



INFORMS Journal on Applied Analytics

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

Optimizing Music Station Playlists on Broadcast Radio

José Antonio Carbajal, Juan Ma, Nannan Chen, Mario Aboytes-Ojeda

To cite this article:

José Antonio Carbajal, Juan Ma, Nannan Chen, Mario Aboytes-Ojeda (2025) Optimizing Music Station Playlists on Broadcast Radio. INFORMS Journal on Applied Analytics 55(5):399-411. <https://doi.org/10.1287/inte.2025.0248>

Full terms and conditions of use: <https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2025, INFORMS

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes. For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Optimizing Music Station Playlists on Broadcast Radio

José Antonio Carbajal,^{a,*} Juan Ma,^b Nannan Chen,^c Mario Aboytes-Ojeda^d

^aPrinceton Consultants, Inc., Princeton, New Jersey 08540; ^biHeartMedia, Inc., San Antonio, Texas 78258; ^cHungryroot, Inc., New York, New York 10010; ^dVerde Studio Engineering, San Antonio, Texas 78212

*Corresponding author

Contact: acarbajal@princeton.com,  <https://orcid.org/0000-0003-4025-1755> (JAC); juliema1205@gmail.com,

 <https://orcid.org/0000-0001-9912-7383> (JM); nannanchen0920@gmail.com (NC); mario.aboytes@gmail.com,

 <https://orcid.org/0000-0003-0965-9840> (MA-O)

<https://doi.org/10.1287/inte.2025.0248>

Copyright: © 2025 INFORMS

Abstract. We developed a mathematical optimization-based engine that generates 24/7 music playlists for radio stations subject to strategic scheduling goals and key business rules. Utilizing song metadata, such as tempo and mood; latest song research results produced by machine learning models; and radio listenership data, the engine produces music playlists that optimize strength and diversity simultaneously. The engine has been successfully deployed to production and has shown its power in efficiently and effectively creating customized music playlists for various radio stations.

History: This paper has been accepted for the *INFORMS Journal on Applied Analytics* Special Issue—2024 Daniel H. Wagner Prize for Excellence in the Practice of Advanced Analytics and Operations Research.

Keywords: media industry • radio broadcasting • music playlist • integer programming • multicriteria optimization

Introduction

Commercial radio broadcasters primarily derive their revenues through selling advertising spots on their radio stations. Except for news, talk, and sports radio stations, most U.S. radio stations (approximately 85%) adhere to various music formats. These stations focus predominantly on playing music from a specific genre, such as contemporary hits radio (CHR), country, and classic rock, catering to listeners' preferences to attract targeted demographics for advertisers.

Unlike broadcast television and cable, which typically have a national presence, commercial radio stations are largely enjoyed by local audiences. This localized focus requires using a dynamic content strategy that aligns with the shifting preferences of listeners to maintain a competitive edge. For music stations, it is crucial to keep a pulse on the latest musical trends in their specific markets and integrate them judiciously into their music playlists. Simultaneously, they must balance this responsiveness with a consistent adherence to their established music strategy and brand identity.

Music radio broadcasters largely rely on the subject matter expertise of program directors to craft their music playlists. This process involves two key steps:

Step 1: Song selection and frequency; program directors leverage research agencies to conduct music research and test songs with curated audience panels to collect their feedback about the music played. They use this information and other analytics to determine which artists and songs should be played on the

stations and the frequencies with which those songs should be played.

Step 2: Song sequence planning: program directors use music scheduling software to determine the specific sequences in which songs are played. The music scheduling software applies business rules aimed to improve the flow from one song into another such as avoiding back-to-back slow songs and ensuring adequate time separation between two plays of the same song or between two plays of different songs by the same artist. Program directors build the playlists iteratively and typically make manual adjustments to finalize them.

Despite this structured approach, the process is highly dependent on the subjective judgments and personal preferences of the program directors. Consequently, two directors may create vastly different playlists using the same data.

In recent years, the media advertising landscape has undergone significant changes. Music radio broadcasters now compete directly with music streaming services. In response, the industry has embarked on major consolidation efforts. The goal of these efforts is to streamline operations by eliminating redundant processes and capitalizing on economies of scale in planning. At the same time, broadcasters are focusing on maintaining a high degree of content localization, which is crucial for remaining competitive.

Crafting localized music playlists at scale across hundreds of stations poses significant challenges, primarily because of the subjective nature of the processes

traditionally followed in the radio industry. These processes have historically depended heavily on the expertise of program directors. Nevertheless, one of the largest radio broadcasters in the United States has successfully developed and implemented large-scale mathematical programming models that blend predictive insights and other analytics with business objectives and restrictions, generating customized music playlists for each station that optimize both music strength and diversity.

The rest of the paper is organized as follows: In the “An Overview on Music Programming and Related Literature” section, we review the related literature, highlighting the differences between music scheduling and the extensively studied ad scheduling. In the “The Music Playlist Creation Process” section, we outline the process for crafting music schedules on broadcast radio. In the “Methods: Leveraging Mathematics to Craft Music Schedules” section, we describe the mathematical models developed to optimize music playlists at scale. In the “Implementation Challenges” section, we discuss the technical and nontechnical challenges encountered during the implementation of our solutions. In the “Impact of the Implementation” section, we evaluate the impact of our implementation. Finally, the appendices include the explicit mathematical formulations and algorithms used in our implementation.

An Overview on Music Programming and Related Literature

Advanced analytics, particularly optimization modeling, has proven effective for revenue management tasks in television and radio broadcasting companies given the perishable nature of advertising spots. Carbajal and Chaar (2017) and Carbajal et al. (2019) summarize the relevant literature in this field, categorizing challenges into strategic, tactical, and operational levels. Strategic challenges include planning advertising capacity and allocating it between the up-front market (ad buys made months in advance for the upcoming year) and the scatter market (ads purchased closer to airtime, typically at higher prices). Tactical challenges include inventory-mix distribution (allocating commercial airtime among spots, promos, and other content) and the development of advertising proposals that establish pricing and number of spots across various programs and weeks. Operational challenges include the scheduling of commercial ads and the management of spot-filler sales. Notably, this area of commercial ad scheduling has received significant attention from both academics and industry professionals; see Venkatachalam et al. (2015) and Souyris et al. (2023).

In contrast to revenue management applications, the adoption of advanced analytics techniques for

content scheduling applications has been slower for two main reasons: (1) these business processes are traditionally seen as closer to an art than a science and show considerable resistance to the adoption of data-driven methods, and (2) the complexity of content curation and optimization varies significantly among media channels. To illustrate this, consider three different media channels: music streaming services, national television networks, and broadcast radio. On the one hand, music streaming apps collect extensive consumer behavior data, enabling them to tailor personalized playlists to individual preferences; see Moor et al. (2023). Broadcast radio, however, must optimize playlists that appeal to hundreds of thousands of listeners without access to the rich, individualized data that music streaming services utilize. On the other hand, national television and cable networks air content with broad appeal to the entire country and, with some exceptions, such as local news, do not need to adjust their programming for local markets; see Horen (1980), Reddy et al. (1998), and Danaher and Mawhinney (2001). Radio playlists, however, must be both localized and individually programmed, preventing radio stations from benefiting from the centralized planning processes on which television and cable networks rely.

Notably, music scheduling in broadcast radio has been overlooked by the optimization and advanced analytics communities. To date, neither theoretical nor practical examples of applications in this context have been documented in the literature. This oversight highlights the reluctance of participants in this application area to embrace new technologies and advanced analytics approaches.

Music scheduling and ad scheduling share some commonalities because both applications determine the order in which media elements are aired, yet there are some notable differences that position music scheduling as a more nuanced problem, including the objective, the scale, and the constraints. Starting with their objectives, ad scheduling has clear goals: to maximize revenue from the advertising spots scheduled and (or) to maximize the audience exposure for advertisers. Both revenue and audiences can be quantified unambiguously. In contrast, music scheduling pursues more abstract goals such as enhancing the strength and diversity of a music log as these are metrics that are not universally defined and that are challenging to translate into mathematical models. Regarding scale, ad scheduling can be broken down into manageable sub-problems specific to certain shows or time slots called dayparts (see Table 1 for examples of dayparts) because of contractual obligations to air a set number of spots during those shows or dayparts. This decomposition allows for independent problem solving, which is not possible in music scheduling. Finally, whereas some constraints in both types of scheduling are similar, ad

Table 1. Examples of Dayparts

Daypart name	Period of time
AM drive	6 a.m.–10 a.m.
Midday	10 a.m.–3 p.m.
PM drive	3 p.m.–7 p.m.
Evening	7 p.m.–12 a.m.
Overnight	12 a.m.–6 a.m.

