BAMS 508 – Discrete Optimization

Assignment 3

Gurpal Bisra, Deanna Garson

Date: November 16, 2016

Question 1

1. Solving a Cutting Stock instance. Consider the instance of the Cutting Stock problem, to satisfy the following demands for small rolls using the smallest possible number of big rolls, where each big roll has width $W = 240$ cm.

Small roll type:	1	2	3	4	5	6
Width (cm)	29	43	47	50	26	23
Number required (rolls)	81	56	96	94	118	121

(a) Define the initial restriction for the Gilmore & Gomory approach, using only patterns each consisting of as many small rolls as possible of a single type. Solve its LP relaxation. Does the optimum objective value of this LP relaxation give, in general, a lower bound or an upper bound on the optimum value of the integer program? Explain.

(b) Solve using the Restriction & Pricing (aka, Column Generation) method, starting from this initial restriction. Detail and explain your calculations.

(a) <u>Information Provided:</u>

- Big roll width W = 240 cm
- 4 small roll types (“standards”)
  - each with width $w_i > 0$ and demand $d_i > 0$ (i = 1, … m)
- Small Roll Types 1 can fit = $\lfloor \frac{240cm}{29cm} \rfloor$ = 8 type-1 rolls
- Small Roll Types 2 can fit = $\lfloor \frac{240cm}{43cm} \rfloor$ = 5 type-2 rolls
- Small Roll Types 3 can fit = $\lfloor \frac{240cm}{47cm} \rfloor$ = 5 type-3 rolls
- Small Roll Types 4 can fit = $\lfloor \frac{240cm}{50cm} \rfloor$ = 4 type-4 rolls
- Small Roll Types 5 can fit = $\lfloor \frac{240cm}{26cm} \rfloor$ = 9 type-5 rolls
- Small Roll Types 6 can fit = $\lfloor \frac{240cm}{23cm} \rfloor$ = 10 type-6 rolls

We are asked to define a cutting pattern y ∈ Z^m where y_i = is the number of small rolls i in the pattern

- pattern y is feasible if its total width c(T) = $\sum_{i \in M} w_i \, y_i \leq W$
- let c_y denote the cost of using pattern y
  - c_y = 1 to minimize the number of big rolls cut
  - c_y = W = $\sum_{i \in M} w_i \, y_i$ to minimize cutting waste

Integer Programming Formulation:

Decision Variables:

- let $x_y =$ denote the number of big rolls cut according to pattern y where y = 1, 2, … 6

Constraints:

1. **Type-1 Rolls:** The number of required type-1 rolls to be cut is 81.
$$8 * x_1 \geq 81$$
2. **Type-2 Rolls:** The number of required type-2 rolls to be cut is 56.
$$5 * x_2 \geq 56$$
3. **Type-3 Rolls:** The number of required type-3 rolls to be cut is 96.
$$5 * x_3 \geq 96$$
4. **Type-4 Rolls:** The number of required type-4 rolls to be cut is 94.
$$4 * x_4 \geq 94$$
5. **Type-5 Rolls:** The number of required type-5 rolls to be cut is 118.
$$9 * x_5 \geq 118$$
6. **Type-6 Rolls:** The number of required type-6 rolls to be cut is 121.
$$10 * x_6 \geq 121$$

Objective:

- To minimize smallest number of big rolls cut.
$$\min Z = \{ c_y = W = \textstyle\sum_{i \,\epsilon\, 6} x_i \}$$

An annotated version of the Excel Solver spreadsheet, for the LP relaxation, with its sensitivity report is found below:

Big roll width	W =	240	cm			
Small roll type:	**1**	**2**	**3**	**4**	**5**	**6**
Width (cm)	29	43	47	50	26	23
Number required (rolls)	81	56	96	94	118	121
# Small Rolls fit on 1 Big Roll	8	5	5	4	9	10
Decision Variables	10.125	11.2	19.2	23.5	13.11111	12.1
# small rolls of type	**x1**	**x2**	**x3**	**x4**	**x5**	**x5**
Objective Function	min (Z = ∑ (i є 6) x i)					
		89.23611	**total rolls**			

Constraints	**Value**		**Minimum**	**Slack**
Type-1 Roll	81	≥	81	0
Type-2 Roll	56	≥	56	0
Type-3 Roll	96	≥	96	0
Type-4 Roll	94	≥	94	0
Type-5 Roll	118	≥	118	0
Type-6 Roll	121	≥	121	0
all xi are integer				

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
B12	Decision Variables W =	10.125	0	1	1E+30	1
C12	Decision Variables	11.2	0	1	1E+30	1
D12	Decision Variables cm	19.2	0	1	1E+30	1
E12	Decision Variables	23.5	0	1	1E+30	1
F12	Decision Variables	13.11111111	0	1	1E+30	1
G12	Decision Variables	12.1	0	1	1E+30	1

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
J7	Type-1 Roll Value	81	0.125	81	1E+30	81
J8	Type-2 Roll Value	56	0.2	56	1E+30	56
J9	Type-3 Roll Value	96	0.2	96	1E+30	96
J10	Type-4 Roll Value	94	0.25	94	1E+30	94
J11	Type-5 Roll Value	118	0.111111111	118	1E+30	118
J12	Type-6 Roll Value	121	0.1	121	1E+30	121

Hence, according to the solution for the LP relaxation, 89.23611 rolls need to be cut.

When I add a 7th constraint, as seen below, I obtain the annotated integer solution.

7. x_y **integer:** The number of each type of roll cut must be integer.

