Ask the Expert: What Are the Ways to Diagnose Infeasibility?

Rob Pratt

April 20, 2021

Rob Pratt

Senior Manager, Advanced Analytics R&D, SAS

Rob Pratt has worked at SAS since 2000 and currently works in the operations research department within SAS R&D's advanced analytics division. He manages a team of developers responsible for the optimization modeling language, constraint programming, project management and discrete-event simulation.

OPTMODEL: Background

- SAS/OR in SAS 9: PROC OPTMODEL
- SAS Optimization in SAS Viya: PROC OPTMODEL, runOptmodel action
- Algebraic modeling language with optimization-oriented syntax:
  - Variables, objectives, constraints
  - Algebraic expression of functions
  - Parameters, arrays, index sets
- Standard and user-defined functions
- Separation between model structure and instance data
- Flexible input/output: read from and create an arbitrary number of data sets
- Direct access to LP, MILP, QP, NLP, CLP, network, and black-box solvers
- Programming statements to support customized algorithms

Scheduling Problem

https://communities.sas.com/t5/Mathematical-Optimization/Project-schedule-via-LP-with-Proc-Optmodel/m-p/698176

- Given set A of activities, durations d_i, and set P of precedence pairs
- Schedule activities to complete in minimum time (*makespan*)
- Can use PROC CPM or PROC CLP, but user requested PROC OPTMODEL
- Decision variables: Start time s_i for activity $i \in A$
- Objective: Minimize makespan $\max_{i \in A} \{s_i + d_i\}$
- Constraints: For each precedence pair $(i, j) \in P$, finish activity i before starting activity j

Linear Programming Formulation

Minimax optimization formulation:

$$
\begin{array}{llll}
\text{minimize} & \max_{i \in A}\{s_i + d_i\} & & (1) \\
\text{subject to} & s_i + d_i \leq s_j & (i,j) \in P & (2) \\
& s_i \geq 0 & i \in A & (3)
\end{array}
$$

Linearization (LINEARIZE option can automate this):

$$
\begin{array}{llll}
\text{minimize} & f & & (4) \\
\text{subject to} & s_i + d_i \leq f & i \in A & (5) \\
& s_i + d_i \leq s_j & (i,j) \in P & (6) \\
& s_i \geq 0 & i \in A & (7)
\end{array}
$$

OPTMODEL Code

$$\begin{array}{lll} \text{minimize} & f & \\ \text{subject to} & s_i + d_i \le f & i \in A \\ & s_i + d_i \le s_j & (i,j) \in P \\ & s_i \ge 0 & i \in A \end{array}$$

```
/* declare variables */
var Start {ACTIVITIES} >= 0;
var Finish;

/* declare objective */
min Makespan = Finish;

/* declare constraints */
con MinMax {i in ACTIVITIES}:
   Start[i] + duration[i] <= Finish;
con Precedence {<i,j> in PRECEDENCES}:
   Start[i] + duration[i] <= Start[j];
```

OPTMODEL Code

$$
\begin{array}{lll}
\text{minimize} & f & \\
\text{subject to} & s_i + d_i \leq f & i \in A \\
& s_i + d_i \leq s_j & (i,j) \in P \\
& s_i \geq 0 & i \in A
\end{array}
$$

```
/* declare variables */
var Start {ACTIVITIES} >= 0;
var Finish;

/* declare objective */
min Makespan = Finish;

/* declare constraints */
con MinMax {i in ACTIVITIES}:
   Start[i] + duration[i] <= Finish;
con Precedence {<i,j> in PRECEDENCES}:
   Start[i] + duration[i] <= Start[j];
```

