

Akıllı Televizyon Programı Planlayıcı

Beyzanur Zeybek

TOBB ETÜ

Yapay Zeka Mühendisliği

Ankara, Türkiye

bzeybek@etu.edu.tr

Özet—Bu çalışma, televizyon izleyicilerinin tercihlerine göre kişiselleştirilmiş ve optimize edilmiş bir izleme planı oluşturmayı amaçlayan bir akıllı televizyon programı planlayıcısının geliştirilmesini kapsamaktadır. Proje kapsamında, web scraping yöntemleri ile televizyon kanallarının haftalık yayın akışları toplanmış ve JSON formatında yapılandırılmış veriler elde edilmiştir. Bu veriler, kullanıcıların seçtiği gün, saat aralığı ve program kategorilerine göre işlenerek en uygun izleme kombinasyonu oluşturulmuştur. BFS, DFS, A*, Greedy ve Genetik Algoritma gibi beş farklı yapay zeka tabanlı algoritma kullanılarak, her birinin öneri kalitesi ve zaman verimliliği karşılaştırılmıştır. Geliştirilen kullanıcı dostu arayüz sayesinde, kullanıcılar farklı algoritmaların sunduğu izleme planlarını karşılaştırmalı olarak görebilmekte ve tercihlerini en iyi şekilde yapabilmektedir. Proje, zaman yönetimi ve içerik seçimi açısından kullanıcı deneyimini iyileştiren, dinamik ve kişiselleştirilmiş bir öneri sistemi sunarak hem günlük kullanım hem de akademik araştırmalar için değerli bir katkı sağlamaktadır.

Github linki: <https://github.com/BeyzanurZeybek/Akilli-TV-Programi-Olusturucu>

Youtube video linki: <https://youtu.be/NaZysDceg2k>

Anahtar Kelimeler—TV yayın planlayıcı, öneri sistemleri, BFS, DFS, A* algoritması, genetik algoritma, Greedy algoritma, web scraping, zaman optimizasyonu, yapay zeka, kullanıcı tercihi, içerik önerisi

I. GİRİŞ

Televizyon yayınları her gün çok çeşitli içeriklerle kullanıcıların karşısına çıkmaktadır. Ancak bu içeriklerin çeşitliliği ve yayın saatlerinin çakışması, kullanıcıların ilgi alanlarına göre en uygun programı seçmesini zorlaştırmaktadır. Özellikle aynı zaman aralığında birden fazla kanalın yayın yaptığı durumlarda, kullanıcıların manuel olarak yayın akışlarını kontrol edip bir tercih yapması zaman alıcı ve zahmetli olabilmektedir.

Bu çalışmada geliştirilen Akıllı Televizyon Programı Planlayıcı, kullanıcıların belirli gün, saat ve kategori tercihlerine göre en uygun izleme planını sunmayı amaçlayan bir öneri sistemidir. Sistem, haftalık yayın akışlarını web scraping teknikleri ile toplayarak, verileri belirli bir yapıya dönüştürmekte ve ardından farklı algoritmalarla optimize edilmiş öneriler üretmektedir.

Proje kapsamında Breadth-First Search (BFS), Depth-First Search (DFS), A* algoritması, Greedy algoritma ve Genetik Algoritma gibi çeşitli arama ve optimizasyon algoritmaları kullanılmıştır. Geliştirilen grafik arayüz sayesinde kullanıcılar farklı algoritmaların sunduğu önerileri karşılaştırarak en uygun izleme planını kolaylıkla seçebilmektedir.

Bu rapor kapsamında sistemin literatürdeki yeri, kullanılan algoritmaların teknik detayları, verilerin toplanma ve işlenme süreçleri, öneri sisteminin gerçekleştirilmesi ve elde edilen sonuçlar detaylı biçimde sunulmaktadır.

II. LİTERATÜR TARAMASI

Öneri sistemleri, dijital içerik yönetimi, elektronik ticaret ve medya dağıtımı gibi birçok alanda yaygın olarak kullanılan karar destek mekanizmaları haline gelmiştir. Kullanıcı tercihlerinin artan çeşitliliği ve içerik bolluğu, bu sistemlerin kişiselleştirme yeteneklerinin gelişimini zorunlu kılmaktadır.

