Mixed Integer Programming Models for Detailed Placement

Shuai Li, Cheng-Kok Koh
ECE, Purdue University
{li263, chengkok}@purdue.edu

Outline

- Background
- MIP models for detailed placement
- Experimental results
- Conclusion

Detailed Placement

- Placement of standard-cell circuits
  - global placement
  - legalization
  - detailed placement
- Objective: Half-perimeter wirelength (HPWL)
- Discrete optimization problem with solution space $O(n!)$, where n is the number of cells
  - Horizontal white space allocation technique

    dynamic programming algorithm for the placement of cells in one row in fixed order
  - Cell swap technique using the concept of optimal region
  - In Domino, detailed placement is transformed into transportation problem solved with a network flow algorithm
  - Sliding window technique

Sliding Window Technique

- Divide-and-conquer
  - the whole chip is partitioned into overlapping windows
  - enumeration approach for the optimization of each window, usually with no more than 6 cells
- Mixed Integer Programming (MIP) approach
  - constrained optimization problem
    - linear objective function & linear constraints
    - integer variables
  - MIP proves to be powerful in solving various NP-hard problems

    e.g. Travelling Salesmen Problem
  - formulate the detailed placement of cells in each window into a MIP problem, solved with

    1) the *branch–and-cut* technique: widely applicable

    2) the *branch-and-price* technique: used for solving the model derived from the Dantzig-Wolfe decomposition

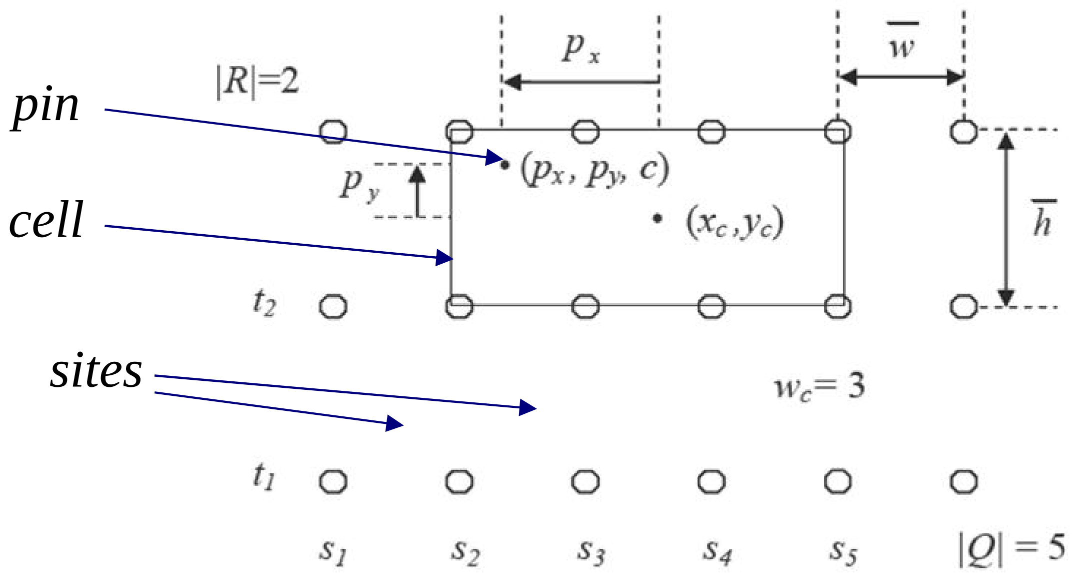
MIP-based detailed placement

- Existing MIP models for detailed placement
  - P. Ramachandaran et al, Optimal placement by branch-and-price, ASPDAC2005
    - efficient in optimizing larger regular mesh grid circuits
    - only applicable for uniform-width cells
  - S. Cauley et al, A parallel branch-and-cut approach for detailed placement, TDAES2011
    - applicable for cells with different widths
    - binary variables determining whether a site is occupied by a cell, referred to as the *site-occupation* variables
- Our contribution
  - two new MIP models applicable for cells with different width, with shorter solution time: the SCP Model, the RQ Model
  - with the ability to optimize larger windows, we developed a MIP-based detailed placer that manages to generate placement results with better HPWL and routed wirelength

Outline

- Background
- MIP models for detailed placement
- Experimental results
- Conclusion

Nomenclature



R: set of rows
Q: set of columns
C: set of cells
w_c: width of cell c
N: set of nets
P_n: set of pins of net n
(p_x, p_y, c): a pin on cell c