OPTMODEL Code

minimize f

subject to

$$s_i + d_i \leq f \quad i \in A$$

$$s_i + d_i \leq s_j \quad (i,j) \in P$$

$$s_i \geq 0 \quad i \in A$$

```
/* declare variables */
var Start {ACTIVITIES} >= 0;
var Finish;

/* declare objective */
min Makespan = Finish;

/* declare constraints */
con MinMax {i in ACTIVITIES}:
   Start[i] + duration[i] <= Finish;
con Precedence {<i,j> in PRECEDENCES}:
   Start[i] + duration[i] <= Start[j];
```


sas

OPTMODEL Code

$$
\begin{array}{lll}
\text{minimize} & f & \\
\text{subject to} & s_i + d_i \leq f & i \in A \\
 & s_i + d_i \leq s_j & (i,j) \in P \\
 & s_i \geq 0 & i \in A
\end{array}
$$

```
/* declare variables */
var Start {ACTIVITIES} >= 0;
var Finish;

/* declare objective */
min Makespan = Finish;

/* declare constraints */
con MinMax {i in ACTIVITIES}:
   Start[i] + duration[i] <= Finish;
con Precedence {<i,j> in PRECEDENCES}:
   Start[i] + duration[i] <= Start[j];
```

OPTMODEL Code

$$
\begin{array}{lll}
\text{minimize} & f & \\
\text{subject to} & s_i + d_i \leq f & i \in A \\
 & s_i + d_i \leq s_j & (i,j) \in P \\
 & s_i \geq 0 & i \in A
\end{array}
$$

```
/* declare variables */
var Start {ACTIVITIES} >= 0;
var Finish;

/* declare objective */
min Makespan = Finish;

/* declare constraints */
con MinMax {i in ACTIVITIES}:
   Start[i] + duration[i] <= Finish;
con Precedence {<i,j> in PRECEDENCES}:
   Start[i] + duration[i] <= Start[j];
```

Input Data

```
data ActivityData;
   input activity $ duration;
   datalines;
A 2
B 3
C 3
D 4
E 8
F 6
G 2
;
```

```
data PrecedenceData;
   input activity $ successor $;
   datalines;
A B
A C
B F
E F
C D
D E
F G
;
```

OPTMODEL Code

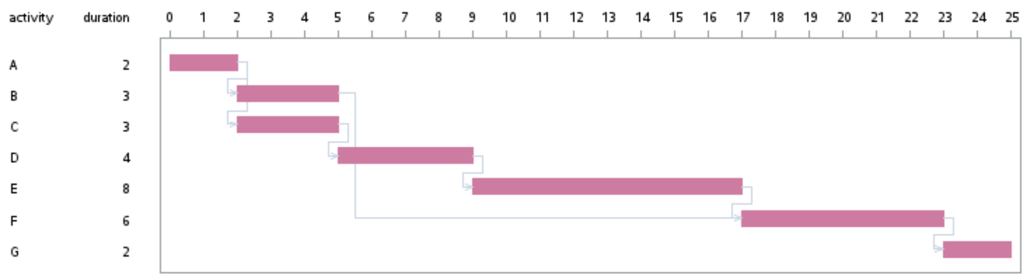
```
proc optmodel;
   /* declare parameters and read data */
   set <str> ACTIVITIES;
   num duration {ACTIVITIES};
   read data ActivityData into ACTIVITIES=[activity] duration;
   set <str,str> PRECEDENCES;
   read data PrecedenceData into PRECEDENCES=[activity successor];

   /* declare variables */
   var Start {ACTIVITIES} >= 0;
   var Finish;

   /* declare objective */
   min Makespan = Finish;

   /* declare constraints */
   con MinMax {i in ACTIVITIES}:
      Start[i] + duration[i] <= Finish;
   con Precedence {<i,j> in PRECEDENCES}:
      Start[i] + duration[i] <= Start[j];

   /* call LP solver */
   solve;
```

OPTMODEL Log

```
NOTE: The problem has 8 variables (1 free, 0 fixed).
NOTE: The problem has 14 linear constraints (14 LE, 0 EQ, 0 GE, 0 range).
NOTE: The problem has 28 linear constraint coefficients.
NOTE: The problem has 0 nonlinear constraints (0 LE, 0 EQ, 0 GE, 0 range).
NOTE: The OPTMODEL presolver is disabled for linear problems.
NOTE: The LP presolver value AUTOMATIC is applied.
NOTE: The LP presolver time is 0.00 seconds.
NOTE: The LP presolver removed all variables and constraints.
NOTE: Optimal.
NOTE: Objective = 25.
```

