TELEVISION COMMERCIAL SCHEDULING WITH INDUSTRY-SPECIFIC CONSTRAINTS AND AUDIENCE-BASED RATINGS

ULVI FINDIK

ÖZYEĞİN UNIVERSITY

AUGUST, 2025

TELEVISION COMMERCIAL SCHEDULING WITH INDUSTRY-SPECIFIC CONSTRAINTS AND AUDIENCE-BASED RATINGS

by
Ulvi Fındık

Thesis
Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

in
Industrial Engineering

Advisor: Prof. Ö. Erhun Kundakçıoğlu

Graduate School of Science and Engineering
Özyeğin University
İstanbul

August, 2025

TELEVISION COMMERCIAL SCHEDULING WITH INDUSTRY-SPECIFIC CONSTRAINTS AND AUDIENCE-BASED RATINGS

Approved by:

Prof. Ö. Erhun Kundakçıoğlu, Advisor
Department of Industrial Engineering
Özyeğin University

Assoc. Prof. Elvin Çoban
Department of Industrial Engineering
Özyeğin University

Assoc. Prof. Mustafa Baydoğan
Department of Industrial Engineering
Boğaziçi University

Approval Date: August 4, 2025

DECLARATION OF ORIGINALITY

I hereby declare that I am the sole author of this thesis and that this is the true copy of my thesis, including the final revisions, approved by my thesis committee. All data and information have been obtained, produced, and presented in accordance with the rules of research ethics and principles of academic honesty. As required by these rules, to the best of my knowledge I have acknowledged ideas, thoughts, and any copyrighted material in accordance with the standard referencing rules. I certify that any part of this thesis has not been submitted for a degree or diploma in another educational institution. *Signature*

Ulvi Fındık

ABSTRACT

Commercials serve as the primary revenue stream for Television (TV) networks. However, these networks must comply with sectoral and governmental regulations governing the scheduling and billing of advertisements. This study addresses the challenge of selecting and scheduling commercials for one of the most prominent TV networks in Turkey, where stringent regulatory constraints and audience-specific requirements complicate the process of finding even a feasible schedule, particularly under high utilization. Moreover, the revenue generated by most commercials is highly dependent on the ratings achieved from their target audience, further increasing the impact of the scheduling process. The commercial scheduling problem can be framed as a non-renewable resource-constrained identical parallel machine scheduling problem, characterized by machine- and time-dependent rewards and constraints, with the objective of maximizing the total reward. To tackle this intricate problem, we present two Mixed-Integer Programming (MIP) formulations and a novel Greedy Randomized Adaptive Search Procedure (GRASP) enhanced with path-relinking and Variable Neighborhood Descent (VND). Additionally, we reformulate the problem as a Team Orienteering Problem with Time Windows (TOPTW) and benchmark the GRASP against a Hybrid Artificial Bee Colony (HABC) algorithm. Experiments with 10 real-world instances demonstrate that the GRASP heuristic consistently achieves near-optimal solutions with minimal computational effort.

Keywords: TV commercials, scheduling, revenue management, combinatorial optimization, greedy randomized adaptive search procedure.

ÖZET

Televizyon (TV) kanalları için başlıca gelir kaynağını reklamlar oluşturmaktadır. Ancak bu kanallar, reklamların çizelgelenmesi ve faturalandırılmasıyla ilgili sektörel ve devlet kaynaklı düzenlemelere uymak zorundadır. Bu çalışma, Türkiye'nin en önde gelen TV kanallarından biri için, sıkı düzenleyici kısıtlar ve hedef kitleye özel gereksinimlerin, özellikle yüksek talep durumlarında uygulanabilir bir çizelge bulmayı dahi zorlaştırdığı reklam seçimi ve çizelgeleme problemini ele almaktadır. Ayrıca, reklamların sağladığı gelirin büyük ölçüde hedef kitleden elde edilen reytinglere bağlı olması, çizelgeleme sürecinin etkisini daha da artırmaktadır. Reklam çizelgeleme problemi; yenilenemeyen kaynak kısıtlı, özdeş paralel makineler üzerinde, makineye ve zamana bağlı ödül ve kısıtlarla tanımlanan bir çizelgeleme problemi olarak çerçevelendirilebilir ve toplam ödülün maksimize edilmesi amaçlanmaktadır. Bu karmaşık problemi çözmek amacıyla, iki farklı Karışık Tamsayılı Programlama (MIP) formülasyonu ile, path-relinking ve Variable Neighborhood Descent (VND) yaklaşımlarıyla geliştirilen özgün bir Greedy Randomized Adaptive Search Procedure (GRASP) sunulmuştur. Ayrıca, problem Team Orienteering Problem with Time Windows (TOPTW) olarak yeniden yapılandırılmış ve GRASP algoritması, Hybrid Artificial Bee Colony (HABC) algoritması ile karşılaştırılmıştır. Gerçek dünyadan alınan 10 örnek üzerinde gerçekleştirilen deneyler, GRASP algoritmasının düşük hesaplama yüküyle tutarlı biçimde optimal çözüme yakın sonuçlar elde ettiğini ortaya koymaktadır.

Anahtar Kelimeler: TV reklamları, çizelgeleme, gelir yönetimi, kombinatoryal optimizasyon, greedy randomized adaptive search algoritması.

ACKNOWLEDGEMENTS

I am deeply grateful to ATV for their generous support and collaboration throughout the development of this study. Their provision of real-world commercial scheduling data, which is not publicly available, was instrumental in both framing the problem and validating the developed models. Their collaboration enabled the practical grounding of this research, and their insights into the television broadcasting industry were crucial to its successful completion. Finally, I gratefully acknowledge the support of the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant 122M839, which made this research possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

TV	Television
PPR	Price Per Rating
MIP	Mixed-Integer Programming
GRASP	Greedy Randomized Adaptive Search Procedure
TOPTW	Team Orienteering Problem with Time Windows
HABC	Hybrid Artificial Bee Colony
TOP	Team Orienteering Problem
TOPTW-TDS	Time Windows and Time-Dependent Scores
VND	Variable Neighborhood Descent
RCL	Restricted Candidate List
TOP	Team Orienteering Problem

1. INTRODUCTION

Advertising, as noted by Stephan Loerke, CEO of the World Federation of Advertisers, plays a crucial role in the economy by fostering competition, driving innovation, and funding media services such as news and entertainment [1]. Alongside these benefits, TV plays a significant role in delivering advertising messages. Although TV is primarily an entertainment medium, it takes advantage of the large number of viewers to convey advertising messages. Their core operation involves building audiences to be purchased by firms and groups looking to reach out to customers.

Historically, TV programs are often sponsored entirely by advertisers. However, it is now rare for a single advertiser to sponsor an entire program. Instead, networks sell ad time slots during entertainment segments, which allows them to maximize revenue. For instance, during a typical broadcast season, major advertisers purchase dozens or even hundreds of commercial slots [2, 3, 4]. Consequently, media broadcasting companies, including TV and radio stations, predominantly generate their revenue from selling advertising space, with commercial advertising being a key revenue driver for TV channels [4].

[5] reports that TV remains a globally trusted advertising platform, even after a decline in 2020. Global TV ad revenue has returned to pre-pandemic levels, with 67% of global brands increasing spending on advanced and connected TV (internet-enabled platforms such as smart TVs, streaming devices, and data-driven ad delivery), and 11% on linear TV (traditional broadcast or cable television with scheduled programming). In North America, TV ad spending reaches nearly $67 billion in 2022, up from $64 billion in 2021, with notable growth in the U.S. and Mexico, though Canada lags behind. Brazil leads in South America, with over $6 billion spent in 2021, surpassing internet spending in countries like Argentina and Peru. Despite a steady decline in spending since 2016, trust in TV advertising remains high in the UK. In Turkey, commercial revenues account

for 96.3% of national TV channel revenues and 79.7% of regional TV channel revenues as of 2017 [6]. Additionally, in 2020, investments in media and advertising in Turkey reach 17.47 billion Turkish Liras ($2.48 billion). On a global scale, advertising spending hits nearly $733 billion in 2023, marking a 2.6% increase from the previous year, and is projected to exceed $1.75 trillion in 2024 [7]. These figures highlight the essential role of TV broadcasting, particularly in developing countries, where substantial investments in TV networks are well-justified by the significant revenue generated from commercials.

Program audience ratings are widely used as a reference in placing advertisements and negotiating advertising prices between TV stations and advertisers [8, 9, 10]. Networks prioritize commercials with higher prices, allocating them to prime-time slots and high-viewership events, such as major sports broadcasts, which attract a broader audience. For instance, national networks like NBC and CBS command significantly higher prices for prime-time slots due to their extensive reach and audience size, making these slots more expensive and preferred over less popular, lower-cost time slots like daytime or non-peak hours.

To fully understand the challenge of revenue maximization objective, it is crucial to grasp the pricing mechanisms behind TV commercials. An expert with experience in commercial scheduling for a major Turkish TV network [11, 12] identifies two primary pricing options: (i) Price Per Rating (PPR), and (ii) fixed price. The majority of TV commercials follow the PPR model, where the cost of the advertisement is determined based on the rating points that the target audience generates. The *unit price* is agreed upon beforehand, and the price is multiplied by the rating points achieved by the target audience during the advertisement's broadcast. It is important to note that the rating points are forecasted by the TV channels and are assumed to remain constant within a specified time window, typically defined on a per-minute basis during the commercial break. In contrast, in fixed-price advertising, a predetermined and unchanging price is set for the advertisement regardless of the target audiences' rating or other variables. The cost is fixed and does not fluctuate based on audience metrics.

TV channels often tackle the commercial scheduling problem multiple times a day, where even minor discrepancies in optimization can lead to significant revenue losses, potentially amounting to millions of dollars annually for major networks like ATV, the most-watched TV channel in Turkey in 2023 [11, 12]. As a result, any proposed solution must be both computationally efficient and capable of consistently delivering high-quality outcomes. The complexity of the problem is compounded by stakeholder demands and stringent government regulations. These insights draw on the extensive experience of the third author in the televised media industry.

To address these challenges, we:

- develop two MIP models to manage the complexities of the commercial scheduling problem,
- propose a novel GRASP that consistently outperforms a commercial solver under realistic time limitations across all tested instances,
- reformulate the problem as the well-known TOPTW and conduct a performance comparison between the GRASP algorithm and the HABC algorithm, and
- make our source code and sample datasets publicly available to facilitate future academic contributions and benchmarking.