- Rows and columns of *sites* in each rectangular sliding window
- Uniform-height *cells* occupying integral number of contiguous sites
  - (x_c, y_c): the centroid of cell c
- *nets* connecting *pins* located on different cells
  - $(u_n^x, l_n^x, u_n^y, l_n^y)$: the bounding box for net n

S Model

- Model base on *site-occupation* variables:

  p_{crq} whether cell c occupies the site at row r and column q
- A variant of the existing model

$$\min \quad \text{HPWL} = \sum_{n \in N} (u_n^x - l_n^x + u_n^y - l_n^y)$$

$$\text{s.t.} \quad l_n^x \le x_c + p_x \le u_n^x, \qquad \forall (p_x, p_y, c) \in P_n, \forall n \in N$$

$$l_n^y \le y_c + p_y \le u_n^y, \qquad \forall (p_x, p_y, c) \in P_n, \forall n \in N$$

definition of bounding box

$$\sum_{r \in R} \sum_{q \in Q} (s_q + \overline{w}/2) p_{crq} = w_c x_c, \qquad \forall c \in C$$

$$\sum_{r \in R} \sum_{q \in Q} (t_r + \overline{h}/2) p_{crq} = w_c y_c, \qquad \forall c \in C$$

definition of cell centroid

$$\sum_{c \in C} p_{crq} \le 1, \qquad \forall r \in R, q \in Q$$

site occupation

S Model (cont'd)

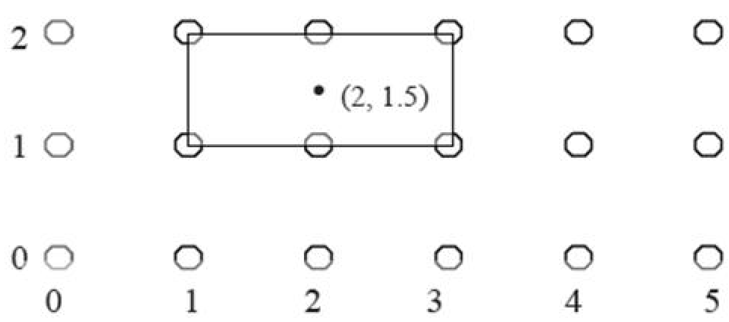
$$\sum_{r\in R}\sum_{q\in Q} p_{crq} = w_c, \qquad \forall c \in C \qquad \textit{cell occupation}$$

$$p_{crq} + p_{cr'q'} \le 1, \qquad \forall c \in C,\ \forall |q-q'| \ge w_c \text{ or } r \ne r' \qquad \textit{contiguity}$$

- branch-and-cut for solving the S Model

- Independent constraints for cell c
  - defines a set of *single-cell-placement patterns* that cell c is legally placed in the window
  - each pattern can be described with the vector of x_c, y_c, p_{crq}

    e.g. a pattern to place cell c in a 2X5 window



(2, 1.5; 0, 0, 0, 0, 0, 0, 1, 1, 0, 0)

K_c: the set of all patterns;

p_c^k: the kth pattern

$$p_c^k = (x_c, y_c; p_{crq}^k) \quad \forall r \in R, q \in Q$$

SCP Model

- Model based on binary *single-cell-placement* (SCP) variables:

$$x_c = \sum_{k\in K_c} x_c^k \lambda_c^k,\quad y_c = \sum_{k\in K_c} y_c^k \lambda_c^k,\quad p_{crq} = \sum_{k\in K_c} p_{crq}^k \lambda_c^k,\quad \sum_{k\in K_c} \lambda_c^k = 1,\quad \lambda_c^k \in \{0,1\}$$

$$\min\quad \mathbf{HPWL} = \sum_{n\in N} (u_n^x - l_n^x + u_n^y - l_n^y)$$

$$s.t.\quad l_n^x \le x_c + p_x \le u_n^x, \qquad \forall (p_x, p_y, c) \in P_n, \forall n \in N$$

$$\Rightarrow\quad l_n^x \le \left(\sum_{k\in K_c} x_c^k \lambda_c^k \right) + p_x \le u_n^x, \qquad \forall (p_x, p_y, c) \in P_n, \forall n \in N$$

$$l_n^y \le y_c + p_y \le u_n^y, \qquad \forall (p_x, p_y, c) \in P_n, \forall n \in N$$

$$\Rightarrow\quad l_n^y \le \left(\sum_{k\in K_c} y_c^k \lambda_c^k \right) + p_y \le u_n^y, \qquad \forall (p_x, p_y, c) \in P_n, \forall n \in N$$

