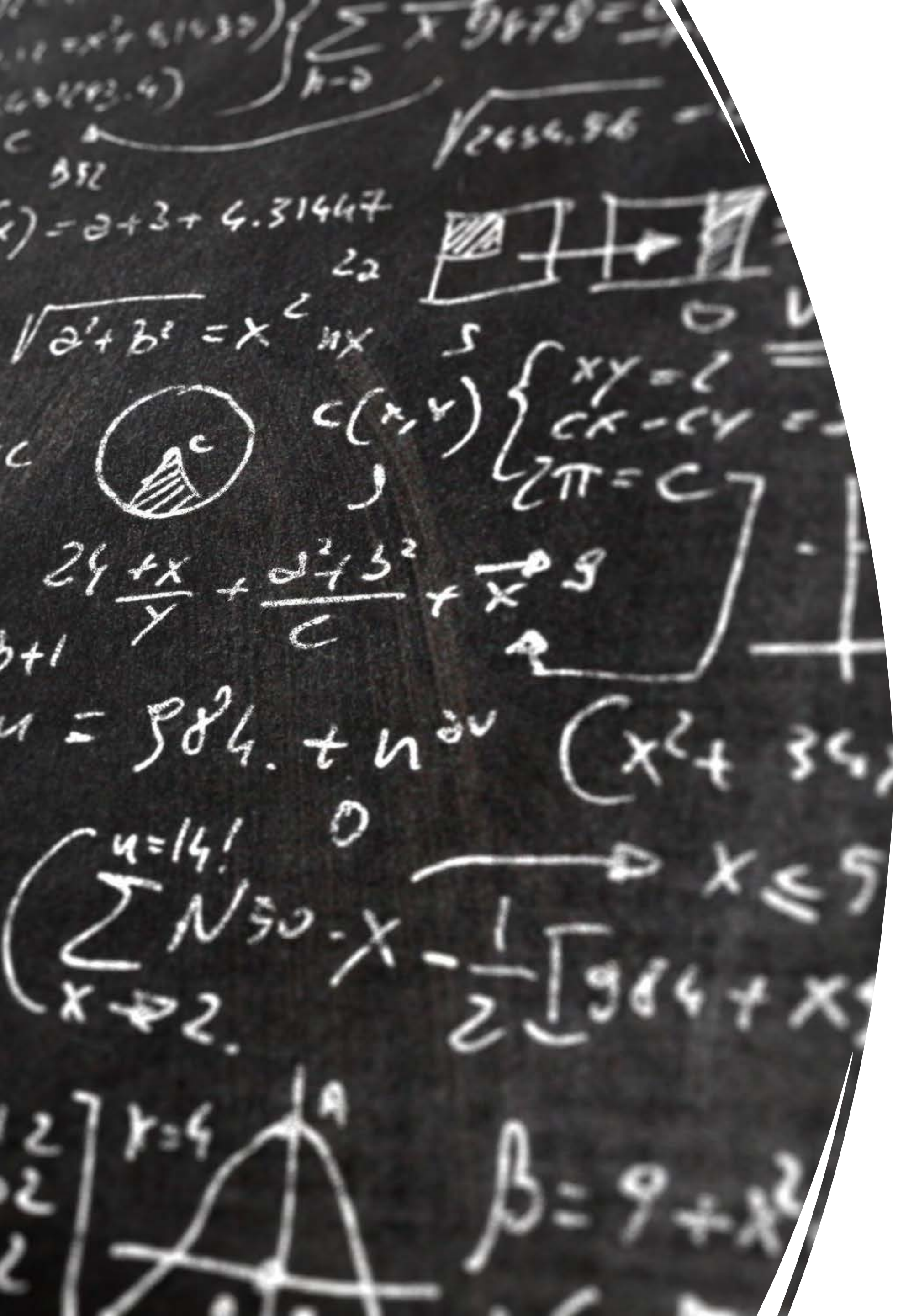


Taming Mathematical Programming in APL (TaMPA)

