Unit Commitment Optimization

Economical Dispatch – LS2N

Authors:

Stephanie M. Matta Bobadilla

Alejandro Moran Velazquez

M1 DREAM OPTIM

Linear Programming - AMPL Minimization

November 2024, Nantes, Fr.

* Both authors contributed equally to all the tasks.

UNIT COMMITMENT PROBLEM

The Unit Commitment Problem (UCP) is a fundamental optimization challenge in power system operations. Its primary objective is to minimize generation costs while ensuring that demand is satisfied at all times. The problem involves scheduling generation units across multiple time periods while adhering to technical and operational constraints, such as generation limits (minimum and maximum), startup costs, minimum generation up time, among others.

In this exercise, we focus on thermal generation units only (nuclear, gas, coal, and oil), excluding renewable and hydro resources, to simplify the problem. The system is designed to classify generation units into three categories based on their operational behavior:

1. Base Units: Represent technologies like nuclear plants
2. Semi-Base Units: Include gas-powered generation units
3. Peak Units: Include coal- and oil-powered plants, which are activated only during peak demand periods.

```
"BASE"          1e6 50     900     1200
"SEMIBASE"      1e3 70     120      500
"PEAK"          0   90     0        300
```

To warm up, the following theoretical example is performed:

Circle Exercise

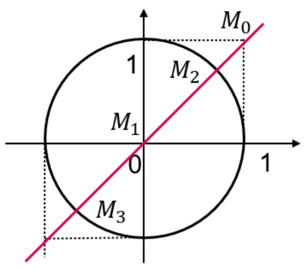
Objective

The goal of this exercise is to minimize the distance between a given point (1,1) and the origin (0,0), constrained by a circular boundary defined as, $x^2 + y^2 \leq 1$.In other words, finding the closest point within the unitary circle to the point (1,1).

The optimization problem is formulated as:

objective: $min\ (x-1)^2 + (y-1)^2$

subject to: $x^2 + y^2 \leq 1$



Model Interpretation

The problem was implemented in AMPL as follows:

```
var x;
var y;
minimize Distance: (x-1)^2 + (y-1)^2;
subject to Circle_Limit: x^2 + y^2 <= 1;


#
solve;
display x,y;
display Circle_Limit;
```

This code defines the variables x and y, the objective function to minimize (squared distance), and the circular constraint. The solver computes the optimal solution using the Knitro Solver. This simple exercise demonstrates the advantage of using AMPL for solving optimization problems. The main challenge lies in accurately formulating the optimization problem and translating it correctly into AMPL syntax.

Results

Using the AMPL Solver we got the following results:

- Final feasibility error (abs / rel) = 9.53e-11 / 9.53e-11 (negligible, constraint satisfied) *
- Final optimality error (abs / rel) = 3.95e-11 / 3.95e-11 (satisfying KKT conditions)**
- Number of Iterations: 5

***Feasibility Error**: This error measures the extent to which the constraints are violated. A small numerical value indicates that the constraints are nearly or fully satisfied, making the error negligible. It is quantified by the maximum constraint violation observed in the solution.

****Optimality Error**: This error evaluates how closely the solution adheres to the KKT conditions, which are necessary for optimality. A numerically small optimality error indicates that the solution is very close to being optimal, as it satisfies the KKT conditions with minimal violations.

The solution was obtained in 5 iterations complying with KKT optimality conditions:

$$\nabla f_0(x^*) + \sum \lambda_k^* \nabla \mathrm{f_k}(\mathrm{x}^*) = 0$$

$$\lambda_k \cdot f_k(x^*) = 0$$
$$\lambda_k \geq 0$$

Optimal Values

- x = 0.707107
- y = 0.707107

Constraint Verification

The circular constraint was satisfied, with:

$x^2 + y^2 = 0.7071072 + 0.7071072 = 0.999999489 \leq 1$

(The Lagrange multipliers (λk) for the constraint exhibit negative values due to the convention of the solver using negative gradients for minimization problems).