An annotated version of the Excel Solver spreadsheet, for the integer programming problem is found below:

Big roll width	W =	240	cm			
Small roll type:	**1**	**2**	**3**	**4**	**5**	**6**
Width (cm)	29	43	47	50	26	23
Number required (rolls)	81	56	96	94	118	121
# Small Rolls fit on 1 Big Roll	8	5	5	4	9	10
Decision Variables	11	12	20	24	14	13
# small rolls of type	**x1**	**x2**	**x3**	**x4**	**x5**	**x5**
Objective Function	min (Z = ∑ (i є 6) x i)					
		94	**total rolls**			

Constraints				
	Value		**Minimum**	**Slack**
Type-1 Roll	88	≥	81	7
Type-2 Roll	60	≥	56	4
Type-3 Roll	100	≥	96	4
Type-4 Roll	96	≥	94	2
Type-5 Roll	126	≥	118	8
Type-6 Roll	130	≥	121	9
all xi are integer				

The **LP relaxation provides a lower bound on the optimum value of the integer program** since fractions of rolls can be used as decision variable values to fulfill the constraints. Because relaxing the integrality constraints yields a linear program (the LP relaxation), to which we can find provably optimum solutions, the slack values of each constraint are 0.

(b) Next, we solve the Restriction & Pricing (aka Column Generation) method, starting from the initial restriction.

Recall from lecture: Restriction & Pricing approach, aka (Delayed) Column Generation

The Pricing Problem for Cutting Stock:

given dual prices $v \in \mathbf{R}^m$

find a cutting pattern $y \in \mathbf{Z}^m_+$ such that $\sum_{i \in M} w_i y_i \leq W$ and

with negative reduced cost $c'_y = c_y - \sum_{i \in M} v_i y_i = 1 - \sum_{i \in M} v_i y_i$

Finding a feasible cutting pattern $y \in \mathbf{Z}^m_+$ with most negative reduced cost is equivalent to solving the integer knapsack problem:

$\max_y \{\sum_{i \in M} v_i y_i : \sum_{i \in M} w_i y_i \leq W;\ y \geq 0, \text{integer}\}$

- so the subproblem can itself be formulated as an IP problem!

- identify a (small) subset of (potentially interesting) variables
  - e.g., all singleton subsets for set covering, packing or partitioning
- ***repeat***
  - define the current restriction, aka the (current) Master Problem, by including known variables
  - solve the current restriction, obtaining the current LP solution x' and dual prices v
    - so x' and v satisfy Complementary Slackness
  - Pricing Problem (aka the Sub-problem): identify one (or several)
  - variable(s) with negative reduced cost (for a minimization problem)
    - these new variables are added to the current restriction
- ***until no variable has a negative reduced cost***
- The final solution x'
  - with all variables xj not in the current restriction set to the value x'j = 0
  - is an optimal solution to the full LP (x' satisfies the LP optimality conditions)

Knapsack sub-problem 1:

- **Decision Variables:** y1, … y6 ≥ 0 as integer
- **Objective Function:** max $\{\frac{1}{8}$ y1 + $\frac{1}{5}$ y2 + $\frac{1}{5}$ y3 + $\frac{1}{4}$ y4 + $\frac{1}{9}$ y5 + $\frac{1}{10}$ y6 $\}$
- **Subject To:**
  - 29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 ≤ 240 cm

KNAPSACK PROBLEM 1												
Decision Variables		0	1	1	3	0	0	**Constraints**				
# small rolls of type		**y1**	**y2**	**y3**	**y4**	**y5**	**y6**		**Value**		**Minimum**	**Slack**
								Iteration 1	240	≤	240	0
Objective Function	max $\{\frac{1}{8}$y1 + $\frac{1}{5}$y2 + $\frac{1}{5}$y3 + $\frac{1}{4}$y4 + $\frac{1}{9}$y5 + $\frac{1}{10}$y6$\}$							29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 ≤ 240 cm				
			1.15	total rolls				**All yi are integer**				

Now, a determined a new cutting pattern which I will denote as:

x_7 = big roll fits 1x type-2 rolls, 1x type-3 rolls, and 3x type-4 rolls

Then, I used this new cutting pattern as a new variable for the linear programming problem. An annotated version of the Excel Solver spreadsheet, for the LP relaxation, with its sensitivity report is found below:

Big roll width	W =	240	cm				
Small roll type:	**1**	**2**	**3**	**4**	**5**	**6**	
Width (cm)	29	43	47	50	26	23	
Number required (rolls)	81	56	96	94	118	121	
# Small Rolls fit on 1 Big Roll	8	5	5	4	9	10	
Decision Variables	10.125	4.933333	12.93333	0	13.11111	12.1	31.3333333
# small rolls of type	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**
Objective Function	min $(Z = \sum_{(i \in 6)} x_i)$						
		84.53611	**total rolls**				

Constraints	**Value**		**Minimum**	**Slack**	**Dual Values**
Type-1 Roll	81	≥	81	0	0.125000001
Type-2 Roll	56	≥	56	0	0.2
Type-3 Roll	96	≥	96	0	0.200000004
Type-4 Roll	94	≥	94	0	0.249999999
Type-5 Roll	118	≥	118	0	0.111111109
Type-6 Roll	121	≥	121	0	0.099999999
all xi are integer					

KNAPSACK PROBLEM 1

Decision Variables	0	1	1	3	0	0
# small rolls of type	**y1**	**y2**	**y3**	**y4**	**y5**	**y6**
Objective Function	$\max\{\frac{1}{8}y1 + \frac{1}{5}y2 + \frac{1}{5}y3 + \frac{1}{4}y4 + \frac{1}{9}y5 + \frac{1}{10}y6\}$					
		1.15	**total rolls**			

Constraints	**Value**		**Minimum**	**Slack**
Iteration 1	240	≤	240	0

29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 ≤ 240 cm

All yi are integer

Initial Pattern	**Pattern 1**	**Pattern 2**	**Pattern 3**	**Pattern 4**	**Pattern 5**	**Pattern 6**	**Pattern 7**
1	8	0	0	0	0	0	0
2	0	5	0	0	0	0	1
3	0	0	5	0	0	0	1
4	0	0	0	4	0	0	3
5	0	0	0	0	9	0	0
6	0	0	0	0	0	10	0

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
\$B\$12	Decision Variables W =	10.125	0	1	1E+30	1
\$C\$12	Decision Variables	4.933333333	0	1	3	0.75
\$D\$12	Decision Variables cm	12.93333333	0	1	3	0.75
\$E\$12	Decision Variables	0	0.2	1	1E+30	0.2
\$F\$12	Decision Variables	13.11111111	0	1	1E+30	1
\$G\$12	Decision Variables	12.1	0	1	1E+30	1
\$H\$12	Decision Variables	31.33333333	0	1	0.15	0.6

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
\$J\$7	Type-1 Roll Value	81	0.125	81	1E+30	81
\$J\$8	Type-2 Roll Value	56	0.2	56	1E+30	24.66666667
\$J\$9	Type-3 Roll Value	96	0.2	96	1E+30	64.66666667
\$J\$10	Type-4 Roll Value	94	0.2	94	74	94
\$J\$11	Type-5 Roll Value	118	0.111111111	118	1E+30	118
\$J\$12	Type-6 Roll Value	121	0.1	121	1E+30	121

The new shadow prices can be used to setup the next knapsack problem for the following iteration.