Öneri sistemleri genel olarak üç temel yaklaşım ile sınıflandırılmaktadır: işbirlikçi filtreleme (Collaborative Filtering), içerik tabanlı filtreleme (Content-Based Filtering) ve bu iki yöntemin birleşiminden oluşan hibrit sistemler. Burke'in çalışması [2], hibrit sistemlerin kullanıcılar arası benzerlik ve içerik özelliklerini bir araya getirerek daha isabetli öneriler sunduğunu göstermektedir. Netflix'in öneri sistemi üzerine yapılan araştırmalar [5], kullanıcı etkileşimlerinin analizine dayalı bireysel önerilerin kullanıcı memnuniyeti üzerinde doğrudan etkili olduğunu ortaya koymaktadır.

Kim ve Lee [1], sosyal TV ortamlarında bağlamsal farkındalık ve dinamik kullanıcı profillerinin etkin öneri üretiminde önemli olduğunu belirtmektedir. Bu tür bağlam odaklı sistemler, yalnızca içerik benzerliğine değil, aynı zamanda zaman, cihaz türü ve sosyal etkileşim gibi parametrelere de odaklanarak öneri kalitesini artırmaktadır.

Zamana duyarlı öneri sistemleri üzerine yapılan çalışmalar, özellikle medya içeriklerinde zaman bilgisinin kritik bir faktör olduğunu vurgulamaktadır. Koren [6], kullanıcı tercihlerinin zamana göre değişebileceğini ve bu durumun öneri sistemlerinin başarımı üzerinde belirleyici olabileceğini göstermiştir.

Veri kalitesinin öneri sistemleri üzerindeki etkisi, web scraping temelli projelerde daha da önem kazanmaktadır. Ferrara ve arkadaşları [4] tarafından yapılan kapsamlı bir incelemede, otomatik veri çekme süreçlerinde elde edilen verilerin eksikliği, yapısal bozukluğu veya güncelliğinin, sistemin genel doğruluğu ve güvenilirliği üzerinde doğrudan etkili olduğu belirtilmiştir.

Öneri sistemlerinin bir diğer önemli boyutu da çok amaçlı optimizasyon yaklaşımlarının uygulanabilirliğidir. Deb ve arkadaşlarının geliştirdiği NSGA-II algoritması [3], aynı anda birden fazla hedefin optimize edilebilmesini mümkün kılarak öneri sistemlerinde esnek yapıların oluşturulmasına katkı sağlamaktadır.

Son olarak, öneri sistemlerinde kullanıcı arayüzünün tasarımı da literatürde sıkça ele alınan bir konudur. Pu ve arkadaşları [7], kullanıcı deneyimini merkeze alan arayüzlerin, önerilerin kullanıcı tarafından kabul edilme oranı üzerinde önemli etkisi olduğunu vurgulamaktadır.

III. VERİ TOPLAMA SÜRECİ

Bu çalışmada, televizyon kanallarının haftalık yayın akışlarını içeren veri seti, web scraping teknikleri kullanılarak toplanmıştır. Veriler, *Hürriyet TV Rehberi* üzerinden çekilmiş ve JSON formatında yapılandırılmıştır. Veri çekimi sürecinde Python programlama dili ile birlikte `Requests` ve `BeautifulSoup` kütüphaneleri kullanılmıştır. Bot algılamalarını önlemek amacıyla HTTP isteklerine kullanıcı aracı (user-agent) başlığı eklenmiş ve her gün için ayrı URL'lere erişilerek HTML sayfalar işlenmiştir.

Scraping işlemi sonucunda elde edilen veriler, günlük bazda program adını, yayın saatini ve (varsa) kategori bilgisini içerecek şekilde düzenlenmiştir. Ancak, yayınlanan web sayfasında kategori bilgisi büyük oranda eksik veya tutarsız olduğundan, programların türleri manuel olarak gözlemlenmiş ve aşağıda listelenen kategorilerden biri ile etiketlenmiştir:

Kullanılan Program Kategorileri: *haber, spor, dram, eğlence, belgesel, aşk, yaşam, polisiye, aile, komedi, aksiyon, gerilim, çizgifilm, magazin, romantikkomedi, gençlik, tarih, bilimkurgu, din, yarışma, tartışma, askeri, macera, çizgidişi, yemek.*

Bu manuel etiketleme işlemi, sistemin kategoriye dayalı filtreleme ve öneri yapabilmesi için kritik bir ön işleme adımı olmuştur.