Stephen M. Mansour, PhD,
Misericordia University

Dyalog 22, Olhao, Portugal

October 12, 2022

Dyalog App Library

- TAMSTAT – TAMing STATistics Package

  New features include

  - Non-Parametric Statistics
  - New Anova Designs
  - Theoretical Probability Graphics
- ADAGE – A Dyalog APL Generalized Equation Solver
- TAMSTOP –TAMing STock OPtions
- TaMPA – Taming Mathematical Programming in APL

What is Mathematical Programming (MP)?

- A mathematical program (MP) has three components:
  1. Decision Variables – e.g. How much product to make $X_1, X_2, \dots, X_n$
  2. Objective Function - e.g. profit $f(X_1, X_2, \dots, X_n)$
  3. Constraints – e.g. resource limitations $g_k(X_1, X_2, \dots, X_n) \leq b_k$
- Linear Programming (LP) is a special case of mathematical programming where

  $f(X_1, X_2, \dots, X_n) = \sum_{i=1}^{n} c_i X_i$, and

  $g_k(X_1, X_2, \dots, X_n) = \sum_{i=1}^{n} a_{ki} X_i$.

Linear Programming

- In linear programming we can replace summations with matrix notation.
- The matrix notation for linear programming (LP) is:

$$\max_{x} c'x \ \text{ subject to } \ Ax \leq b$$

- Where c is the vector of coefficients in the objective function, A is a matrix of coefficients for the constraints, b is a vector of resource limitations, and x is a vector of decision variables.
- APL is a natural way to handle linear programming due to its array handling capabilities.

APL Syntax for LP/NLP

- We propose the following syntax for linear programming (LP) or (NLP) :

```
  NS ← ⌈ optimize c x subjectTo A x ≤     b
  NS ← ⌊ optimize f x subjectTo G x ≥     0
--------- -------------- ------------------ ---------   ---
    ↑         ↑                  ↑              ↑          ↑
  Result     Runs the          builds a       creates a  right
 Namespace   LP/NLP            tableau        namespace  arg
 maximize ← ⌈ optimize   ⍝ Monadic operator with left
 minimize ← ⌊ optimize   ⍝ oper. produces max or min
```

Key: Array Function Operator NameSpace

Example 1: Blue Ridge Hot Tubs

Hot Tub Brand:	Aqua-Spa	Hydro-Luxe	Typhoon-Lagoon	Resources Available:
Unit Profit:	$350	$300	$320	
Pumps Required	1	1	1	200 pumps
Labor Required	9 hours	6 hours	8 hours	1566 hours
Tubing Needed	12 feet	16 feet	13 feet	2880 feet

Questions to ask

- How many hot tubs of each type should Blue Ridge produce?
- What is the maximum profit?
- How much additional profit can be realized with additional resources?
- What are the costs of deviating from the optimal solution?

Problem formulation in mathematical notation

X_1 = Number of Aqua-Spas to produce
X_2 = Number of Hydro-Luxes to produce
X_3 = Number of Typhoon-Lagoons to produce

Maximize $350X_1 + 300X_2 + 320X_3$

Subject to:

$$\begin{aligned} X_1 + X_2 + X_3 &\leq 200 \\ 9X_1 + 6X_2 + 8X_3 &\leq 1{,}566 \\ 12X_1 + 16X_2 + 13X_3 &\leq 2{,}880 \\ X_1, X_2, X_3 &\geq 0 \end{aligned}$$

Let

$$x = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \quad c = \begin{bmatrix} 350 \\ 300 \\ 320 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 9 & 6 & 8 \\ 12 & 16 & 13 \end{bmatrix} \quad b = \begin{bmatrix} 200 \\ 1566 \\ 2880 \end{bmatrix}$$

