

Atf İçin: Demir Y, 2021. Tekrarlı Açgözlü Algoritma Üzerine Kapsamlı Bir Analiz. Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 11(4): 2716-2728.

To Cite: Demir Y, 2021. A Comprehensive Analysis on the Iterated Greedy Algorithm. Journal of the Institute of Science and Technology 11(4): 2716-2728.

Tekrarlı Açgözlü Algoritma Üzerine Kapsamlı Bir Analiz

Yunus DEMİR

ÖZET: Genel olarak optimizasyon, belirli kısıtlar altında en iyi sonucun elde edilmesi için yapılan çalışmaların bütünüdür. Temel olarak optimizasyon problemlerinin çözümü için geliştirilen yaklaşımlar kesin çözüm yöntemleri ve yaklaşık çözüm yöntemleri olmak üzere iki grup altında incelenir. Kesin çözüm yöntemleri optimumu garanti ederler ancak NP-Zor yapıdaki büyük boyutlu gerçek hayat problemleri için kabul edilebilir bir zamanda çözüm üretmezler. Bu yüzden araştırmacılar, kabul edilebilir zamanda, kaliteli çözümler verebilmeleri nedeni ile yaklaşık çözüm yöntemlerinden meta-sezgisel yöntemlere büyük ilgi göstermektedir. Bu çalışmada uygulaması kolay ve etkili bir meta-sezgisel olan tekrarlı açgözlü algoritmaya yönelik detaylı bir analiz çalışması yapılmıştır. İlgili meta-sezgisel olan her bir operatör alt başlıklar halinde ele alınmıştır. Çeşitli problemler için geliştirilen tekrarlı açgözlü algoritma yaklaşımları avantaj ve dezavantajlarıyla okuyucuya sunulmuştur. Özetle bu çalışmada tabu, tavlama benzetimi, tekrarlı yerel arama gibi çeşitli meta-sezgiseller ile ortak birçok yönü bulunan tekrarlı açgözlü algoritma hakkında Türkçe literatüre katkıda bulunmak amaçlanmıştır.

Anahtar Kelimeler: Kombinatoriyal en iyileme, meta-sezgisel teknikler, tekrarlı açgözlü algoritma

A Comprehensive Analysis on the Iterated Greedy Algorithm

ABSTRACT: Generally, optimization is the achievement of the best result under certain constraints. Basically, the approaches developed for the solution of optimization problems are examined under two groups as exact solution methods and approximate solution methods. Exact solution methods guarantee the optimum, but they cannot produce solutions in an acceptable time for large-scale real-life problems of NP-Hard. Therefore, the researchers draw great attention to the meta-heuristic methods, which are one of the approximate solution methods, because they can provide quality solutions at an acceptable time. In this study, a detailed analysis of the iterative greedy algorithm, which is an easy-to-apply and effective meta-heuristic, was conducted. In the relevant meta-heuristic, each operator is discussed under subheadings. Iterative greedy algorithm approaches developed for various problems are presented to the reader with their advantages and disadvantages. In summary, in this study, it is aimed to contribute to the Turkish literature about the iterative greedy algorithm, which has many aspects in common with various meta-heuristics such as taboo, annealing simulation, and iterative local search.

Keywords: Combinatorial optimization, meta-heuristics, iterated greedy algorithm

Yunus DEMİR ([Orcid ID: 0000-0003-3868-1860](https://orcid.org/0000-0003-3868-1860)), Bursa Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Endüstri Mühendisliği Bölümü, Bursa, Türkiye

