# Branch and Bound Examples

Katta G. Murty Lecture slides

TSP: $c$ = cost matrix. Problem said to be:

**Symmetric TSP:** if $c_{ij} = c_{ji} \quad \forall i, j$.

**Assymmetric TSP:** if $c$ not symmetric.

**Euclidean TSP:** If $c$ satisfies triangle inequality. Here cities can be represebted by points in the $2 - d$ plane.

**Lower bounding based on assignment relaxation:** The first LB strategy proposed for the TSP in our paper in the early 1960's. Can solve problems with upto 50 cities within reasonable computer time.

**Fathoming strategy:** Check whether the *Relaxed optimum assignment (ROA)* is a tour.

EXAMPLE: $n = 6$, $c =$

| To | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| From 1 | × | 27 | 43 | 16 | 30 | 26 |
| 2 | 7 | × | 16 | 1 | 30 | 25 |
| 3 | 20 | 13 | × | 35 | 5 | 0 |
| 4 | 21 | 16 | 25 | × | 18 | 18 |
| 5 | 12 | 46 | 27 | 48 | × | 5 |
| 6 | 23 | 5 | 5 | 9 | 5 | × |

Original problem ROA $\{(1,4),(2,1),(3,5),(4,2),(5,6),(6,3)\}$, LB = 54

Final reduced cost matrix $c_0$

| To | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| From 1 | × | 7 | 23 | 0 | 10 | 11 |
| 2 | 0 | × | 11 | 0 | 25 | 25 |
| 3 | 13 | 8 | × | 34 | 0 | 0 |
| 4 | 3 | 0 | 9 | × | 2 | 7 |
| 5 | 0 | 36 | 17 | 42 | × | 0 |
| 6 | 16 | 0 | 0 | 8 | 0 | × |

For all assignments $x$, we have $\quad z_c(x) = 54 + z_{c_0}(x)$.

Using this, we never use the cost matrix $c$ again. For children of the original problem, we use the cost matrix $c_0$.

Other details of this B & B for TSP will be discussed in class.

General MIP: Use LP relaxation for lower bounding. Let RO (relaxed optimum) denote an optimum sol. obtained for LP relaxation.

Select an integer variable with a fractional value in the RO as the BV. Usually, penalties computed from opt. simplex tableau are used for the selection.

EXAMPLE: RO for following MIP is: $(y = (3/2, 5/2)^T, x = (4, 0, 0, 0)^T)$. Original problem not fathomed. We select $y_2$ with a value of $5/2$ in RO as the BV.

| $y_1$ | $y_2$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $-z$ | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | $-2$ | 1 | 0 | 3/2 |
| 0 | 1 | 0 | 2 | 1 | $-1$ | 0 | 5/2 |
| 0 | 0 | 1 | $-1$ | 1 | 1 | 0 | 4 |
| 0 | 0 | 0 | 3 | 4 | 5 | 1 | $-20$ |

$y_1, y_2 \geq 0$ & integer; $x_1$ to $x_4 \geq 0$; min $z$ .

0-1 Knapsack problem: Use LP rexation as LB strategy.

In any CP here, for that CP delete all objects with weight $>$ remaining knapsack's weight capacity from consideration (i.e., set corresponding variable $= 0$).

Here LP relaxation can be solved by a simple special rule. It consists of loading knapsack with objects in decreasing order of density ($=$ value/weight) until at some stage one of the following two events occurs.

*knapsacks capacity fully used up exactly:* In this case, the set of objects loaded into knapsack at this stage, is the optimum set of objects to be loaded. This CP is fathomed.

*Knapsack has positive capacity remaining, but it is $<$ weight of next object to be loaded:* Make value of corresponding variable $=$ (remaining knapsacks weight capacity)/(weight of that object); and make variables corresponding to all remaining objects $= 0$. This gives an RO.

If a CP is not fathomed, exactly one variable has a fractional value in the RO, that variable could be selected as the BV for branching this CP.

| $j$ | Weight $w_j$ | Value $v_j$ | Density $v_j/w_j$ |
|---|---|---|---|
| 1 | 3 | 21 | 7 |
| 2 | 4 | 24 | 6 |
| 3 | 3 | 12 | 4 |
| 4 | 21 | 168 | 8 |
| 5 | 15 | 135 | 9 |
| 6 | 13 | 26 | 2 |
| 7 | 16 | 192 | 12 |
| 8 | 20 | 200 | 10 |
| 9 | 40 | 800 | 20 |

capacity $w_0 = 35$

The asymmetric assignment problem: $c_{n \times n} = (c_{ij})$ is cost matrix . Need a min cost asymmetric assignment, i.e., one in which $\forall i, j \quad x_{ij} = 1 \Rightarrow x_{ji} = 0$ (this automatically $\Rightarrow x_{ii} = 0 \forall i$).

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 4 | 15 | 10 | 14 | 13 | 20 |
| 2 | 15 | × | 16 | 18 | 18 | 8 |
| 3 | 10 | 16 | 34 | 28 | 25 | 24 |
| 4 | 14 | 18 | 28 | × | 3 | 17 |
| 5 | 13 | 18 | 25 | 3 | × | 13 |
| 6 | 20 | 8 | 24 | 17 | 13 | 30 |

The symmetric assignment problem: $n = 2p$ objects.

Need to form these objects into $p$ couples each with 2 objects.