Maximize $c'x$ subject to $Ax \leq b$

Problem formulation using TAMPA

```
   c←350 300 320    ⍝ Objective coefficients
  A←3 3⍴1 1 1 9 6 8 12 16 13    ⍝ Constraint coefficients
 1  1  1
 9  6  8
12 16 13
  b←200 1566 2880                 ⍝ Resource limitations
  NS←maximize c x subjectTo A x ≤ b ⍝ Perform the LP
  NS.Decision                    ⍝ Produce 122 Aqua Spas and 78 Hydro-Luxes
122 78 0
  NS.ShadowPrice   ⍝ Each add'l pump available contributes $200 to profit
200 16.66666667 0 ⍝ Each add'l labor hour contributes $16.67 profit
  NS.ReducedCost   ⍝ Each Typhoon-Lagoon produced reduces profit by $13.33
0 0 ¯13.33333333
```

NS←maximize c x subjectTo A x ≤ b

```
      NS← A x≤b
      ↓NS.⎕nl 2 3
A     b    rel
      NS.rel
≤
      NS←  c x subjectTo NS
      ↓NS.⎕nl 2 3
A    T     b       c     rel
      NS.T      ⍝ Tableau
 200   1   1   1  1  0  0
1566   9   6   8  0  1  0
2880  12  16  13  0  0  1
```

```
      NS←maximize NS
      NS.⎕nl 2 3
A
Decision
Objective
ReducedCost
ShadowPrice
T
b
c
optimum
rel
```

The x operator depends upon the structure and class of its operands

```
x←{⍝⍺⍺:  A Matrix  a[i;j] = coefficient of ith constraint, jth variable  or function array
   ⍝⍵⍵:  Relation, e.g. ≤ or train (≤,=,≥) or subjectTo function
   ⍝⍵:   b - right hand side of constraints
   ⍝←:   Namespace containing values
     2=⎕NC'⍵':⍺⍺{NS←⎕NS''           ⍝ Create namespace
         NS.b←⍵ ⋄ NS.rel←⍵⍵         ⍝ Assign values
         NS.A←⍺⍺ ⋄ NS}⍵⍵ ⍵
     NS←⍵                           ⍝ Right argument is namespace
     3=⎕NC'⍺⍺':⍺⍺{⍵.c←⍺⍺ ⋄ G←⍵⍵    ⍝ Is left operand an array?
         ⍵}⍵⍵ ⍵                     ⍝ Is objective a function?
     NS.T←⍵⍵ ⍵                      ⍝ Build tableau
     c←NS.⍺⍺                        ⍝ Coefficients
     NS.c←c
     NS}
```