λ = -0.414214,

The fact that we obtained a Lagrange multiplier, λ=−0.414214, which satisfies the KKT conditions, **confirms that the point is optimal.**

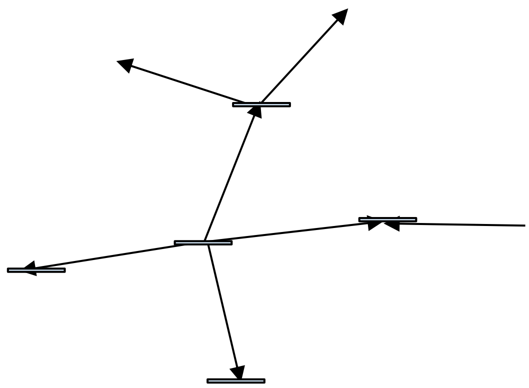
Data Structure

The system is described using three primary data files:

1. Technology File:
    - Columns:
        - UnitCost: Cost of switching the unit on.
        - Cost: The cost of producing 1 unit of power with the specified technology.
        - Pmin: Minimum power injection required when the unit is active.
        - Pmax: Maximum power capacity of the unit.
2. Buses File:
    - Represents the electrical buses (substations) in the network.
    - Each bus is uniquely identified and may have one or more generation units (base, semi-base, or peak) connected to it.
    - Buses act as nodes where power is injected into or withdrawn from the system.
3. Branches File:
    - Models the transmission lines that connect the buses.
    - Defines the flow of power between nodes, constrained by the capacity of each line.

Power Flow convention:

To the node positive - Production , To the node negative – Demand, as showed in the following simplified topology:



Part 1

The approach begins with modelling the core variables, such as power production ($P_{i,t}$), associated costs (C_i), and demand ($d_{i,t}$), and incrementally refining the constraints to address practical challenges in generation scheduling.

Variables:

$P_{i,t}$ Power output of generator i at time t

d_t Demand at time t

C_i Production cost per unit of generator i at time t

- **Objective Function:** Minimize the total production cost across all generation units and time periods

$$min \sum_{i \in Generators,} \sum_{t \in TimeSteps} C_{i,t} * P_{i,t}$$

Power Balance: The total power generated at each time step must equal the total demand**, first and only initial constraint considered.**

$$\sum_{i \in Generators,} \sum_{t \in TimeSteps} P_{i,t} = d_t \ \{\forall t\}$$

Generation Lower Bound: Non-Negative Production added constraint, since production cannot be negative physically.

$$P_{i,t} \geq \boldsymbol{0} \{\forall i, t\}$$

Generation Upper Bound: Each generator has a maximum capacity ($P_{MAX,i}$) to avoid relying solely on the cheapest technology

$$P_{i,t} \leq P_{MAX,i} \{\forall i, t\}$$

Implementation

The model was implemented in AMPL and solved using the Knitro Solver. The first step was to define the objective function and first constraint (power balance) in AMPL syntax.

```
# Objective Sum of all Costs t times Production t
minimize production_cost:
    sum{bus in POWER_BUSES,ts in USED_TS,s in SAMPLE}COST[B_TECH[bus]]*u[bus,ts,s];
```

Initial Results: When the model was executed, the solution was marked as "unbounded or infeasible", indicating missing constraints. Specifically, the lack of non-negative production (lower bound) constraints.

```
EXIT: Problem is unbounded or infeasible.

Final Statistics
----------------
Final objective value               =   9.52906280210000e+03
Final feasibility error (abs / rel) =   1.34e+03 / 1.00e+00
Final optimality error  (abs / rel) =   1.80e+308 / 1.80e+308
# of iterations                     =          0
# of CG iterations                  =          0
# of function evaluations           =          0
# of gradient evaluations           =          0
# of Hessian evaluations            =          0
Total program time (secs)           =       0.002 (     0.000 CPU time)
Time spent in evaluations (secs)    =       0.000

===============================================================================

Knitro 13.0.0: Problem is unbounded or infeasible.
objective 9529.062802; feasibility error 1.34e+03
0 iterations; 0 function evaluations

suffix feaserror OUT;
suffix opterror OUT;
suffix numfcevals OUT;
suffix numiters OUT;
production_cost = 9529.06

PS Microsoft.PowerShell.Core\FileSystem::\\data-pfe\AMORANVELA2024\Documents\OPTIMIZATION\DREAM TP\dream_tp>
```

