Non-contiguous satellite channel allocation under cumulative interference: A hybrid price-and-branch approach

Vicky Mak-Hau[a], Angus Gaudry[a]

[a]*School of Information Technology, Deakin University, Victoria, Australia*

Abstract

We study a non-contiguous channelisation problem arising in multi-beam geostationary satellite communications. Carriers have heterogeneous subchannel demands and priorities; carriers within the same beam cannot share a subchannel, while reuse across beams is permitted subject to carrier-specific cumulative-interference budgets. Demand may be partially satisfied, with a priority-weighted objective favouring higher-priority traffic when spectrum or interference constraints are binding.

We formulate the problem as a compact mixed-integer program and derive an equivalent pattern formulation that removes subchannel-permutation symmetry. Building on this formulation, we develop a pattern-based Price-and-Branch algorithm combining randomized Greedy seeding, dual-guided screening of a Greedy-generated pattern pool, and an exact mixed-integer pricing problem, which we show to be NP-hard. We also develop two alternative methods: a Greedy-Seeded Relax-and-Cut algorithm and a Pairwise Co-occurrence and Clique Reconstruction heuristic.

Computational experiments on eight industry-informed synthetic instances and 30 controlled perturbations show that Price-and-Branch returns a feasible solution on all 38 instances and attains the best or tied-best objective on 37. For the 22 instances with certified column-generation convergence, the median certified optimality gap is 0.06%. The results demonstrate robust performance under perturbations of interference tolerances and carrier populations.

Keywords: Integer Programming, Column Generation, Price-and-Branch, Cumulative Interference, Satellite Channel Allocation

1. Introduction

1.1. Multi-Beam Satellite Channelisation under Cumulative Interference

We consider a frequency-assignment problem arising in `SatDreamOpt`, an optimisation-based planning system for multibeam geostationary earth orbit (GEO) satellite communication systems. `SatDreamOpt` decomposes the overall satellite resource-planning problem into a sequence of interdependent combinatorial optimisation problems. The problem studied in this paper occurs after the beam laydown optimisation (BLO) problem has been solved. The BLO determines the locations and sizes of the satellite spot beams, assigns users to beams, and determines a frequency-reuse scheme (Gaudry et al., 2023; Li et al., 2024). We refer to the frequency-reuse classes produced by the BLO as *colours*. For a given colour, multiple beams may share the same spectrum, and each beam contains a collection of carriers with heterogeneous bandwidth demands. Figure 1 illustrates an example frequency-reuse scheme in which multiple beams are assigned the same colour. Each colour defines an independent channelisation problem, since beams belonging to different colours do not generate co-channel interference.

The available spectrum for each beam is discretised into subchannels, and the demand of each carrier is specified as the number of subchannels it requests. These demands are determined by an earlier solution module of `SatDreamOpt` and are treated as input parameters in this study. Demand need not be fully satisfied: a carrier may be allocated any number of subchannels up to its requested demand. Our model does not impose a minimum allocation conditional on serving a carrier. Introducing carrier-specific minimum service levels would instead combine resource allocation with an admission-control decision and would require application-dependent QoS thresholds that are not specified in the problem considered here. Each carrier is also assigned a priority level from 1 (highest) to 5 (lowest). We associate these levels with decreasing allocation weights and maximise the

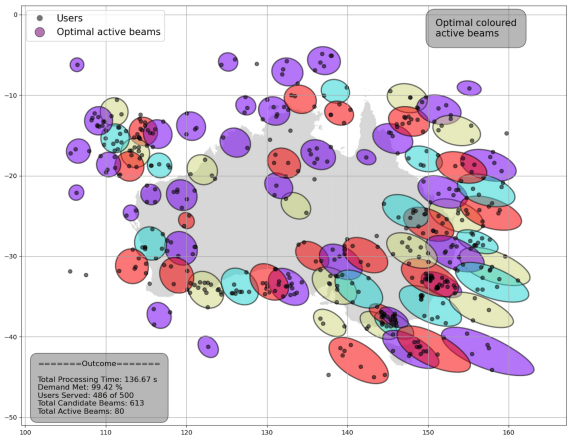


Figure 1: An example beam laydown solution with four frequency-reuse colours. Each colour gives rise to an independent non-contiguous channelisation problem among the beams sharing that colour.

weighted number of assigned subchannels. This objective favours the satisfaction of higher-priority demand while retaining the flexibility to allocate capacity to lower-priority carriers, rather than enforcing a strict lexicographic ordering across priority classes. Motivated by modern digitally channelised systems, we allow the subchannels assigned to a carrier to be non-contiguous.

The resulting problem is a priority-weighted combinatorial allocation problem with packing and cumulative-interference constraints. The optimisation determines which subchannels should be assigned to each carrier and may leave some demand unsatisfied when spectrum availability or interference constraints prevent full allocation. Assignments within a beam must be disjoint, while carriers from different beams may reuse a subchannel only when the total interference remains within all carrier tolerance limits.

This cumulative-interference structure distinguishes the problem from classical graph-colouring and frequency-assignment models based primarily on pairwise conflicts. Although interference contributions are generated by carrier pairs, the feasibility of a subchannel assignment depends on the complete set of carriers simultaneously assigned to that subchannel. A carrier pair may therefore be mutually compatible while their joint inclusion with additional carriers creates an infeasible interference accumulation.

Consequently, the fundamental challenge is not merely assigning subchannels, but identifying jointly interference-feasible sets of carriers that can share a subchannel. This observation motivates a pattern-based representation of the problem, where each subchannel is viewed as a feasible pattern of simultaneously active carriers.

1.2. Related Literature

Future multibeam satellite systems must assign radio resources flexibly according to the traffic demand distribution (Kisseleff et al., 2021). Within `SatDreamOpt`, the channelisation problem studied here is solved after beam laydown, which fixes the position and size of each beam, the users it serves, and its frequency-reuse colour (Gaudry et al., 2023; Li et al., 2024).

Traditional approaches avoid inter-beam interference by not allowing adjacent beams to use the same frequency band (Li et al., 2024). Abdu et al. (2021) solve a carrier assignment problem before optimising power for the active carriers, and Ramírez et al. (2022) solve a mixed-binary quadratic user–carrier assignment within each beam. Kawamoto et al. (2020) use a digital channeliser to allow adjacent beams to share subchannels at cluster boundaries, and the present problem also shares subchannels between beams, since carriers in beams of the same colour may reuse a subchannel subject to cumulative interference budgets.

Non-contiguous allocation is supported in terrestrial OFDMA systems, where a bitmap can assign a user resource block groups that are not adjacent in frequency (3GPP, 2011, 2020), and the present problem adopts non-contiguous subchannel allocation from these systems. Subchannel allocation in OFDMA and NOMA networks is often formulated as an integer or mixed-integer program subject to co-channel interference constraints, with the weighted sum of user data rates a common objective (Mokari et al., 2016; Hadi & Ghazizadeh, 2020). Gaudry & Mak-Hau (2026b) note that these methods do not transfer directly to satellite channelisation because of the cumulative interference structure of flexible payloads.

Within `SatDreamOpt`, Li et al. (2025) develop ALG-MUSIC and ALG-BEAT, which maximise full-bandwidth allocations and weighted subchan-

nel activation, respectively, by permitting partial allocation and relaxing the interference constraints in a two-pass hybrid approach, and the eight industry-informed synthetic instances of Section 6 were also used in that study. Gaudry & Mak-Hau (2026b) introduce the interference-limited channel allocation problem (IL-CAP), which maximises the number of carriers whose total demand is met in a single contiguous block of subchannels, and solve it with integer programming, constraint programming and problem-specific heuristics. The present paper differs from both Gaudry & Mak-Hau (2026b) and Li et al. (2025) by allowing partial demand fulfilment over non-contiguous subchannels under a priority-weighted objective. It further differs from Li et al. (2025) by incorporating cumulative-interference budgets as constraints in a pattern-based master problem, for which exact pricing establishes convergence of the column-generation procedure and hence the LP bound of the full master problem.

The frequency assignment problem allocates frequencies from a limited spectrum to transmitters while avoiding interference (Hale, 1980), and it is classically modelled on an interference graph whose edges join transmitters close enough to interfere (Aardal et al., 2007). For terrestrial networks with cumulative interference, Palpant et al. (2008) combine constraint programming with large neighbourhood search, and Nicholas & Hoffman (2019) use integer programming with lazily generated packing constraints. Kiatmanaroj et al. (2012) assign a single frequency per user in an SDMA satellite system, maximising the number of served users under hard cumulative-interference constraints via Integer Linear Programming and greedy heuristics, without the beam/subchannel structure or multi-subchannel demands considered here. Kiatmanaroj et al. (2016) relax cumulative interference into pairwise disjunctive constraints, an approach that does not guarantee feasibility for the original model.

Column generation has been applied to several frequency and channel assignment problems with structures somewhat related to the problem considered here. (Mehrotra & Trick, 1996) developed a column-generation approach for the classical graph-colouring problem, where each column represents an independent set of vertices that can share a colour. (Jaumard

et al., 2001) considered multi-channel assignment in cellular networks, with the objective of minimising unmet channel demand subject to co-channel, adjacent-channel, and antenna-spacing constraints. They proposed two alternative column-generation formulations, including one in which a column represents a set of cells that can simultaneously reuse a particular channel. (Hemazro et al., 2008) subsequently developed this formulation into a column-generation and branch-and-cut framework for both maximising channel supply and minimising interference. (Koster & Tieves, 2012) considered a related multi-frequency assignment problem for slow frequency-hopping networks, where each transceiver group requires multiple frequencies and column generation is used to determine groups of transceivers that share frequencies while accounting for expected co-channel interference.

The problem considered in this work shares several features with these studies, particularly the allocation of multiple, potentially non-contiguous channels to each user and the possibility of frequency reuse between spatially separated entities. (Jaumard et al., 2001) and (Hemazro et al., 2008) are close in that demand may be only partially satisfied and the objective is related to maximising the amount of allocated spectrum. However, their interference models are predominantly based on pairwise co-channel or adjacent-channel incompatibilities, while (Koster & Tieves, 2012) account for pairwise interference through an aggregate interference objective. In contrast, the present multi-beam satellite problem imposes a hard, carrier-specific cumulative interference budget: the interference received by a carrier from all other carriers sharing its allocated sub-channels is accumulated across the carrier's complete allocation and must remain within its budget.

Gaudry and Mak-Hau (Gaudry & Mak-Hau, 2026a) study non-contiguous channelisation under an all-or-nothing carrier-admission policy, maximising the priority-weighted number of fully served carriers using a pattern-based cooperative portfolio of matheuristics. In contrast, the present paper considers priority-weighted partial demand fulfilment, develops different mathematical formulations, and introduces a distinct suite of algorithms comprising dual-guided screening of a pre-generated Greedy pattern pool before exact pricing, relax-and-cut with minimal-violation-set cuts, and a novel

pairwise co-occurrence and clique reconstruction heuristic.

1.3. Contributions and Paper Organization

The main contributions of this paper are as follows.

1. **Problem formulations.** We formulate the problem as a compact subchannel-indexed MILP and derive a pattern formulation by eliminating the explicit indexing of interchangeable subchannels. We show the integer equivalence of the two formulations.
2. **Column generation with a nontrivial pricing problem.** We derive a pattern-based master problem for column generation, together with an exact pricing formulation, and show that the pricing problem is NP-hard by reduction from Maximum Weight Independent Set.
3. **A hybrid Price-and-Branch algorithm.** We develop a Price-and-Branch method that combines randomized Greedy seeding, dual-guided screening of a Greedy-generated pattern pool, and exact MIP pricing when the pool contains no improving patterns. This reduces reliance on exact pricing while retaining the ability to certify full-master LP bounds when column generation converges.
4. **Alternative algorithms.** We develop two alternative methods: Greedy-Seeded Relax-and-Cut, using dynamically generated interference cuts, and Pairwise Co-occurrence and Clique Reconstruction, combining a pairwise relaxation with clique-based reconstruction.
5. **Computational evaluation on industry-informed instances.** We evaluate the methods on eight industry-informed synthetic instances and 30 controlled perturbations. Price-and-Branch returns feasible solutions on all 38 instances, attains the best or tied-best objective on 37, and achieves a median certified optimality gap of 0.06% on the 22 instances with certified column-generation convergence.

The remainder of the paper is organised as follows. Section 2 presents the Compact and Pattern Formulations and establishes their equivalence. Section 3 introduces the randomized Greedy heuristic. Section 4 develops Price-and-Branch. Section 5 presents the alternative methods. Section 6 reports the computational study, and Section 7 concludes the paper.

2. Problem Formulation

In Table 1 below, we provide a summary of frequently-used notation. Section-specific notation will be introduced in their relevant sections. In

Table 1: Notation for the single-colour subchannel allocation problem.

