Lecture 5 - Introduction to Combinatorial Optimization & Modelling

Fair & Optimal Deision Trees (ML) - construct the best decision tree based on historical data

Combinatorial Optimization Problem: $\min F(X) \leftarrow$ objective function

Solution $\rightarrow X \in C \subseteq \mathbb{N}^n$

Set of feasible solutions, implicitly defined through constraints

1) Each node must be assigned shape: $X_{\{node, shape\}} \in \{0, 1\}$

$\rightarrow$ There is a circle on node T: $X_{\{T, circle\}} = 1$, $X_{\{T, star\}} = 0$, $X_{\{T, cane\}} = 0$

2) Garlands always connect two of the same type of ornaments.

$(n_1, n_2) \in edges(garland)$, $s \in shapes: X_{\{n_1, s\}} \leftrightarrow X_{\{n_2, s\}}$

3) Lights are always connected to a circle at one or the other end.

$\forall (n_1, n_2) \in edges(lights): X_{\{n_1, circle\}} \vee X_{\{n_2, circle\}}$

Modelling

Variables	Constraints	Objective function
(decisions we can make)	(feasibility)	(quality, soft constraints)
e.g. $X_{\{N, circle\}} \in \{0, 1\}$	e.g. 3)	e.g. use least number of lights

Optimal solution $X^*: \forall X \in C: F(X^*) \leq F(X)$

Graph Colouring Problem

Variables: $X_{node} \in \{1, 2, \dots, k\} \rightarrow$ colours

Constraints: $\forall a, b \in Edges: X_a \neq X_b$

High School Timetabling Problem

Variables $\begin{cases} X_{i,t,c} \in \{0, 1\} \text{ - is lecturer } i \text{ teaching course } c \text{ at time } t\text{?} \\ L_{i,t} \in \{0, 1\} \text{ - is lecturer } i \text{ busy at time } t\text{?} \end{cases}$

Constraints $\begin{cases} L_{i,t} \rightarrow \sum_c X_{i,t,c} = 1, \quad X_{i,t,c} \rightarrow L_{i,t} \\ \forall c \sum_{i,t} X_{i,t,c} = 1 \rightarrow \text{each course must be assigned time and lecturer} \end{cases}$

Objective function - minimize number of holes for lecturers, $\min \sum_{i,t} H_{i,t}$

$B_{i,t} \Leftrightarrow \sum_{k<t} L_{i,k} \geq 1$ - lecturer i busy before time t

$A_{i,t} \Leftrightarrow \sum_{k>t} L_{i,k} \geq 1$ - lecturer i busy after time t

$H_{i,t} \Leftrightarrow B_{i,t} \land \overline{L_{i,t}} \land A_{i,t}$ - lecturer i has a hole at time t

Resource-Constrained Project Scheduling Problem

$task_i = (s_i, D_i, R_i)$

s_i - possible starting time (var.)

D_i - task duration (const.)

R_i - resource requirements (const.)

R_{max} - maximum number of available resources (const.)

Constraints $\begin{cases} \text{Precedence Relations, } task_i < task_j \\ \text{Resource Limitation, } task_i = (s_i, D_i, R_i) \end{cases}$

$S_{i,t} \in \{0,1\}$ - is task i starting at time t?

$\forall i: \sum_t S_{i,t} = 1$ - each task has exactly one starting time.

$\forall task_i < task_j: S_{i,t} \to \bigwedge_{k\in[0,t+dur(i)-1]} \overline{S_{j,k}}$ - Precedence Constraint

$S_{i,t} \to \bigwedge_{k\in[t,t+dur(i)-1]} X_{i,k}$

$\forall t: \sum_i R_i \cdot X_{i,t} \leq R_{max}$ - resource capacity must be respected at each time t

Modelling Patterns

Assignments - i^{th} entity is assigned j^{th} object

1) Integer variable, $Y_i \in \{1, 2, \dots, n\}$ s.t. $Y_i = j$
2) Boolean variable, $X_{i,j} \in \{0,1\}$

Lecturer can only teach at most 2 different days, $\forall i: \sum_d M_{i,d} \leq 2$

$M_{i,d} \in \{0,1\}$ - does lecturer i teach on day d?