Knapsack sub-problem 2:

- **Decision Variables:** y1, … y6 ≥ 0 as integer
- **Objective Function:** max { $\frac{1}{8}$ y1 + $\frac{1}{5}$ y2 + $\frac{1}{5}$ y3 + $\frac{1}{5}$ y4 + $\frac{1}{9}$ y5 + $\frac{1}{10}$ y6 }
- **Subject To:**
  - 29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 ≤ 240 cm

Big roll width	W =	240	cm						
Small roll type:	**1**	**2**	**3**	**4**	**5**	**6**			
Width (cm)	29	43	47	50	26	23			
Number required (rolls)	81	56	96	94	118	121			
# Small Rolls fit on 1 Big Roll	8	5	5	4	9	10			
Decision Variables	10.125	0	12.93333	0	13.11111	11.60667	31.3333333	4.93333333	
# small rolls of type	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	
Objective Function	min (Z = ∑ (i ϵ 6) x_i)								
		84.04278	**total rolls**						

Constraints					
	Value		**Minimum**	**Slack**	**Dual Values**
Type-1 Roll	81	≥	81	0	0.125000001
Type-2 Roll	56	≥	56	0	0.2
Type-3 Roll	96	≥	96	0	0.200000004
Type-4 Roll	94	≥	94	0	0.249999999
Type-5 Roll	118	≥	118	0	0.111111109
Type-6 Roll	121	≥	121	0	0.099999999
all xi are integer					

KNAPSACK PROBLEM 2

Decision Variables	0	5	0	0	0	1
# small rolls of type	**y1**	**y2**	**y3**	**y4**	**y5**	**y6**
Objective Function	max { $\frac{1}{8}$ y1 + $\frac{1}{5}$ y2 + $\frac{1}{5}$ y3 + $\frac{1}{5}$ y4 + $\frac{1}{9}$ y5 + $\frac{1}{10}$ y6 }					
	1.1	**total rolls**				

Constraints				
	Value		**Minimum**	**Slack**
Iteration 1	238	≤	240	-2
29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 ≤ 240 cm				
All yi are integer				

Initial Pattern		**Pattern 1**	**Pattern 2**	**Pattern 3**	**Pattern 4**	**Pattern 5**	**Pattern 6**	**Pattern 7**	**Pattern 8**
	1	8	0	0	0	0	0	0	0
	2	0	5	0	0	0	0	1	5
	3	0	0	5	0	0	0	1	0
	4	0	0	0	4	0	0	3	0
	5	0	0	0	0	9	0	0	0
	6	0	0	0	0	0	10	0	1

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
B12	Decision Variables W =	10.125	0	1	1E+30	1
C12	Decision Variables	0	0.1	1	1E+30	0.1
D12	Decision Variables cm	12.93333333	0	1	3.1	0.65
E12	Decision Variables	0	0.173333333	1	1E+30	0.173333333
F12	Decision Variables	13.11111111	0	1	1E+30	1
G12	Decision Variables	11.60666667	0	1	6.5	1
H12	Decision Variables	31.33333333	0	1	0.13	0.62
I12	Decision Variables	4.933333333	0	1	0.1	0.65

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
K7	Type-1 Roll Value	81	0.125	81	1E+30	81
K8	Type-2 Roll Value	56	0.18	56	580.3333333	24.66666667
K9	Type-3 Roll Value	96	0.2	96	1E+30	64.66666667
K10	Type-4 Roll Value	94	0.206666667	94	74	94
K11	Type-5 Roll Value	118	0.111111111	118	1E+30	118
K12	Type-6 Roll Value	121	0.1	121	1E+30	116.0666667

From solving the knapsack problem first, I determined a new cutting pattern which I will denote as:

x_8 = big roll fits 5x type-2 rolls, 1x type-6 rolls

The new shadow prices can be used to setup the next knapsack problem for the following iteration.

Knapsack sub-problem 3:

- **Decision Variables:** y1, … y6 ≥ 0 as integer
- **Objective Function:** max { $\frac{1}{8}$ y1 + $\frac{9}{50}$ y2 + $\frac{1}{5}$ y3 + $\frac{31}{150}$ y4 + $\frac{1}{9}$ y5 + $\frac{1}{10}$ y6 }
- **Subject To:**
  - 29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 ≤ 240 cm

Big roll width	W =	240	cm						
Small roll type:	**1**	**2**	**3**	**4**	**5**	**6**			
Width (cm)	29	43	47	50	26	23			
Number required (rolls)	81	56	96	94	118	121			
# Small Rolls fit on 1 Big Roll	8	5	5	4	9	10			
Decision Variables	8.071934	0	14.0283	0	13.11111	0	25.8584906	6.02830189	16.4245283
# small rolls of type	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**
Objective Function	min (Z = ∑_(i ϵ 6) x_i)								
		83.52267	**total rolls**						

Constraints	**Value**		**Minimum**	**Slack**	**Shadow Price**
Type-1 Roll	81	≥	81	0	0.125
Type-2 Roll	56	≥	56	0	0.18
Type-3 Roll	96	≥	96	0	0.2
Type-4 Roll	94	≥	94	0	0.206666667
Type-5 Roll	118	≥	118	0	0.111111111
Type-6 Roll	121	≥	121	0	0.1
all xi are integer					

