CTW 2020

Coupling Machine Learning and Integer Programming for Optimal TV Promo Scheduling

Joint work with: **R. Seccia**[1],
G. Leo[2],
M. Vahdat[2],
Q. Gao[2],
H. Wali[3]

1: Department of Computer, Control, and Management Engineering "Antonio Ruberti"
2: IBM Data Science Elite Team, Germany,
3: IBM Data Science Center of Excellence, Saudi Arabia

SAPIENZA
UNIVERSITÀ DI ROMA

Table of content

Available airtime allocation

TV Media companies divide the available airtime into several TV programs, separated by breaks.

Program | *Break* | *Program* | *Break* | *Program* | *Break* → *Airtime*

Available airtime allocation

TV Media companies divide the available airtime into several TV programs, separated by breaks.

Within breaks we have:

Commercials:

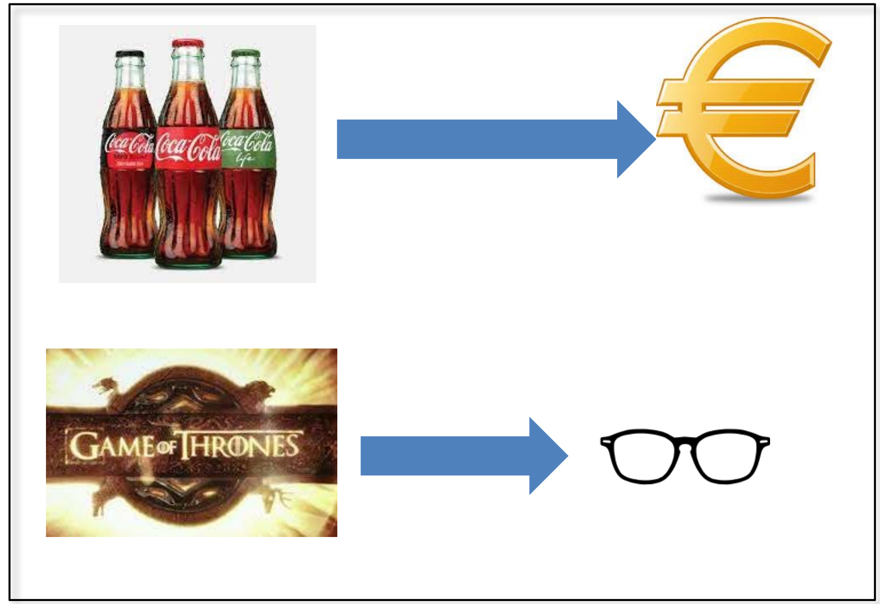
- Promote products or services offered by third-party companies (e.g. Coca-Cola)
- Affect company's **income**

Promos:

- Advertise TV programs (e.g. "The Voice") or other products offered by the media group itself (e.g. "Rai Replay")
- Affect channel/program's **viewership**

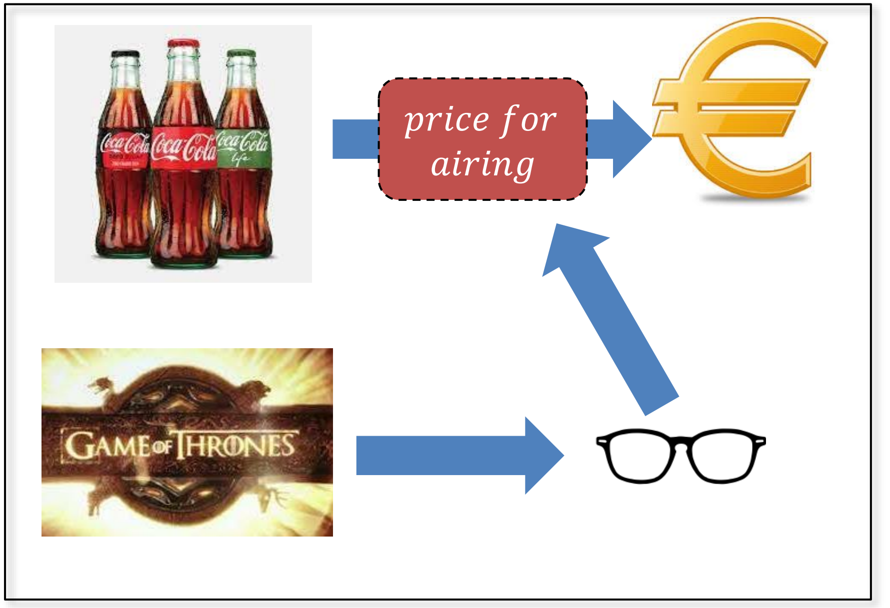
What is Optimal TV Promo Scheduling (OPS)