$\forall t \in times(d), c \in courses(i): X_{i,t,c} \to M_{i,d}$

$M_{i,d} \to \bigvee_{t\in times(d), c\in courses(i)} X_{i,t,c}$

Auxiliary Variables (help model constraints)

Lecture 6 - Search & Inference

Model

Variables	Constraints	Objective Function
Boolean	Clauses	$\sum_i w_i \cdot x_i$
$x_i \in \{0,1\}$	$(x_i \lor x_j \lor \overline{x_z})$	(Maximum) Satisfiability - (Max)SAT

Constraints - Linear Inequalities ($\sum w_i \cdot x_i \geq k$) -> Pseudo-Boolean (PB)
Variables - Integer -> Integer Program (IP)
Variables - Real-Values -> Linear Program (LP)
Constraints - Predicates ($C: X^n \to \{0,1\}$, $C(x_1, x_2, x_3)$) -> Constraint Programming (CP)

$x_i \in \{0, 1\}$

$w_1: x_1 - x_3 \geq 0$	$w_5: x_1 - x_2 - x_6 \geq -1$		1) $x_1 = 0$, $w_1: x_3 = 0$
$w_2: x_1 + x_2 + x_3 + x_6 \geq 1$	$w_6: x_4 + x_5 \geq 1$		2) $x_2 = 0$, $w_2: x_6 = 1$
$w_3: x_3 - x_4 - x_5 \geq -1$	$w_7: x_4 - x_5 \geq 0$		3) $x_4 = 0$, $w_3: x_5 = 1$, w_7 - conflict
$w_4: x_6 - x_2 \geq 0$	$w_8: x_5 - x_4 \geq 0$		2) $x_4 = 1$, w_0 - conflict
			1) $x_2 = 1$, $w_4: x_6 = 1$, w_5 - conflict

Search (with Pruning/Inference/Propagation)

1) Select Variable } Heuristic
2) Assign value from domain }
3) Propagate $\rightarrow$ (Global) Constraint $C: Y^n \to \{0,1\}$, $C(y_1, y_2, \dots, y_n)$, $y_i \in D \subset \mathbb{N}_0$
4) Conflict? Constraint defines feasibility.

Propagator defines inference for constraint.

$f: D^n \to D^n$, $f(D_1, D_2, \dots, D_n) = (D_1', D_2', \dots, D_n')$ $D_i \subset \mathbb{N}_0$, $D_i' \subseteq D_i$

$x \neq y$, $x \in D_1$, $y \in D_2$ | $x = \{1, 2\}$ $y = \{2, 3, 4, 5\}$ -> no propagation

$D_i = \{a\}$ -> remove a from D_j | $x = \{2\}$ $y = \{2, 3, 4, 5\}$ -> propagate $y \neq \{2\}$

$x \geq y$, $x \in D_1$, $y \in D_2$

$x = \{1, 2, 3\}$ $y = \{1, 2, 3, 4, 5\}$ -> propagate $y \neq \{4, 5\}$ -> $y \leq \text{UpperBound}(x)$

$x = \{1, 2, 3, 4, 5\}$ $y = \{4, 5\}$ -> propagate $x \neq \{1, 2, 3\}$ -> $x \geq \text{LowerBound}(y)$

$y_1 + 2y_2 + 3y_3 \geq 17$ $y_1 \in \{0, 1, 4\}$ $y_2 \in \{0, 2\}$ $y_3 \in \{0, 1, 2, 3, 4\}$

$y_2 = UB(y_2) = 2$, $y_3 = UB(y_3) = 4$ => $LB(y_1) = 1$ -> propagate $y_1 \neq \{0\}$

$y_1 = UB(y_1) = 4$, $y_3 = UB(y_3) = 4$ => $LB(y_2) = 2$ -> propagate $y_2 \neq \{0\}$

$y_1 = UB(y_1) = 4$, $y_2 = UB(y_2) = 2$ => $LB(y_3) = 3$ -> propagate $y_3 \neq \{0, 1, 2\}$

After propagation: $y_1 \in \{1, 4\}$, $y_2 \in \{2\}$, $y_3 \in \{3, 4\}$

$s = \{1, 2, 3\}$ $D = 4$ $T = 6$ => T_3 & T_4 - compulsory resourse consumption

$s_1, s_2 = \{1, 2, 3\}$ $D_1 = D_2 = 4$ $R_{max} = 1$ $T = 6$ -> Compulsory reasoning reveals feasibility.