***Sorumlu Yazar/Corresponding Author:** Yunus DEMİR, e-mail: yunus.demir@btu.edu.tr

GİRİŞ

Tekrarlı Açgözlü Algoritmanın Tanımı

Yerel arama tabanlı bir algoritma olan tekrarlı açgözlü algoritma (TAA) ilk defa Jacobs and Brusco (1995) küme örtüleme (*set covering*) probleminin çözümü için kullanılmıştır. Farklı araştırma gruplarınca, uyarlanabilir büyük komşuluk araması (*Adaptive Large Neighborhood Search*) (Shaw 1997) veya yinelenen düzleştirme araması (*Iterated Flattening Search*) (Cesta et al. 2000) gibi farklı isimlerle geliştirilmiştir (Pranzo and Pacciarelli 2016). Genel olarak TAA, yıkım ve yapım adımları ile yinelenen bir açgözlü algoritma üzerinden çözüm uzayını araştırır (Kim et al. 2017). Olasılıklı yerel arama algoritmalarından biri olan TAA, çözüme rassal veya probleme özgü bir sezgisel ile oluşturulmuş tek bir uygun çözümle (S_0) başlar (Tasgetiren et al. 2013). Sonrasında iki ayrı tekrarlı faz ile daha iyi sonuçlara erişmeye çalışır. Bunlardan ilki, önceki aday çözümlerin belirli parçalarının çıkarıldığı yıkım (*destruction*) fazı ve diğeri, çıkarılan parçaların açgözlü yapıcı sezgiseller yardımı ile tamamlanarak yeni aday çözümün (S_C) oluşturulduğu yeniden yapılandırma (Y-Y) (*Reconstruction*) fazıdır. Daha sonra, önceden tanımlanan kabul kriterine göre mevcut çözümün (S_i), aday çözüm (S_C) ile değiştirilip değiştirilmeyeceğine karar verilir (Ruiz and Stützle 2007). Bu iki temel faza ilaveten başlangıç çözümü ve yeni oluşan aday çözümü iyileştirmek üzere ana döngüden önce ve kabul aşamasından önce isteğe bağlı olarak yerel arama fazı uygulanabilmektedir (Lin et al. 2011). Bu aşamalar durdurma kriteri karşılanana kadar tekrarlanır. TAA'nın bu genel yapısı Çizelge 1'de sunulmuştur.

Çizelge 1. Tekrarlı açgözlü algoritmasının genel yapısı

Prosedür: Tekrarlı Açgözlü Algoritma

$S_0 :=$ Başlangıç çözümün oluşturulması	
$S_i :=$ Yerel arama(S_0)	% Seçmeli
While (Sonlandırma kriteri karşılanmadığı sürece)	
$S_D :=$ Yıkım(S_i)	
$S' :=$ Yeniden yapım(S_D, S_R)	
$S_C :=$ Yerel arama(S')	% Seçmeli
$S_i :=$ Karar kriteri(S_C, S_i)	
endWhile	
return $S_{max/min}$	

TAA'nın diğer yöntemlere göre avantajları şöyle sıralanabilir: TAA, ilgilenilen probleme ait spesifik bilgiye dayanmaz, karmaşık algoritmalarda olduğu gibi çok fazla kontrol parametresi gerektirmez, başka meta-sezgisellerin bileşenleri (operatörleri) ile uyumlu bir şekilde çalışabilecek esnekliğe sahiptir (García-Martínez et al 2014). Başka bir deyişle hibritleme daha kolaydır.

Literatür Taraması

TAA farklı birçok kombinatoriyal/kesikli optimizasyon probleminde kullanılmıştır. Bunlardan bazıları şu şekildedir: Yük trenlerinin çizelgelenmesi (Yuan et al. 2008), köşe örtme (*vertex cover*) problemi (Bouamama et al. 2012), yazılım/donanım işbirlikçi tasarımı problemlerinden olan yazılım/donanım bölümlenme problemi (Lin 2013), dağıtık hesaplama sitemine görev atama problemi (Kang et al. 2013), çapraz-sevkiyat kamyon çizelgeleme problemi (Naderi et al. 2014), içecek dağıtımını yapan bir firmanın müşteri bölümlendirme (*customer segmentation*) üzerine bir gerçek hayat problemi (bir tür kümeleme problemi) (Huerta-Muñoz et al. 2017), karesel çoklu sırt çantası (*quadratic multiple knapsack*) problemi (García-Martínez et al. 2014).

Ancak TAA daha ziyade çizelgeleme problemleri için uygulandığı gözlemlenmiştir. Çizelgeleme problemlerinin çözümü için ilk defa (Ruiz and Stütze 2007) tarafından kullanılmıştır. Bu çalışmayı takiben başta akış tipi çizelgeleme problemi olmak üzere, çeşitli tipteki çizelgeleme problemleri için TAA tercih edilmiştir.