Optimal Solution

[1]	Start	duration
A	0	2
B	2	3
C	2	3
D	5	4
E	9	8
F	17	6
G	23	2



Random Data

20 activities, random durations, 42 random precedence pairs

```
NOTE: There were 20 observations read from the data set WORK.ACTIVITYDATA.
NOTE: There were 42 observations read from the data set WORK.PRECEDENCEDATA.
NOTE: The problem has 21 variables (1 free, 0 fixed).
NOTE: The problem has 62 linear constraints (62 LE, 0 EQ, 0 GE, 0 range).
NOTE: The problem has 124 linear constraint coefficients.
NOTE: The problem has 0 nonlinear constraints (0 LE, 0 EQ, 0 GE, 0 range).
NOTE: The OPTMODEL presolver is disabled for linear problems.
NOTE: The LP presolver value AUTOMATIC is applied.
NOTE: The LP presolver time is 0.00 seconds.
NOTE: Constraint Precedence[A3,A19] causes the problem to be infeasible.
NOTE: The LP presolver found this problem to be infeasible.
NOTE: Infeasible.
NOTE: The IIS= option can be used to help diagnose the cause of infeasibility.
```

Irreducible Infeasible Set (IIS)

- Whole problem is infeasible, but why?
- Want smaller infeasible set of constraints to diagnose
- Irreducible: if you remove any constraint from set, resulting set is feasible
- IIS= option for LP, MILP, QP, NLP solvers

```
solve with lp / iis=true;
expand / iis;
```

IIS Output

```
NOTE: The LP solver is called.
NOTE: The IIS= option is enabled.
                          Objective
      Phase Iteration         Value          Time
       P 1          1    2.376122E+01          0
       P 1         28    5.711673E+00          0
NOTE: Applying the IIS sensitivity filter.
NOTE: The sensitivity filter removed 43 constraints and 20 variable bounds.
NOTE: Applying the IIS deletion filter.
NOTE: Processing constraints.
...
NOTE: The deletion filter removed 16 constraints and 0 variable bounds.
NOTE: The IIS= option found this problem to be infeasible.
NOTE: The IIS= option found an irreducible infeasible set with 0 variables and 3 constraints.
NOTE: The IIS solve time is 0.11 seconds.


Constraint Precedence[A9,A10]: Start[A9] - Start[A10] <= -0.05670104
Constraint Precedence[A10,A20]: Start[A10] - Start[A20] <= -0.079830517
Constraint Precedence[A20,A9]: Start[A20] - Start[A9] <= -0.436688164
```

Remove One Identified Constraint and Solve Again

```
PRECEDENCES = PRECEDENCES diff {<'A10','A20'>};
solve;
```

```
NOTE: The problem has 21 variables (1 free, 0 fixed).
NOTE: The problem has 61 linear constraints (61 LE, 0 EQ, 0 GE, 0 range).
NOTE: The problem has 122 linear constraint coefficients.
NOTE: The problem has 0 nonlinear constraints (0 LE, 0 EQ, 0 GE, 0 range).
NOTE: The OPTMODEL presolver is disabled for linear problems.
NOTE: The LP presolver value AUTOMATIC is applied.
NOTE: The LP presolver time is 0.00 seconds.
NOTE: Constraint Precedence[A3,A19] causes the problem to be infeasible.
NOTE: The LP presolver found this problem to be infeasible.
NOTE: Infeasible.
NOTE: The IIS= option can be used to help diagnose the cause of infeasibility.
```