- Commercials directly affect company's income
- Promos affect the programs/channels' viewership



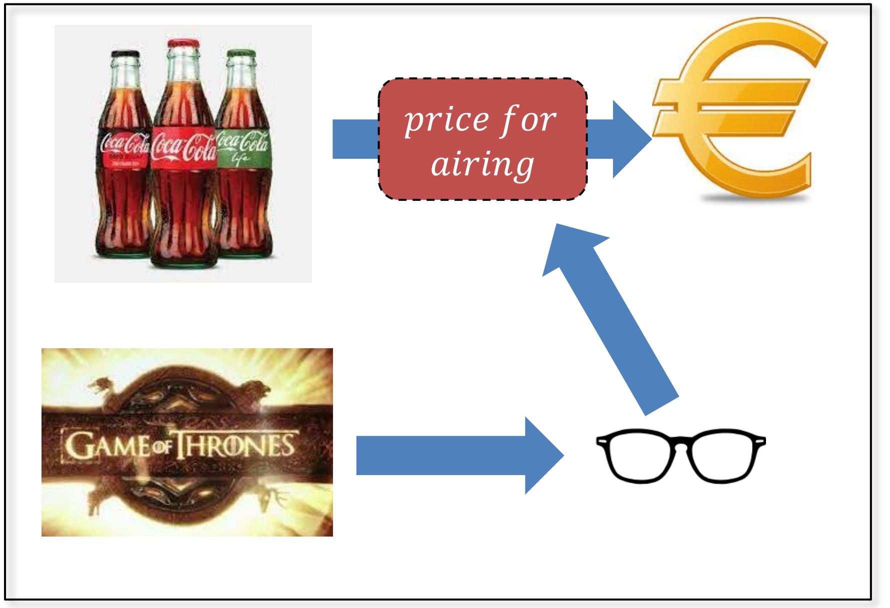
What is Optimal TV Promo Scheduling (OPS)

- Commercials directly affect company's income
- Promos affect the programs/channels' viewership
- Increasing the viewership increases the price for selling airtime to commercials
- Increasing the price for airtime increases the company's income



What is Optimal TV Promo Scheduling (OPS)

- Commercials directly affect company's income
- Promos affect the programs/channels' viewership
- Increasing the viewership increases the price for selling airtime to commercials
- Increasing the price for airtime increases the company's income



Objective in OPS:

Find the optimal scheduling of promos within breaks in order to maximize programs/channels' viewership (i.e. revenue)

What is Optimal TV Promo Scheduling (OPS)

Given a set of breaks $\{B_1, B_2, B_3, \ldots\}$

We need to assign them a set of promos $\{P_1, P_2, P_3, P_4, P_5, P_6, P_7\}$, i.e.

$$B_1 \rightarrow \{P_1, P_3, P_6\}$$
$$B_2 \rightarrow \{P_2, P_3, P_5\}$$
$$B_3 \rightarrow \{P_1, P_4\}$$

In order to:

