

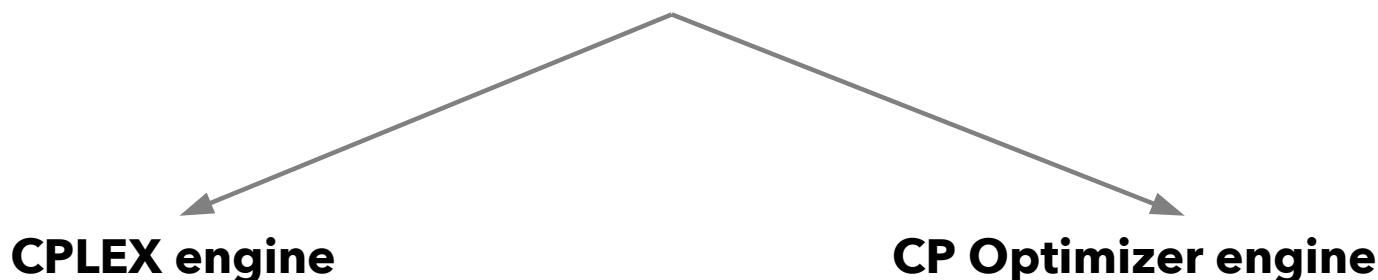
CPLEX Optimization Studio

Combinatorial Optimization

Let:

x	A vector of decision variables
$f(x)$	A function
$C(x)$	Some constraints limiting the possible combinations of values for decision variables

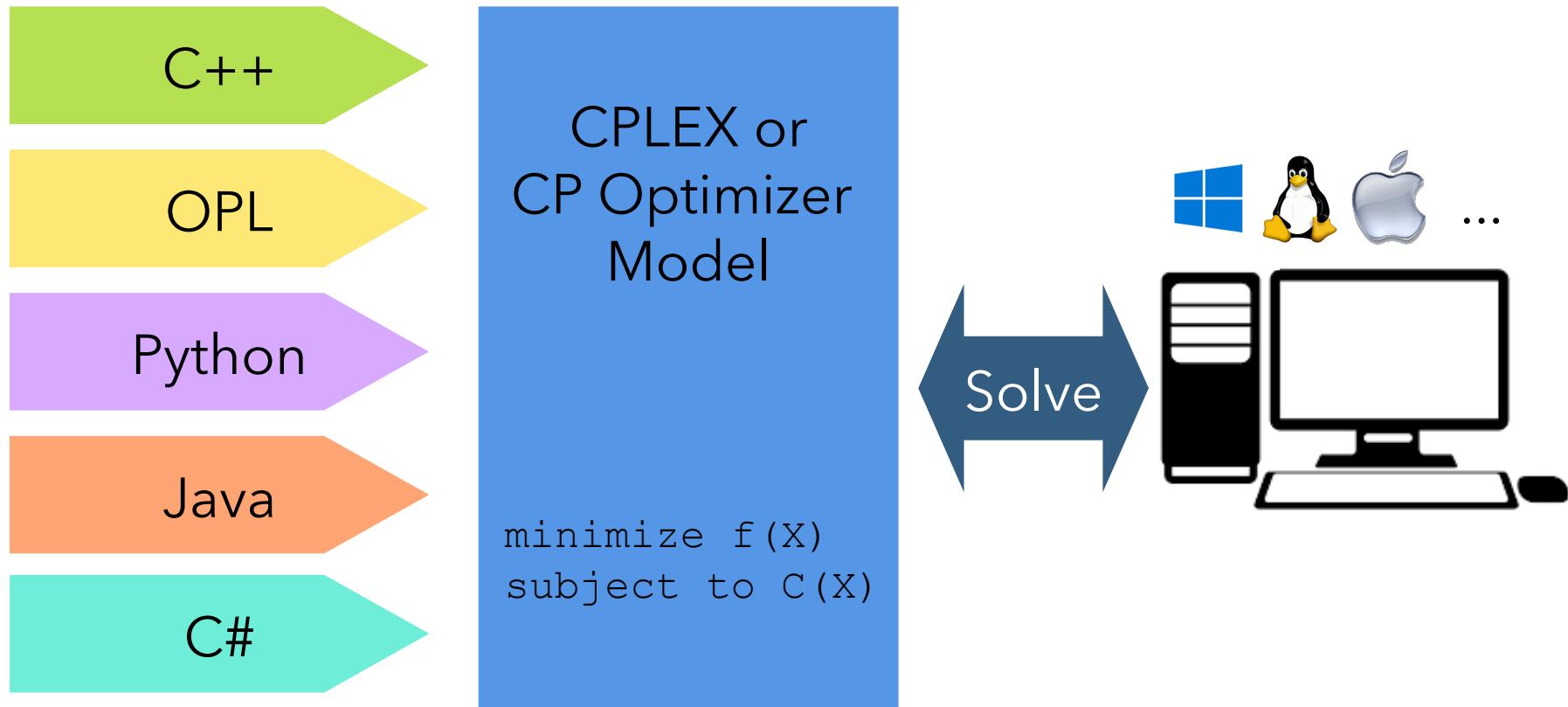
Problem:

$$\begin{aligned} & \text{minimize } f(x) \\ & \text{subject to } C(x) \end{aligned}$$


x	Numerical or Integer variables
$f(x)$	Linear or Quadratic function
$C(x)$	Linear or Quadratic constraints

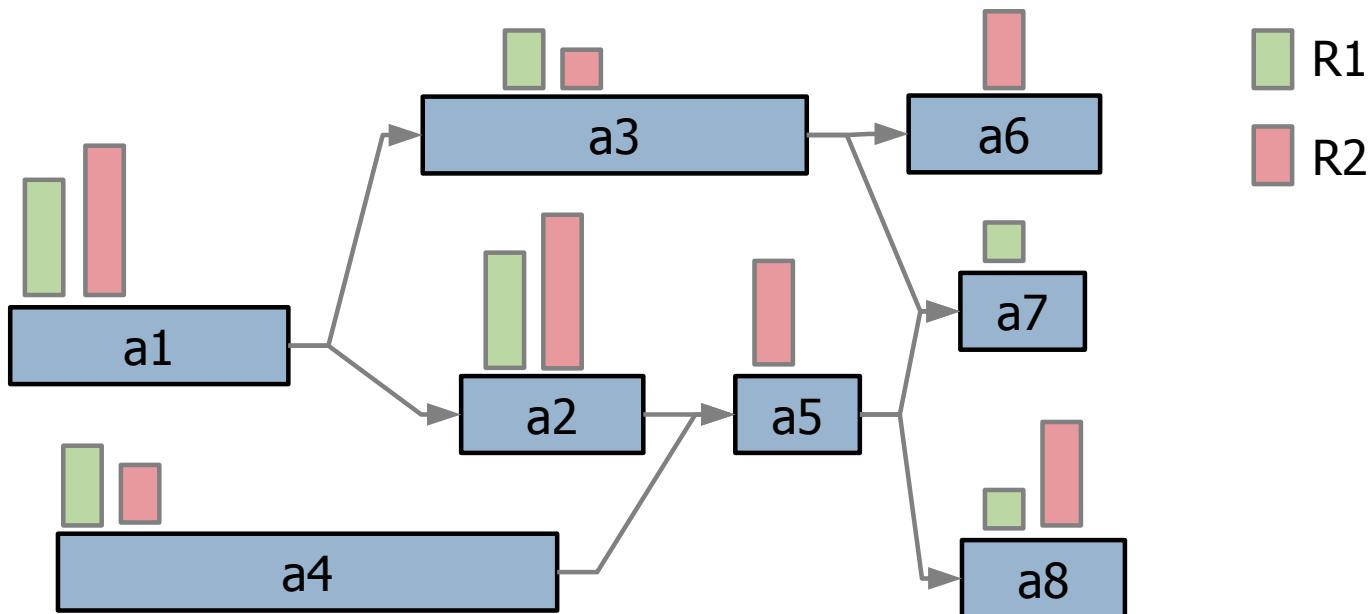
x	Integer or Interval variables
$f(x)$	General function
$C(x)$	General constraints

CPLEX Optimization Studio



Example of a classical scheduling problem (RCPSP)