Notes. Dayparts are periods of time in a day during which TV and radio stations broadcast programs (Nielsen 2016). Audience composition varies across dayparts.

scheduling does not face cross-day and cross-week monotony-breaking rules or the particularly stringent segue rules that music scheduling does. Segue rules, set by program directors, govern the transitions between songs to maintain a desired musical flow, reflecting their expert judgment on what constitutes an effective schedule. These rules make it necessary to explicitly model the order of songs. We explore additional complexities and differences in music scheduling in further detail below.

The Music Playlist Creation Process

Creating music playlists for radio broadcasting involves a detailed and strategic process, primarily designed to meet listener preferences and comply with radio station policies. We now provide a simplified overview of this process, focusing on its most relevant components. The first step in this process is to understand the preferences of listeners who are being targeted by the radio station. Because of the nature of broadcasting, receiving direct feedback at scale is not possible; thus, radio broadcasters rely on third-party research companies to collect feedback and determine which songs and artists resonate most with their target audiences. Traditionally, the process of collecting feedback from listeners is called “callout” research. Callout research involves selecting a sample of songs and playing short, recognizable snippets to a representative group of listeners, typically via phone interviews or online surveys. Participants then rate each song based on a scale that includes *favorite*, *like*, *no opinion*, *developed dislike*, *negative*, and *unfamiliar*. These ratings are then consolidated into a so-called POP score, which is a number that measures the strength of the popularity of a song at the time of the research.

The second major step consists of analyzing the data from callout research to assess the songs’ popularity and listener fatigue. Radio stations use these insights to adjust their music rotations, promoting tracks that resonate well with their audiences and removing those that do not, thereby enhancing listener engagement and, subsequently, advertising revenue. Unfortunately, the callout research process is time-consuming and expensive; thus, it is only performed

for a sample of songs and stations. These limitations have traditionally meant that program directors must rely on their expertise and gut feeling to guess the popularity of the songs not covered by research. Our first application of advanced analytics in the music research context consisted of developing a series of machine learning models that estimated the popularity of songs at specific stations at different stages of their life cycles, thus filling in the gaps in callout research. One set of our machine learning models is a series of XGBoost regressions, one per music format, that produce a POP score estimate, that is, a synthetic POP score that can be calculated based on readily available information about a song, such as historical POP scores for the song and the artist, number of audio streams and video streams, sales, number of times it has been played in different stations both nationally and at the market level, and number of times it has been thumbed up or down in a music streaming app. These models provide program directors with unbiased, quantitative estimates about the popularity of all the songs they consider playing at their stations.

Synthetic POP scores as well as additional data points are used to determine which songs will be included in the planning horizon as well as their frequency or average rotation. Songs are meticulously classified according to the station’s format into predefined categories such as *power* (songs with the highest rotations) and *new* (recently released songs with the lowest rotations). These categories, standardized within each music format, are designed to be flexible and tailored to meet specific format needs. For example, the classic rock format might use either strength-based categories reflecting varying levels of popularity or era-based categories that align with the songs’ original release periods. Each format typically includes 12 to 16 different categories. Decisions are also made regarding the number of songs in each category and target average frequency for their airplay.

Finally, the third major step consists of determining the specific order in which the songs will be played during the planning horizon. The legacy music scheduling process utilizes a rules engine–based software to create music playlists for each station. Program directors provide the rules and restrictions for the software to follow and iteratively refine music playlists until they can make manual adjustments. However, there are several pain points associated with the legacy process, and these are summarized in Table 2.

Methods: Leveraging Mathematics to Craft Music Schedules

The primary goals and rules in creating music playlists aim to provide the best possible experiences for

Table 2. Pain Points Associated with the Legacy Process

Limited song information and flexibility
<ul style="list-style-type: none"> • The legacy system relies primarily on song categories and target rotations to determine song appearances in the playlist over the planning horizon. • It lacks the flexibility to integrate new market signals and (or) additional metrics. • The system is unable to incorporate scores from state-of-the-art machine learning models that provide a more granular understanding of a song’s strength or popularity throughout its life cycle.
Complex process and user parameters
<ul style="list-style-type: none"> • Users must set numerous parameters to set priorities around the restrictions on the schedule. • The system includes various alternatives to achieve similar goals, often confusing users about whether they are setting requirements correctly. • Users frequently need to redo schedules multiple times when trying different parameter values.
Subjectivity and bias
<ul style="list-style-type: none"> • The legacy process depends heavily on the program director’s expertise and intuition, introducing potential bias. • Different program directors can produce vastly different music logs from the same input data, causing inconsistencies. • Turnover among program directors can lead to abrupt changes in music schedules.
Scalability issues
<ul style="list-style-type: none"> • The process requires experts to categorize songs into appropriate frequency buckets, impeding the streamlining and automation of music scheduling. • This hampers the ability to support localized music scheduling with centralized planning.

radio listeners and strengthen relationships with record labels. To achieve this, the overarching approach is to maximize both the strength and diversity of the 24/7 music playlists, all adhering to specific business rules. This approach has led us to the use of mathematical optimization in our application.

As described in the “The Music Playlist Creation Process” section, synthetic POP scores estimated with machine learning models are agreed-upon metrics that quantify the strength of a song at a particular station. Quantifying diversity is more complex as different program directors consider various dimensions for different stations. However, song title diversity and artist/vocalist diversity are fundamental and consistent across all stations. Beyond these two dimensions, diversity can be assessed using various music codes as detailed in Table 3. To address this challenge, we designed and implemented a pick-and-rank mechanism that allows program directors to select and prioritize diversity dimensions

according to their station’s branding. This approach helps balance the scalability of the solution with the need for diverse preferences across different stations although it necessitates multiple iterations of model refinements and parameter tuning.

Furthermore, because of the size and complexity of the problem, we decompose it into two sequential phases: assignment and sequencing. Initially, the assignment phase determines which songs to play during each quarter hour over a one-week scheduling period. Program directors typically schedule one week’s worth of music at a time with quarter-hour segments serving as the building blocks of this schedule because they align with Nielsen’s radio audience measurement intervals; see Nielsen (2024). Following this, the sequencing phase establishes the playback order of the songs within each quarter hour.

Table 3. Diversity Music Codes and Example Values

Music code	Examples
Era	Late ‘60s, early ‘70s, late ‘90s, early ‘00s, etc.
Tempo	Slow, medium, fast, etc.
Mood	Sad, happy, etc.
Soundcode	Station-specific song markers that categorize the music sound of a song within a station, such as rock, pop, alternative, hard, or southern in a classic rock station
Genre	K-pop, dance, emo, etc.
Vocal	Male, female, duet
Content	Core, noncore, etc.
CRG	Current-recurrent-gold song markers for which current indicates newly released songs, recurrent indicates established popular songs with decreasing appeal, and gold indicates timeless hits

The Assignment Phase

During the assignment phase, we solve a mixed-integer, multicriteria optimization problem. The main decision variables determine whether to assign a song to a specific day, hour, and quarter hour slot, aiming to achieve several objectives: (1) strength goals, (2) song title diversity goals, (3) artist and vocalist diversity goals, (4) current-recurrent-gold (CRG) diversity goals, (5) other music code diversity goals (see Table 3 for examples), and (6) weekly play commitments to record label partners. Among these, the goals for song title diversity, artist and vocalist diversity, and weekly play commitments to record label partners are treated as hard constraints because of their strategic importance. The remaining goals are treated as soft constraints (SC).