$$\sum_{c\in C} p_{crq} \le 1, \qquad \forall r \in R, q \in Q$$

$$\Rightarrow\quad \sum_{c\in C} \left(\sum_{k\in K_c} p_{crq}^k \lambda_c^k \right) \le 1, \qquad \forall r \in R, q \in Q$$

$$\sum_{k\in K_c} \lambda_c^k = 1 \qquad \forall c \in C$$

SCP Model

- Model based on binary *single-cell-placement* (SCP) variables:

$$x_c = \sum_{k \in K_c} x_c^k \lambda_c^k, \quad y_c = \sum_{k \in K_c} y_c^k \lambda_c^k, \quad p_{crq} = \sum_{k \in K_c} p_{crq}^k \lambda_c^k, \quad \sum_{k \in K_c} \lambda_c^k = 1, \quad \lambda_c^k \in \{0,1\}$$

$$\min \quad \mathbf{HPWL} = \sum_{n \in N} (u_n^x - l_n^x + u_n^y - l_n^y)$$

$$s.t. \quad l_n^x \leq \sum_{k \in K_c} x_c^k \lambda_c^k + p_x \leq u_n^x, \qquad \forall (p_x, p_y, c) \in P_n, \forall n \in N$$

$$l_n^y \leq \sum_{k \in K_c} y_c^k \lambda_c^k + p_y \leq u_n^y, \qquad \forall (p_x, p_y, c) \in P_n, \forall n \in N$$

$$\sum_{k \in K_c} \sum_{c \in C} p_{crq}^k \lambda_c^k \leq 1, \quad \forall r \in R, q \in Q$$

$$\sum_{k \in K_c} \lambda_c^k = 1 \qquad \forall c \in C$$

SCP Model (cont'd)

- Two advantages over the S Model
  - tighter model derived from the Dantzig-Wolfe decomposition
  - fewer binary variables for cell c

    the S Model: $|R||Q|$ the SCP Model: $|R|(|Q|-w_c+1)$
- Branch-and-cut for solving the SCP Model
- As a comparison, the existing branch-and-price model
  - derived from the Dantzig-Wolfe decomposition as well
  - defined with binary *single-net-placement* variables, each of which corresponds to a legal pattern to place multiple cells connected by a single net in the window
  - has a huge number of *single-net-placement* variables
  - branch-and-price must be used for its solution

RQ Model

- Model based on *row-occupation* and *column-occupation* variables:
  - $\hat{p}_{cr}$ whether cell *c* occupies row *r*
  - $\bar{p}_{cq}$ whether cell *c* occupies column *q*

$$\min \quad \text{HPWL} = \sum_{n \in N} (u_n^x - l_n^x + u_n^y - l_n^y)$$

$$s.t. \quad l_n^x \le x_c + p_x \le u_n^x, \qquad \forall (p_x, p_y, c) \in P_n, \forall n \in N$$

$$l_n^y \le y_c + p_y \le u_n^y, \qquad \forall (p_x, p_y, c) \in P_n, \forall n \in N$$

definition of bounding box

$$\sum_{r \in R} \sum_{q \in Q} (s_q + \bar{w}/2) \bar{p}_{cq} = w_c x_c, \qquad \forall c \in C$$

$$\sum_{r \in R} \sum_{q \in Q} (t_r + \bar{h}/2) \hat{p}_{cr} = y_c, \qquad \forall c \in C$$

definition of cell centroid

$$\hat{p}_{cr} + \bar{p}_{cq} + \hat{p}_{c'r} + \bar{p}_{c'q} \le 3, \quad \forall r \in R, q \in Q, c \ne c'$$

site occupation

$$\sum_{q \in Q} \bar{p}_{cq} = w_c, \qquad \forall c \in C$$

$$\sum_{r \in R} \hat{p}_{cr} = 1, \qquad \forall c \in C$$

cell occupation

$$\bar{p}_{cq} + \bar{p}_{cq'} \le 1, \qquad \forall c \in C, \ \forall |q - q'| \ge w_c$$

contiguity

RQ Model (cont'd)

