Solving the Assortment and Trim Loss Problem with Branch-and-Price-and-Cut

Cláudio Alves* and J.M. Valério de Carvalho†
Departamento de Produção e Sistemas, Escola de Engenharia, Universidade do Minho
4710-057 Braga, Portugal
*Email: claudio@dps.uminho.pt
† Email: vc@dps.uminho.pt

***Abstract*— In this paper, we address the combined assortment and trim loss minimization problem. In real settings, when different stock lengths are available, a common problem is to select a subset of stock lengths from which to cut the ordered items. In the cutting and packing literature, this problem is known as the assortment problem. Solving the assortment and trim loss minimization problem in a single stage yields better cutting plans since one takes into account the whole set of stock lengths while computing the best set of cutting patterns.**

Here, we present an exact branch-and-price-and-cut algorithm to solve both the assortment and the trim loss minimization problem in a single stage. We use an integrated Integer Linear Programming model to compute lower bounds at the nodes of the branch-and-bound tree, and we derive a robust branching scheme to find the integer optimal solution. We report on an extensive set of computational experiments for random instances.

***Keywords*— Cutting Stock Problem; Assortment Problem; Branch-and-Price-and-Cut.**

I. Introduction

A problem that is directly related to the multiple length cutting stock problem is to select from a set of possible stock lengths the assortment that will be used in the cutting plan. Indeed, even when a production manager has access to different stock lengths, only a limited number of different lengths can be kept in stock. Typically, the inventory holding costs increase with the dimension of the assortment. The assortment of stock lengths is a major concern since it directly impacts on the final assignment of the small items to the stock pieces.

In [8], Hinxman distinguished between the assortment and the trim loss minimization problem. Solving these two problems in a single stage yields cutting plans that are better in terms of material usage. The minimum trim loss cutting plan can be computed by taking into account all the available stock lengths and the constraint on the number of different stock lengths that can be used. The cutting patterns are not restricted to a pre-selected (usually suboptimal) set of stock lengths.

Recently, some researchers tried to derive efficient solution algorithms for the combined trim loss and assortment problem. These approaches are essentially heuristic. New results concerning the cutting stock problem with different stock types were also published recently [1], [2]. The authors proposed different schemes to accelerate the generation of columns, and they derived a robust branch-and-price-and-cut algorithm to exactly solve this type of problems. They also used for the first time valid dual inequalities [11] at all the nodes of a branching tree.

In [4], the authors proposed a binary non-linear model for the trim loss and assortment problem with inventory holding costs and service levels. Their solution method relies on a Dantzig-Wolfe reformulation with a linear restricted master problem and non-linear pricing subproblems. The former is solved by column generation, while the subproblems are solved heuristically using a marginal cost procedure. Their algorithm may return non-optimal solutions. A major drawback of their approach is the complexity of the pricing subproblems. Ideally, the pricing subproblems should remain computationally tractable since they have to be solved at each column generation iteration. Another drawback is that branching is done on the variables of the reformulated model. It is well known that this may cause the regeneration of columns, thus forcing to find the second, third (and so on) most attractive column. Complexity increases as the algorithm goes into deeper nodes of the branching tree.

Arbib and Marinelli [3] developed a heuristic algorithm to solve a particular 2-dimensional assortment and trim loss minimization problem for an Italian plant. They used an approximation based on a p-median model with additional constraints, and they compared their approach with the solutions provided by the operators. For all the instances tested, the authors obtained solutions with a smaller average trim loss. However, the problem they addressed is quite restrictive since each item size is forced to be cut from the same type of stock.

Other heuristics have been proposed for the trim loss minimization and assortment problem. Some of them address the two objectives simultaneously, while others solve them separately [9]. There are very few results concerning the computation of good and efficient lower bounds or related to exact methods. A weak lower bound was proposed by Holthaus [9].

