How Birchbox Transformed Its Operations With Mathematical Optimization

Thank you for joining us. We will be starting shortly.

GUROBI OPTIMIZATION

The World's Fastest Solver

PRINCETON CONSULTANTS
Information Technology and Management Consulting

BIRCHBOX

Today's Speakers

Irv Lustig, PhD

Optimization Principal
Princeton Consultants

Rob Randall, PhD

Senior Optimization Specialist
Princeton Consultants

David Bendes

VP, Global Business Technology
Birchbox

How Birchbox Transformed Its Operations With Mathematical Optimization

Irv Lustig, PhD, Optimization Principal at Princeton Consultants

Rob Randall, PhD, Senior Optimization Specialist at Princeton Consultants

GUROBI OPTIMIZATION

The World's Fastest Solver

PRINCETON CONSULTANTS

Information Technology and Management Consulting

BIRCHBOX

About Princeton Consultants (www.princeton.com)

Princeton New Jersey

Information Technology & Management Consulting

Advanced Analytics *Strategy*
Application Development *Process Improvement*

Optimization into Production

New York City

- **Stability**: Founded in 1981; over 1800 successful completed projects for many industry leaders and fast-growing innovators.

GUROBI OPTIMIZATION PREMIER PARTNER

- **Representative Clients**: 2020 US Census, US Tennis Association, NBCUniversal, Wells Fargo, NetJets, The Wall Street Journal, Schneider.

- **Analytics-focused Professional Staff**: 65 full-time consultants, 2/3rds with graduate STEM degrees; plus network of top independent consultants and university professors. 50+ experienced data scientists/developers.

- **Experience**: Senior Staff (top 28 consultants) average 15+ years experience; Firm Leadership (6 Directors) average 20+ years experience.

PRINCETON CONSULTANTS

Information Technology and Management Consulting

GUROBI OPTIMIZATION

Formulation Matters: Reciprocating Integer Programming for Birchbox Product Assortment

Irv Lustig, PhD, Optimization Principal

Patricia Randall, PhD, Director

Robert Randall, PhD, Senior Optimization Specialist

BIRCHBOX

Birchbox

› Birchbox is an online monthly subscription service that sends its customers a box of beauty product samples each month

Launched in 2010

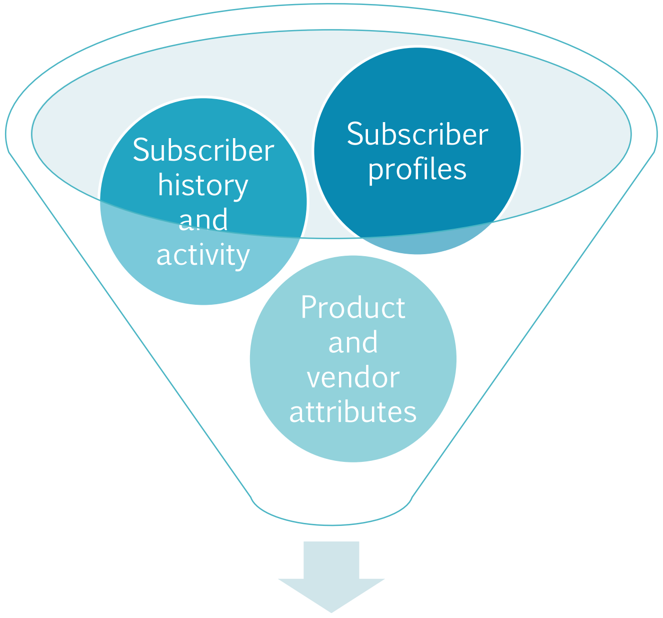
Operates in 3 countries, with over 2.5 million active customers

Samples tailored to each subscriber's needs and preferences

500+ best-in-class prestige brand partners with products including skincare items, perfumes, and other cosmetics

The Competitive Edge

- A subscription box service's operational success is powered by its "black box" – grouping subscribers into clusters and creating configurations of available products for each subscriber cluster


Subscriber history and activity
Subscriber profiles
Product and vendor attributes

Optimal box creation

- Birchbox uses a mixed integer programming (MIP) model with Gurobi to optimize this process by using their proprietary objectives and constraints

Problems with the Initial Model

- The product grouping and box creation MIP worked for several years but as the customer base grew and offerings expanded, the run time increased to multiple days
  - Operations was up against production deadlines every month and forced to monitor the run closely over the course of days
  - Alternative business models could not be tested
  - New improvements to the box experience could not be adequately handled by the existing model

Engagement History

- Birchbox contacted Princeton due to performance issues with their existing MIP model that used Gurobi

- Princeton Offer:
  - Advanced Analytics Model Review & Validation Service

- Benefits
  - Validate and improve optimization and predictive models with an expert third party review
    - Uncover new ideas for improvement
    - Benchmark your models and group against best practices
    - Give business leaders more confidence in your solutions
    - Help your specialists improve their development skills

Simplified Problem Description

- 100 Different Products
  - Each product has different attributes
    - Weight, Volume, Value, etc.
  - There is a maximum inventory of each product
  - There is a desired minimum number of subscribers to send each product to
    - Often zero
- 2000 to 5000 “Subscribers”
  - Depends on number of clusters from machine learning
  - For each subscriber/product combination, define “eligibility”:
    - A subscriber can receive the product
    - A subscriber must receive the product
    - A subscriber must not receive the product

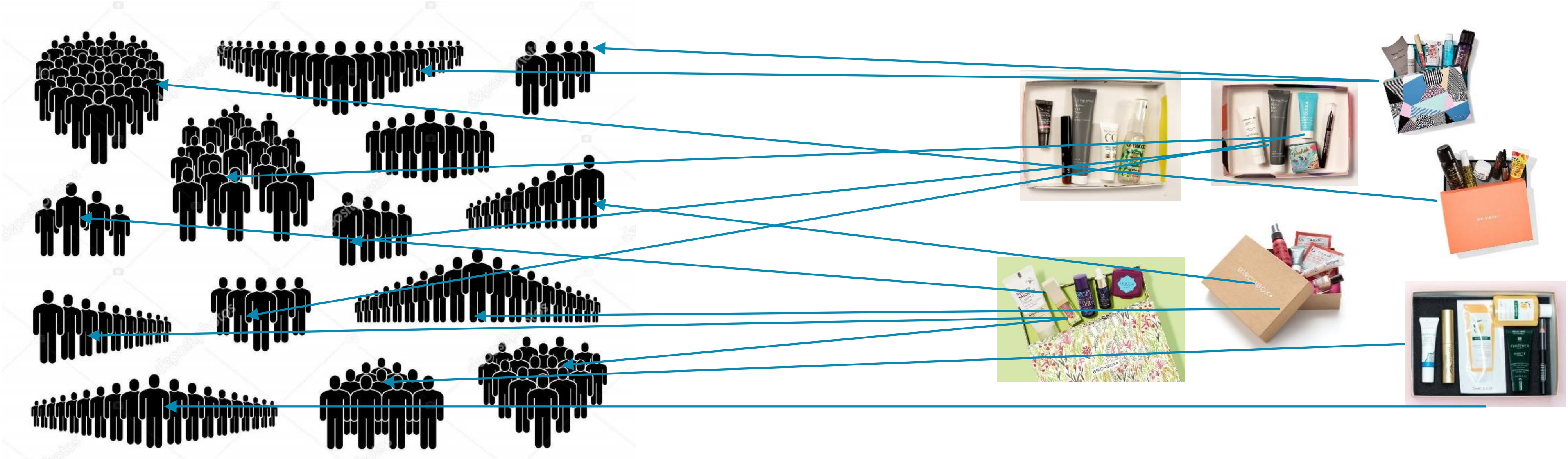
Goal 1: Create Box Configurations

› A Configuration is a group of products

- Configurations satisfy constraints on the attributes
  - Range on number of products included
    - E.g., at least 5, no more than 7
  - Maximum weight, volume
  - Ranges on other attributes
  - Etc.