A constraint for positive production was added to ensure $P_{i,t} \geq \boldsymbol{0}\, \{\forall i, t\}$. After incorporating this, the solver produced a feasible solution and better performance for the optimization problem.

```
var u{POWER_BUSES, USED_TS, SAMPLE}>=0;
```

```
EXIT: Optimal solution found.

Final Statistics
----------------
Final objective value               =   9.88948438292406e+06
Final feasibility error (abs / rel) =   1.31e-12 / 1.02e-15
Final optimality error  (abs / rel) =   5.21e-08 / 5.79e-10
# of iterations                     =          7
# of CG iterations                  =          0
# of function evaluations           =          0
# of gradient evaluations           =          0
```

Base Technology Dominance: Initially, the solution relied entirely on base technology (nuclear) due to its low production cost. Consequently, the semi-base and peak technologies are not utilised due to their higher costs, which is also unfeasible and limited. This result was expected but highlighted a need for further constraints to avoid over-reliance on a single technology.

```
57    display{bus in POWER_BUSES,ts in USED_TS,s in SAMPLE}(B_TECH[bus], u[bus,ts,s]);

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

2    1 1   BASE        1043.52
4    0 1   SEMIBASE       4.24586e-08
4    1 1   SEMIBASE       4.29774e-08
5    0 1   BASE        1059.43
5    1 1   BASE        1041.12
6    0 1   BASE        1060.14
6    1 1   BASE        1041.6
9    0 1   BASE        1055.19
9    1 1   BASE        1036.14
10   0 1   PEAK           1.41301e-08
10   1 1   PEAK           1.41619e-08
12   0 1   SEMIBASE       4.20214e-08
12   1 1   SEMIBASE       4.24833e-08
13   0 1   BASE        1080.93
13   1 1   BASE        1061.12
14   0 1   BASE        1072.4
14   1 1   BASE        1055.31
15   0 1   PEAK           1.41441e-08
15   1 1   PEAK           1.41373e-08
19   0 1   PEAK           1.4177e-08
```

Generation Limit Constraint ($P_{MAX,i}$): To distribute production across multiple technologies (e.g., gas, coal, and oil), a constraint was added to cap the production of each generator:

$$P_{i,t} \leq P_{MAX,i} \ \{\forall i, t\}$$

```
# expand power_balance;
subject to MaxPowerG{bus in POWER_BUSES,ts in USED_TS,s in SAMPLE}:
    #Genereted Power should be positive and lower than the Pmax for every Technology.
    u[bus, ts, s]<=PMAX[B_TECH[bus]];
```

Outcome: Despite the added constraints, the optimization **still relied** heavily on nuclear technology due to the absence of further considerations like transmission limits or ramping costs.

```
objective 9889484.383; feasibility error 6.25e-13
13 iterations; 0 function evaluations

suffix feaserror OUT;
suffix opterror OUT;
suffix numfcevals OUT;
suffix numiters OUT;
production_cost = 9889480

:  B_TECH[bus] u[bus,ts,s]    :=
;

PS Microsoft.PowerShell.Core\FileSystem::\\data-pfe\AMORANVELA2024\Documents\OPTIMIZATION\DREAM TP\dream_tp>
```

In the previous figure, we aimed to display instances of generation technologies different from the base (nuclear) producing more than 1 unit of power. However, no such cases were observed, indicating that the optimization process continues to rely exclusively on nuclear technology due to its lower cost. This highlights the need for additional constraints to encourage the utilization of semi-base and peak technologies, which will be addressed in Part 2 of the model.