Nouri and Ladhari (2016) yine bloklama kısıtı altında akış tipi çizelgeleme problemini ele almışlardır. Pan and Ruiz (2014) permutasyon akış tipi çizelgeleme probleminin özel bir halini ele almıştır. Yazarlar; fiberglas işleme, dökümhane, entegre devre üretimi gibi bir çok alanda karşılaşılan, işler arası bazı makinelerin durdurulmasının ekonomik açıdan maliyetli veya teknik açıdan zor olduğu durumu ele alınmıştır. İlgili çalışmada ekleme komşuluk operatörü için yeni bir hızlandırma (*speed-up method*) yaklaşımı ve yıkım-yeniden yapım prosedürü geliştirmişlerdir. (Pan et al 2008) permutasyon akış tipi çizelgeleme problemi için toplam akış süresini minimize etmek istemişlerdir. Yazarlar çalışmalarında “*reference local search*” ismini verdikleri yeni bir komşuluk arama yaklaşımı geliştirmişlerdir. Rajendran and Ziegler (2004)’in “*job-index based insertion scheme*” olarak isimlendirdikleri metottan esinlenen bu yaklaşım ile çözüm uzayı genişletilmektedir. Temel olarak ekleme operasyonunun referans bir çözümden alınan referans pozisyona bağlı olarak yapılması esasına dayanmaktadır. Riahi et al. (2020) benzer şekilde gerçek hayatta fiberglas işleme sürecinde ve dökümhanelerde karşılaşılan bir problemi ele almışlardır. Yazarlar boşluksuz (Bir makine üzerindeki işi bitirdikten sonra boş kalmaksızın diğer işe başlaması gerekliliğini ifade etmektedir) (*no-idle*) permutasyon akış tipi çizelgeleme problemi için TAA kullanmıştır. Bu durum. Al Aqel et al. (2019), TAA’yı esnek atölye tipi çizelgeleme ortamında Cmax minimizasyonu için kullanmışlardır. Problemi iki alt probleme ayırarak ele alan yazarlar her alt problem için iki faz halinde TAA uygulamışlardır. Modifiye TAA ismini verdikleri yaklaşımı literatürde yapılmış diğer çalışmalar ile karşılaştırmışlardır. Pranzo and Pacciarelli (2016) bloklama kısıtı altında iki farklı versiyonu için atölye tipi çizelgeleme problemini ele almışlardır. Kim et al. (2017) sıra bağımlı hazırlık süreli atölye tipi çizelgeleme problemi için toplam aile akış süresini minimize etmek üzere TAA’yı kullanmıştır. Deng and Gu (2014) sıra bağımlı hazırlık zamanlı tek makine çizelgeleme probleminin çözümü için TAA’yı kullanmıştır. Yazarlar çözüm hızını artırma adına ekleme-komşuluk operatörü için Tasgetiren et al. (2009) tarafından geliştirilen hızlandırma metotlarını kullanmışlardır. Ayrıca değiş-tokuş komşuluk operatörü için kendileri eleme kuralları ve hızlandırma metotları geliştirmiştir. Framinan and Leisten (2008) ele aldıkları problem için değişken komşuluk arama ve tekrarlı açgözlü arama algoritmalarını hibretleyerek bir çözüm yöntemi oluşturmuştur. Değişken komşuluk araması, algoritmanın yerele takılmasını engellemek amacıyla kullanılmıştır. Tasgetiren et al. (2013) ele aldığı akış tipi çizelgeleme probleminin çözümü için üç farklı tipte TAA önermiştir. Bunlar: Geleneksel TAA, değişken TAA ve diferansiyel evrim algoritması ile hibritleşmiş TAA. Değişken TAA, değişken komşuluk arama algoritmasından esinlenerek oluşturulmuştur. Koşum boyunca yıkım aşamasında sökülecek parça sayısının değişkenliği esasına dayanır. İlk başta $d=1$ kabul edilir ve her iterasyonda bu değer bir artırılır ($d=d+1$). Eğer iyileşme olursa bu değer tekrar bir indirilir. Aksi takdirde $d_{max}=n-1$ olana kadar artırılır. Yazarlar tarafından önerilen üçüncü yaklaşımda (Bouamama et al. 2012)’nin çalışmasına benzer şekilde popülasyon temelli bir yapı kurulmuştur. TAA standart hali ile tek çözümlü bir sezgiseldir ancak Bouamama et al. (2012) klasik TAA’yı popülasyon temelli bir yapıya modifiye etmiştir. Tasgetiren et al. (2013) popülasyondaki bireye TAA uygulanma olasılığı ve yıkım boyutunu optimize etmek üzere diferansiyel evrim algoritması kurulmuştur. Li et al. (2019), her fabrikada tek bir robotun olduğu ve maksimum yayılımın (Cmax) en aza indirilmesi gereken dağıtılmış permutasyon akış tipi çizelgeleme problemini ele almıştır. İlgili problemde robotlar, her işi önceki makineden sonraki makineye aktarmak için kullanılmaktadır. Makineler arasında bir bloklama kısıtı bulunmaktadır. Bu kısıt çerçevesinde, işler robotu beklerken