Symbol	Description
$\mathcal{B}$	Set of beams sharing a common frequency colour.
$\mathcal{C}$	Set of carriers.
C_b	Set of carriers belonging to beam $b \in \mathcal{B}$.
T	Number of available subchannels in each beam.
$\mathcal{T}$	Index set of subchannels, i.e., $\mathcal{T} = \{1, \ldots, T\}$.
beam(i)	Beam to which carrier i belongs.
d_i	Demand of carrier i as number of subchannels.
Γ_i	Interference budget of carrier i. ($\Gamma_i > 0$ for all carriers).
φ_i	Priority level of carrier i (1 highest, 5 lowest).
p_{ji}	Interfering power induced by carrier j on carrier i share a when they subchannel. (In general, $p_{ij} \neq p_{ji}$)
v_i	Objective weight of carrier i.

Figure 1, we can see beam footprints in different colours. Beams sharing the same colour will share the same frequency spectrum, and with polarisation, the number of colours may double. In any case, each distinct colour and polarisation gives us a mathematical optimisation problem instance.

A problem instance is defined by the set of beams $\mathcal{B}$, the set of carriers allocated to each beam C_b, the demand of a carrier d_i (in terms of number of subchannels), and interference tolerance budget Γ_i, and the interfering power p_{ij} between each pair of carriers i, j for each sub-channel they share.

A *feasible allocation* assigns, for each subchannel t, at most one carrier per beam, such that every carrier i receives no more than d_i subchannels in total, and the cumulative interference it receives across all of its allocated subchannels does not exceed Γ_i. We refer to the composition of a single subchannel – the set of carriers simultaneously placed in it (at most one per beam) – as a *pattern*. A feasible allocation is exactly a multiset of at most T patterns, one per used subchannel.

2.1. Compact Subchannel-Indexed Formulation

Let:

- $x_{it} \in \{0,1\}$ be a binary decision variable with $x_{it} = 1$ indicating Carrier i is allocated to Subchannel t, and 0 otherwise;
- $v_i = 6 - \varphi_i$ the "value" (weight) of Carrier i in the objective function;
- $\mathcal{I}_{it} \geq 0$ an auxiliary variable to bound from above the interference received by carrier i if it is allocated to Subchannel t.

The mixed integer linear programming (MILP) formulation is given as follows.

Model 1. *Compact Formulation*

$$\max \quad \sum_{i \in \mathcal{C}} \sum_{t=1}^{T} v_i \, x_{it} \tag{1}$$

$$s.t. \quad \sum_{i \in C_b} x_{it} \;\leq\; 1, \qquad \forall b \in \mathcal{B},\ \forall t \in \mathcal{T} \tag{2}$$

$$\sum_{t=1}^{T} x_{it} \;\leq\; d_i, \qquad \forall i \in \mathcal{C} \tag{3}$$

$$\sum_{j \in \mathcal{C} \,:\, \text{beam}(j) \neq \text{beam}(i)} p_{ji} \, (x_{it} + x_{jt} - 1) \;\leq\; \mathcal{I}_{it}, \quad \forall i \in \mathcal{C},\ \forall t \in \mathcal{T} \tag{4}$$

$$\sum_{t=1}^{T} \mathcal{I}_{it} \;\leq\; \Gamma_i, \qquad \forall i \in \mathcal{C} \tag{5}$$

$$x_{it} \in \{0,1\}, \qquad \forall i \in \mathcal{C},\ \forall t \in \mathcal{T} \tag{6}$$

$$\mathcal{I}_{it} \geq 0 \qquad \forall i \in \mathcal{C},\ \forall t \in \mathcal{T} \tag{7}$$

The objective maximises the priority-weighted number of allocated subchannels. Constraints (2) and (3) enforce beam exclusivity and carrier demand limits, respectively. Constraints (4)–(5) impose each carrier's cumulative-interference budget: when carrier i is assigned to subchannel t, I_{it} bounds the interference received from co-channel carriers, and these contributions are accumulated over all subchannels.

Numerical stabilisation. Although the aggregated interference constraints are exact for binary assignments, their large implicit big-M coefficients can amplify numerical errors arising from finite integrality tolerances. For computation, we therefore use an integer-equivalent reformulation that caps interference coefficients, tightens big-M using beam exclusivity, and normalises the interference budgets. This preserves the feasible integer allocations without introducing additional variables or constraints. The reformulation, its justification, and computational comparisons with alternative linearisations are provided in the Supplementary Material.

2.2. Pattern Formulation

Given that sub-channel allocations to carriers need not be contiguous, the ordering of sub-channels is immaterial. We therefore introduce the notion of a pattern and present a pattern-based formulation.

Definition 1. *A pattern $p = (c_b)_{b\in\mathcal{B}}$ specifies the carrier assigned to each beam, where $c_b \in C_b \cup \{0\}$ for every $b \in \mathcal{B}$. The value 0 indicates that no carrier is assigned to beam b. We denote by*

$$V(p) = \{c_b : b \in \mathcal{B},\ c_b \neq 0\}$$

the set of carriers contained in pattern p.

Let $\mathcal{P}$ be the set of all patterns, and any $y_p \in \mathbb{Z}_{\geq 0}$, for $p \in \mathcal{P}$, the multiplicity of pattern p in a solution.

Model 2. *Pattern Formulation*

$$\max \quad \sum_{p\in\mathcal{P}} \Big(\sum_{i\in V(p)} v_i \Big) y_p \tag{8}$$

$$s.t. \quad \sum_{p\in\mathcal{P}\,:\,i\in V(p)} y_p \leq d_i \qquad \forall i \in \mathcal{C} \tag{9}$$

$$\sum_{p\in\mathcal{P}\,:\,i\in V(p)} \left(\sum_{j\in V(p),\, j\neq i} p_{ji} \right) y_p \leq \Gamma_i \qquad \forall i \in \mathcal{C} \tag{10}$$

$$\sum_{p\in\mathcal{P}} y_p \leq T \tag{11}$$

$$y_p \in \mathbb{Z}_{\geq 0} \qquad \forall p \in \mathcal{P}. \tag{12}$$

The objective function (8) maximises the total value of the selected patterns. Constraint (9) ensures that the allocation of each carrier does not exceed its demand. Constraint (10) limits the cumulative interference received by each carrier i to its interference threshold Γ_i. Finally, Constraint (11) ensures that the total number of selected patterns, corresponding to allocated sub-channels, does not exceed the available number of sub-channels T.

2.3. Equivalence of the Compact and Pattern Formulations

Proposition 1. *The Compact Formulation 1 and the Pattern Formulation 2 are equivalent at the integer level. More precisely, every feasible integer solution of the Compact Formulation induces a feasible integer solution of the Pattern Formulation with the same carrier allocations and objective value, and every feasible integer solution of the Pattern Formulation can be expanded into a feasible integer solution of the Compact Formulation with the same objective value. Hence, the two formulations have the same optimal integer objective value.*

The equivalence is established by aggregating each feasible compact solution into pattern multiplicities and, conversely, expanding any feasible integer pattern solution onto distinct subchannels; both transformations preserve feasibility and objective value. The full proof is provided in the Supplementary Material.

3. Common Construction Heuristic

This section introduces a common randomized construction heuristic used both as a standalone solution method and as a supporting component within the algorithms developed later in the paper. The heuristic generates feasible subchannel allocations quickly and provides useful initial solutions and candidate patterns for subsequent optimisation procedures.

3.1. Adaptive Score-Weighted Randomized Greedy Heuristic

The Adaptive Score-Weighted Randomized Greedy Heuristic (henceforth referred to as `Greedy`) is a construction heuristic that provides warm starts and primal lower bounds for some of the methods presented in this section.

`Greedy` fills the T subchannels one at a time; within each subchannel it visits every beam in turn and attempts to place one of that beam's carriers with unmet demand, via a score-weighted random draw. Let:

- A be the set of carriers already placed in the current subchannel;
- a_i be the number of subchannels assigned to Carrier i so far;
- score(i) be the selection score of Carrier i;
- $u_i = d_i - a_i$ be the remaining unallocated demand of Carrier i; and
- N be the number of independent randomised `Greedy` runs.

Candidate score. Consider the candidate pool for a Beam b of subchannel t given as $\{i \in C_b : a_i < d_i\}$. For a candidate carrier i being considered for the current subchannel t, given the carriers A already accepted into t from earlier beams in this pass, the score for Carrier i is calculated as:

$$\text{score}(i) = \underbrace{\frac{1}{\varphi_i}}_{\text{priority}} \times \underbrace{\frac{u_i}{d_i}}_{\text{unmet demand fraction}} \times \underbrace{\eta_i(A)}_{\text{interference impact}}, \tag{13}$$

$$\text{for} \quad \eta_i(A) = \begin{cases} 1, & \text{if } \max\limits_{j \in C_b \,:\, u_j > 0} \sum\limits_{k \in A} p_{jk} = 0, \\ 1 - (1-\varepsilon)\dfrac{\sum_{k \in A} p_{ik}}{\max\limits_{j \in C_b \,:\, u_j > 0} \sum\limits_{k \in A} p_{jk}}, & \text{otherwise.} \end{cases} \tag{14}$$

for $\varepsilon = 0.1$. The candidate score is realised as a weighted random draw via Python's `random.choices`, which accepts arbitrary non-negative relative weights and performs the proportional sampling internally, provided they do not all sum to zero. When every candidate's score is zero, we instead sample uniformly from the candidate pool. The interference term equals 1 whenever the candidate i induces zero interference on the carriers already accepted into the current subchannel; this holds trivially when $A = \emptyset$. Overall, this score function favours carriers with higher priority, carriers with greater unmet

demand, and carriers that would induce the least additional interference on those already placed in the subchannel.

A sampled carrier is accepted only if its addition preserves the cumulative interference budget of every carrier already placed in the current subchannel, as well as its own budget. Full implementation details and pseudocode are provided in the Supplementary Material. Since carrier selection is randomised, independent runs (different seeds) generally produce different total allocations. Every downstream method that uses Greedy as a subroutine therefore runs it N times (we set $N = 100$ by default) and keeps the run with the largest objective value.

4. Pattern-Based Price-and-Branch

4.1. The Master Problem

The equivalence established in Section 2.3 allows the explicit subchannel indices in the Compact Formulation to be replaced by pattern counts. The resulting Pattern Formulation can be viewed as an aggregated Dantzig–Wolfe reformulation: because feasibility and objective value are invariant to permutations of subchannel indices, subchannels with the same carrier-allocation pattern can be represented by a single pattern-count variable. The master constraints then enforce carrier demand (3) and cumulative interference (5) limits across the selected patterns.

Recall that $\mathcal{P}$ denotes the set of all admissible patterns and $V(p)$ the set of carriers contained in pattern $p \in \mathcal{P}$. We introduce new definitions.

Table 2: New parameters.

Symbol	Description
$c_p := \sum_{i \in V(p)} v_i$	the value of Pattern p
a_{ip}	with $a_{ip} = 1$ if $i \in V(p)$ and $a_{ip} = 0$ otherwise
g_{ip}	with $g_{ip} = \sum_{j \in V(p),\ j \neq i} p_{ji}$ if $i \in V(p)$, $g_{ip} = 0$ otherwise

Recall that $y_p \in \mathbb{Z}_{\geq 0}$ is the number of subchannels pattern p occupies. Model 2 is rewritten as follows for convenience of column generation.

Model 3. *The Master Problem*

$$\max \quad \sum_{p\in\mathcal{P}} c_p y_p \tag{15}$$

$$s.t. \quad \sum_{p\in\mathcal{P}} a_{ip} y_p \leq d_i \qquad \forall i \in C, \qquad [\pi_i] \tag{16}$$

$$\sum_{p\in\mathcal{P}} g_{ip} y_p \leq \Gamma_i \qquad \forall i \in C, \qquad [\mu_i] \tag{17}$$

$$\sum_{p\in\mathcal{P}} y_p \leq T, \qquad [\rho] \tag{18}$$

$$y_p \in \mathbb{Z}_{\geq 0} \qquad \forall p \in \mathcal{P}. \tag{19}$$

4.2. The Column Generation Pricing Subproblem

The column generation pricing subproblem is as below. Let:

- (π, μ, ρ) be the dual prices from the LP relaxation of Model 3;
- $x_i \in \{0, 1\}$, for each carrier $i \in \mathcal{C}$, with $x_i = 1$ indicating Carrier i is included in the pattern, and $x_i = 0$ otherwise; and
- $z_{ij} \in [0, 1]$, for each pair of carriers (i, j) with $\text{beam}(i) < \text{beam}(j)$, be the product $x_i x_j$.