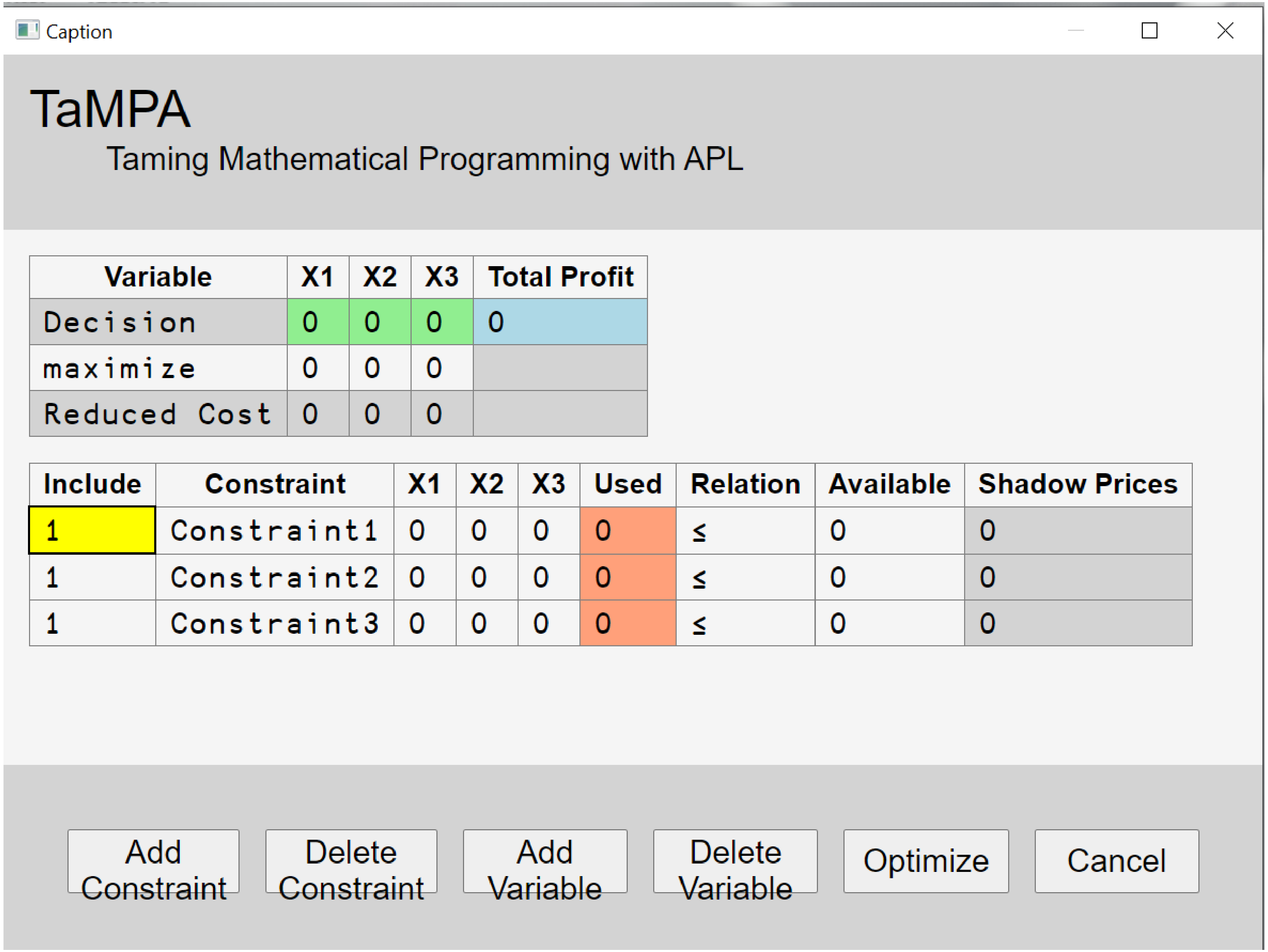
The optimize operator

```
optimize←{
⍝αα:  ⌈ - Maximize; ⌊ - Minimize; ∩ - Goal Programming
⍝ω:   A - Namespace containing Tableau
⍝ω:   b - right hand side of constraints
⍝⍎:
    c←ω.c                                 ⍝ Save original coefficients
    ω.c←c×¯1*~b←αα 0.5                    ⍝ Negate them if minimum
    NS←{                                  ⍝ Perform the optimization
        NS←ω
        3=⎕NC'NS.A':NLP ω                 ⍝
        r←(≢ω.b)⍴↑0 1-.ω.rel 1 0          ⍝ Relation:   1 ≤, 0 =, ¯1 ≥
        M←1000×⌈/|ω.c                     ⍝ Big M
        c←-ω.c                            ⍝ Use neg coefficients
        T0←(↑⌽⍴ω.T)↑0,c,M×r≠1             ⍝ Assemble tableau
        T0-←M×+⌿(r≠1)⌿ω.T                 ⍝ Adjust first row to remove bigM
        Z←Primal⍣{0∧.≤1↓ω[0;]}T0⍪ω.T      ⍝ Apply primal until reduced costs ≥ 0
        B←Z[;1+⍳≢c]                       ⍝ Determine the basis
        B←B×⍤1⊢B[0;]=0                    ⍝ Zero out non-basis columns
        ω.Objective←↑Z                    ⍝ Objective
        ω.Decision←Z[;0]+.×B              ⍝ Primal Solution (Decision Variables)
        J←(1+≢c)+⍳≢ω.b                    ⍝ Indices for Dual Solution
        ((r=¯1)/J)←(-r+.=¯1)↑⍳1⊃⍴Z        ⍝ Get last n columns for equality constraints
        ω.ShadowPrice←Z[0;J]-M×r=0        ⍝ Dual Solution (Shadow Prices)
        ω.ReducedCost←-Z[0;1+⍳≢c]         ⍝ Reduced Costs
        ω
    }ω
```