Playing songs with the same synthetic POP score at different time slots, each with varying projected

Downloaded from informs.org by [216.73.216.31] on 06 June 2026, at 18:40. For personal use only, all rights reserved.

ratings, can significantly impact listenership. As a result, it is common practice to play the most popular songs during peak hours, such as the morning and afternoon drive times when listenership is typically highest. Additionally, because listeners tune in at different times and for various durations, program directors strive to create diverse playlists for each hour, daypart, and day. For instance, some program directors aim to ensure that listeners hear a mix of new, familiar, and classic tracks within any given daypart. To achieve these objectives, strength and diversity goals are also addressed at various levels of granularity during the assignment phase. This process is informed by extensive listener research accumulated over the years, and this includes both historical and projected Nielsen ratings for different stations and time frames throughout the week. It also incorporates intelligence gathered at the individual panelist level, detailing the specific days, times, and durations during which panelists listen to various stations.

Finally, some listeners consistently tune in at the same times and on consecutive days, and some listeners tune in on the same day of the week over several weeks. Thus, cross-day and cross-week playlist diversity is also desired. These rules and goals collectively contribute to crafting music playlists that provide the best possible listening experiences.

Specifically, the objective function for the assignment phase aims to maximize the total synthetic POP scores of the playlist, weighted by expected audience sizes at different times. At the same time, it seeks to minimize deviations from various soft constraints. These SC include (SC 1) achieving a desired mix of CRG song types within a given daypart; (SC 2) striving for an equitable distribution of songs sharing the same music code values, weighted by priority levels set by program directors; and (SC 3) avoiding monotonous playlist patterns by preventing the repetition of song sequences in a predefined look-back time window, such as the previous week. The hard constraints in the assignment model are categorized and detailed as follows:

- Scheduling songs at eligible times: Ensuring songs are played at appropriate times. For instance, slow and sad songs are excluded from morning drive hours.
- Maintaining song title diversity: This entails (1) ensuring desired time separation between plays of the same song and (2) preventing songs being repeated in the same hour or quarter hour on the next two business days to break monotony. This rule also applies to different versions of the same song, including covers. A song cover refers to a new performance or recording by an artist other than the original performer or composer.
- Maintaining artist/vocalist diversity: This entails (1) ensuring adequate time separation between two plays of songs by the same artist or vocalist and (2)

preventing artists/vocalists from being repeated in the same hour or quarter hour on the next two business days to break monotony. Although “artist” and “vocalist” are similar concepts, they differ when it comes to collaborations of multiple artists, which is common in formats such as hip-hop. For instance, hip-hop program directors enforce vocalist diversity, whereas program directors for other formats, such as classic rock, typically enforce artist diversity. Another complicating factor involves artist/vocalist groups, in which related artists or vocalists are grouped together. For instance, program directors want to separate songs by The Beatles from solo tracks by John Lennon.

- Maintaining play limits: Adhering to minimum and maximum plays for songs by day, daypart, and week. Day- and daypart-level play limits are designed to enhance song title diversity such that popular songs are played in each daypart without omission. Meanwhile, weekly play limits are crucial for meeting commitments to record label partners, ensuring that agreed-upon levels of exposure are maintained.

- Accommodating specialty programming needs: Honoring position-specific requirements, such as Twofer Tuesdays in which two songs by the same band or artist are played consecutively, or world premieres when newly debuted songs are played every hour as well as songs of specific duration during syndication shows.

- Following program directors’ segue rules: Aiming to govern transitions between songs, segue rules are crucial for maintaining the desired music flow of a radio station. These rules help reinforce the station’s identity and enhance its appeal to target listeners. For example, consider a station with a primary genre of rock and two subgenres, rock+ and rock–. Each song is categorized under one of these three labels. A typical segue rule might be no back-to-back rock+/rock– songs, stipulating that two songs labeled rock+ or two songs labeled rock– should not be played consecutively to prevent deviating from the central genre for too long. Thus, an appealing playlist might alternate tracks such as rock → rock+ → rock → rock– → rock, maintaining rock as the main theme and intermittently introducing songs from the subgenres to vary the playlist. In contrast to the broader constraints applied during the assignment phase, segue rules focus on these finer, more detailed aspects of playlist organization. Thus, they are only accounted for partially during the assignment phase and explicitly enforced during the sequencing phase.

The detailed mathematical formulation of the assignment model can be found in Appendix A.

The Sequencing Phase

The sequencing phase determines the playback order of the songs assigned to the same quarter hour. This

phase aims to construct a playlist that minimizes the total penalties incurred from violations of segue rules with these penalties weighted according to user-defined priorities. For example, consider the previously mentioned segue rule: no back-to-back rock+/rock− songs. Whereas the assignment phase ensures that the overall count of rock+ and rock− songs within each quarter hour stays below a certain threshold, the sequencing phase penalizes violations of this rule such that the number of occurrences of back-to-back rock+ or rock− songs is minimized. Because the flow of music is crucial to programming operations, program directors often establish dozens of segue rules based on specific song music codes to control music diversity on each radio station. Consequently, the sequencing phase can face complex segue rules involving chains of two, three, or more songs to meet the diverse preferences of program directors.

For sequencing methods, we initially employed heuristics during the proof-of-concept and pilot phases of development. This approach provided greater flexibility to adapt to dynamic business needs and was straightforward to explain to stakeholders, facilitating user adoption. The use of heuristics also enabled us to effectively translate the artistic flow of music into mathematical terms during the early stages when requirements were still evolving. Once user requirements stabilized, we developed a dedicated mathematical programming model for the sequencing phase. This transition aimed to minimize global segue violations and enhance computational efficiency.

We initially developed a scoring greedy heuristic algorithm to balance the trade-offs among all segue bans across multiple music codes. The algorithm sequentially builds the final music playlist in quarter-hour blocks, starting from the morning time daypart of the first day and progressing through the overnight daypart of the last day of the scheduling window. During each step of sequencing a quarter-hour block, if a segue rule violation is detected in the default order from the assignment phase, the algorithm searches for alternative song orders. For each potential order, it evaluates chains of two, three, or more songs based on every segue rule and calculates the total weighted score of violations. The song order with the minimum weighted score is selected as the final playlist of the quarter-hour block. This heuristic is flexible and extensible, and it simplified the translation of rules, assisting with user validation and adoption. This approach quickly delivered solutions and provided a mechanism for business users to clarify ambiguous requirements, such as quantifying a good music flow when not all segue rules are met as well as providing feedback to tune in the penalty weights across all music codes. Appendix B presents the pseudo-code of the heuristic.

After the initial models and heuristics were put into production, thoroughly tested, and stabilized, we identified an opportunity to enhance the sequencing phase with an optimization model. Developing a separate mixed-integer programming model for the sequencing phase offers several benefits. First, it improves the solution quality by evaluating segue rule violations holistically across the entire scheduling time window rather than the quarter-hour local optimization used in the heuristic method. Second, it enhances runtime efficiency. Because the heuristic involves evaluating permutations, its time complexity is $O(n!)$, where n equals the number of songs assigned to the quarter hour. Although these running times are generally acceptable in most business use cases—in which each quarter-hour segment contains up to five songs—certain edge cases requiring additional songs needed special handling in the heuristic to avoid extended runtimes.

Our initial optimization approach modeled the sequencing phase as a minimum cost Hamiltonian path problem with additional side constraints, represented over a network in which nodes corresponded to songs assigned to each quarter hour and arcs represented transitions within and across quarter hours. The arc costs quantified two-song segue violations, whereas the side constraints captured longer segue violations. Although this approach improved solution quality and was faster than the heuristics in several instances, it suffered from a weak linear programming relaxation, leading to excessively long running times in some routine scenarios.

A key insight that enabled us to generate a formulation improving both solution quality and running times was to shift from modeling segue violations using songs as building blocks to modeling them in terms of positions in the schedule. Although this new approach requires quantifying segue violations across each segue ban separately, it leads to a stronger formulation that finds the optimal solution in seconds. In this model, we use binary variables to decide whether a song is scheduled in a specific position in each quarter hour based on the quarter-hour song assignment information from the assignment phase. Segue violations are quantified using decision variables that track whether a segue violation occurs. The objective function minimizes the total violation of segue rules, weighted by their priorities, over the full scheduling time window. The constraints considered across all formats are

- Ensuring that each position in the quarter hour has a song scheduled.
- Ensuring that each song assigned to the quarter hour is given a position.
- Accounting for violations across segue bans that occur between groups of consecutively scheduled songs.