The remainder of this paper is structured as follows: Chapter 2 reviews related literature, Chapter 3 introduces the problem formulation and our MIP model, Chapter 4 describes the GRASP algorithm, and Chapter 5 presents computational results demonstrating the effectiveness of our approach. Chapter 6 concludes the paper with directions for future research.

2. LITERATURE REVIEW

Research on scheduling problems in the TV industry has spanned decades, with most of the literature focusing primarily on program scheduling rather than advertisements [13, 14, 15, 16]. The main focus of these works is optimizing audience ratings, with additional studies like those by [17] offering insights into prime-time TV program scheduling.

In contrast, the scheduling of TV spots has received relatively less attention. The early work by [18] addresses the challenges of manually scheduling TV spots, while [19] explores the effectiveness of different advertising schedules through simulation. Following this, research diversified, with various authors addressing advertisement scheduling from perspectives such as content, viewer interest, and sponsor preferences, often focusing on the optimal timing of ads [20, 21, 22].

A significant advancement in this area comes from [23], who pioneer the application of mathematical programming to TV advertisement scheduling. Their study introduces an algorithm for NBC that optimizes ad placement, generating substantial revenue gains. This work evolves with further studies by [2] and [24], which propose mixed-integer programming and heuristic methods to evenly distribute commercials and prioritize ad placements at the start and end of breaks for higher audience ratings. These models set the foundation for subsequent research focused on integrating constraints such as budget and ad frequency to improve overall scheduling effectiveness [25].

Building on these foundational studies, other researchers like [26] and [27] expand the scope by addressing problems such as ad allocation through hierarchical approaches and the TV break packing problem, respectively.

Subsequent works by [28, 29, 30] introduce various heuristic methods, including ant colony optimization and genetic algorithms, to tackle the challenges of commercial scheduling. These approaches are further refined by [4], who explore combinatorial auction models for prime-time ad slots.

Despite advancements in scheduling algorithms, a significant gap remains in the literature: no study specifically addresses the assignment and scheduling of commercials in commercial breaks. This critical aspect of advertisement scheduling, which directly influences revenue generation and adherence to advertiser requirements, has been largely overlooked. Our research fills this gap by developing approaches to optimize commercial placement and scheduling within breaks, balancing revenue maximization with the constraints imposed by scheduling requirements.

The problem we address can be categorized as a non-renewable resource-constrained identical parallel machine scheduling problem with machine- and time-dependent rewards, aiming to maximize total rewards. Each machine operates under limited processing time, and all machines share a common resource constraint: the hourly broadcasting limit. This complexity is further compounded by additional constraints, such as advertiser preferences and sequential constraints making it a challenging and relevant issue for the TV industry.

While identical parallel machine scheduling has been extensively studied, existing algorithms do not directly apply due to the unique and unconventional constraints in commercial scheduling. For instance, the problem lacks traditional precedence constraints but includes a distinct restriction against consecutive scheduling of commercials from the same group, such as those within the same product category or from competing brands. Additionally, the time- and machine-dependent reward associated with each job introduces further complexity. While variations in time-dependent setup times [31, 32] and time-dependent processing durations [33, 34] are well-documented, the concept of time-dependent costs or rewards remains largely unexplored. Another distinction from conventional machine scheduling is that jobs are optional rather than mandatory, adding the challenge of not only scheduling but also selecting which commercials to broadcast to optimize total rewards within resource constraints.

To effectively address this complex scheduling problem, we employ GRASP. Initially introduced by [35], GRASP is applied to a wide range of scheduling problems. Early applications include single-machine scheduling with flow time and earliness penalties [36], as well as setup costs and delay penalties [37]. Later, the algorithm is also adapted to more complex settings such as job shop scheduling [38, 39, 40], flexible job shop scheduling [41, 42, 43], dynamic job shop scheduling [44] and flow shop [45]. The literature also presents GRASP applications in parallel machine scheduling under various problem settings. These include scenarios with time windows [46], sequence-dependent setup times with total tardiness objective [47], and settings involving setup costs and additional limited resources [48].

Other notable applications of GRASP include project scheduling [49], production scheduling [50], bus driver scheduling [51], technician and intervention scheduling [52], transporter scheduling [53], and tactical harvest planning [54]. However, to the best of our knowledge, no GRASP algorithm has been applied to a scheduling problem comparable to commercial scheduling, which presents unique constraints and objectives. Next, we define the specific characteristics of the commercial scheduling problem and outline the mathematical formulation used to model it.

3. PROBLEM FORMULATION

Advertisers often impose strict requirements on the scheduling of their commercials, including precise preferences regarding the placement of ads within a commercial break, specific broadcast timing, and restrictions on airing commercials from the same group nonconsecutively. Additionally, government regulations introduce further complexity, commonly limiting the total duration of advertisements to 12 minutes per hour, accounting for 20% of the total broadcast time [55].

The efficient allocation of advertisement time slots is crucial for maximizing revenue while ensuring compliance with regulatory constraints. The proposed solution must adeptly manage the schedule and timing of commercials, accounting for time-sensitive ratings to optimize revenue generation, all while addressing the diverse requirements of multiple stakeholders.

In this study, we define commercials as *flagged* if they have at least one specific sequencing or timing preference set by advertisers, which we term as *flags*. While it is not mandatory for a commercial to have a flag, any flagged commercial will have at least one scheduling requirement tied to a particular commercial break. Crucially, a flagged commercial may only have one flag per break. For instance, a commercial may be required to air first in one break, while a different flag for the same commercial could dictate that it must be aired within the first 30 seconds of another break. The types of flags considered in this study are as follows:

- `F`: The commercial must be aired as the `FIRST` commercial of the break.
- `L`: The commercial must be aired as the `LAST` commercial of the break.
- `M`: The commercial must be aired within the first `MINUTE` of the break.
- `H`: The commercial must be aired within the first `HALF-MINUTE` of the break.

Adhering to these flags is critical when selecting commercials for airing. Failure to meet these requirements results in an infeasible solution, potentially leading to contractual penalties or lost revenue, thereby underscoring the importance of precise scheduling.

Table 3.1: *Sets and parameters.*

$\mathcal{C}$	Set of commercials
$\mathcal{I}$	Set of inventories
$\mathcal{G}$	Set of product groups that are advertised
$\mathcal{C}_i$	Set of commercials suitable to be aired in commercial break $i \in \mathcal{I}$
$\mathcal{I}_c$	Set of inventories suitable for commercial $c \in \mathcal{C}$
$\mathcal{W}_i$	Set of time windows in commercial break $i \in \mathcal{I}$
$\mathcal{N}_i$	Set of available slots in commercial break $i \in \mathcal{I}$
$\mathcal{H}$	Set of hours in the planning horizon
$\mathcal{I}_h$	Set of commercial breaks (referred to as inventories) $i \in \mathcal{I}$ that start in hour $h \in \mathcal{H}$
$\mathcal{C}^f$	Set of commercials with flag f, $f \in \{\mathtt{F}, \mathtt{L}, \mathtt{H}, \mathtt{M}\}$
Δ_i	Total available broadcast time for commercial break $i \in \mathcal{I}$
d_c	Duration of commercial $c \in \mathcal{C}$ in seconds
p_{ciw}	Revenue of commercial $c \in \mathcal{C}$, if aired in commercial break $i \in \mathcal{I}$, time window $w \in \mathcal{W}_i$
g_c	Group of commercial $c \in \mathcal{C}$
α_{ci}	Flag of commercial $c \in \mathcal{C}$ for commercial break $i \in \mathcal{I}$
s^H	Duration of H flag (e.g., 30 sec)
s^M	Duration of M flag (e.g., 60 sec)
s^O	Maximum total hourly commercial broadcast duration limit (e.g., 720 sec)

In this problem context, there are three distinct sets: a set of commercials $\mathcal{C}$, a set of commercial breaks $\mathcal{I}$, and a set of product groups $\mathcal{G}$. Each commercial belongs to a specific group within $\mathcal{G}$, and commercials from the same group cannot be aired consecutively. The revenue generated by airing a commercial c during a specific time window w within a commercial break i is represented by p_{ciw} and is assumed to be predetermined, based on forecasted rating values for the target audience during that time window. Additionally, each commercial is aimed at a particular audience segment, which means it should be scheduled in commercial breaks most likely to reach that audience, as determined by the stakeholders. Consequently, for each commercial $c \in \mathcal{C}$, there is a corresponding set of suitable commercial breaks, denoted as $\mathcal{I}_c$.

The objective of the problem is to maximize total revenue generated while simultaneously adhering to advertiser preferences and ensuring compliance with government regulations. We propose two formulations to address the distinct challenges posed by this problem, each providing a unique approach to its intricacies.

3.1 Continuous-Time Formulation

Table 3.2: *Decision variables.*

O_{cin}	Indicator variable for commercial $c \in \mathcal{C}$ being aired in break $i \in \mathcal{I}$ at n^{th} slot
Z_{ciw}	Indicator variable for commercial $c \in \mathcal{C}$ being aired in break $i \in \mathcal{I}$ in time window $w \in \mathcal{W}_i$
S_{cin}	Start time of commercial $c \in \mathcal{C}$ in break $i \in \mathcal{I}$ at n^{th} slot

The continuous-time formulation monitors time within a continuous framework, utilizing the continuous variable S_{cin} to represent the precise airing time of each commercial. The decision variable O_{cin} tracks the slot in which a commercial is aired within a specific commercial break, while the binary variable Z_{ciw} indicates the time window in which the commercial is scheduled to air. Together, these variables ensure that the model optimizes revenue while satisfying both advertiser specifications and regulatory constraints.

