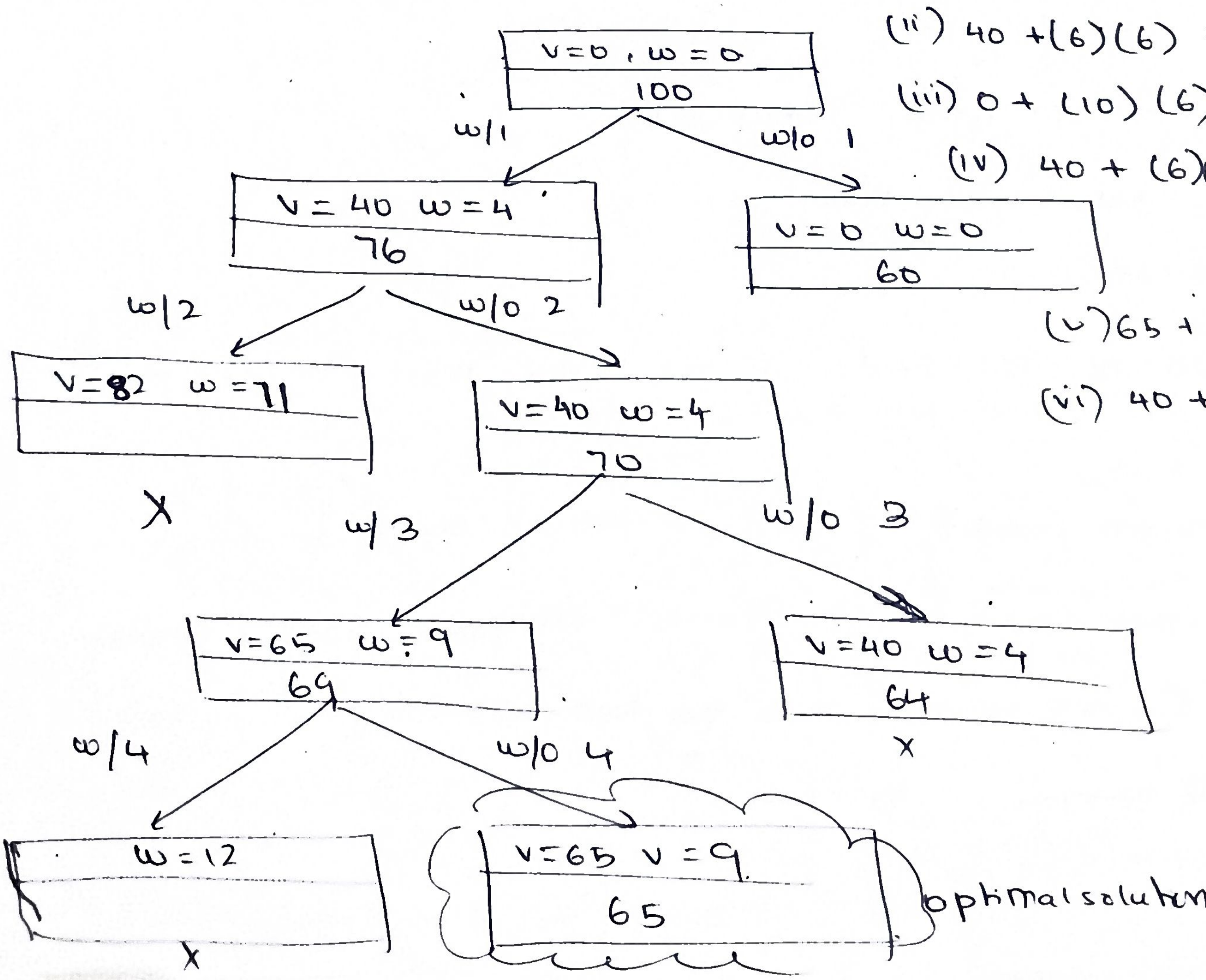
(ii) $40 + (6)(6) = 76$

(iii) $0 + (10)(6) = 60$

(iv) $40 + (6)(5) = 70$

(v) $65 + (1)(4) = 69$

(vi) $40 + (6)(4) = 64$



Example 2

item	v_i	w_i	v_i/w_i
1	10	2	5
2	10	4	2.5
3	12	6	2
4	18	9	2

$W = 15$

$v + (W - w)(v_{i+1}/w_{i+1})$

(i) $0 + (15)(5)$

(ii) $10 + (13)(2.5) = 42.5$

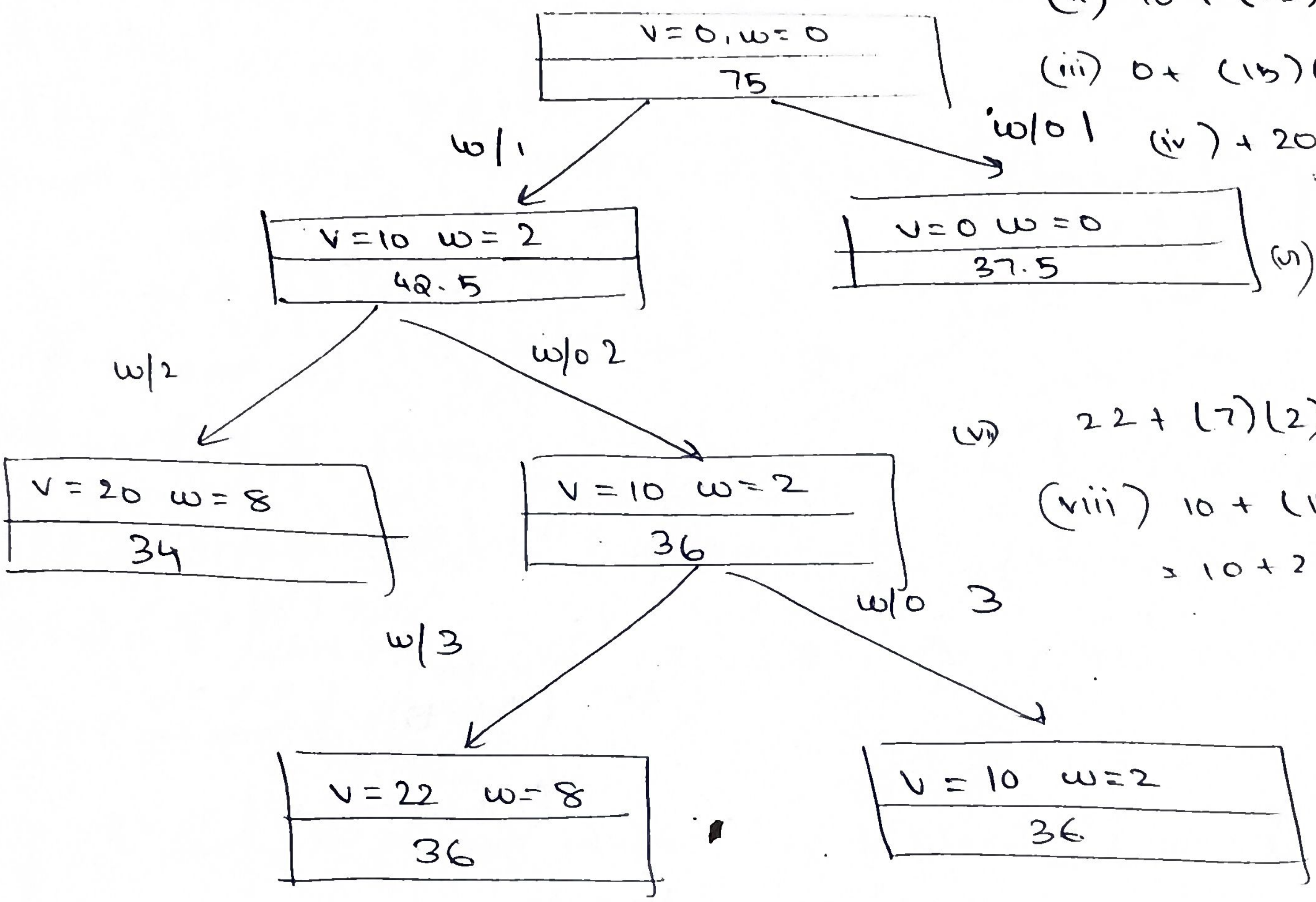
(iii) $0 + (15)(2.5)$

(iv) $+ 20 + (7)(2) = 34$

(v) $10 + (13)(2) = 36$

