

Challenge July 2026

A Brick Factory

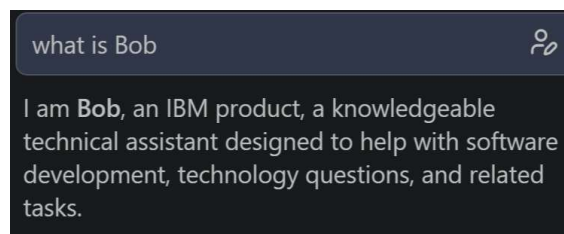
A solution with IBM Bob and OPL CPLEX by Alex Fleischer
afleischer@fr.ibm.com

For this challenge I knew OPL CPLEX could help but I wanted to give it a go with Generative AI in general and IBM Bob in particular. (Bob seems to be good at building, check his clothes!)



What is Bob ?

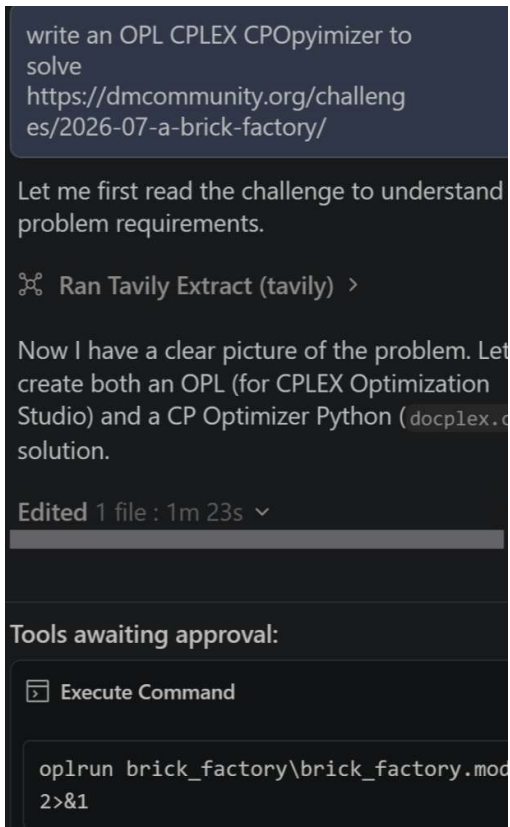
Let me ask Bob.



So within Bob I asked

“write an OPL CPLEX CPOptimizer to solve
<https://dmcommunity.org/challenges/2026-07-a-brick-factory/>”

I even made a typo since I was in a hurry!



1 minute later I got an OPL CPLEX model:

```
/*
 * Brick Factory Scheduling - OPL / CP Optimizer model
 * https://dmcommunity.org/challenges/2026-07-a-brick-factory/
 *
 * Five orders (A-E) must each be scheduled as a single, uninterrupted
 * interval within an 11-day horizon so that the oven's time-varying
 * capacity is never exceeded at any point in time.
 *
 * Order data
 * -----
 * Order Batches/day Duration (days)
 * A      2          1
 * B      1          4
 * C      1          4
 * D      1          2
 * E      2          4
 *
 * Capacity profile
 * -----
 * Days 0-1 : 2 batches/day
```

```

* Days 1-2 : 1 batch/day
* Days 2-3 : 0          (oven offline)
* Days 3-5 : 1 batch/day
* Days 5-10 : 3 batches/day
* Days 10-11 : 1 batch/day
*/

```

```

using CP;

```

```

// -----
// Data
// -----

```

```

int Horizon = 11;
int nOrders = 5;

```

```

range Orders = 0..nOrders-1;

```

```

int dur[Orders]    = [1, 4, 4, 2, 4]; // duration in days
int batchRate[Orders] = [2, 1, 1, 1, 2]; // batches consumed per day

```

```

string orderName[Orders] = ["A", "B", "C", "D", "E"];

```

```

// -----
// Decision variables
// -----

```

```

dvar interval task[i in Orders]
    in 0..Horizon
    size dur[i];

```

```

// -----
// Cumulative function: total oven load
// -----

```

```

cumulFunction ovenLoad =
    sum(i in Orders) pulse(task[i], batchRate[i]);

```

```

// -----
// Objective - minimise makespan
// (any feasible schedule satisfies the challenge)
// In CP models the objective must appear before subject-to.
// -----
minimize max(i in Orders) endOf(task[i]);

```

```

// -----
// Constraints
// -----

```

```

subject to {

```

```

    // Enforce the time-varying capacity via alwaysIn on the cumul function.
    // alwaysIn(f, tmin, tmax, vmin, vmax) - load must stay in [vmin, vmax]
    // over [tmin, tmax).