Appendix C provides the mathematical formulation of the baseline model. We omit explicit details of some additional constraints that are applicable in specific scenarios, such as those requiring differentiation in the direction of segue rules. For instance, the segue rule $\text{no rock+} \rightarrow \text{rock-}$ segue may have a different priority weight compared with the $\text{no rock-} \rightarrow \text{rock+}$ segue. Additionally, special programming, such as twofer events, necessitates further constraints.

Finally, tuning parameters in the objective function are necessary to set the proper priority weights to each of the numerous segue rules. We next discuss some details of our parameter tuning.

Tuning Sequencing Weights Using Meta Heuristics

The objective in the sequencing phase is to minimize the number of weighted segue violations across all music codes. Music codes (and sometimes specific music code values) have different priorities for each station and music format. For instance, a rock station may give more importance to segue violations involving eras (e.g., no three consecutive songs from the early '90s, late '90s, or early '00s) and lower importance to violations involving tempo (e.g., no back-to-back songs with slow tempo), whereas a CHR station may prioritize segue violations involving slow tempo over segue violations involving vocals (e.g., no three consecutive songs with male-only vocals). The importance of each goal is represented by weights on the auxiliary variables quantifying segue violations for each music code or music code value as appropriate. Although the greedy sequencing heuristic established a baseline for setting these priority weights, a goal-coefficient smart search approach, which combines a meta-heuristic with mathematical programming, was employed to fine-tune the weights or goal coefficients in the model's objective function.

Initial weights in the sequencing phase are stored and set as the goal coefficients; after that, the model is used to generate music logs for a set of test instances to generate an associated performance metric that is compared with the corresponding metric of benchmark logs (i.e., high-quality logs on which program directors agree). The deviation is stored as the initial and best value found so far, and then, a perturbation is created among the goal coefficients to change the status of the system. A new set of music logs is created and compared with the set of real instances to estimate its deviation; if the deviation is smaller or any other search criteria is met, parameters and system status are updated and/or stored. The process of perturbation-system update is repeated until the algorithm reaches a predefined number of iterations. The weight coefficients that show the smallest deviation in

a validation phase are the tuned coefficients for the sequencing model.

Implementation Challenges

Our solution comprises a two-step optimization engine integrated into a music scheduling application. This engine is customized to cater to specific musical genres and data utilization methods commonly employed by program directors. Typically, a program director manages stations within one or similar music formats, and our engine can accommodate these specificities. The engine utilizes standardized inputs and outputs across music formats, but it incorporates unique logic within the mathematical models and algorithms to address the diverse programming needs of each format. Figure 1 shows the end-to-end flow of information in the application.

We overcame several challenges to make the application of our models a success. In the remainder of this section, we describe a few major challenges and our approaches to address them.

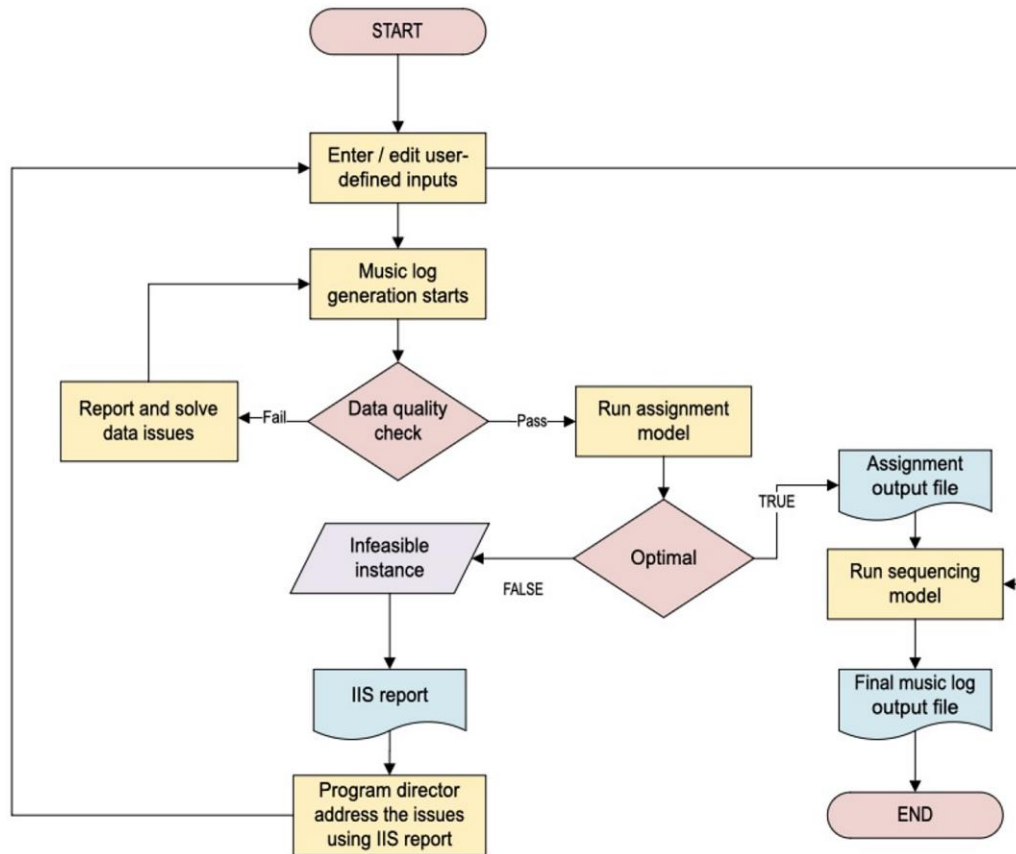
Dealing with Ambiguous Requirements

Criteria such as a strong log and good music flow are inherently subjective and vary among stakeholders. To address this, we used qualitative research methods, including interviews and focus groups to gather insights on these subjective criteria. We also ensured that our models are extendable and can accommodate metrics of music strength other than the synthetic POP scores that we had leveraged so far. The initial use of heuristics in the sequencing phase also afforded us flexibility and speed to validate our results and receive ongoing feedback based on evolving preferences.

Enforcing Data Governance

The legacy process was less sensitive to data quality issues because users often spent considerable time iterating and manually adjusting playlists or over-constraining specific songs when creating schedules. In some extreme cases, the legacy software was used merely as a template with schedules being manually entered one song at a time without utilizing the scheduling functions. During our proof-of-concept phase, we observed that, although our optimization framework delivered strong results for stations with clean data, it produced suboptimal results for stations with messy data. By quantifying the performance improvements achieved at stations with good data, we were able to build a business case that encouraged our business partners to devote time and resources to a data-alignment project aimed at cleaning and standardizing the input data for our models.

Additionally, we implemented a concept called a data contract when integrating with upstream and

Figure 1. (Color online) The Flow of Information Through the Music Scheduling Application

Note. The IIS report identifies the minimal set of constraints and bounds responsible for the infeasibility, which is then used to explain the root cause to the program director, enabling corrective action.

downstream processes to ensure that data passed to the models are concise, accurate, and robust. The concept, originated from the data engineering field, involves a formal agreement between two parties, often a client and a server, defining the structure and format of the exchanged data. It specifies data elements, their types, and any validation rules or constraints.

Incorporating Explainability of Infeasible Instances

The success of the music scheduling application is boosted by its ability to inform users when their chosen combination of constraints and parameters leads to an infeasible solution. The application utilizes the solution status provided by its solver to determine the appropriate actions for the application's output and to communicate relevant information to the end user.

When an infeasible status is detected, the application runs an irreducible infeasible set (IIS) analysis. The IIS represents the smallest set of constraints and variable bounds that, when one is removed, make the previously infeasible mathematical program feasible. The results of the IIS analysis are compiled into an IIS report, which is then processed to generate a message

that identifies the root cause of the infeasibility. The message is communicated to the program director through the user interface, providing feedback on the issue and prompting the user to adjust the parameters or constraints. The process might be repeated until the optimization yields a feasible solution.