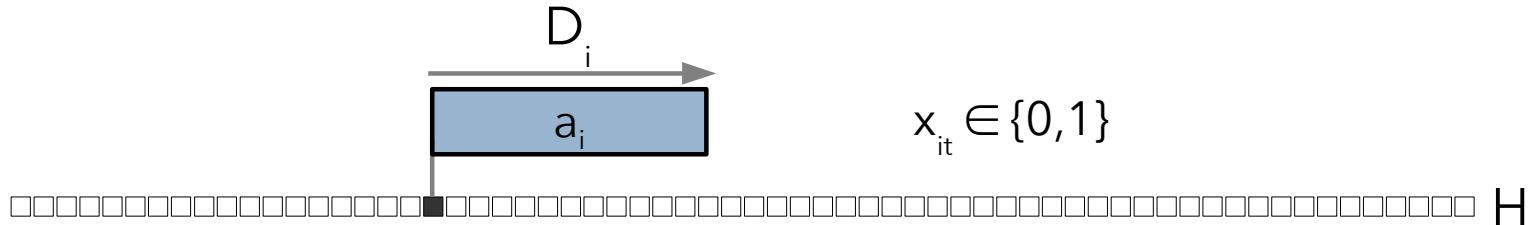
- Given n tasks, m finite capacity resources and precedence constraints ...



- Find a schedule that minimizes project makespan

Example of a classical scheduling problem (RCPSP)

- A standard CPLEX (MIP) formulation (time-indexed)



minimize c

$$\sum_{t \in H} x_{it} = 1 \quad \forall i \in N$$

$$\sum_{t \in H} t x_{it} \leq c \quad \forall i \in N$$

$$\sum_{t \in H} t x_{it} + D_i \leq \sum_{t \in H} t x_{jt} \quad \forall (i, j) \in P$$

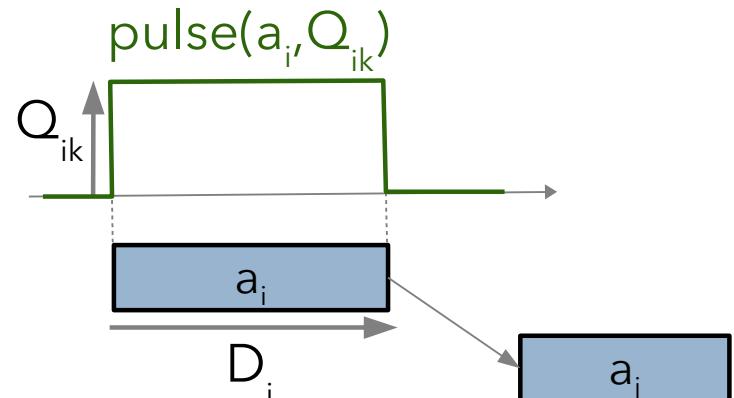
$$\sum_{i \in N, t \leq \tau < t+D_i} Q_{ik} x_{it} \leq R_k \quad \forall \tau \in H, \forall k \in M$$

$$\text{integer } x_{it} \in \{0, 1\} \quad \forall i \in N, \forall t \in H$$

$$\text{integer } c \in H$$

Example of a classical scheduling problem (RCPSP)