tamamlanan makinede kalmalıdır. Bu çalışmada ayrıca yükleme ve boşaltma süreleri ve bozulma süresi (*deteriorating time*) de dikkate alınmıştır. Kısaca bilgi verilen bu çalışmalar Graham et al. (1979) tarafından oluşturulan çizelgeleme sınıflandırma metodolojisine göre derlenerek ve Tablo 1’de sunulmuştur.

Çizelge 1. Çizelgeleme problemleri için önerilen TAA literatürü

Referans	Gösterim
Framinan and Leisten 2008	$F pmtn \Sigma T_j$
Fanjul-Peyro and Ruiz 2010	$R C_{max}$
Urlings et al. 2010	$FFm S_{ij}, r_j C_{max}$
Lin et al. 2011	$R S_{mjk}, d_j \Sigma T_j$
Tasgetiren et al. 2013	$F pmtn, no_idle C_{max}, \Sigma F_j$
Rodriguez et al. 2013	$R \Sigma w_j C_j$
Lin et al. 2013	$DF pmtn C_{max}$
Fernandez-Viagas and Framinan 2014	$DF pmtn C_{max}$
Pan and Ruiz 2014	$F prmu; mixed\ no_idle C_{max}$
Deng and Gu 2014	$1 s_{ij} \Sigma w_j T_j$
Nouri and Ladhari 2016	$Fm block C_{max}$
Lin et al. 2016	$Rm C_{max}; Pw_j C_j; \Sigma w_j T_j$
Karabulut 2016	$F prmu \Sigma T_j$
Kim et al. 2017	$JS s_{ij} \Sigma F_j$
Pranzo and Pacciarelli-2016	$JS block C_{max}$
Aqel et al. 2018	$FJS C_{max}$
Lee 2018	$Pm s_{ij} \Sigma w_i T_i$
Ribas et al. 2019	$F block \Sigma T_j$
Al-Behadili et al. 2019	$F brkdwn utility, stability, robustness$
Li et al. 2019	$DF block C_{max}$
Riahi et al. 2020	$F no_idle \Sigma T_j$
Deng et al. 2020	$JS no_wait \Sigma F_j$
Pinheiro and Arroyo 2020	$1 s_{ij} \Sigma T_j$

Aç gözlü algoritmanın tek amaçlı problemler için etkili çözümler ürettiği yukarıda sunulan literatür göstermektedir. Aç gözlü temelli yaklaşımlar çok amaçlı problemler için de uygulanmış ve başarılı çözümler elde edilmiştir (Lin et al. 2016). Örneğin Framinan and Leisten (2008) ve Minella et al. (2011) permutasyon akış tipi çizelgeleme probleminde, C_{max} ve toplam akış süresini minimize etmek için TA kullanmıştır. Ciavotta et al. (2013) aynı problemin sıra bağımlı hazırlık süreli versiyonu için pareto açgözlü bir yöntem önermiştir. Ying et al. (2014) iki amaçlı tekrar girişli (*reentrant*) hibrit akış tipi çizelgeleme problemi için pareto aç gözlü önermiştir. Lin et al. (2016), çok amaçlı ilişkisiz paralel makine çizelgeleme problemi için, Ying et al. (2014) tarafından oluşturulan tekrarlı pareto açgözlü algoritmaya tabu listesi eklenerek genişletilmiş bir versiyonunu önermişlerdir. Al-Behadili et al. (2019) üç amaçlı dinamik permutasyon akış tipi çizelgeleme problemi için TAA’yı önyargılı rastgeleleştirme süreci (*biased randomisation process*) ile hibritleyerek robust bir yaklaşım geliştirmişlerdir.

MATERYAL VE METOT

Başlangıç Çözüm

TAA’nın bu bileşeninde, araştırmacılar genelde problemle ilgili daha önce geliştirilmiş kısa zamanda uygun çözüm üreten sezgisellere başvurmuşlardır. Başlangıç çözümü rassal bir şekilde oluşturan tabu arama ve tavlama benzetimi gibi algoritmaların aksine TAA başlangıç çözüm için yapıcı bir sezgisel kullanmaktadır. Bu yaklaşım TAA açısından ciddi bir avantajdır (García-Martínez et al.