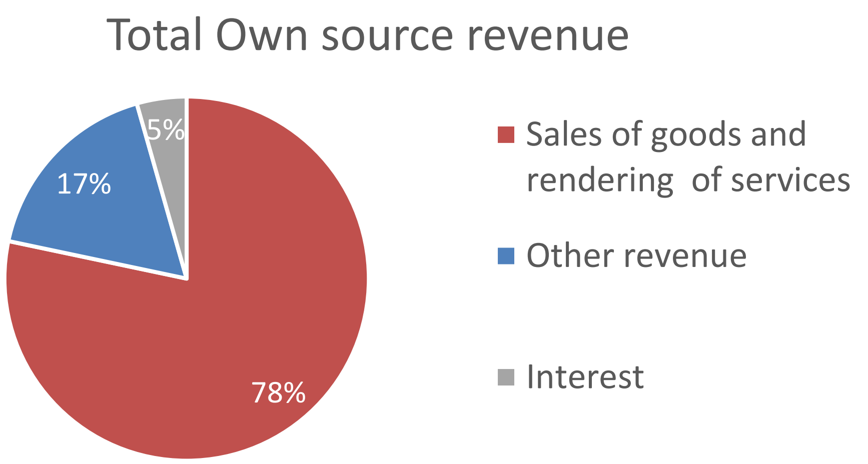
1. Maximize the (**expected**) viewership generated by placing a promo in a specific break
2. Satisfy all the logical and business constraints

What is Optimal TV Promo Scheduling (OPS)

Intuition on the magnitude of the market

An Example:

Australian TV Media company **ABC**



The Voice paid $ 150,000 for 30-second TV spot on 2013 (for the last episode)

$$\frac{\$150{,}000}{\$124{,}106{,}000} = \mathbf{0.12}\%$$

Only **one spot** made up 0.12% of the main source of revenue for ABC

- https://about.abc.net.au/wp-content/uploads/2014/12/ABCAnnualReport2014Accessible.pdf
- https://www.news.com.au/entertainment/tv/k-for-30-secs-you8217ve-got-to-be-kidding-the-most-expensive-tv-shows-for-advertisers-in-australia-revealed/news-story/dd39a4a74c83dfbcd1c020f1b9a1f2e8

Challenges in OPS problem

Three main challenges:

1. **Estimating** the impact of placing a promo within a specific break requires the definition of **accurate forecasting models**
2. The **uncertainty** of the setting requires the definition of **fast-to-solve formulations** (unexpected events may result in sudden changes in the overall airing schedule)
3. **Many business rules** must be taken into consideration while scheduling promos. It makes the **optimization problem hard** to tackle

State of the art

Most of the literature has been focused on solving the scheduling problem for airing commercials.

However it mostly differs from scheduling for promos (e.g. different objectives, different business constraints, different prediction task)

For scheduling promos very few studies have been done and only **heuristic methods** have been developed (see e.g. [1,2] where genetic algorithms are implemented)

This is the first attempt to solve it through an **exact approach**

Currently, OPS is mostly **done manually** by business experts within media companies

[1] Fontes, D. B. M. M., Paulo A. Pereira, and F. A. C. C. Fontes. "A Decision Support System for TV self-promotion Scheduling." International Journal of Advanced Trends in Computer Science and Engineering 8.2 (2019): 134-140.
[2] Pereira, Paulo A., Fernando ACC Fontes, and Dalila BMM Fontes. "A Genetic Algorithm Approach for the TV Self-Promotion Assignment Problem." AIP Conference Proceedings. Vol. 1168. No. 1. American Institute of Physics, 2009.

How we tackle the OPS problem

Setting

- Scheduling day by day (because of business constraints)
- Focus on one channel (but easy to extend to multichannel setting)
- Commercials already assigned to timeslots

We define two main frameworks:

Machine Learning framework:

Estimates the viewership resulted from placing a promo in a specific break

Optimization Framework:

Finds the assignments Promos-Break that maximize the overall viewership while satisfying all the business rules

How we tackle the OPS problem

Machine Learning framework

Reach used to measure the viewership of a promo/program:

$$Reach \ = \ \% \ of \ unique \ households \ reached \ by \ the \ aired \ content$$

Promos for Programs:	Promote TV programs (e.g. The Voice)	Focus on the Client Program Reach (**CR**)

How we tackle the OPS problem

Machine Learning framework

Reach used to measure the viewership of a promo/program:

$$Reach \; = \; \% \; of \; unique \; households \; reached \; by \; the \; aired \; content$$

Promos for Programs:	Promote TV programs (e.g. The Voice)	Focus on the Client Program Reach (**CR**)
Non-channel promos:	Promote non-channel media group's products (e.g. Rai Play)	Focus on the Promo Reach (**PR**)