$$\max \sum_{c \in \mathcal{C}} \sum_{i \in \mathcal{I}_c} \sum_{w \in \mathcal{W}_i} p_{ciw} Z_{ciw} \tag{3.1a}$$

$$\text{s.t.} \sum_{i \in \mathcal{I}_c} \sum_{n \in \mathcal{N}_i} O_{cin} \leq 1, \qquad c \in \mathcal{C} \tag{3.1b}$$

$$\sum_{c\in\mathcal{C}} O_{cin} \leq 1 \qquad i \in \mathcal{I}, n \in \mathcal{N}_i \qquad (3.1c)$$

$$\sum_{c\in\mathcal{C}_i} O_{cin} \geq \sum_{c\in\mathcal{C}_i} O_{(ci(n+1))}, \qquad i \in \mathcal{I}, n \in \mathcal{N}_i \qquad (3.1d)$$

$$\sum_{n\in\mathcal{N}_i} \sum_{c\in\mathcal{C}_i} d_c O_{cin} \leq \Delta_i, \qquad i \in \mathcal{I} \qquad (3.1e)$$

$$\sum_{i\in\mathcal{I}_h} \sum_{c\in\mathcal{C}_i} \sum_{n\in\mathcal{N}_i} d_c O_{cin} \leq s^O, \qquad h \in \mathcal{H} \qquad (3.1f)$$

$$S_{cin} \leq \Delta_i O_{cin} \qquad c \in \mathcal{C}, i \in \mathcal{I}_c, n \in \mathcal{N}_i \qquad (3.1g)$$

$$S_{cin} \leq \sum_{c'\in\mathcal{C}} \sum_{m\in\mathcal{N}_i, m<n} d_{c'} O_m^{c'i} \qquad c \in \mathcal{C}, i \in \mathcal{I}_c, n \in \mathcal{N}_i \qquad (3.1h)$$

$$S_{cin} \geq \sum_{c'\in\mathcal{C}} \sum_{m\in\mathcal{N}_i, m<n} d_{c'} O_m^{c'i} - \Delta_i(1 - O_{cin}) \qquad c \in \mathcal{C}, i \in \mathcal{I}_c, n \in \mathcal{N}_i \qquad (3.1i)$$

$$\sum_{n\in\mathcal{N}_i} S_{cin} \leq s^M w + \Delta_i(1 - Z_{ciw}) - \epsilon \qquad c \in \mathcal{C}, i \in \mathcal{I}_c, w \in \mathcal{W}_i \qquad (3.1j)$$

$$\sum_{n\in\mathcal{N}_i} S_{cin} \geq s^M (w-1) Z_{ciw} \qquad c \in \mathcal{C}, i \in \mathcal{I}_c, w \in \mathcal{W}_i \qquad (3.1k)$$

$$\sum_{n\in\mathcal{N}_i} O_{cin} \geq Z_{ciw} \qquad c \in \mathcal{C}, i \in \mathcal{I}_c, w \in \mathcal{W}_i \qquad (3.1l)$$

$$O_w^{c_1 i} + O_{w+1}^{c_2 i} \leq 1 \qquad \begin{array}{r} c_1, c_2 \in \mathcal{C}, g_{c_1} = g_{c_2}, \\ i \in \mathcal{I}_{c_1} \cap \mathcal{I}_{c_2}, n \in \mathcal{N}_i \end{array} \qquad (3.1m)$$

$$O_w^{c_2 i} + O_{w+1}^{c_1 i} \leq 1 \qquad \begin{array}{r} c_1, c_2 \in \mathcal{C}, g_{c_1} = g_{c_2}, \\ i \in \mathcal{I}_{c_1} \cap \mathcal{I}_{c_2}, n \in \mathcal{N}_i \end{array} \qquad (3.1n)$$

$$\sum_{n\in\mathcal{N}_i, n>0} O_{cin} = 0 \qquad c \in \mathcal{C}^{\mathtt{F}}, i \in \mathcal{I}_c, \alpha_{ci} = \mathtt{F} \qquad (3.1o)$$

$$O_{cin} \leq 1 - \sum_{\bar{c}\in\mathcal{C}_i} O_{(n+1)}^{\bar{c}i} \qquad \begin{array}{r} c \in \mathcal{C}^{\mathtt{L}}, i \in \mathcal{I}_c, \\ \alpha_{ci} = \mathtt{L}, n \in \mathcal{N}_i \end{array} \qquad (3.1p)$$

$$\sum_{i\in\mathcal{I}_c, \alpha_{ci}=\mathtt{H}} \sum_{n\in\mathcal{N}_i} S_{cin} \leq s^H \qquad c \in \mathcal{C}^{\mathtt{H}} \qquad (3.1q)$$

$$\sum_{i \in \mathcal{I}_c, \alpha_{ci}=\mathrm{M}} \sum_{n \in \mathcal{N}_i} S_{cin} \leq s^M \qquad c \in \mathcal{C}^{\mathrm{M}} \tag{3.1r}$$

$$O_{cin} \in \{0, 1\} \qquad c \in \mathcal{C}, i \in \mathcal{I}_c, n \in \mathcal{N}_i \tag{3.1s}$$

The objective function (3.1a) seeks to maximize total revenue generated. Constraint (3.1b) requires each commercial to be aired at most once, while constraint (3.1c) requires that no more than 1 commercial is assigned to any slot in a commercial break. Constraint (3.1d) stipulates that slot $n + 1$ cannot be occupied unless its predecessor, slot n, is occupied, thereby ensuring that commercials are aired sequentially starting at the designated time.

Constraint (3.1e) guarantees that the total duration of commercials within a break does not exceed the capacity of the commercial break, while constraint (3.1f) enforces government regulations on the maximum allowable duration of advertisements per hour.

Constraints (3.1g, 3.1h, 3.1i) establish the relationship between O_{cin} and S_{cin}. Similarly, constraints (3.1j, 3.1k) link S_{cin} to Z_{ciw}. Additionally, constraint (3.1l) mandates that Z_{ciw} be set to 0 if commercial $c \in \mathcal{C}$ is not scheduled in any slot of the commercial break $i \in \mathcal{I}$.

Constraints (3.1m, 3.1n) prevent commercials from the same group from being aired consecutively. Finally, constraints (3.1o, 3.1p, 3.1q, 3.1r) ensure that all specified flags are adhered to.

3.2 Discrete-Time Formulation

The problem can be discretized to streamline the time-tracking component. Given that both commercial and inventory durations are assumed to be integers in seconds, discretizing the problem with a unit increment of 1 preserves all relevant information in real-life data. This approach ensures no loss of detail while simplifying the modeling process.

Table 3.3: *Sets and parameters.*

$\mathcal{T}_i$	Set of discrete time points (e.g., seconds) in commercial break $i \in \mathcal{I}$
$\mathcal{T}_{ci}$	Set of feasible discrete time points that commercial $c \in \mathcal{C}$ can air in $i \in \mathcal{I}_c$
$\mathcal{R}_{ct}$	$\left\{ \bar{t} \middle\vert \begin{array}{ll} t - d_c < \bar{t} < t, & \text{if } t - d_c \geq 0 \\ 0 \leq \bar{t} < t, & \text{otherwise} \end{array} \right\}$
p_{cit}	Revenue of commercial $c \in \mathcal{C}$, if aired in commercial break $i \in \mathcal{I}_c$ at time $t \in \mathcal{T}_{ci}$
X_{cit}	Binary variable indicating whether commercial $c \in \mathcal{C}$ airs in break $i \in \mathcal{I}_c$ at time $t \in \mathcal{T}_{ci}$

Building on Table 3.1, we introduce the set of discrete time points, $\mathcal{T}$, and the set $\mathcal{R}_{ct}$, which includes all time points at which commercial c, if started, would end after t. We reduce the 3 decision variable setting in 3.1 into single binary variable X_{cit} that represents airing of commercial $c \in \mathcal{C}$ in commercial break $i \in \mathcal{I}_c$ at time $t \in \mathcal{T}_{ci}$. The feasible airing times, $t \in \mathcal{T}_{ci}$, are defined based on the flags of the commercial as follows:

$$\mathcal{T}_{ci} = \begin{cases} \{0\} & \text{, if } \alpha_{ci} = \texttt{F} \\ \{t \,|\, 0 \leq t \leq \min\{s^H, \Delta_i - d_c\}\} & \text{, if } \alpha_{ci} = \texttt{H} \\ \{t \,|\, 0 \leq t \leq \min\{s^M, \Delta_i - d_c\}\} & \text{, if } \alpha_{ci} = \texttt{M} \\ \{t \,|\, 0 \leq t \leq \Delta_i - d_c\} & \text{, otherwise} \end{cases} \tag{3.2}$$

$$\max \sum_{c \in \mathcal{C}} \sum_{i \in \mathcal{I}_c} \sum_{t \in \mathcal{T}_{ci}} p_{cit} X_{cit} \tag{3.3a}$$

$$\text{s.t.} \sum_{i \in \mathcal{I}_c} \sum_{t \in \mathcal{T}_{ci}} X_{cit} \leq 1, \qquad c \in \mathcal{C} \tag{3.3b}$$

$$\sum_{i \in \mathcal{I}_h} \sum_{c \in \mathcal{C}_i} \sum_{t \in \mathcal{T}_{ci}} d_c X_{cit} \leq s^O, \qquad \forall h \in \mathcal{H} \tag{3.3c}$$

$$\sum_{c \in \mathcal{C}_i} \sum_{\bar{t} \in \mathcal{R}_{ct}} X_{ci\bar{t}} \leq 1, \qquad \forall i \in \mathcal{I}, t \in \mathcal{T}_i \tag{3.3d}$$

$$\sum_{\substack{c \in \mathcal{C}_i, \\ d_c \leq t}} X_{ci(t-d_c)} \geq \sum_{c \in \mathcal{C}_i} X_{cit}, \qquad \forall i \in \mathcal{I}, t \in \mathcal{T}_{ci} \setminus 0 \tag{3.3e}$$

$$\sum_{\substack{c' \in \mathcal{C}_i, \\ g_c = g_{c'} \\ d_{c'} \leq t}} X_{c'i(t-d_{c'})} \leq 1 - X_{cit}, \qquad \forall c \in \mathcal{C}, i \in \mathcal{I}_c, t \in \mathcal{T}_{ci} \tag{3.3f}$$

$$X_{cit} \leq 1 - \sum_{c' \in \mathcal{C}_i, c' \neq c} X_{c'i(t+d_c)}, \qquad \forall \mathcal{C}^{\mathtt{L}}, i \in \mathcal{I}_c, \alpha_{ci} = \mathtt{L}, t \in \mathcal{T}_i \tag{3.3g}$$

$$X_{cit} \in \{0, 1\}, \qquad \forall c \in \mathcal{C}, i \in \mathcal{I}_c, t \in \mathcal{T}_{ci} \tag{3.3h}$$

The objective function 3.3a is defined to maximize the generated revenue. Constraint 3.3b ensures that no commercial is aired more than once and constraint 3.3c enforces hourly commercial broadcasting limit.

Constraint 3.3d ensures that for any time point in any commercial break no commercial is overlapping. Constraint 3.3e is defined to ensure no idle time in commercial breaks meaning that after each commercial next one required to start immediately. Constraints 3.3f and 3.3g set group and L flag constraint respectively. Note that all other flag requirements are satisfied within $\mathcal{T}_{ci}$.