(End of Optimization Problem Part 1)

Part 2

As previously mentioned, although the earlier model computed an optimal value, we logically understand that this value is not feasible in reality. In the second part of the problem, we focused on constructing a more robust model and utilized the AMPL Xpress solver, which is a MILP (Mixed-Integer Linear Programming) solver.

Key Challenges:

1. **Satisfying Demand with Minimum Production:**

Considering the following scenario:

Let Demand d_t = 900 MW

Generator 1: $P_{MIN\ 1} = 500, \quad P_{MAX1} = 1000$

Generator 2: $P_{MIN\ 2} = 500, \quad P_{MAX2} = 1000$

While one generator can satisfy the demand, splitting the load equally (450 MW each) between the two generators is preferable to avoid operating a generator near its maximum capacity. However, turning on both generators simultaneously violates the minimum production constraint ($P_{MIN\ 1}$ +$P_{MIN\ 2} > d_t$), rendering the problem infeasible.

2. **Startup Costs and Binary Variables:**

To address these challenges, binary variables were introduced to "turn on" or "turn off" generators, enabling the activation or deactivation of P_{MIN} and P_{MAX} constraints.

Reformulated Optimization Model

Objective Function: Minimize production and startup costs:

$$min \sum_i \sum_t C_i * P_{i,t} + C_{i,start} * start_{i,t}$$

Updated Constraints:

- **Production Bounds:** $P_i min * on_{i,t} \leq P_{i,t} \leq P_i max * on_{i,t} \quad \{\forall i, \forall t\}$
- **Binary Variables:** $on_{i,t} \in \{0,1\} \quad \{\forall i, \forall t\}$
- **Startup Cost:** $start_{i,t} = on_{i,t}(1 - on_{i,t-1}) \quad \{\forall i, \forall t\}$

Linearization:

In the case when we have a non linear constraint, (*i.e.* $z = x.y$ *in startup costs)* but one of the two variables is a binary variable, the constraint can be linearized using the following relationships:

Let x be a continuous variable in $[L,U]$ and y a binary variable (worth 0 or 1), show that the following set describes the points (x,y,z) such that $z=x{\cdot}y$.

$$\begin{cases} z - x - Ly + L \leq 0 \\ z - Ly \geq 0 \\ z - x - yU + U \geq 0 \\ z - Uy \leq 0 \end{cases}$$

Note the inclusion of the ***start*** binary variable. The purpose of introducing this variable is to have a value that is 1 only when the previous time step was 0. This allows us to represent the startup cost of turning on a generator, which can then be added to the objective function.

$$start_{i,t} = on_{i,t}(1 - on_{i,t-1}), \{\forall i, \forall t\}$$

With this revised model, we then converted the optimization language into AMPL

Implementation in AMPL

1. Binary Variable Definition: $on_{i,t} \in \{0,1\}$, $start_{i,t} = on_{i,t}(1 - on_{i,t-1})$, $\{\forall i, \forall t\}$

```
#Start of binary variables

var on{POWER_BUSES, USED_TS} binary;

var start{POWER_BUSES, USED_TS};
```

2. Adding the *on* binary variable to Pmin and Pmax Constraints. $P_i min * on_{i,t} \leq P_{i,t} \leq P_i max * on_{i,t}$ $\{\forall i, \forall t\}$

```
subject to positive_generation{bus in POWER_BUSES, ts in USED_TS}: u[bus,ts]
>= 0;

subject to max_generation{bus in POWER_BUSES, ts in USED_TS}: u[bus,ts] <=
PMAX[B_TECH[bus]]*on[bus, ts];

subject to min_generation{bus in POWER_BUSES, ts in USED_TS}: u[bus,ts] >=
PMIN[B_TECH[bus]]*on[bus, ts];
```

3. Linearizing the nonlinear constraint, $start_{i,t} = on_{i,t}(1 - on_{i,t-1})$

$$\begin{cases} z - x - Ly + L \leq 0 \\ z - Ly \geq 0 \\ z - x - yU + U \geq 0 \\ z - Uy \leq 0 \end{cases}$$

