Branch & Bound–algoritmer

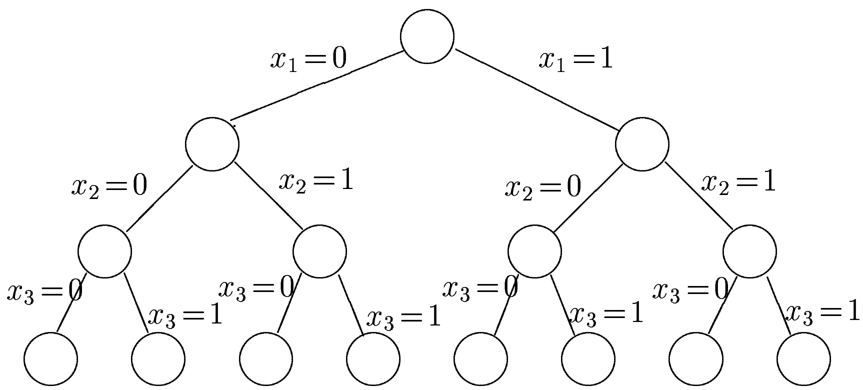
Generell princip för att finna optimala lösningar till optimeringsproblem med heltalskrav. Stor frihetsgrad gör att den kan anpassas till olika typer av modeller; kan även kombineras med andra (ev. icke-optimerande) algoritmer. (Synonymer: trädsökning, implicit uppräkning.)

$$\text{(P)} \qquad z^* = \min \; c^T x \quad \text{då} \; x \in X$$

Idé: Räkna upp alla tillåtna lösningar genom att successivt partitionera X i en familj av delmängder. Uppräkningen organiseras i ett träd med hjälp av grafsökning, och görs implicit genom att man utnyttjar uppskattningar av z^*; dessa fås från lösningar till relaxeringar av (P).

B&B har en exponentiell värsta falls–komplexitet, ty det kan inträffa att samtliga tillåtna lösningar räknas upp explicit.

Exempel: Partition via förgrening: $X = \{0,1\}^3$



Fyra centrala begrepp

- **Relaxering:** en förenkling av (P) där vissa bivillkor tas bort.
  - **Syfte:** att erhålla enkla (polynomiskt lösbara) subproblem, samt optimistiska uppskattningar av z^*.
- **Förgreningsstrategi:** regler för hur en delmängd av X skall partitioneras.
  - **Syfte:** att utesluta lösningen till en relaxering ifall den ej är tillåten i (P); att närma oss en tillåten lösning.
- **Trädsökningsstrategi:** definierar den ordning i vilken noderna i B&B–trädet skapas och avsöks.
  - **Syfte:** att snabbt finna bra tillåtna lösningar, begränsa trädets storlek.
- **Nodkapningskriterier:** regler för när en delmängd av X ej skall partitioneras vidare.
  - **Syfte:** att undvika att avsöka delar av trädet som ej kan innehålla en kandidat till optimallösning.

Dessa fyra begrepp måste specificeras för att en komplett B&B–algoritm skall vara definierad, d.v.s:

Förenkla **F**örgrena **F**ortsätta **F**örkasta

Relaxeringar

Exempel på relaxeringar av (P):

- Tag bort heltalskrav $\Rightarrow$ linjärt kontinuerligt subproblem.
- Tag bort komplicerande bivillkor
- Lagrange–relaxera komplicerande bivillkor

Om optimallösningen till det relaxerade problemet ej är tillåten i (P) söker vi uppfylla de relaxerade bivillkoren m.h.a. en förgrening i B&B–trädet.

För att få pessimistiska uppskattningar av z^* (motsvarande tillåtna lösningar till (P)) används ofta heuristiker.

Det optimala värdet för ett subproblem ger en optimistisk uppskattning av det värde som kan nås i grenen under motsvarande nod i trädet, ty varje förgrening tillför restriktioner.

Förenklingen av (P) genom relaxeringen får inte vara **för** stor; då krävs troligen väldigt många förgreningar innan tillåtenhet åstadkommes, d.v.s., ett stort, djupt B&B–träd.

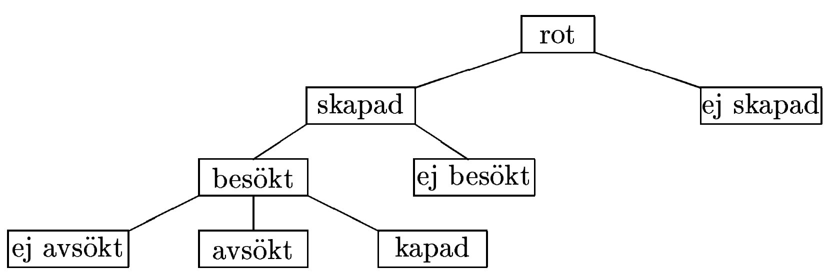
Förgreningar

Förgrening används för att stärka en relaxering. Helst ska förgreningen motsvara en partition av den aktuella delmängden av X, så att varje tillåten lösning till (P) återfinns på exakt en plats i trädet (äkta uppdelning).

Förgreningen måste innebära att lösningen som erhållits i subproblemet utesluts; dock får ingen tillåten lösning till (P) uteslutas.

Trädsökning

B&B-trädet är inte givet på förhand, utan skapas under algoritmens gång. För att konstruera trädet och avsöka dess noder används grafsökning.



Förgreningsprioritet: anger om efterföljarna skapas simultant eller ej, och i vilken ordning.

- Djup–först–ordning: endast en av efterföljarna skapas.
- Bredd–först–ordning: alla efterföljare skapas på en gång.

Besöksprioritet: anger vilken av de aktiva (d.v.s., skapade men ej besökta eller avsökta) noderna som skall besökas härnäst. Hybrider finns, men i huvudsak används följande två:

- Djup–först–ordning i både förgrening och besök. Ger snabbt tillåtna lösningar; djupa, smala träd.
- Bäst–först–ordning: välj den aktiva nod som motsvarar den bästa optimistiska uppskattningen av z^*. Vi tror nämligen att optimum finns i just den grenen. Kräver mycket minne; ger minst träd.

Nodkapning

Om lösningen x^R till ett relaxerat problem inte är tillåten i (P), men har ett målfunktionsvärde z^R som är lägre (min-problem) än målfunktionsvärdet $\overline{z}$ (pessimistisk uppskattning) för den bästa kända tillåtna lösningen till (P), kan nod R **inte** kapas.

Kapningsregler: kapa en nod R i B&B–trädet ifall motsvarande subproblem

- saknar lösning,
- har en optimallösning som är tillåten i (P) $\Rightarrow$ kandidat till optimal lösning (pessimistisk uppskattning), eller
- har ett optimalt målfunktionsvärde, $z^R \geq \overline{z}$, där $\overline{z}$ är målfunktionsvärdet hos någon känd tillåten lösning.

I inget av dessa fall kan en tillåten lösning till (P), med ett lägre målfunktionsvärde (min-problem) än det bästa kända (pessimistisk uppskattning), återfinnas efter en ytterligare förgrening i nod R.

Lös nedanstående heltalsproblem med Branch & Bound–algoritmen

$$
\begin{array}{rcrcrcll}
\max & z = & 5x_1 & + & 3x_2 & & & \\
\text{då} & & 2x_1 & - & x_2 & \geq & 2 & \quad (1) \\
 & & -x_1 & + & 3x_2 & \geq & 3 & \quad (2) \\
 & & 5x_1 & + & 6x_2 & \leq & 60 & \quad (3) \\
 & & & & x_1, x_2 & \geq & 0, \text{ heltal} &
\end{array}
$$