In this paper, we propose an exact branch-and-price-and-cut algorithm to solve the assortment and trim loss minimization problem in a single stage. We use a column generation model inspired on the well-know Gilmore and Gomory model [6], [7] for the standard cutting stock problem to derive lower bounds at the nodes of the branch-and-bound tree. The columns of the model are mainly related to feasible cutting patterns. A set of binary variables are used to determine wether a particular stock length is used or not. Our branching scheme is based on these binary variables, and on the variables of an original formulation. It is well-known that branching on the variables of an original formulation does not induce any intractable complexity to the pricing subproblems [10]. The details of the Integer Linear Programming (ILP) model are described in Section II. The algorithm is presented in Section III. The cutting planes that we used to strengthen the ILP model are discussed in Section IV. Computational results are reported on Section V.

II. An ILP Formulation

The assortment and trim loss minimization problem is characterized by K stock types with a length of W_k units. There are B_k units of stock type k. The items to cut from these stock pieces have a size of w_i, $i = 1, \dots, m$, and there is a demand of b_i units for the item size i. The maximum number of stock types that can be used in a cutting plan is restricted to K'.

The assortment and trim loss minimization problem can be modeled using a column generation formulation. The model has an exponential number of columns (say p). We will use r, $r = 1, \dots, p$ to index the columns of the model. The cutting patterns consist in vectors with the form $(a_{1kr}, a_{2kr}, \dots, a_{mkr}; \dots, 1, \dots; 0)^T$. The coefficients a_{ikr} determine how often an item of size w_i is cut from a stock piece of length W_k in the pattern r. The set of cutting patterns associated to a stock type k is denoted by P^k. All the data is assumed to be integer. The λ_p^k variables determine the number of times the cutting pattern p defined over a stock of length W_k is used. The binary variables μ_k, $k = 1, ..., K$ indicate if a stock piece of length W_k is used or not.

	λ_1^1	λ_2^1	λ_1^2	λ_2^2	λ_1^3	λ_2^3	μ_1	μ_2	μ_3		
$w_i = 3$	2		1		1					$\geq$	3
2		3	1	2		2				$\geq$	3
1	1	1		1	1					$\geq$	2
$W_k = 7$	1	1					-2			$\leq$	0
5			1	1				-1		$\leq$	0
4					1	1			-3	$\leq$	0
							1	1	1	$\leq$	2
	7	7	5	5	4	4	0	0	0		

Fig. 1. Restricted Master (Example 2.1)

The ILP model for the assortment and trim loss minimization problem is as follows:

$$\min \quad \sum_{k=1}^{K} \sum_{p \in P^k} W_k \lambda_p^k \tag{1}$$

s.to

$$\sum_{k=1}^{K} \sum_{p \in P^k} a_{ip}^k \lambda_p^k \geq b_i, \;\; i = 1, ..., m, \tag{2}$$

$$\sum_{p \in P_k} \lambda_p^k \leq B_k \mu_k, \;\; k = 1, ..., K, \tag{3}$$

$$\sum_{k=1}^{K} \mu^k \leq K', \tag{4}$$

$$\lambda_p^k \geq 0, \text{and integer}, k = 1, ..., K, p \in P^k, \tag{5}$$

$$\mu_k \in \{0, 1\}, \;\; k = 1, ..., K. \tag{6}$$

The following example illustrates a small instance of the assortment and trim loss problem and its corresponding model (1)-(6).

Example 2.1: Consider an instance with the following set of stock lengths $W = (7, 5, 4)$ and corresponding availabilities $B = (2, 1, 3)$, and a set of items sizes $w = (3, 2, 1)$ with corresponding demands $b = (3, 3, 2)$. Assume that only two stock lengths can be used.

Figure 2.1 represents a valid restricted master related to formulation (1)-(6) for this instance. □

Given the number of columns of (1)-(6), for any practical instance, the model can be solved only by dynamic column generation. The pricing subproblems remain bounded knapsack problems even if the master problem has K more columns than the classical column generation formulation for the multiple length cutting stock problem [1]. In fact, the dual variable associated to (4) has no incidence in the dual price of the other columns (cutting patterns). A pricing subproblem must be solved for each stock length W_k. Let π and δ denote the dual variables associated to (3) and (4), respectively. The pricing subproblem related to a stock of length W_k reads:

$$\begin{aligned} \max \quad & z_{SP}^k \sum_{i=1}^{m} \pi_i a_i \\ \text{s.to} \quad & \\ & \sum_{i=1}^{m} w_i a_i \leq W_k, \\ & a_i \leq b_i, \; i = 1, \dots, m, \\ & a_i \geq 0, \text{ and integer.} \end{aligned}$$

The decision variables are denoted by a_i, $i = 1, \dots, m$. They represent the number of items of size w_i that are added to the knapsack. A cutting pattern is attractive only if $z_{SP}^k + \delta_k < W_k$. In our implementation, only the most attractive column for each stock length is added to the master in each iteration.