Shortening Assignment Phase Runtimes with a Rolling Horizon Approach

Given the size and complexity of our assignment models, we devised a rolling horizon approach to reduce solution times, maintaining the desired solution quality. This process defines a rolling horizon that initially covers only a few days (typically one to three) and optimizes song assignments within this shorter period. The song assignments for the first day of the rolling horizon are then fixed, the horizon is expanded by one day, and song assignments over the expanded horizon are optimized. At each step, the song assignments of an additional day are fixed, and the rolling horizon expanded by one day. The process continues until the rolling horizon spans the whole planning horizon, typically a week, and all song assignments are established. The length of the rolling

horizon is fine-tuned for each specific music format and station to optimize both running times and solution quality.

Impact of the Implementation

The music scheduling application that serves as the user interface to our optimization engine was successfully deployed to production at the end of 2023. Since then, it has been integrated into more than 100 stations across three formats with plans to roll it out to additional stations as well as to other formats in the coming months. The application has made an impact in the following key areas:

- **Simplified the scheduling process:** The number of parameters that program directors need to input was reduced by a factor of 10. Additionally, in the application user interface, many parameters come with recommended ranges or default settings, which are informed by data and algorithms. This simplification addresses previous challenges with legacy systems in which parameter settings were complex and burdensome, and these were aggravated by staff turnovers, leading to excessive manual efforts to create quality playlists. Furthermore, this simplification allows program directors to concentrate on the artistic and strategic aspects of music programming, letting the optimization engine identify the best playlists that balance all their scheduling requirements.

- **Increased the process efficiency:** Most stations can now generate a seven-day music playlist in less than 10 minutes, a marked improvement from the past, in which the same process could take several hours. This efficiency allows program directors to more freely experiment with different scheduling scenarios.

- **Enabled the use of localized music intelligence at scale:** The legacy system was unable to directly ingest music research data and state-of-the-art predictive analytics that provide up-to-date estimates of song performance for specific stations. Our optimization engine was designed to exploit this localized intelligence with flexibility to accommodate additional metrics that may be developed in the future.

- **Enhanced music playlist quality:** The quality of music playlists consistently improved because of the integration of the latest song performance intelligence and the transition from a rule-based to an optimization-based engine. On average, the strength of music schedules was increased by 20% and the number of segue violations was decreased by 24%. Moreover, the increased quality of produced playlists also reduces the manual effort needed to fine-tune music playlists postgeneration.

- **Decreased playlist subjectivity:** Built around an optimization engine and other advanced analytics algorithms, the music scheduling application reduces

much of the human subjectivity in the music scheduling process. This ensures that playlists generated by different program directors using the same input data are more robust and consistent.

These improvements underscore the application's effectiveness in boosting the operational efficiency and quality of music scheduling across radio stations.

Appendix A. Assignment Phase Optimization Formulation

This appendix details the optimization model in the assignment phase, including sets, parameters, derived sets and parameters, decisions variables, and model formulation.

A.1. Sets

- \mathcal{D} : The set of all days in the scheduling time window indexed by d .

- \mathcal{H} : The set of all hours in each day indexed by h .

- \mathcal{Q} : The set of quarter hours in each hour indexed by q .

- \mathcal{S} : The set of all songs in the song library indexed by s

- \mathcal{A} : The set of all artists with songs in the song library indexed by a .

- $\mathbb{A} = \{\mathcal{A}'_1, \mathcal{A}'_2, \dots\}$: The set of all user-defined artist groups. Let $\mathcal{A}_{\text{other}}$ denote artists not associated with any artist group. \mathbb{A} is a partition of $\mathcal{A} \setminus \mathcal{A}_{\text{other}}$.

- \mathcal{V} : The set of all vocalists with songs in the song library indexed by v .

- $\mathbb{V} = \{\mathcal{V}'_1, \mathcal{V}'_2, \dots\}$: The set of all vocalist groups. Let $\mathcal{V}_{\text{other}}$ denote vocalists not associated with any vocalist group. \mathbb{V} is a partition of $\mathcal{V} \setminus \mathcal{V}_{\text{other}}$.

- \mathcal{C} : The set of all music codes program directors create for songs indexed by c . Note that CRG is not a member of \mathcal{C} because the type of requirements for CRG differs from other music codes. CRG is defined in Table 3 and in the "Parameters" section below.

- \mathcal{M} : The set of current, recurrent, and gold categorization indexed by m .

- \mathcal{G} : The set of all song groups determined by program directors indexed by g .

- \mathcal{P} : The set of all Nielsen dayparts indexed by n .

- $\tilde{\mathcal{V}}_s$: The set of vocalists with which song s is associated.

- \mathcal{B} : The set of segue bans indexed by b .

- $\Omega_s = \{(d_1, h_1, q_1), (d_2, h_2, q_2), \dots\}$: The set of all quarter hours when song $s \in \mathcal{S}$ is eligible to air indexed by (d, h, q) .

- (specialty programming) $\tilde{\mathcal{H}} = \{(d_1, h_1), (d_2, h_2), \dots\}$: The set of all hours when twofer programming is requested indexed by (d, h) . Hours with position-level requirements such as twofer that are less than 10% in a week.

- (specialty programming) \mathcal{L}_{dh} : The set of all twofer locations within hour $(d, h) \in \tilde{\mathcal{H}}$ indexed by l . In each twofer location, two or more songs from the same artist are played consecutively.

- (specialty programming) \mathcal{T}_{dh} : The set of all song positions within hour $(d, h) \in \tilde{\mathcal{H}}$ indexed by p . Song positions define the playing order of twofer songs.

A.2. Parameters

- G_s : The song group/category to which song $s \in \mathcal{S}$ is assigned by program directors.

- M_s : The current, recurrent, gold categorization for song $s \in \mathcal{S}$. It is also called the CRG label for song s .
- A_s : The artist with which song $s \in \mathcal{S}$ is associated. Each song is associated with one artist.
- C_{cs} : The value of code $c \in \mathcal{C}$ for song $s \in \mathcal{S}$.
- H_s : The strength score for song $s \in \mathcal{S}$.
- N_{dhgs} : The desired number of times for song group $g \in \mathcal{G}$ to play in each day $d \in \mathcal{D}$, hour $h \in \mathcal{H}$, and quarter hour $q \in \mathcal{Q}$.
- (specialty programming) N'_{dhqlp} : The number of twofer calls in each (d, h, q) , where $(d, h) \in \tilde{\mathcal{H}}$, location $l \in \mathcal{L}_{dh}$, and position $p \in \mathcal{T}_{dh}$.
- R_{dhq} : Listener ratings for each day $d \in \mathcal{D}$, hour $h \in \mathcal{H}$, and quarter hour $q \in \mathcal{Q}$.
- P_{dhq} : The daypart with which each $d \in \mathcal{D}$, hour $h \in \mathcal{H}$, and quarter hour $q \in \mathcal{Q}$ is associated. That is, $P_{dhq} \in \mathcal{P}$.
- T_s^0 : The number of minimum quarter hours to separate two same-day airings of song $s \in \mathcal{S}$.
- T_s^1, T_s^2 : The number of minimum quarter hours to separate song $s \in \mathcal{S}$ from when it airs the previous day and two previous days, respectively.
- E_a^0, F_v^0 : The number of minimum quarter hours to separate two same-day airings of artist a who is not assigned to any artist groups and vocalist v who is not assigned to any vocalist groups, respectively.
- E'_j, F'_j : The number of minimum quarter hours to separate two same-day airings of artists from artist group $\mathcal{A}'_j \in \mathbb{A}$ and two same-day airings of vocalists from vocalist group $\mathcal{V}'_j \in \mathbb{V}$, respectively.
- E_a^1, F_v^1 : The number of minimum quarter hours to separate artist $a \in \mathcal{A}$ and vocalist $v \in \mathcal{V}$, respectively, from when the artist's songs air the previous day.
- L_{ds}, U_{ds} : The minimum and maximum times to play song $s \in \mathcal{S}$ on day $d \in \mathcal{D}$.
- $\tilde{L}_{ns}, \tilde{U}_{ns}$: The minimum and maximum times to play song $s \in \mathcal{S}$ in daypart $n \in \mathcal{P}$.
- \tilde{L}_s, \tilde{U}_s : The minimum and maximum times to play song $s \in \mathcal{S}$ in the entire scheduling time window.
- J_{mn} : The percentage of songs with CRG label $m \in \mathcal{M}$ that a user desires in daypart $n \in \mathcal{P}$. This is also referred to as CRG goals.
- K_1, K_2 , and K_c : The user-entered priority ranking for total strength scores, for CRG goals, and for spreading music code $c \in \mathcal{C}$, respectively.
- B_b^{type} : The type of segue ban $b \in \mathcal{B}$. A segue ban type is a type of music code in \mathcal{C} . That is, $B_b^{\text{type}} \in \mathcal{C}$.
- B_b^{val} : The value of segue ban $b \in \mathcal{B}$, which is a set of one or more string values that program directors use to populate music code B_b^{type} for songs. Typically, $B_b^{\text{val}} \subset \{C_{B_b^{\text{type}}_s} | s \in \mathcal{S}\}$.
- B_b^{com} : The length of the segue chain for segue ban $b \in \mathcal{B}$. In this work, we consider $B_b^{\text{com}} \in \{2, 3, 4\}$.

