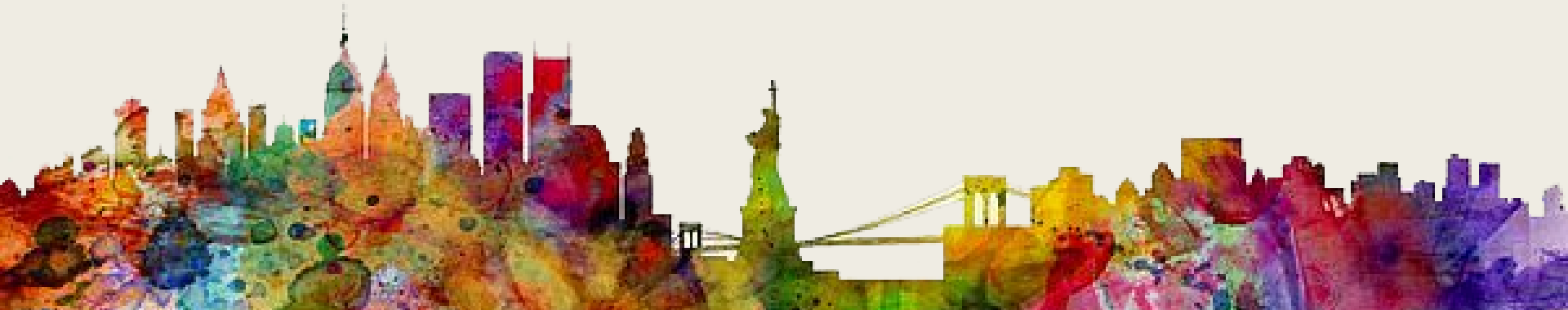




ADVERTISEMENT SCHEDULING IN

# BROADCAST TELEVISION

PRODUCTION SCHEDULING, APRIL 26, 2016 | ANDELYN RUSSELL KARA ODUM CHLOE SHIH HONGLI YANG



# BACKGROUND

---

# BACKGROUND

---

KEY PLAYERS

# BACKGROUND

---

## KEY PLAYERS



# BACKGROUND

## KEY PLAYERS

TV  
NETWORK

+

AD  
AGENCY



# BACKGROUND

## KEY PLAYERS

TV  
NETWORK

+

AD  
AGENCY

 CBS



**B | B | C**



## TV ADVERTISING

SCATTERED  
DEMAND

vs.

VIEWERSHIP  
GUARANTEES

# BACKGROUND

## KEY PLAYERS

TV  
NETWORK

+

AD  
AGENCY

 CBS



**B | B | C**



## TV ADVERTISING

SCATTERED  
DEMAND

vs.

VIEWERSHIP  
GUARANTEES

# BACKGROUND

## KEY PLAYERS



+



HOW IT  
WORKS

## TV ADVERTISING

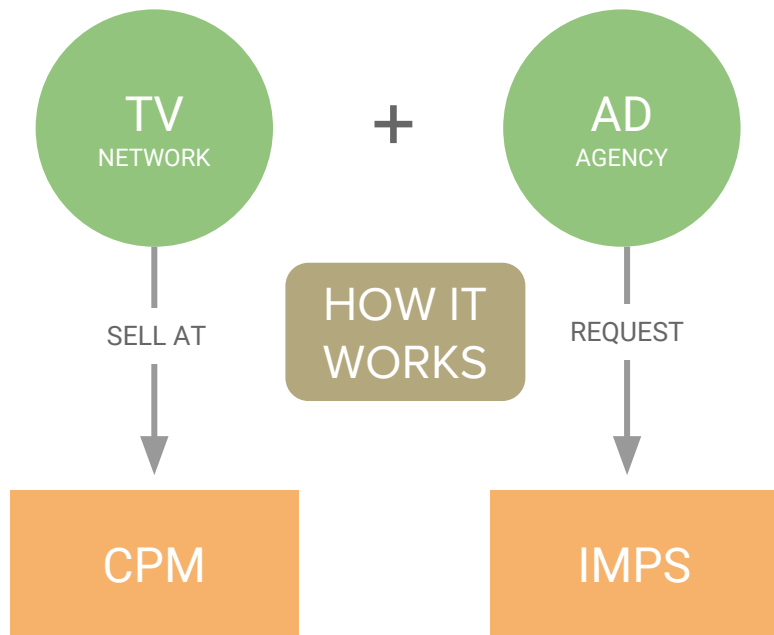
SCATTERED  
DEMAND

vs.