The pricing subproblem has not the integrality property, and hence, the bound given by the linear programming relaxation (LP) of (1)-(6) may be improve the bound of other original formulations.

III. Branch-and-Bound

At each node of the branch-and-bound tree, we solve the LP relaxation of (1)-(6). We start with a model that has an artificial column plus K columns for the μ_k binary variables. The first restricted master problem is further initialized with columns obtained with a First-Fit Decreasing heuristic that only takes into account the K' largest stock lengths.

Our branching scheme has two levels. We branch first on the μ_k variables by imposing the following constraints each time μ_k is fractional:

$$\mu_k = 0, \tag{7}$$

and

$$\mu_k = 1. \tag{8}$$

When constraint (7) is enforced, stock type k is excluded from the optimal cutting plan, and hence, this stock type can be removed from the instance. When (8) holds, stock type k is assumed to be used, and hence only $K'-1$ stock types other than k can effectively be used. However, in practice, it may happen that there is no cutting patterns associated to stock type k in the optimal cutting plan.

If all the variables μ_k are integer, we proceed by branching on the variables of an arc-flow model for the multiple length cutting stock problem [2]. This arc-flow model is defined as follows:

$$\min. \sum_{k=1}^{K} W_k z_k \tag{9}$$

subject to

$$-\sum_{(r,s)\in A} x_{rs} + \sum_{(s,t)\in A} x_{st} = \begin{cases} \sum_{k=1}^{K} z_k, & \text{if } s = 0, \\ -z_k, & \text{for } s = W_k, \; k = 1, \dots, K, \\ 0, & \text{otherwise,} \end{cases} \tag{10}$$

$$\sum_{(r,r+w_i)\in A} x_{r,r+w_i} \geq b_i, \; i = 1, \dots, m \tag{11}$$

$$z_k \leq B_k, \; k = 1, \dots, K, \tag{12}$$

$$x_{rs} \geq 0 \text{ and integer}, \; \forall (r,s) \in A, \tag{13}$$

$$z_k \geq 0 \text{ and integer}, \; k = 1, \dots, K. \tag{14}$$

In (9)-(14), the multiple length cutting stock problem is formulated as a minimum weighted flow problem, with additional constraints on the items' demand (11) and rolls' availability (12). Constraints (10) are the flow conservation constraints. The x_{ij} variables represent the number of items of size $j - i$ placed at position i from the leftmost border of the roll. The z_k variables denote the number of rolls of length W_k that are used.

The fractional λ_p^k variables of the linear relaxation of (1)-(6) can be converted into a set of flows in the arc-flow model. Indeed, cutting patterns can be seen as paths in a graph. Each arc in these paths correspond to a particular item within the pattern. In our implementation, the items of a cutting pattern are converted into arc flows in decreasing order of their sizes. Then the variables are checked for integrality. Branching constraints are enforced on the x_{ij} arc-flow variables, and on the backward arcs (z_k variables). This scheme allows us to implicitly determine a subset of cutting patterns (λ_p^k variables) on which to branch [12].

The branching nodes are selected in a depth first manner. At most two nodes are created when branching, and the one with (8) or a "greater than or equal to" branching constraint is selected first. At the root node, there are K knapsack subproblems to solve per iteration of the column generation procedure. These subproblems can be solved by running once a pseudo-polynomial dynamic programming algorithm. At the other nodes, even when constraints of type (7) are enforced, attractive patterns can still be priced out in a single run of a dynamic programming algorithm.