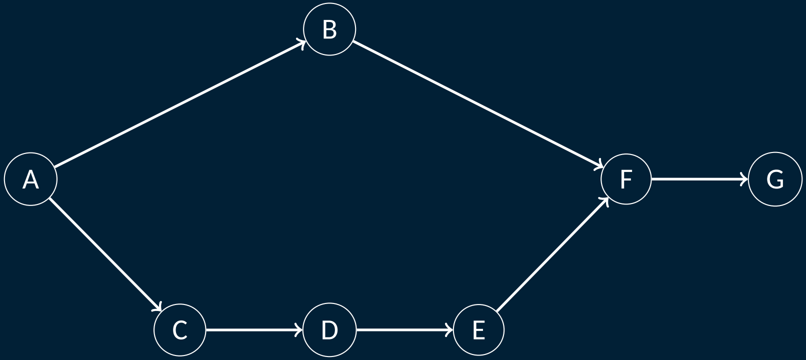
Find Another IIS

```
solve with lp / iis=true;
expand / iis;
```

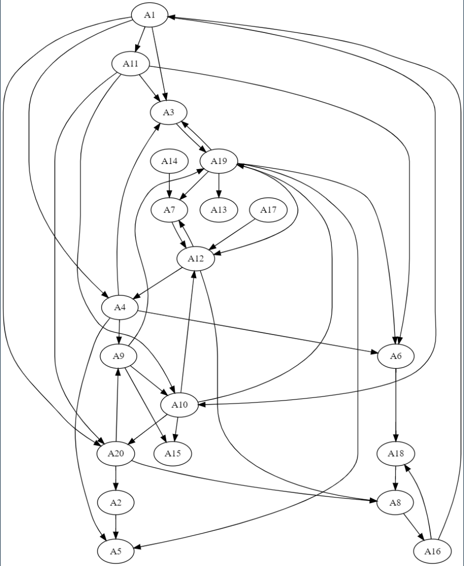
```
NOTE: The LP solver is called.
NOTE: The IIS= option is enabled.
                           Objective
      Phase Iteration        Value         Time
       P 1          1    2.368139E+01         0
       P 1         24    5.631828E+00         0
NOTE: Applying the IIS sensitivity filter.
NOTE: The sensitivity filter removed 43 constraints and 20 variable bounds.
NOTE: Applying the IIS deletion filter.
NOTE: Processing constraints.
...
NOTE: The deletion filter removed 16 constraints and 0 variable bounds.
NOTE: The IIS= option found this problem to be infeasible.
NOTE: The IIS= option found an irreducible infeasible set with 0 variables and 2 constraints.
NOTE: The IIS solve time is 0.04 seconds.


Constraint Precedence[A7,A12]: Start[A7] - Start[A12] <= -0.337594608
Constraint Precedence[A12,A7]: Start[A12] - Start[A7] <= -0.225850861
```

Network Model

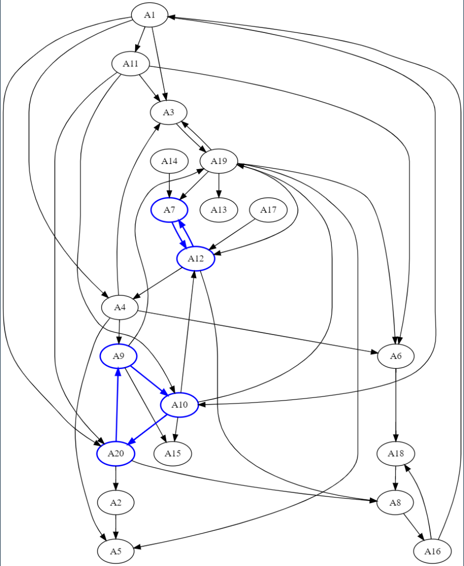
- Node for each activity
- Directed arc for each precedence pair



Random Network



Cycles Cause Infeasibility



Network Algorithms

- Biconnected components and articulation points
- Clique enumeration
- Connected components
- Cycle enumeration
- Path enumeration
- Transitive closure
- Linear assignment
- Minimum-cost network flow
- Minimum cut
- Minimum spanning tree
- Shortest path
- Traveling salesman problem
- Vehicle routing problem