Toplanan ham veriler üzerinde ayrıca çeşitli temizlik ve dönüşüm işlemleri gerçekleştirilmiştir. Yayın saatleri "HH:MM - HH:MM" formatından dakika bazlı zaman aralıklarına çevrilmiş, sabaha karşı (00:00–06:00) başlayan programlar önceki güne atanmış ve her gün için kanallar kendi ID'leriyle gruplanmıştır. Tüm bu işlemlerin ardından nihai JSON veri yapısı oluşturulmuş ve algoritmalar tarafından işlenmeye hazır hale getirilmiştir.

Veri işleme süreci üç aşamadan oluşmaktadır:

- 1) **Ham verinin çekilmesi ve parse edilmesi:** HTML sayfalar `BeautifulSoup` ile ayrıştırılmış ve her bir günün yayın akışı elde edilmiştir.
- 2) **Kanal yapısının oluşturulması:** Programların zaman bilgilerine göre yeni bir kanal düzeni oluşturulmuş ve ardışık programlar uygun şekilde gruplandırılmıştır.
- 3) **Verinin düzenlenmesi ve etiketlenmesi:** Eksik veya hatalı bilgiler temizlenmiş, kategori bilgileri manuel olarak güncellenmiş ve JSON formatında son hali oluşturulmuştur.

Sonuç olarak elde edilen veri seti, her gün için birden fazla kanal içeren ve her programın adı, süresi ve kategorisi gibi bilgileri barındıran, öneri algoritmalarının çalışmasına uygun yapılandırılmış bir formata sahiptir. Bu yapı sayesinde kullanıcı tercihleri (saat aralığı, kategori) doğrultusunda algoritmalar tarafından uygun izleme planları üretilebilmiştir.

IV. ÖNERİ SİSTEMİNİN TASARIMI VE KULLANILAN ALGORİTMALAR

Bu projede geliştirilen öneri sistemi, kullanıcının belirli bir gün, saat aralığı ve program kategorisi tercihinine göre en uygun izleme planını sunmayı hedeflemektedir. Sistemin temel işleyişi, TV yayın akışlarının belirli bir veri yapısına dönüştürülmesi, kullanıcıdan alınan parametrelere göre uygun programların filtrelenmesi ve çeşitli algoritmalar kullanılarak izleme sıralamasının optimize edilmesi adımlarına dayanmaktadır.

Veri yapısı, her gün için birden fazla kanalın içerdiği, her kanalın ise bir dizi programdan oluştuğu hiyerarşik bir JSON formatındadır. Programlar zaman bilgileriyle birlikte kategori etiketleri taşımaktadır. Kullanıcıdan alınan girişler (gün, saat aralığı ve ilgi kategorileri), bu veri seti üzerinde filtreleme ve optimizasyon işlemlerinin gerçekleştirilmesini sağlar.

A. Sistem Akışı

Öneri sistemi aşağıdaki adımlarla çalışmaktadır:

- 1) Kullanıcıdan izlemek istediği gün, saat aralığı ve program kategorileri alınır.
- 2) Belirtilen parametrelere uygun olan programlar veri seti içerisinden filtrelenir.
- 3) Filtrelenmiş programlar kanal bazlı ağaç yapıları olarak yapılandırılır.
- 4) Beş farklı algoritmadan biri kullanılarak en uygun izleme sıralaması oluşturulur.
- 5) Algoritma çıktısı kullanıcı arayüzünde görsel olarak sunulur.

B. Kullanılan Algoritmaların Uygulanışı

Projede öneri süreci için beş farklı algoritma kullanılmıştır. Her algoritma farklı bir optimizasyon yaklaşımı izleyerek, kullanıcıya en uygun izleme planını üretmeyi hedeflemektedir.