VIEWERSHIP  
GUARANTEES

# BACKGROUND

## KEY PLAYERS



## TV ADVERTISING

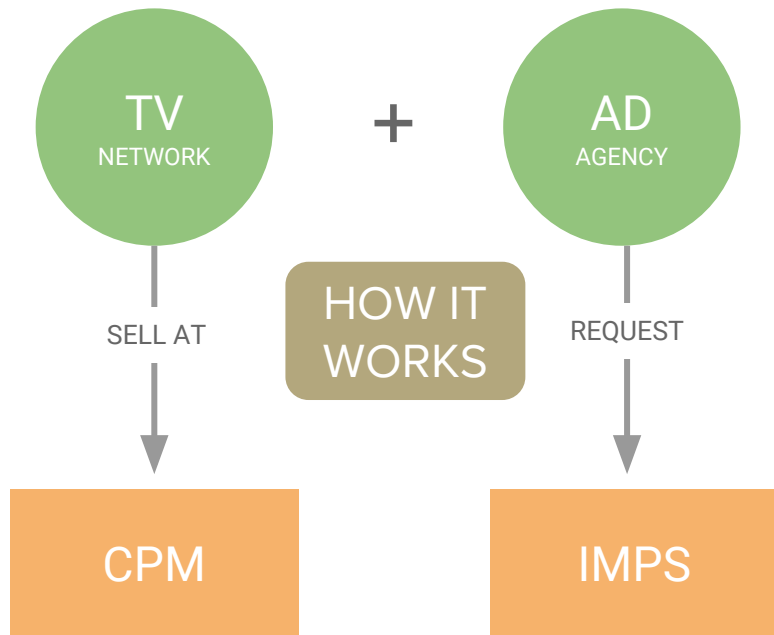
SCATTERED  
DEMAND

vs.

VIEWERSHIP  
GUARANTEES

# BACKGROUND

## KEY PLAYERS



## TV ADVERTISING

SCATTERED  
DEMAND

vs.

VIEWERSHIP  
GUARANTEES

## OUR PROJECT

- CBS
- Prime Time (8 - 11 pm, Weeknights)
- No competitor's ad
- Viewers stay for entire show
- 30 sec per ad
- 2 ad blocks per show
- Deadline to finish assigned impressions
- Underdeliver vs. Overdeliver

# OUR DATA

<b>Show</b>	Name of television show
<b>Week</b>	Indicates which week the show was on; Values 1-4
<b>Day</b>	Day of the week; Monday through Friday
<b>Numbered Day</b>	Enumerated day of the week; Values 1-4
<b>Time</b>	Time slot of television show
<b>Cost to Advertise</b>	Cost for advertiser to advertise on a particular show
<b>Viewership</b>	Total views of a particular show during a particular showtime
<b>CPM</b>	Cost per thousand impressions

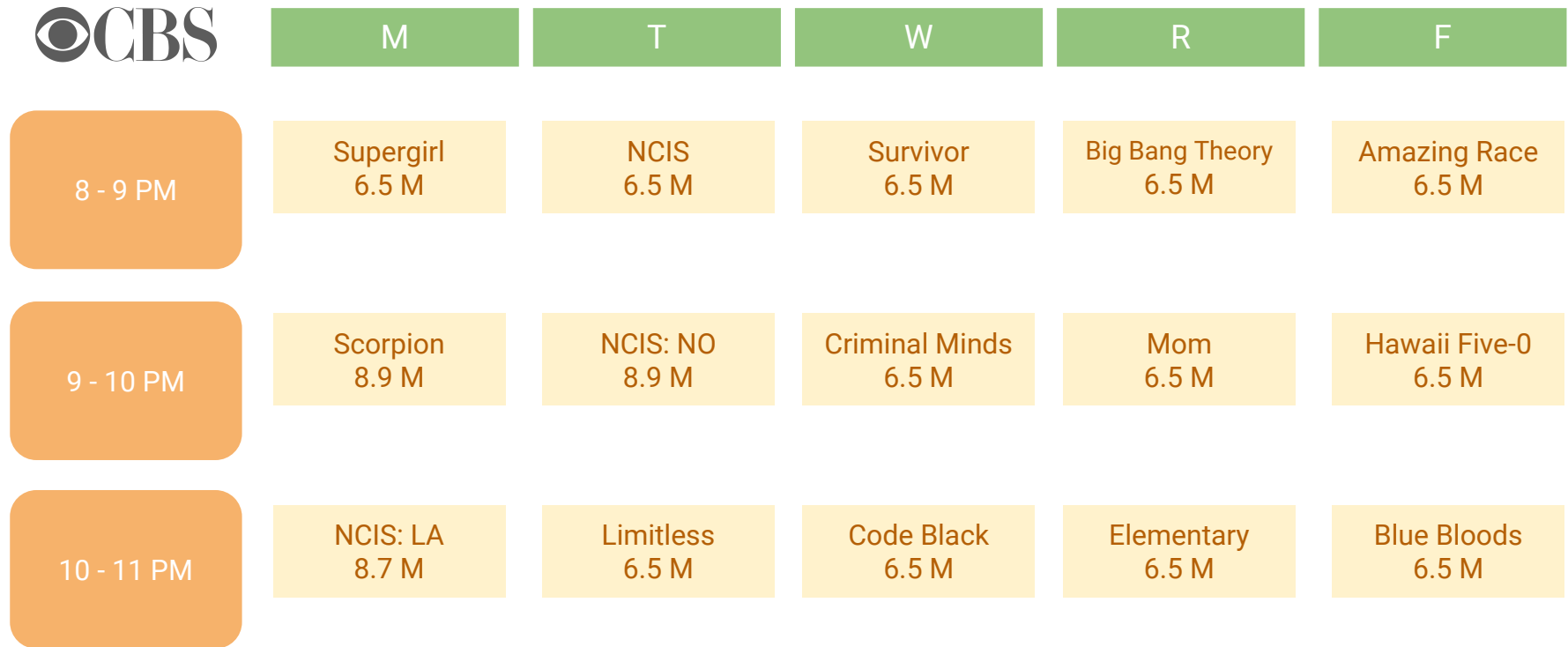
**Table 1.** Viewership Dataset Variables on CBS Shows

<b>Advertiser</b>	Company that is requesting advertising from CBS
<b>Budget per Week</b>	Company's weekly budget for advertising
<b>Imp Requested per Week</b>	Number of impressions/views that a company requests CBS to satisfy per week

**Table 2.** Ad Requests Dataset Variables from TV Advertisers






# OUR DATA



**Figure 1.** Visualization of Viewership Data for Week 1

# OUR DATA

Advertiser	Budget per Week	Imp Requested per Week
 at&t	30.9 M	1.6 M
 GEICO	21.4 M	1.1 M
 NCAA	17.1 M	0.9 M
...		