Where z = start [i,ts] , x = on[i, ts] , y = (1-on[i, ts - 1]), L = 0 , U = 1

```
subject to binary_constraint_1: start[i, ts] - on[i, ts] - L*(1-on[i, ts -
1]) + L <= 0;

subject to binary_constraint_2: start[i, ts] - on[i, ts] - U*(1-on[i, ts -
1]) + U >= 0;

subject to binary_constraint_3: start[i, ts] - L*(1-on[i, ts - 1]) >=0;
```

```
subject to binary_constraint_4: start[i, ts] - U*(1-on[i, ts - 1]) <=0;
```

Updating the objective function to include the startup costs, $min \sum_i \sum_t C_i * P_{i,t} + C_{i,start} * start_{i,t}$

```
minimize production_cost:

sum{bus in POWER_BUSES,ts in USED_TS} (

    COST[B_TECH[bus]]*u[bus,ts] + UNITCOST[B_TECH[bus]]*start[bus,ts]);

solve;
```

With these updated constraints and objective function the simulated yielded the following results:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

  Solution time / primaldual integral :          0s/ 98.849245%
  Number of solutions found / nodes   :           1 /         1
  Max primal violation      (abs/rel) : 2.922e-11 / 2.922e-11
  Max integer violation     (abs    ) :       0.0
XPRESS 39.01.01: Global search complete
Best integer solution found 9889484.383
1 branch and bound node
No basis.
production_cost = 9889480
```

Optimal Cost Computed by AMPL: 9,889,484.383

The optimal solution remains unchanged compared to the previous iteration. This happens because the model assumes that all units are already "on" at the start of the simulation. Consequently, startup costs are neglected, as no unit transitions from an "off" to an "on" state during the optimization.

Adding a Constraint to initialize all units as 'off'

To properly account for startup costs, a constraint was added to ensure that all generators begin in the "off" state. This is achieved by defining the start variable to equal the on variable at t=0.

The start variable takes the value of 1 only when a generator transitions from an "off" state (on=0) to an "on" state (on=1). Since the model lacks access to a t = −1 state, this relationship approximates the startup behaviour at t=0:

on[bus,0] = start[bus,0]; , $on_{i,0} = start_{i,0}$

This implies that if a generator is "on" at t=0, it is treated as having just started up, and its associated startup cost is added to the objective function.

```
#Add initial state - all off
```

```
subject to initial_off {bus in POWER_BUSES}: on[bus,0] = start[bus,0];
```

With this additional consideration the simulation gives us the following results:

```
Final MIP bound                            : 7.122400098770954e+07
  Solution time / primaldual integral :          0s/ 47.254033%
  Number of solutions found / nodes   :          1 /          1
  Max primal violation      (abs/rel) : 2.149e-11 / 2.149e-11
  Max integer violation     (abs     ) :       0.0
XPRESS 39.01.01: Global search complete
Best integer solution found 71224071.89
1 branch and bound node
No basis.
production_cost = 71224100
```

Optimal Cost Computed by AMPL: 71,224,100

This result is consistent with expectations, as the inclusion of startup costs forces the model to be more selective in determining which generators to turn on. By accurately accounting for the cost of transitioning units from "off" to "on," the model now reflects a more cautious and realistic operation strategy, avoiding unnecessary startups and prioritizing cost efficiency.

Minimum duration constraint

To create a more robust model that better reflects reality, we account for the condition that when a generator is switched on, it must remain on for a minimum duration.

We can represent this logically and mathematically like:

If start = 1 (meaning a generator was switched on) then

$on_{i,t'} = 1, \forall t' \in \{t, t + MD\}$, meaning ***on*** must be 1 for a subset t , **from** t where the unit was switched on, **to** t + an arbitrary duration.

This can be viewed mathematically as:

$start_{i,t} \leq on_{i,t'}, \forall t' \in \{t + MD\}$, ***start*** is forced to 0, when ***on*** is 0 during a time period.

$t \leq t' \leq t + MD$, MD = 10 time steps

Translating this constraint to AMPL we write:

```
subject to minimum_on_duration {bus in POWER_BUSES, ts in USED_TS, t_prime in USED_TS: t_prime >= ts and t_prime <= ts + MD}: start[bus, ts] <= on[bus,t_prime];
```

The final step in the simulation was to increase the time step to 50 to observe the effects of the newly added constraint. **Note that MAX_TS is limited to 50,** as using the full dataset would demand significant computational resources, in the previous simulations MAX_TS = 1, thus an increment can be observed.