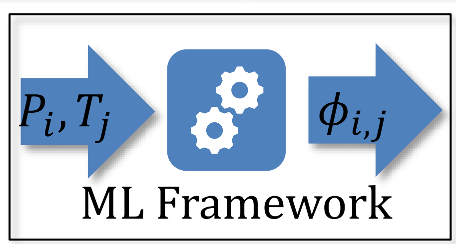
How we tackle the OPS problem

Machine Learning framework

Reach used to measure the viewership of a promo/program:

$$Reach = \%\ of\ unique\ households\ reached\ by\ the\ aired\ content$$

Promos for Programs:	Promote TV programs (e.g. The Voice)	Focus on the Client Program Reach (**CR**)
Non-channel promos:	Promote non-channel media group's products (e.g. Rai Play)	Focus on the Promo Reach (**PR**)



How we tackle the OPS problem

How to combine ML with Optimization

Two opposite needs:

- We need to solve a **combinatorial** optimization problem given a **nonlinear objective** function (it derives from the ML model)
- We need to solve it **fast** (unexpected events may implies to drastically change the schedule in few seconds)

How we tackle the OPS problem

How to combine ML with Optimization

Two opposite needs:

- We need to solve a **combinatorial** optimization problem given a **nonlinear objective** function (it derives from the ML model)
- We need to solve it **fast** (unexpected events may implies to drastically change the schedule in few seconds)

Solution:

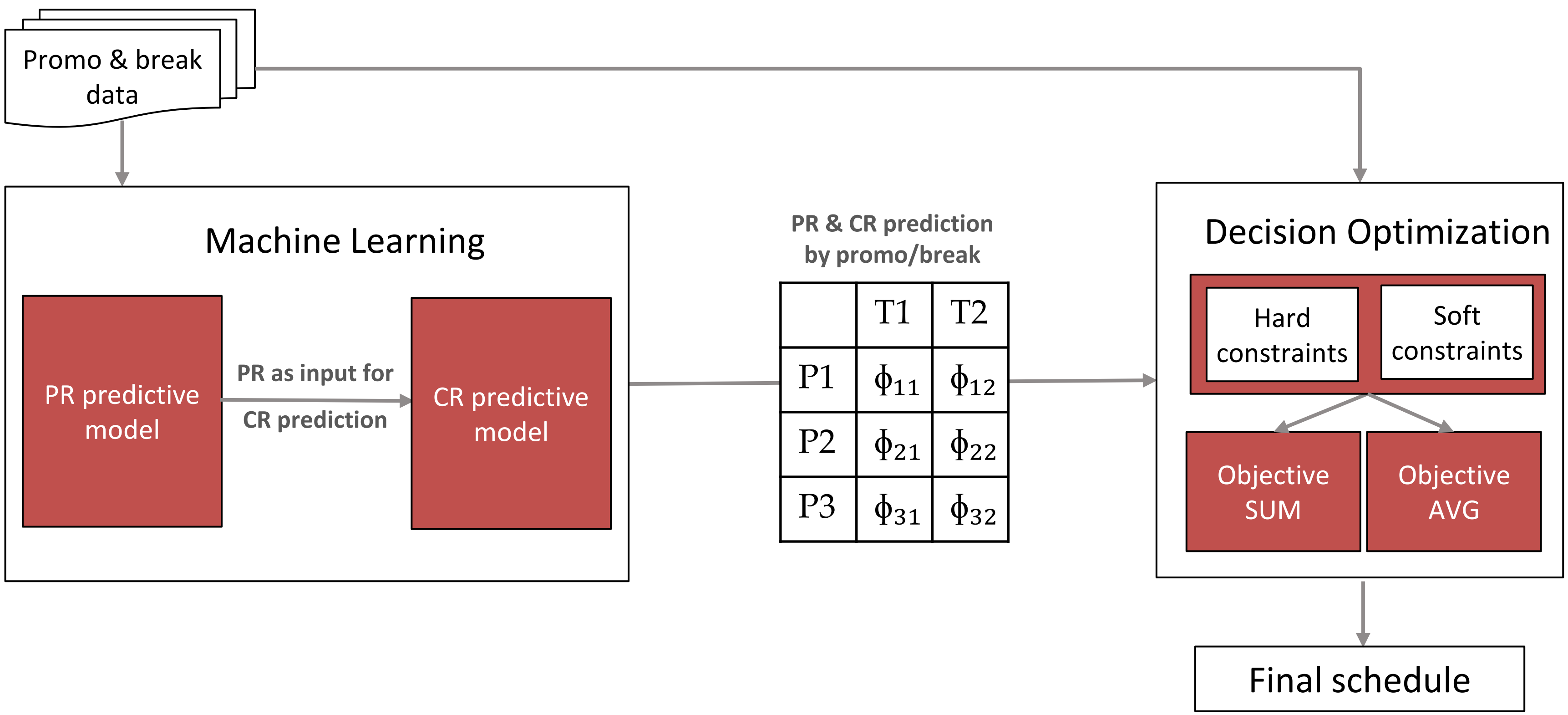
Compute the outcome of all the potential combinations Promo- breaks and formulate the problem as a **linear** one