1) **BFS (Genişlik Öncelikli Arama) Algoritması:** BFS algoritması, tüm kanallardaki programları eşzamanlı olarak değerlendirerek geniş kapsamlı bir izleme planı oluşturmayı hedefler. Kuyruk (queue) yapısı kullanılarak programlar sıralı şekilde işlenir. Her kanalın programları zaman sırasına göre kuyruk içerisine eklenir ve kullanıcının belirlediği başlangıç saatinden itibaren uygun olan ilk program seçilerek izleme listesine eklenir. Zaman güncellendikçe sıradaki programlar değerlendirilir ve böylece farklı kanallar arasında en erken başlayan uygun programlar seçilir. Kodda `collections.deque` yapısı ve `visited` kümesiyle tekrar eden programlar filtrelenmiştir. Bu yaklaşım, daha dengeli ve çeşitlilik içeren bir izleme planı sunar.

2) **DFS (Derinlik Öncelikli Arama) Algoritması:** DFS algoritması, bir kanal üzerinde derinlemesine ilerlemeyi tercih eder. Sistem, kullanıcının başlangıç saatinden itibaren bir kanalda olabildiğince uzun süre kesintisiz izleme sağlayacak programları seçer. Kanal değişimi minimum düzeyde tutulur. Her programın başlama zamanı kontrol edilerek önceki programla çakışmaması sağlanır. Kodda sırasıyla kanalların programları dolaşılır ve `visited` seti kullanılarak tekrarlar

engellenir. Bu strateji, istikrarlı bir izleme deneyimi isteyen kullanıcılar için uygundur.

3) *A* (A-Star) Algoritması*: A* algoritması, izleme planını oluştururken hem geçerli sürece ($g(n)$) hem de gelecekteki olasılıklara ($h(n)$) göre karar verir. Projede, her programın süresi bir tür heuristik ($h(n)$) olarak kullanılır ve toplam izleme süresiyle ($g(n)$) toplanarak bir maliyet fonksiyonu ($f(n)$) hesaplanır. heapq yapısı ile açık liste (open list) öncelikli kuyrukta tutulur. Algoritma, en yüksek $f(n)$ değerine sahip olan rotaları önceliklendirerek izleme süresini maksimize eden en uygun planı oluşturur. Program sıralaması esnasında saat uyumu ve kategori filtrelerine de dikkat edilmektedir.

4) *Greedy (Açgözlü) Algoritma*: Greedy algoritması, her adımda o an için en avantajlı (en uzun süren) programı seçerek ilerler. Programlar zaman sırasına göre sıralanır ve mevcut zaman noktasından itibaren izlenebilecek en uzun program seçilir. Bu süreç döngüsel olarak devam eder. Kısa sürede karar veren bu yapı, kullanıcıya hızlı ama her zaman optimal olmayan bir izleme planı sunar. Kodda `available_programs` listesi içerisinde minimum başlangıç zamanına sahip ve en uzun süreli program belirlenerek plana dahil edilmektedir.

5) *Genetik Algoritma*: Genetik algoritma, rastgele oluşturulan program dizilimlerini evrimsel işlemler (seçim, çaprazlama, mutasyon) ile geliştirerek en uygun izleme sıralamasını üretir. Her birey, kullanıcı kriterlerine uygun bir program listesiyle temsil edilir. Fitness fonksiyonu toplam izleme süresi, kategori uyumu ve çakışma cezasına göre hesaplanır. Turnuva seçimi ile en iyi bireyler seçilir; ardından çaprazlama ve mutasyon işlemleriyle yeni jenerasyonlar oluşturulur. En iyi birey, iterasyon sonunda kullanıcıya öneri olarak sunulur. Projenin bu aşaması, karmaşık ancak güçlü bir optimizasyon sağlar ve sistemin kişiselleştirme yeteneğini artırır.

V. ARAYÜZ VE UYGULAMA GELİŞTİRME

Projenin kullanıcı ile etkileşimini sağlayan arayüz, öneri sisteminin kullanılabilirliğini artırmak amacıyla tasarlanmıştır. Kullanıcı, geliştirilmiş arayüz sayesinde gün, saat aralığı ve program kategorisi gibi tercihlerde bulunarak farklı algoritmalarla oluşturulmuş izleme planlarını grafiksel olarak görüntüleyebilmektedir.