The objective function is given as:

$$\max \sum_{i\in\mathcal{C}} (v_i - \pi_i) x_i - \sum_{\substack{i,j\in\mathcal{C} \\ \text{beam}(i)<\text{beam}(j)}} \big(\mu_i p_{ji} + \mu_j p_{ij}\big) z_{ij}. \tag{20}$$

Notice that the constant dual term $-\rho$ is omitted from the optimisation objective but is subtracted afterwards. Suppose that the optimal value of (20) is q^*, the reduced cost of the generated pattern will therefore be

$$\bar{c} = q^* - \rho, \tag{21}$$

and an improving column exists when $q^* - \rho > 0$. Again, we have the beam exclusivity constraints to ensure that no more than 1 carrier is allocated to

each beam in the pattern.

$$\sum_{i \in C_b} x_i \leq 1 \qquad \forall b \in \mathcal{B} \tag{22}$$

As z_{ij} is to capture the quadratic term $x_i x_j$, for each pair (i, j) with beam$(i) <$ beam(j), we have the McCormick linearisation constraints:

$$z_{ij} \leq x_i, \tag{23}$$

$$z_{ij} \leq x_j, \tag{24}$$

$$z_{ij} \geq x_i + x_j - 1 \tag{25}$$

$$x_i \in \{0, 1\}, z_{ij} \geq 0. \tag{26}$$

As interference budget is ensured in the Master Problem, we did not add any interference budget violation elimination constraints here. Therefore, it is possible that a pattern generated this way can have some carriers with total interference in the pattern exceeding their interference budgets.

Proposition 2. *The exact pricing subproblem described in Section 4.2 is NP-hard in general.*

The NP-hardness result follows by a polynomial-time reduction from the Maximum Weight Independent Set (MWIS) problem. For an arbitrary MWIS instance, we construct a corresponding pricing instance with one carrier for each vertex, vertex weights represented by the linear terms in the pricing objective, and pairwise penalties set to a sufficiently large value for edges of the MWIS graph and to zero for nonedges. The large penalties ensure that no optimal pricing solution selects both endpoints of an MWIS edge, so that the optimal pricing solution corresponds exactly to a maximum-weight independent set. The full construction and proof are provided in the Supplementary Material. Since the exact pricing subproblem is NP-hard, we use a two-tiered pricing strategy to reduce the per-iteration computational cost, particularly for large instances.

4.2.1. Dual-Guided Pool Screening: The Heuristic Pricer

We employ dual-guided screening of a Greedy-generated pattern pool to reduce the frequency of exact pricing calls. For fixed master dual prices, the reduced cost $\bar{c}(p)$ of any known pattern p can be evaluated in closed form, allowing candidate patterns to be re-scored efficiently as the dual prices change. The N `Greedy` runs are ranked by objective value. The best run is used to seed the initial master (Section 4.1), while patterns from the next K runs are retained in a candidate pool $\mathcal{H}$ outside the master, excluding duplicates and patterns already in the master. Each run of `Greedy` solution contributes to at most T distinct patterns. At each column-generation iteration, all remaining patterns in $\mathcal{H}$ are re-scored using the current dual prices, and the pattern with the largest reduced cost is admitted if $\bar{c} > \varepsilon$. An admitted pattern is removed from $\mathcal{H}$, and the pool is not replenished.

4.2.2. Exact Pricer

If no pattern in $\mathcal{H}$ is improving, the exact pricer given below, is called.

Model 4. *Pricing subproblem* (20)*, subject to* (22)–(26)

Benefit of seeding with best-of-N Greedy solution.. An initial pool of singletons alone induces a highly degenerate master LP — every singleton column has identical budget usage (of 1) and identical, all-zero interference consumption (as there is no other carriers in the column), and, within each priority tier, identical cost as well. Seeding with a real, already-feasible best-of-N Greedy solution both improves the starting solution and reduces this degeneracy behaviour.

4.3. Price-and-Branch Algorithm

Given that the pricing subproblem is NP-hard, instead of implementing a full branch-and-price, we implement an integer solve (through a solver's standard branch-and-bound) after the exact-pricer termination proves the full master LP optimal or pricing time budget is reached.

If the exact pricing MIP returns an OPTIMAL status and the maximum reduced cost does not exceed ε, column-generation convergence is certified

Algorithm 1 Column generation with heuristic pool pricer

Run Greedy N times and rank the solutions by objective value
$\mathcal{P}' \leftarrow$ singleton patterns plus patterns from the best Greedy run
$\mathcal{H} \leftarrow$ patterns from Greedy runs $2, \ldots, K+1$, excluding patterns already in $\mathcal{P}'$
repeat
    solve LP relaxation of the Master Problem over $\mathcal{P}'$; read duals (π, μ, ρ)
    $(p^*, \bar{c}) \leftarrow \arg\max_{p \in \mathcal{H}} \bar{c}(p)$ if $\mathcal{H}$ not empty ▷ heuristic pool pricer, §4.2.1
    if $\mathcal{H} = \emptyset$ or $\bar{c} \leq \varepsilon$ **then**
        $(p^*, \bar{c}) \leftarrow$ solve the exact pricing subproblem given (π, μ, ρ) ▷ §4.2.2
    else
        $\mathcal{H} \leftarrow \mathcal{H} \setminus \{p^*\}$
    if $\bar{c} > \varepsilon$ **then**
        $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{p^*\}$
until $\bar{c} \leq \varepsilon$ or a time/iteration limit is reached
integer solve of the Master Problem over $\mathcal{P}'$ ▷ final solution

within the specified numerical tolerances. In this case, the restricted-master LP objective provides a numerical bound on the full-master LP optimum, subject to these tolerances. Otherwise, including termination due to computational time limits, the restricted-master LP objective does not constitute a certified upper bound on the original problem. When pricing is terminated, we implement a final integer solve of the Master Problem over $\mathcal{P}'$ – this approach is commonly known as *price-and-branch*: optimising over the columns generated at the root, without further pricing within the branch-and-bound tree. This approach does not however provide the global optimality guarantee of full branch-and-price.

4.4. Pricer Calls in Price-and-Branch

Figure 2 illustrates the interaction between the heuristic pool pricer and exact pricing for two representative instances. Colour 1 obtains most of its columns from the heuristic pool but reaches the computational limit without certified convergence, whereas Colour 7 uses exact pricing much more

extensively and ultimately certifies column-generation convergence. Price-

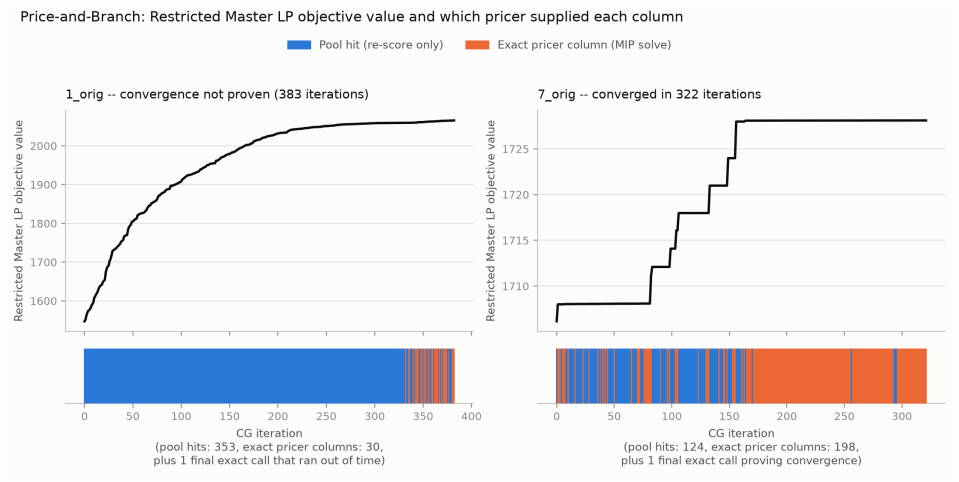


Figure 2: Column generation on the original instances: Colour 1 and Colour 7. The black line is the restricted Master LP objective value after each iteration. The coloured strips are the pricer that supplied the iteration's column: with blue strips representing a heuristic pricer call (Pool Hit) and orange strips an exact pricer call. Colour 1 is time-capped at 383 iterations without a proof of convergence; Colour 7 converges in 322 iterations.

and-Branch uses GLOP for the restricted-master LP and HiGHS for exact pricing and the final integer solve; implementation details and solver-specific safeguards are given in the Supplementary Material.

5. Alternative Methods

5.1. Greedy-Seeded Relax-and-Cut

The Greedy-Seeded Relax-and-Cut (`GSRC`) method starts from the Compact Formulation with interference constraints (4)–(7) removed then iteratively adds valid cuts until an interference-feasible allocation is obtained or the time limit is reached. In the unseeded case, the relaxed optimum is an upper bound on the original problem.

Analytical upper bound. The relaxed optimum of the unseeded case can be decomposed by beams, and computed analytically without a solver. For each beam b, order its carriers as $i_1, \ldots, i_{|C_b|}$ such that $v_{i_1} \geq v_{i_2} \geq \cdots \geq v_{i_{|C_b|}}$.

Allocate subchannels in this order, satisfying each carrier's demand up to the remaining capacity. Specifically,

$$a_{i_r}^{UB} = \min\left\{ d_{i_r}, \left(T - \sum_{s=1}^{r-1} d_{i_s} \right)^+ \right\}.$$

The resulting upper bound is

$$UB_{\text{GSRC}} = \sum_{b \in B} \sum_{r=1}^{|C_b|} v_{i_r} a_{i_r}^{UB}.$$

Seeding. We seed `GSRC` with the best of the N Greedy solutions by fixing its carrier–subchannel assignments, i.e., setting the corresponding $x_{it} = 1$. The interference-relaxed model may introduce additional assignments to improve the objective, potentially creating interference violations. We then iteratively add cuts to eliminate these violations.

Repair. In each iteration of the repair loop, we solve the (increasingly constrained) relaxed master. For each Carrier k currently with interference exceeding its budget, we generate two tiers of cuts:

Tier-1: Pattern ban. For a subchannel containing Carrier k, rank the other co-located carriers in non-increasing order of their interfering power on k, giving $i_1, i_2, \ldots$. Let m be the smallest index such that $\sum_{r=1}^{m} p_{i_r k} > \Gamma_k$. Then, $\sum_{r=1}^{m-1} p_{i_r k} \leq \Gamma_k$ and $\mathcal{S} = \{k, i_1, \ldots, i_m\}$ forms a minimal violation set (MVS). A constraint in the following form can be generated to eliminate such a interference budget violation:

$$\sum_{i \in \mathcal{S}} x_{it} \leq |\mathcal{S}| - 1 \qquad \forall t. \tag{27}$$

Since every subchannel is structurally interchangeable, the same violating carrier combination will not appear in future iterations.

Tier-2: No-good cut. If Tier 1 does not identify a violating carrier combination within a single subchannel for a carrier k whose cumulative interference exceeds Γ_k, we consider the individual interference contributions

received by k across all subchannels to which it is assigned.

Let T_k denote the set of subchannels currently assigned to Carrier k, and let A_t denote the set of carriers assigned to subchannel t. We form the collection $\mathcal{C}_k = \{(j, t, p_{jk}) : t \in T_k,\ j \in A_t \setminus \{k\}\}$, and order its elements in non-increasing order of p_{jk}. Let $(j_1, t_1), \ldots, (j_\ell, t_\ell)$ be the shortest prefix of this ordering for which $\sum_{r=1}^{\ell} p_{j_r k} > \Gamma_k$. Let $S_k = \{t_r : r = 1, \ldots, \ell\}$ be the distinct subchannels appearing in this prefix, and define the corresponding witness set: $\mathcal{W}_k = \{(j_r, t_r) : r = 1, \ldots, \ell\} \cup \{(k, t) : t \in S_k\}$. We then add the no-good cut

$$\sum_{(i,t)\in\mathcal{W}_k} x_{it} \leq |\mathcal{W}_k| - 1. \tag{28}$$

To assess the effect of Greedy seeding, we compare seeded and unseeded `GSRC` on two representative instances, `7_orig` and `6_sub80`. The seeded runs obtain feasible solutions in 14.1 and 41.6 seconds, respectively, whereas neither corresponding unseeded run reaches feasibility within the 600-second computational budget.

Figure 3 illustrates the repair dynamics of `GSRC` on two contrasting original instances, Colours 1 and 7. Colour 7 reaches feasibility after 33 repair iterations using almost exclusively Tier-1 cuts, whereas Colour 1 reaches the 600-second limit after 145 iterations with carriers still violating their interference budgets and increasing reliance on Tier-2 cuts.

5.2. Pairwise Co-occurrence and Clique Reconstruction

Pairwise Co-occurrence and Clique Reconstruction (`PCCR`) is a two-stage heuristic. Stage 1 solves an integer relaxation in terms of pairwise carrier co-occurrences, and Stage 2 reconstructs these pairwise fragments into feasible subchannel patterns using cliques.