```
### TIME HORIZON

set TS;

param MIN_TS := 0;

param MAX_TS := 50;
```

To test the constraints, we ran the simulation with and without the minimum duration constraints, obtaining the following results:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

  Max dual violation        (abs/rel) : 1.237e-10 / 1.164e-10
  Max complementarity viol. (abs/rel) :       0.0 /       0.0

Starting root cutting & heuristics

 Its Type    BestSoln    BestBound   Sols    Add    Del     Gap
k         315115299.1  315094661.1      1                   0.01%
STOPPING - MIPRELSTOP target reached (MIPRELSTOP=0.0001  gap=6.5493
 *** Search completed ***
Uncrunching matrix
Final MIP objective                   : 3.151152990893352e+08
Final MIP bound                       : 3.150946611322528e+08
  Solution time / primaldual integral :         1s/ 99.758187%
  Number of solutions found / nodes   :         1 /         0
  Max primal violation      (abs/rel) : 3.075e-11 / 3.075e-11
  Max integer violation     (abs    ) :       0.0
XPRESS 39.01.01: Global search complete
Best integer solution found 315115299.1
0 branch and bound nodes
production_cost = 315115000
```

Optimal Cost Computed by AMPL (without minimum duration constraint): 315,115,000

Final Model:

Objective Function: Minimize production and startup costs:

$$min \sum_{i} \sum_{t} \quad C_i * P_{i,t} + C_{i,start} * start_{i,t}$$

Updated Constraints:

- **Production Bounds:** $P_i min * on_{i,t} \leq P_{i,t} \leq P_i max \ * on_{i,t} \quad \{\forall i, \forall t\}$
- **Binary Variables:** $on_{i,t} \in \{\boldsymbol{0,1}\} \quad \{\forall i, \forall t\}$
- **Startup Cost:** $start_{i,t} = on_{i,t}(\boldsymbol{1} - on_{i,t-\boldsymbol{1}}) \quad \{\forall i, \forall t\}$
- **Initial Units are Off** $on_{i,\boldsymbol{0}} = start_{i,\boldsymbol{0}}$
- **Minimum On Duration Constraint** $start_{i,t} \leq \ on_{i,t'}, \forall t, \forall t' \in \{t + MD\}$

Results with minimum on duration constraint added:

```
Cuts in the matrix         : 718
Cut elements in the matrix : 20196
 *** Search completed ***
Uncrunching matrix
Final MIP objective                   : 3.154554905814697e+08
Final MIP bound                       : 3.154340244791424e+08
  Solution time / primaldual integral :         2s/ 63.586244%
  Number of solutions found / nodes   :          1 /         0
  Max primal violation      (abs/rel) : 1.455e-11 / 1.455e-11
  Max integer violation     (abs    ) :       0.0
XPRESS 39.01.01: Global search complete
Best integer solution found 315455490.6
0 branch and bound nodes
production_cost = 315455000
```

Optimal Cost Computed by AMPL (without minimum duration constraint): 315,455,000

Recalling that this simulation is solved by a MILP solver, note that the production cost is higher because adding another constraint reduces the feasible domain of the optimization problem.

In conclusion, this lab helped us understand optimization language better and apply it using AMPL software. Working with a real-life scenario related to our field made it more practical and something we will definitely find useful in the future. The model could still be improved by adding features like minimum stop time constraints or including storage units to make it even more realistic.