2014). Başlangıç çözüm için ve yeniden yapım için uygulanan açgözlü yaklaşımlar problem özgüdür. Bu bölümde TAA literatürü başlangıç çözüm operatörü bağlamında incelenmiştir.

Ribas et al. (2019) toplam gecikmenin minimize edilmeye çalışıldığı blok kısıtlı paralel akış tipi çizelgeleme problemi için geliştirdikleri TAA için, çözüm kalitesi ve CPU zamanı açısından altı farklı yapıcı sezgiseli (NEH1, NEH2, RC3_m, FPD_m, A3B2_m, RC4) değerlendirmişlerdir. Ayrıca yazarlar bu çalışmada algoritmanın farklı varyasyonlarını test ederek algoritmanın bileşenlerinin performans üzerine etkisini araştırmışlardır. Sonuçta başlangıç çözümün kalitesinin TAA ile ulaşılan son sonucun kalitesine etkisinin olmadığı çıkarımında bulunmuşlardır. Bu yüzden kaliteli başlangıç çözüm elde etmek için harcanan zamanın daha fazla iterasyona ayrılmasının daha faydalı olacağı sonucuna varmışlardır. Akış tipi çizelgeleme problemi için geliştirilen TAA'larda başlangıç çözüm operatörü olarak genelde Nawaz et al. (1983) tarafından geliştirilen NEH (Nawaz, Enscore and Ham) algoritması kullanılmıştır. NEH, akış tipi çizelgeleme problemi için geliştirilmiş diğer en çok bilinen sezgiseller olan; Palmer algoritması (Palmer 1965), Johnson kuralı (Johnson 1954) ve CDS metodundan (Campbell et al 1970) daha iyi olduğundan birçok yazar tarafından (Pan and Ruiz 2014; Lin et al. 2013; Nouri and Ladhari 2016; Li et al. 2019) NEH tabanlı yaklaşım tercih edilmiştir. Sonrasında klasik NEH'in yeni NEH1 ve NEH2 olmak üzere yeni versiyonları türetilmiştir (Naderi and Ruiz 2010). Rad et al. (2009), akış tipi problemler için başlangıç çözüm olarak FRB1-FRB5 olarak isimlendirdikleri 5 farklı metod önermiştir. Geliştirilen metotlar, NEH'e göre dahi iyi performans gösterse de hesaplama maliyeti daha yüksektir (Pan and Ruiz 2014). Pan and Ruiz (2014), başlangıç çözüm için FRB4'i özelleştirerek FRB4k ismini verdikleri sezgiseli kullanmıştır. Bu yaklaşıma göre, p pozisyonuna bir iş yerleştirildikten sonra p pozisyonu etrafındaki k adet iş, daha iyi sonuç bulmak üzere muhtemel tüm pozisyonlar için denir. Burada k parametresinin ayarlanması çözüm kalitesi ve hesap yükü arasında dengenin sağlanmasında önemli bir etkidir. Al-Behadili et al. (2019) permutasyon akış tipi çizelgeleme probleminin dinamik versiyonu için Juan et al. (2014) tarafından geliştirilmiş NEH algoritmasından türeyen BR-NH isimli algoritmayı kullanmıştır. Bu algoritma klasik NEH algoritması ile biased randomiastion sürecinin kombinasyonundan oluşmaktadır. Kim et al. (2017) başlangıç çözüm için Choi and Korkmaz (1997) tarafından geliştirilmiş sıra bağımlı hazırlık süreli atölye tipi çizelgeleme sezgiseli kullanmıştır. Deng and Gu (2014) tek makine çizelgeleme problemi ağırlıklı gecikmeyi minimize etmek için geliştirdikleri TAA için başlangıç çözümde hazırlık zamanlı görünen gecikme maliyeti (apparent tardiness cost with setup-ATCS) kullanmışlardır. Pranzo and Pacciarelli (2016) bloklu atölye tipi çizelgeleme problemini, Mascis and Pacciarelli (2002) tarafından geliştirilen ve ayrık şebeke modelinden özelleşmiş alternatif şebeke modeli (alternative graph model) şeklinde modellemiştir. Bahsedilen bu şebeke modeli ile pratik hayatta karşılaşılan birçok karmaşık özel çizelgeleme durumları başarı ile çözülebilmektedir. Bu özel durumlara örnek olarak; bir iş için maksimum depolama zamanının bulunması, bir işin ardışık operasyonları arası zaman kısıtının bulunması, bir makinenin sınırlı yarı ürün depolama kapasitesi olması (buffer), belirli zaman aralıklarında makinelerin çalışmıyor olduğu durumlar veya işlerin işlenmek için daha önceden tanımlanmış bir zaman aralığında (varış ve teslimat tarihleri belli) yapılması gerektiği durumlar verilebilir (Pacciarelli 2002). Yazarlar başlangıç çözümüm bu model üzerine kurgulamıştır. Deng et al. (2020) atölye tipi çizelgeleme için popülasyon temelli yaklaşım kullanmıştır. Başlangıçta belirli sayıda uygun çözüm oluşturmuştur. Popülasyondaki her birey için TAA prosedürü uygulanmıştır. Daha iyi çözüme ulaşıldığında birey yeni aday ile değiştirilmiştir. Benzer şekilde AlAqel et al. (2018) esnek atölye tipi çizelgeleme probleminin çözümü için popülasyon tabanlı bir yaklaşım kullanmıştır. Klasik yaklaşımlardan farklı olarak çözüm uzayının keşfi (*exploration*) için popülasyon içerisinde farklı kalitelerde bireylerin iyiden en kötüye doğru sıralayarak oluşturmuştur. Bunun için