IV. Cutting Planes

An optimal solution to the assortment and trim loss minimization problem must correspond to an integer combination of at most K' stock lengths. Strong cutting planes for the LP relaxation of (1)-(6) can be derived by using this principle. When we know the value of the continuous bound given by (1)-(6), we just have to find the smallest integer combination of stock lengths larger than or equal to this bound. The total length of this combination is clearly a lower bound for the value of the optimum, and it may improve the continuous bound of the corresponding assortment and trim loss minimization problem.

At a node q of the branch-and-bound tree, the continuous bound given by the LP relaxation of (1)-(6) (say z^q_{LP}) may be dominated by z^q_{IP}, the optimal value of the following optimization problem:

$$z^q_{IP} = \min \quad \sum_{k=1}^{K} W_k y_k \tag{15}$$

$$\text{subject to} \quad \sum_{k=1}^{K} W_k y_k \geq \left\lceil z^q_{LP} \right\rceil, \tag{16}$$

$$y_k \leq B_k \mu_k, \; k = 1, \ldots, K, \tag{17}$$

$$\sum_{k=1}^{K} \mu^k \leq K', \tag{18}$$

$$y_k \geq 0 \text{ and integer}, k = 1, ..., K \tag{19}$$

A constraint can be enforced in the LP master at each node of the branch-and-bound tree, forcing the LP optimum to be greater than or equal to z^q_{IP}. The right hand side of this cut is computed with a dynamic programming algorithm that has a state space defined as follows: $(n, level)$, where n is the number of stock lengths used, and $level$ identifies a reachable length obtained by combining no more than K' different stock lengths.

The stock lengths for which there is a branching constraint of type (8) are treated first in our implementation of the dynamic programming algorithm. These stock lengths may not be part of the combination. However, transition between two states in the stages related to these stock lengths is always done from a state (n, l_1) to another state $(n+1, l_2)$, with l_1 not necessarily different from l_2. The stock lengths for which a branching constraint of type (7) has been enforced are removed from the instance, and obviously they are not considered in the computation of the above cut.

V. Computational Experiments

A set of computational experiments were conducted on 160 random instances. These tests were performed on a 3GHz Pentium IV computer with 512MBytes of RAM. To generate the test problems, we used the CUTGEN1 generator described in [5], with a seed equal to 1994. Sixteen groups of ten instances were used. Their main characteristics are summarized in Table I. The instances have at most 50 different item sizes and between 5 and 20 stock lengths in the interval $[100, 300]$. The average demand per item type is always 10 units. Hence, for the instances with $m = 50$, for example, there will be a total of 500 items to cut from the rolls. In the subsequent tables, m represents the parameter of the CUTGEN1 generator that is related to the number of item sizes, while $\overline{m}$ is the real average number of different item sizes in the instances.

For each problem set, our algorithm was run four times. In the first run, all the K stock lengths can be used, and the problems reduce to the standard multiple length cutting stock problem. In the remaining three runs, we restrict the set of stock lengths to 75%, 50% and 25% of K, respectively. In the subsequent tables, the column designated by K' identifies the respective percentage of stock lengths that can be used.

Table II reports on the average computational results obtained with each problem set. Note that these averages do not take into account the instances that were not solved within a time limit of 900 seconds.

The entries of the Table II are the following:

- $cols_{IN}$: number of columns before column generation;
- sp_{LP}: number of pricing subproblems solved before branching;
- $cols_{LP}$: number of generated columns during the resolution of the LP relaxation;
- sp_{BB}: number of pricing subproblems solved in the branch-and-bound phase;
- $cols_{BB}$: number of generated columns in the branch-and-bound phase;
- nod_{BB}: number of branching nodes explored;
- t_{PP}: time in seconds spent with preprocessing (FFD type heuristic);
- t_{LP}: solution time in seconds for the LP relaxation;
- t_{BB}: time in seconds spent with branch-and-bound;
- t_{TOT}: total computing time in seconds;
- opt: number of instances solved to optimality.

Our algorithm found the optimal integer solution within the time limit for 97% of the instances. On average, the computing times are rather small. Some of the high values that appear are essentially due to a very small number of instances in the sets for which the algorithm performed poorly.