	T_1	T_2
P_1	ϕ_{11}	ϕ_{12}
P_2	ϕ_{21}	ϕ_{22}
P_3	ϕ_{31}	ϕ_{32}

$\boldsymbol{P_i}$: i –th Promo
$\boldsymbol{T_j}$: j –th Break
$\boldsymbol{\phi_{ij}}$: viewership when placing promo i in break j

How we tackle the OPS problem

The whole picture



Optimization framework

Variable definition:

$$\delta_{pt} = \begin{cases} 1 & \text{if promo } p \text{ is assigned to break } t \\ 0 & \text{otherwise} \end{cases}$$

Constraints:

- ***Hard constraints:***
  Constraints that cannot be violated
  Implemented as mathematical constraints
  (e.g. cannot air trailer of Horror movies during programs for kids)

- ***Soft constraints :***
  Constraints that should be preferably satisfied but that are not strict requirements
  Implemented as penalization terms
  (i.e. ratio of self promos)

It finds the **match promos-break**. Then in the post-processing we determine the order within each break.

Terminology

Break	**Timeslot** within which many promos and commercials can be aired
Promos for programs	Promote **TV programs** (e.g. “The Voice”);
Non-channel promos	Promote **non-channel** media group's **products** (e.g. “Rai Replay”);

Terminology

Break	**Timeslot** within which many promos and commercials can be aired
Promos for programs **Non-channel promos**	Promote **TV programs** (e.g. “The Voice”); Promote **non-channel** media group's **products** (e.g. “Rai Replay”);
Specific promos **Generic promos**	Advertise a **specific episode** of a program (“The Voice Ep2”) Advertise the **program in general** (e.g. “The Voice”)

Sapienza Università di Roma

Terminology

Break	**Timeslot** within which many promos and commercials can be aired
Promos for programs **Non-channel promos**	Promote **TV programs** (e.g. “The Voice”); Promote **non-channel** media group's **products** (e.g. “Rai Replay”);
Specific promos **Generic promos**	Advertise a **specific episode** of a program (“The Voice Ep2”) Advertise the **program in general** (e.g. “The Voice”)
Self-promos **Cross-promos**	Promos promoting content aired on the **same channel** where the promo is aired on. Promos promoting content aired on **other channels** where the promo is aired on.

Terminology

Break	**Timeslot** within which many promos and commercials can be aired
Promos for programs **Non-channel promos**	Promote **TV programs** (e.g. “The Voice”); Promote **non-channel** media group's **products** (e.g. “Rai Replay”);
Specific promos **Generic promos**	Advertise a **specific episode** of a program (“The Voice Ep2”) Advertise the **program in general** (e.g. “The Voice”)
Self-promos **Cross-promos**	Promos promoting content aired on the **same channel** where the promo is aired on. Promos promoting content aired on **other channels** where the promo is aired on.
Host program **Client program**	The program which **hosts the aired promo**; The **program promoted** by the promo;