(vi) $22 + (7)(2) = 36$

(viii) $10 + (13)(2)$
$= 10 + 26 = 36$

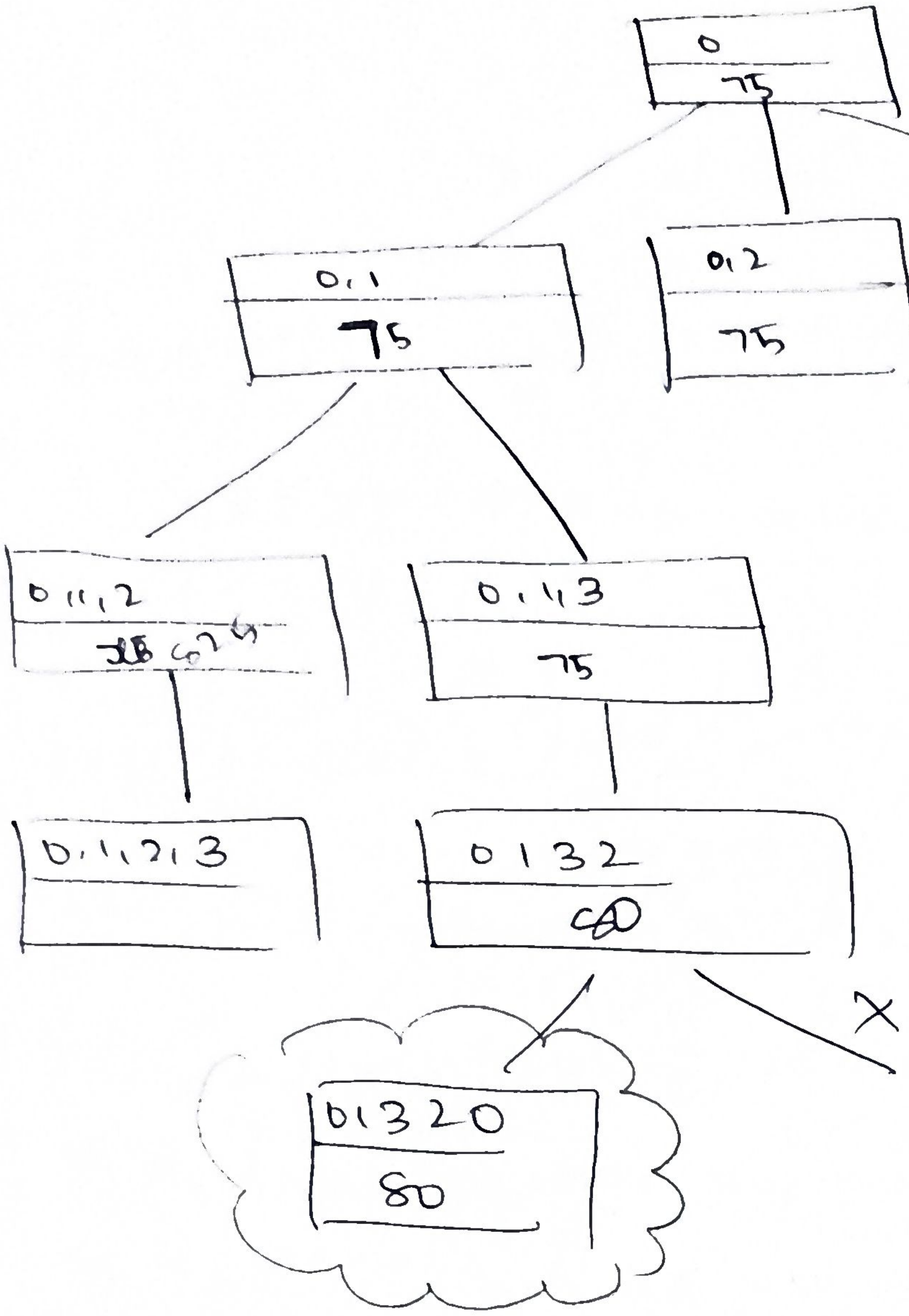


* Travelling Salesman Problem

Find the lower bound : for each city i, $1 \le i \le n$, find the sum s_i of the distances from city i to the 2 nearest cities, compute the sum of the numbers, divide the result by 2

$$lb = \lceil s/2 \rceil$$

If a particular edge has to be included, then the lb should be computed with the inclusion of the required edges.



$lb = (10+15) + (10+25)$
$+ (15+30) + (20+25)$

$(0,1) \Rightarrow lb = (10+15)$
$+ (10+25)$
$+ (15+30)$
$+ (20+25)$

$(0,2) = (10+15) + (10+25)$
$+ (15+30) + (20+25)$
$= 75$

$(0,3) = (10+20) +$
$(10+25) + (15+30)$
$+ (20+25)$

$lb = (10+15) + (10+35)$
$+ (15+30) + (20+25$

$lb = (\overset{10+15}{\cancel{10+20}}) + (\overset{10+35}{\cancel{25+35}}) +$
$(15+30) + (25+20)$

$lb = (10+15) + (10+35)$