A.3. Derived Sets and Parameters

- $w_1 \in \mathbb{R}_+$, $w_2 \in \mathbb{R}_+$, $w_c \in \mathbb{R}_+$: Nonnegative auxiliary parameters representing tuned weights for total strength scores, for CRG goals, and for spreading music code $c \in \mathcal{C}$, respectively. Note that function \mathcal{F} maps user-entered priority ranking to weights in the objective $\mathcal{F} : (K_1, K_2, K_c) \rightarrow (w_1, w_2, w_c)$.
- $\beta(B_b^{\text{com}}) \in (0, 1)$: Auxiliary parameters representing the threshold for segue ban combination B_b^{com} . As with w_1, w_2 , etc., these are parameters tuned by modelers.

- $\tilde{\Omega}_n = \{(d, h, q) | P_{dhq} = n\}$: The set of all quarter hours that map to daypart $n \in \mathcal{P}$.
- $\Omega_t^0 = \{(d_1, h_1, q_1), (d_2, h_2, q_2), \dots\}$: The set of all (d, h, q) that belong to the same-day cluster of separation interval $t \in \{T_s^0 | s \in \mathcal{S}\} \cup \{E_a^0 | a \in \mathcal{A}\} \cup \{F_v^0 | v \in \mathcal{V}\} \cup \{E'_j | \mathcal{A}'_j \in \mathbb{A}\} \cup \{F'_j | \mathcal{V}'_j \in \mathbb{V}\}$.
- $\Omega_t^1 = \{(d_1, h_1, q_1), (d_2, h_2, q_2), \dots\}$: The set of all (d, h, q) that belong to two-day cluster of separation interval $t \in \{T_s^1 | s \in \mathcal{S}\} \cup \{E_a^1 | a \in \mathcal{A}\} \cup \{F_v^1 | v \in \mathcal{V}\}$.
- $\Omega_t^2 = \{(d_1, h_1, q_1), (d_2, h_2, q_2), \dots\}$: The set of all (d, h, q) that belong to the three-day cluster of separation interval $t \in \{T_s^2 | s \in \mathcal{S}\}$.

A.4. Decision Variables

- $x_{dhqs} \in \{0, 1\}^{|\Omega_s|}$: A binary decision variable representing whether to assign song s to $(d, h, q) \in \Omega_s$.
- $y_{mn}^+ \geq 0, y_{mn}^- \geq 0$: Nonnegative decision variables representing positive and negative deviations from desired CRG spin percentage J_{mn} for CRG label $m \in \mathcal{M}$ and daypart $n \in \mathcal{P}$.
- $z_{dhci}^+ \geq 0, z_{dhci}^- \geq 0$: Nonnegative decision variables representing positive and negative deviations from average hourly airings of songs with music code $c \in \mathcal{C}$ assigned with value i for day $d \in \mathcal{D}$ and hour $h \in \mathcal{H}$. Note that i is any value used for music code c . That is, $i \in \{C_{cs} | s \in \mathcal{S}\}$.
- (specialty programming) $x_{dhqs} \in \{0, 1\}$: A binary decision vector representing whether to assign song s to (d, h, q) , where $(d, h) \in \tilde{\mathcal{H}}$ and position $p \in \mathcal{T}_{dh}$.
- (specialty programming) $\alpha_{dhla} \in \{0, 1\}$: A binary decision vector representing whether to assign artist $a \in \mathcal{A}$ to $(d, h) \in \tilde{\mathcal{H}}$ in a location $l \in \mathcal{L}_{dh}$.

A.5. Formulation

$$\max_{x_{dhqs} \in \{0, 1\}^{|\Omega_s|}} \left\{ w_1 \sum_{s \in \mathcal{S}} \sum_{(d, h, q) \in \Omega_s} H_s R_{dhq} x_{dhqs} - w_2 \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{P}} \frac{(y_{mn}^+ + y_{mn}^-)}{|\tilde{\Omega}_n|} - \sum_{c \in \mathcal{C}} \sum_{i \in \{C_{cs} | s \in \mathcal{S}\}} \sum_{d \in \mathcal{D}} \sum_{h \in \mathcal{H}} w_c (z_{dhci}^+ + z_{dhci}^-) \right\} \quad (\text{A.1})$$

subject to

$$\sum_{s \in \mathcal{S} | G_s = g} x_{dhqs} = N_{dhgs}, \quad \forall d \in \mathcal{D}, h \in \mathcal{H}, q \in \mathcal{Q}, g \in \mathcal{G}, \quad (\text{A.2})$$

$$\sum_{s \in \mathcal{S} | A_s = a} x_{dhqs} = N'_{dhqlp} \alpha_{dhla}, \quad \forall (d, h) \in \tilde{\mathcal{H}}, q \in \mathcal{Q}, l \in \mathcal{L}_{dh}, p \in \mathcal{T}_{dh}, a \in \mathcal{A}, \quad (\text{A.3})$$

$$\sum_{s \in \mathcal{S} | A_s = a, G_s = G_{dhqlp}} x_{dhqs} = N'_{dhqlp} \alpha_{dhla}, \quad \forall (d, h) \in \tilde{\mathcal{H}}, q \in \mathcal{Q}, l \in \mathcal{L}_{dh}, p \in \mathcal{T}_{dh}, a \in \mathcal{A}, \quad (\text{A.4})$$

$$\sum_{s \in \mathcal{S}} x_{dhqs} \leq 1, \quad \forall (d, h) \in \tilde{\mathcal{H}}, q \in \mathcal{Q}, p \in \mathcal{T}_{dh}, \quad (\text{A.5})$$

$$\sum_{a \in \mathcal{A}} \alpha_{dhla} = 1, \quad \forall (d, h) \in \tilde{\mathcal{H}}, l \in \mathcal{L}_{dh}, \quad (\text{A.6})$$

$$\sum_{p \in \mathcal{T}_{dh}} x_{dhqs} = x_{dhqs}, \quad \forall (d, h) \in \tilde{\mathcal{H}}, q \in \mathcal{Q}, s \in \mathcal{S}, \quad (\text{A.7})$$

$$\sum_{(d, h, q) \in \Omega_t^0} x_{dhqs} \leq 1, \quad \forall s \in \mathcal{S}, \quad (\text{A.8})$$

$$\sum_{(d,h,q) \in \Omega_1^1 | t=T_1^1} x_{dhqs} \leq 1, \quad \forall s \in \mathcal{S}, \quad (\text{A.9})$$

$$\sum_{(d,h,q) \in \Omega_1^2 | t=T_2^2} x_{dhqs} \leq 1, \quad \forall s \in \mathcal{S}, \quad (\text{A.10})$$