Goal 2: Assign Box Configurations to Subscribers

- Each configuration must be sent to a minimum number of subscribers and to at most a maximum number of subscribers
- Every subscriber must be assigned a configuration
  - Identify quickly if this is not possible due to eligibility or lack of inventory



Configuration Goals

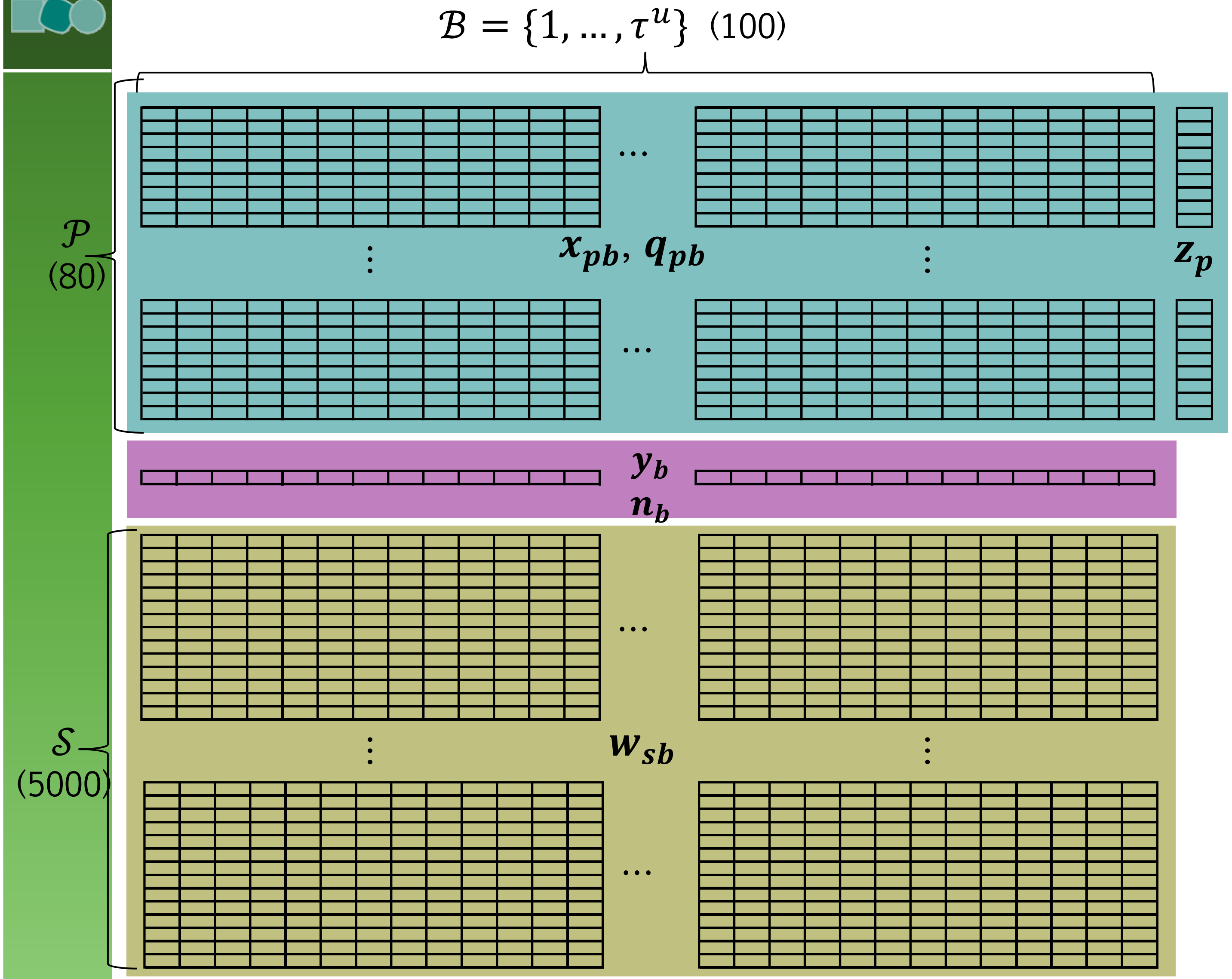
- Use a small number of configurations
  - Typically below 100.
  - For grooming products, could be as low as 30
- Respect product maximum inventory used
- Some subscribers must receive certain products.
- Some subscribers must not receive certain products.

Notation

- $\mathcal{P}$ is the set of products •$\mathcal{S}$ is the set of subscribers •$\mathcal{B}$ is the set of box configurations
- $\mathcal{A}$ is the set of attributes that exist for all products
  - μ_{ap} is the value of attribute a for product p
- $\widehat{\mathcal{S}}_p \subseteq \mathcal{S}$ are subscribers that must receive product p
  - $\widehat{\mathcal{P}}_s \subset \mathcal{P}$ are the products that subscriber s must receive.
- $\widetilde{\mathcal{S}}_p \subset \mathcal{S}$ are subscribers that must not receive product p
  - $\widetilde{\mathcal{P}}_s \subset \mathcal{P}$ are the products that subscriber s must not receive.

Property	Scope	Minimum	Maximum
Number of subscribers per box	Global across boxes	σ^ℓ	σ^u
Number of Different Box Configurations	Global	τ^ℓ	τ^u
Number of products per box	Global	γ^ℓ	γ^u
Total Product Quantity Across subscribers	Per product $p \in \mathcal{P}$	ϕ_p^ℓ(soft)	ϕ_p^u
Total Attribute Value	For all boxes, per attribute $a \in \mathcal{A}$	α_a^ℓ	α_a^u

Initial MIP Formulation: Variables



x_{pb}: (binary) Is product p in box b?

q_{pb}: Number of times product p is in box b

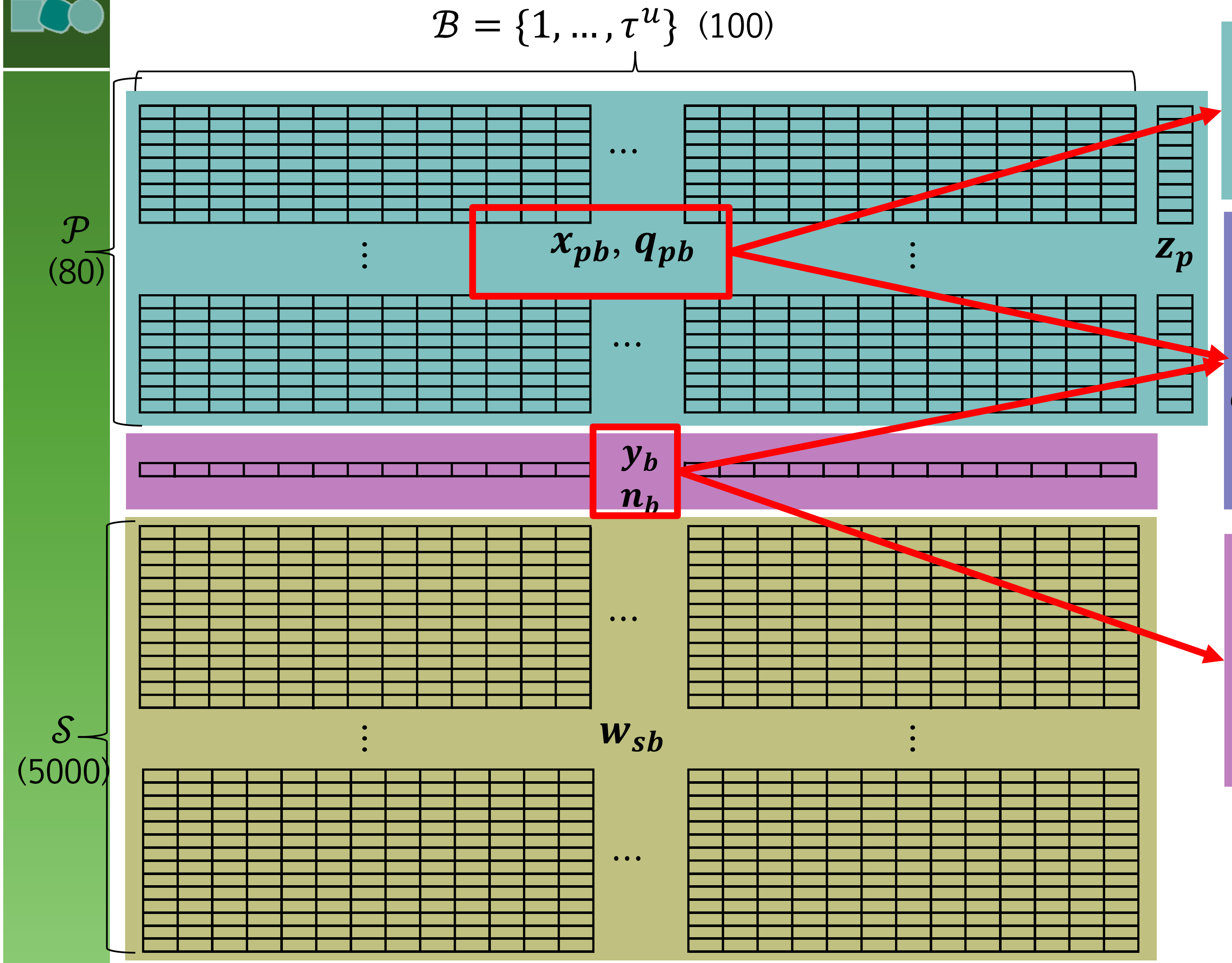
z_p: Error for not satisfying minimum product quantity

y_b: (binary) Is box b used?

n_b: Number of times box b is used (number of subscribers assigned)

w_{sb}: (binary) Is subscriber s assigned to box b?