Network Solver: Find One Cycle

```
set <num,num,str,str> CYCLESLINKS; /* id, order, from, to */
solve with network / cycle
   direction=directed links=(include=PRECEDENCES) out=(cycleslinks=CYCLESLINKS);
put CYCLESLINKS=;
```

```
NOTE: The number of nodes in the input graph is 20.
NOTE: The number of links in the input graph is 42.
NOTE: The network solver is called.
NOTE: Processing cycle enumeration using 1 threads across 1 machines.
NOTE: Required number of cycles found. You can increase this value using the MAXCYCLES=
option.
NOTE: The algorithm found 1 cycles.
NOTE: Processing cycle enumeration used 0.00 (cpu: 0.00) seconds.
CYCLESLINKS={<1,1,'A1','A3'>,<1,2,'A3','A19'>,<1,3,'A19','A6'>,<1,4,'A6','A18'>,
<1,5,'A18','A8'>,<1,6,'A8','A16'>,<1,7,'A16','A1'>}
```

Network Solver: Find All Cycles

```
solve with network / cycle=(maxcycles=all)
   direction=directed links=(include=PRECEDENCES) out=(cycleslinks=CYCLESLINKS);
set CYCLES init {};
set <str,str> PRECEDENCES_cycle {CYCLES} init {};
for {<id,order,from,to> in CYCLESLINKS} do;
   CYCLES = CYCLES union {id};
   PRECEDENCES_cycle[id] = PRECEDENCES_cycle[id] union {<from,to>};
end;
```

```
NOTE: The number of nodes in the input graph is 20.
NOTE: The number of links in the input graph is 42.
NOTE: The network solver is called.
NOTE: Processing cycle enumeration using 1 threads across 1 machines.
NOTE: The algorithm found 96 cycles.
NOTE: Processing cycle enumeration used 0.00 (cpu: 0.00) seconds.
```

sas

Remove Links to Become Feasible: Cycle Formulation

```
/* find minimum number of links to remove */
var IsRemoved {PRECEDENCES} binary;
min NumRemoved = sum {<i,j> in PRECEDENCES} IsRemoved[i,j];
/* force at least one link to be removed from each cycle */
con Cover {c in CYCLES}:
   sum {<i,j> in PRECEDENCES_cycle[c]} IsRemoved[i,j] >= 1;
solve;
```

```
NOTE: The problem has 42 variables (0 free, 0 fixed).
NOTE: The problem has 42 binary and 0 integer variables.
NOTE: The problem has 96 linear constraints (0 LE, 0 EQ, 96 GE, 0 range).
NOTE: The problem has 801 linear constraint coefficients.
NOTE: The problem has 0 nonlinear constraints (0 LE, 0 EQ, 0 GE, 0 range).
NOTE: The OPTMODEL presolver is disabled for linear problems.
NOTE: The initial MILP heuristics are applied.
NOTE: The MILP presolver value AUTOMATIC is applied.
NOTE: The MILP presolver removed all variables and constraints.
NOTE: Optimal.
NOTE: Objective = 5.
```

Removed Links in Optimal Solution

[1]	[2]	IsRemoved
A12	A4	1
A3	A19	1
A7	A12	1
A8	A16	1
A9	A10	1

Verify That All Cycles are Eliminated

```
put (card(PRECEDENCES))=;
PRECEDENCES = {<i,j> in PRECEDENCES: IsRemoved[i,j].sol < 0.5};
put (card(PRECEDENCES))=;
solve with network / cycle
   direction=directed links=(include=PRECEDENCES) out=(cycleslinks=CYCLESLINKS);
```

```
CARD(PRECEDENCES)=42
CARD(PRECEDENCES)=37
NOTE: The number of nodes in the input graph is 20.
NOTE: The number of links in the input graph is 37.
NOTE: The network solver is called.
NOTE: Processing cycle enumeration using 1 threads across 1 machines.
NOTE: The algorithm found 0 cycles.
NOTE: Processing cycle enumeration used 0.00 (cpu: 0.00) seconds.
```