5.2.1. Stage 1: Pairwise Relaxation

Let Π be the set of unordered pairs of carriers from different beams, and let $w_{ij} \in \mathbb{Z}_{\geq 0}$ denote the number of subchannels in which pair $(i, j) \in \Pi$ occurs. Recall that C_b denote the set of carriers associated with beam b. The pairwise relaxation is:

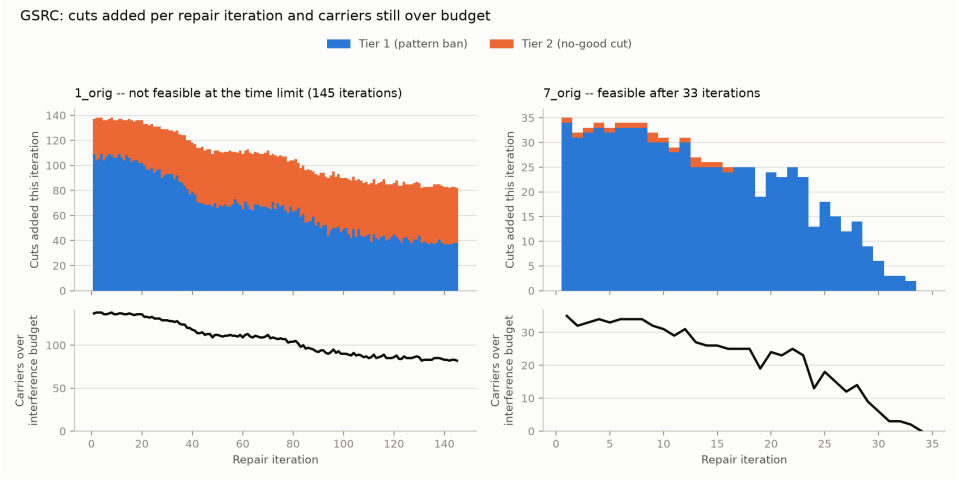


Figure 3: Tier-1 (pattern-ban) and Tier-2 (no-good) cuts added per iteration (stacked bars, left axis) and the number of carriers exceeding their interference budget (line, right axis) for Colour 1 (left) and Colour 7 (right).

Model 5. *Pairwise Co-occurrence Relaxation*

$$\max \quad \sum_{\substack{i,j\in\mathcal{C}\\ \text{beam}(i)<\text{beam}(j)}} w_{ij} \tag{29}$$

$$s.t. \quad \sum_{j\in C_{b'}} w_{ij} \le d_i \qquad \forall i \in \mathcal{C},\ \forall b' \neq \text{beam}(i), \tag{30}$$

$$\sum_{\substack{j\in\mathcal{C}\\ \text{beam}(i)\neq\text{beam}(j)}} p_{ji} w_{ij} \le \Gamma_i \qquad \forall i \in \mathcal{C}, \tag{31}$$

$$w_{ij} \in \mathbb{Z}_{\ge 0} \qquad \forall i,j \in \mathcal{C} : \text{beam}(i) < \text{beam}(j). \tag{32}$$

Constraint (30) imposes a demand cap for each carrier, imposed separately for each "partner" beam b', rather than aggregated across all other beams. The latter may induce infeasibility as multiple carrier pairs involving i may belong to the same subchannel. Constraint (31) enforces its cumulative-interference budget. The objective function (29) maximises the total number of pairwise carrier co-occurrences as a surrogate for maximising allocation. The variables w_{ij} record only pairwise co-occurrence counts and do not enforce consistency among different pairs. Consequently, a feasible solution to

the PCCR need not correspond to any feasible set of subchannel patterns.

5.2.2. Stage 2: Clique-Based Reconstruction

Let w^* be an optimal solution to Stage 1, we build an undirected, edge-weighted graph $G = (\mathcal{C}, E)$ with $E = \{(i,j) \in \Pi : w^*_{ij} > 0\}$, weighted by w^*_{ij}. Recall that (i,j) is unordered. Since every edge joins carriers from different beams, every clique of G satisfies beam exclusivity and therefore defines a candidate subchannel pattern. Let $\mathcal{K}$ denote the set of all nonempty cliques in G, including singleton cliques $\{i\}$ for every $i \in C$. A singleton clique represents a subchannel allocated exclusively to one carrier and therefore contributes no pairwise interference. Including singletons allows the reconstruction model to allocate otherwise unused subchannels to carriers with unmet demand.

Let $x_c \in \mathbb{Z}_{\geq 0}$ denote the multiplicity of Clique $c \in \mathcal{K}$. The Stage 2 reconstruction model is:

Model 6. *Clique Reconstruction*

$$\max \quad \sum_{c \in \mathcal{K}} \left(\sum_{i \in c} v_i \right) x_c \tag{33}$$

$$s.t. \quad \sum_{c \in \mathcal{K}} x_c \leq T \tag{34}$$

$$\sum_{i \in c,\ c \in \mathcal{K}} x_c \leq d_i \qquad \forall i \in \mathcal{C}, \tag{35}$$

$$\sum_{\{i,j\} \subseteq c,\ c \in \mathcal{K}} x_c \leq w^*_{ij} \qquad \forall (i,j) \in E, \tag{36}$$

$$x_c \in \mathbb{Z}_{\geq 0} \qquad \forall c \in \mathcal{K}. \tag{37}$$

Constraint (36) links the reconstruction to Stage 1 by preventing any carrier pair from occurring more often than prescribed by w^*_{ij}; Constraints (34) and (35) enforce the subchannel and demand limits. Given that the reconstructed pairwise co-occurrences never exceed w^*_{ij}, the cumulative-interference

feasibility is also guaranteed:

$$\sum_{\substack{j\in\mathcal{C}\\ \text{beam}(i)\neq\text{beam}(j)}} p_{ji}\left(\sum_{\substack{c\in\mathcal{K}\\ \{i,j\}\subseteq c}} x_c\right) \leq \sum_{\substack{j\in\mathcal{C}\\ \text{beam}(i)\neq\text{beam}(j)}} p_{ji}w^*_{ij} \leq \Gamma_i, \qquad \forall i \in \mathcal{C}. \tag{38}$$

Limitation. Although singleton patterns are permitted, multi-carrier patterns can only be constructed from cliques in the support graph induced by the Stage 1 solution. Carrier combinations containing pairs not represented by this graph cannot be recovered. Moreover, the Stage 1 surrogate objective rewards pairwise co-occurrences without accounting for the feasibility of assembling those pairs into valuable multi-carrier patterns. These restrictions may limit the quality of the reconstructed solution. These limitations motivate the pattern-based column-generation approach of Section 4.

6. Computational Study

6.1. Problem Instances and Experimental Setup

We consider 38 test instances in total. The eight original instances were obtained from the Beam Laydown Optimisation module of `SatDreamOpt` and were also used in Li et al. (2025): their beams, carriers, demands, priorities, interference tolerances, and pairwise interference values p_{ji} are therefore derived from industry-informed synthetic data.

Table 3 summarises the eight original instances. Here, $|\mathcal{B}|$ and $|\mathcal{C}|$ denote the numbers of beams and carriers, $\sum_i d_i$ is total requested subchannel demand, and all instances have $T = 200$ available subchannels. We also report the percentage of ordered carrier pairs with nonzero interference and the median and 90th percentile of p_{ji}/Γ_i.

The distributions of nonzero pairwise co-channel interference across the eight original instances are reported in Supplementary Figure S1

To assess robustness beyond the eight original cases, we derive 30 additional instances using controlled perturbations of interference tolerances and carrier populations:

Table 3: Characteristics of the eight operational colour instances. All instances have $T = 200$ available subchannels.

Colour	$\lvert\mathcal{B}\rvert$	$\lvert\mathcal{C}\rvert$	$\sum_i d_i$	Nonzero-interference density (%)	Median p_{ji}/Γ_i	90th pct. p_{ji}/Γ_i
1	8	206	1130	83.1	4.29×10^{-2}	1.43
2	6	170	916	82.2	1.17×10^{-2}	5.32×10^{-1}
3	8	92	470	71.9	3.49×10^{-4}	4.02×10^{-2}
4	6	96	636	74.5	1.74×10^{-3}	8.96×10^{-2}
5	5	75	501	70.6	2.69×10^{-4}	3.03×10^{-2}
6	5	123	647	74.4	3.74×10^{-3}	2.92×10^{-1}
7	5	94	691	73.9	2.91×10^{-3}	2.83×10^{2}
8	7	144	882	80.8	1.16×10^{-2}	6.51×10^{-1}

- **Tolerance perturbation.** Every carrier's interference tolerance Γ_i is generated independently from a uniform distribution as $\Gamma_i \cdot U(1 - r, 1 + r)$ for $r \in \{0.10, 0.20\}$, while leaving every p_{ij} untouched. As p_{ij} is derived upstream from each carrier's antenna gain pattern and the transmitting/receiving beams' relative geometry – that is why it is directional ($p_{ij} \neq p_{ji}$) – so perturbing it directly would fabricate numbers with no physical grounding. Tolerance, by contrast, is the link margin obtained from an other module of `SatDreamOpt`, not a physical measurement, having a different value is legitimate. Tolerance perturbation is applied to all eight operational colours at both magnitudes of r described above: this gives us 16 new instances.

- **Carrier subsampling.** A random fraction of each beam's carriers is sampled to be kept, stratified by beam, and applied at 80% and 90% retention rate to the five instances with the heaviest interference load (colours 1, 2, 6, 7, 8 – identified from the p_{ij} distributions in Supplementary Figure S1 and at 85% to the remaining three: this gives us 13 new instances.

- **Combined.** One instance (the single hardest operational colour, 1) carriers subsampled at 85% and tolerance-perturbed at ±15%: this gives us 1 new instance.

6.2. Computer Setup

The Python implementations were developed with assistance from Claude Code (Anthropic), under the authors' direction. The authors specified the mathematical formulations and algorithmic procedures, reviewed the implementations, and independently validated the computational results. All computations were run sequentially on a single Apple M5 (10-core) machine with 24 GB RAM under macOS 27. The optimisation models were implemented in Python 3.14.7 using Google OR-Tools 9.15.6755 through its `MPSolver` interface. HiGHS 1.12.0 was used for the Compact Formulation, the `GSRC` models, both stages of `PCCR`, and Price-and-Branch's exact pricing and final integer solve. The restricted master LP in Price-and-Branch was solved with GLOP because reliable dual values are required for column generation; solver-specific implementation details are provided in the Supplementary Material.

Each instance–method combination was run separately, so the reported wall-clock times are unaffected by concurrent resource contention. A 600-second time limit was imposed on each method, except for the `GSRC` upper bound, which requires only a single LP-relaxation solve. All optimisation backends used in the study are open source, allowing the computational implementation to be reproduced without proprietary solver licences. The experiments therefore use a common open-source optimisation environment, with GLOP used only for the Price-and-Branch restricted-master LP.

6.3. Methods

Five methods are compared throughout this section:

- **Greedy** (Section 3.1): the best of $N = 100$ independent runs.
- **Greedy-seeded Relax-and-Cut** (`GSRC`, Section 5.1): an interference-relaxed Compact Formulation with dynamically generated cuts, seeded by the best Greedy solution. The corresponding unseeded relaxation provides an upper bound.

- **Pairwise Co-occurrence and Clique Reconstruction** (`PCCR`, Section 5.2): a two-stage heuristic combining a pairwise relaxation with clique-based reconstruction of feasible subchannel patterns.
- **Pattern-Based Price-and-Branch** (Section 4): column generation over the implicit pattern universe, with heuristic pool pricing before invoking the exact MIP pricer when required.
- **Compact Formulation** (Section 2.1): Model 1 solved directly using the numerically stabilised, normalised formulation described in the Supplementary Material, without decomposition or a warm start.

All experiments use the priority-weighted objective, $N = 100$ Greedy runs, and pricing-pool depth $K = 20$. Every reported solution was independently checked by a common validator for beam exclusivity, demand, and cumulative-interference feasibility.

6.4. Overall Computational Performance

To compare solution quality across instances with different objective scales, let z_{mi} denote the objective value obtained by method m on instance i, and define the best feasible objective found as

$$z_i^{\text{best}} = \max_m \{z_{mi} : \text{method } m \text{ returns a feasible solution on instance } i\}.$$

For every feasible method–instance pair, let: $\delta_{mi} = 100(z_i^{\text{best}} - z_{mi})/z_i^{\text{best}}$. In this way, $\delta_{mi} = 0$ when method m attains the best feasible objective found for instance i. For a method that does not return a feasible solution, we set $\delta_{mi} = +\infty$. For a deviation threshold $\tau \geq 0$, we then define:

$$F_m(\tau) = \frac{1}{38} \left| \{i : \delta_{mi} \leq \tau\} \right|,$$

so that $F_m(\tau)$ is the proportion of all 38 test instances on which method m returns a feasible solution whose objective is within $\tau\%$ of the best feasible objective found.