$s_1 = \{1, 2, 3\}$ $s_2 = \{2, 3, 4, 5\}$ $D_1 = 4$ $D_2 = 2$ -> propagate $s_2 \neq \{2, 3, 4\}$ -> $s_2 = 5$ -> $s_1 = 1$

Lecture 7 - Exponential Growth, Globals, Symmetries

Search as Tree: Node $\equiv$ Variable, Edge $\equiv$ Assignment
Brute force - traverse all paths, propagation - edge removal
Search - Depth-first, Tree size - exponential
2^n binary strings of length n
2^n memory addresses for n-bit architecture
$\begin{cases} 2^{32} \sim 4\text{ GB} \\ 2^{64} \sim 18\ 446\ 744\ 073\text{ GB} \end{cases}$
$10! \sim 3 \times 10^6$, $20! \sim 2.4 \times 10^{18}$

Global Constraints

All-Different$(x_1, x_2, \dots, x_n)$, $x_i \in D_i \subset \mathbb{N}$, all variables must take distinct values
1) Value-Consistency: once a variable is fixed, remove its value from other domains
2) Domain-Consistency: bipartite between variables and union of domains

Circuit, x_i - successor of i, variables x_i represent Hamiltonian cycle
-> Check Algorithm: 1) All-Different on variables 2) Cycle Detection

Symmetries

$X \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ $\quad |X| = 3 \rightarrow$ fixed cardinality
$x_1, x_2, x_3 \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, All-Different$(x_1, x_2, x_3)$
symmetry - different variable assignments represent the same subset
$x_1 < x_2 < x_3 \rightarrow$ symmetry breaking constraints or $x_i \in \{0, 1\}$, $x_i = 1$ if i in set, $\sum x_i = 3$
Colour nodes -> symmetry in renaming colours
1) $x_i = \{1, 2, \dots, n\}$ 2) $y_{i,j} = \{0, 1\}$, $\forall i, j, k \ \ y_{i,j} \wedge y_{j,k} \rightarrow y_{i,k}$

$x_{i,s,d} \in \{0, 1\}$ - employee i assigned shift s at day d
$\forall i, d: \sum_s x_{i,s,d} = 1$ $\quad$ shifts: R, M, D, N
$\forall s, d: \sum x_{i,s,d} \geq M_{d,s}$
$\forall i, d \in$ nonworking_days(i): $x_{i,R,d} = 1$
$\forall i, d: \ x_{i,N,d} \rightarrow \overline{x_{i,M,d}}$
$\forall i, d: \ x_{i,N,d} \wedge x_{i,N,d+1} \longrightarrow \overline{x_{i,N,d+2}}$
Employees with same nonworking hours -> symmetry
Lexicographical ordering for identical employees: R < M < D < N

Lecture 8 - Optimisation, Relaxations, Look Ahead

$y_{i,d} \in \{R, M, D, N\}$ $\Big\{$ $[y_{i,0} < y_{j,0}] \vee ([y_{i,0} = y_{j,0}] \wedge [y_{i,1} < y_{j,1}]) \vee ([y_{i,0} = y_{j,0}] \wedge [y_{i,1} = y_{j,1}] \wedge [y_{i,2} < y_{j,2}]) \vee \ldots$

$R < M < D < N$ $\Big|$ $[y_{i,0} = y_{j,0}] \wedge [y_{i,1} < y_{j,1}] \Leftrightarrow z_{i,j,1} < N \Rightarrow [y_{i,0} < y_{j,0}] \vee z_{i,j,1} \vee z_{i,j,2} \vee \ldots$