The formulations presented above address the unique challenges associated with commercial scheduling in distinct ways. While the continuous-time formulation 3.1 encounters difficulties with time-dependent rewards, the discrete-time formulation 3.3 faces challenges in enforcing constraints related to the sequencing of commercials, such as the group and last flag constraints. It is worth emphasizing that the discrete-time formulation (3.3) demonstrates superior performance compared to the continuous-time formulation (3.1). Moreover, the discrete-time approach offers significantly tighter bounds. Nevertheless, both formulations face challenges in producing high-quality solutions when applied to larger problem instances.

3.3 Existing Approaches

The commercial scheduling problem can be viewed as a special case of the non-renewable resource-constrained homogeneous parallel machine scheduling problem, where both time- and machine-dependent rewards and constraints are incorporated, with the objective of maximizing the total reward. However, the inclusion of optional jobs and reward-based objectives is highly unconventional in traditional scheduling theory. These concepts, while rare in scheduling problems, are commonly addressed in routing problems. For example, the well-studied Team Orienteering Problem (TOP) can be reformulated as a scheduling problem by assuming zero travel times, thereby simplifying the structure to align with scheduling tasks.

Moreover, concepts such as due dates, release times, and resource constraints can be incorporated into routing problems through the introduction of time windows. In a similar vein, the commercial scheduling problem can be reformulated as a Team Orienteering Problem with Time Windows and Time-Dependent Scores (TOPTW-TDS). In this reformulation, flags such as the `H` can be modeled using time windows (e.g., `H` with a time window of [0, 30], or `F` with a window of [0, 0]). The primary exception is the `L`, which can be represented by assigning infinite distances between the associated node and others, or by restricting connections solely to the depot. However, since the eligibility and flags associated with commercials are dependent on commercial breaks, each vehicle will function within its own distinct network.

The Commercial Scheduling problem is thus formulated as a multi-commodity network $G = \{G^i = (V^i, E^i) \mid i \in \mathcal{I}\}$, where each network G^i represents a unique commercial break (vehicle) i within $\mathcal{I}$. For each network, the node set is defined as $V^i = \{n \mid n \in (\{0\} \cup \mathcal{C}_i)\}$, where 0 serves as the depot and $\mathcal{C}_i$ represents the commercials eligible for i. The directed edge set is denoted as $E^i = \{(a, b) \mid a \neq b,\ g_a \neq g_b,\ \alpha_{ai} \neq \mathtt{L},\ a, b \in V^i\}$. To address flag `L`, connections from such nodes are restricted to the depot. Additionally, group constraints are implemented by prohibiting connections between nodes within the same group. Time windows are assigned to nodes based on each commercial's specific flag requirements: `F` is assigned $[0, 0]$, `H` is assigned $[0, 30]$, and `M` is assigned $[0, 60]$. Finally, the regulatory limit on hourly broadcasting is enforced as a shared service duration constraint across groups of vehicles.

However, existing algorithms perform inadequately on this incomplete multi-commodity network. Specifically, we apply the HABC algorithm proposed by [56] for the Team Orienteering Problem with TOPTW-TDS. In doing so, we extend the fitness evaluation process, as discussed in [56], to exclude nodes that are either infeasible or inaccessible from the preceding node. However, due to the incomplete nature of the graph, the HABC algorithm demonstrates restricted effectiveness in handling this network structure. To address these limitations, we introduce a GRASP algorithm with Path Relinking, specially designed for the Commercial Scheduling problem, to achieve efficient and robust solutions.

4. VND-GRASP WITH PATH RELINKING

Our framework builds on the traditional GRASP methodology by introducing VND and Mixed-Path Relinking to enhance the optimization process. VND broadens the search scope by exploring multiple neighborhoods, allowing the algorithm to effectively escape local optima and uncover higher-quality solutions. Subsequently, the Mixed-Path Relinking algorithm is employed to further refine the solutions identified by VND.

Algorithm 1 VND-GRASP with Path Relinking

while stopping criterion not satisfied **do**
    $S \leftarrow$ SemiGreedyConstruction
    $S \leftarrow$ VND(S, Ω, β)
    $\mathcal{E} \leftarrow \emptyset$
    if $|\mathcal{E}| > 2$ **then**
        $S_g \leftarrow$ Random $S \in \mathcal{E}$
        $S \leftarrow$ MixedPathRelinking(S, S_g, β)
        $S \leftarrow$ VND(S, Ω, β)
    end if
    Update elite set $\mathcal{E}$
end while
return $\arg\max_{S \in \mathcal{E}}\{f(S)\}$

In Algorithm 1, we present the procedure for VND-GRASP with Path Relinking. The variable S represents a candidate solution. The neighborhood function Ω defines the local search space, and β serves as a parameter within the range [0, 1], controlling the randomness of the improvement phase. $\mathcal{E}$ denotes the elite set, consisting of up to ten solutions with the highest objective values encountered during the search. Additionally, the distance between any pair of solutions in $\mathcal{E}$ is strictly greater than zero. The elite set is dynamically updated to ensure it always contains the top-performing and sufficiently diverse solutions. The function $f(\cdot)$ is used to evaluate the objective function value of a given solution.

4.1 Construction Phase

During the construction phase, a greedy randomized solution is generated. The effectiveness of the GRASP algorithm depends significantly on the quality of the solutions produced in this phase. Therefore, it is essential to develop an efficient constructive heuristic. In our approach, a commercial is inserted into an inventory at each iteration of the heuristic, with the insertion randomly selected from a Restricted Candidate List (RCL). The RCL is defined as a set of tuples (c, i, s), where each tuple represents the insertion of commercial c into inventory i with an associated score s. The RCL is formally defined by the equation:

$$RCL = \{(c, i, s) \mid s \geq s_{max} - \alpha(s_{max} - s_{min}), \forall (c, i, s) \in \Theta(S, C, I)\} \tag{4.1}$$

where $\Theta(S, C, I)$ encompasses all feasible insertion triplets, consisting of the set of commercials $\mathcal{C}$ and the set of inventories $\mathcal{I}$ into the partial solution S. Here, s_{max} and s_{min} represent the maximum and minimum scores, respectively, across all potential insertion triplets in $\Theta(S, C, I)$.

The quality (score) of an insertion is quantified using a greedy score function:

$$\text{GreedyScore}(c, i, t) = \pi_{\alpha_{ci}} \frac{p_{cit}}{d_c} \tag{4.2}$$

Here, c represents a commercial, i denotes an inventory, and t specifies the time window during which commercial c begins broadcasting in the inventory. The score is multiplied by a random coefficient π_a, defined as follows to introduce further randomness to the algorithm:

$$\pi_a = \begin{cases} \text{Uniform}(0.5, 2) & \text{, if } a \in \{\mathtt{F}, \mathtt{L}, \mathtt{M}, \mathtt{H}\} \\ 1 & \text{, otherwise} \end{cases} \tag{4.3}$$

The pseudocode for the Semi-Greedy Construction Heuristic is detailed in Algorithm 2. During each iteration, a candidate is randomly selected from the RCL and added

Algorithm 2 Semi-Greedy Construction Heuristic
INPUT: $\mathcal{C}, \mathcal{I}, \alpha$
OUTPUT: S

$S \leftarrow \emptyset$
$\tau \leftarrow$ Map of (inventory, utilization) pairs. Utilizations are initially 0.
while $C \neq \emptyset$ **do**
  RCL $\leftarrow \{(c, i, s) \mid s \geq s_{max} - \alpha(s_{max} - s_{min}), \quad \forall (c, i, s) \in \Theta(S, \mathcal{C}, \mathcal{I})\}$
  if RCL = $\emptyset$ **then**
    break
  end if
  $c, i, s \leftarrow$ Randomly selected triplet from RCL
  $\tau[i] \leftarrow \tau[i] + d_c$
  $S \leftarrow S \cup (i, c, \tau[i])$
  $\mathcal{C} \leftarrow \mathcal{C} \setminus c$
end while
return S

to the current partial solution S. Insertions that would render the solution infeasible are excluded from the RCL. Thus, the process continues until all commercials are assigned to inventories or no further feasible insertions remain.

4.2 Improvement Phase

Within the GRASP framework, the solution initially generated by the semi-greedy algorithm is enhanced through subsequent local search and intensification strategies. We employ VND alongside path-relinking techniques to refine and improve the solution's quality. This phase is designed to explore multiple neighborhoods and strategic paths to further optimize the preliminary solution.

4.2.1 Variable Neighborhood Descent

VND is a local search methodology that leverages multiple neighborhood structures to deepen the exploration of the solution space. The VND algorithm systematically navigates through these neighborhoods, cycling back to the initial neighborhood upon finding an improvement. The sequence in which neighborhoods are arranged is pivotal for an effective search. Each neighborhood, denoted as $\Omega_k^\beta(\cdot)$, is designed to yield either the best or the first improving feasible move for a given solution S. With a probability of β, it instead returns a random feasible move. In cases where no feasible solution is identified, $\Omega_k^\beta(\cdot)$ is assumed to return a neutral move that preserves the existing solution without alteration.

For clarity, we define *position* as the sequential order in which a commercial is broadcast within an inventory. For instance, position 1 signifies that a commercial is the first to be aired. Furthermore, the symbol $\oplus$ is used to represent the move operation. Here are the detailed definitions of specific move operations:

- *shift*(i, n_1, n_2): Moves a commercial from position n_1 to position n_2 within the same inventory i.
- *intra-swap*(n_1, n_2, i): Exchanges the positions of commercials at n_1 and n_2 within inventory i, assuming $n_1 < n_2$.
- *transfer*(n_1, n_2, i_1, i_2): Moves a commercial from position n_1 in inventory i_1 to position n_2 in inventory i_2, with $i_1 \neq i_2$.
- *inter-swap*(n_1, n_2, i_1, i_2): Swaps commercials between positions n_1 in inventory i_1 and n_2 in inventory i_2, assuming the duration of the commercial at n_1 is less than that at n_2.
- *insert*(c, i, k): Adds a previously unscheduled commercial c into inventory i to position k.
- *out-of-pool*(c, n, i): Replaces the commercial at position n in inventory i with an unscheduled commercial c.

Measuring infeasibility in the context of commercial scheduling is complex due to the numerous constraints involved. As a result, infeasible moves are strictly prohibited.

Algorithm 3 Variable Neighborhood Descent
INPUT: S, Ω, β
OUTPUT: S^*

$S^* \leftarrow S$
$k \leftarrow 1$
while $k < |\Omega|$ **do**
  $move \leftarrow \Omega_k^\beta(S)$
  $\bar{S} \leftarrow S \oplus move(\cdot)$
  if $f(\bar{S}) > f(S^*)$ **then**
    $k \leftarrow 1$
    $S^* \leftarrow \bar{S}$
  end if
  $k \leftarrow k + 1$
end while

However, verifying the feasibility of each potential solution can be computationally expensive. To mitigate this, we have developed specific feasibility criteria tailored to each type of move, enabling a more efficient evaluation process without compromising the integrity of the solution.