Arayüz, Python dili kullanılarak geliştirilen masaüstü tabanlı bir grafiksel kullanıcı arayüzüdür (GUI). Tkinter kütüphanesi temel alınarak hazırlanmış olup, kullanıcı dostu ve sade bir yapı hedeflenmiştir. Kullanıcıdan alınan parametreler, öneri sistemine iletilmekte ve her bir algoritma ayrı ayrı çalıştırılarak sonuçlar Gantt grafiği biçiminde görselleştirilmektedir. Bu grafikler, programların zaman çizelgesi ve kanal bilgisi ile birlikte sunulmasını sağlar.

Her algoritmanın ürettiği izleme planı, Matplotlib kullanılarak oluşturulan çubuk grafikler ile gösterilir. Bu grafiklerde yatay eksen zaman çizelgesini, dikey eksen ise kanal adlarını temsil eder. Her program, yayın süresi kadar bir çubukla gösterilir ve üzerine program adı eklenir. Böylece kullanıcılar,

Fig. 1. Kullanıcıdan seçim alınan arayüz ekranı

hangi saat aralıklarında hangi programların izlenebileceğini görsel olarak kolayca anlayabilir. Arayüzün başlıca işlevleri şu şekilde özetlenebilir:

- Kullanıcıdan gün, başlangıç ve bitiş saati, program kategorisi bilgilerini almak,
- Seçilen parametrelere uygun programları filtrelemek,
- Beş farklı algoritmayı çalıştırarak her biri için ayrı izleme planı oluşturmak,
- Gantt grafikleri aracılığıyla önerilen izleme planlarını kullanıcıya sunmak.

VI. ÖRNEK SENARYO UYGULAMASI

Sistemin işleyişini görselleştirmek amacıyla aşağıdaki kullanıcı senaryosu örnek olarak ele alınmıştır:

- **Gün:** Salı
- **Saat Aralığı:** 08:00–18:00
- **Seçilen Kategoriler:** yaşam, aşk, eğlence

Bu kriterlere göre sistem, uygun programları filtreleyerek bir ağaç yapısı oluşturur. Bu yapı, her kanalın belirli zaman aralığında yayınladığı ve seçilen kategorilere uygun programlardan oluşur. Figür 2’de oluşturulan bu ağaç yapısı şematik olarak gösterilmiştir. Bu yapı, algoritmaların sıralama ve seçim işlemlerini uygulayabilmesi için temel veri modeli olarak kullanılmaktadır.

Aynı senaryo için beş farklı algoritma çalıştırılmış ve her biri kullanıcı tercihlerini dikkate alarak farklı izleme planları üretmiştir. Bu planlar, programların zaman ve kanal bilgilerine göre oluşturulmuş Gantt grafikleri yardımıyla görsel olarak karşılaştırılmış; böylece algoritmaların sunduğu öneriler hem içerik yapısı hem de zaman dağılımı açısından analiz edilebilir hale getirilmiştir.

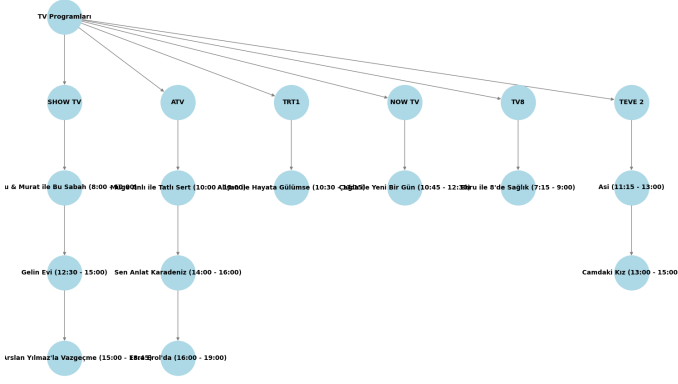


Fig. 2. Salı 08:00–18:00 arası aşk, yaşam ve eğlence kategorileri için oluşturulan ağaç yapısı