**Figure 2.** Visualization of Ad Requests Data

# OUR GOAL



	M	T	W	R	F
8 - 9 PM	Supergirl 6.5 M	NCIS 6.5 M	Survivor 6.5 M	Big Bang Theory 6.5 M	Amazing Race 6.5 M
9 - 10 PM	Scorpion 8.9 M	NCIS: NO 8.9 M	Criminal Minds 6.5 M	Mom 6.5 M	Hawaii Five-0 6.5 M
10 - 11 PM	NCIS: LA 8.7 M	Limitless 6.5 M	Code Black 6.5 M	Elementary 6.5 M	Blue Bloods 6.5 M

# OUR GOAL



	M	T	W	R	F
8 - 9 PM	Supergirl 6.5 M	NCIS 6.5 M	Survivor 6.5 M	Big Bang Theory 6.5 M	Amazing Race 6.5 M
9 - 10 PM	Scorpion 8.9 M	NCIS: NO 8.9 M	Criminal Minds 6.5 M	Mom 6.5 M	Hawaii Five-0 6.5 M
10 - 11 PM	NCIS: LA 8.7 M	Limitless 6.5 M	Code Black 6.5 M	Elementary 6.5 M	Blue Bloods 6.5 M

AD BLOCK 1 (indicated by an arrow pointing to the 9-10 PM slot on Monday)

AD BLOCK 2 (indicated by an arrow pointing to a red block in the 8-9 PM slot on Monday)

# OUR GOAL



	M	T	W	R	F
8 - 9 PM	Supergirl 6.5 M	NCIS 6.5 M	Survivor 6.5 M	Big Bang Theory 6.5 M	Amazing Race 6.5 M
9 - 10 PM	Scorpion 8.9 M	NCIS: NO 8.9 M	Criminal Minds 6.5 M	Mom 6.5 M	Hawaii Five-0 6.5 M
10 - 11 PM	NCIS: LA 8.7 M	Limitless 6.5 M	Code Black 6.5 M	Elementary 6.5 M	Blue Bloods 6.5 M

# OUR GOAL



	M	T	W	R	F
8 - 9 PM	Supergirl 6.5 M	NCIS 6.5 M	Survivor 6.5 M	Big Bang Theory 6.5 M	Amazing Race 6.5 M
9 - 10 PM	Scorpion 8.9 M	NCIS: NO 8.9 M	Criminal Minds 6.5 M	Mom 6.5 M	Hawaii Five-0 6.5 M
10 - 11 PM	NCIS: LA 8.7 M	Limitless 6.5 M	Code Black 6.5 M	Elementary 6.5 M	Blue Bloods 6.5 M

ADS



# OUR GOAL



	M	T	W	R	F
8 - 9 PM	Supergirl 6.5 M	NCIS 6.5 M	Survivor 6.5 M	Big Bang Theory 6.5 M	Amazing Race 6.5 M
9 - 10 PM	Scorpion 8.9 M	NCIS: NO 8.9 M	Criminal Minds 6.5 M	Mom 6.5 M	Hawaii Five-0 6.5 M
10 - 11 PM	NCIS: LA 8.7 M	Limitless 6.5 M	Code Black 6.5 M	Elementary 6.5 M	Blue Bloods 6.5 M
ADS					

# OUR GOAL



	M	T	W	R	F
8 - 9 PM	Supergirl 6.5 M	NCIS 6.5 M	Survivor 6.5 M	Big Bang Theory 6.5 M	Amazing Race 6.5 M
9 - 10 PM	Scorpion 8.9 M	NCIS: NO 8.9 M	Criminal Minds 6.5 M	Mom 6.5 M	Hawaii Five-0 6.5 M
10 - 11 PM	NCIS: LA 8.7 M	Limitless 6.5 M	Code Black 6.5 M	Elementary 6.5 M	Blue Bloods 6.5 M

ADS



# OUR GOAL



	M	T	W	R	F
8 - 9 PM	Supergirl 6.5 M	NCIS 6.5 M	Survivor 6.5 M	Big Bang Theory 6.5 M	Amazing Race 6.5 M
9 - 10 PM	Scorpion 8.9 M	NCIS: NO 8.9 M	Criminal Minds 6.5 M	Mom 6.5 M	Hawaii Five-0 6.5 M
10 - 11 PM	NCIS: LA 8.7 M	Limitless 6.5 M	Code Black 6.5 M	Elementary 6.5 M	Blue Bloods 6.5 M

ADS



# OUR GOAL



	M	T	W	R	F
8 - 9 PM	Supergirl 6.5 M	NCIS 6.5 M	Survivor 6.5 M	Big Bang Theory 6.5 M	Amazing Race 6.5 M
9 - 10 PM	Scorpion 8.9 M	NCIS: NO 8.9 M	Criminal Minds 6.5 M	Mom 6.5 M	Hawaii Five-0 6.5 M
10 - 11 PM	NCIS: LA 8.7 M	Limitless 6.5 M	Code Black 6.5 M	Elementary 6.5 M	Blue Bloods 6.5 M