To concisely represent feasibility conditions, we introduce the functions $\mathcal{F}(\cdot)$ and $\mathcal{G}(\cdot)$. We make the assumption that any undefined c_{in} (e.g., where $n = 0$) is treated as if it was not included when passed to $\mathcal{G}(\cdot)$. Furthermore, if the same commercial is passed consecutively to $\mathcal{G}(\cdot)$, it is processed as though it were submitted only once. This is particularly relevant in cases such as *intra-swap*(n_1, n_2, i), where $n_2 - n_1 = 1$.

Table 4.2 details the conditions required to assess the feasibility of a move. It is crucial to note that for some moves, multiple conditions may apply. Each relevant condition must be verified accordingly. Definitions of the notations utilized are available in Table 4.1.

4.2.2 Mixed Path Relinking

Path Relinking is an intensification strategy widely utilized across numerous metaheuristics, where it has consistently led to marked improvements in solution quality and

Table 4.1: *Nomenclature.*

Notation	Description
c_{in}	Commercial in inventory $i \in \mathcal{I}$ at position n.
t_{in}	Start time of a commercial in inventory $i \in \mathcal{I}$ at position n.
e_{in}	End time of a commercial in inventory $i \in \mathcal{I}$ at position n.
$\mathcal{F}(c, i, n, t, N)$	Checks if commercial $c \in \mathcal{C}$ satisfies the flag when assigned to position n with start time t in inventory $i \in \mathcal{I}_c$, given N commercials assigned to i after the assignment. If commercial is not flagged, it returns true.
$\mathcal{G}([\cdot])$	Given a list of commercials $[\cdot]$, checks if $[\cdot]$ contains any consecutive commercials belonging to the same group $g \in \mathcal{G}$.
δ_i	Total broadcasting time of inventory $i \in \mathcal{I}$.
N_i	Position of the last commercial in inventory $i \in \mathcal{I}$.
h_i	Hour in which inventory $i \in \mathcal{I}$ is scheduled.
δ_h	Total broadcasting time in hour h.
α_{ci}	Flag of commercial $c \in \mathcal{C}$ for inventory $i \in \mathcal{I}_c$.

reductions in computation time [57]. Typically, this technique involves exploring the paths between two distinct solutions: an initial solution S_i and a guiding solution S_g. The algorithm methodically traverses the paths connecting these solutions, effectively evolving S_i towards S_g. We define $\Omega^{\beta}(S \to S_g) \subseteq \Omega^{\beta}(S)$ as a restricted neighborhood of S such that the moves in $\Omega^{\beta}(S \to S_g)$ reduce the distance between S and S_g. The restricted neighborhood function, $\Omega^{\beta}(S \to S_g)$, is designed to return the best move; however, with a probability β, it may instead return a random move.

In the Path Relinking algorithm, similar to the VND approach, infeasible moves are strictly prohibited. As a result, a path may reach a dead end before it successfully reaches the guiding solution. To broaden the search scope and explore solutions that are closer to the guiding solution, we implement the Mixed Path Relinking algorithm. This method constructs two paths simultaneously: one originates from the initial solution and the other from the guiding solution, both converging towards each other. The proposed algorithm is outlined in 4.

Table 4.2: *Feasibility conditions.*

Move	*Condition*	Check
shift(i, n_1, n_2)	*None*	$\mathcal{G}([c_{i(n_1-1)}, c_{i(n_1+1)}])$
	$n_1 < n_2$	$\mathcal{F}(c_{in_1}, i, n_2, e_{in_2} - d_{in_1}, N_i)$, $\mathcal{G}([c_{in_2}, c_{in_1}, c_{i(n_2+1)}])$
	$n_1 \geq n_2$	$\mathcal{F}(c_{il}, i, l+1, t_c + d_{c_i n_1}, N_i)\, \forall n = n_2, \ldots, n_1 - 1$, $\mathcal{F}(c_{in_1}, i, n_2, t_{in_2}, N_i)$, $\mathcal{G}([c_{in_2-1}, c_{in_1}, c_{in_1}])$
	$n_2 = N_i$	$\alpha^i_{c_{iN_i}} \neq \mathtt{L}$
intra-swap(n_1, n_2, i)	*None*	$\mathcal{F}(c_{n_1}, i, n_2, t_{n_2} + d_{n_2} - d_{n_1}, N_i)$, $\mathcal{F}(c_{n_2}, i, n_1, t_{n_1}, N_i)$, $\mathcal{G}([c_{i(n_1)-1}, c_{i(n_2)}, c_{i(n_1)+1}])$, $\mathcal{G}([c_{i(n_2)-1}, c_{i(n_1)}, c_{i(n_2)+1}])$
	$d_{n_1} < d_{d_{n_2}}$	$\mathcal{F}(c_n, i, n, t_n + d_{n_2} - d_{n_2}, N_i)\, \forall\, n = n_1 + 1, \ldots, n_2 - 1$
transfer (n_1, n_2, i_1, i_2)	*None*	$\mathcal{F}(c_{i_2 n}, i_2, n+1, t_{i_2 n} + d_{c_{i_1 n_1}}, N_{i_2} + 1)\, \forall n = n_2, \ldots, N_{i_2}$, $\mathcal{G}([c_{i_1(n_1-1)}, c_{i_1(n_1+1)}])$, $\mathcal{G}([c_{i_2(n_2-1)}, c_{i_1 n_1}, c_{i_2 n_2}])$, $\delta_{i_2} + d_{c_{i_1 n_1} \leq D_i}$
	$1 < n_2 = N_{i_2} + 1$	$\alpha^{i_2}_{c_{i_2 N_{i_2}}} \neq \mathtt{L}$, $\mathcal{F}(c_{i_1 n_1}, i_2, n_2, \delta_{i_2}, N_{i_2} + 1)$
	$n_2 = 1$	$\mathcal{F}(c_{i_1 n_1}, i_2, n_2, 0, N_{i_2} + 1)$
	$1 < n_2 < N_{i_2} + 1$	$\mathcal{F}(c_{i_1 n_1}, i_2, n_2, t_{i_2 n_2}, N_{i_2} + 1)$
	$h_{i_1} \neq h_{i_2}$	$\delta_h + d_{i_1 n_1} \leq s^O$
inter-swap (n_1, n_2, i_1, i_2)	*None*	$\mathcal{F}(c_{i_1 n}, i_1, n, t_{i_1 n} + (d_{c_{i_2 n_2}} - d_{c_{i_1 n_1}}), N_{i_1})\, \forall n = n_1 + 1, \ldots, N_{i_1}$, $\mathcal{F}(c_{i_1 n_1}, i_2, n_2, t_{i_2 n_n} 2, N_{i_2})$, $\mathcal{F}(c_{i_2 n_2}, i_1, n_1, t_{i_1 n_n} 1, N_{i_1})$, $\mathcal{G}([c_{i_1(n_1-1), c_{i_2 n_2}, c_{i_1(n_1+1)}}])$, $\mathcal{G}([c_{i_2(n_2-1), c_{i_1 n_1}, c_{i_2(n_2+1)}}])$, $\delta_{i_1} + (d_{c_{i_2 n_2}} - d_{c_{i_1 n_1}}) \leq D_{i_1}$
	$h_{i_1} \neq h_{i_2}$	$\delta_{h_{i_1}} + (d_{c_{i_2 n_2}} - d_{c_{i_1 n_1}}) \leq s^O$
insert (c, i, n)	*None*	$\mathcal{F}(c_{il}, i, l+1, t_{in} + d_c, N_i + 1)\, \forall l = n, \ldots, N_i$, $\mathcal{G}([c_{i(n-1)}, c, c_{in}])$, $\delta_i + d_c \leq D_i$, $\delta_{h_i} + d_c \leq s^O$
	$n \neq N_i + 1$	$\mathcal{F}(c, i, n, t_{in}, N_i + 1)$
	$n = N_i + 1$	$(\mathcal{F}(c, i, n, \delta_i, N_i + 1), \alpha^i_{c_{iN_i}} \neq \mathtt{L})$ or $N_i = 0$
out-of-pool (c, n, i)	*None*	$\mathcal{F}(c, i, n, t_{in}, N_i)$, $\mathcal{G}([c_{i(n-1)}, c, c_{i(n+1)}])$
	$d_c > d_{c_{in}}$	$\mathcal{F}(c_{il}, i, l, e_{il} + (d_c - d_{c_{in}}), N_i)\, \forall l = n+1, \ldots, N_i$, $\delta_i + (d_c - d_{c_{in}}) \leq D_i$, $\delta_{h_i} + (d_c - d_{c_{in}}) \leq s^O$

Algorithm 4 Mixed Path Relinking
INPUT: S_i, S_g, β
OUTPUT: S^*

```
S* ← S_i
e ← true
while Δ(S̄_i, S̄_g) > 1 do
    ω ← Ω^β(S_i → S_g)
    if ω is None then
        if e is false then
            return S*
        end if
        e ← false
        continue
    end if
    S_i ← S_i ⊕ ω
    S* ← arg max{f(S_i), f(S*)}
    if e is true then
        S_i, S_g ← S_g, S_i
    end if
end while
return S*
```

4.2.2.1 Distance Function

We define the distance between two solutions, $\Delta(S, S_g)$, as the number of differing commercial-inventory assignments between S and S_g. An efficient algorithm for calculating $\Delta(S, S_g)$ is outlined in Algorithm 5. It is important to note that sorted assignments need only be calculated during the initial construction phase. During the local search phase, this list can be efficiently updated as needed, without requiring full re-sorting.