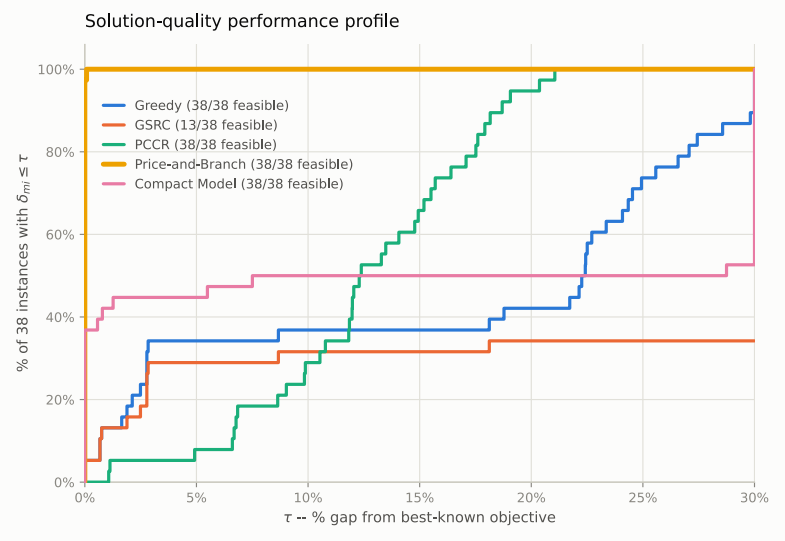
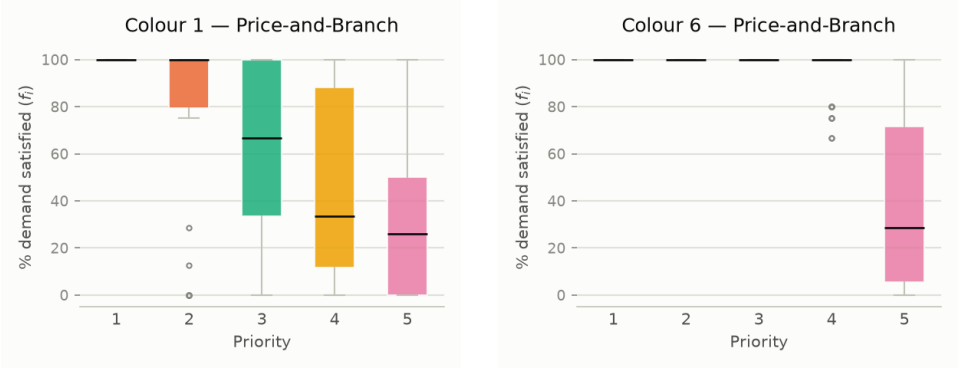


Figure 4: Solution-quality performance profile across the 38 test instances. For each relative-deviation threshold τ, the figure shows the proportion of instances on which each method attains a feasible solution within τ% of the best feasible objective found.

Figure 4 shows that Price-and-Branch consistently outperforms the alternative methods in solution quality, attaining the best or tied-best objective on 37 of 38 instances. The sole exception is 3_tol20, where the Compact Model achieves an objective of 1271 compared with 1270 for Price-and-Branch. Greedy, PCCR, Price-and-Branch, and the Compact Model all return feasible solutions on every instance, whereas GSRC succeeds on 13. Despite the improved feasibility of the numerically stabilised Compact Model, its solution quality remains substantially below that of Price-and-Branch on many instances. Overall, Price-and-Branch provides the strongest combination of solution quality and reliability.

Complete instance-level results for all five methods are reported in Supplementary Table S2. For methodological transparency, these report the best feasible solution produced by each method itself, without replacing it by the better of that solution and the corresponding `Greedy` solution. Price-and-Branch uses a heuristic pricing-pool size of $K = 20$ throughout the main computational comparison. Its exact pricing MIP is solved with a relative



(a) Colour 1.

(b) Colour 6.

Figure 5: Carrier-level demand fulfilment by priority class for Price-and-Branch on Colours 1 and 6. For carrier i, $f_i = 100a_i/d_i$ is the percentage of requested demand allocated. Priority 1 is the highest priority and Priority 5 the lowest.

MIP-gap tolerance of 10^{-9} and HiGHS's default absolute MIP-gap tolerance of 10^{-6}. Column-generation convergence is declared when the exact pricing solver returns an OPTIMAL status and the returned reduced cost does not exceed $\varepsilon = 10^{-6}$. This constitutes a numerical convergence certificate subject to the solver's tolerances. The Compact Formulation uses HiGHS's default relative MIP-gap tolerance, whereas the remaining MIP solves, apart from exact pricing, use a relative tolerance of 0.5%. For Price-and-Branch, the gaps reported in Supplementary Table S1 are calculated relative to the last solved restricted-master LP objective. When column-generation convergence is certified, these provide numerical optimality-gap estimates based on the full-master LP bound. Otherwise, the reported values, marked with an asterisk, are descriptive LP–integer differences and should not be interpreted as certified optimality gaps for the original problem. Price-and-Branch returns feasible solutions on all 38 instances and achieves the best or tied-best objective on 37, demonstrating its consistently strong solution quality and reliability across the test set.

6.5. Demand Fulfilment by Priority

Figure 5 illustrates two contrasting allocation outcomes. In Colour 1, where spectrum and interference constraints are more restrictive, fulfilment

generally decreases with priority, with substantial variation among carriers in Priorities 3–5. In contrast, Colour 6 achieves nearly complete fulfilment for Priorities 1–4, while most unmet demand is concentrated in Priority 5. These results illustrate how the priority-weighted objective favours higher-priority traffic when resources are limited. Since the objective is weighted rather than lexicographic, individual carrier fulfilment need not be strictly ordered by priority. The corresponding boxplots for the other six original instances, together with additional fulfilment distributions by priority, are provided in the Supplementary Material.

6.6. Price-and-Branch Ablations

We investigate three design choices in Price-and-Branch: Greedy seeding of the initial restricted master, the size K of the heuristic pricing pool, and the number of patterns admitted B before re-solving the master. All experiments use the eight original colour instances and the same 600-second wall-clock budget as the main computational study.

6.6.1. Greedy Seeding and Pricing-Pool Size

We first compare a Greedy-seeded initial master with a singleton-only initial master for $K \in \{0, 5, 10, 20\}$. For the Greedy-seeded variant, we additionally test $K = 40$ and $K = 50$ to examine whether further enlargement of the heuristic pool continues to improve performance.

Without a heuristic pool ($K = 0$), Greedy seeding improves five of the eight instances, ties on two, and is slightly worse on one, with a mean objective advantage of 242.9. This advantage falls to 7.1 at $K = 5$, 2.5 at $K = 10$, and 3.25 at $K = 20$. For the Greedy-seeded variant, increasing K from 20 to 50 improves the aggregate objective by only 0.61%, with non-monotonic changes at intermediate values. We therefore retain $K = 20$ as a moderate fixed setting rather than tuning it separately for individual instances.

6.6.2. Number of Pool Columns Added per Iteration

We also tested admitting multiple improving pool columns before re-solving the master, using batch sizes $B \in \{1, 5, 10\}$. Relative to $B = 1$,

Table 4: Effect of Greedy seeding and heuristic pricing-pool size K. Entries are objective value / wall-clock time (s).

	Greedy-seeded					
Colour	$K = 0$	$K = 5$	$K = 10$	$K = 20$	$K = 40$	$K = 50$
1	1528 / 540.3	1879 / 569.2	1941 / 547.1	1969 / 560.7	1938 / 600.6	1969 / 600.7
2	1736 / 542.7	1912 / 569.9	1961 / 600.4	1942 / 600.4	1946 / 600.5	2017 / 582.3
3	1271 / 1.3	1271 / 1.3	1271 / 1.3	1271 / 1.3	1271 / 1.3	1271 / 1.3
4	1652 / 339.3	1652 / 137.0	1650 / 131.4	1657 / 92.0	1654 / 97.0	1654 / 80.2
5	1328 / 0.9	1328 / 0.9	1328 / 0.9	1328 / 1.0	1328 / 1.0	1328 / 0.9
6	1628 / 540.4	1633 / 536.4	1638 / 364.7	1638 / 408.5	1635 / 335.5	1638 / 295.9
7	1726 / 229.6	1728 / 166.0	1728 / 157.4	1728 / 213.6	1728 / 148.7	1728 / 145.3
8	1806 / 540.6	1996 / 549.6	2004 / 546.5	2013 / 544.7	2027 / 454.2	2024 / 396.3
	Singleton-only					
1	856 / 540.3	1857 / 547.4	1933 / 548.5	1970 / 550.6		
2	1097 / 540.2	1894 / 571.7	1933 / 584.3	1934 / 600.4		
3	1271 / 32.5	1268 / 1.5	1270 / 1.6	1268 / 1.8		
4	1650 / 472.1	1655 / 141.3	1649 / 124.2	1651 / 91.3		
5	1326 / 1.7	1322 / 1.1	1323 / 1.1	1328 / 1.4		
6	1628 / 540.5	1635 / 540.4	1635 / 435.5	1635 / 362.3		
7	1728 / 309.7	1728 / 167.8	1728 / 181.9	1728 / 167.7		
8	1176 / 540.2	1983 / 544.7	2030 / 596.5	2006 / 403.6		

batching reduces the total number of column-generation iterations by 54% and 62% for $B = 5$ and $B = 10$, respectively. However, the aggregate objective changes by only $+0.06\%$ and -0.07%, respectively, with negligible changes in total computation time (-0.58% and $+0.82\%$). We therefore retain the one-column-at-a-time strategy ($B = 1$).

6.7. Robustness Across Perturbed Instances

This section examines the robustness of the five methods under controlled perturbations of carrier interference tolerances and carrier populations, with complete results reported in Supplementary Table S1.

Interference-tolerance perturbations. Price-and-Branch maintains its superior performance under perturbations of the carrier interference tolerances. Across the 16 interference-tolerance perturbations, Price-and-Branch returns feasible solutions on every instance and attains the best or tied-best objective on 15, falling short by only one objective unit on 3_tol20. Column-generation convergence is certified for 10 of these instances, comprising both

Table 5: Effect of heuristic pool batch size B. Entries report objective value / time (s) / column-generation iterations.

Colour	$B = 1$	$B = 5$	$B = 10$
1	1969 / 560.6 / 383	1968 / 567.7 / 115	1971 / 577.9 / 80
2	1942 / 600.4 / 435	1944 / 600.5 / 140	1971 / 600.6 / 105
3	1271 / 1.3 / 10	1271 / 1.3 / 6	1267 / 1.4 / 6
4	1657 / 92.0 / 202	1657 / 90.7 / 94	1657 / 88.8 / 76
5	1328 / 1.0 / 5	1328 / 1.0 / 3	1328 / 1.0 / 3
6	1638 / 411.2 / 430	1635 / 416.7 / 195	1638 / 404.6 / 163
7	1728 / 213.1 / 322	1728 / 200.4 / 258	1728 / 188.5 / 229
8	2013 / 563.5 / 479	2023 / 550.7 / 222	1977 / 600.3 / 192

perturbations of Colours 3–7, while those derived from Colours 1, 2, and 8 remain computationally more challenging and do not achieve certified convergence within the time limit.

Carrier-population perturbations. Price-and-Branch also demonstrates strong robustness to changes in carrier populations, achieving the best or tied-best objective on all 13 subsampled instances, with certified column-generation convergence on seven of them. Greedy, PCCR, Price-and-Branch, and the Compact Formulation return feasible solutions on all 13 subsampled instances, whereas GSRC succeeds on six. Price-and-Branch nevertheless achieves the best or tied-best objective on every subsampled instance, demonstrating its superior solution quality.

7. Conclusions and Future Research Directions

We studied a priority-weighted non-contiguous channelisation problem for multibeam satellite communications with carrier-specific cumulative interference constraints. Exploiting the interchangeability of subchannels, we developed a pattern formulation and a Price-and-Branch algorithm with Greedy seeding, heuristic pool pricing, and exact pricing when required.

Across 38 test instances, Price-and-Branch returned a feasible solution in every case and attained the best or tied-best objective among the five methods on 37 instances. Column-generation convergence was numerically certified on 22 instances, with a median certified optimality gap of 0.06%. The

computational results also demonstrate that the priority-weighted objective achieves the intended preferential fulfilment of higher-priority demand.

The computational budget is important in the intended application: channelisation is one component of `SatDreamOpt`, whose optimisation modules may be solved repeatedly while exploring improved system-level solutions. Our emphasis is therefore on obtaining high-quality solutions within a practically useful time budget rather than pursuing exactness at arbitrary computational cost.

Future research may explore more efficient exact-pricing formulations, adaptive construction and replenishment of the heuristic pattern pool, and warm starts between related `SatDreamOpt` modules to further improve computational efficiency and solution quality. Broader testing on additional operationally representative instances and system configurations would also help assess the practical applicability of the approach across a wider range of satellite systems.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this manuscript, the authors used ChatGPT (OpenAI) to assist with drafting, editing, and technical review, and Claude Code (Anthropic) to assist with software implementation and debugging. The authors developed the mathematical formulations and algorithmic procedures, reviewed and verified the AI-assisted outputs, and independently validated the computational results. The authors take full responsibility for the content of the manuscript.