Terminology

Break	**Timeslot** within which many promos and commercials can be aired
Promos for programs **Non-channel promos**	Promote **TV programs** (e.g. “The Voice”); Promote **non-channel** media group's **products** (e.g. “Rai Replay”);
Specific promos **Generic promos**	Advertise a **specific episode** of a program (“The Voice Ep2”) Advertise the **program in general** (e.g. “The Voice”)
Self-promos **Cross-promos**	Promos promoting content aired on the **same channel** where the promo is aired on. Promos promoting content aired on **other channels** where the promo is aired on.
Host program **Client program**	The program which **hosts the aired promo**; The **program promoted** by the promo;
Others	Each promo might belong to a **campaign** The **genre** of a promo is the genre of the product promoted

Optimization framework

Parameters definition

P	set of promos
T	set of breaks
Q	set of client programs
K	set of genres

Optimization framework

Parameters definition

P	set of promos
T	set of breaks
Q	set of client programs
K	set of genres
$P(c) \subseteq P$	subset of promos for campaign $c \in C$
$P(q) \subseteq P$	subset of promos for client program $c \in C$
$P(k) \subseteq P$	subset of promos for genre $k \in K$
$S \subset P$	subset of self promos
$P_1 \subset P$	set of *promos for programs*
$P_2 \subset P$	set of *non-channel promos* ($P_1 \cap P_2 = \emptyset$)
$A \subset P$	set of generic promos
$B \subset P$	set of specific promos ($G \cap Z = \emptyset$)

SAPIENZA
UNIVERSITÀ DI ROMA

Optimization framework

Parameters definition

P	set of promos
T	set of breaks
Q	set of client programs
K	set of genres
$P(c) \subseteq P$	subset of promos for campaign $c \in C$
$P(q) \subseteq P$	subset of promos for client program $c \in C$
$P(k) \subseteq P$	subset of promos for genre $k \in K$
$S \subset P$	subset of self promos
$P_1 \subset P$	set of *promos for programs*
$P_2 \subset P$	set of *non-channel promos* ($P_1 \cap P_2 = \emptyset$)
$A \subset P$	set of generic promos
$B \subset P$	set of specific promos ($G \cap Z = \emptyset$)
d_p	duration of promo p
D_t	duration of break t
r	required ratio of self promos
k_p	minimum number of times promo p is aired
l_p, u_p	*nominal* minimum and maximum number of times promo p can be aired

Optimization framework

Objective function

$$F(\delta) := f(\delta) + G(\delta)$$

$f(\delta)$	final objective we want to achieve

Sum of the viewership:
(IP-SUM)

$$f(\delta) := \sum_{p \in P} \sum_{t \in T} \delta_{pt} \phi_{pt}$$

Average of the viewership.
(IP_AVG)

$$f(\delta) := \sum_{q \in Q} \left(\frac{\sum_{p \in P(q)} \sum_{t \in T} \delta_{pt} \phi_{pt}}{\sum_{p \in P(q)} \sum_{t \in T} \delta_{pt}} \right)$$

$G(\delta)$	Sum of penalty terms with $\gamma_j \geq 0$

$$G(\delta) := \sum_{j \in \{SR, SC, SG, PV, FG\}} \gamma_j \; g_j(\delta),$$

Note that $f(\delta)$ is a surrogate of the real unknown function we want to optimize

Optimization framework

Hard constraints

The feasible region $\mathcal{P}$ is defined by the following constraints:

$$\sum_{p \in P} \delta_{pt} d_p \leq D_t \qquad \forall t \in T \tag{1}$$

$$\sum_{t \in T} \delta_{pt} \geq k_p \qquad \forall p \in P \tag{2}$$

$$\sum_{p \in P(j)} \delta_{pt} \leq 1 \qquad \forall t \in T, \forall j \in J = \{C, Q, K\} \tag{3}$$

(1) In each break the sum of **promos' duration** cannot be higher than the break duration itself;

(2) Each promo needs to be aired a **minimum number of times per day**;

(3) At most one promo of each specific **campaign, or client or genre** can be placed in each break (non-channel promos do not have a genre).