$$\sum_{(d,h,q) \in \Omega_1^0 | t=E_a^0} \sum_{s \in \mathcal{S} | A_s=a} x_{dhqs} \leq 1, \quad \forall a \in \mathcal{A} \setminus \left\{ \bigcup_{A_j \in \mathcal{A}} A_j \right\}, \quad (\text{A.11})$$

$$\sum_{(d,h,q) \in \Omega_1^0 | t=E_j^0} \sum_{s \in \mathcal{S} | A_s \in A_j'} x_{dhqs} \leq 1, \quad \forall A_j' \in \mathcal{A}, \quad (\text{A.12})$$

$$\sum_{(d,h,q) \in \Omega_1^1 | t=E_a^1} \sum_{s \in \mathcal{S} | A_s=a} x_{dhqs} \leq 1, \quad \forall a \in \mathcal{A}, \quad (\text{A.13})$$

$$\sum_{(d,h,q) \in \Omega_1^0 | t=F_v^0} \sum_{s \in \mathcal{S} | v \in \mathcal{V}_s} x_{dhqs} \leq 1, \quad \forall v \in \mathcal{V} \setminus \left\{ \bigcup_{\mathcal{V}_j' \in \mathcal{V}} \mathcal{V}_j' \right\}, \quad (\text{A.14})$$

$$\sum_{(d,h,q) \in \Omega_1^0 | t=F_j^0} \sum_{s \in \mathcal{S} | \mathcal{V}_s \cap \mathcal{V}_j' \neq \emptyset} x_{dhqs} \leq 1, \quad \forall \mathcal{V}_j' \in \mathcal{V}, \quad (\text{A.15})$$

$$\sum_{(d,h,q) \in \Omega_1^1 | t=F_b^1} \sum_{s \in \mathcal{S} | v \in \mathcal{V}_s} x_{dhqs} \leq 1, \quad \forall v \in \mathcal{V}, \quad (\text{A.16})$$

$$L_{ds} \leq \sum_{h \in \mathcal{H}} \sum_{q \in \mathcal{Q}} x_{dhqs} \leq U_{ds}, \quad \forall d \in \mathcal{D}, s \in \mathcal{S}, \quad (\text{A.17})$$

$$\tilde{L}_{ns} \leq \sum_{(d,h,q) \in \tilde{\Omega}_n} x_{dhqs} \leq \tilde{U}_{ns}, \quad \forall n \in \mathcal{P}, s \in \mathcal{S}, \quad (\text{A.18})$$

$$\hat{L}_s \leq \sum_{(d,h,q) \in \Omega_s} x_{dhqs} \leq \hat{U}_s, \quad \forall s \in \mathcal{S}, \quad (\text{A.19})$$

$$\sum_{s \in \mathcal{S} | C_b \text{ type } s \in B_b^{\text{val}}} x_{dhqs} \leq \beta(B_b^{\text{com}}) \sum_{s \in \mathcal{S}} x_{dhqs}, \quad \forall b \in \mathcal{B}, d \in \mathcal{D}, h \in \mathcal{H}, q \in \mathcal{Q}, \quad (\text{A.20})$$

$$\sum_{(d,h,q) \in \tilde{\Omega}_n} \sum_{s \in \mathcal{S} | M_s=m} x_{dhqs} - y_{mn}^+ + y_{mn}^- = J_{mn} \sum_{(d,h,q) \in \tilde{\Omega}_n} \sum_{s \in \mathcal{S}} x_{dhqs}, \quad \forall m \in \mathcal{M}, n \in \mathcal{P}, \quad (\text{A.21})$$

$$\sum_{s \in \mathcal{S} | C_{cs}=i} \sum_{q \in \mathcal{Q}} x_{dhqs} - z_{dhci}^+ + z_{dhci}^- = \frac{1}{|\mathcal{H}|} \sum_{s \in \mathcal{S} | C_{cs}=i} \sum_{h' \in \mathcal{H}} \sum_{q \in \mathcal{Q}} x_{dh'qs}, \quad \forall c \in \mathcal{C}, i \in \{C_{cs} | s \in \mathcal{S}\}, d \in \mathcal{D}, h \in \mathcal{H}, \quad (\text{A.22})$$

$$y_{mn}^+, y_{mn}^- \geq 0, \quad \forall m \in \mathcal{M}, n \in \mathcal{P}, \quad (\text{A.23})$$

$$z_{dhci}^+, z_{dhci}^- \geq 0, \quad \forall c \in \mathcal{C}, i \in \{C_{cs} | s \in \mathcal{S}\}, d \in \mathcal{D}, h \in \mathcal{H}. \quad (\text{A.24})$$

The objective function (A.1) maximizes the total synthetic POP scores of the playlist, weighted by expected audience sizes at different times, also minimizing the weighted penalties for deviating from desired CRG goals for each daypart and from the desired spread for various music code values. Constraints (A.2) ensure that a song is scheduled for each slot. Constraints (A.3)–(A.7) ensure compliance with twofer programming requirements. Constraints (A.8)–(A.10) enforce title diversity in terms of same-day time separation and up-to-two-business-day horizontal time separation. Constraints (A.11)–(A.13) ensure artist and artist group diversity in terms of same-day time separation and up-to-two-business-day horizontal time separation. Similarly, Constraints (A.14)–(A.16) ensure vocalist and vocalist group diversity in terms of same-day time separation and up-to-two-business-day horizontal

time separation. Constraints (A.17)–(A.19) enforce minimum and maximum spin limits by day, daypart, and week. Constraints (A.20) ensure that the desired play ratios are met for given music code values for which segue bans are specified. Constraints (A.21) address CRG goals, whereas Constraints (A.22) pertain to the spread goals for various music code values.

Appendix B. Sequencing Phase Heuristic Algorithm Pseudo-Code

This appendix contains a description of the heuristic initially developed for the sequencing phase, starting with notations followed by the pseudo-code. Once the assignment phase is completed, the song assignment decision in each quarter hour is fixed and passed as an input to the sequencing phase. Other inputs into the heuristic are the user-defined segue bans and their corresponding priorities.

B.1. Sets

- \mathcal{I} : The ordered set of all quarter hours in the entire planning horizon indexed by i . Each (d, h, q) can be mapped to $i \in \mathcal{I}$. For a 24x7 scheduling time window, $\mathcal{I} = \{1, 2, \dots, 168\}$.
- \mathcal{O} : The set of song order positions for the entire planning horizon indexed by o . For a music playlist with 2000 songs, for instance, $\mathcal{O} = \{1, 2, \dots, 2000\}$.
- Γ_i : Set of songs assigned to quarter hour $i \in \mathcal{I}$ from the assignment phase. That is, $\Gamma_i = \{s | x_{idhqs} = 1 \text{ where } (d, h, q) \rightarrow i\}$.
- Φ_i : All possible orders for songs assigned to quarter hour $i \in \mathcal{I}$ indexed by ϕ_i .

B.2. Parameters

- ϕ_i^0, ϕ_i^* : An initial order and final order for songs assigned to quarter hour $i \in \mathcal{I}$.
- O_i^{\min}, O_i^{\max} : The smallest position index and largest position index in quarter hour $i \in \mathcal{I}$.
- \tilde{w}_b : Penalty weight for segue ban $b \in \mathcal{B}$.
- $v_{\phi_i^b}$: Violation counts for segue ban $b \in \mathcal{B}$ of a specific song sequencing order $\phi_i \in \Phi_i$ given ϕ_{i-1}^* .
- ξ_{ϕ_i} : Violation score for a specific song sequencing order ϕ_i . Furthermore, $\xi_{\phi_i} = \sum_{b \in \mathcal{B}} \tilde{w}_b v_{\phi_i^b}$.
- ξ' : Tuned penalty threshold, which serves as stopping criteria in sequencing each quarter hour.

The pseudo-code for our sequencing heuristic is described in Algorithm B.1.