öncelikle bir uygun çözüm oluşturulmuş ve daha sonra bu çözüm optimize edilerek çözüm kalitesi yükseltilecek ikinci birey ve üçüncü birey şeklinde başlangıç çözüm oluşturma süreci tamamlanmıştır.

Çizelgeleme problemlerinin dışında diğer problem tiplerine bakıldığında, benzer şekilde probleme özgü geliştirilmiş etkili yapıcı yöntemlerin başlangıç çözüm olarak kullanıldığı görülmektedir. García-Martínez et al. (2014) karesel çoklu sırt çantası problemi için Hiley and Julstrom (2006) tarafından geliştirilen açgözlü yapıcı algoritmayı kullanmıştır. Huerta-Muñoz et al. (2017) bir tür kümeleme problemi için başlangıç operatöründe Hartigan & Wong (1979) tarafından geliştirilen k-means algoritmasından türettiği ettiği Modified k -Means kullanmıştır.

Yıkım (Destruction) ve Yeniden Yapılandırma (Reconstruction)

Tekrarlı yerel arama sezgiselindeki (*iterated local search-ILS*) perturbasyon operatörünün Ruiz and Stutzle (2007) tarafından uyarlanmış halidir (Deng and Gu 2014). Her defasında en baştan açgözlü bir yaklaşım ile aday çözümler oluşturmak yerine, her iterasyonda elde edilen başarılı çözümlerin parçalarını muhafaza etmek esasına dayanmaktadır. Ayrıca bu yaklaşım ile çözüm daha hızlı oluşturulmaktadır (Hoos and Stutzle 2004). Bu aşama, n işten oluşan permutasyonuna (S-uygun çözüm sıralamasına) uygulanır. Tekrarsız bir şekilde d adet iş rassal bir şekilde seçilir. Seçilen bu işler S permutasyonundan çıkarılır. Bu işlerin ardından iki alt permutasyon oluşur. Bunlardan ilki $n-d$ işten oluşan kısmi sıralama (partial sequence) S_D ve diğeri çıkarılan d işten oluşan küme S_R 'dir.

Başlangıç çözüm oluşturma aşamasında olduğu gibi, yeniden yapım aşamasında da çıkarılan işler (S_R) probleme özgü yapıcı bir sezgisel ile S_D içerisine tekrar eklenerek aday çözüm S_C oluşturulur (Ruiz and Stutzle 2007). TAA'da çizelgeleme problemlerinde yeniden yapım aşamasında genelde NEH (Nawaz et al. 1983) sezgiseli kullanılmıştır. Bu sezgisele göre S_R alt permutasyonundaki ilk iş S_D alt permutasyonundaki tüm muhtemel pozisyonlara ($n-d+1$) yerleştirilir. Bunlar arasından en iyi amaç değerli permutasyon, yeni parçalı permutasyon (S_D) olarak seçilir. Bu işlem S_R alt permutasyonundaki tüm işler bitene kadar devam eder. Bu şekilde yeni aday çözüm S_D oluşturulur. Yıkım-yeniden yapılandırma aşamasına ait pseudo kod Çizelge 2'de verilmiştir (Pan et al. 2008).