TABLE I
CHARACTERISTICS OF THE RANDOM INSTANCES

Set	m	K	v_1	v_2	$\bar{b}$
1	20	5	0.1	0.8	10
2		10			
3		15			
4		20			
5	30	5	0.1	0.8	10
6		10			
7		15			
8		20			
9	40	5	0.1	0.8	10
10		10			
11		15			
12		20			
13	50	5	0.1	0.8	10
14		10			
15		15			
16		20			

VI. CONCLUSIONS

In this paper, we described a new exact algorithm for the assortment and trim loss minimization problem, which is based on column generation, branch-and-bound and cutting planes. The branching scheme adopted is robust in the sense that it does not induce any intractable modification to the pricing subproblems. We conducted a set of computational experiments on instances generated randomly using a publicly available generator. Our results show that the problem can be solved efficiently in a reasonable amount of time. The main element contributing to these results seems to be the cutting planes that are enforced in the LP master. The cuts become stronger as one goes deeper in the branching tree, since some of the branching constraints may have an impact on its value.

ACKNOWLEDGEMENTS

This work was partially supported by the portuguese Science and Technology Foundation (Projecto POSC/ 57203/ EIA/ 2004) and by the Algoritmi Research Center of the University of Minho, and was developed in the Industrial and Systems Engineering Group.

REFERENCES

[1] C. Alves and J. M. Valério de Carvalho, “Accelerating column generation for variable sized bin-packing problems,” *European Journal of Operational Research (in press - doi:10.1016/j.ejor.2005.07.033)*, 2005.

[2] ——, “A stabilized branch-and-price-and-cut algorithm for the multiple length cutting stock problem,” *Computers and Operations Research (in press - doi:10.1016/j.cor.2006.08.014)*, 2006.

[3] C. Arbib and F. Marinelli, “An optimization model for trim loss minimization in an automotive glass plant,” *European Journal of Operational Research (in press)*, 2006.

[4] A. Chauhan, A. Martel, and S. Amour, “Roll assortment optimization in a paper mill: An integer programming approach,” *Computers and Operations Research (in press)*, 2006.

[5] T. Gau and G. Waescher, “CUTGEN1: A problem generator for the standard one-dimensional cutting stock problem,” *European Journal of Operational Research*, vol. 84, pp. 572–579, 1995.

[6] P. C. Gilmore and R. E. Gomory, “A linear programming approach to the cutting stock problem,” *Operations Research*, vol. 9, pp. 849–859, 1961.

[7] ——, “A linear programming approach to the cutting stock problem: Part II,” *Operations Research*, vol. 11, pp. 863–888, 1963.

[8] A. I. Hinxman, “The trim-loss and assortment problems: A survey,” *European Journal of Operational Research*, vol. 5, pp. 8–18, 1980.

[9] O. Holthaus, “On the best number of different standard lengths to stock for one-dimensional assortment problems,” *International Journal of Production Economics*, vol. 83, pp. 233–246, 2003.

[10] M. E. Luebbecke and J. Desrosiers, “Selected topics in column generation,” *Operations Research*, vol. 53, pp. 1007–1023, 2005.

[11] J. M. Valério de Carvalho, “Using extra dual cuts to accelerate convergence in column generation,” *INFORMS Journal on Computing*, vol. 17, no. 2, pp. 175–182, 2005.

[12] F. Vanderbeck, “On Dantzig-Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm,” *Operations Research*, vol. 48, no. 1, pp. 111–128, 2000.