Algorithm 5 iteratively scans the lists N_S and N_{S_g}, representing the current assignment combinations. When these lists are equal, i.e., all tuples are identical, the assignments in both solutions match. However, if there are differences, quantified as $\Delta(S, S_g) > 0$, three types of discrepancies can arise:

1. A commercial is present in both solutions but in different inventories.

2. A commercial is present in S but absent in S_g.

Algorithm 5 Distance Function
INPUT: S, S_g
OUTPUT: n

$n \leftarrow 0$
$N_S \leftarrow$ Array of (commercial, inventory) assignment pairs (c, i) of S sorted in ascending order by commercial ID.
$N_{S_g} \leftarrow$ Array of (commercial, inventory) assignment pairs (c, i) of S_g sorted in ascending order by commercial ID.
$k \leftarrow 1$
$l \leftarrow 1$
while $k \leq |N_S|$ and $l \leq |N_{S_g}|$ **do**
  $(c_S, i_S) \leftarrow N_S[k]$
  $(c_{S_g}, i_{S_g}) \leftarrow N_{S_g[l]}$
  if $c_S = c_{S_g}$ **then**
    if $i_S \neq i_{S_G}$ **then**
      $n \leftarrow n + 1$
    end if
    $k \leftarrow k + 1$
    $l \leftarrow l + 1$
  else if $c_S < c_{S_g}$ **then**
    $n \leftarrow n + 1$
    $k \leftarrow k + 1$
  else
    $n \leftarrow n + 1$
    $l \leftarrow l + 1$
  end if
end while
return $n + max\{|N_S| - k, |N_{S_g}| - l\}$ + 1

3. A commercial is absent in S but present in S_g.

These discrepancies are efficiently identified by iterating over the sorted assignment tuples. Consider two tuples $(c_S, i_S) \in N_S$ and $(c_{S_g}, i_{S_g}) \in N_{S_g}$. The following checks are applied:

- If $c_S = c_{S_g}$ but $i_S \neq i_{S_g}$, it indicates that commercial c_S is assigned to different inventories in the two solutions.
- If $c_S < c_{S_g}$, it signifies that commercial c_S is not scheduled in S_g, as no further

tuple in N_{S_g} will contain c_S due to the sorted order.

- Conversely, if $c_{S_g} < c_S$, it indicates that commercial c_{S_g} is not scheduled in S, although it is present in S_g.

4.2.2.2 Restricted Neighborhood

Exploring every possible move can be computationally intensive. To address this, we define a restricted neighborhood $\Omega^{\beta}(S \to S_g)$ as outlined in Algorithm 6. It is critical to note that when $\Omega^{\beta}(S \to S_g)$ is used iteratively, the algorithm is assured to converge to a state where $\Delta(S, S_g) = 0$ within $\Delta(S, S_g)$ moves, assuming no dead ends are encountered. This proof of convergence is straightforward. $\Omega^{\beta}(S \to S_g)$ includes the operations: $insert(\cdot)$, $transfer(\cdot)$, and $remove(\cdot)$. Each operation is specifically tailored to reduce the distance by one. Consequently, in $\Delta(S, S_g)$ iterations, S will align precisely with the assignment configuration of S_g.

Since VND does not utilize the $remove(\cdot)$ operation, we have not defined it until now. $remove(n_1, i)$ is defined as an operation that removes the commercial $c \in \mathcal{C}$ at position n_1 in inventory $i \in \mathcal{I}_c$. The only feasibility requirement for $remove(n_1, i)$ is $\mathcal{G}([c_{i(n_1-1)}, c_{i(n_1+1)}])$.

With the definition of the neighborhood function and the key operations for our solution methodology complete, we now turn our attention to evaluating the performance of the proposed approach. While the algorithm is designed to optimize the commercial scheduling problem effectively, its practical impact and efficiency can only be understood through rigorous computational testing. In the following section, we present the results of our experiments, which assess the performance of both the MIP model and the GRASP algorithm across a range of real-world problem instances.

Algorithm 6 Restricted Neighborhood Function $\Omega^{\beta}(S \to S_g)$
INPUT: S, S_g, β
OUTPUT: ω

$N_S \leftarrow$ List of commercial-inventory assignment pairs (c, i) of S sorted by commercial ID.
$N_{S_g} \leftarrow$ List of commercial-inventory assignment pairs (c, i) of S_g sorted by commercial ID.
$\Omega \leftarrow \emptyset$
$k \leftarrow 1$
$l \leftarrow 1$
while $k \leq |N_S|$ and $l \leq |N_{S_g}|$ **do**
    $(c_S, i_S) \leftarrow N_S[k]$
    $(c_{S_g}, i_{S_g}) \leftarrow N_{S_g[l]}$
    if $c_S = c_{S_g}$ **then**
        if $i_S \neq i_{S_G}$ **then**
            $\hat{\Omega} \leftarrow$ For solution S, all feasible *transfer* moves that transfer c_S to i_{S_g}
        end if
        $k \leftarrow k + 1$
        $l \leftarrow l + 1$
    else if $c_S < c_{S_g}$ **then**
        $\hat{\Omega} \leftarrow$ For solution S *remove* move that removes c_S from S, if feasible, otherwise Ø
        $k \leftarrow k + 1$
    else
        $\hat{\Omega} \leftarrow$ For solution S, all feasible *insert* moves that insert c_{S_g} to i_{S_g}
        $l \leftarrow l + 1$
    end if
    $\Omega \leftarrow \Omega \cup \hat{\Omega}$
end while
if $l \leq |N_{S_g}|$ **then**
    $\hat{\Omega} \leftarrow$ All possible insert moves for tuple $(c, i) \in N_{S_g}$ that are after index $l - 1$
else if $k \leq |N_S|$ **then**
    $\hat{\Omega} \leftarrow$ All remove moves for tuple $(c, i) \in N_S$ that are after index $k - 1$
end if
$\Omega \leftarrow \Omega \cup \hat{\Omega}$
if probability β **then**
    return Random move in Ω
end if
return $\arg\max_{\omega \in \Omega}\{f(\omega \oplus S)\}$

5. COMPUTATIONAL RESULTS

This section evaluates the performance of the MIP model, the HABC algorithm [56], and the proposed GRASP algorithm across ten real-world problem instances. For heuristic algorithms, each configuration are run three times per instance, and the reported results represent the average outcomes. To maintain confidentiality, the actual values for commercial prices and ratings have been masked by multiplying them by a random constant $X \sim \text{Uniform}(0.2, 0.8)$. Consequently, the objective values reported in this study do not reflect real-world figures but are instead based on these masked and perturbed revenue values. For the purposes of this study, readers may consider the prices of commercials as being expressed in abstract units. In addition, generic commercial slots in the dataset are excluded from the analysis, as they fall outside the scope of this study, along with the corresponding generic commercials. Furthermore, any commercials or inventories missing specific rating information are omitted from the dataset. A summary of the studied problem instances is provided in Table 5.1. These datasets are publicly available [1]. In this study, we compare performances of different approaches through GAP. The GAP is calculated as $\frac{|best\ bound - obj|}{|obj|}$, where the best bound refers to the best objective bound found by the solver using the discrete-time formulation (3.3) within a two-hour computational time limit.

5.1 Comparison of MIP Model Performances

Figure 5.1 compares the performance of two MIP models evaluated at various time points. The Y-axis indicates the MIP GAP for each evaluation. Bars with reduced opacity represent the continuous-time formulation, whereas opaque bars depict the discrete-time

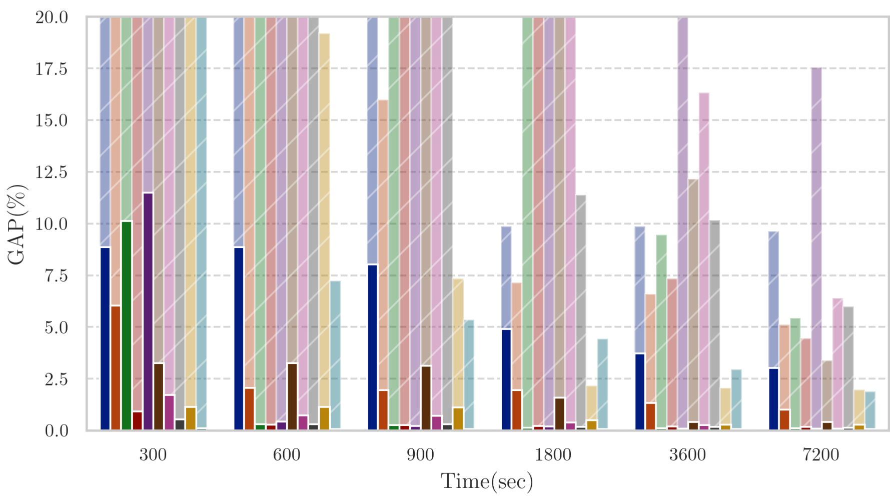
[1] https://github.com/OEKundakcioglu/ThesisTVCommercialScheduling

Table 5.1: *Detailed instance information.*

Instance	# Commercials	# Inventories	# Hours	# Target Audiences
1	163	18	5	17
2	162	18	5	16
3	195	18	5	21
4	202	18	5	21
5	190	18	5	14
6	162	18	5	17
7	192	18	5	13
8	161	18	5	13
9	103	14	3	9
10	105	11	3	11

formulation. The results clearly demonstrate that the discrete-time formulation consistently outperforms the continuous-time formulation across all instances and evaluated time points.

Figure 5.1: *Comparative performance of MIP models within 0-20% gap against the best upper bound obtained. Lighter (darker) shades represent continuous (discrete) time formulations.*



Although not explicitly depicted in Figure 5.1, the continuous-time formulation struggles to produce tight bounds, with solver-reported GAP values exceeding 100% even after two hours of computation. It is important to note that the solver calculates the gap relative to the upper bound determined by the respective model, whereas Figure 5.1 illustrates the gap relative to the upper bound obtained from the discrete-time formulation after two hours of runtime. In contrast, the discrete-time formulation demonstrates significantly better performance, achieving much tighter bounds with greater computational efficiency.

For simpler instances, the discrete-time formulation attains a MIP GAP of approximately 1% within 5–10 minutes. Nonetheless, it does not establish optimality even after two hours of computation. Although this formulation is highly effective in generating high-quality solutions for less complex instances, its performance declines as the problem becomes more complex. These findings underscore the necessity of heuristic methods to effectively tackle larger and more intricate problem instances, where exact methods encounter substantial computational challenges.

5.2 Performance of HABC

The HABC algorithm is adapted to address the specific challenges of the Commercial Scheduling Problem. To account for the graph's incomplete nature and incorporate service time constraints, we extend the fitness evaluation process described by [56], excluding infeasible nodes and those inaccessible from the preceding node. Additionally, the termination conditions in [56] are replaced with a fixed time limit to provide the algorithm with additional runtime. To enhance convergence, the algorithm is initialized with the constructive heuristic developed in this study.