References

3GPP (2011). Evolved universal terrestrial radio access (E-UTRA); physical layer procedures. Technical Specification TS 36.213 V10.2.0, 3rd Generation Partnership Project.

3GPP (2020). NR; physical layer procedures for data. Technical Specification TS 38.214 V16.2.0, 3rd Generation Partnership Project.

Aardal, K. I., van Hoesel, S. P. M., Koster, A. M. C. A., Mannino, C., & Sassano, A. (2007). Models and solution techniques for frequency assignment problems. *Annals of Operations Research*, 153, 79–129. https://doi.org/10.1007/s10479-007-0178-0

Abdu, T. S., Kisseleff, S., Lagunas, E., & Chatzinotas, S. (2021). Flexible resource optimization for GEO multibeam satellite communication system. *IEEE Transactions on Wireless Communications*, 20(12), 7888–7902. https://doi.org/10.1109/TWC.2021.3088609

Gaudry, A., Li, R., & Mak-Hau, V. (2023). FlexBeamOpt: Hybrid solution methodologies for high-throughput GEO satellite beam laydown and resource allocation. *Int. J. Satell. Commun. Netw.*, 41(6), 515–538.

Gaudry, A. & Mak-Hau, V. (2026a). *Non-contiguous channel allocation in multi-beam satellites: Formulations, upper bounds and cooperative matheuristics*. https://arxiv.org/abs/2610.17906

Gaudry, A. & Mak-Hau, V. (2026b). Optimised channel allocation in satellite communication systems: A comparison of models and solution techniques. *Optimization and Engineering*.

Hadi, M. & Ghazizadeh, R. (2020). Sub-channel assignment and power allocation in OFDMA-NOMA based heterogeneous cellular networks. *AEU – International Journal of Electronics and Communications*, 120, 153195. https://doi.org/10.1016/j.aeue.2020.153195

Hale, W. K. (1980). Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 68(12), 1497–1514.

Hemazro, T. D., Jaumard, B., & Marcotte, O. (2008). A column generation and branch-and-cut algorithm for the channel assignment problem. *Computers & Operations Research*, 35(4), 1204–1226. https://doi.org/10.1016/j.cor.2006.07.012

Jaumard, B., Marcotte, O., Meyer, C., & Vovor, T. (2001). Comparison of column generation models for channel assignment in cellular networks.

Discrete Applied Mathematics, 112(1–3), 217–240. https://doi.org/10.1016/S0166-218X(00)00317-6

Kawamoto, Y., Kamei, T., Takahashi, M., Kato, N., Miura, A., & Toyoshima, M. (2020). Flexible resource allocation with inter-beam interference in satellite communication systems with a digital channelizer. *IEEE Transactions on Wireless Communications*, 19(5), 2934–2945. https://doi.org/10.1109/TWC.2020.2969173

Kiatmanaroj, K., Artigues, C., & Houssin, L. (2016). On scheduling models for the frequency interval assignment problem with cumulative interferences. *Engineering Optimization*, 48(5), 740–755. https://doi.org/10.1080/0305215X.2015.1056789

Kiatmanaroj, K., Artigues, C., Houssin, L., & Messine, F. (2012). Hybrid discrete-continuous optimization for the frequency assignment problem in satellite communication system. *IFAC Proceedings Volumes*, 45(6), 1419–1424. https://doi.org/10.3182/20120523-3-RO-2023.00197

Kisseleff, S., Lagunas, E., Abdu, T. S., Chatzinotas, S., & Ottersten, B. (2021). Radio resource management techniques for multibeam satellite systems. *IEEE Communications Letters*, 25(8), 2448–2452. https://doi.org/10.1109/LCOMM.2020.3033357

Koster, A. & Tieves, M. (2012). Column generation for frequency assignment in slow frequency hopping networks. *EURASIP Journal on Wireless Communications and Networking*, 2012(1), 253. https://doi.org/10.1186/1687-1499-2012-253

Li, R., Gaudry, A., & Mak-Hau, V. (2024). CG-FlexBeamOpt: Advanced solution methodology for high throughput GEO satellite beam laydown and resource allocation. *Int. J. Satell. Commun. Netw.*, 42(4), 286–312.

Li, R., Mak-Hau, V. H., Lam, T., & Gaudry, A. (2025). Heuristic-enhanced mathematical optimization algorithms for channel assignment and interference mitigation. *MILCOM 2025 - 2025 IEEE Military Communications Conference (MILCOM)*, 1–6.

Mehrotra, A. & Trick, M. A. (1996). A column generation approach for graph coloring. *INFORMS Journal on Computing*, 8(4), 344–354. https://doi.org/10.1287/ijoc.8.4.344

Mokari, N., Alavi, F., Parsaeefard, S., & Le-Ngoc, T. (2016). Limited-feedback resource allocation in heterogeneous cellular networks. *IEEE Transactions on Vehicular Technology*, 65(4), 2509–2521. https://doi.org/10.1109/TVT.2015.2428997

Nicholas, P. J. & Hoffman, K. L. (2019). Efficient solution methods for the cumulative-interference channel assignment problem using integer optimization and constraint programming. *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, 443–460. https://doi.org/10.1007/978-3-030-19212-9_29

Palpant, M., Oliva, C., Artigues, C., Michelon, P., & Didi Biha, M. (2008). Models and methods for frequency assignment with cumulative interference constraints. *International Transactions in Operational Research*, 15(3), 307–324. https://doi.org/10.1111/j.1475-3995.2008.00630.x

Ramírez, T., Mosquera, C., & Alagha, N. (2022). Flexible user mapping for radio resource assignment in advanced satellite payloads. *IEEE Transactions on Broadcasting*, 68(3), 723–739. https://doi.org/10.1109/TBC.2022.3178018

Supplementary Material for "Non-contiguous satellite channel allocation under cumulative interference: A hybrid price-and-branch approach"

Vicky Mak-Hau[a], Angus Gaudry[a]

[a]*School of Information Technology, Deakin University, Victoria, Australia*

S1. Proof of Proposition 1

This section provides the full proof of Proposition 1 in the main manuscript, establishing the integer equivalence of the Compact Formulation and the Pattern Formulation.

Consider first a feasible solution $(x, \mathcal{I})$ of the Compact Formulation. For each subchannel t, define the corresponding pattern $p(t)$ by the carrier assigned to each beam on that subchannel, with 0 indicating no carrier from the beam is assigned. Constraint (2) ensures that $p(t)$ is well-defined, since at most one carrier from each beam can be assigned to any subchannel. Let y_p be the multiplicity of Pattern p. We have that

$$\sum_{p \in \mathcal{P}} y_p \leq T.$$

For every carrier i, as

$$\sum_{\substack{p \in \mathcal{P}: \\ i \in V(p)}} y_p = \sum_{t=1}^{T} x_{it} \leq d_i,$$

the demand constraints of the Pattern Formulation are satisfied. Moreover,

whenever $x_{it} = 1$, Constraint (4) gives us

$$\sum_{\substack{j \in \mathcal{C}: \\ \text{beam}(j) \neq \text{beam}(i)}} p_{ji} x_{jt} \leq \mathcal{I}_{it}.$$

Summing over the subchannels assigned to carrier i, using Constraint (5) we get:

$$\sum_{\substack{p \in \mathcal{P}: \\ i \in V(p)}} \left(\sum_{\substack{j \in V(p) \\ j \neq i}} p_{ji} \right) y_p \leq \Gamma_i.$$

Hence the interference constraint of the Pattern Formulation is also satisfied. Since each occurrence of carrier i contributes v_i in both formulations, the objective value is unchanged.

Conversely, consider a feasible integer solution y of the Pattern Formulation. Since

$$\sum_{p \in \mathcal{P}} y_p \leq T,$$

the selected pattern copies can be assigned arbitrarily to distinct subchannels $t = 1, \ldots, T$. For each such subchannel, set $x_{it} = 1$ if carrier i belongs to its assigned pattern and set $x_{it} = 0$ otherwise; set all variables associated with unused subchannels to zero. By the definition of a pattern, Constraint (2) is satisfied, while Constraint (9) directly implies Constraint (3).

For each carrier i and subchannel t, define

$$\mathcal{I}_{it} = \begin{cases} \displaystyle\sum_{\substack{j \in \mathcal{C}: \\ \text{beam}(j) \neq \text{beam}(i)}} p_{ji} x_{jt}, & \text{if } x_{it} = 1, \\ 0, & \text{if } x_{it} = 0. \end{cases}$$

When $x_{it} = 1$, Constraint (4) is satisfied at equality. When $x_{it} = 0$, the left-hand side of the constraint is non-positive because $p_{ji} \geq 0$ and $x_{jt} \leq 1$, and hence the constraint is satisfied by $\mathcal{I}_{it} = 0$. Finally, Constraint (10)

gives

$$\sum_{t=1}^{T} \mathcal{I}_{it} \leq \Gamma_i,$$

so Constraint (5) is satisfied. The resulting Compact-Formulation solution therefore is feasible and has the same objective value as the pattern solution.

Consequently, the two formulations describe the same feasible allocations for any arbitrary permutation of the subchannel indices, and have identical optimal integer objective values. Therefore, the proposition is proved. □

S2. Alternative Interference Linearisations

The original Compact Formulation (O) has an implicit big-M coefficient

$$M_i^O = \sum_{j:\text{beam}(j)\neq\text{beam}(i)} p_{ji},$$

which can amplify small integrality deviations. We investigate three alternative linearisations. Recall that $p_{ji} \geq 0$, $\Gamma_i > 0$. For simplicity of exposition, denote $C_i^{\times} = \{j \in C : \text{beam}(j) \neq \text{beam}(i)\}$. We explain the three methods used (A, N, and B).

A. Coefficient-capped tight-big-M. We capped the interference coefficients at $\widehat{p}_{ji} = \min\{p_{ji}, \kappa\Gamma_i\}$ for $\kappa = 1.1$, and tightened the big-M coefficient by exploiting beam exclusivity:

$$M_i^A = \sum_{\substack{b\in B \\ b\neq\text{beam}(i)}} \max_{j\in C_b} \widehat{p}_{ji}.$$

The interference constraints become

$$\sum_{j\in C_i^{\times}} \widehat{p}_{ji} x_{jt} - M_i^A(1 - x_{it}) \leq I_{it}, \qquad \forall i, t, \tag{S1}$$

while retaining the original cumulative budgets $\sum_t I_{it} \leq \Gamma_i$ and $I_{it} \geq 0$.

N. Normalised tight-big-M. We divide the constraints of A by Γ_i. Let:

$$q_{ji} = \frac{\widehat{p}_{ji}}{\Gamma_i}, \qquad M_i^N = \frac{M_i^A}{\Gamma_i}.$$

The resulting formulation is

$$\sum_{j \in C_i^\times} q_{ji} x_{jt} - M_i^N (1 - x_{it}) \leq H_{it}, \qquad \forall i, t, \tag{S2}$$

with $\sum_t H_{it} \leq 1$ and $H_{it} \geq 0$.

B. Disaggregated interference. We introduce a set of additional variables $y_{ijt} \geq 0$ for each unordered cross-beam carrier pair $\{i, j\}$ and subchannel t, use the following interference constraints instead.

$$y_{\{i,j\}t} \geq x_{it} + x_{jt} - 1, \qquad \forall \{i, j\}, t, \tag{S3}$$

$$\sum_t \sum_{j \in C_i^\times} p_{ji} y_{\{i,j\}t} \leq \Gamma_i, \qquad \forall i \in C. \tag{S4}$$

All three alternatives (A, N, and B) preserve the feasible binary allocations of O. In A, capping only affects coefficients already exceeding Γ_i, while beam exclusivity ensures the validity of M_i^A; N is a positive rescaling of A. In B, setting $y_{ijt} = \max\{0, x_{it} + x_{jt} - 1\}$ for binary x recovers the original cumulative interference. Thus, all four formulations have the same optimal integer objective value.

Although all four formulations are exact for binary assignments, finite integrality and feasibility tolerances can still lead to violations after rounding. Formulations A and N substantially reduce this sensitivity through coefficient capping and tighter big-M bounds, while disaggregation (B) eliminates the aggregated big-M mechanism but remains subject to numerical tolerances. Consequently, all reported allocations are independently validated against the original interference coefficients and budgets.

S2.1. Computational Comparison

The four formulations were tested on all 38 instances, with a nominal 600-second computational budget per instance, including model construction. All returned allocations were independently validated using the original interference data. Table S1 summarises the results.