TABLE II
COMPUTATIONAL RESULTS FOR RANDOM INSTANCES

Set	m	$\overline{m}$	K	K'	$cols_{IN}$	sp_{LP}	$cols_{LP}$	sp_{BB}	$cols_{BB}$	nod_{BB}	t_{PP}	t_{LP}	t_{BB}	t_{TOT}	opt
1	20	17.20	5.00	100	23.80	15.00	60.70	13.20	6.50	11.50	0.02	0.01	0.10	0.13	10
		17.20	5.00	75	23.80	15.00	60.70	30.40	33.60	21.30	0.02	0.01	0.15	0.17	10
		17.20	5.00	50	23.80	15.00	60.70	36.60	31.50	26.40	0.02	0.01	0.14	0.17	10
		17.20	5.00	25	23.80	15.00	60.70	83.80	46.20	62.80	0.02	0.01	0.17	0.20	10
2	20	17.20	10.00	100	28.80	10.30	87.50	33.80	43.00	28.10	0.02	0.01	0.71	0.75	10
		17.20	10.00	75	28.80	10.30	87.50	41.60	93.20	29.50	0.02	0.01	0.62	0.65	10
		17.20	10.00	50	28.80	10.30	87.50	55.50	125.90	31.50	0.02	0.01	0.39	0.42	10
		17.20	10.00	25	28.80	10.30	87.50	72.10	150.40	35.70	0.02	0.01	0.32	0.35	10
3	20	17.20	15.00	100	33.80	10.00	130.70	54.50	112.50	44.00	0.02	0.02	2.15	2.19	10
		17.11	15.00	75	33.67	10.00	130.56	57.67	137.00	43.33	0.02	0.02	1.59	1.63	9
		17.20	15.00	50	33.80	10.00	130.70	99.80	239.80	57.80	0.03	0.02	1.39	1.44	10
		17.20	15.00	25	33.80	10.00	130.70	139.40	263.40	83.20	0.02	0.02	1.03	1.08	10
4	20	17.20	20.00	100	38.80	9.20	161.10	86.40	190.50	71.50	0.02	0.04	5.53	5.59	10
		16.88	20.00	75	38.88	9.13	159.00	97.00	242.88	73.38	0.02	0.02	4.21	4.25	8
		17.20	20.00	50	38.80	9.20	161.10	362.50	268.40	262.70	0.03	0.03	7.70	7.75	10
		17.22	20.00	25	38.78	9.56	168.00	162.89	499.22	80.00	0.02	0.03	1.81	1.86	9
5	30	25.00	5.00	100	33.50	15.80	70.90	26.70	30.00	19.60	0.05	0.02	0.30	0.36	10
		25.00	5.00	75	33.50	15.80	70.90	43.00	68.70	24.40	0.05	0.01	0.29	0.36	10
		25.00	5.00	50	33.50	15.80	70.90	37.50	58.10	19.70	0.05	0.02	0.19	0.26	10
		25.00	5.00	25	33.50	15.80	70.90	118.00	146.00	59.50	0.05	0.01	1.34	0.60	10
6	30	25.00	10.00	100	38.50	12.30	112.30	51.80	77.70	42.40	0.05	0.03	1.78	1.85	10
		25.00	10.00	75	38.50	12.30	112.30	68.50	160.70	46.90	0.04	0.03	1.56	1.63	10
		25.00	10.00	50	38.50	12.30	112.30	152.30	296.20	83.40	0.04	0.03	1.71	1.78	10
		25.00	10.00	25	38.50	12.30	112.30	130.80	288.30	58.80	0.05	0.03	0.98	1.06	10
7	30	25.00	15.00	100	43.50	11.30	153.90	255.90	201.60	234.70	0.05	0.05	15.76	15.86	10
		25.00	15.00	75	43.50	11.30	153.90	99.90	316.70	72.50	0.05	0.04	4.58	4.68	10
		25.00	15.00	50	43.44	11.00	149.56	1030.67	547.56	710.22	0.05	0.05	25.46	25.56	9
		25.00	15.00	25	43.50	11.30	153.90	291.40	862.10	132.40	0.06	0.04	4.05	4.15	10
8	30	25.00	20.00	100	48.50	10.60	191.40	111.00	332.80	87.10	0.05	0.07	10.96	11.07	10
		25.00	20.00	75	48.50	10.60	191.40	126.30	468.80	93.00	0.05	0.07	8.39	8.50	10
		24.63	20.00	50	48.00	10.13	182.00	223.63	619.50	118.88	0.05	0.07	6.97	7.09	8