Initial MIP Formulation: Box/Product Logical Constraints



$q_{pb} \le \phi_p^u x_{pb}$ Number of products in box controlled by whether product is in box

$z_p \le \phi_p^u$ Error limited by maximum

$x_{pb} \le y_b$

$q_{pb} \le \phi_p^u y_b$ Product in box if box is used

$q_{pb} \ge n_b - M(1 - x_{pb})$

$q_{pb} \le n_b$ Number of products in box is equal to number of subscribers if the product is in the box

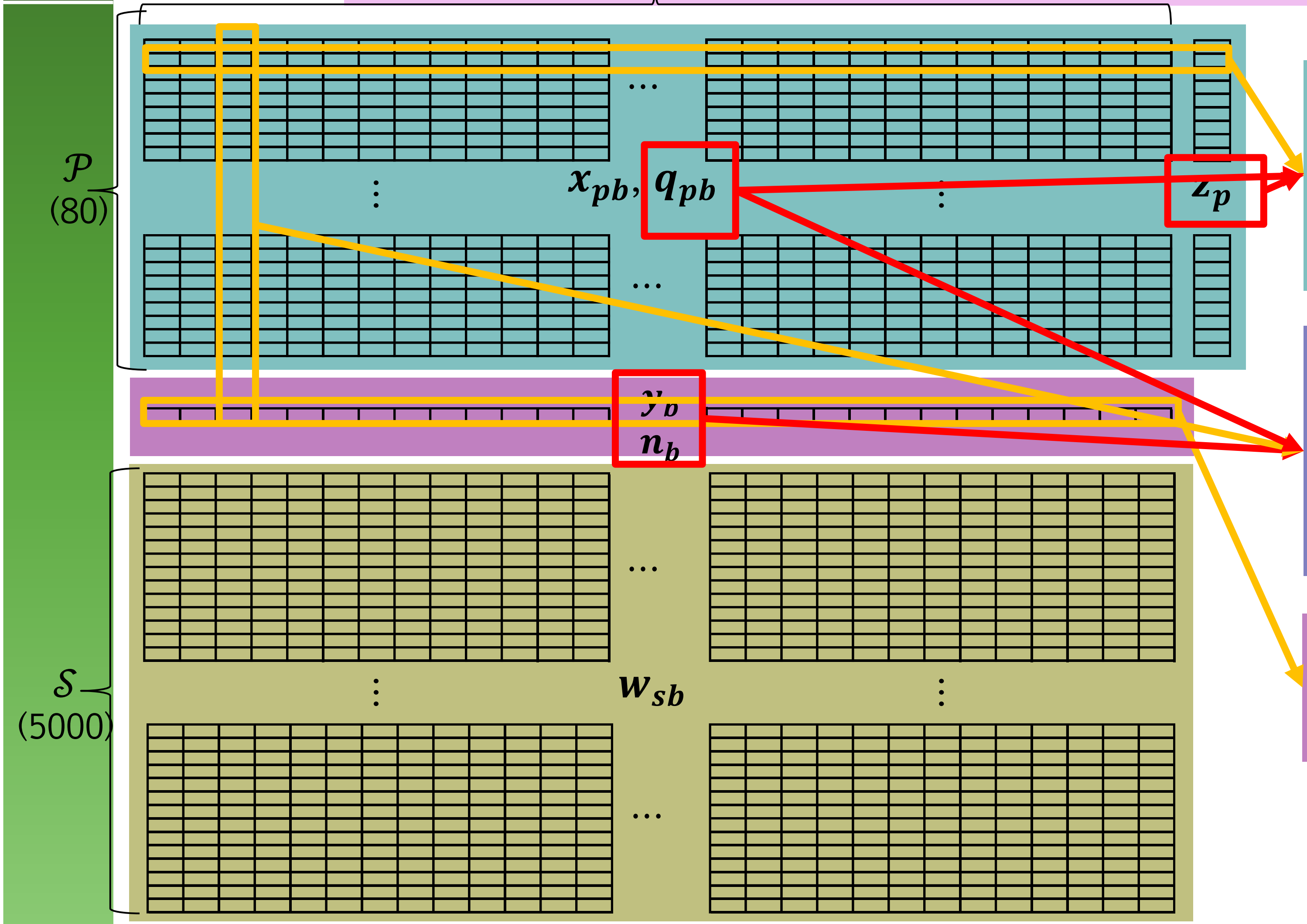
$n_b \le \sigma^u y_b$

$n_b \ge \sigma^\ell y_b$ Number of products in box controlled by whether box is used and controlled by minimum and maximum subscribers per box (for each box)

Initial MIP Formulation: Box/Product Quantity Constraints

$\mathcal{B} = \{1, \dots, \tau^u\}$ (100)

Enforces maximum number of box configurations



$$\sum_{b \in \mathcal{B}} q_{pb} + z_p \geq \phi_p^\ell$$

$$\sum_{b \in \mathcal{B}} q_{pb} \leq \phi_p^u$$

For each product, across boxes, total product used is limited by minimums and maximums

$$\sum_{p \in \mathcal{P}} q_{pb} \geq \gamma^\ell n_b$$

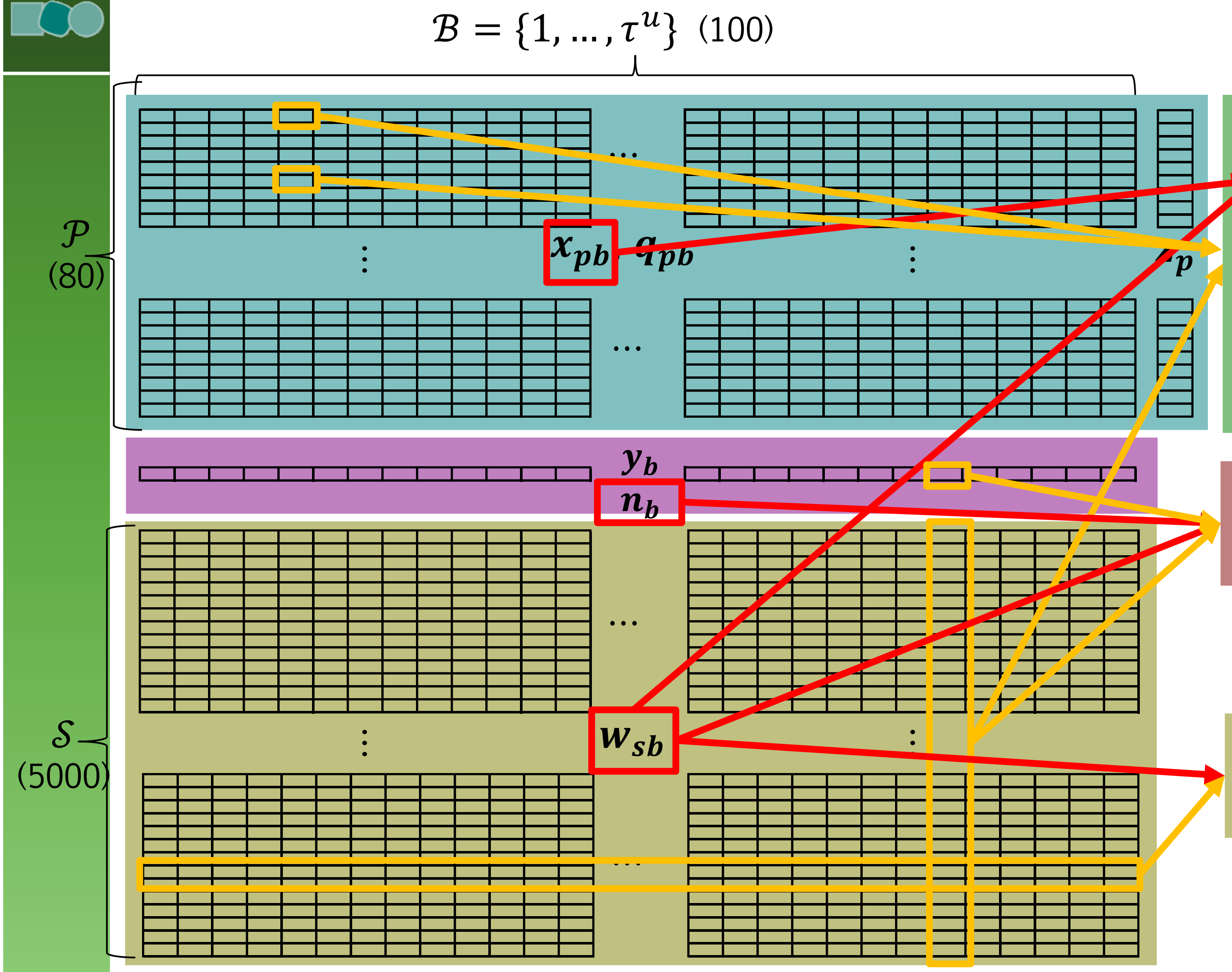
$$\sum_{p \in \mathcal{P}} q_{pb} \leq \gamma^u n_b$$