Algorithm B.1 (Greedy Sequencing)

- 1: **procedure** SEQUENCINGQUARTERHOUR($\phi_{i-1}^*, \phi_i^0, \Gamma_i, \xi'$)
- 2: $\phi_i^* \leftarrow \phi_i^0$
- 3: **if** $|\Gamma_i| > 1$ and $\xi_{\phi_i^*} > \xi'$ **then**
- 4: Create Φ_i based on Γ_i
- 5: **for** $\phi_i \in \Phi_i \setminus \phi_i^0$ **do**
- 6: **if** $\xi_{\phi_i} < \xi_{\phi_i^*}$ **then**
- 7: $\phi_i^* \leftarrow \phi_i$
- 8: **if** $\xi_{\phi_i^*} \leq \xi'$ **then**
- 9: Break
- 10: **end if**
- 11: **end if**
- 12: **end for**

```

13: end if
14: return  $\phi_i^*$ 
15: end procedure
16: procedure SEQUENCINGPLAYLIST( $I, \phi_i^0$  for  $\forall i \in I, \Gamma_i$  for  $\forall i \in I, \xi'$ )
17: for  $i \in \mathcal{I}$  do
18:   if  $i = 1$  then SEQUENCINGQUARTERHOUR( $[], \phi_1^0, \Gamma_1, \xi'$ )
19:   else SEQUENCINGQUARTERHOUR( $\phi_{i-1}^*, \phi_i^0, \Gamma_i, \xi'$ )
20:   end if
21: end for
22: end procedure

```

Appendix C. Sequencing Phase Optimization Formulation

This appendix describes the optimization model during the sequencing phase. Note that sets and parameters not explicitly defined in this appendix inherit their definitions from Appendices A and B.

C.1. Sets

- $\Gamma_{ib}^{\text{conflict}}$: The subset of songs assigned to quarter hour $i \in \mathcal{I}$ that can create a violation associated with segue ban $b \in \mathcal{B}$. That is, songs in $\Gamma_{ib}^{\text{conflict}}$ share values for music code $c \in \mathcal{C}$ associated with segue ban b . Typically, $\Gamma_{ib}^{\text{conflict}} \subset \{s \in \Gamma_i | C_{cs} \in B_b^{\text{val}}\}$.

C.2. Decision Variables

- y_{ios} : Binary variables representing whether to assign song s to quarter hour i in position o .
- v_{ob} : Binary variables representing whether segue ban $b \in \mathcal{B}$ has been violated by a chain of songs of length B_b^{com} with starting position $o \in \mathcal{O}$.

C.3. Formulation

$$\min \sum_{b \in \mathcal{B}} \sum_{o \in \mathcal{O}} \bar{w}_b v_{ob} \quad (\text{C.1})$$

subject to

$$\sum_{o \in \{O_i^{\min}, \dots, O_i^{\max}\}} y_{ios} = 1, \quad \forall i \in \{\mathcal{I} | |\Gamma_i| > 0\}, \quad s \in \{\Gamma_i\}, \quad (\text{C.2})$$

$$\sum_{s \in \Gamma_i} y_{ios} = 1, \quad \forall i \in \{\mathcal{I} | |\Gamma_i| > 0\}, \quad o \in \{O_i^{\min}, \dots, O_i^{\max}\}, \quad (\text{C.3})$$

$$\sum_{p \in \{1, \dots, B_b^{\text{com}}\}} \sum_{s \in \Gamma_{(o+p-1)^b}^{\text{conflict}}} y_{i(o+p-1)(o+p-1)s} \leq B_b^{\text{com}} - 1 + v_{ob}, \quad \forall b \in \mathcal{B}, o \in \{1, \dots, N - B_b^{\text{com}} + 1\}, \quad (\text{C.4})$$

$$y_{ios} \in \{0, 1\}, \quad \forall i \in \{\mathcal{I} | |\Gamma_i| > 0\}, \quad s \in \{\Gamma_i\}, \quad o \in \{O_i^{\min}, \dots, O_i^{\max}\}, \quad (\text{C.5})$$

$$v_{ob} \geq 0, \quad \forall b \in \mathcal{B}, o \in \mathcal{O}. \quad (\text{C.6})$$

The objective function (C.1) minimizes the total weighted penalties resulting from violations of the segue bans throughout the planning horizon. Constraints (C.2) and (C.3) ensure that, within each quarter hour, each song is assigned a position, and each position has a song assigned to it, respectively. Constraints (C.4) link the song position assignments to segue ban violations. For each segue ban and each position in the schedule, the right-hand side adds all the conflicting song

assignments across consecutive positions composing a chain of the same length as the segue ban. The sum is then limited to be greater than or equal to the ban length minus one plus the variable representing the segue violation. Thus, the only way that all positions in the chain can be filled with conflicting songs is if a violation is accounted for by the violation variable on the right-hand side. The notation $i_{(o)}$ represents the quarter hour index associated with position o . Note that variables v_{ob} are declared as continuous nonnegative variables; however, at optimality, they are guaranteed to take values in $\{0, 1\}$, which matches their definition as binary variables. This outcome is ensured by two observations: (1) Constraints (C.3) guarantee that the left-hand side of Constraints (C.4) will always be $\leq B_b^{\text{com}}$, ensuring that $v_{ob} \leq 1$, and (2) because Constraints (C.4) are independent, for any feasible solution in which v_{ob} takes fractional values ($0 < v_{ob} < 1$), there is always an alternative feasible solution at a lower cost with $v_{ob} = 0$.

References

- Carbajal JA, Char W (2017) Turner optimizes the allocation of audience deficiency units. *INFORMS J. Appl. Anal.* 47(6):518–536.
- Carbajal JA, Williams P, Popescu A, Char W (2019) Turner blazes a trail for audience targeting on television with operations research and advanced analytics. *INFORMS J. Appl. Anal.* 49(1): 64–89.
- Danaher PJ, Mawhinney DF (2001) Optimizing television program schedules using choice modeling. *J. Marketing Res.* 38(3):298–312.
- Horen J (1980) Scheduling of network television programs. *Management Sci.* 26(4):354–370.
- Moor D, Yuan Y, Mehrotra R, Dai Z, Lalmas M (2023) Exploiting sequential music preferences via optimisation-based sequencing. *Proc. 32nd ACM Internat. Conf. Inform. Knowledge Management (Association for Computing Machinery, New York)*, 4759–4765.
- Nielsen (2016) Glossary. Accessed May 1, 2025, <https://tapweb.nielsen.com/help/main/glossary.htm>.
- Nielsen (2024) Audio measurement. Accessed July 17, 2024, <https://www.nielsen.com/solutions/audience-measurement/audio/>.
- Reddy SK, Aronson JE, Stam A (1998) Spot: Scheduling programs optimally for television. *Management Sci.* 44(1):83–102.
- Souyris S, Seshadri S, Subramanian S (2023) Scheduling advertising on cable television. *Oper. Res.* 71(6):2217–2231.
- Venkatachalam S, Wong F, Uyar E, Ward S, Aggarwal A (2015) Media company uses analytics to schedule radio advertisement spots. *INFORMS J. Appl. Anal.* 45(6):485–500.

José Antonio Carbajal is a decision intelligence consultant with over 20 years of experience in analytics, data science, and operations research, applied to challenges in scheduling, revenue management, resource allocation, pricing, and forecasting. He has served in executive leadership roles, and his work has been recognized by INFORMS and the Association of National Advertisers. He holds an MS in operations research and a PhD in industrial engineering from the Georgia Institute of Technology.

Juan Ma is a technical manager of data science at iHeartMedia, Inc., specializing in decision-making challenges in ad sales, revenue management, content research, and digital advertising monitoring. She holds a PhD in industrial engineering and management from Oklahoma State University and a bachelor's degree in logistics systems engineering from Huazhong University of Science and Technology.

Nannan Chen is currently a senior operations research engineer at Hungryroot. She has applied her expertise in optimization and

advanced analytics on various scheduling, resource allocation, and forecasting problems across the airlines, media, and online retail industries. She earned her PhD in industrial engineering from Texas A&M University.

Mario Aboytes-Ojeda is a mechanical engineer and applied scientist with over 10 years of experience integrating machine

learning and optimization for applications in energy, logistics, media, and mechanical, electrical, and plumbing engineering. He holds a PhD in mechanical engineering from The University of Texas at San Antonio and an MS in operations research from the National Autonomous University of Mexico. He is an active INFORMS member and a published researcher.