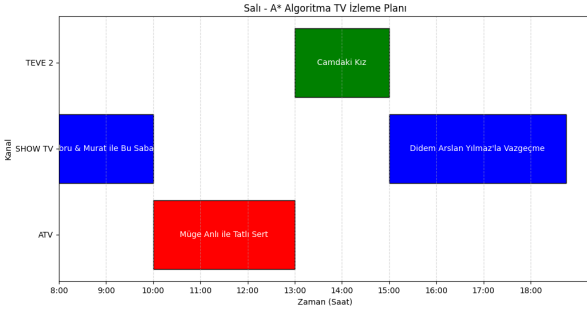


Fig. 3. A* algoritması ile oluşturulan izleme planı

A*: Maksimum izleme süresi için optimizasyon yapılmıştır. Programlar hem süre hem kategori uyumu açısından değerlendirilmiştir.

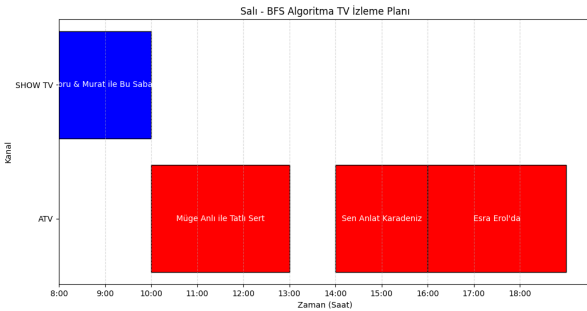


Fig. 4. BFS algoritması ile oluşturulan izleme planı

BFS: Tüm kanalları aynı anda değerlendiren bir stratejiyle daha çeşitli bir plan sunmuştur.

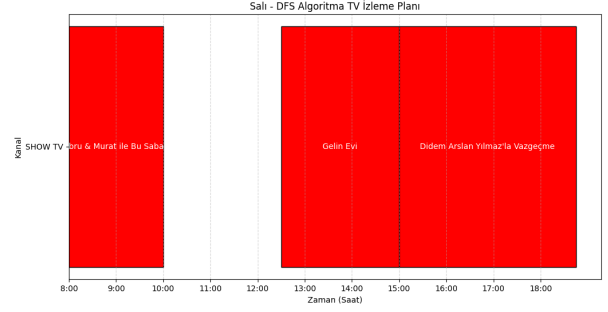


Fig. 5. DFS algoritması ile oluşturulan izleme planı

DFS: Tek bir kanal üzerinde derinlemesine ilerleyerek kesintisiz bir izleme planı üretmiştir.

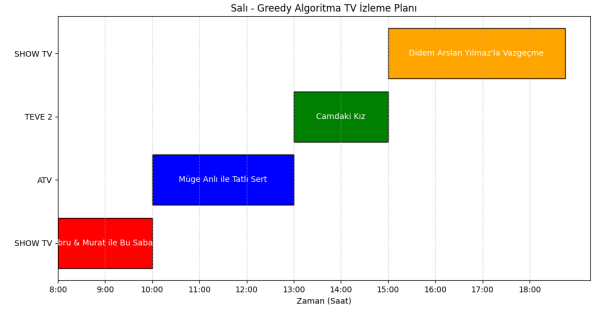


Fig. 6. Greedy algoritması ile oluşturulan izleme planı

Greedy: En erken başlayan ve en uzun süren programları seçerek hızlı bir öneri sunmuştur.

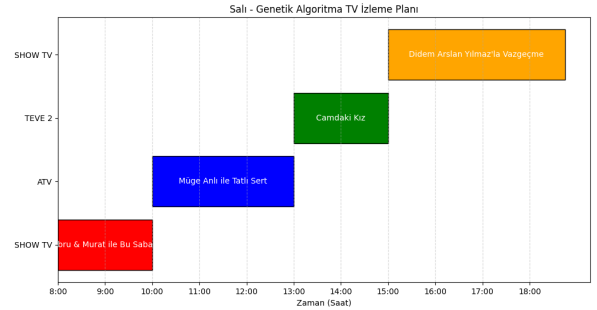


Fig. 7. Genetik algoritma ile oluşturulan izleme planı

Genetik Algoritma: Kategori uyumu, süre ve çakışma gibi farklı faktörleri değerlendirerek evrimsel süreçle optimize edilmiş bir plan sunmuştur.