ADS



# OUR GOAL



	M	T	W	R	F
8 - 9 PM	Supergirl 6.5 M	NCIS 6.5 M	Survivor 6.5 M	Big Bang Theory 6.5 M	Amazing Race 6.5 M
9 - 10 PM	Scorpion 8.9 M	NCIS: NO 8.9 M	Criminal Minds 6.5 M	Mom 6.5 M	Hawaii Five-0 6.5 M
10 - 11 PM	NCIS: LA 8.7 M	Limitless 6.5 M	Code Black 6.5 M	Elementary 6.5 M	Blue Bloods 6.5 M



ADS

# OUR GOAL



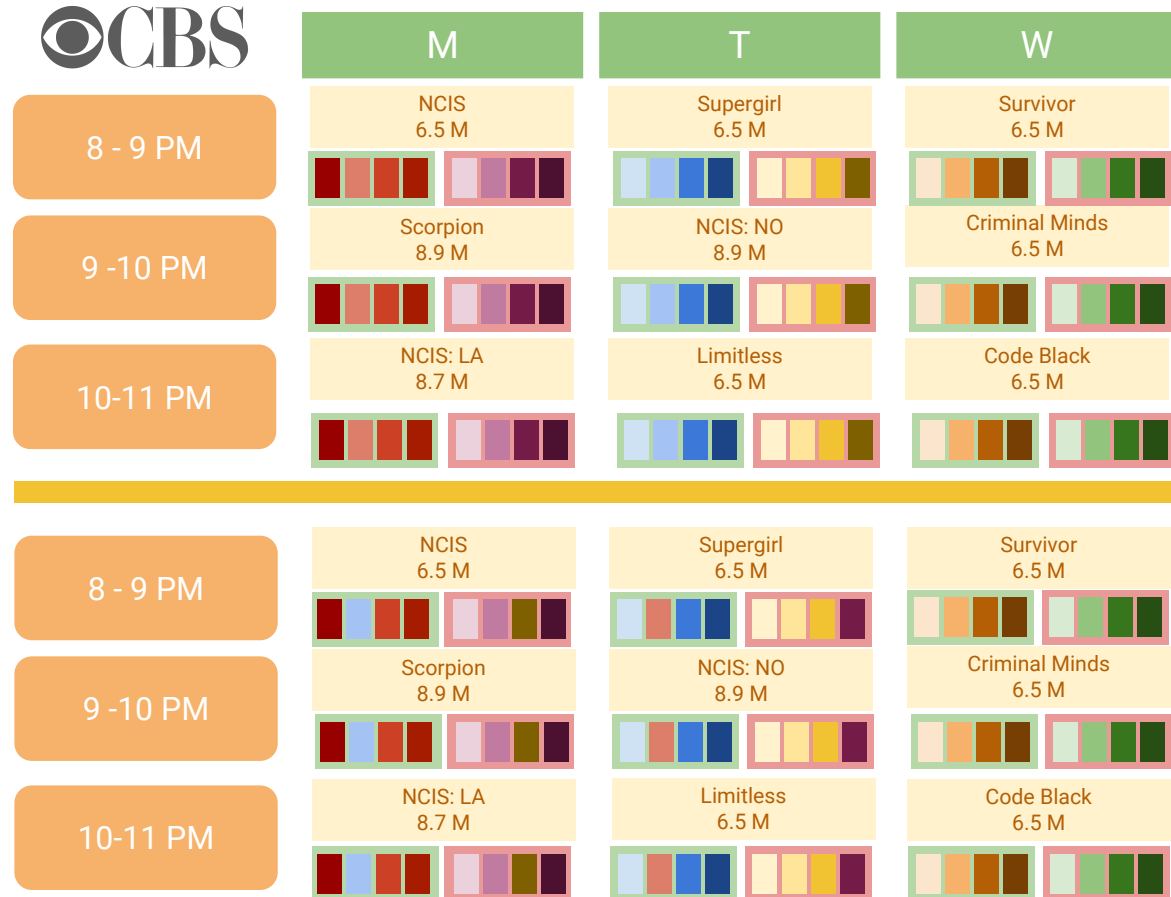
	M	T	W	R	F
8 - 9 PM	Supergirl 6.5 M 	NCIS 6.5 M 	Survivor 6.5 M 	Big Bang Theory 6.5 M 	Amazing Race 6.5 M 
9 - 10 PM	Scorpion 8.9 M 	NCIS: NO 8.9 M 	Criminal Minds 6.5 M 	Mom 6.5 M 	Hawaii Five-0 6.5 M 
10 - 11 PM	NCIS: LA 8.7 M 	Limitless 6.5 M 	Code Black 6.5 M 	Elementary 6.5 M 	Blue Bloods 6.5 M 

ADS

# FORMULATION

$Q \mid \text{pmpt} \mid \sum T_j$

- Jobs: advertising requests
- $p_j$ : required viewers
- $Q$ : shows are related machines
- $\text{pmpt}$ : changing show after a week
- $\sum T_j$
- Must fulfill requests