KNAPSACK PROBLEM 3

Decision Variables	1	0	0	1	0	7
# small rolls of type	**y1**	**y2**	**y3**	**y4**	**y5**	**y6**
Objective Function	max { $\frac{1}{8}$y1 + $\frac{9}{50}$y2 + $\frac{1}{5}$y3 + $\frac{31}{150}$y4 + $\frac{1}{9}$y5 + $\frac{1}{10}$y6 }					
		1.031667	**total rolls**			

Constraints	**Value**		**Minimum**	**Slack**
Iteration 1	240	≤	240	0

29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 ≤ 240 cm

All yi are integer

Initial Pattern		**Pattern 1**	**Pattern 2**	**Pattern 3**	**Pattern 4**	**Pattern 5**	**Pattern 6**	**Pattern 7**	**Pattern 8**	**Pattern 9**
	1	8	0	0	0	0	0	0	0	1
	2	0	5	0	0	0	0	1	5	0
	3	0	0	5	0	0	0	1	0	0
	4	0	0	0	4	0	0	3	0	1
	5	0	0	0	0	9	0	0	0	0
	6	0	0	0	0	0	10	0	1	7

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
B12	Decision Variables W =	8.071933962	0	1	5.4	0.253333333
C12	Decision Variables	0	0.095518868	1	1E+30	0.095518868
D12	Decision Variables cm	14.02830189	0	1	0.475	0.660714286
E12	Decision Variables	0	0.174528302	1	1E+30	0.174528302
F12	Decision Variables	13.11111111	0	1	1E+30	1
G12	Decision Variables	0	0.044811321	1	1E+30	0.044811321
H12	Decision Variables	25.85849057	0	1	0.132142857	0.095
I12	Decision Variables	6.028301887	0	1	0.096428571	0.660714286
J12	Decision Variables	16.4245283	0	1	0.031666667	0.675

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
L7	Type-1 Roll Value	81	0.125	81	1E+30	64.5754717
L8	Type-2 Roll Value	56	0.180896226	56	580.3333333	30.42857143
L9	Type-3 Roll Value	96	0.2	96	1E+30	70.14150943
L10	Type-4 Roll Value	94	0.206367925	94	91.28571429	78.31428571
L11	Type-5 Roll Value	118	0.111111111	118	1E+30	118
L12	Type-6 Roll Value	121	0.095518868	121	456.3333333	116.0666667

From solving the knapsack problem first, I determined a new cutting pattern which I will denote as:

x_9 = big roll fits 1x type-1 rolls, 1x type-4 rolls, and 7x type-6 rolls.

The new shadow prices can be used to setup the next knapsack problem for the following iteration.

Knapsack sub-problem 4:

- **Decision Variables:** y1, … y6 ≥ 0 as integer
- **Objective Function:** max { $\frac{1}{8}$ y1 + (0.180896226*y2) + $\frac{1}{5}$ y3 + (0.206367925*y4) + $\frac{1}{9}$ y5 + (0.095518868*y6) }
- **Subject To:**
  - 29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 ≤ 240 cm

Big roll width	W =	240	cm							
Small roll type:	**1**	**2**	**3**	**4**	**5**	**6**				
Width (cm)	29	43	47	50	26	23				
Number required (rolls)	81	56	96	94	118	121				
# Small Rolls fit on 1 Big Roll	8	5	5	4	9	10				
Decision Variables	0	0	14.05253	0	10.25724	0	25.7373358	3.48405253	16.7879925	12.8424015
# small rolls of type	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**x10**
Objective Function	min (Z = ∑_(i ϵ 6) x_i)									
		83.16156	**total rolls**							
KNAPSACK PROBLEM 4										
Decision Variables	5	1	0	0	2	0				
# small rolls of type	**y1**	**y2**	**y3**	**y4**	**y5**	**y6**				
Objective Function	max { $\frac{1}{8}$ y1 + (0.180896226*y2) + $\frac{1}{5}$ y3 + (0.206367925*y4) + $\frac{1}{9}$ y5 + (0.095518868*y6) }									
		1.028118	**total rolls**							

Initial Pattern		**Pattern 1**	**Pattern 2**	**Pattern 3**	**Pattern 4**	**Pattern 5**	**Pattern 6**	**Pattern 7**	**Pattern 8**	**Pattern 9**	**Pattern 10**
	1	8	0	0	0	0	0	0	0	1	5
	2	0	5	0	0	0	0	1	5	0	1
	3	0	0	5	0	0	0	1	0	0	0
	4	0	0	0	4	0	0	3	0	1	0
	5	0	0	0	0	9	0	0	0	0	2
	6	0	0	0	0	0	10	0	1	7	0

Constraints					**Shadow**
	Value		**Minimum**	**Slack**	**Price**
Type-1 Roll	81	≥	81	0	0.125
Type-2 Roll	56	≥	56	0	0.180896226
Type-3 Roll	96	≥	96	0	0.2
Type-4 Roll	94	≥	94	0	0.206367925
Type-5 Roll	118	≥	118	0	0.111111111
Type-6 Roll	121	≥	121	0	0.095518868
all xi are integer					

Constraints				
	Value		**Minimum**	**Slack**
Iteration 1	240	≤	240	0
29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 ≤ 240 cm				
All yi are integer				

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
B12	Decision Variables W =	0	0.044736294	1	1E+30	0.044736294
C12	Decision Variables	0	0.096310194	1	1E+30	0.096310194
D12	Decision Variables cm	14.05253283	0	1	0.393333333	0.659880051
E12	Decision Variables	0	0.174317282	1	1E+30	0.174317282
F12	Decision Variables	10.25724411	0	1	0.59	0.126533019
G12	Decision Variables	0	0.036898061	1	1E+30	0.036898061
H12	Decision Variables	25.73733583	0	1	0.13197601	0.078666667
I12	Decision Variables	3.484052533	0	1	0.097777778	0.141931217
J12	Decision Variables	16.7879925	0	1	0.026222222	0.684444444
K12	Decision Variables	12.8424015	0	1	0.028118449	0.131111111

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
M7	Type-1 Roll Value	81	0.119407963	81	88.42857143	64.5754717
M8	Type-2 Roll Value	56	0.180737961	56	596.5333333	17.68571429
M9	Type-3 Roll Value	96	0.2	96	1E+30	70.26266417
M10	Type-4 Roll Value	94	0.20642068	94	53.05714286	77.94318182
M11	Type-5 Roll Value	118	0.111111111	118	1E+30	92.315197
M12	Type-6 Roll Value	121	0.096310194	121	456.3333333	119.3066667

From solving the knapsack problem first, I determined a new cutting pattern which I will denote as:

x_{10} = big roll fits 5x type-1 rolls, 1x type-2 rolls, and 2x type-5 rolls.