		25.00	20.00	25	48.50	10.60	191.40	765.00	1552.10	418.60	0.05	0.07	17.54	17.66	10
9	40	30.00	5.00	100	37.40	18.70	86.20	47.00	52.40	35.40	0.06	0.02	0.88	0.96	10
		30.00	5.00	75	37.40	18.70	86.20	41.70	64.80	27.30	0.06	0.02	0.55	0.63	10
		30.00	5.00	50	37.40	18.70	86.20	182.00	132.60	139.70	0.07	0.02	1.70	1.78	10
		30.00	5.00	25	37.40	18.70	86.20	115.00	169.70	48.50	0.06	0.02	0.48	0.56	10
10	40	30.00	10.00	100	42.40	13.70	126.70	123.70	186.30	96.70	0.06	0.04	6.23	6.34	10
		30.00	10.00	75	42.40	13.70	126.70	130.40	337.80	81.70	0.06	0.05	4.36	4.47	10
		30.00	10.00	50	42.40	13.70	126.70	145.80	367.40	67.80	0.06	0.04	2.33	2.43	10
		30.00	10.00	25	42.40	13.70	126.70	228.60	490.20	102.80	0.06	0.03	2.37	2.47	10
11	40	30.11	15.00	100	47.33	13.11	181.33	1015.00	389.22	972.22	0.06	0.08	98.82	98.96	9
		30.00	15.00	75	47.40	13.00	179.70	252.50	598.70	177.10	0.06	0.06	16.40	16.52	10
		30.00	15.00	50	47.40	13.00	179.70	311.50	751.90	151.70	0.06	0.08	9.62	9.76	10
		30.00	15.00	25	47.40	13.00	179.70	2063.10	1568.10	1624.00	0.06	0.09	53.15	53.29	10
12	40	30.11	20.00	100	52.33	12.78	234.89	188.89	603.44	137.67	0.06	0.12	26.42	26.60	9
		29.88	20.00	75	52.25	13.13	241.75	209.25	756.38	141.13	0.06	0.12	19.69	19.88	8
		29.71	20.00	50	52.14	12.00	219.14	310.71	1003.43	148.57	0.06	0.11	14.46	14.63	7
		29.78	20.00	25	52.44	12.11	221.56	556.89	2724.89	196.89	0.07	0.11	17.41	17.58	9
13	50	36.10	5.00	100	43.60	20.30	95.40	65.80	60.60	50.30	0.07	0.03	1.68	1.79	10
		36.10	5.00	75	43.60	20.30	95.40	83.50	131.60	49.80	0.08	0.03	1.36	1.46	10
		36.10	5.00	50	43.60	20.30	95.40	1129.70	189.20	1039.20	0.08	0.03	20.27	20.37	10
		36.10	5.00	25	43.60	20.30	95.40	112.70	191.90	41.40	0.08	0.03	0.57	0.68	10
14	50	36.10	10.00	100	48.60	15.50	144.30	148.40	262.80	112.90	0.08	0.06	10.71	10.84	10
		36.10	10.00	75	48.60	15.50	144.30	134.50	346.80	82.90	0.08	0.06	6.45	6.59	10
		36.00	10.00	50	48.56	15.44	143.67	257.89	587.56	111.78	0.08	0.07	5.31	5.45	9
		36.10	10.00	25	48.60	15.50	144.30	393.50	873.50	159.40	0.08	0.06	5.80	5.94	10
15	50	35.78	15.00	100	53.44	14.56	199.22	203.67	412.89	145.78	0.08	0.24	27.22	27.55	9
		36.10	15.00	75	53.60	14.70	200.20	278.80	688.60	170.30	0.08	0.26	26.45	26.79	10
		35.89	15.00	50	53.00	14.56	197.56	1011.56	1017.33	737.78	0.08	0.24	60.25	60.56	9
		36.10	15.00	25	53.60	14.70	200.20	1522.90	1937.90	819.10	0.08	0.18	55.86	56.12	10
16	50	36.10	20.00	100	58.60	14.50	263.00	273.40	624.80	192.20	0.09	0.46	57.43	57.99	10
		35.78	20.00	75	58.33	13.78	247.78	400.89	1195.78	221.89	0.10	0.45	49.30	49.84	9
		36.10	20.00	50	58.60	14.50	263.00	768.40	1746.50	368.00	0.10	0.38	50.07	50.54	10
		36.00	20.00	25	58.22	14.67	265.56	994.33	4125.00	312.00	0.09	0.30	40.76	41.15	9