# EXPLORING 3 ALGORITHMS

---

- 1 Dynamic Programming
- 2 Brand & Bound
- 3 LRPT-FM

# EXPLORING 3 ALGORITHMS

---

1

Dynamic Programming



2

Brand & Bound

3

LRPT-FM

# BRAND & BOUND

---

- **Cost:** No preference
- **Revenue:** Min CPM
- **Capacity**
- **Feasible solution:** CPM vs Visibility
- Not too relevant in this case

# BRAND & BOUND

	Advertiser	Budget per Week	Imp Requested per Week	Free Views
1	AT&T	30,940,000	1,628,421,053	-158,947
2	Geico	21,490,000	1,131,052,632	-9,277,368
3	NCAA	17,180,000	904,210,526	-4,699,474
4	Capital One	14,290,000	752,105,263	-2,424,737
5	Subaru	12,960,000	682,105,263	-2,044,737
6	Audi	12,930,000	680,526,316	-2,063,684
7	State Farm	12,690,000	667,894,737	-5,440,263
8	Apple iPhone	12,670,000	666,842,105	-1,392,895
9	Taco Bell	11,910,000	626,842,105	-1,672,895
10	Samsung Mobile	11,680,000	614,736,842	-518,158
11	Buick	12,503,470	658,077,368	-3,227,632
12	Universal Pictures	12,462,734	655,933,368	-2,581,632
13	T-Mobile	12,394,212	652,326,947	-428,053
14	Ford	11,557,984	608,314,947	-1,810,053
15	Nationwide Insurance	10,781,073	567,424,895	-6,205,105
16	Verizon	10,386,960	546,682,105	-3,082,895
17	McDonald's	10,569,615	556,295,526	-4,089,474
18	Warner Bros	10,322,320	543,280,000	-1,475,000
19	Procter & Gamble	12,602,739	663,302,053	-297,947
20	General Motors	8,493,151	447,007,947	-202,053
21	Comcast	8,219,178	432,588,316	-1,076,684
22	American Express	6,575,342	346,070,632	-1,764,368
23	L'Oréal	6,027,397	317,231,421	-7,338,579
24	Walt Disney	5,753,425	302,811,842	-63,788,158
25	Fiat Chrysler	6,027,397	317,231,421	-80,763,579
				SUM
				-207,824,368

**Table 3.** Results of Scheduling Problem using Branch & Bound

# FEASIBLE SOLUTION

Numbered Day	Time	Cost to Advertise	Viewership	CPM	1	2	3	4	5	6	7	8
1	08:00 PM - 09:00 PM	147,933	6,530,000	22.65436447	5	10	13	20	23	10	24	6
1	09:00 PM - 10:00 PM	142,108	8,955,000	15.86912339	6	16	17	19	14	24	24	16
1	10:00 PM - 11:00 PM	109,940	8,781,000	12.5202141	7	8	20	21	22	23	25	22
2	08:00 PM - 09:00 PM	151,738	17,260,000	8.791309386	1	2	3	1	2	4	5	1
2	09:00 PM - 10:00 PM	125,920	11,967,000	10.52226957	3	6	7	8	3	10	7	6
2	10:00 PM - 11:00 PM	113,900	5,627,000	20.24169184	8	15	18	20	21	22	14	21
3	8:00 PM - 9:00 PM	125,449	9,700,000	12.9328866	3	9	11	12	3	11	12	9
3	9:00 PM - 10:00 PM	133,983	9,300,000	14.40677419	9	11	13	14	15	13	11	13
3	10:00 PM - 11:00 PM	129,626	7,114,000	18.22125387	9	10	14	15	17	18	19	15
4	08:00 PM - 09:00 PM	348,300	15,290,000	22.77959451	1	2	4	1	2	4	5	1
4	09:00 PM - 10:00 PM	144,660	9,087,000	15.91944536	8	12	13	8	14	12	15	14
4	10:00 PM - 11:00 PM	106,695	6,327,000	16.86344239	12	16	17	18	19	23	16	18
5	08:00 PM - 09:00 PM	65,517	6,087,000	10.76343026	6	7	9	11	14	16	17	7
5	09:00 PM - 10:00 PM	77,683	8,876,000	8.752027941	4	10	17	20	21	25	25	4
5	10:00 PM - 11:00 PM	75,965	10,922,000	6.95522798	19	1	2	18	19			