Remove Precedences to Become Feasible: Big-M Formulation

Original constraint:

```
var Start {ACTIVITIES} >= 0;
con Precedence {<i,j> in PRECEDENCES}:
   Start[i] + duration[i] <= Start[j];
```

Binary variables and relaxed (big-M) constraint:

```
var Start {ACTIVITIES} >= 0 <= sum {i in ACTIVITIES} duration[i];
var IsRemoved {PRECEDENCES} binary;
min NumRemoved = sum {<i,j> in PRECEDENCES} IsRemoved[i,j];
con Precedence {<i,j> in PRECEDENCES}:
   Start[i] + duration[i] - Start[j]
<= (Start[i].ub + duration[i] - Start[j].lb) * IsRemoved[i,j];

solve;
print {<i,j> in PRECEDENCES: IsRemoved[i,j].sol > 0.5} IsRemoved;
```

OPTMODEL Log

```
NOTE: The problem has 63 variables (1 free, 0 fixed).
NOTE: The problem has 42 binary and 0 integer variables.
NOTE: The problem has 62 linear constraints (62 LE, 0 EQ, 0 GE, 0 range).
NOTE: The problem has 166 linear constraint coefficients.
NOTE: The problem has 0 nonlinear constraints (0 LE, 0 EQ, 0 GE, 0 range).
NOTE: The OPTMODEL presolver is disabled for linear problems.
NOTE: The initial MILP heuristics are applied.
NOTE: The MILP presolver value AUTOMATIC is applied.
NOTE: The MILP presolver removed 7 variables and 21 constraints.
NOTE: The MILP presolver removed 50 constraint coefficients.
NOTE: The MILP presolver modified 8 constraint coefficients.
NOTE: The presolved problem has 56 variables, 41 constraints, and 116 constraint coefficients.
NOTE: The MILP solver is called.
NOTE: The parallel Branch and Cut algorithm is used.
NOTE: The Branch and Cut algorithm is using up to 4 threads.
          Node   Active   Sols    BestInteger      BestBound      Gap    Time
             0        1      4      6.0000000      0.6530030  818.83%      0
             0        1      4      6.0000000      4.0799396   47.06%      0
             0        1      5      5.0000000      4.0799396   22.55%      0
             0        1      5      5.0000000      4.0799396   22.55%      0
NOTE: The MILP solver added 15 cuts with 43 cut coefficients at the root.
NOTE: Optimal.
NOTE: Objective = 5.
```

Removed Precedences in Optimal Solution

[1]	[2]	IsRemoved
A12	A4	1
A12	A7	1
A19	A3	1
A20	A9	1
A8	A16	1

Remove Precedences to Become Feasible: Indicator Formulation

Big-M constraint:

```
con Precedence {<i,j> in PRECEDENCES}:
   Start[i] + duration[i] - Start[j]
<= (Start[i].ub + duration[i] - Start[j].lb) * IsRemoved[i,j];
```

Indicator constraint:

```
con Precedence {<i,j> in PRECEDENCES}:
   IsRemoved[i,j] = 0 implies Start[i] + duration[i] <= Start[j];
```

OPTMODEL Log

```
NOTE: The problem has 63 variables (1 free, 0 fixed).
NOTE: The problem has 42 binary and 0 integer variables.
NOTE: The problem has 62 linear constraints (62 LE, 0 EQ, 0 GE, 0 range).
NOTE: The problem has 124 linear constraint coefficients.
NOTE: The problem has 0 nonlinear constraints (0 LE, 0 EQ, 0 GE, 0 range).
NOTE: The OPTMODEL presolver is disabled for linear problems.
NOTE: The initial MILP heuristics are applied.
NOTE: The MILP presolver value AUTOMATIC is applied.
NOTE: The MILP presolver removed 7 variables and 21 constraints.
NOTE: The MILP presolver removed 50 constraint coefficients.
NOTE: The MILP presolver modified 8 constraint coefficients.
NOTE: The presolved problem has 56 variables, 41 constraints, and 116 constraint coefficients.
NOTE: The MILP solver is called.
NOTE: The parallel Branch and Cut algorithm is used.
NOTE: The Branch and Cut algorithm is using up to 4 threads.
          Node   Active   Sols    BestInteger      BestBound      Gap    Time
             0        1      4      6.0000000      0.6530030  818.83%       0
             0        1      4      6.0000000      4.0799396   47.06%       0
             0        1      5      5.0000000      4.0799396   22.55%       0
             0        1      5      5.0000000      4.0799396   22.55%       0
NOTE: The MILP solver added 15 cuts with 43 cut coefficients at the root.
NOTE: Optimal.
NOTE: Objective = 5.
```