Optimization framework

Soft constraints 1/3

Ratio of self promos (SR). Each channel wants to broadcast more self-promos than cross-promos according to some business strategy

$$g_{\mathrm{SR}}(\delta) := -\left| \sum_{p \in S} \sum_{t \in T} \delta_{pt} - r \sum_{p \in P} \sum_{t \in T} \delta_{pt} \right|$$

Placement of self promos (SP). Each break should start and end with a self promo. This constraint (considered in the post-processing of the solution) implies that we should guarantee at least two self promos per break, however, this requirement cannot be always satisfied. As a consequence, it can be modeled as follows:

$$g_{\mathrm{SP}}(\delta) := -\sum_{t \in T} h_t(\delta)$$

Where

$h_t(\delta)$= 1 if $\sum_{p \in S} \delta_{pt} \leq 1$, 0 otherwise

Optimization framework

Soft constraints 2/3

Specifics versus generics (SG). Media companies want to give higher priority to specific promos over generic promos. This constraint is modeled directly in the objective function by adding the term:

$$g_{\mathrm{SG}}(\delta) := \sum_{p \in Z} \sum_{t \in T} \delta_{pt} - \sum_{p \in G} \sum_{t \in T} \delta_{pt}$$

Fill gap (FG). This rule is introduced to fill as much as possible all the available spaces within each break.

$$g_{\mathrm{FG}}(\delta) := -\left| \sum_{t \in T} \left(D_t - \sum_{p \in P} \delta_{pt} d_{pt} \right) \right|$$

Optimization framework

Soft constraints 3/3

Promos variation (PV). This rule guarantees that each promo is aired possibly no more than its ***desired number of times***, called θ_p, which depends on the relevance (i.e. the expected viewership) of the product advertised by the promo. The maximum frequency of each promo that the planning should target is defined as

$$\theta_p := \left\lceil l_p + (u_p - l_p)\,\frac{\mu_p - m}{M - m} \right\rceil \qquad \forall p \in P$$

where: $\mu_{p\in P} := \frac{\sum_{t\in T} \phi_{pt}}{T}$, $m := \min_{p\in P} \mu_p$ and $M := \max_{p\in P} \mu_p$

Then, the rule can be modeled by introducing the following function:

$$g_{\mathrm{PV}}(\delta) := -\sum_{p\in P} \max\left\{0, \sum_{t\in T} \delta_{pt} - \theta_p\right\}$$

Formulation as MILP

The final IP problem can be formulated as

$$\underset{\delta}{\text{maximize}} \quad \left\{F(\delta) : \delta \in \mathcal{P} \cap \{0,1\}^{|P| \times |T|}\right\}$$

Being both objective function and constraints linear (or can be linearized), both **IP-SUM** and **IP-AVG** can be written as MILP at the cost of adding some variables

What we obtain is an **easy-to-solve** MILP formulation to compute the optimal promo schedule

Numerical Results

Setting

- Real airtime data concerning a TV channel during 2018 were provided by a large media company
- ≈ 30 TB of data
- Approximately 65 breaks, 42 promos and 40 programs per day
- The entire solution has been deployed in **IBM Cloud Pak for Data**
- ML algorithms developed in **Python** with open-source libraries, like pandas, sklearn and xgboost
- Optimization models have been developed in Python with the IBM Decision Optimization **docplex** APIs, then solved with IBM Decision Optimization on **Cloud** APIs
- Cplex environment: IBM ILOG CPLEX 12.9 in an environment with 10 cores and 60 GB of RAM

Numerical Results

ML results

- Different approaches were tested and benchmarked
- 75 % for training and 25% for testing
- Hyper-parameter tuning and a cross validation was performed with expanding windows

	Predicting PR:	*Predicting CR:*
Best model:	XGBoost	XGBoost stacked with a NN (high accuracy + capturing the changes of viewership when placing a promo in different breaks)
R^2	$> 90\%$	$\approx 90\ \%$
MAPE	$\approx 10\%$	6 %
Most important features:	• Duration of the host program • Airing time	• Historical data of the client program • Historical promos' statistics (e.g. frequency of promos per day)

Numerical Results

Optimization results

IP-SUM takes less than 10 sec to solve the problem, while IP-AVG is not able to close the opt. gap but reach acceptable values within 5 mins which was set as limit time (lim).

