4.1

The Branch and Bound Approach

Katta G. Murty Lecture slides

Assume original problem minimization problem. Let K_0 = its set of feasible solutions.

During B&B K_0 is partitioned into many simpler subsets, each subset is set of feasible sols. of a problem called a Candidate Problem or CP.

Each CP is the original problem, augmented with additional constraints called Branching Constraints.

Branching constraints are simple constraints generated by an operation called Branching.

Whenever a new CP is generated, an

LB = Lower Bound for min. obj. value in it

is computed by a procedure called Lower bounding strategy.

For some CPs, the LB strategy may actually produce a minimum cost feasible sol. in it. In this case, that CP is said to be Fathomed, it need not be processed any further, so is taken out from further consideration.

Among the optimum solutions of fathomed CPs, the best is called the incumbent at this stage, and it is stored and updated. So, the objective value of incumbent is an Upper Bound for the min obj. value in original problem.

The incumbent and upper bound change whenever a new and better feasible sol. appears in method due to fathoming.

In each stage, method selects one CP to examine, called Current CP.

- If LB for current CP $\geq$ current Upper Bound, this CP is Pruned, i.e., discarded. The Partial enumeration property of method comes from this.
- Otherwise, set of feasible solutions of this CP is partitioned into 2 or more subsets by applying branching strategy on it.

Main Steps in B&B

Bounding: B&B uses both :

Upper Bound for min objective value in original problem: Changes whenever incumbent does, and decreases when it changes.

Lower Bound for min obj. value in each CP: Calculated by applying LB strategy on it.

Pruning: Deleting some CPs from further consideration. A CP is pruned

- if its LB $\geq$ Current UB
- if it is fathomed
- if it is found infeasible

Branching: This operation on a CP (Called Parent Node, generates two or more new CPs (called its Children).

The various steps:

The LB strategy: Most commonly used LB strategy is based on solving a relaxed problem.

To find LB for a CP, this strategy relaxes (i.e., ignores) difficult constraints in it until remaining problem can be solved by an efficient algo. Opt. sol. of relaxed problem called Relaxed Optimum. Objective value of relaxed opt. is a LB for the CP.

Fathoming Criterion: If relaxed opt. satisfies the relaxed constraints,it is in fact an opt. sol. for that CP, in this case the CP is Fathomed. Thus lower bounding based on relaxation lends itself very easily to fathoming.

Quality of Lower Bounds: The higher the quality of LB (i.e., higher it is, or closer it is to the min obj. value) the more pruning, thus reducing no. of CPs algo. has to examine. In designing an LB strategy, need strike a balance between: (1) Quality of LB (Higher the better), and (2) Computational effort needed to get it (Smaller the better).

Examples: TSP

0–1 Knapsack

Pure 0–1 IP

MIP.

When a problem can be modeled as IP in different ways, the LB obtained by LP relaxations of different formulations may differ. That leading to highest LB is best.

Lagrangian Relaxation This LB technique does not relax integer requirements, but relaxes some of the linear constraints. Suppose a CP is: $\min z(x)$ s. to $f_p(x) = 0$ $p = 1$ to k, $g_i(x) \geq 0$ $i = 1$ to r and $x \in X$.

Here $x \in X$ includes integer requirements, & all other linear constraints not to be relaxed. The 1st $k + r$ constraints are to be relaxed.

Relaxed problem is: $\min L(x, u, v) = z(x) - \Sigma_{p=1}^{k} u_p f_p(x) - \Sigma_{i=1}^{r} v_i g_i(x)$ s. to $x \in X$.

Linear constraints to relax choosen to make sure relaxed problem very easy to solve.

Theorem: For any $u = (u_1, \ldots, u_k)$ & $v = (v_1, \ldots, v_r) \geq 0$ let $\bar{x}(u, v)$ be relaxed opt and $L(\bar{x}, u, v)$ the opt. obj. value in relaxed prob. Then $L(\bar{x}, u, v)$ is an LB for min obj. value in CP.

How to choose Multiplier vectors in Lagrangian relaxation?

LB $= L(\bar{x}, u, v) = L(u, v)$ say. Select (u, v) to $\max L(u, v)$ s. to $v \geq 0$.

For some problems, an NLP technique called **Subgradient Opt. Algo.** helps to get very good solutions to this very quickly. On those, Lagrangian relaxation provides excellent LB strategy.

Example: Constrained assignment. Consider assigning n salesman to n zones. c_{ij}, s_{ij} = cost, sales volume by assigning salesman i to zone j. g = min goal for total sales generated. Problem is:

$$\begin{aligned}
&\min \sum\sum c_{ij}x_{ij} \\
&\text{s. to} \quad \sum_j x_{ij} = 1, \quad \forall i \qquad (1) \\
&\qquad \sum_i x_{ij} = 1, \quad \forall j \qquad (2) \\
&\qquad \sum\sum s_{ij}x_{ij} \geq g \qquad (3) \\
&\qquad 0 \leq x_{ij} \leq 1 \quad \forall i, j \qquad (4) \\
&\qquad x_{ij} \quad \text{integer } \forall i, j \qquad (5)
\end{aligned}$$

LP relaxation would relax (5)

Lagrangian relaxation cannot relax (5), but can relax any or all of (1), (2), (3).

LR Strategy 1: Relax (3)

LR Strategy 2: Relax (1) and (3)

Numerical example: $g = 90, c = \begin{pmatrix} 13 & 19 & 25 \\ 12 & 17 & 11 \\ 9 & 13 & 6 \end{pmatrix}, S = \begin{pmatrix} 15 & 25 & 17 \\ 30 & 10 & 40 \\ 50 & 70 & 80 \end{pmatrix}$

Bounding using Cuts

K = feasible space of LP relaxation.

Valid ineq. One which does not eliminate any integer feasible sols. when added to LP relaxation.

Facetal ineq. One that leads to a facet of integer hull.

Derive some facetal ineq. & add to LP relaxation, leading to tighter bounds. Additional facetal ineqs. for CPs could be added in later stages. B & B based on this strategy called **Branch and cut**.

The Braching Strategy:

Usually carried out by selecting a Branching Variable, one that is likely to make LBs for children as high as possible.

If branching variable is a 0–1 variable x_1, branching constraints are:

If branching variable is an integer variable x_1 whose value in present relaxed optimum is the nonintegral $\bar{x}_1$, branching constraints are:

1 Union of sets of feasible solutions of child problems is always the set of feasible solutions of parent.

2 Every child always inherits all branching constraints in its parent. So always, LB for child $\geq$ LB for parent.

Branching constraints, BVs selected to make LB for children as high as possible. One technique uses **evaluation coeffs.** for each potential BV for this selection.

The Search Strategy:

List refers to the set of all unexplored CPs in the present stage, i.e., set of all Live nodes, those not yet branched, fathomed or pruned.

One strategy picks current CP to branch to be the one in list with least lower bound.

Another is a backtrack search strategy based on depth first search.

Search terminates when list becomes $\emptyset$. Incumbent then is an optimum solution.