- Objective and most constraints are similar with those in the S Model, except the site occupation constraint, necessary to avoid two cells overlapping at a site:

$$\hat{p}_{cr} + \bar{p}_{cq} + \hat{p}_{c'r} + \bar{p}_{c'q} \leq 3, \quad \forall r \in R, q \in Q, c \neq c'$$

$$\Leftrightarrow \quad \hat{p}_{cr}\bar{p}_{cq} + \hat{p}_{c'r}\bar{p}_{c'q} \leq 1, \quad \forall r \in R, q \in Q, c \neq c'$$

$$\Leftrightarrow \quad \sum_{c \in C} \hat{p}_{cr}\bar{p}_{cq} \leq 1, \quad \forall r \in R, q \in Q \quad \Leftrightarrow \quad \sum_{c \in C} p_{crq} \leq 1, \quad \forall r \in R, q \in Q$$

- Advantage: fewer binary variables
  - the RQ Model: $O(|C|(|R|+|Q|))$ the S Model: $O(|C||R||Q|)$
- Disadvantage: more constraints $O(|C|^2|R||Q|)$

Outline

- Background
- MIP models for detailed placement
- Experimental results
- Conclusion

Comparison of MIP models

- Implemented with CPLEX
- Besides the three MIP models, we generalized branch-and-price model with single-net-placement variables
  - applicable for cells with different widths by adding contiguity constraints
  - branch-and-price solver for the model
- 2-row windows and 8-row windows with different numbers of cells
  - randomly extracted from benchmark circuit ibm01e
  - for each window size, over 300 windows
- Tolerance time: 40s
  - not all the windows can be fully optimized
  - the percentage of windows that are fully optimized

Comparison of MIP models (cont'd)

- Results obtained with the S, the RQ, the SCP Models

	#cell	S Model				RQ Model				SCP Mode			
		t(s)	red.	opt	o.t(s)	t(s)	red.	opt	o.t(s)	t(s)	red.	opt	o.t(s)
2-row	8	7.88	138.4	94.7%	5.28	6.27	138.5	96.3%	4.68	0.82	138.5	100%	0.82
	9	15.11	199.1	88.0%	10.12	13.46	194.1	86.0%	7.38	1.17	202.7	100%	1.17
	10	23.64	232.4	73.8%	13.17	22.83	237.7	78.6%	15.44	1.93	250.4	99.7%	1.74
	12	42.39	222.6	31.1%	20.88	41.41	205.3	28.1%	19.66	5.86	321.2	96.4%	4.22
	14	–	–	–	–	–	–	–	–	14.90	453.6	88.7%	9.34
8-row	10	4.74	133.0	98.0%	3.89	0.46	133.7	100%	0.46	0.34	133.7	100%	0.34
	11	11.46	159.9	91.8%	7.93	0.91	168.8	100%	0.91	0.65	168.8	100%	0.65
	12	15.04	192.6	87.5%	10.33	2.50	216.7	99.0%	2.03	1.22	217.1	99.7%	1.09
	14	43.18	184.0	37.9%	23.58	13.94	293.8	87.0%	8.17	3.40	305.8	98.3%	2.61
	16	–	–	–	–	28.94	250.1	59.1%	13.29	7.37	313.4	94.7%	5.03

- Results obtained with the branch-and-price model with single-net-placement variables

	#cell	t(s)	red.	opt	o.t(s)	#site
2-row	8	40.12	49.6	23.9%	28.81	2×41.3
	9	44.36	41.1	6.3%	35.61	2×44.1
8-row	10	35.64	91.5	38.5%	21.11	8×30.0
	11	43.44	93.7	22.7%	21.25	8×29.7

Experiments on large-scaled benchmark circuits

- IBM version-2 benchmark circuits
  - Ibm01(12208 cells) ~ ibm12 (68735 cells)
- Original placement results
  - detailed placement is optimized
  - each circuit is scanned by the enumeration approach for 6 times
- MIP-based detailed placer based on the SCP Model
  - scan each circuit for 5 times
  - 2-row windows or 4-row windows with no more than 13 cells in each scan
  - parallelized with *Message Passing Interface* (MPI) in Master-Slave structure
    implemented on computing system with 33 cores
    run-time 110min~570min
- Enumeration approach as a comparison
  - each circuit is further optimized for the same amount of time
- Besides HPWL, placement results are routed with Cadence WROUTE