- A CP Optimizer formulation



$$\text{minimize} \quad \max_{i \in N} \text{endOf}(a_i)$$

$$\sum_{i \in N} \text{pulse}(a_i, Q_{ik}) \leq C_k \quad \forall k \in M$$

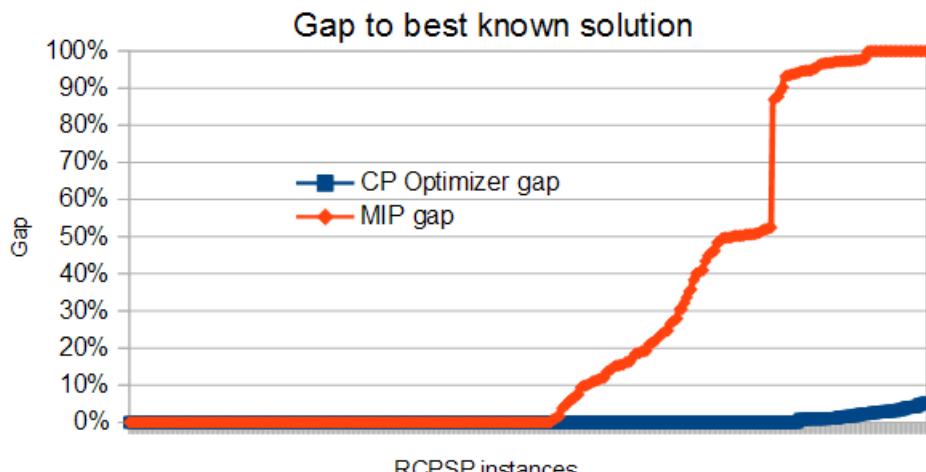
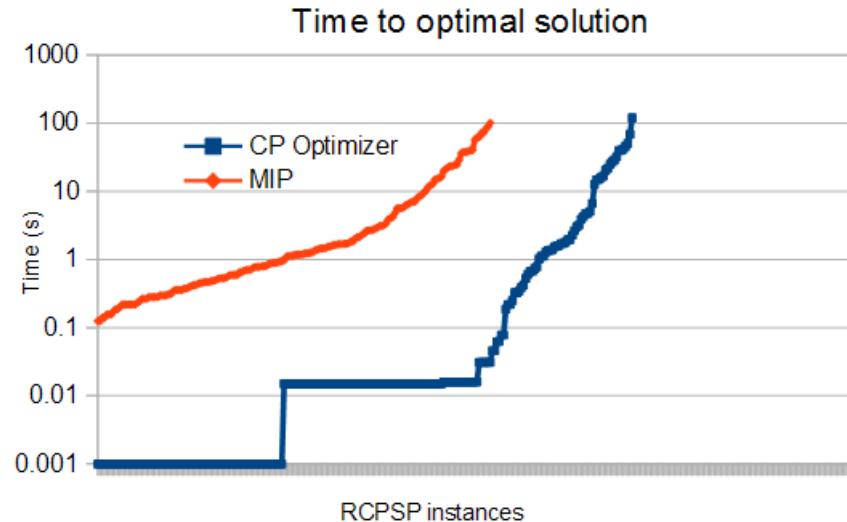
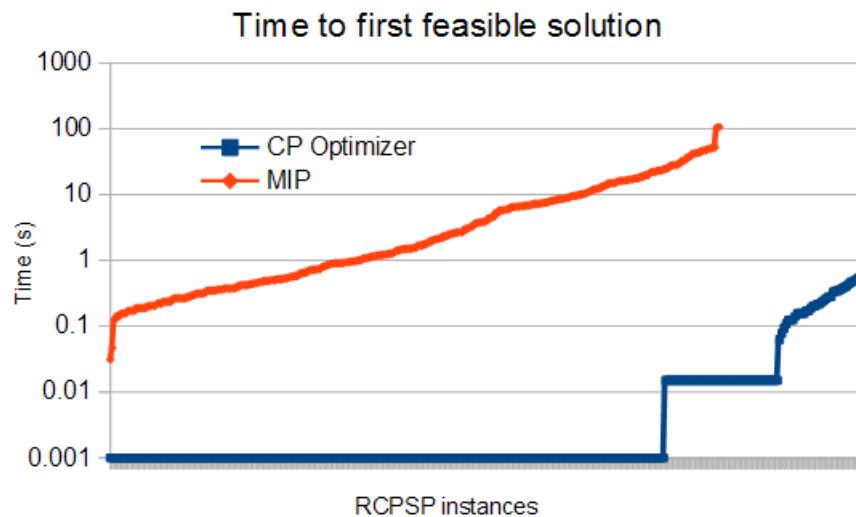
$$\text{endBeforeStart}(a_i, a_j) \quad \forall (i, j) \in P$$

$$\text{interval } a_i, \text{ size} = D_i \quad \forall i \in N$$

- No enumeration of time (H)
- Formulation size grows linearly with size of the data
- Can find good quality solutions for problems with several 10.000 tasks

Example of a classical scheduling problem (RCPSP)