For each box, total product used must correspond to minimums and maximums per box

$$\sum_{b \in \mathcal{B}} y_b \geq \tau^\ell$$

Must build a minimum number of box configurations

Initial MIP Formulation: Subscriber Constraints



$$\sum_{s\in\hat{\mathcal{S}}_p} w_{sb} \le |\hat{\mathcal{S}}_p| x_{pb}$$

For each product and box, force certain subscribers to be assigned

$$\sum_{s\in\tilde{\mathcal{S}}_p} w_{sb} \le |\tilde{\mathcal{S}}_p|(1 - x_{pb})$$

For each product and box, force certain subscribers to *not* be assigned

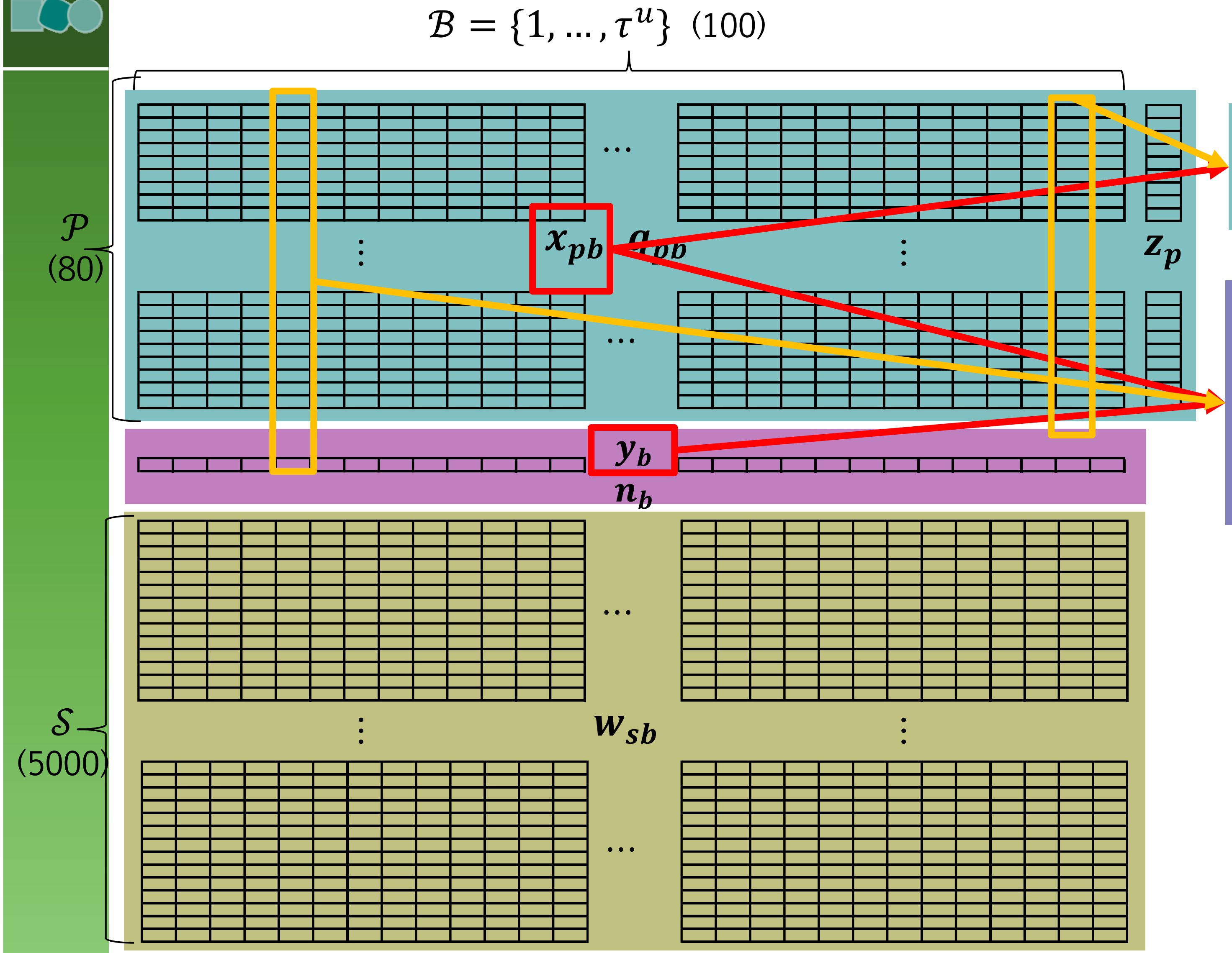
$$\sum_{s\in\mathcal{S}} w_{sb} = n_b$$

Count the number of subscribers assigned for each box

$$\sum_{b\in\mathcal{B}} w_{sb} \le 1$$

Limit each subscriber to be assigned to one box

Initial MIP Formulation: Box Composition Constraints



$$\alpha_a^\ell \le \sum_{p\in\mathcal{P}} \mu_{ap}\, x_{pb} \le \alpha_a^u$$

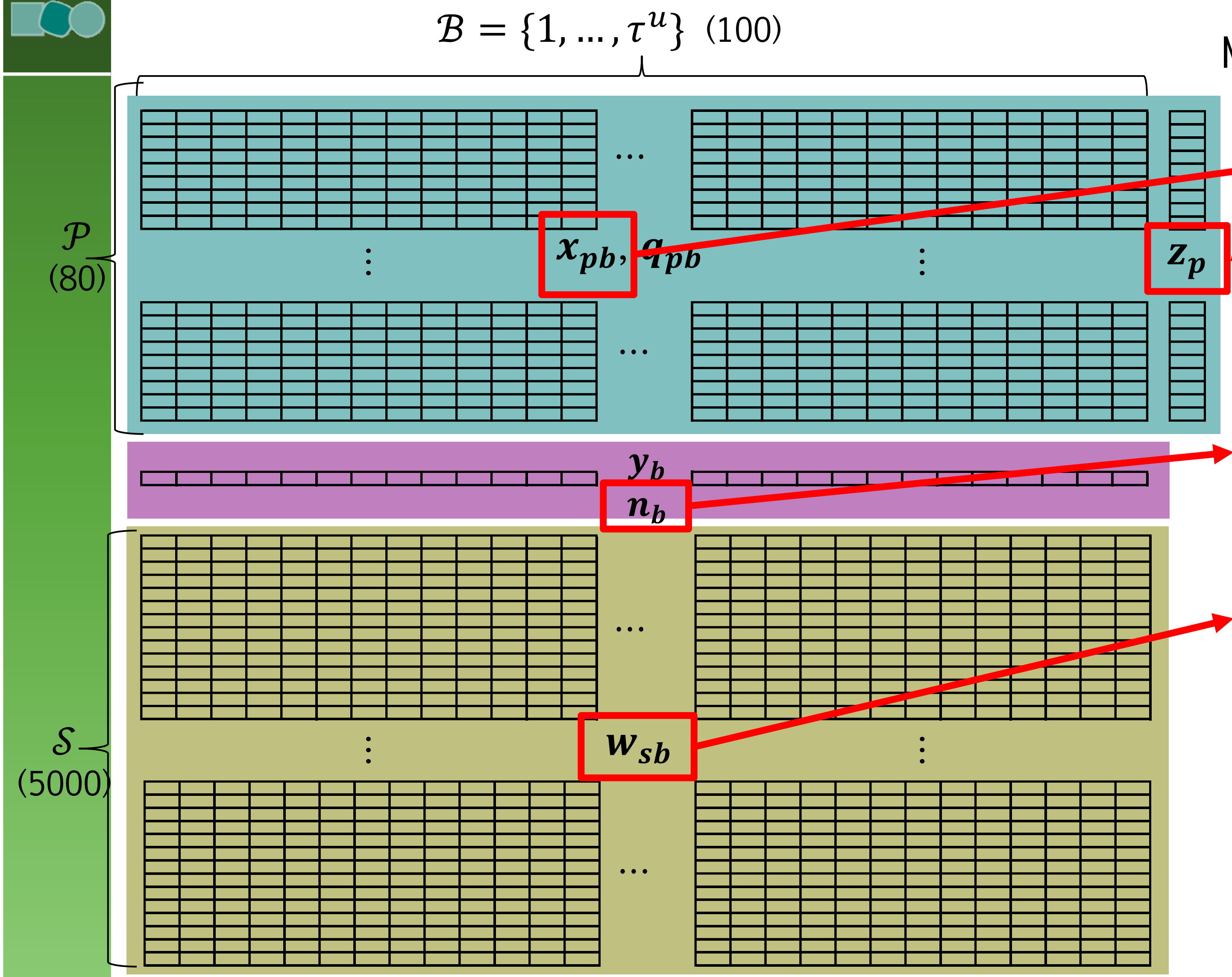
Attribute Constraints

$$\sum_{p\in\mathcal{P}} x_{pb} \le \gamma^u y_b$$

$$\sum_{p\in\mathcal{P}} x_{pb} \ge \gamma^\ell y_b$$

Min/Max Different Products per Box

Initial MIP Formulation: Objective Function



MAXIMIZE

$$\sum_{b\in\mathcal{B}}\sum_{p\in\mathcal{P}}\frac{1}{3600}\cdot x_{pb}$$

"Generosity" – More products are better

$$-\sum_{p\in\mathcal{P}}\frac{2}{(\phi_p^u+1)}\cdot z_p$$

Error in reaching desired minimum product quantities

$$+\frac{1}{2}\sum_{b\in\mathcal{B}} n_b$$

Total number of boxes built

$$+\frac{1}{2}\sum_{s\in\mathcal{S},b\in\mathcal{B}} w_{sb}$$

Number of Subscribers Assigned to a box

Note: Because $\sum_{s\in\mathcal{S}} w_{sb} = n_b$,

$$\frac{1}{2}\sum_{b\in\mathcal{B}} n_b + \frac{1}{2}\sum_{s\in\mathcal{S},b\in\mathcal{B}} w_{sb} = \sum_{s\in\mathcal{S},b\in\mathcal{B}} w_{sb}$$