```

```

alwaysIn(ovenLoad, 0, 1, 0, 2); // days 0-1 : cap 2
alwaysIn(ovenLoad, 1, 2, 0, 1); // days 1-2 : cap 1
alwaysIn(ovenLoad, 2, 3, 0, 0); // days 2-3 : oven offline
alwaysIn(ovenLoad, 3, 5, 0, 1); // days 3-5 : cap 1
alwaysIn(ovenLoad, 5, 10, 0, 3); // days 5-10 : cap 3
alwaysIn(ovenLoad, 10, 11, 0, 1); // days 10-11 : cap 1

}

// -----
// Solution output
// -----
execute DISPLAY_SOLUTION {
  writeln("=== Brick Factory Schedule ===");
  writeln("Order Start End Duration Rate");
  writeln("-----");
  var makespan = 0;
  for (var i in Orders) {
    // Inside execute blocks, interval variable properties are
    // accessed via .start, .end, .size (not startOf/endOf/sizeOf).
    writeln(" " + orderName[i]
      + " " + task[i].start
      + " " + task[i].end
      + " " + task[i].size
      + " " + batchRate[i]);
    if (task[i].end > makespan) makespan = task[i].end;
  }
  writeln("");
  writeln("Makespan: " + makespan + " days");
}

```

Which I can see within a text editor, the OPL IDE or Bob

```

brick_factory.mod ×
rick_factory > brick_factory.mod > ...
60 // -----
61 minimize max(i in Orders) endOf(task[i]); rick_factory > brick_factory.mod > ...
62 // -----
63 // Constraints
64 // -----
65 // -----
66 subject to {
67 // Enforce the time-varying capacity v
68 // alwaysIn(f, tmin, tmax, vmin, vmax)
69 // over [tmin, tmax).
70 // -----
71 alwaysIn(ovenLoad, 0, 1, 0, 2); // 69
72 alwaysIn(ovenLoad, 1, 2, 0, 1); // 70
73 alwaysIn(ovenLoad, 2, 3, 0, 0); // 71
74 alwaysIn(ovenLoad, 3, 5, 0, 1); // 72
75 alwaysIn(ovenLoad, 5, 10, 0, 3); // 73
76 alwaysIn(ovenLoad, 10, 11, 0, 1); // 74
77 }
78

```

I see the result

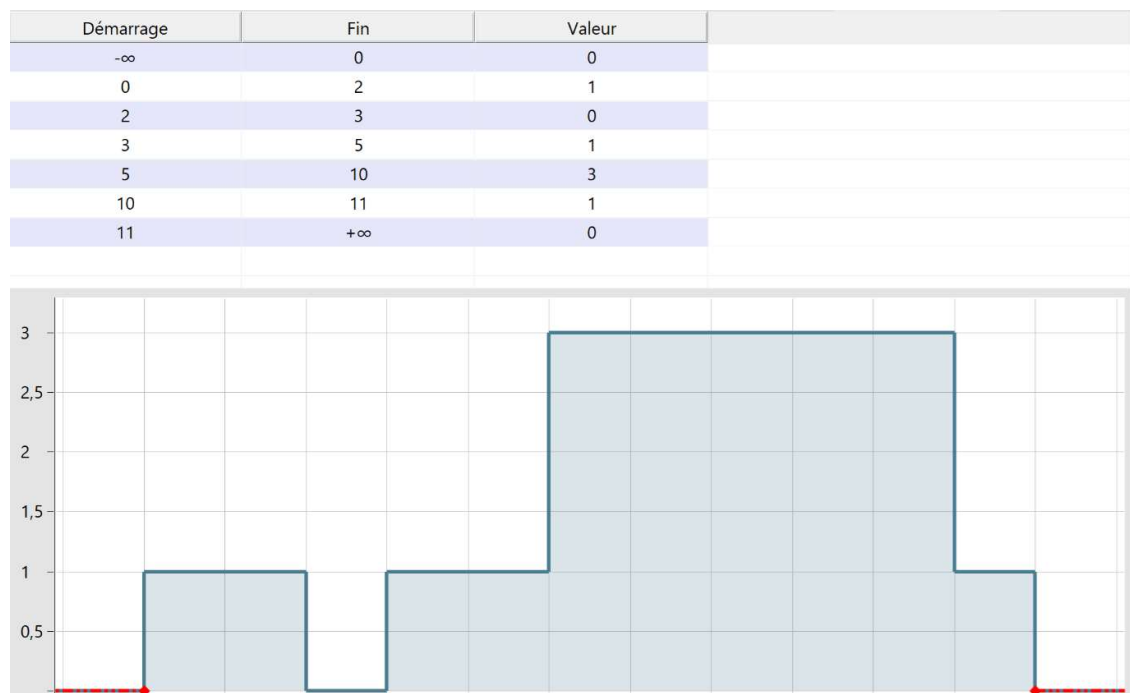
```

// solution with objective 11
=== Brick Factory Schedule ===
Order Start End Duration Rate
-----
A      5      6      1      2
B      3      7      4      1
C      7     11      4      1
D      0      2      2      1
E      6     10      4      2

```

Makespan: 11 days

And within the OPL IDE I can see for oven load



So Bob helped me be fast and I can check the model since the model is in OPL which is very easy to read.

Thanks Bob!