Introduce Nonnegative Slack Variables and Minimize Infeasibility

Original constraint:

```
con Precedence {<i,j> in PRECEDENCES}:
   Start[i] + duration[i] <= Start[j];
```

Slack variables and relaxed constraint:

```
var Slack {PRECEDENCES} >= 0;
min Infeasibility = sum {<i,j> in PRECEDENCES} Slack[i,j];
con Precedence {<i,j> in PRECEDENCES}:
   Start[i] + duration[i] <= Start[j] + Slack[i,j];

solve;
print {<i,j> in PRECEDENCES: Slack[i,j].sol > 0} Slack;
```

OPTMODEL Log

```
NOTE: The problem has 63 variables (1 free, 0 fixed).
NOTE: The problem has 62 linear constraints (62 LE, 0 EQ, 0 GE, 0 range).
NOTE: The problem has 166 linear constraint coefficients.
NOTE: The problem has 0 nonlinear constraints (0 LE, 0 EQ, 0 GE, 0 range).
NOTE: The OPTMODEL presolver is disabled for linear problems.
NOTE: The LP presolver value AUTOMATIC is applied.
NOTE: The LP presolver time is 0.00 seconds.
NOTE: The LP presolver removed 11 variables and 11 constraints.
NOTE: The LP presolver removed 29 constraint coefficients.
NOTE: The presolved problem has 52 variables, 51 constraints, and 137 constraint coefficients.
NOTE: The LP solver is called.
NOTE: The Dual Simplex algorithm is used.
                               Objective
      Phase Iteration            Value         Time
       D 2          1     0.000000E+00            0
       P 2         39     5.711688E+00            0
NOTE: Optimal.
NOTE: Objective = 5.7116881642.
NOTE: The Dual Simplex solve time is 0.00 seconds.
```

Positive Slack Variables in Optimal Solution

[1]	[2]	Slack
A1	A11	0.581700
A10	A20	0.028167
A11	A10	0.051663
A12	A4	0.595692
A12	A7	0.200592
A16	A1	0.846208
A18	A8	0.296669
A19	A12	0.018005
A20	A9	0.465222
A3	A19	0.096185
A4	A9	0.074706
A6	A18	0.484932
A7	A12	0.362854
A8	A16	1.529263
A9	A10	0.079831

Summary

- Irreducible Infeasible Set (IIS)
- Network solver (cycle algorithm)
- Network solver output as MILP solver input
- Relax constraints as little as possible
  - Big-M constraints
  - Indicator constraints
  - Nonnegative slack variables

support.sas.com

Explore Helpful Resources

Ask the Expert
View other user webinars that provide insights into using SAS products to make your job easier.

FREE Training
Learn from home — free for 30 days. Get software labs to practice and online support if needed.

SAS Support Communities
Ask questions, get answers and share insights with SAS users.

SAS Analytics Explorers
An exclusive platform to collaborate, learn and share your expertise. Gain access to a diverse network to advance your career. Special rewards and recognition exclusively for SAS users.

SAS Users YouTube Channel
A plethora of videos on hundreds of topics, just for SAS users.

Newsletters
Get the latest SAS news plus tips, tricks and more.

Users Groups
Meet local SAS users, network and exchange ideas — virtually.

SAS Profile
If you haven't already done so, create your SAS Profile to access free training, SAS Support Communities, technical support, software downloads, newsletters and more.

Thank you

for joining us for
this SAS webinar