The primal algorithm does most of the work

```
Primal←{
 ⍝∇ Full Tableau Implementation of the Simplex Method
 ⍝∊ Written by Steve Mansour 9/23/2005
 ⍝⍵ Starting Tableau (Numeric Matrix)
 ⍝← Revised Tableau (Numeric Matrix)
     x←1↓⍵[;0]                 ⍝ x-vector - solution to Ax=b
     c←1↓⍵[0;]                 ⍝ Reduced cost vector
      ⍝j←1+c⍳⌊/c                 ⍝ Index of Entering variable
     α←1+c⍳⌊/c                 ⍝ Index of Entering variable
     u←1↓⍵[;α]                 ⍝ jth column of A matrix
     r←x÷u+u=0                 ⍝ Get smallest ratio
     p←u>0                     ⍝ Restrict to u>0
     b←p∧r=⌊/p/r               ⍝ Bland's Anticycling Rule
     l←(b\0=⍋⍋¯1⌽(0,b)⌿⍵)⍳1   ⍝ Index of Leaving variable
     pr←⍵[l+1;]÷u[l]           ⍝ Normalize pivot row
     z←⍵-⍵[;↑α]∘.×pr           ⍝ Subtract various multiples of pivot row
     z[1+l;]←pr                ⍝ Replace with normalized value
     z                         ⍝ New Tableau
 }
```

```
RunTAMPA←{
    ⍝ Add Insert/Delete buttons
    ⍝⍎ GUI.RunTAMPA HotTubs
    ⍝⍎ GUI.RunTAMPA WeedWacker
    ⍝⍎ GUI.RunTAMPA Blank
    w←{0≡⍵:##.Blank ⋄
        2=≢⍵:Init ⍵ ⋄ ⍵}⍵
    ⎕THIS.H←#.Abacus.Main
    #.Abacus.DialogBox.H←#.Abacus.Main
    f←#.Abacus.TriDocument.New 0
    hd←H.GetHeader f
    h1←hd H.New'h1' 'TaMPA'
    h2←hd H.New'h2' 'Taming Mathematical Programming with APL'
    m←H.GetMain f
    m.Content←#.Abacus.Grid.New ObjectiveGrid 2↑w
    m.Content,←#.Abacus.Grid.New(0⌷1⊃w)ConstraintGrid 2⊃w
    t←m.Content[1]
    BC←⍉H.BodyCells t
    BC[;3].class←⊂'orange'
    BC[;0].class←⊂'grey'
    ⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝ Temporary Fix ⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝
⍝    _←BC[;2].Content←⊂''
⍝    _←BC[;2]#.Abacus.DropDownList.New¨⊂,¨'<=' '=' '>='
    ⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝⍝
    BOPT←⍉H.BodyCells m.Content[0]
```


