

OPTIMIZING PRICING STRATEGY WITH ROYALTY CONSTRAINTS

MUSTAFIF KHAN

1. PROBLEM FORMULATION

Suppose there are n platforms or sales formats (e.g., Ebook, Paperback, Retail), each with:

- A unit price p_i for format i
- A royalty margin $m_i \in [0, 1]$ such that you receive $m_i \cdot p_i$ as earnings per unit

We seek to determine the optimal price vector $\mathbf{p} = [p_1, p_2, \dots, p_n]$ that:

1. Maximizes total royalty revenue per unit:

$$\max_{\mathbf{p}} R(\mathbf{p}) = \sum_{i=1}^n m_i p_i$$

2. Subject to price ordering to reflect increasing value or production cost:

$$p_1 < p_2 < \dots < p_n$$

3. And royalty-per-unit support to reflect preference for higher-margin formats:

$$m_1 p_1 \geq m_2 p_2 \geq \dots \geq m_n p_n$$

4. With bounds:

$$p_i \in [\underline{p}_i, \bar{p}_i] \quad \forall i = 1, \dots, n$$

This is a constrained nonlinear optimization problem that can be solved using numerical methods (e.g., Sequential Least Squares Programming).

2. INTERPRETATION

The resulting prices $\{p_i\}$ ensure:

- Readers save more by choosing digital or direct options
- Your royalty per sale does not decrease as price increases
- The price structure aligns with perceived value and incentivizes support

3. ALGORITHMIC EXTENSIONS FOR PRACTICAL PRICING OPTIMIZATION

To address the real-world challenges of fair and strategic pricing across multiple publishing platforms, we enhance the core optimization model with several algorithmic improvements. These adjustments ensure that the model remains robust, interpretable, and aligned with both business constraints and reader incentives.

3.1. Grouped Margin Constraints.

We group platforms by their royalty margins m_i and enforce ordering only between different groups. This reduces constraint complexity while ensuring that higher-margin platforms receive no less royalty per unit than lower-margin ones.

$$m_i p_i \geq m_j p_j + \varepsilon \quad \text{if } m_i > m_j$$

Here, $\varepsilon \geq 0$ is a small slack variable used to relax the constraint. Only a representative platform from each group is selected to avoid over-constraining the model.

Why this helps:

- Reduces redundant comparisons (e.g., Kobo vs. B&N if both have 70% margin)
- Preserves interpretable tiering of platforms
- Scales better with many distribution channels

3.2. Slack-Relaxed Constraint Design.

Constraints between groups are softened using a margin-specific slack term ε , chosen adaptively depending on the closeness of the margins.

$$m_i p_i - m_j p_j \geq \varepsilon_{ij}$$

This prevents infeasibility in near-equal margin scenarios.

Why this helps:

- Avoids optimization failure when margins are close (e.g., 0.70 vs. 0.68)
- Allows minor trade-offs while preserving intended ranking
- Encourages feasible and realistic solutions

3.3. Penalty and Reward Adjustment in Objective Function.

To strengthen the incentive structure, we define a modified objective function:

$$\max \left[\sum_{i=1}^n m_i p_i + \lambda_r \cdot \text{Reward}(p) - \lambda_p \cdot \text{Penalty}(p) \right]$$

Where:

- $\text{Reward}(p)$: encourages good separation between royalty tiers
- $\text{Penalty}(p)$: discourages near-equal royalties across tiers
- λ_r, λ_p : tunable coefficients controlling influence

Why this helps:

- Avoids scenarios where high-margin formats yield similar or lower royalties
- Promotes meaningful separation in earnings
- Reinforces incentive-aligned pricing structures

3.4. Intra-Group Price Differentiation.

For platforms within the same margin group, we allow soft variability in pricing, while bounding the maximum price difference:

$$|p_i - p_j| \leq \Delta \quad \text{for } i, j \in \text{same group}$$

With $\Delta \approx 5$ dollars.

Why this helps:

- Reflects real-world variation across similarly-royaltied stores
- Prevents artificially flat or uniform pricing
- Provides flexibility to account for user base, store UX, etc.

3.5. Smart Initialization Based on Margin Rank.

The initial guess for pricing is informed by royalty margin rank:

$$p_i^{(0)} = \underline{p}_i + \left(\frac{r(m_i)}{R} \right) (\bar{p}_i - \underline{p}_i)$$

Where $r(m_i)$ is the rank of margin m_i and R is the total number of unique margin levels.

Why this helps:

- Improves convergence of the solver
- Reduces likelihood of landing in poor local minima
- Encourages desirable separation right from initialization

3.6. Constraint Diagnostics and Validation.

The model logs constraint values, satisfaction status, and relative royalty differences between groups to ensure solution validity. This supports both verification and transparency for adoption in academic, open-access, or educational publishing models.

Why this helps:

- Confirms correctness of solution
- Aids interpretability and justification for pricing tiers
- Facilitates deployment in systems with trust or auditing needs

3.7. Summary.

Together, these enhancements ensure that the optimization model:

- Maintains structural fairness (higher margin \Rightarrow higher royalty)
- Encourages readers to support authors while still saving money
- Produces robust, explainable, and incentive-aligned pricing strategies