Çizelge 2. Yıkım-yeniden yapılandırma aşamasına ait pseudo kod

Prosedür: Yıkım-Yeniden Yapılandırma (S)

Yıkım

for $i:=1$ to d **do**

$P[1, n+1-i]$ setinden rassal bir j pozisyonu seç;

$S_D := j$ pozisyonundaki işi S' 'den çıkar. Çıkarılan işi S_R setine ekle;

endfor

Yeniden Yapılandırma

for $i:=1$ to d **do**

S_R setinden rassal bir iş seç

$S_D :=$ Seçilen işi S_D alt setindeki muhtemel tüm pozisyonlarda deneyerek en iyi alternatifi belirle

endfor

return S'

Mevcut çözümden çıkarılacak parça sayısı (d) genellikle deneysel çalışmalar sonucunda sabit bir sayı olarak belirlenmektedir (Framinan and Leisten 2008; Fanjul-Peyro and Ruiz 2010; Kang et al. 2013). Ancak bu değer belirlenirken farklı yaklaşımlar uygulandığı gözlemlenmiştir. Rodriguez et al. (2013) çıkarılacak parça sayısını (d), problem boyutuna ve sıfır bir aralığında belirlenen ve yıkım seviyesi olarak isimlendirilen (*degree of destruction*) ($q \in [0,1]$) iki parametrenin çarpımı ($q \cdot n$) ile bulmaktadır. Bu bölümün devam eden kısımlarında önce yıkım aşaması sonra yapım aşaması detaylı bir şekilde işlenmiştir.

Ribas et al. (2019) tarafından perturbation operatörü olarak da isimlendirilen yıkım aşamasının iyi tasarlanması çözüm kalitesi üzerinde önemli etkiye sahiptir. Öncelikle sökülecek bileşen sayısı, mevcut çözümü yerel optimumdan çıkaracak kadar büyük ve mevcut çözüm ile kazanılmış iyi özellikleri kaybettirmeyecek kadar küçük olmalıdır. Aksi takdirde çözüm yerele takılabilir ya da bulunan yeni çözüm elde edilen yerel çözümde daha kötüye gidebilmektedir. Karabulut (2016) yıkım boyutunu sabit ($d=4$) tutmuştur (Bu değer, Ruiz and Stützle (2007) tarafından deneysel bulunmuştur). Fanjul-Peyro and Ruiz (2010) “*tabu-enhanced procedure*” ismini verdiği yaklaşım ile çıkarılacak bileşenlerin seçimini rassal olarak değil de olasılıksal temellere dayalı bir sezgisel ile yapmaktadır. Her ne kadar TAA problem bağımlı bir sezgisel olmasa da yazarlar probleme özgü veriyi kullanarak daha iyi sonuçlar elde etmek adına daha akıllı seçimler yapılabileceğini göstermişlerdir. Benzer şekilde Riahi et al. (2020) seçilecek işlerin rasgele seçildiği klasik yıkım prosedüründen farklı olarak, gecikme güdümlü bir iş seçme yapısı geliştirmişlerdir. Bu şekilde sıralamadaki en problemlili bileşenler düzeltilmeye çalışılmıştır. Ayrıca yazarlar yıkım aşamasında sökülecek iş sayısını (d) sabit almayıp problem boyutuna göre belirli bir fonksiyon ile tespit etmektedir. Elliden az işli problemler için $d=3$, 50’den büyük problemler için $B*n$ (B algoritma parametresi) şeklinde belirlemiştir. García-Martínez et al. (2014) Tabu listeli yeni bir yıkım mekanizması geliştirmiştir. Bu yaklaşıma göre klasik yıkımdan farklı olarak sökülen parçaları kısa dönemli bir hafızada, bir listede kaydedip aynı parçaların tekrar sökülmesine engel olunmaktadır. Benzer yaklaşımı Lin et al. (2016) çok amaçlı ilişkisiz paralel makine çizelgeleme problemi için kullanmıştır. Lee (2018) özdeş paralel makine çizelgeleme problemi için geliştirdiği TAA’da yıkım için iki farklı yaklaşım kurgulamıştır. Bunlardan ilkinde paralel makine seti içinden C_{max} değeri en küçük olandan rastgele bir iş seçmiştir. Diğerinde tüm makinelerden rastgele bir iş seçimi yapmıştır.