Satisfaction Problem -> feasible solution, verification

Optimisation Problem -> best solution

Satisfaction ⊆ Optimisation, Satisfaction - Optimization with trivial objective function

Combinatorial Optimisation Problem - $\min\limits_{x \in C \subseteq \mathbb{N}^n} w \cdot x$

Linear Search

1) feasible solution

2) constraint $f(X) <$ cost (best_solution) - infeasible -> best solution

3) solve

anytime - terminating at any time provides feasible solution

complete - given enough time finds optimal solution

Value Selection Heuristic - Solution-Guided Search -> search near previous solution

=> assign variable the value from best solution so far

Relaxation - simplified version of the original problem

$\max \sum_j w_j \cdot x_j$, $\forall i: \sum_j c_{ij} \cdot x_j \geq k_j$, $x_i \in \{0, 1\}$ - Integer Program (difficult)

$\max \sum_j w_j \cdot x_j$, $\forall i: \sum_j c_{ij} \cdot x_j \geq k_j$, $x_i \in [0, 1]$ - Linear Program (easy)

bound on optimal objective function value -> more constraints worsen the objective

Look Ahead - stronger inference at root level, limited BFS

$\forall v \in \text{domain}(x_i)$: 1) $x_i = v$ 2) propagate 3) unassigned x_i 4) conflict? -> remove v from domain(x_i)

extra inference - track assignments of other variables

$x_i \in \{0, 1\}$ | $x_1 = 0$ no propagation

$c_1: x_1 + x_2 - x_3 \geq 0$ | $x_1 = 1$ no propagation

$c_2: -x_2 + x_3 \geq 0$ | look ahead: $x_2 = 0 \Rightarrow c_3: x_3 = 1$, $c_1: x_1 = 1$, $c_4: x_4 = 1$

$c_3: x_2 + x_3 \geq 1$ | $x_2 = 1 \Rightarrow c_2: x_3 = 1$, $c_4: x_4 = 1$

$c_4: -x_3 + x_4 \geq 0$ | => Regardless of x_2, $x_3 = 1$, $x_4 = 1$

$c_5: x_5 + x_6 - x_3 \geq 1$ $\quad$ $x_3 = 1,\ x_4 = 1$

$c_6: x_5 - x_6 - x_4 \geq -1$ $\quad$ look ahead: $x_5 = 0 \Rightarrow c_5$: conflict! $\Rightarrow x_5 \neq 0$

$x_5 = 1 \Rightarrow c_5: x_6 = 1$

Look ahead - more propagation, computationally expensive

<u>Preprocessing</u> - simplify the problem before solving (expensive/special reasoning)

→ remove duplicate constraints

→ remove subsumed constraints (1) $x_1 + x_2 \geq 1$ 2) $x_1 + x_2 + x_3 \geq 1$ ⇒ remove 2) if $x_3 \in \{0,1\}$)

→ reason over combinations of constraints

$$\left.\begin{array}{l} x_1 + x_2 + x_3 \geq 1 \\ x_1 + x_2 - x_3 \geq 0 \end{array}\right\} 2x_1 + 2x_2 \geq 1 \xrightarrow[x_i \in \mathbb{N}_0]{\text{saturation}} x_1 + x_2 \geq 1$$

Colour Nodes Problem

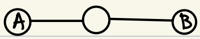
→ Decomposition to strongly connected components, O(#nodes), can solve independently

→ Dominance Rules

node with only one edge

1) same colour as neighbour, Penalty ≥ 1

2) different colour from neighbour, Penalty = 1

subgraph 

Every colouring leads to at least 1 violation.

constraint: nodes A and B have different colours

↳ Removes solutions, at least 1 optimal solution remains

(Exhaustive) Search - iteratively extend feasible partial solution, sophisticated brute force

↳ guaranteed to find (optimal) solution, but slow if weak inference

Local Search - iteratively changing an infeasible solution

↳ fast with weak inference, but not guaranteed to find (optimal) solution

Problem → Model {Modelling Patterns, Variables, (Global) Constraints, Objective Function} → Solve {Search, Propagation, Look Ahead, Relaxations}