Table 5.2: *Best solutions identified by the HABC across all parameter configurations in 5 minutes.*

Instances	1	2	3	4	5	6	7	8	9	10	Avg
GAP(%)	55.59	121.38	21.46	23.06	14.39	28.02	50.23	16.37	72.63	708.02	111.11

The parameter configurations evaluated include all combinations of initial temperature values $T_0 \in 0.99, 0.995, 0.9999$, cooling factors $\alpha \in 0.9, 0.95, 0.99$, and population sizes $B \in 1000, 3000, 6000$. Table 5.2 summarizes the best solutions obtained for each instance, selecting the best solution among all runs and configurations within the fixed 5-minute time limit. Despite testing all configurations, the algorithm demonstrates limited ability to improve upon the initial solution generated by the constructive heuristic. This underperformance likely stems from the restrictive and complex constraints characteristic to the commercial scheduling problem, which significantly limit the applicability of conventional local search moves. As a result, the algorithm struggles to explore the solution space effectively within the given runtime.

5.3 Performance of GRASP

In contrast to the HABC algorithm, which struggles to improve upon initial solutions, the GRASP algorithm exhibits superior performance across all tested instances. The algorithm is evaluated under multiple parameter configurations, denoted as $\theta|\alpha|\beta$, where $\theta \in B, F$ specifies the improvement strategy: B represents best improvement, and F represents first improvement in the local search. The parameter α defines the method of assigning its value to the constructive heuristic at each iteration, either as a fixed value, represented as $F(\cdot)$, or drawn from a uniform distribution, denoted as $U(\cdot)$. Finally, the parameter β controls the degree of randomization introduced into the algorithm.

The final GAPs obtained by all configurations of the GRASP algorithm are presented in Table 5.3. The results demonstrate that GRASP consistently achieves near-optimal solutions, with an average GAP of approximately 1%. Across all parameter configurations, the algorithm maintains robust performance, consistently identifying high-quality solutions irrespective of the specific parameter settings. However, given sufficient computational time, discrete-time formulation can outperform the GRASP algorithm in

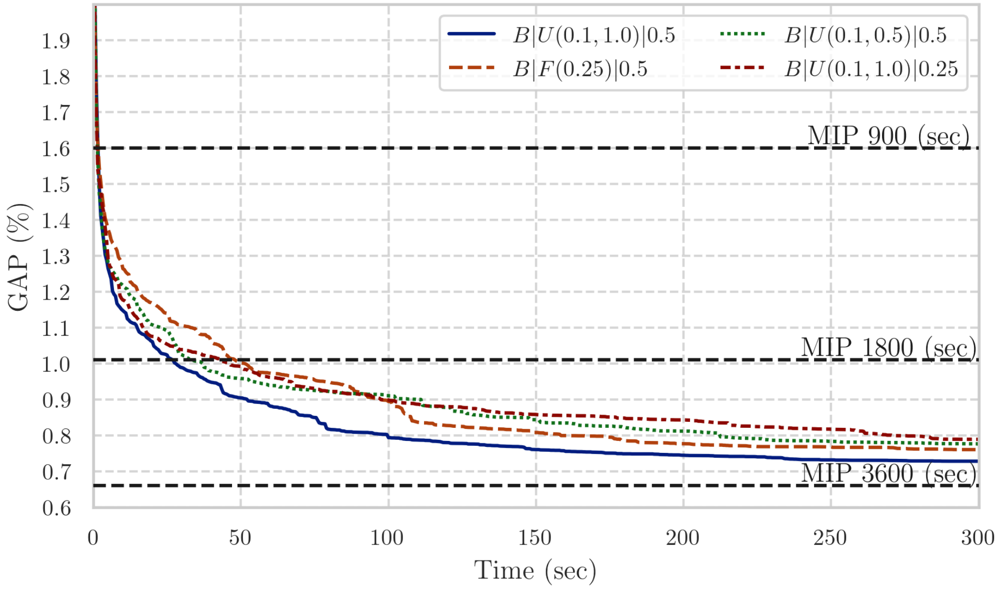
Table 5.3: *Average GAPs achieved by different configurations within 5 minutes. The top 5 performing configurations are highlighted in bold.*

Configuration			Instances										
θ	α	β	1	2	3	4	5	6	7	8	9	10	Avg
B	$F(0.25)$	0.0	1.77	**0.25**	1.48	0.62	0.98	0.65	1.13	0.81	2.75	0.63	1.11
		0.25	1.78	**0.23**	1.10	0.46	0.65	0.53	0.92	**0.53**	1.95	0.57	0.87
		0.5	1.89	0.27	1.05	**0.41**	**0.49**	**0.41**	**0.82**	**0.61**	**1.34**	0.63	**0.79**
	$F(0.5)$	0.0	**1.70**	0.31	1.48	0.75	0.96	0.69	1.08	0.80	2.59	0.66	1.10
		0.25	1.76	0.30	1.36	0.55	0.78	0.60	0.87	**0.60**	1.65	**0.55**	0.90
		0.5	1.73	0.27	1.03	**0.44**	0.89	**0.34**	**0.84**	0.71	1.50	**0.54**	**0.83**
	$U(0.1, 0.5)$	0.0	**1.71**	0.27	1.40	0.65	0.71	0.72	1.09	0.89	2.75	0.60	1.08
		0.25	1.81	**0.23**	1.12	**0.41**	0.66	0.47	0.92	**0.63**	2.40	**0.55**	0.92
		0.5	1.73	**0.26**	**0.95**	**0.42**	**0.62**	**0.42**	**0.71**	**0.58**	1.82	**0.54**	**0.81**
	$U(0.1, 1.0)$	0.0	1.79	0.29	1.44	0.73	0.86	0.67	1.04	0.85	**1.49**	0.64	0.98
		0.25	1.75	0.27	1.17	0.47	**0.58**	0.49	0.84	0.65	**1.44**	**0.54**	**0.82**
		0.5	**1.55**	**0.25**	**0.97**	0.45	0.62	0.48	**0.73**	0.72	**1.26**	0.55	**0.76**
F	$F(0.25)$	0.0	1.75	0.34	1.40	0.79	0.92	0.71	1.12	0.87	2.72	0.69	1.13
		0.25	1.72	0.33	**0.96**	0.45	**0.60**	**0.41**	0.85	0.87	2.50	0.62	0.93
		0.5	1.90	0.39	1.19	**0.44**	**0.53**	0.47	0.99	0.84	**1.41**	0.68	0.88
	$F(0.5)$	0.0	**1.63**	0.40	1.26	0.78	0.97	0.74	1.03	0.98	2.32	0.70	1.08
		0.25	1.75	0.36	**0.98**	0.51	0.88	**0.38**	0.95	0.94	1.72	0.62	0.91
		0.5	2.59	0.44	1.04	0.49	0.96	0.61	**0.77**	0.88	1.61	0.70	1.01
	$U(0.1, 0.5)$	0.0	**1.68**	0.35	1.54	0.70	0.84	0.64	0.99	0.82	2.54	0.67	1.08
		0.25	1.71	0.36	1.01	0.47	0.70	0.56	0.86	0.90	2.24	0.59	0.94
		0.5	2.04	0.41	**0.95**	0.50	0.70	0.48	0.88	0.88	2.18	0.64	0.97
	$U(0.1, 1.0)$	0.0	1.72	0.36	1.40	0.60	0.96	0.72	1.16	0.97	1.69	0.65	1.02
		0.25	1.94	0.29	1.23	0.49	0.72	0.43	0.88	0.81	1.67	0.57	0.90
		0.5	2.05	0.37	1.48	0.46	0.68	0.46	0.94	0.91	1.90	0.66	0.99

terms of solution quality on smaller instances.

The convergence behavior of GRASP is illustrated in Figure 5.2, which depicts the averaged performance of the four best-performing configurations over time. The results show that GRASP reliably converges to a GAP of approximately 1% within a relatively short computational timeframe. The algorithm is capable of producing higher-quality solutions than those obtained by the discrete-time formulation (3.3) within a half-hour

Figure 5.2: *Averaged performance in terms of gap against the best bound obtained by discrete-time formulation across all instances for the top 4 GRASP configurations over time.*



computational limit, and the solutions found by GRASP are only worse by less than 0.1% compared to those obtained by the discrete-time formulation (3.3) within a one-hour computational timeframe.

6. CONCLUSION

In this study, we address a practical challenge in the TV advertising industry known as the commercial scheduling problem. This problem involves determining the optimal assignment and scheduling of commercials within designated breaks while complying with stringent stakeholder requirements and regulatory mandates. The revenue generated from a broadcast is highly dependent on both the specific break and the advertisement's start time. The commercial scheduling problem can be viewed as a specialized case of a non-renewable resource-constrained identical parallel machine scheduling problem, characterized by machine- and time-dependent rewards, with the primary objective of maximizing these rewards.

To tackle this problem, we propose two MIP models, develop a GRASP heuristic (VND-GRASP with Path Relinking), and reformulate the problem as a TOPTW-TDS and apply HABC algorithm proposed in [56]. We evaluate these approaches using ten real-world problem instances. The results show that the discrete-time formulation consistently outperforms the continuous-time formulation in terms of solution quality and the ability to provide tight upper bounds. Traditional methodologies struggle with the intricate constraints of commercial scheduling, while the proposed heuristic reliably achieves a MIP gap of approximately 1% for all instances within about 100 seconds. Although the discrete-time formulation remains more effective on smaller instances, the GRASP approach effectively handles larger and more complex scenarios. We also introduce feasibility conditions that enhance the local search phase of the heuristic, further improving performance.

Despite the strong results, the discrete-time formulation is not able to prove optimality within the given time limit. Therefore, future research will focus on refining exact solution approaches, improving the problem formulation, and advancing exact solution methodologies. Finally, all source code and sample datasets are available publicly available [1], providing valuable resources for ongoing research and methodological improvements.

[1] https://github.com/OEKundakcioglu/ThesisTVCommercialScheduling

REFERENCES

[1] EACA, "Social benefits." https://eaca.eu/news/advertising-delivers-powerful-economic-benefits-across-eu/, 2024. Last visited 2024-09-20.

[2] S. Bollapragada, M. R. Bussieck, and S. Mallik, "Scheduling commercial videotapes in broadcast television," *Operations Research*, vol. 52, no. 5, pp. 679–689, 2004.

[3] R. Bai and J. Xie, "Heuristic algorithms for simultaneously accepting and scheduling advertisements on broadcast television," *Journal of Information and Computer Science*, vol. 1, no. 4, pp. 245–251, 2006.

[4] R. Alaei and F. Tari, "Development of a genetic algorithm for advertising time allocation problems," *Journal of Industrial and Systems Engineering*, 01 2011.

[5] M. Majidi, "TV advertising worldwide - statistics & facts," 2023. Last visited 2024-09-20.