Initial Formulation: Issues

- Gurobi 8.0
- Solve Times ranged from minutes for easy problems to multiple days for most problems
  - Using 30 cores of 8 processor/4 cores per processor, AMD Opteron 6328, 3.2Ghz, 2MB cache
- Small changes in the input data caused huge variance in performance
  - E.g. just changing γ^u (maximum number of products per box) significantly increased solve times.
- Blended Objectives
  - Get every subscriber a box
  - Try to meet product minimums
- Symmetry
  - No differentiation between x_{pb_1} and x_{pb_2} for two different box numbers b_1 and b_2

Solution: Pattern Generation for Each Subscriber

$$\sum_{s\in\hat{\mathcal{S}}_p} w_{sb} \le |\hat{\mathcal{S}}_p| x_{pb}$$

For each product and box, force certain subscribers to be assigned

$$\sum_{s\in\tilde{\mathcal{S}}_p} w_{sb} \le |\tilde{\mathcal{S}}_p|(1-x_{pb})$$

For each product and box, force certain subscribers to *not* be assigned

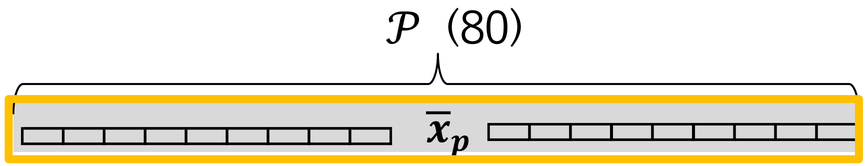
$$\alpha_a^{\ell} \le \sum_{p\in\mathcal{P}} \mu_{ap}\, x_{pb} \le \alpha_a^{u}$$

Attribute Constraints

$$\sum_{p\in\mathcal{P}} x_{pb} \le \gamma^{u} y_b$$

$$\sum_{p\in\mathcal{P}} x_{pb} \ge \gamma^{\ell} y_b$$

Min/Max Different Products per Box



$$\bar{x}_p = 1 \qquad \forall p \in \hat{\mathcal{P}}_s$$
$$\bar{x}_p = 0 \qquad \forall p \in \tilde{\mathcal{P}}_s$$

S

$$\alpha_a^{\ell} \le \sum_{p\in\mathcal{P}} \mu_{ap}\, \bar{x}_p \le \alpha_a^{u}$$

$$\sum_{p\in\mathcal{P}} \bar{x}_p \le \gamma^{u}$$

$$\sum_{p\in\mathcal{P}} \bar{x}_p \ge \gamma^{\ell}$$

Maximize

$$\sum_{p\in\mathcal{P}} \bar{\pi}_p\, \bar{x}_p$$

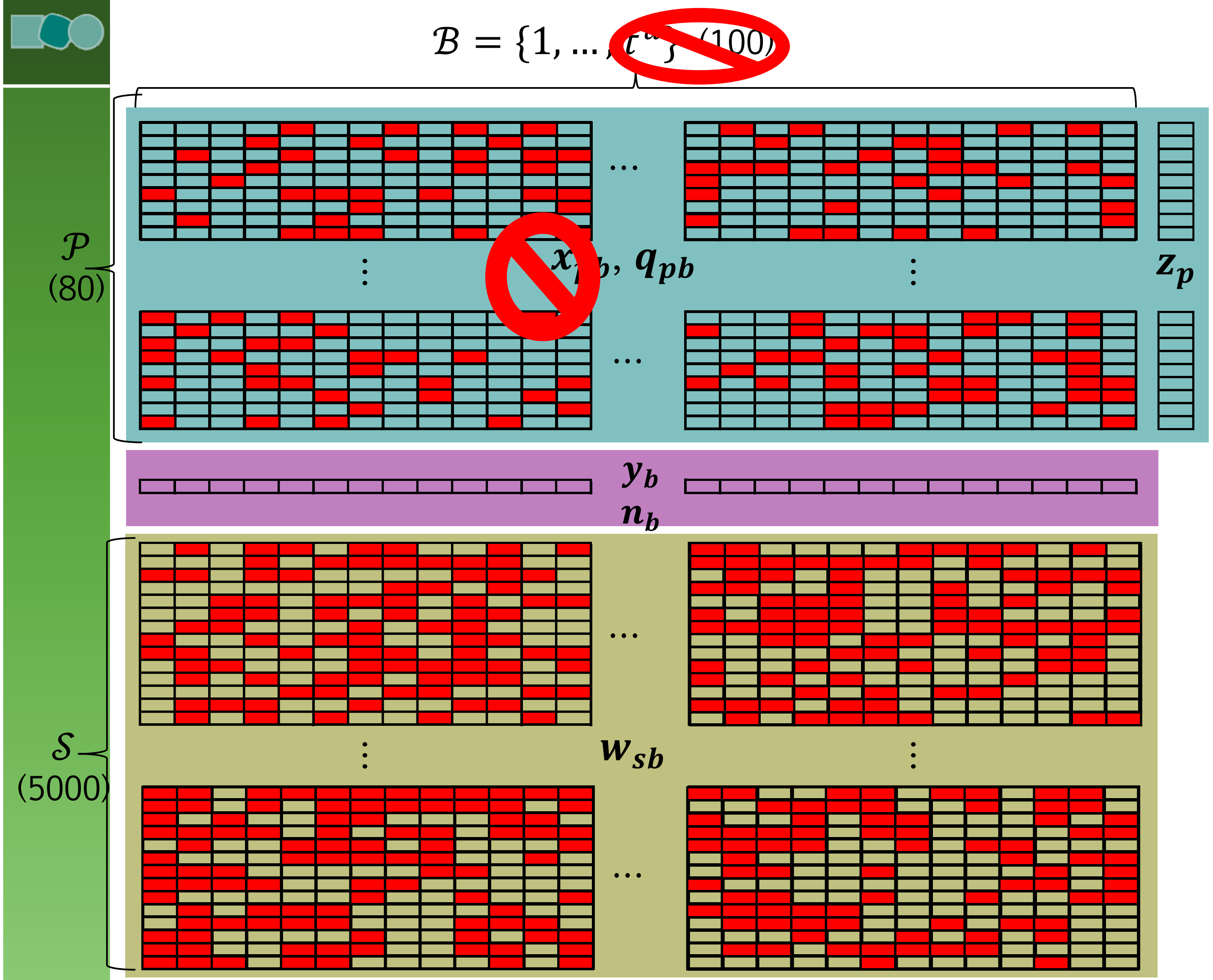
($\bar{\pi}_p$ are “duals”)