The new shadow prices can be used to setup the next knapsack problem for the following iteration.

Knapsack sub-problem 5:

- **Decision Variables:** y1, … y6 ≥ 0 as integer
- **Objective Function:**

$\max \{(0.119407963*y1) + (0.180737961*y2) + \frac{1}{5}y3 + (0.20642068*y4) + \frac{1}{9}y5 + (0.096310194*y6)\}$

- **Subject To:**
  - $29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 \leq 240$ cm

Small roll type:	1	2	3	4	5	6					
Width (cm)	29	43	47	50	26	23					
Number required (rolls)	81	56	96	94	118	121					
# Small Rolls fit on 1 Big Roll	8	5	5	4	9	10					
Decision Variables	0	0	0	0	6.353763	0	25.7373358	3.48405253	16.7879925	12.8424015	17.565666
# small rolls of type	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11
Objective Function	min (Z = ∑_(i ε 6) x_i)										
		82.77121	total rolls								
KNAPSACK PROBLEM 5											
Decision Variables	0	0	4	0	2	0					
# small rolls of type	y1	y2	y3	y4	y5	y6					
Objective Function	max {(0.119407963*y1) + (0.180737961*y2) + 1/5 y3 + (0.20642068*y4) + 1/9 y5 + (0.096310194*y6)}										
		1.022222	total rolls								

Initial Pattern		Pattern 1	Pattern 2	Pattern 3	Pattern 4	Pattern 5	Pattern 6	Pattern 7	Pattern 8	Pattern 9	Pattern 10	Pattern 11
	1	8	0	0	0	0	0	0	0	1	5	0
	2	0	5	0	0	0	0	1	5	0	1	0
	3	0	0	5	0	0	0	1	0	0	0	4
	4	0	0	0	4	0	0	3	0	1	0	0
	5	0	0	0	0	9	0	0	0	0	2	2
	6	0	0	0	0	0	10	0	1	7	0	0

Constraints					Shadow Price
	Value		Minimum	Slack	
Type-1 Roll	81	≥	81	0	0.119407963
Type-2 Roll	56	≥	56	0	0.180737961
Type-3 Roll	96	≥	96	0	0.2
Type-4 Roll	94	≥	94	0	0.20642068
Type-5 Roll	118	≥	118	0	0.111111111
Type-6 Roll	121	≥	121	0	0.096310194
all xi are integer					

Constraints				
	Value		Minimum	Slack
Iteration 1	240	≤	240	0
29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 ≤ 240 cm				
All yi are integer				

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
B12	Decision Variables W =	0	0.044819679	1	1E+30	0.044819679
C12	Decision Variables	0	0.096049614	1	1E+30	0.096049614
D12	Decision Variables cm	0	0.027777778	1	1E+30	0.027777778
E12	Decision Variables	0	0.166979362	1	1E+30	0.166979362
F12	Decision Variables	6.353762768	0	1	1.082857143	0.1
G12	Decision Variables	0	0.039503857	1	1E+30	0.039503857
H12	Decision Variables	25.73733583	0	1	0.126420455	0.084222222
I12	Decision Variables	3.484052533	0	1	0.097513228	0.142195767
J12	Decision Variables	16.7879925	0	1	0.028074074	0.682592593
K12	Decision Variables	12.8424015	0	1	0.02817086	0.14037037
L12	Decision Variables	17.56566604	0	1	0.022222222	0.505681818

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
N7	Type-1 Roll Value	81	0.11939754	81	88.42857143	64.5754717
N8	Type-2 Roll Value	56	0.180790077	56	596.5333333	17.68571429
N9	Type-3 Roll Value	96	0.194444444	96	114.3677298	70.26266417
N10	Type-4 Roll Value	94	0.208255159	94	53.05714286	77.94318182
N11	Type-5 Roll Value	118	0.111111111	118	1E+30	57.18386492
N12	Type-6 Roll Value	121	0.096049614	121	456.3333333	119.3066667

From solving the knapsack problem first, I determined a new cutting pattern which I will denote as:

x_{11} = big roll fits 4x type-3 rolls, 2x type-5 rolls

The new shadow prices can be used to setup the next knapsack problem for the following iteration.

Knapsack sub-problem 6:

- **Decision Variables:** y1, … y6 ≥ 0 as integer
- **Objective Function:**

max {(0.119407963*y1) + (0.180737961*y2) + $\frac{1}{5}$ y3 + (0.20642068*y4) + $\frac{1}{9}$ y5 + (0.096310194*y6)}

- **Subject To:**
  - 29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 ≤ 240 cm

Big roll width	W =	240	cm									
Small roll type:	**1**	**2**	**3**	**4**	**5**	**6**						
Width (cm)	29	43	47	50	26	23						
Number required (rolls)	81	56	96	94	118	121						
# Small Rolls fit on 1 Big Roll	8	5	5	4	9	10						
Decision Variables	0	0	0	0	0	0	25.7719298	4.21052632	16.6842105	9.1754386	17.5570175	9.21929825
# small rolls of type	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**x10**	**x11**	**x12**
Objective Function	min (Z = ∑_(i ϵ 6) x_i)											
		82.61842	**total rolls**									

KNAPSACK PROBLEM 6						
Decision Variables	2	0	0	0	7	0
# small rolls of type	**y1**	**y2**	**y3**	**y4**	**y5**	**y6**
Objective Function	max {(0.119407963*y1) + (0.180737961*y2) + $\frac{1}{5}$ y3 + (0.20642068*y4) + $\frac{1}{9}$ y5 + (0.096310194*y6)}					
		1.016573	**total rolls**			