[6] RATEM, "Radio and television broadcasting sectoral report for Turkey (in Turkish) 2019," 2019. Last visited 2024-09-20.

[7] J. G. Navarro, "Ad spending worldwide 2022-2023," May 2024. Last visited 2024-09-20.

[8] M. Givon and A. Grosfeld-Nir, "Using partially observed markov processes to select optimal termination time of TV shows," *Omega*, vol. 36, no. 3, pp. 477–485, 2008. Special Issue on Multiple Criteria Decision Making for Engineering.

[9] P. J. Danaher, T. S. Dagger, and M. S. Smith, "Forecasting television ratings," *International Journal of Forecasting*, vol. 27, no. 4, pp. 1215–1240, 2011.

[10] L. Song, Y. Shi, G. K. F. Tso, and H. P. Lo, "Forecasting week-to-week television ratings using reduced-form and structural dynamic models," *International Journal of Forecasting*, vol. 37, no. 1, pp. 302–321, 2021.

[11] Wikipedia, "Television in turkey," 2024. Last visited 2024-09-20.

[12] TIAK, "TV audience research company," 2024. Last visited 2024-09-20.

[13] A. S. C. Ehrenberg, G. J. GoodHardt, and M. Collins, "The way that people watch television," *New Society*, pp. 262–263, 1975.

[14] R. S. Headen, J. E. Klompmaker, and R. T. Rust, "The duplication of viewing law and television media schedule evaluation," *Journal of Marketing Research*, vol. 16, no. 3, pp. 333–340, 1979.

[15] M. D. Henry and H. J. Rinne, "Predicting program shares in new time slots.," *Journal of Advertising Research*, 1984.

[16] R. T. Rust and N. V. Eechambadi, "Scheduling network television programs: A heuristic audience flow approach to maximizing audience share," *Journal of Advertising*, vol. 18, no. 2, pp. 11–18, 1989.

[17] S. K. Reddy, J. E. Aronson, and A. Stam, "Spot: Scheduling programs optimally for television," *Management Science*, vol. 44, no. 1, pp. 83–102, 1998.

[18] A. R. Brown, "Selling television time: An optimisation problem," *The Computer Journal*, vol. 12, pp. 201–207, 08 1969.

[19] R. Balachandra, "A simulation model for predicting the effect of advertisement schedules," in *Proceedings of the 9th Conference on Winter Simulation - Volume 2*, WSC '77, p. 580–589, Winter Simulation Conference, 1977.

[20] H. Simon, "Adpuls: An advertising model with wearout and pulsation," *Journal of Marketing Research*, vol. 19, no. 3, pp. 352–363, 1982.

[21] V. Mahajan and E. Muller, "Advertising pulsing policies for generating awareness for new products," *Marketing Science*, vol. 5, no. 2, pp. 89–106, 1986.

[22] G. L. Lilien, P. Kotler, and K. S. Moorthy, *Marketing models*. Prentice Hall, 1992.

[23] S. Bollapragada, H. Cheng, M. Phillips, M. Garbiras, M. Scholes, T. Gibbs, and M. Humphreville, "Nbc's optimization systems increase revenues and productivity," *Interfaces*, vol. 32, no. 1, pp. 47–60, 2002.

[24] S. Bollapragada and M. Garbiras, "Scheduling commercials on broadcast television," *Operations Research*, vol. 52, no. 3, pp. 337–345, 2004.

[25] A. Mihiotis and I. Tsakiris, "A mathematical programming study of advertising allocation problem," *Applied Mathematics and Computation*, vol. 148, no. 2, pp. 373–379, 2004.

[26] X. Zhang, "Mathematical models for the television advertising allocation problem," *International Journal of Operational Research*, vol. 1, no. 3, pp. 302–322, 2006.

[27] T. Benoist, E. Bourreau, and B. Rottembourg, "The TV-break packing problem," *European Journal of Operational Research*, vol. 176, no. 3, pp. 1371–1386, 2007.

[28] A. Kimms and M. Muller-Bungart, "Revenue management for broadcasting commercials: the channel's problem of selecting and scheduling the advertisements to be aired," *International Journal of Revenue Management*, vol. 1, no. 1, pp. 28–44, 2007.

[29] M. S. Wuang, C. L. Yang, R. H. Huang, and S. P. Chuang, "Scheduling of television commercials," in *2010 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 803–807, 2010.

[30] J. Mao, J. Shi, J. Wanitwattanakosol, and S. Watanabe, "An ACO-based algorithm for optimising the revenue of TV advertisement using credit information," *International Journal of Revenue Management*, vol. 5, no. 2-3, pp. 109–120, 2011.

[31] Y. H. Lee and M. Pinedo, "Scheduling jobs on parallel machines with sequence-dependent setup times," *European Journal of Operational Research*, vol. 100, no. 3, pp. 464–474, 1997.

[32] R. Logendran, B. McDonell, and B. Smucker, "Scheduling unrelated parallel machines with sequence-dependent setups," *Computers & Operations Research*, vol. 34, no. 11, pp. 3420–3438, 2007.

[33] Z.-L. Chen, "Parallel machine scheduling with time dependent processing times," *Discrete Applied Mathematics*, vol. 70, no. 1, pp. 81–93, 1996.

[34] Y. Ouazene and F. Yalaoui, "Identical parallel machine scheduling with time-dependent processing times," *Theoretical Computer Science*, vol. 721, pp. 70–77, 2018.

[35] T. A. Feo and M. G. C. Resende, "A probabilistic heuristic for a computationally difficult set covering problem," *Operations Research Letters*, vol. 8, no. 2, pp. 67–71, 1989.

[36] T. A. Feo, K. Venkatraman, and J. F. Bard, "A GRASP™ for a difficult single machine scheduling problem," *Computers & Operations Research*, vol. 18, no. 8, pp. 635–643, 1991.

[37] T. A. Feo, K. Sarathy, and J. McGahan, "A GRASP for single machine scheduling with sequence dependent setup costs and linear delay penalties," *Computers & Operations Research*, vol. 23, no. 9, pp. 881–895, 1996.

[38] S. Binato, W. J. Hery, D. M. Loewenstern, and M. G. C. Resende, *A GRASP for Job Shop Scheduling*, pp. 59–79. Boston, MA: Springer US, 2002.

[39] R. M. Aiex, S. Binato, and M. G. C. Resende, "Parallel GRASP with path-relinking for job shop scheduling," *Parallel Computing*, vol. 29, no. 4, pp. 393–430, 2003.

[40] C. Bierwirth and J. Kuhpfahl, "Extended GRASP for the job shop scheduling problem with total weighted tardiness objective," *European Journal of Operational Research*, vol. 261, no. 3, pp. 835–848, 2017.

[41] M. Rajkumar, P. Asokan, T. Page, and S. Arunachalam, "A GRASP algorithm for the integration of process planning and scheduling in a flexible job-shop," *International Journal of Manufacturing Research*, vol. 5, no. 2, pp. 230–251, 2010.

[42] P. A. M. Rajkumar and V. Vamsikrishna, "A GRASP algorithm for flexible job-shop scheduling with maintenance constraints," *International Journal of Production Research*, vol. 48, no. 22, pp. 6821–6836, 2010.

[43] N. A. M. Rajkumar, P. Asokan and T. Page, "A GRASP algorithm for flexible job-shop scheduling problem with limited resource constraints," *International Journal of Production Research*, vol. 49, no. 8, pp. 2409–2423, 2011.

[44] A. Baykasoğlu and F. S. Karaslan, "Solving comprehensive dynamic job shop scheduling problem by using a GRASP-based approach," *International Journal of Production Research*, vol. 55, no. 11, pp. 3308–3325, 2017.

[45] G. Prabhaharan, B. S. H. Khan, and L. Rakesh, "Implementation of GRASP in flow shop scheduling," *The International Journal of Advanced Manufacturing Technology*, vol. 30, pp. 1126–1131, Oct 2006.

[46] S. Rojanasoonthon and J. Bard, "A GRASP for parallel machine scheduling with time windows," *INFORMS Journal on Computing*, vol. 17, no. 1, pp. 32–51, 2005.

[47] V. A. Armentano and M. F. de França Filho, "Minimizing total tardiness in parallel machine scheduling with setup times: An adaptive memory-based GRASP approach," *European Journal of Operational Research*, vol. 183, no. 1, pp. 100–114, 2007.

[48] J. C. Yepes-Borrero, F. Villa, F. Perea, and J. P. Caballero-Villalobos, "GRASP algorithm for the unrelated parallel machine scheduling problem with setup times and additional resources," *Expert Systems with Applications*, vol. 141, p. 112959, 2020.

[49] R. Alvarez-Valdes, E. Crespo, J. M. Tamarit, and F. Villa, "GRASP and path relinking for project scheduling under partially renewable resources," *European Journal of Operational Research*, vol. 189, no. 3, pp. 1153–1170, 2008.

[50] S. K. Monkman, D. J. Morrice, and J. F. Bard, "A production scheduling heuristic for an electronics manufacturer with sequence-dependent setup costs," *European Journal of Operational Research*, vol. 187, no. 3, pp. 1100–1114, 2008.

[51] R. D. Leone, P. Festa, and E. Marchitto, "A bus driver scheduling problem: a new mathematical model and a GRASP approximate solution," *Journal of Heuristics*, vol. 17, pp. 441–466, Aug 2011.

[52] H. Hashimoto, S. Boussier, M. Vasquez, and C. Wilbaut, "A GRASP-based approach for technicians and interventions scheduling for telecommunications," *Annals of Operations Research*, vol. 183, pp. 143–161, Mar 2011.

[53] C. Park and J. Seo, "A GRASP approach to transporter scheduling and routing at a shipyard," *Computers & Industrial Engineering*, vol. 63, no. 2, pp. 390–399, 2012.

[54] J. E. Gómez-Lagos, M. C. González-Araya, W. E. Soto-Silva, and M. M. Rivera-Moraga, "Optimizing tactical harvest planning for multiple fruit orchards using a metaheuristic modeling approach," *European Journal of Operational Research*, vol. 290, no. 1, pp. 297–312, 2021.

[55] Ofcom, "Rules on the amount and distribution of advertising," 2024. Last visited 2024-09-20.

[56] V. F. Yu, P. Jewpanya, S.-W. Lin, and A. P. Redi, "Team orienteering problem with time windows and time-dependent scores," *Computers & Industrial Engineering*, vol. 127, pp. 213–224, 2019.

[57] M. G. C. Resende and C. C. Ribeiro, *Optimization by GRASP*. Springer, 2016.