**Table 5.** Feasible Solution for Branch & Bound Algorithm

# LONGEST & SHORTEST PROCESSING TIMES

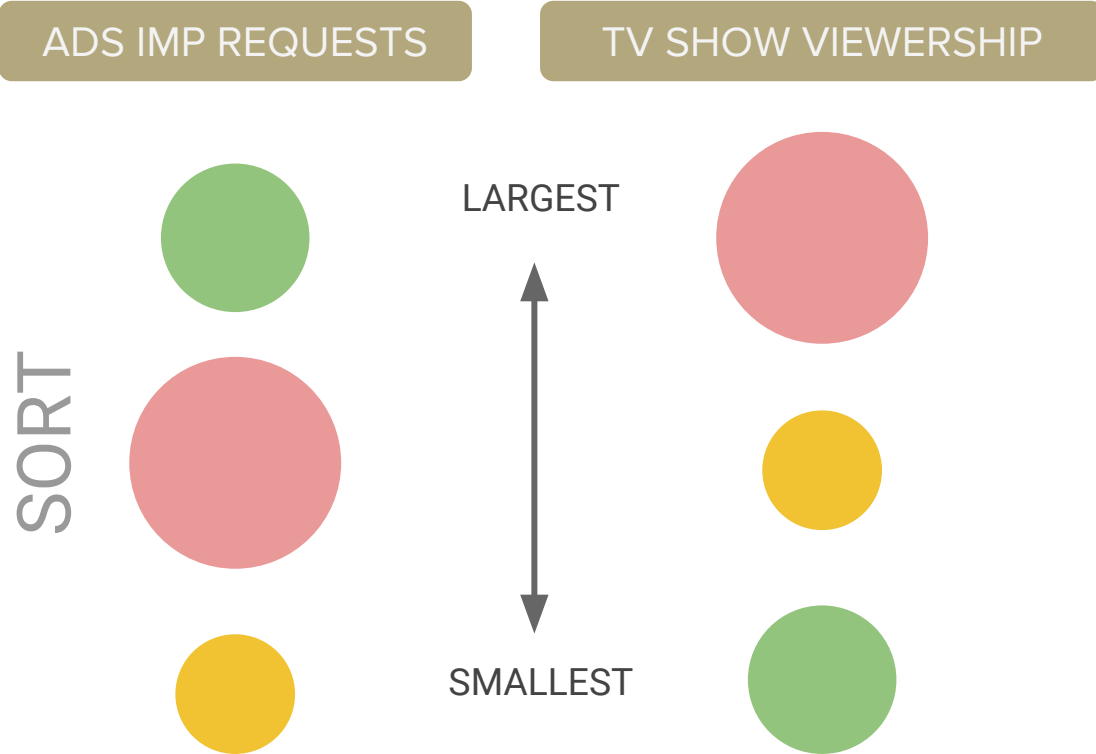
## Longest Remaining Processing Times on Fastest Machine

- Ads with the most impressions go on the most popular shows
- Helps ensure that advertisers get their guaranteed views
- If not enough ad slots, then it minimizes the number of missed views

## Shortest Remaining Processing Times on Fastest Machine

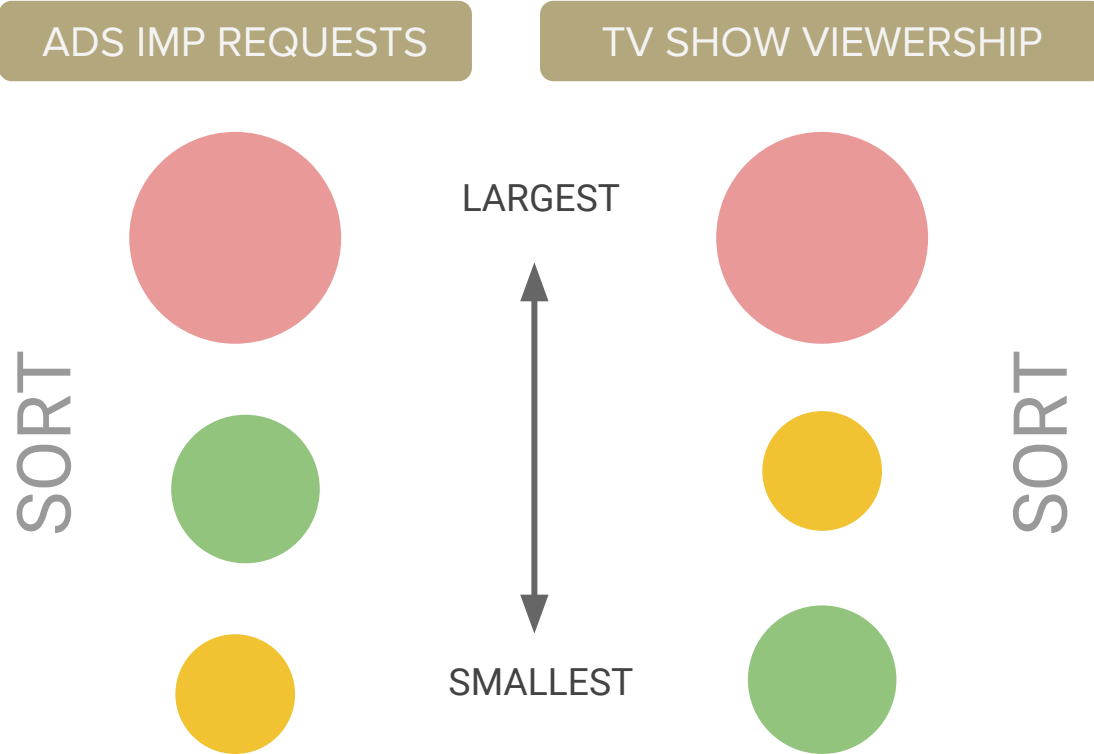
- Ads with the least impressions go on the most popular shows
- Best for  $Q \mid \text{pmtn} \mid \sum C_j$