Prevent Pattern Regeneration

$$\sum_{p\in\bar{\mathcal{P}}_b} \bar{x}_p - \sum_{p\notin\bar{\mathcal{P}}_b} \bar{x}_p \le |\bar{\mathcal{P}}_b| - 1$$

Feasible New box **b**: $\bar{x}_p^b$ → $\bar{\mathcal{P}}_b = \{p \in \mathcal{P} : \bar{x}_p^b = 1\}$

Master MIP Problem: Variables



~~x_{pb}: Is product p in box b?~~

q_{pb}: Number of times product p is in box b, only if product in the box

z_p: Error for not satisfying minimum product quantity

y_b: Is box b used?

n_b: Number of times box b is used (number of subscribers assigned)

w_{sb}: Is subscriber s assigned to box b?

(Only if subscriber s is eligible for box b)

Master MIP Problem: Box/Product Logical Constraints

$q_{pb} \le \phi_p^u x_{pb}$	Number of products in box controlled by whether product is in box
$z_p \le \phi_p^u$	Error limited by maximum
$x_{pb} \le y_b$	Product in box if box is used
$q_{pb} \le \phi_p^u y_b$ $q_{pb} \ge n_b - M(1 - x_{pb})$ $q_{pb} \le n_b$	Number of products in box is equal to number of subscribers if the product is in the box
$n_b \le \sigma^u y_b$ $n_b \ge \sigma^\ell y_b$	Number of products in box controlled by whether box is used and controlled by minimum and maximum subscribers per box (for each box)

$z_p \le \phi_p^u$	Error limited by maximum
$q_{pb} \le \phi_p^u y_b$	Product in box if box is used
$q_{pb} = n_b$	Number of products in box is equal to number of subscribers assigned to the box
$n_b \le \sigma^u y_b$ $n_b \ge \sigma^\ell y_b$	Number of products in box controlled by whether box is used and controlled by minimum and maximum subscribers per box (for each box)

Master MIP Problem: Box/Product Quantity Constraints

$\mathcal{B} = \{1, \ldots, \tau^u\}$

Enforces maximum number of box configurations

$$\sum_{b \in \mathcal{B}} q_{pb} + z_p \geq \phi_p^\ell$$

$$\sum_{b \in \mathcal{B}} q_{pb} \leq \phi_p^u$$

For each product, across boxes, total product used is limited by minimums and maximums

$$\sum_{p \in \mathcal{P}} q_{pb} \geq \gamma^\ell n_b$$

$$\sum_{p \in \mathcal{P}} q_{pb} \leq \gamma^u n_b$$

For each box, total product used must correspond to minimums and maximums per box

$$\sum_{b \in \mathcal{B}} y_b \geq \tau^\ell$$

Must build a minimum number of box configurations

$$\sum_{b \in \mathcal{B}} y_b \leq \tau^{\boldsymbol{u}}$$

Must build at most a certain number of box configurations (HARD option)

$$\sum_{b \in \mathcal{B}} q_{pb} + z_p \geq \phi_p^\ell$$

$$\sum_{b \in \mathcal{B}} q_{pb} \leq \phi_p^u$$

For each product, across boxes, total product used is limited by minimums and maximums

$$\sum_{p \in \mathcal{P}} q_{pb} \geq \gamma^\ell n_b$$

$$\sum_{p \in \mathcal{P}} q_{pb} \leq \gamma^u n_b$$

For each box, total product used must correspond to minimums and maximums per box

$$\sum_{b \in \mathcal{B}} y_b \geq \tau^\ell$$

Must build a minimum number of box configurations

Master MIP Problem: Subscriber Constraints

$\sum_{s\in\hat{\mathcal{S}}_p} w_{sb} \le \lvert\hat{\mathcal{S}}_p\rvert x_{pb}$	For each product and box, force certain subscribers to be assigned
$\sum_{s\in\tilde{\mathcal{S}}_p} w_{sb} \le \lvert\tilde{\mathcal{S}}_p\rvert (1 - x_{pb})$	For each product and box, force certain subscribers to ***not*** be assigned
$\sum_{s\in\mathcal{S}} w_{sb} = n_b$	Count the number of subscribers assigned for each box
$\sum_{b\in\mathcal{B}} w_{sb} \le 1$	Limit each subscriber to be assigned to one box

→

$\sum_{s\in\mathcal{S}} w_{sb} = n_b$	Count the number of subscribers assigned for each box
$\sum_{b\in\mathcal{B}} w_{sb} \le 1$	Limit each subscriber to be assigned to one box

Master MIP Problem: Hierarchical Objective Function

MAXIMIZE

$$\sum_{b\in\mathcal{B}}\sum_{p\in\mathcal{P}}\frac{1}{3600}\cdot x_{pb}$$

“Generosity” – More products are better

$$-\sum_{p\in\mathcal{P}}\frac{2}{(\phi_p^u+1)}\cdot z_p$$

Error in reaching desired minimum product quantities

$$+\frac{1}{2}\sum_{b\in\mathcal{B}} n_b$$

Total number of boxes built

$$+\frac{1}{2}\sum_{s\in\mathcal{S},b\in\mathcal{B}} w_{sb}$$

Number of Subscribers Assigned to a box

MAXIMIZE

$$\sum_{s\in\mathcal{S},b\in\mathcal{B}} w_{sb}$$

Number of Subscribers Assigned to a box

MINIMIZE

$$\sum_{b\in\mathcal{B}} y_b$$

Number of boxes used
(Target is τ^u)
(SOFT Option)

MINIMIZE

$$\sum_{p\in\mathcal{P}}\frac{1}{(\phi_p^u+1)}\cdot z_p$$

Error in reaching desired minimum product quantities

New Formulation

- Full Master Problem has all possible patterns
  - Is probably very large!