	Original			MIP			Enumeration		
	HPWL	r-WL	#vias	HPWL	r-WL	#vias	HPWL	r-WL	#vias
ibm01e	47.58	0.6245	85848	46.19	0.6163	83873	47.20	0.6221	85702
ibm01h	47.45	0.6127	85266	45.94	0.6065	83638	47.05	0.6125	85103
ibm02e	132.00	1.6823	194294	129.49	1.7047	192181	130.70	1.7164	193997
ibm02h	128.94	1.7301	198177	127.15	1.7175	196303	128.33	1.7467	197279
ibm07e	296.46	3.3743	351669	291.72	3.3489	346840	294.67	3.3768	352340
ibm07h	289.52	3.3754	359261	285.43	3.3533	353902	288.38	3.3724	357979
ibm08e	310.64	3.7528	430905	306.91	3.7315	424543	309.44	3.7447	429130
ibm08h	306.18	3.7289	432689	302.89	3.7055	427173	305.32	3.7212	431259
ibm09e	267.25	2.9201	368283	262.68	2.8864	361312	266.14	2.9129	366435
ibm09h	260.98	2.8690	368610	256.79	2.8331	361360	260.08	2.8614	366320
ibm10e	505.19	5.6628	584716	498.05	5.6051	574144	503.06	5.6485	582531
ibm10h	498.20	5.6252	584079	492.21	5.5770	574712	496.86	5.6225	582262
ibm11e	408.05	4.3640	474085	401.23	4.3109	464933	405.75	4.3519	471439
ibm11h	395.66	4.2695	479599	390.17	4.2283	469384	394.27	4.2592	477032
ibm12e	712.10	8.1989	707431	699.91	8.0897	692005	705.20	8.1601	702873
ibm12h	697.47	8.0553	703139	686.61	7.9658	689523	692.71	8.0288	698971
Avg. Red.				**1.684%**	**0.827%**	**1.707%**	**0.549%**	**0.00002%**	**0.376%**
Norm.	1.0000	1.0000	1.0000	**0.9849**	**0.9907**	**0.9825**	**0.9946**	**0.9986**	**0.9957**

Outline

- Background
- MIP models for detailed placement
- Experimental results
- Conclusion

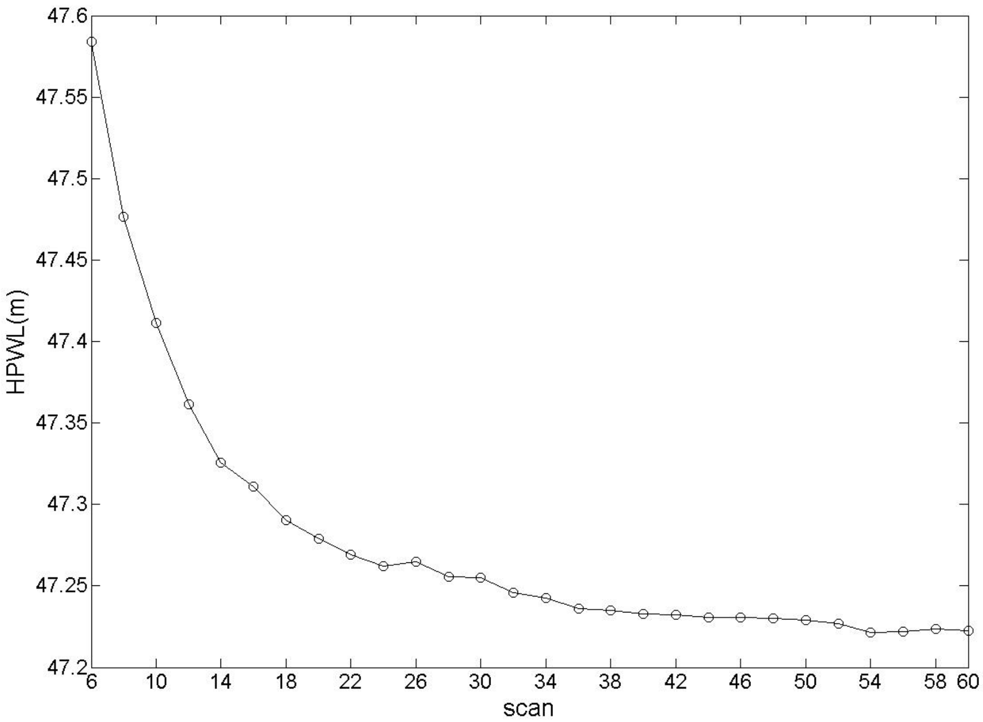
Conclusion

- MIP-based detailed placement technique
  - an alternative to the enumeration approach in the sliding window technique
- Two new MIP models for detailed placement
  - the RQ Model with fewer integer variables
  - the SCP Model derived from the Dantzig-Wolfe decomposition
  - more efficient than the S Model and the existing branch-and-price model with single-net-placement variables
  - by means of the proposed MIP models, we developed an MIP-based detailed placer that can optimize larger windows
  - results in better placement solutions in terms of HPWL, routed wirelength, and number of vias

Thank you!