APPENDIX

Circle example results

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS Microsoft.PowerShell.Core\FileSystem::\\data-pfe\AMORANVELA2024\Documents\OPTIMIZATION\Ex_1> ampl .\CircleExcercise1.run
##### This license is only intended for use by RTE. #####
##### License is valid until Sep 15, 2025 #####
Artelys Knitro 13.0.0: ##### This license is only intended for use by RTE. #####
##### License is valid until Sep 15, 2025 #####

=======================================
          Commercial License
         Artelys Knitro 13.0.0
=======================================

No start point provided -- Knitro computing one.

Knitro presolve eliminated 0 variables and 0 constraints.

concurrent_evals:        0
datacheck:               0
hessian_no_f:            1
The problem is identified as a convex QCQP.

Problem Characteristics                                 (   Presolved)
-----------------------
Objective goal:  Minimize
Objective type:  quadratic
Number of variables:                                  2 (          2)
    bounded below only:                               0 (          0)
    bounded above only:                               0 (          0)
    bounded below and above:                          0 (          0)
    fixed:                                            0 (          0)
    free:                                             2 (          2)
Number of constraints:                                1 (          1)
    linear equalities:                                0 (          0)
    quadratic equalities:                             0 (          0)
    gen. nonlinear equalities:                        0 (          0)
    linear one-sided inequalities:                    0 (          0)
    quadratic one-sided inequalities:                 1 (          1)
    gen. nonlinear one-sided inequalities:            0 (          0)
    linear two-sided inequalities:                    0 (          0)
    quadratic two-sided inequalities:                 0 (          0)
    gen. nonlinear two-sided inequalities:            0 (          0)
Number of nonzeros in Jacobian:                       2 (          2)
Number of nonzeros in Hessian:                        2 (          2)

Knitro using the Interior-Point/Barrier Direct algorithm.
```

```
Knitro using the Interior-Point/Barrier Direct algorithm.

  Iter      Objective      FeasError   OptError    ||Step||    CGits
--------  --------------  ----------  ----------  ----------  -------
       0    1.470379e+00   0.000e+00
       5    1.715729e-01   9.530e-11   3.948e-11   1.512e-06        0

EXIT: Optimal solution found.

Final Statistics
----------------
Final objective value               =   1.71572875214333e-01
Final feasibility error (abs / rel) =   9.53e-11 / 9.53e-11
Final optimality error  (abs / rel) =   3.95e-11 / 3.95e-11
# of iterations                     =          5
# of CG iterations                  =          0
# of function evaluations           =          0
# of gradient evaluations           =          0
# of Hessian evaluations            =          0
Total program time (secs)           =       0.002 (     0.000 CPU time)
Time spent in evaluations (secs)    =       0.000

===============================================================================

Knitro 13.0.0: Locally optimal or satisfactory solution.
objective 0.1715728752; feasibility error 9.53e-11
5 iterations; 0 function evaluations

suffix feaserror OUT;
suffix opterror OUT;
suffix numfcevals OUT;
suffix numiters OUT;
x = 0.707107
y = 0.707107

Circle_Limit = -0.414214
```

```
#Start of binary variables

var on{POWER_BUSES, USED_TS} binary;

var start{POWER_BUSES, USED_TS};

param L = 0;

param U = 1;

subject to binary_constraint_1 {bus in POWER_BUSES, ts in USED_TS}:
start[bus, ts] - on[bus, ts]<= 0;

subject to binary_constraint_2 {bus in POWER_BUSES, ts in USED_TS: ts>0}:
start[bus, ts] - on[bus, ts] - U*(1-on[bus, ts - 1]) + U >= 0;

subject to binary_constraint_3 {bus in POWER_BUSES, ts in USED_TS}:
start[bus, ts] >= 0;
```

```
subject to binary_constraint_4 {bus in POWER_BUSES, ts in USED_TS: ts>0}:
start[bus, ts] - U*(1-on[bus, ts - 1]) <= 0;
```

```
# expand power_balance;

minimize production_cost:

sum{bus in POWER_BUSES,ts in USED_TS} (

    COST[B_TECH[bus]]*u[bus,ts] + UNITCOST[B_TECH[bus]]*start[bus,ts]);

solve;

display production_cost;
```