Initial Pattern		**Pattern 1**	**Pattern 2**	**Pattern 3**	**Pattern 4**	**Pattern 5**	**Pattern 6**	**Pattern 7**	**Pattern 8**	**Pattern 9**	**Pattern 10**	**Pattern 11**	**Pattern 12**
	1	8	0	0	0	0	0	0	0	1	5	0	2
	2	0	5	0	0	0	0	1	5	0	1	0	0
	3	0	0	5	0	0	0	1	0	0	0	4	0
	4	0	0	0	4	0	0	3	0	1	0	0	0
	5	0	0	0	0	9	0	0	0	0	2	2	7
	6	0	0	0	0	0	10	0	1	7	0	0	0

Constraints					**Shadow Price**
	Value		**Minimum**	**Slack**	
Type-1 Roll	81	≥	81	0	0.11939754
Type-2 Roll	56	≥	56	0	0.180790077
Type-3 Roll	96	≥	96	0	0.194444444
Type-4 Roll	94	≥	94	0	0.208255159
Type-5 Roll	118	≥	118	0	0.111111111
Type-6 Roll	121	≥	121	0	0.096049614
all xi are integer					

Constraints				
	Value		**Minimum**	**Slack**
Iteration 1	240	≤	240	0
29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 ≤ 240 cm				
All yi are integer				

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
B12	Decision Variables W =	0	0.036297641	1	1E+30	0.036297641
C12	Decision Variables	0	0.095961887	1	1E+30	0.095961887
D12	Decision Variables cm	0	0.021098004	1	1E+30	0.021098004
E12	Decision Variables	0	0.168784029	1	1E+30	0.168784029
F12	Decision Variables	0	0.024047187	1	1E+30	0.024047187
G12	Decision Variables	0	0.040381125	1	1E+30	0.040381125
H12	Decision Variables	25.77192982	0	1	0.127747253	0.086129032
I12	Decision Variables	4.210526316	0	1	0.097465438	0.102040816
J12	Decision Variables	16.68421053	0	1	0.028709677	0.682258065
K12	Decision Variables	9.175438596	0	1	0.020215633	0.041666667
L12	Decision Variables	17.55701754	0	1	0.016873299	0.510989011
M12	Decision Variables	9.219298246	0	1	0.016572858	0.052345216

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
O7	Type-1 Roll Value	81	0.120462795	81	94.69387755	40.88140162
O8	Type-2 Roll Value	56	0.180807623	56	593.0967742	21.38248848
O9	Type-3 Roll Value	96	0.195780399	96	114.3677298	70.20683399
O10	Type-4 Roll Value	94	0.207803993	94	66.28571429	78.02380952
O11	Type-5 Roll Value	118	0.108439201	118	143.0849057	57.18386492
O12	Type-6 Roll Value	121	0.095961887	121	303.34	118.6193548

From solving the knapsack problem first, I determined a new cutting pattern which I will denote as:

x_{12} = big roll fits 2x type-1 rolls, 5x type-5 rolls

The new shadow prices can be used to setup the next knapsack problem for the following iteration.

Knapsack sub-problem 7:

- **Decision Variables:** y1, … y6 ≥ 0 as integer
- **Objective Function:**

max {(0.120462795*y1) + (0.180807623*y2) + (0.195780399*y3) + (0.207803993*y4) + (0.108439201*y5) + (0.095961887 *y6)}

- **Subject To:**
  - 29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 ≤ 240 cm

Big roll width	W =	240	cm									
Small roll type:	**1**	**2**	**3**	**4**	**5**	**6**						
Width (cm)	29	43	47	50	26	23						
Number required (rolls)	81	56	96	94	118	121						
# Small Rolls fit on 1 Big Roll	8	5	5	4	9	10						
Decision Variables	0	0	0	0	0	0	25.7719298	4.21052632	16.6842105	9.1754386	17.5570175	9.21929825
# small rolls of type	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**x10**	**x11**	**x12**
Objective Function	min (Z = ∑_(i ϵ 6) x_i)											
		82.61842	**total rolls**									

KNAPSACK PROBLEM 7						
Decision Variables	0	5	0	0	0	1
# small rolls of type	**y1**	**y2**	**y3**	**y4**	**y5**	**y6**
Objective Function	max {(0.119407963*y1) + (0.180737961*y2) + $\frac{1}{5}$ y3 + (0.20642068*y4) + $\frac{1}{9}$ y5 + (0.096310194*y6)}					
		1	**total rolls**			

Initial Pattern		**Pattern 1**	**Pattern 2**	**Pattern 3**	**Pattern 4**	**Pattern 5**	**Pattern 6**	**Pattern 7**	**Pattern 8**	**Pattern 9**	**Pattern 10**	**Pattern 11**	**Pattern 12**
	1	8	0	0	0	0	0	0	0	1	5	0	2
	2	0	5	0	0	0	0	1	5	0	1	0	0
	3	0	0	5	0	0	0	1	0	0	0	4	0
	4	0	0	0	4	0	0	3	0	1	0	0	0
	5	0	0	0	0	9	0	0	0	0	2	2	7
	6	0	0	0	0	0	10	0	1	7	0	0	0

Constraints					**Shadow Price**
	Value		**Minimum**	**Slack**	
Type-1 Roll	81	≥	81	0	0.120462795
Type-2 Roll	56	≥	56	0	0.180807623
Type-3 Roll	96	≥	96	0	0.195780399
Type-4 Roll	94	≥	94	0	0.207803993
Type-5 Roll	118	≥	118	0	0.108439201
Type-6 Roll	121	≥	121	0	0.095961887
all xi are integer					

Constraints				
	Value		**Minimum**	**Slack**
Iteration 1	238	≤	240	-2
29*y1 + 43*y2 + 47*y3 + 50*y4 + 26*y5 + 23*y6 ≤ 240 cm				
All yi are integer				