Computational environment. The interference-linearisation experiments were conducted on a separate machine from the main computational study: an Apple Mac Studio (Mac13,2) with an M1 Ultra processor (20 CPU cores: 16 performance and 4 efficiency), 64 GiB unified memory, and macOS 27.0.1. All formulations were implemented in Python 3.14.8 using OR-Tools 9.15.6755 (`MPSolver`) and HiGHS 1.12.0. Each solver was restricted to a single thread, while independent instances were executed concurrently in separate processes. Reported wall-clock times therefore reflect single-threaded execution under concurrent load rather than isolated runs. All four formulations were solved using HiGHS's default relative MIP-gap tolerance.

Table S1: Comparison of interference linearisations. Objective totals are reported for (i) the 23 instances on which all four formulations returned feasible solutions, and (ii) the complete set of 38 instances, with missing solutions assigned zero. Variable counts refer to the largest instance.

Formulation	Relative big-M	Obj. (23)	Obj. (38)	No solution	Variables (1_orig)
O: Original	1.37×10^5	21,884	25,305	0	82,400
A: Clipped	6.23	**25,682**	25,991	0	82,400
N: Normalised	6.23	25,323	**29,120**	0	82,400
B: Disaggregated	—	20,375	20,375	15	3,549,600

Both A and N improve the aggregate objective over O on the 23 common instances. A performs best on this subset, whereas N achieves the highest aggregate objective over all 38 instances. Although A and N are algebraically equivalent, their different scaling produces different computational outcomes. Disaggregation avoids the aggregated big-M sensitivity but substantially increases model size, returning no solution on 15 instances.

We therefore adopt N for the Compact Model experiments. These experiments investigate the effects of alternative interference linearisations and numerical scaling; their computational results are not directly comparable with those of the main study because of differences in hardware and execution conditions.

S3. Proof of Proposition 2

This section provides the full proof of Proposition 2 in the main manuscript, establishing that the exact pricing problem is NP-hard.

We reduce the Maximum Weight Independent Set (MWIS) problem to the pricing subproblem. Let $G = (V, E)$ be an arbitrary MWIS instance with nonnegative vertex weights w_i. For each vertex $i \in V$, create one carrier and assign each carrier to a distinct beam, so that the beam-exclusivity constraints are redundant. Set $v_i - \pi_i = w_i$. For each pair of carriers, we define

$$q_{ij} := \mu_i p_{ji} + \mu_j p_{ij} = \begin{cases} M, & \{i,j\} \in E, \\ 0, & \{i,j\} \notin E, \end{cases}$$

where $M > \sum_{i \in V} w_i$. Such coefficients can be obtained, for example, by setting $\mu_i = 1$ and $p_{ij} = p_{ji} = M/2$ for every $\{i,j\} \in E$. The pricing objective therefore becomes

$$\max \sum_{i \in V} w_i x_i - M \sum_{\{i,j\} \in E} x_i x_j.$$

Any solution containing both endpoints of an edge incurs a penalty M, which exceeds the total weight of all vertices, and therefore no optimal solution will contain adjacent vertices. The selected vertices therefore form an independent set, and among all such sets the objective maximises their total weight. Thus, solving the pricing subproblem solves MWIS as a special case. Since MWIS is NP-hard, the pricing subproblem is NP-hard. □

S4. Additional Details of the Greedy Construction Heuristic

This section provides implementation details and pseudocode for the Adaptive Score-Weighted Randomized Greedy heuristic described in Section 3 of the main manuscript.

Acceptance test.. Let TI_i denote the cumulative interference received by Carrier i over all completed subchannels, and let SG_i denote the interference accumulated so far in the current subchannel (t). A sampled candidate i is accepted only if

$$\mathrm{TI}_i + \mathrm{SG}_i + \sum_{k \in A} p_{ki} \leq \Gamma_i$$

and

$$\mathrm{TI}_k + \mathrm{SG}_k + p_{ik} \leq \Gamma_k \qquad \forall k \in A.$$

In this way, our acceptance test accounts for both interference accumulated on previous subchannels and the running interference within the current subchannel. Once all beams have been processed for subchannel t, the realised within-subchannel interference is added to the cumulative interference totals. A pseudocode for `Greedy` is given in Algorithm S1.

S5. Details for Price-and-Branch

S5.1. Implementation Notes

Solver split. As a solver-specific workaround, we use different solvers for the Master LP and the remaining optimisation problems. In the OR-Tools build used in this project (9.15.6755, HiGHS 1.12.0), `Constraint.dual_value()` returns the constraint's primal activity rather than its shadow price. We therefore solve the Master LP relaxation with GLOP, as we have not observed GLOP's duals exhibiting the same failure mode in any of the 38 instances tested. The exact pricer and the final integer solver used HiGHS, since neither requires dual values.

Algorithm S1 Greedy subchannel allocation

for $t = 1$ to T **do**
  Set $A \leftarrow \emptyset$ ▷ carriers accepted into subchannel t so far
  Initialise within-subchannel interference to zero
  for each beam $b \in \mathcal{B}$ **do**
    Form the candidate pool $\leftarrow \{i \in C_b : a_i < d_i\}$
    while pool $\neq \emptyset$ **do**
      draw candidate i from pool with probability $\propto$ score(i)
      if accepting i keeps the cumulative interference of every carrier in $A \cup \{i\}$ within budget **then**
        accept i into subchannel t; $A \leftarrow A \cup \{i\}$; $a_i \leftarrow a_i + 1$
        update the within-subchannel interference for carriers in A
        break
      else
        remove i from pool
      end if
    end while
  end for
  add the realised interference in subchannel t to the cumulative interference totals
end for

Solver-specific safeguards. The final integer solve uses a small relative MIP-gap tolerance (0.5%) rather than HiGHS's tighter default. The Compact Formulation instead uses the default HiGHS relative MIP-gap tolerance. Without this tolerance, hitting a hard time limit can leave HiGHS in a status from which this OR-Tools build cannot extract *any* solution values, including those of an incumbent already found. Allowing a small optimality gap typically allows the solver terminate cleanly, for every one of the 38 problem instances we have tested, before reaching this failure mode, ensuring that the best solution found remains retrievable.

S6. Additional Computational Details

S6.1. Instance diagnostics

Figure S1 shows the distribution of nonzero pairwise co-channel interference in log scale, we can see that Colour 1's box sits well above the others.

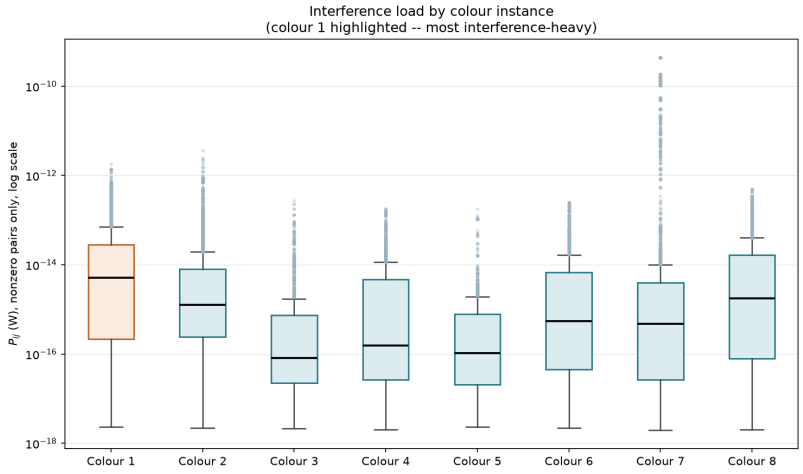


Figure S1: Distribution of pairwise co-channel interference p_{ij} (nonzero pairs only, log scale) across the eight colour instances.

S7. Complete and Supplementary Computational Results

S7.1. Complete results for all 38 instances

Table S2: Performance of the five solution methods across all 38 test instances.

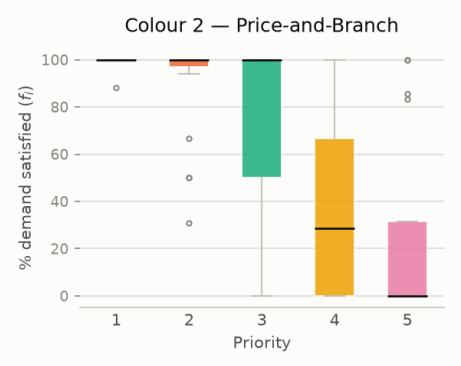
	Greedy				GSRC					PCCR				Price-and-Branch (K=20)					Compact Model				
Instance	Obj	Alc	Unm	Time	UB	Obj	Alc	Unm	Time	Obj	Alc	Unm	Time	Obj	Alc	Unm	Gap	Time	Obj	Alc	Unm	Gap	Time
1_orig	1308	441	689	6.3	3145	–	–	–	601.1	1692	484	646	28.7	**1969**	**627**	**503**	4.68%*	560.3	593	177	953	80.35%	601.5
2_orig	1512	524	392	4.3	2628	–	–	–	600.6	1594	466	450	7.1	**1942**	**608**	**308**	6.54%*	600.5	903	289	627	59.78%	601.2
3_orig	1250	458	12	1.0	1274	–	–	–	600.2	1187	412	58	0.1	**1271**	**467**	**3**	0.04%	1.4	**1271**	**467**	**3**	0.24%	600.5
4_orig	1163	458	178	2.1	1683	–	–	–	600.2	1374	449	187	0.2	**1657**	**610**	**26**	0.19%	98.0	1636	604	32	2.79%	600.6
5_orig	1319	490	11	1.0	1338	1319	490	11	168.0	1168	406	95	0.1	**1328**	**491**	**10**	0.06%	0.9	**1328**	**491**	**10**	0.00%	572.1
6_orig	1266	483	164	2.1	1724	–	–	–	600.4	1384	437	210	0.4	**1638**	**572**	**75**	1.02%	407.4	830	276	371	51.86%	600.8
7_orig	1680	**509**	**182**	1.6	2125	1680	**509**	**182**	13.0	1514	417	274	0.2	**1728**	**509**	**182**	0.01%	208.1	**1728**	**509**	**182**	0.00%	246.9
8_orig	1478	483	399	3.8	2597	–	–	–	601.5	1603	447	435	1.2	**2013**	**612**	**270**	2.07%*	544.4	950	305	577	63.02%	601.1
1_tol10	1337	450	680	6.0	3145	–	–	–	602.6	1678	481	649	120.4	**1969**	**629**	**501**	4.62%*	581.1	5	1	1129	99.93%	601.5
1_tol20	1309	445	685	6.1	3145	–	–	–	601.0	1689	494	636	60.6	**1893**	**605**	**525**	7.09%*	600.5	5	1	1129	99.93%	601.7
2_tol10	1515	526	390	4.0	2628	–	–	–	600.0	1663	486	430	8.4	**1955**	**605**	**311**	6.32%*	600.5	844	272	644	62.41%	601.2
2_tol20	1499	527	389	4.0	2628	–	–	–	600.1	1688	501	415	2.9	**1915**	**597**	**319**	6.82%*	600.5	915	287	629	59.24%	601.2
3_tol10	1244	457	13	1.0	1274	–	–	–	600.1	1186	406	64	0.1	**1271**	**467**	**3**	0.03%	1.3	**1271**	**467**	**3**	0.24%	600.8
3_tol20	1247	457	13	1.0	1274	1247	457	13	67.9	1184	412	58	0.1	1270	466	4	0.10%	1.4	**1271**	**467**	**3**	0.24%	600.7
4_tol10	1180	460	176	1.8	1683	–	–	–	600.2	1352	442	194	0.2	**1652**	**608**	**28**	0.49%	94.5	1528	564	72	9.21%	600.5
4_tol20	1152	456	180	1.8	1683	–	–	–	600.3	1362	445	191	0.3	**1653**	**609**	**27**	0.49%	69.3	1640	605	31	2.55%	600.6
5_tol10	1318	490	11	0.7	1338	1318	490	11	305.7	1171	412	89	0.1	**1328**	**491**	**10**	0.05%	0.9	**1328**	**491**	**10**	0.00%	178.4
5_tol20	1319	490	11	0.7	1338	1319	490	11	168.8	1169	412	89	0.0	**1328**	**491**	**10**	0.06%	1.0	**1328**	**491**	**10**	0.00%	260.1
6_tol10	1245	470	177	2.2	1724	–	–	–	600.3	1371	437	210	0.3	**1640**	**573**	**74**	0.80%	376.2	840	278	369	51.28%	600.7
6_tol20	1240	465	182	2.3	1724	–	–	–	600.3	1390	440	207	0.3	**1639**	**574**	**73**	0.89%	402.3	1549	561	86	10.15%	600.7
7_tol10	1680	**509**	**182**	1.6	2125	1680	**509**	**182**	15.9	1558	434	257	0.1	**1728**	**509**	**182**	0.01%	151.5	**1728**	**509**	**182**	0.00%	91.2
7_tol20	1679	**509**	**182**	1.6	2125	1679	**509**	**182**	18.6	1572	438	253	0.2	**1728**	**509**	**182**	0.01%	177.6	**1728**	**509**	**182**	0.00%	547.8
8_tol10	1526	497	385	4.3	2597	–	–	–	601.9	1592	450	432	4.5	**1967**	**595**	**287**	4.20%*	600.3	937	302	580	63.53%	601.0
8_tol20	1508	498	384	4.3	2597	–	–	–	602.8	1586	440	442	18.0	**2009**	**609**	**273**	1.93%*	561.6	1028	339	543	60.23%	600.9
1_sub80	1445	492	436	4.8	2583	–	–	–	601.9	1634	485	443	19.1	**1863**	**594**	**334**	3.39%*	553.4	600	188	740	76.30%	601.2
1_sub90	1347	453	543	5.7	2752	–	–	–	602.4	1598	466	530	120.3	**1847**	**595**	**401**	3.20%*	571.8	5	1	995	99.93%	601.4
2_sub80	1375	482	263	3.1	2135	–	–	–	600.4	1536	468	277	1.3	**1822**	**588**	**157**	3.07%*	553.0	776	255	490	59.83%	601.0
2_sub90	1370	503	328	3.6	2354	–	–	–	600.5	1535	460	371	3.4	**1888**	**598**	**233**	3.34%*	554.2	731	241	590	64.13%	601.0
6_sub80	1285	480	48	1.5	1420	1285	480	48	179.1	1268	418	110	0.3	**1407**	**516**	**12**	0.92%	4.3	1399	512	16	1.48%	600.7
6_sub90	1202	459	114	1.9	1512	1202	459	114	543.5	1273	425	148	0.4	**1468**	**537**	**36**	1.05%	208.7	1046	376	197	30.82%	600.6
7_sub80	1487	**450**	**113**	1.2	1760	1487	**450**	**113**	9.6	1450	421	142	0.1	**1525**	**450**	**113**	0.01%	78.1	**1525**	**450**	**113**	0.00%	53.3
7_sub90	1609	**493**	**139**	1.4	1946	1609	**493**	**139**	9.3	1512	425	207	0.1	**1655**	**493**	**139**	0.01%	109.8	**1655**	**493**	**139**	0.00%	68.6
8_sub80	1362	456	249	3.3	2001	–	–	–	600.2	1542	483	222	1.7	**1752**	**575**	**130**	0.99%*	524.9	732	245	460	63.42%	600.6
8_sub90	1470	479	315	3.8	2352	–	–	–	601.2	1582	455	339	7.1	**1918**	**590**	**204**	1.27%*	564.4	904	288	506	61.56%	600.7
3_sub85	**1122**	**396**	**0**	1.2	1122	**1122**	**396**	**0**	1.1	1110	384	12	0.1	**1122**	**396**	**0**	0.00%	1.0	**1122**	**396**	**0**	0.00%	157.4
4_sub85	1103	435	89	1.8	1366	–	–	–	600.2	1266	444	80	0.1	**1358**	516	8	0.59%	2.0	**1358**	**518**	**6**	0.59%	600.5
5_sub85	**1066**	**418**	**0**	0.6	1066	**1066**	**418**	**0**	0.8	1054	406	12	0.1	**1066**	**418**	**0**	0.00%	0.7	**1066**	**418**	**0**	0.00%	70.5
1_sub85tol15	1385	470	507	5.1	2732	–	–	–	602.0	1665	497	480	25.7	**1861**	**603**	**374**	3.28%*	583.2	722	214	763	71.80%	601.0