Caption
TaMPA
Taming Mathematical Programming with APL
Variable | X1 | X2 | X3 | Total Profit
Decision | 0 | 0 | 0 | 0
maximize | 0 | 0 | 0
Reduced Cost | 0 | 0 | 0
Include | Constraint | X1 | X2 | X3 | Used | Relation | Available | Shadow Prices
1 | Constraint1 | 0 | 0 | 0 | 0 | ≤ | 0 | 0
1 | Constraint2 | 0 | 0 | 0 | 0 | ≤ | 0 | 0
1 | Constraint3 | 0 | 0 | 0 | 0 | ≤ | 0 | 0
Add Constraint
Delete Constraint
Add Variable
Delete Variable
Optimize
Cancel

Caption

TaMPA

Taming Mathematical Programming with APL

Variable	AquaSpa	Hydro-Luxe	Typhoon-Lagoon	Total Profit
Decision	0	0	0	0
maximize	350	300	320	
Reduced Cost	0	0	0	

Include	Constraint	AquaSpa	Hydro-Luxe	Typhoon-Lagoon	Used	Relation	Available	Shadow Prices
1	Pumps	1	1	1	0	≤	200	0
1	Labor	9	6	8	0	≤	1566	0
1	Tubing	12	16	13	0	≤	2880	0

Add Constraint | Delete Constraint | Add Variable | Delete Variable | Optimize | Cancel

Caption

TaMPA

Taming Mathematical Programming with APL

Variable	AquaSpa	Hydro-Luxe	Typhoon-Lagoon	Total Profit
Decision	122	78	0	66100
maximize	350	300	320	
Reduced Cost	0	0	¯13.33333333	

Include	Constraint	AquaSpa	Hydro-Luxe	Typhoon-Lagoon	Used	Relation	Available	Shadow Prices
1	Pumps	1	1	1	200	≤	200	200
1	Labor	9	6	8	1566	≤	1566	16.66666667
1	Tubing	12	16	13	2712	≤	2880	0

Add Constraint | Delete Constraint | Add Variable | Delete Variable | Optimize | Cancel

Caption

Removing a Constraint

TaMPA

Taming Mathematical Programming with APL

Variable	AquaSpa	Hydro-Luxe	Typhoon-Lagoon	Total Profit
Decision	108	99	0	67500
maximize	350	300	320	
Reduced Cost	0	0	¯10.55555556	

Include	Constraint	AquaSpa	Hydro-Luxe	Typhoon-Lagoon	Used	Relation	Available	Shadow Prices
0	Pumps	1	1	1		≤	200	
1	Labor	9	6	8	1566	≤	1566	27.77777778
1	Tubing	12	16	13	2880	≤	2880	8.333333333

Add Constraint | Delete Constraint | Add Variable | Delete Variable | Optimize | Cancel

Caption

Adding a Constraint

TaMPA

Taming Mathematical Programming with APL

Variable	AquaSpa	Hydro-Luxe	Typhoon-Lagoon	Total Profit
Decision	113.2	73.6	13.2	65924
maximize	350	300	320	
Reduced Cost	0	0	0	

Include	Constraint	AquaSpa	Hydro-Luxe	Typhoon-Lagoon	Used	Relation	Available	Shadow Prices
1	Pumps	1	1	1	200	≤	200	176
1	Labor	9	6	8	1566	≤	1566	14
1	Tubing	12	16	13	2707.6	≤	2880	0
1	MaterialCost	600	500	400	110000	≤	110000	0.08

Add Constraint | Delete Constraint | Add Variable | Delete Variable | Optimize | Cancel

Example 2: Weedwacker Company – Make or Buy

- The company produces two types of lawn trimmers: Electric and Gas

	Electric Trimmers	Gas Trimmers	Total Hours Available
Production	0.20 hours	0.40 hours	10,000
Assembly	0.30 hours	0.50 hours	15,000
Packaging	0.10 hours	0.10 hours	5,000
Cost to Make	$55	$85	
Cost to Buy	$67	$95	
Number required	15,000	30,000	