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
H12	Decision Variables	25.77192982	0	1	2.046774194	0.629120879
I12	Decision Variables	4.210526316	0	1	2.709183673	0.918202765
J12	Decision Variables	16.68421053	0	1	6.427419355	0.682258065
K12	Decision Variables	9.175438596	0	1	1.691037736	0.536725067
L12	Decision Variables	17.55701754	0	1	2.516483516	0.78288479
M12	Decision Variables	9.219298246	0	1	1.874117647	0.67260788

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
O7	Type-1 Roll Value	81	0.120462795	81	94.69387755	40.88140162
O8	Type-2 Roll Value	56	0.180807623	56	593.0967742	21.38248848
O9	Type-3 Roll Value	96	0.195780399	96	114.3677298	70.20683399
O10	Type-4 Roll Value	94	0.207803993	94	66.28571429	78.02380952
O11	Type-5 Roll Value	118	0.108439201	118	143.0849057	57.18386492
O12	Type-6 Roll Value	121	0.095961887	121	303.34	118.6193548

From solving the knapsack problem first, I determined a no new cutting pattern. Hence, I believe I determined all six optimum cutting patterns.

Finally, using these new values, we solved the integer programming model by adding the following constraint:

7. x_y **integer:** The number of each type of roll cut must be integer.

Big roll width	W =	240	cm									
Small roll type:	**1**	**2**	**3**	**4**	**5**	**6**						
Width (cm)	29	43	47	50	26	23						
Number required (rolls)	81	56	96	94	118	121						
# Small Rolls fit on 1 Big Roll	8	5	5	4	9	10						
Decision Variables	0	0	0	0	0	0	26	4	17	10	18	9
# small rolls of type	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**x10**	**x11**	**x12**
Objective Function	min (Z = ∑_(i ϵ 6) x_i)											
		84	**total rolls**									

Constraints	Value		Minimum	Slack	Shadow Price
Type-1 Roll	85	≥	81	4	0.120462795
Type-2 Roll	56	≥	56	0	0.180807623
Type-3 Roll	98	≥	96	2	0.195780399
Type-4 Roll	95	≥	94	1	0.207803993
Type-5 Roll	119	≥	118	1	0.108439201
Type-6 Roll	123	≥	121	2	0.095961887
all xi are integer					

Hence, according to the solution for the LP relaxation, <u>84 rolls need to be cut</u> to solve the integer programming problem using the following 6 optimized cutting patterns:

$\boldsymbol{x_7}$ = 26 [i.e. big roll fits 1x type-2 rolls, 1x type-3 rolls, and 3x type-4 rolls]
$\boldsymbol{x_8}$ = 4 [i.e. big roll fits 5x type-2 rolls, 1x type-6 rolls]
$\boldsymbol{x_9}$ = 17 [i.e. big roll fits 1x type-1 rolls, 1x type-4 rolls, and 7x type-6 rolls]
$\boldsymbol{x_{10}}$ = 10 [i.e. big roll fits 5x type-1 rolls, 1x type-2 rolls, and 2x type-5 rolls]
$\boldsymbol{x_{11}}$ = 18 [i.e. big roll fits 4x type-3 rolls, 2x type-5 rolls]
$\boldsymbol{x_{12}}$ = 9 [i.e. big roll fits 2x type-1 rolls, 5x type-5 rolls]

Question 2

__2. Product mix with learning.__ The Acme Company is planning the production of its two products, the popular Widget and the new SuperWidget soon to be introduced to the market. Each Widget sells for \$40 and requires \$7 for materials. Each SuperWidget will require \$8 of materials and, during the planning period, will be sold at the promotional price of \$30. Widgets and SuperWidgets each go through two machining operations. For the planning period, machine 1 is limited to a total of 320 hours and machine 2 to a total of 240 hours. Labour costs \$5 per hour on each machine. Each Widget takes four hours on machine 1, and two hours on machine 2. Producing a total of x SuperWidgets during the planning period will take a total of $t(x)$ hours on machine 1, and the same amount $t(x)$ on machine 2. The function $t(.)$ is a continuous, increasing function of the cumulative production x and, because of learning effects, it grows less than linearly. Namely, the larger the accumulated production x of SuperWidgets, and the more experienced the workers will be in machining the new product. Thus $t(.)$ is concave and increasing. Based on past experience with similar products, the following time estimates are available:

x (SuperWidgets)	0	30	60	120	240
$t(x)$ (hours)	0	60	96	153.6	245.76

Assume that any quantity of both products that Acme can produce will be sold during the planning period at the specified prices. The (short-sighted) goal of the Acme Company is to produce the combination of Widgets and SuperWidgets that will maximize its total contribution margin (revenues from sales, minus materials and labour costs) for the planning period. Using an appropriate (concave and increasing) approximation for $t(.)$, formulate as a mixed integer linear programming problem. (Note: there is no need to constrain the quantities of Widgets and SuperWidgets to be integer, a solution with fractional values is acceptable; i.e., you may use continuous variables for these.) Solve using the computer.

Below is the mathematical formulation of the mixed integer program that characterizes Acme's production problem. Given the data relating the quantity of product produced and its associated production time, we estimate the function t(x) piecewise linearly.

Decision Variables

W = number of units of Widgets produced

SW = number of units of Super Widgets produced

T = Production time required for Super Widgets given the quantity produced

U_1= binary indicator variable for whether at least 30 SW are produced

U_2=binary indicator variable for whether at least 60 SW are produced

U_3= binary indicator variable for whether at least 120 SW are produced

G_1= continuous variable indicating the quantity of SW produced below the [0,30] unit threshold

G2= continuous variable indicating the quantity of SW produced within the [30,60] unit threshold.

G3= continuous variable indicating the quantity of SW produced within the [60,120] unit threshold

G4= continuous variable indicating the quantity of SW produced above the 120 unit threshold.