- Address via “column generation” techniques
  - Generate patterns as needed
  - Form a Restricted Master Problem (RMP)

- Not true column generation
  - Adding new variables and new constraints
- Not “Branch-and-price”
  - Could be considered “Price-and-branch”

Column and Constraint Generation

- Each new pattern $\boldsymbol{b}$ adds variables

$\boldsymbol{q_{pb}}$: Number of times product $\boldsymbol{p}$ is in box $\boldsymbol{b}$ for each product

$\boldsymbol{y_b}$: (binary) Is box $\boldsymbol{b}$ used?
$\boldsymbol{n_b}$: Number of times box $\boldsymbol{b}$ is used (number of subscribers assigned)

$\boldsymbol{w_{sb}}$: (binary) Is subscriber $\boldsymbol{s}$ assigned to box $\boldsymbol{b}$? (for each eligible subscriber)

- Each new pattern $\boldsymbol{b}$ adds constraints

$$q_{pb} \le \phi_p^u y_b$$

Product in box if box is used

$$q_{pb} = n_b$$

Number of products in box is equal to number of subscribers assigned to the box

$$\sum_{p\in\mathcal{P}} q_{pb} \ge \gamma^\ell n_b$$

$$\sum_{p\in\mathcal{P}} q_{pb} \le \gamma^u n_b$$

For each box, total product used must correspond to minimums and maximums per box

$$n_b \le \sigma^u y_b$$
$$n_b \ge \sigma^\ell y_b$$

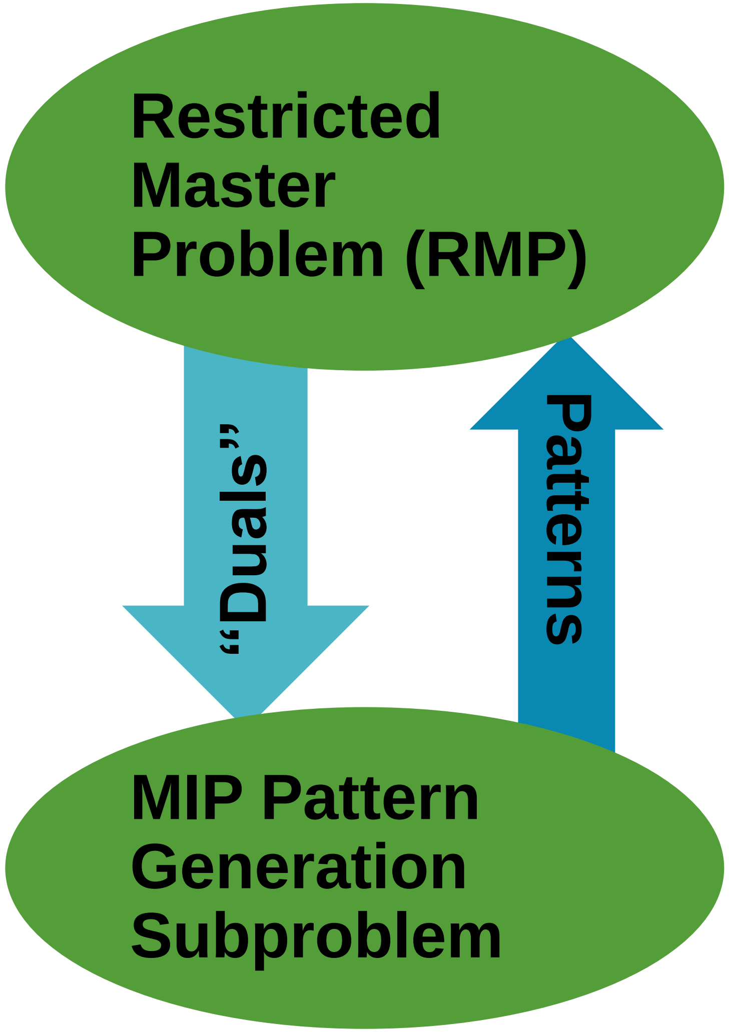
Number of products in box controlled by whether box is used and controlled by minimum and maximum subscribers per box (for each box)

$$\sum_{s\in\mathcal{S}} w_{sb} = n_b$$

Count the number of subscribers assigned for each box

Reciprocating Integer Programming (RIP)

- Two MIPs
  - Communicate by “reciprocating” information
- Pattern Generation: Solve a MIP!!!
  - Bootstrap
    - Generate a lot of patterns, enough for each subscriber, preferring products with lots of inventory
  - Generate New Patterns
- RMP: First solve an LP, then solve a MIP!
  - Get all subscribers a box with the first objective (maximize number of assigned subscribers)
  - Then move to MIP
- RMP generates “Duals” for objective of subproblem
  - Even when the RMP is a MIP!


Restricted Master Problem (RMP)
“Duals”
Patterns
MIP Pattern Generation Subproblem

RIP Step 1: Bootstrap

- Solve the Pattern Generation MIP
  - Just generate feasible solutions
- Generate enough patterns for each subscriber
- Make sure each product has enough patterns

- Dynamically adjust objective function
  - Initially, $\bar{\pi}_p = \phi_p^u$. Choose products with the most inventory
  - Dynamically adjust to incent products to be used at same rate
    - $\mathtt{\textbf{ProdCnt}}[p]$ is number of patterns containing product p

$$\bar{\pi}_p = \frac{\phi_p^u \max_{\hat{p} \in \mathcal{P}} \mathtt{\textbf{ProdCnt}}[\hat{p}]}{1 + \mathtt{\textbf{ProdCnt}}[p]}$$

RIP Step 2: Relax

- Based on the Sifting Method: Dynamically add patterns to build up the RMP
  - Bixby, Gregory, Lustig, Marsten, Shanno (1992)
- Solve LP relaxation of RMP
  - Use Gurobi's deterministic concurrent optimization
    - Can use basis from previous solve to start
- Remove nonbasic variables and associated patterns with negative reduced costs
- Add duals from specific constraints to compute $\hat{\pi}_p$
- Use $\hat{\pi}_p$ to add back previously deleted patterns
- Adjust objective function for subproblem (v_p is unused inventory)

$$\bar{\pi}_p = \begin{cases} \frac{1}{2}\left(\min_{p\in\mathcal{P}:\hat{\pi}_p>0} \hat{\pi}_p\right)\left(\frac{v_p}{1+\max_{p\in\mathcal{P}} v_p} - 1\right) & \text{if } \hat{\pi}_p = 0 \\ -\hat{\pi}_p & \text{if } \hat{\pi}_p > 0 \end{cases}$$

- Stop when RMP master problem objective is equal to the number of subscribers

RIP Step 3: Populate

- Create the MIP version of RMP
  - Do *not* include all variables from the LP relaxation!
- Choose all patterns included in the LP relaxation solution basis or patterns that are assigned (even fractionally) to a subscriber
- For each pattern, ensure sufficient subscribers per pattern
  - Do not include all eligible subscribers for each pattern
- For each product with positive values of ϕ_p^ℓ, ensure that there are enough patterns (and associated subscribers) that include that product
- For each subscriber, ensure sufficient patterns per subscribers
  - Do not include all eligible patterns for each subscriber
- Overriding goal – try to not make the MIP have too many variables (less than 300,000 worked well)

RIP Step 4: Subscribe

- Solve the MIP form of the RMP
- Stop if
  - Upper bound is proven to be smaller than the number of subscribers
  - Solution found equal to the number of subscribers (Proof of optimality not required!)
  - Gurobi hits a small node limit
- If adequate solution not found, add back all deleted patterns
  - Solve MIP form of the RMP again
- If adequate solution still not found, generate new patterns using surrogate dual values

$$\bar{\pi}_p = \frac{1}{v_p+1}$$

RIP Steps 5 and 6: Boxes and Product

- Boxes Step: First Add Constraint: $\sum_{b \in \bar{\mathcal{B}}, s \in \mathcal{S}} w_{sb} \geq |\mathcal{S}|$
  - Enforce every subscriber gets a box

HARD OPTION	SOFT OPTION
› Skip to Product Step	› Minimize Number of Boxes $\sum_{b \in \bar{\mathcal{B}}} y_b$ › Stop when objective at or below τ^u › Add Constraint $\sum_{b \in \bar{\mathcal{B}}} y_b \leq \tau^u$

- Product Step:
  - Minimize $\sum_{p \in \mathcal{P}} \frac{1}{(\phi_p^\ell + 1)} z_p$ (violation of product minimums)
  - Stop if
    - Optimal
    - User terminates (provide values of intermediate solutions to allow abort)
    - After 100 nodes of branching