- Comparison CP Optimizer / MIP performance on RCPSP

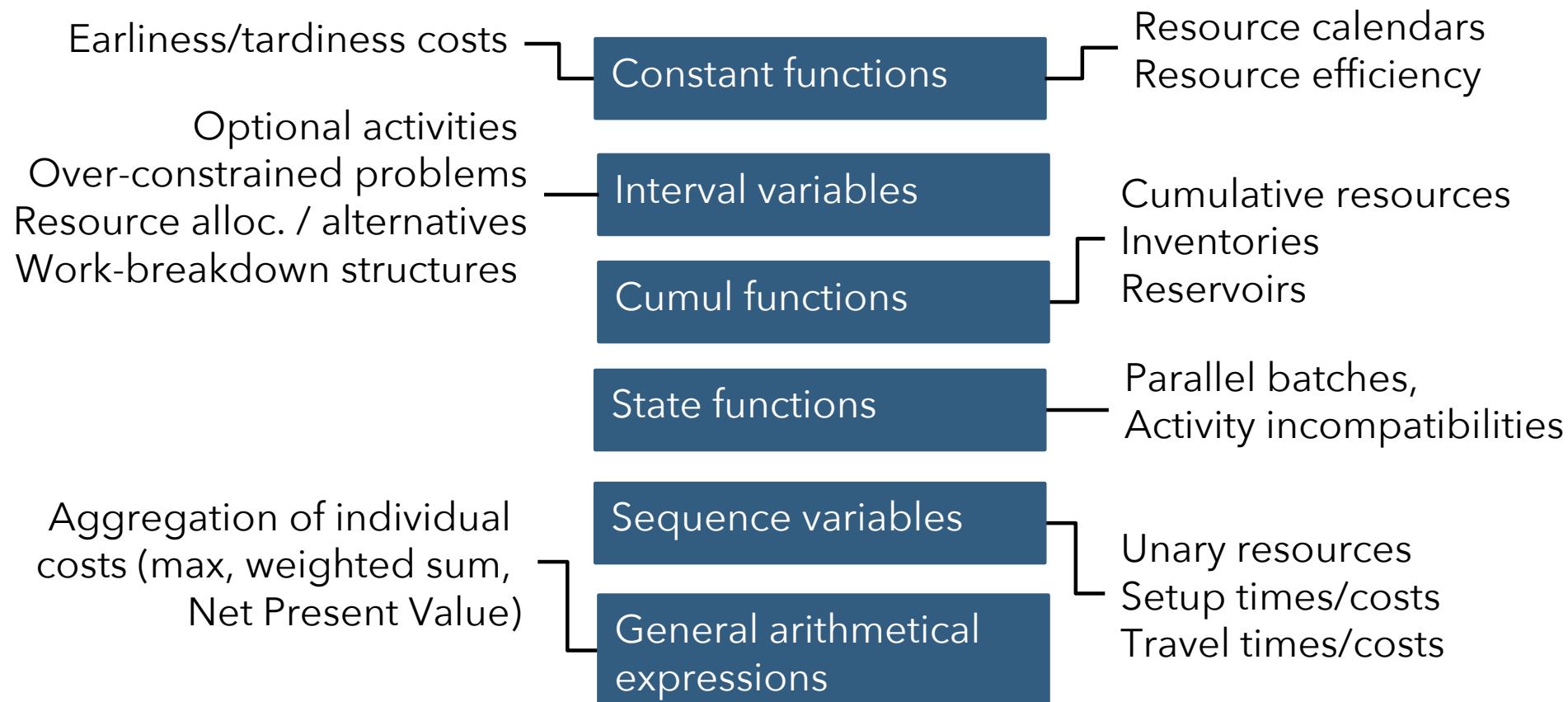


Comparison on 300 classical small RCPSP instances (30-120 tasks) + 40 medium-size ones (900 tasks)

Time-limit: 2 minutes, 4 threads

Beyond RCPSP

- CP Optimizer has mathematical concepts that naturally map to features invariably found in industrial scheduling problems



CP Optimizer RCPSP formulation in Java

- Creation of an instance of CP Optimizer engine:

```
IloCP cp = new IloCP();
```

- Creation of interval variables and end expressions:

```
IloIntervalVar[] a = new IloIntervalVar[n];
IloIntExpr[] ends = new IloIntExpr[n];
for (int i = 0; i < n; i++) {
    a[i] = cp.intervalVar(D[i]);
    ends[i] = cp.endOf(a[i]);
}
```

- Creation of precedence constraints:

```
cp.add(cp.endBeforeStart(a[i],a[j]));
```

- Etc .

- Creation of objective function:

```
cp.add(cp.minimize(cp.max(ends)));
```

- Problem resolution:

```
cp.solve();
```