Bu görseller sayesinde, kullanıcıların her algoritmanın sunduğu farklı yaklaşımları daha net şekilde karşılaştırabilmesi mümkün olmaktadır. Ayrıca sistemin esnekliği ve kişiselleştirme gücü görsel olarak da desteklenmiştir.

VII. SONUÇLAR VE DEĞERLENDİRME

Sistem tarafından beş farklı algoritma ile oluşturulan izleme planları, belirlenen örnek senaryo üzerinden karşılaştırmalı olarak değerlendirilmiştir. Her algoritma farklı bir optimizasyon stratejisi kullandığından, ürettikleri izleme planları hem içerik sıralaması hem de toplam izleme süresi bakımından değişiklik göstermektedir.

Yapılan testlerde aşağıdaki bulgulara ulaşılmıştır:

- **A* algoritması**, hem süreyi maksimize etme hem de kategori uyumunu dikkate alma açısından en dengeli sonuçları sunmuştur. Ancak çalışması diğer algoritmalarla göre daha uzun sürebilmektedir.
- **BFS algoritması**, tüm kanallar üzerinde geniş kapsamlı bir arama yaptığı için oldukça çeşitli programlar sunmuştur. Ancak bazı durumlarda izleme süreleri kısa kalabilmektedir.
- **DFS algoritması**, bir kanalda ardışık programları tercih ettiği için daha tutarlı ve kanal değiştirmeyen planlar üretmiştir. Bu yapı, özellikle kanal sabitliği isteyen kullanıcılar için avantajlıdır.
- **Greedy algoritma**, en hızlı çalışan yöntemlerden biri olup her adımda en uzun programı seçmiştir. Ancak bu yaklaşım genellikle küresel optimal çözümler yerine yerel maksimumlara ulaşmıştır.
- **Genetik algoritma**, kategori uyumu, süre ve çakışma gibi çok sayıda faktörü değerlendirerek kişiselleştirilmiş çözümler sunmuştur. Evrimsel sürecin doğası gereği daha kaliteli sonuçlar alınsa da, hesaplama süresi en uzun olan yöntemdir.

Algoritmaların ürettiği izleme planları, Matplotlib ile oluşturulan Gantt grafikleri yardımıyla görsel olarak sunulmuş ve kullanıcıya her seçeneğin anlaşılır biçimde sunulması sağlanmıştır. Özellikle görsel sunumlar, kullanıcıların karar verme sürecini kolaylaştırmış ve sistemin açıklanabilirliğini artırmıştır.

Sonuçlar göstermiştir ki, her algoritmanın belirli kullanıcı senaryoları için uygunluk düzeyi farklıdır. Örneğin; zaman kısıtlı kullanıcılar için Greedy algoritması uygunken, daha kişiselleştirilmiş içerik arayan kullanıcılar için Genetik veya A* algoritmaları daha etkilidir. Bu nedenle sistem, farklı kullanıcı profillerine hitap eden esnek bir yapı sunarak öneri sistemlerinin kişiselleştirme kabiliyetini ön plana çıkarmaktadır.

REFERENCES

- [1] Fotis Aisopos, Angelos Valsamis, Alexandros Psychas, Andreas Menychtas, and Theodora Varvarigou. Efficient context management and personalized user recommendations in a smart social tv environment. In *Economics of Grids, Clouds, Systems, and Services: 13th International Conference, GECON 2016, Athens, Greece, September 20-22, 2016, Revised Selected Papers 13*, pages 102–114. Springer, 2017.
- [2] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12:331–370, 2002.
- [3] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
- [4] Emilio Ferrara, Pasquale De Meo, Giacomo Fiumara, and Robert Baumgartner. Web data extraction, applications and techniques: A survey. *Knowledge-based systems*, 70:301–323, 2014.
- [5] Carlos A Gomez-Urbe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):1–19, 2015.
- [6] Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 447–456, 2009.
- [7] Pearl Pu, Li Chen, and Rong Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 157–164, 2011.