P.S. The trend of HPWL reduction

- Enumeration approach
- Benchmark circuit ibm01e
  - Original placement result has been scanned for 6 times
  - HPWL after 6~60 scans



	Original			MIP			Enumeration		
	HPWL	r-WL	#vias	HPWL	r-WL	#vias	HPWL	r-WL	#vias
ibm01e	47.58	0.6245	85848	46.19	0.6163	83873	47.20	0.6243	85398
ibm01h	47.45	0.6127	85266	45.94	0.6065	83638	47.06	0.6131	85379
ibm02e	132.00	1.6823	194294	129.49	1.7047	192181	130.71	1.7144	194071
ibm02h	128.94	1.7301	198177	127.15	1.7175	196303	128.33	1.7554	197124
ibm07e	296.46	3.3743	351669	291.72	3.3489	346840	294.67	3.3781	352105
ibm07h	289.52	3.3754	359261	285.43	3.3533	353902	288.38	3.3761	358464
ibm08e	310.64	3.7528	430905	306.91	3.7315	424543	309.45	3.7456	429679
ibm08h	306.18	3.7289	432689	302.89	3.7055	427173	305.33	3.7234	430502
ibm09e	267.25	2.9201	368283	262.68	2.8864	361312	266.14	2.9137	366252
ibm09h	260.98	2.8690	368610	256.79	2.8331	361360	260.08	2.8609	366780
ibm10e	505.19	5.6628	584716	498.05	5.6051	574144	503.06	5.6515	582472
ibm10h	498.20	5.6252	584079	492.21	5.5770	574712	496.86	5.6199	582660
ibm11e	408.05	4.3640	474085	401.23	4.3109	464933	405.76	4.3520	471026
ibm11h	395.66	4.2695	479599	390.17	4.2283	469384	394.30	4.2622	476849
ibm12e	712.10	8.1989	707431	699.91	8.0897	692005	705.21	8.1555	702892
ibm12h	697.47	8.0553	703139	686.61	7.9658	689523	692.71	8.0201	698443
Avg. Red.				**1.684%**	**0.827%**	**1.707%**	**0.546%**	**-0.062%**	**0.383%**
Norm.	1.0000	1.0000	1.0000	**0.9849**	**0.9907**	**0.9825**	**0.9946**	**0.9987**	**0.9956**

P.S. Comparison of MIP models

	#cell	S Model				RQ Model				SCP Mode			
		t(s)	red.	opt	o.t(s)	t(s)	red.	opt	o.t(s)	t(s)	red.	opt	o.t(s)
2-row	8	7.88	138.4	94.7%	5.28	6.27	138.5	96.3%	4.68	0.82	138.5	100%	0.82
	9	15.11	199.1	88.0%	10.12	13.46	194.1	86.0%	7.38	1.17	202.7	100%	1.17
	10	23.64	232.4	73.8%	13.17	22.83	237.7	78.6%	15.44	1.93	250.4	99.7%	1.74
	12	42.39	222.6	31.1%	20.88	41.41	205.3	28.1%	19.66	5.86	321.2	96.4%	4.22
	14	–	–	–	–	–	–	–	–	14.90	453.6	88.7%	9.34
4-row	8	3.22	147.4	98.7%	2.51	0.59	147.8	99.7%	0.46	0.18	147.8	100%	0.18
	9	11.27	158.8	91.7%	7.32	2.71	173.3	99.3%	2.43	0.95	172.7	99.3%	0.67
	10	16.77	202.7	84.4%	10.45	7.03	222.3	94.4%	4.34	1.70	232.8	99.3%	1.33
	12	38.76	167.8	35.1%	20.33	27.37	250.1	66.9%	14.82	8.66	299.6	93.0%	5.25
	14	–	–	–	–	–	–	–	–	20.45	484.9	77.7%	9.85
8-row	10	4.74	133.0	98.0%	3.89	0.46	133.7	100%	0.46	0.34	133.7	100%	0.34
	11	11.46	159.9	91.8%	7.93	0.91	168.8	100%	0.91	0.65	168.8	100%	0.65
	12	15.04	192.6	87.5%	10.33	2.50	216.7	99.0%	2.03	1.22	217.1	99.7%	1.09
	14	43.18	184.0	37.9%	23.58	13.94	293.8	87.0%	8.17	3.40	305.8	98.3%	2.61
	16	–	–	–	–	28.94	250.1	59.1%	13.29	7.37	313.4	94.7%	5.03