Parallel MIP

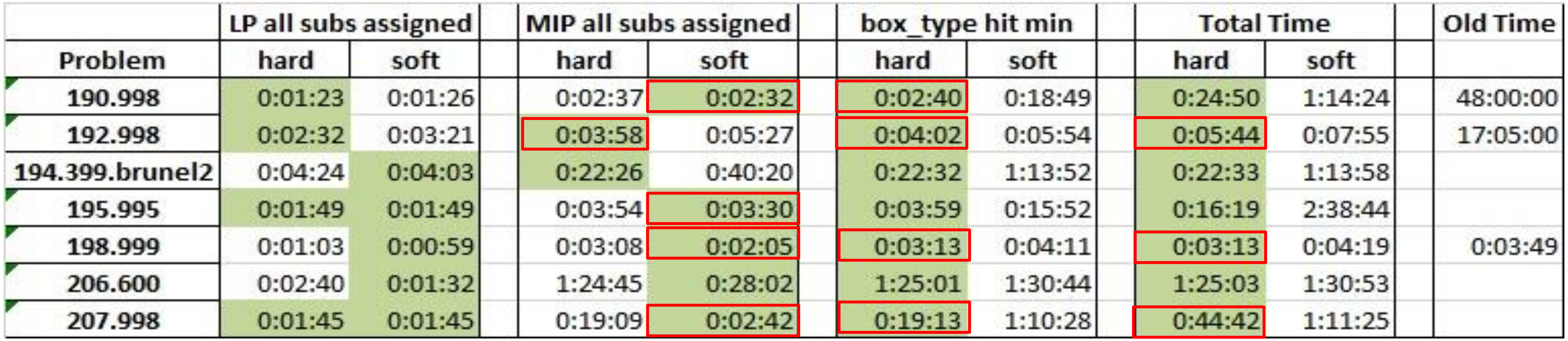
- Two Options
  - Use all processors on one MIP of restricted master problem
  - Use Gurobi Concurrent Optimization
    - Set a random seed for each parallel run
    - Can help the Gurobi heuristics find feasible solutions faster

- Computational Results using Birchbox hardware:
  - 24 cores of 8 processor/4 cores per processor, AMD Opteron 6328, 3.2Ghz, 2MB cache
  - Concurrent Optimization for MIP: 6 parallel solves, 4 threads each

Computational Results (Gurobi 9.0)

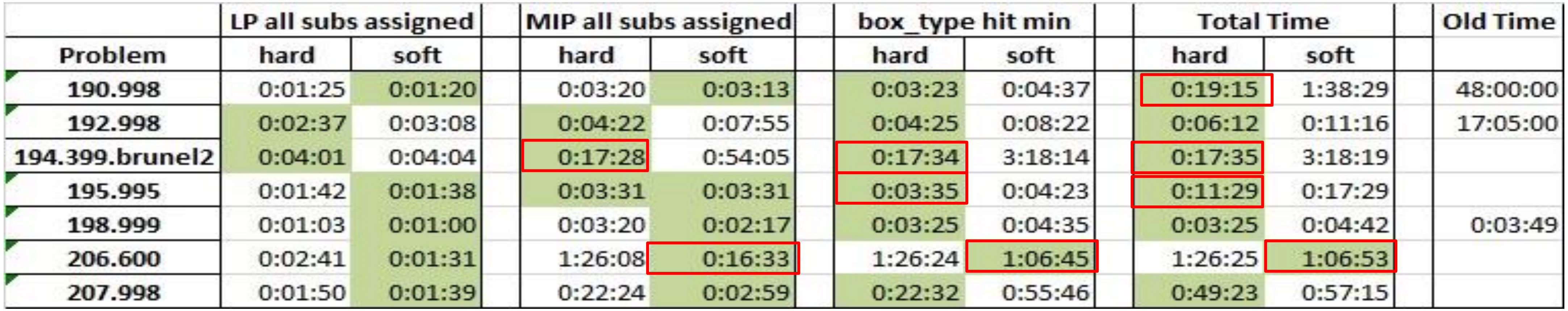
Wall Clock Times in HH:MM:SS

Concurrent MIP



	LP all subs assigned		MIP all subs assigned		box_type hit min		Total Time		Old Time
Problem	hard	soft	hard	soft	hard	soft	hard	soft	
190.998	0:01:23	0:01:26	0:02:37	0:02:32	0:02:40	0:18:49	0:24:50	1:14:24	48:00:00
192.998	0:02:32	0:03:21	0:03:58	0:05:27	0:04:02	0:05:54	0:05:44	0:07:55	17:05:00
194.399.brunel2	0:04:24	0:04:03	0:22:26	0:40:20	0:22:32	1:13:52	0:22:33	1:13:58	
195.995	0:01:49	0:01:49	0:03:54	0:03:30	0:03:59	0:15:52	0:16:19	2:38:44	
198.999	0:01:03	0:00:59	0:03:08	0:02:05	0:03:13	0:04:11	0:03:13	0:04:19	0:03:49
206.600	0:02:40	0:01:32	1:24:45	0:28:02	1:25:01	1:30:44	1:25:03	1:30:53	
207.998	0:01:45	0:01:45	0:19:09	0:02:42	0:19:13	1:10:28	0:44:42	1:11:25	

Default MIP



	LP all subs assigned		MIP all subs assigned		box_type hit min		Total Time		Old Time
Problem	hard	soft	hard	soft	hard	soft	hard	soft	
190.998	0:01:25	0:01:20	0:03:20	0:03:13	0:03:23	0:04:37	0:19:15	1:38:29	48:00:00
192.998	0:02:37	0:03:08	0:04:22	0:07:55	0:04:25	0:08:22	0:06:12	0:11:16	17:05:00
194.399.brunel2	0:04:01	0:04:04	0:17:28	0:54:05	0:17:34	3:18:14	0:17:35	3:18:19	
195.995	0:01:42	0:01:38	0:03:31	0:03:31	0:03:35	0:04:23	0:11:29	0:17:29	
198.999	0:01:03	0:01:00	0:03:20	0:02:17	0:03:25	0:04:35	0:03:25	0:04:42	0:03:49
206.600	0:02:41	0:01:31	1:26:08	0:16:33	1:26:24	1:06:45	1:26:25	1:06:53	
207.998	0:01:50	0:01:39	0:22:24	0:02:59	0:22:32	0:55:46	0:49:23	0:57:15	

Hard or Soft Fastest

Concurrent or Default Fastest

Lessons Learned: FORMULATION MATTERS!!!

- Solvers have gotten faster but the business problems have gotten bigger and harder and expectations can be unrealistic
- Even with the tremendous performance improvements we have seen in solvers in the last several years, a poor formulation can still doom a model and the determination of whether MIP is appropriate
- It is easy to formulate a model; it can be very difficult to formulate it so that it can
  - Handle the scale and complexities of business operations
  - Provide that solution in the required timeframe
  - Provide robust performance for varying data sets

Lessons Learned: Proof of optimality may not be critical

- It is important to understand the objectives in the context of the business problem to determine whether a “good” solution is good enough
- When there are multiple objectives, understand which ones align with feasibility and which ones measure the quality of the solution
- In our problem, two of the three objectives were indicating whether the problem could meet basic solution requirements, and this was critical to know as soon as possible
- The third objective measured the quality of one solution vs. another and was only important once feasibility was determined

Lessons Learned: Miscellaneous

- Curate a representative data set for continual testing
  - Tweaks to the model can help performance for some data sets and hurt it for others so you need to compare against the baseline for multiple data sets before declaring success

- Don’t be afraid to challenge the status quo when it comes to the business problem
  - Sometimes the Business subject matter experts can get used to equating the model to the business constraints. An outside resource can help push the team to consider new angles and think outside the (Birch)box.

Lessons Learned: Leverage the Power of Gurobi MIP

- In many column generation schemes, people build heuristics to generate new columns/patterns
- MIP can do often this just as well
- Use various features of Gurobi
  - Concurrent MIP Optimization
  - Stopping criteria
    - Objective target
    - Best bound
  - Concurrent LP Optimization at the root
    - Barrier is often the winner at the root!

“Life-Altering” Improvements

“The increase in speed and flexibility from the new model impacts every piece of our business across multiple teams. Now we can strategically invest that time in everything from building more personal customer experiences to optimizing our production process. With this new model, Birchbox has truly entered a new operating universe.”

David Bendes

Vice President of Global Business Technology at Birchbox

Thank You – Questions?

GUROBI OPTIMIZATION

The World's Fastest Solver

PRINCETON CONSULTANTS

Information Technology and Management Consulting

BIRCHBOX

Your Next Steps

- Try Gurobi 9.0 Now!
  - Get a 30-day commercial trial license of Gurobi at www.gurobi.com/free-trial
  - Academic and research licenses are free!
- For questions about Gurobi pricing, please contact sales@gurobi.com or sales@gurobi.de
- A recording of this webinar, including the slides, will be available in roughly one week.