# LONGEST & SHORTEST PROCESSING TIMES



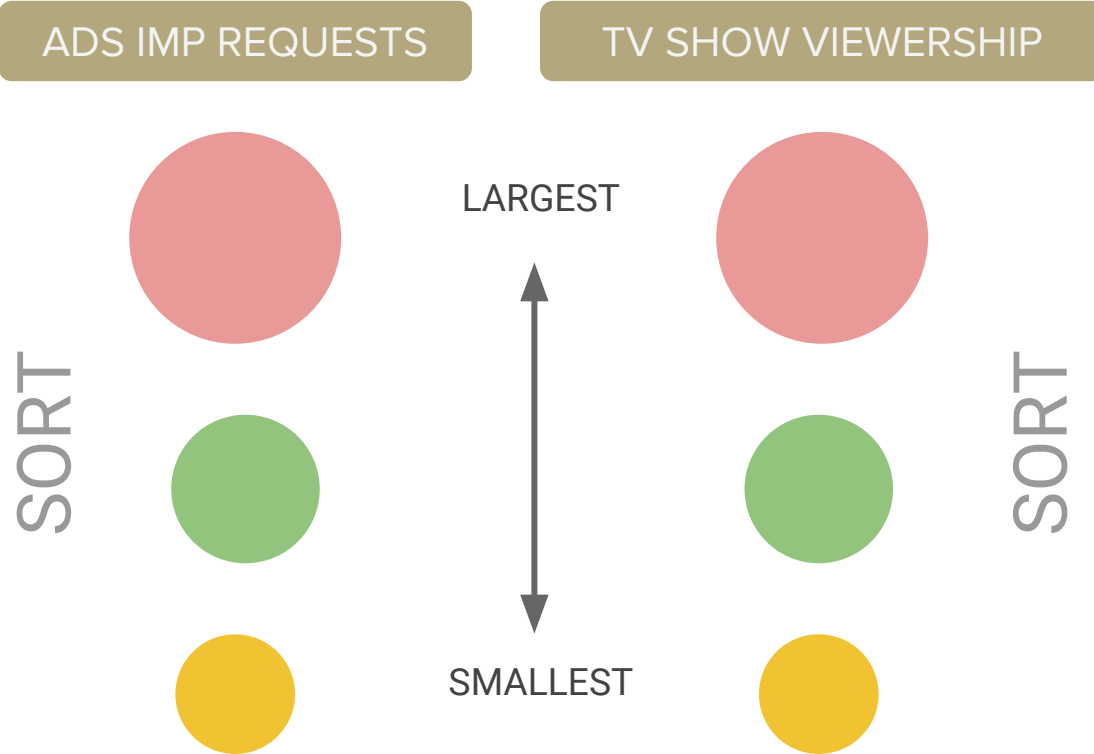
Sort in  
decreasing  
order

# LONGEST & SHORTEST PROCESSING TIMES



Sort in decreasing order

# LONGEST & SHORTEST PROCESSING TIMES



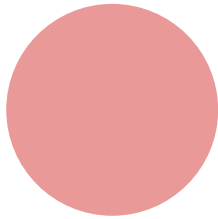
Sort in  
decreasing  
order

# LONGEST & SHORTEST PROCESSING TIMES

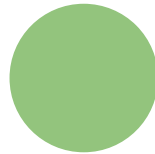
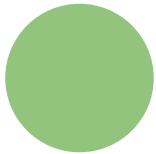
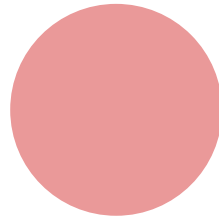
ADS IMP REQUESTS

TV SHOW VIEWERSHIP

REMAINING IMPS



ASSIGN



XX

XX



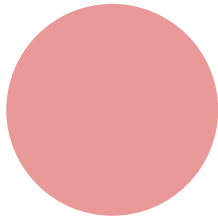
XX

# LONGEST & SHORTEST PROCESSING TIMES

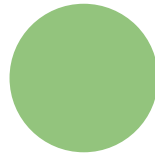
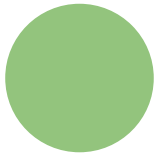
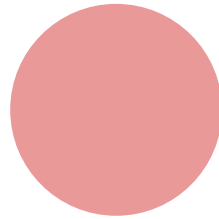
ADS IMP REQUESTS

TV SHOW VIEWERSHIP

TARDINESS



ASSIGN



YY

YY



YY

# RESULTS

Week	# Ads needed	Free Impressions	Percent of Total Impressions
1	112	149,000,000	9.3%
2	126	105,000,000	6.6%
3	115	147,000,000	9.2%
4	111	93,200,000	5.8%

**Table 6.** Results of Scheduling Problem using LRPT-FM

# CONCLUSION

---

- Problem is iterative
  - Multiple assignment process to create a viable schedule
  - Dynamic programming theoretically works but the specific conditions of the case don't hold
- Branch and Bound helps maximize revenue
  - Identifies potential schedules with higher CPM
  - Creates a schedule favoring higher price per unit ads
- LRPT-FM
  - Has minimum number of free views while actively fulfilling ad contracts

# NEXT STEPS

J Sched (2012) 15:193–200  
DOI 10.1007/s10951-010-0198-5

## Minimizing total tardiness on parallel machines with preemptions

Svetlana A. Kravchenko · Frank Werner

Published online: 16 September 2010  
© Springer Science+Business Media, LLC 2010

**Abstract** The basic scheduling problem we are dealing with in this paper is the following one. A set of jobs has to be scheduled on a set of parallel uniform machines. Each machine can handle at most one job at a time. Each job becomes available for processing at its release date. All jobs have the same execution requirement and arbitrary due dates. Each machine has a known speed. The processing of any job may be interrupted arbitrarily often and resumed later on any machine. The goal is to find a schedule that minimizes the sum of tardiness, i.e., we consider problem  $Q | r_j, p_j = p, \text{pntn} | \sum T_j$  whose complexity status was open. Recently, Tian et al. (J. Sched. 9:343–364, 2006) proposed a polynomial algorithm for problem  $1 | r_j, p_j = p, \text{pntn} | \sum T_j$ . We show that both the problem  $P | \text{pntn} | \sum T_j$  of minimizing total tardiness on a set of parallel machines with allowed preemptions and the problem  $P | r_j, p_j = p, \text{pntn} | \sum T_j$  of minimizing total tardiness on a set of parallel machines with release dates, equal processing times and allowed preemptions are NP-hard. Moreover, we give a polynomial algorithm for the case of uniform machines without release dates, i.e., for problem  $Q | p_j = p, \text{pntn} | \sum T_j$ .

