→ we can compare node's bound value with the value of best solution seen so far.

→ if the b value is not better than best solution seen so far - ie; not smaller for a min problem & not larger for max problem. - the node is non-promi. & can be terminated because no solution obtained from it can yield a better solution than are already available.

Reason for terminating a search path at the current node in state - space tree:-

* The value of node's bound is not better than the value of best solution seen so far.

* The node represent's no feasible solution because the constraints of problem are already violated.

* The subset of feasible solution represented by node consists of single point.

eg:-

→ Assignment problem:-

In this, n agents are to be assigned for n-tasks, each agent having exactly one task to perform. The problem is to assign agents to tasks so as to minimize the total cost of executing the n-tasks.

weight not exceeding W. Our aim is to fill the knapsack in a way that max the value of included objects, while respecting the capcity constraint

→ Travelling Sales person problem:-

find shortest tour through 'n' citis with known positive integer distances b/w them.

Algorithm:-

```
Roodus
Procedure B&B ()
begin
  E: node pointer;
  E: new(node)
  H: heap;
while (true) do
if (E is a final leaf) then
print out the path from E to root;
return;
end if
Expand(E);
if (H is empty) then
  report that there is no solution
  return
end if
E: = delete_top (H);
```