For $j > i$ $\quad d_{ij}$ = cost of forming objects $i, j$ into a couple.

| $j =$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $i = 1$ | × | 4 | 6 | 110 | 116 | 126 | 118 | 120 | 116 | 114 |
| 2 | | × | 8 | 118 | 114 | 124 | 106 | 102 | 118 | 106 |
| 3 | | | × | 112 | 116 | 112 | 110 | 122 | 116 | 124 |
| 4 | | | | × | 2 | 4 | 218 | 216 | 226 | 230 |
| 5 | | | | | × | 6 | 212 | 212 | 234 | 232 |
| 6 | | | | | | × | 338 | 306 | 316 | 308 |
| 7 | | | | | | | × | 4 | 2 | 16 |
| 8 | | | | | | | | × | 6 | 8 |
| 9 | | | | | | | | | × | 10 |
| 10 | | | | | | | | | | × |

THis is a min cost perfect matching problem.

B & B for pure 0-1 IP

Called **Implicit enumeration methods**.

Consider $\min z = cx$, s. to $Ax \leq b$ $x_j$ binary $\forall j$.

**0-variable:** One fixed at 0 in a CP

**1-variable:** One fixed at 1 in a CP

**Free variable:** One not fixed in a cP

**Partial sol.:** $(U_0, U_1, U_f)$ – in this all $x_j \in U_0$ are fixed at 0, all $x_j \in U_1$ are fixed at 1; while all $x_j \in U_f$ are free. Each CP in algo. corresponds to a partial sol.

A **completion** of a partial sol. is obtained by giving 0-1 values to free variables.

If $c_j < 0$ substitute $x_j = 1 - y_j$. $y_j$ is also 0-1. With this we can assume $c \geq 0$.

## Simple fathoming criterion for CP $(U_0, U_1, U_f)$

Remaining problem is: $\min c_f x_f + (\Sigma_{j \in U_1} c_j)$ s. to $A_f x_f \leq b' = b - \Sigma_{j \in U_1} A_{.j}$ $x_f$ is 0-1.

If $b' \geq 0$, $x_f = 0$ is an opt. sol. for this problem. This is fathoming criterion. If satisfied, complete by making all free vars. into 0-vars, gives opt. sol. for CP.

## Computations to be carried out on the CP $(U_0, U_1, U_f)$ if it's not fathomed

Let $\bar{z}$ = incumbent obj. value at this stage. To avoid solving the LP relaxation, they apply several 0-1 feasibility tests to see if this CP can have a 0-1 sol. with obj. val. $< \bar{z}$, i.e. check feasibility of:

$$\begin{aligned} \sum_{j \in U_f} A_{.j} x_j &\leq b' = b - \sum_{j \in U_1} A_{.j} \\ \sum_{j \in U_f} a_{m+1,j} x_j &\leq b'_{m+1} = \bar{z} - \sum_{j \in U_1} a_{m+1,j} \\ x_j &= 0 \text{ or } 1 \ \forall j \in U_f \end{aligned}$$

where $a_{m+1,j} = c_j \ \ \forall j$.

An example test: If $\Sigma_{j \in U_f} \min\{a_{ij}, 0\} > b'_i$ for any $i$ above system infeasible, prune this CP. Try $3x_1 + 4x_2 - 4x_3 - 6x_4 \leq -11$.

Other tests: determine that a free var. is a 0-var. or 1-var. at all feasible sols. of system. For example, if $k \in U_f$ & $\Sigma_{j \in U_f} \min\{a_{ij}, 0\} + |a_{ik}| > b'_i$ for any $i$ and $a_{ik} < 0$ $[a_{ik} > 0]$ then $x_k$ is a 1-var. [0-var.]. Try $3x_1 + 4x_2 - 4x_3 - 6x_4 \leq -9$. Both $x_3, x_4$ are 1-vars. in every feasible sol. Alter sets of CP accordingly.

Using Surrogate constraints: For above system, it is a nonnegative comb. of constraints, i.e., $\Sigma_{j \in U_f} (\Sigma_{i=1}^{m+1} \mu_i a_{ij}) x_j \leq \Sigma_{i=1}^{m+1} \mu_i b'_i$, where $\mu_i \geq 0 \quad \forall i$. For example:

$$\begin{aligned} x_1 - x_2 &\leq -1 \\ -x_1 + 2x_2 &\leq -1 \end{aligned}$$

Taking $\mu = (1, 1)$ leads to the surrogate constraint $x_2 \leq -2$.

From each of the 2 constraints in above system, we cannot conclude that system has no 0-1 sol., but from surrogate constraint we can. Apply all tests on surrogate constraint.

Best $\mu$ comes from dual opt. sol. of LP relaxation.

Using surrogate based 0-1 Knapsack model: One relaxation is: $\min \Sigma_{j \in U_f} c_j x_j$ s. to $\Sigma_{j \in U_f} (\Sigma_{i=1}^{m} \mu_i a_{ij}) x_j \leq \Sigma_{i=1}^{m} \mu_i b_i'$ $x_{U_f}$ binary.

$\Sigma_{j \in U_1} c_j$ + min obj. val. in this 0-1 knapsack problem is an LB for this CP. Best $\mu$ is negative opt. dual sol. of LP relaxation. Prune if LB can be shown to be $\geq$ cost of incumbent.

After all tests etc. if CP not pruned, and is: $(U_0', U_1', U_f')$, then can take $\Sigma_{j \in U_1'} c_j$ as a LB for it.

Normally use backtrack search. CP to explore next is selected from LIST (maintained as an ordered list) by LIFO. It can be branched using a free var. in it as BV.