**Keywords** Parallel machines · Total tardiness · Preemptive problems · Linear programming · Polynomial algorithm · NP-hardness

S.A. Kravchenko  
United Institute of Informatics Problems, Surganova St. 6,  
220012 Minsk, Belarus

F. Werner (✉)  
Fakultät für Mathematik, Otto-von-Guericke-Universität,  
39106 Magdeburg, Germany  
e-mail: frank.werner@mathematik.uni-magdeburg.de

### 1 Introduction

The problem considered can be stated as follows. There are  $n$  independent jobs and  $m$  parallel uniform machines. For each job  $J_j$ ,  $j = 1, \dots, n$ , there is given its processing time  $p_j = p$ , its release date  $r_j \geq 0$ , and its due date  $d_j \geq 0$ . Each machine  $M_q$ ,  $q = 1, \dots, m$ , has some speed  $s_q$ , i.e., the execution of job  $J_j$  on machine  $M_q$  requires  $p/s_q$  time units. Any machine can process any job but only one job at a time. Furthermore, a job can be processed only on one machine at a time. Preemptions of processing are allowed, i.e., the processing of any job may be interrupted at any time and resumed later, possibly on a different machine. We assume that all numerical data are integers. For a schedule  $s$ , let  $T_j(s) = \max\{0, C_j(s) - d_j\}$  denote the tardiness of job  $J_j$  in  $s$ , where  $C_j(s)$  is the time at which the processing of job  $J_j$  is completed. If no ambiguity arises, we drop the reference to schedule  $s$  and write  $T_j$  and  $C_j$ . The problem is to schedule all jobs so as to minimize the optimality criterion  $\sum_{j=1}^n T_j$ . The described problem can be denoted as  $Q | r_j, p_j = p, \text{pntn} | \sum T_j$ .

Note that problem  $P | r_j, \text{pntn} | \sum C_j$  and, therefore, problem  $P | r_j, \text{pntn} | \sum T_j$  are unary NP-hard (Baptiste et al. 2007). Recall that  $P$  in the notation of the problem means that all machines have identical speeds.

Problem  $1 | r_j, p_j = p, \text{pntn} | \sum T_j$  can be solved in  $O(n^2)$  time (Tian et al. 2006). In Du and Leung (1990), it has been shown that problems  $1 | \text{pntn} | \sum T_j$  and  $1 | \sum T_j$  are NP-hard in the ordinary sense whereas in Lawler (1977), a pseudopolynomial algorithm has been proposed for these problems. As it was noted in Du and Leung (1990), preemptions cannot reduce the total tardiness on one machine but for parallel machines, it is easy to see that preemptions can reduce the total tardiness.

In this paper, it is shown that both problems  $P | \text{pntn} | \sum T_j$  and  $P | r_j, p_j = p, \text{pntn} | \sum T_j$  are NP-hard under

### Minimize

$$\sum_{k=1}^n \sum_{i=k}^n (C_k^i - d_i) \quad (2.2)$$

subject to

$$d_i = C_0^i \leq C_1^i \leq \dots \leq C_n^i \leq C_{n+1}^i = d_{i+1},$$

$$i = 0, \dots, n, \quad (2.3)$$

$$\sum_{q=1}^m \frac{v_j^q([C_k^i, C_{k+1}^i])}{s_q} \leq C_{k+1}^i - C_k^i,$$

$$i = 1, \dots, n, \quad j = 1, \dots, n, \quad k = 1, \dots, n, \quad (2.4)$$

$$\sum_{j=1}^n \frac{v_j^q([C_k^i, C_{k+1}^i])}{s_q} \leq C_{k+1}^i - C_k^i,$$

$$i = 1, \dots, n, \quad q = 1, \dots, m, \quad k = 1, \dots, n, \quad (2.5)$$

$$\sum_{i=1}^n \sum_{k=1}^n \sum_{q=1}^m v_j^q([C_k^i, C_{k+1}^i]) = p, \quad j = 1, \dots, n, \quad (2.6)$$

$$v_j^q([C_k^i, C_{k+1}^i]) = 0, \quad i = 0, \dots, n, \quad j = 1, \dots, n,$$

$$q = 1, \dots, m, \quad k = j, \dots, n, \quad (2.7)$$

$$C_k^i \geq 0, \quad i = 0, \dots, n, \quad k = 0, \dots, n+1, \quad (2.8)$$

$$v_j^q([C_k^i, C_{k+1}^i]) \geq 0, \quad i = 0, \dots, n, \quad j = 1, \dots, n,$$

$$k = 0, \dots, n, \quad q = 1, \dots, m. \quad (2.9)$$



# THANK YOU

PRODUCTION SCHEDULING, APRIL 26, 2016 | ANDELYN RUSSELL KARA ODUM CHLOE SHIH HONGLI YANG