		day-1	**day-2**	**day-3**	**day-4**	**day-5**	**day-6**	**day-7**
IP-SUM	time (sec)	3.918	3.272	5.448	8.524	8.789	0.663	3.633
	gap (%)	0	0	0	0	0	0	0
	# bin var	2880	3293	3210	3108	4113	2923	2340
	# cont var	24	25	25	24	24	22	23
	# lin cons	4054	4327	4513	4312	5485	4271	3321
	# ind cons	72	77	75	74	96	79	60
IP-AVG	time (sec)	291.278	lim	lim	lim	lim	lim	lim
	gap (%)	0	0.028	0.022	0.019	0.136	0.166	0.261
	# bin var	2900	3313	3232	3129	4133	2942	2359
	# cont var	2852	3261	3182	3079	4061	2885	2322
	# lin cons	6882	7563	7670	7367	9522	7134	5620
	# ind cons	5708	6529	6367	6163	8150	5786	4639

Numerical Results

Optimization results

δ': Business expert solution (human)

$\delta^\star$: Optimized solution

$$Gain := \frac{KPI(\delta^\star) - KPI(\delta')}{1 \times 10^{-6} + |KPI(\delta')|}$$

Day	KPI	IP-SUM			IP-AVG		
		$\delta = \delta'$	$\delta = \delta^\star$	Gain (%)	$\delta = \delta'$	$\delta = \delta^\star$	Gain (%)
day-2	$F(\delta)$	$1.23 \cdot 10^6$	$4.26 \cdot 10^6$	247.06	$1.80 \cdot 10^7$	$1.93 \cdot 10^7$	7.14
	$f_{\text{SUM}}(\delta(P_1))/F(\delta^\star)$ (%)	$2.07 \cdot 10^1$	$6.61 \cdot 10^1$	219.00	–	–	–
	$f_{\text{AVG}}(\delta(P_1))/F(\delta^\star)$ (%)	$2.89 \cdot 10^0$	$2.88 \cdot 10^0$	−0.47	$9.88 \cdot 10^{-6}$	$1.01 \cdot 10^{-5}$	1.80
	$f_{\text{SUM}}(\delta(P_2))/F(\delta^\star)$ (%)	$2.89 \cdot 10^1$	$4.39 \cdot 10^1$	51.71	–	–	–
	$f_{\text{AVG}}(\delta(P_2))/F(\delta^\star)$ (%)	$1.78 \cdot 10^0$	$2.03 \cdot 10^0$	13.81	$1.97 \cdot 10^{-6}$	$2.36 \cdot 10^{-6}$	13.34
	$\gamma_{\text{PV}} g_{\text{PV}}(\delta)$	$-6.00 \cdot 10^4$	$0.00 \cdot 10^0$	100.00	$-6.00 \cdot 10^4$	$0.00 \cdot 10^0$	100.00
	$\gamma_{\text{SR}} g_{\text{SR}}(\delta)$	$-1.01 \cdot 10^6$	$-2.55 \cdot 10^6$	−152.78	$-1.01 \cdot 10^6$	$-1.68 \cdot 10^5$	83.33
	$\gamma_{\text{SG}} g_{\text{SG}}(\delta)$	$2.63 \cdot 10^2$	$1.67 \cdot 10^3$	534.29	$2.63 \cdot 10^2$	$3.75 \cdot 10^2$	42.86
	$\gamma_{\text{FG}} g_{\text{FG}}(\delta)$	–	–	–	$5.67 \cdot 10^3$	$6.07 \cdot 10^3$	6.99

- Both IP-SUM and IP-AVG **improve human-tuned solution**
- Some of the **human-tuned solutions were not feasible**
- The penalty terms γ_j allow us to **tune the solution** according to the strategic requirements specific to each TV media company.

Conclusion

- OPS effectively solved in a very fast way
- We always obtain better solutions than those found by business experts
- γ_j parameters allow us to tune the solution according to the specific business strategy
- The framework can be easily extended to consider more channels and include commercials

SAPIENZA
UNIVERSITÀ DI ROMA

Thanks for the attention.
Questions?

ruggiero.seccia@uniroma1.it