Note. Obj = priority-weighted objective value; Alc = allocated subchannels; Unm = unmet demand; Time = wall-clock seconds; UB = upper bound obtained from the interference-relaxed GSRC model; A dash indicates that no feasible incumbent was returned. Boldface denotes the best objective, largest allocation, or smallest unmet demand among the five methods; ties are included. For Price-and-Branch, an unmarked Gap is a numerically certified optimality-gap estimate, since exact pricing established column-generation convergence within the specified tolerances. A superscript * indicates that convergence was not certified; these values are restricted-master LP diagnostics and should not be interpreted as certified optimality gaps or upper bounds on the true optimality gap. Of the 38 instances, 22 have certified Price-and-Branch gaps. Gap = $100(U - z)/U$, where z is the objective value of the feasible integer solution returned by the respective method (reported under Obj). For Price-and-Branch, U is the last solved restricted-master LP objective; for the Compact Formulation, U is the best bound returned by HiGHS for that formulation. The Compact Formulation uses HiGHS's default relative MIP-gap tolerance; a displayed gap of 0.00% does not necessarily imply exact optimality.

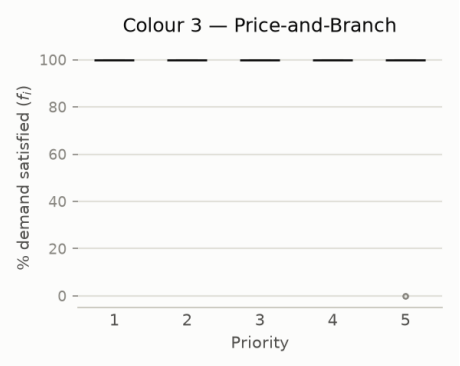
S7.2. Additional Demand-Fulfilment Results

The main manuscript presents Colours 1 and 6 as two contrasting examples of demand fulfilment under priority-weighted allocation. Figure S2 reports the corresponding results for the remaining six original instances. Colours 2, 7, and 8 exhibit more pronounced shortfalls among lower-priority carriers, whereas Colours 3–5 generally achieve high fulfilment across most priority classes. These results further illustrate how the allocation outcomes depend on the relative availability of spectrum and interference resources.

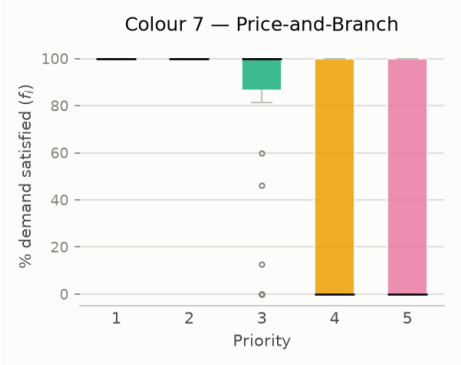
Figures S3 and S4 show the distributions of carriers across demand-fulfilment intervals, disaggregated by priority class, for all eight original instances. These figures provide a more detailed view of the extent to which demand remains unsatisfied within each priority class.



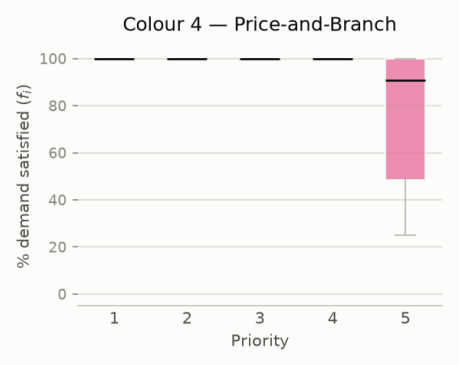
(a) Colour 2.



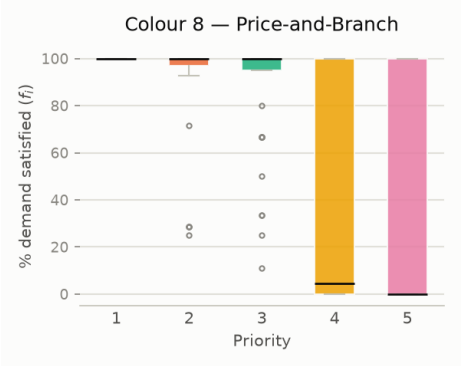
(b) Colour 3.



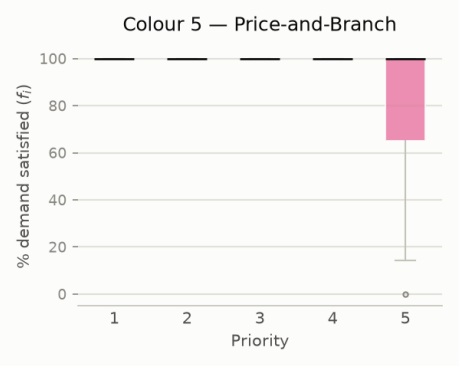
(c) Colour 7.



(d) Colour 4.

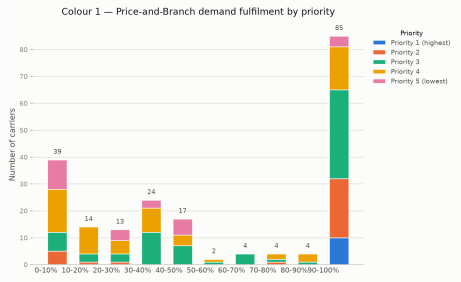


(e) Colour 8.

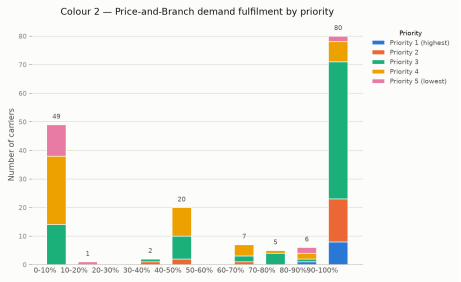


(f) Colour 5.

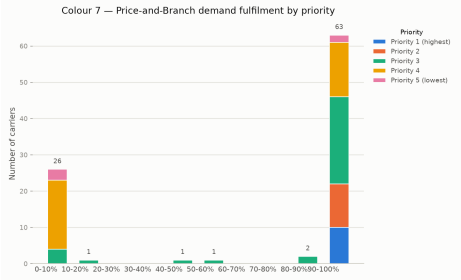
Figure S2: Carrier-level demand fulfilment by priority class for Price-and-Branch on the six original instances not shown in the main manuscript. The left column shows Colours 2, 7, and 8; the right column shows Colours 3, 4, and 5. For carrier i, $f_i = 100a_i/d_i$ denotes the percentage of requested demand allocated. Priority 1 is the highest priority and Priority 5 the lowest.



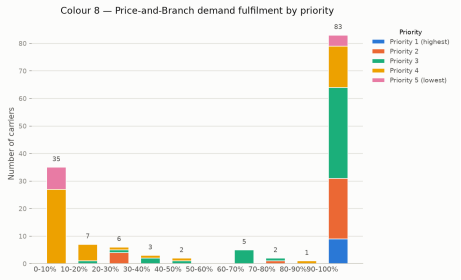
(a) Colour 1.



(b) Colour 2.

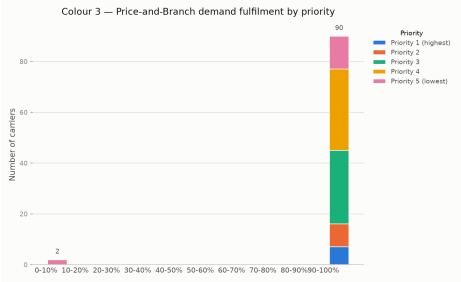


(c) Colour 7.

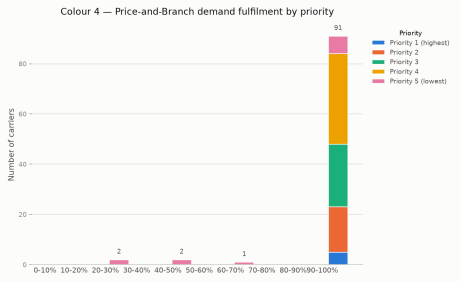


(d) Colour 8.

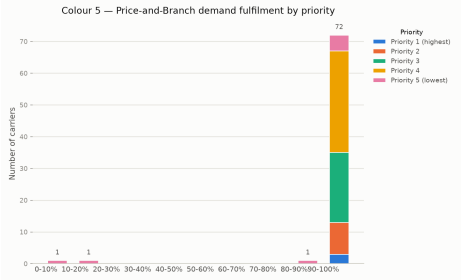
Figure S3: Distribution of carrier-level demand fulfilment for Price-and-Branch on Colours 1, 2, 7, and 8. Bars show the number of carriers in each fulfilment interval, disaggregated by priority class.



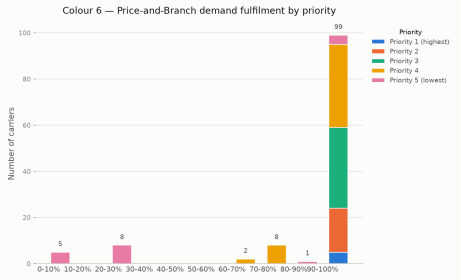
(a) Colour 3.



(b) Colour 4.



(c) Colour 5.



(d) Colour 6.

Figure S4: Distribution of carrier-level demand fulfilment for Price-and-Branch on Colours 3–6. Bars show the number of carriers in each fulfilment interval, disaggregated by priority class.