Minimize $55M_1 + 85\,M_2 + 67\,B_1 + 95\,B_2$

ST $M_1 + B_1 = 30{,}000$

$M_2 + B_2 = 15{,}000$

$0.20M_1 + 0.40M_2 \leq 10{,}000$

$0.30M_1 + 0.50M_2 \leq 15{,}000$

$0.10M_1 + 0.10M_2 \leq 5{,}000$

$M_i, B_i \geq 0$

```
c←55 85 67 95
A←1 0 1 0 0 1 0 1 .2 .4 0 0
A,←.3 .5 0 0 .1 .1 0 0
A←5 4⍴A
b←30000 15000 10000 15000 5000
rel←=,=,≤,≤,≤
NS←minimize c x subjectTo A x rel b
NS.Decision
30000 10000 0 5000
    NS.Objective  ⍝ Total Cost
2975000
    NS.ShadowPrice
60 95 ¯25 0 0
      NS.ReducedCost
0 0 7 0
```

Caption

TaMPA

Taming Mathematical Programming with APL

Variable	M1	M2	B1	B2	Total Cost
Decision	30000	10000	0	5000	2975000
minimize	55	85	67	95	
Reduced Cost	0	0	7	0	

Include	Constraint	M1	M2	B1	B2	Used	Relation	Available	Shadow Prices
1	D1	1	0	1	0	30000	=	30000	60
1	D2	0	1	0	1	15000	=	15000	95
1	Production	0.2	0.4	0	0	10000	≤	10000	¯25
1	Assembly	0.3	0.5	0	0	14000	≤	15000	0
1	Packaging	0.1	0.1	0	0	4000	≤	5000	0

Add Constraint | Delete Constraint | Add Variable | Delete Variable | Optimize | Cancel

Example 3: Garden City Beach – How Many Lifeguards?

- Each summer, the city hires lifeguards to assign five consecutive days each week followed by two days off. The city's insurance company requires the minimum number of lifeguards each day:

Day	Sun Day 0	Mon Day 1	Tue Day 2	Wed Day 3	Thu Day 4	Fri Day 5	Sat Day 6
Lifeguards Required	18	17	16	16	16	14	19

- Let X_i = Number of workers who start on the following Day: i.e. Day 7 | i + 1

MIN $X_0 + X_1 + X_2 + X_3 + X_4 + X_5 + X_6$

ST
$$X_1 + X_2 + X_3 + X_4 + X_5 \geq 18$$
$$X_2 + X_3 + X_4 + X_5 + X_6 \geq 17$$
$$X_0 + X_3 + X_4 + X_5 + X_6 \geq 16$$
$$X_0 + X_1 + X_4 + X_5 + X_6 \geq 16$$
$$X_0 + X_1 + X_2 + X_5 + X_6 \geq 16$$
$$X_0 + X_1 + X_2 + X_3 + X_6 \geq 14$$
$$X_0 + X_1 + X_2 + X_3 + X_4 \geq 19$$
$$X_i \geq 0$$

```
      EX3←⎕NS ''
      EX3.A←(-⍳7)⌽⍤0 1⊢1 5 1/0 1 0
      EX3.b←18 17 16 16 16 14 19
      EX3.c←7/1
      EX3.optimum←⌊
      EX3.rel←≥,≥,≥,≥,≥,≥,≥
      EX3←LP EX3
      EX3.Decision
4.6 1.6 5.6 1.6 5.6 3.6 0.6
      EX3←IP EX3 ⍝ Must be integer
      EX3.Decision
3 3 5 0 8 2 3
      EX3.Objective
24
```

Conclusion

- Optimization techniques can extended to the following
  - LP – Linear Programming
  - IP - Integer Programming
  - TP – Transportation Problem
  - NLP – Non-linear Programming
- We can use HTML Renderer via Abacus to generate the user interface.
  - Insert and Delete Rows and Columns
  - Use Checkboxes and Drop-Downs where useful.
- Web site and Documentation
  - www.tamstat.com
  - Paper: Optimizing with Defined Operators