Constraints

1) **Machine 1 hours**: total production time on machine 1 used to produced W and SW units must not exceed 320 hours.

$$4W+T \leq 320 \quad \text{(hours)}$$

2) **Machine 2 hours**: total production time on machine 2 used to produce U and SW units must not exceed 240 hours.

$$2W + T \leq 240 \quad (hours)$$

3) **SW Production Intervals**: we define the support of each 'piece' of the piecewise linear production time function.

$$G1 \leq 30 \quad (SW\ units)$$
$$G2 \leq 30 \quad (SW\ units)$$
$$G3 \leq 60 \quad (SW\ units)$$

4) **Total SW Production Quantity**: Let SW be defined by the sum of SW units produced within each support segment of the piecewise linear production time function.

$$SW = G1 + G2 + G3 + G4 \quad (units)$$

5) **Total SW Production Time:** Let T be defined by the sum of the G1,G2,G3 support intervals of the piecewise linear function times the respective slopes of each support's segment of the function.

$$T = 2G_1 + 1.2G_2 + 0.96G_3 + 0.768G_4 \quad (hours)$$

6) **Active Support Interval Sequence:** the support intervals must be activated in the correct ordinal sequence.

$$U2 \leq U1$$
$$U3 \leq U2$$

7) **Active Support Interval bounds**: support intervals must take on permitted values based on the given piecewise linear function, and bounds must be activated in their correct ordinal sequence.

$$30U_1 \leq G1 \quad (SW\ units)$$

$$30U_2 \leq G2 \leq 30U_1 \quad (SW\ units)$$

$$60U_3 \leq G3 \leq 60U_2 \quad (SW\ units)$$

$$G4 \leq 240U_3 \quad (SW\ units)$$

8) **Non-negativity**: All decision variables take on non-negative values.

$$W, SW, T, G1, G2, G3, G4, U1, U2, U3 \geq 0$$

9) **Binary Constraints**: Indicator variables for 'kinks' in piecewise linear production time function for SW take on values of {0,1} only.

$$U1, U2, U3 \in \{0,1\}$$

Objective

Find values of SW,W,T so as to maximize profits from the production of W and SW units

$$Maximize\ \ 3W + 22SW - 10T$$

Full Formulation

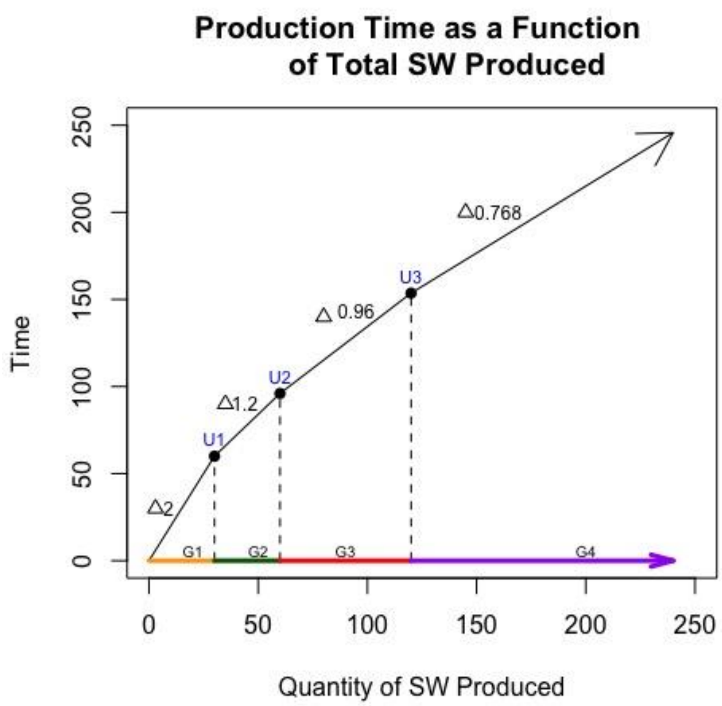
Find quantities of W,SW, and T so as to:

$$Maximize\ \ 3W + 22SW - 10T$$

Subject to:

$$4W + T \leq 320 \quad (Machine\ 1\ hours)$$
$$2W + T \leq 240 \quad (Machine\ 2\ hours)$$
$$G1 \leq 30 \quad (SW\ units)$$
$$G2 \leq 30 \quad (SW\ units)$$
$$G3 \leq 60 \quad (SW\ units)$$
$$SW - G1 - G2 - G3 - G4 = 0 \quad \text{(SW units)}$$
$$T - 2G_1 - 1.2G_2 - 0.96G_3 - 0.768G_4 = 0 \quad (hours)$$
$$U2 - U1 \leq 0 \quad (first\ and\ second\ interval\ activation\ order)$$
$$U3 - U2 \leq 0 \quad (second\ and\ third\ interval\ activation\ order)$$
$$30U_1 - G1 \leq 0 \quad (SW\ units\ in\ first\ interval)$$
$$30U_2 \leq G2 \leq 30U_1 \quad (SW\ units\ in\ second\ interval)$$
$$60U_3 \leq G3 \leq 60U_2 \quad (SW\ units\ in\ third\ interval)$$
$$G4 - 240U_3 \leq 0 \quad (SW\ units\ in\ fourth\ interval)$$
$$W, SW, T, G1, G2, G3, G4, U1, U2, U3 \geq 0 \quad (non-negativity)$$
$$U1, U2, U3 \in \{0,1\} \quad \text{(binary)}$$

Below is a graphic illustration of the piece wise linear function used to approximate T(x) based in the data given. The MIP is formulated based on the interval bounds of this function.



Solving the MIP using AMPL's Gurobi solver, I obtained the following results:

```
ampl: solve;
Gurobi 7.0.0: optimal solution; objective 2715
9 simplex iterations
plus 9 simplex iterations for intbasis
ampl: display w,sw,T;
w = 0
sw = 232.5
T = 240

ampl: display u1,u2,u3;
u1 = 1
u2 = 1
u3 = 1

ampl: display g1,g2,g3,g4;
g1 = 30
g2 = 30
g3 = 60
g4 = 112.5
```

A total of 232.5 Super Widgets were produced while zero Widgets were produced. The total production time spent manufacturing Super Widgets was 240 hours (all machine 2 time available). The objective value was found to be $2,715, which represents the maximized profits.