Yeniden yapılandırma aşaması sökülen parçaların belirli bir düzen çerçevesinde kısmi çözüme eklendiği süreçtir. Bu aşamada açgözlü yapısal sezgiseller kullanılmaktadır. Çıkarılmış işler seti S_R içinden işler, rastgele veya belirli bir sezgisel göre yapılabilmektedir. Pan et al (2008) yeniden yapılandırmada NEH sezgiselini kullanmıştır. Bu yaklaşıma göre seçilen iş tüm muhtemel sıralara ($n-d-1$) yerleştirilmiş ve en iyi sonuç veren sıralama seçilmiştir. Bu işlem S_R kümesi boşalana kadar devam ettirilmiştir. Pan and Ruiz (2012) etkili tasarlanmış açgözlü yeniden yapılandırmanın TAA’nın performansı üzerinde önemli bir etkiye sahip olduğu vurgulanmıştır. Pinheiro and Arroyo (2020) α $[0,1]$ arasında rassal bir değer olmak üzere $\alpha*n$ kadar işi çıkarıp çıkan her bir işi permutasyondaki tüm sıralara yerleştirmiştir.

TAA en genel yapısında çıkarılan işler rassal bir şekilde seçilerek parçalı çözüme eklenir. Ancak bazı çalışmalarda bu işlem için daha etkili sonuçlar veren yaklaşımlar geliştirilmiştir (Framinan and Leisten 2008). Deng and Gu (2014), Fernandez-Viagas and Framinan (2014), Naderi et al. (2014) da aynı şekilde “*Insertion Perturbation*” ismini verdikleri yaklaşımda, rassal şekilde d kadar işi uygun çözümden çıkarmış ve çıkan işleri sırayla çıkarıldığı yer hariç diğer tüm pozisyonlara eklemiştir. Yazarlar bu sürecin zorluğunu $O(dn^2)$ olarak hesaplamışlardır. AlAqel et al. (2018) yeniden yapımlı operatörü olarak sıralama kurallarını kullanmıştır. Yazarlar aynı probleme yönelik yaptıkları başka bir çalışmada problemi iki alt probleme (makine atma ve sıralama) ayırmış ve her biri için yeniden yapımlı operatörü olarak sıralama kurallarını kullanmıştır (AlAqel et al. 2019). Rodriguez et al. (2013) özdeş olmayan paralel makine çizelgeleme problemi için 7 farklı sezgisel kullanmıştır. Pan and Ruiz (2014), FRB4k (Akış tipi çizelgeleme başlangıç çözüm oluşturmak için geliştirilmiştir) isimli sezgiselden esinlenerek sökülen iş tekrar eklendikten sonra ilgili işin öncesi ve sonrasındaki işler de permutasyondaki tüm sıralara eklenmiştir. Bu küçük modifikasyon ile oluşturdukları “*eDC*” ismini verdikleri operatör ile kayda değer iyileştirme sağlamışlardır.

Yerel Arama

Yerel arama, yıkım ve yeniden yapım aşamasından sonrası üretilen her bir çözümü daha da iyileştirmek üzere opsiyonel olarak kullanılan TAA operatörüdür. Bu bölümde, literatürde yerel arama sürecinde kullanılan bazı yaklaşımlara yer verilmiştir.

Kim et al. (2017) ele aldıkları atölye tipi çizelgeleme problemini ayrık şebeke ile ifade etmişlerdir. Yazarlar, Nowicki and Smutnicki (1996) tarafından sunulan komşuluk üretme metodundan esinlenerek kritik yol üzerindeki komşu operasyonları değişimine dayalı iki farklı komşu çözüm üretmişlerdir.

Şebekenin kritik yolu Cmax değerini veren ardışık operasyonlardan oluşmaktadır. Pan and Ruiz (2014) sökülecek işin rastgele seçildiği klasik yaklaşımın yerine Pan et al. (2008) tarafından geliştirilen “referenced local search” isimli yapıyı kullanmışlardır. Bu yaklaşımda sökülecek iş rastgele değil de referans listedeki sıraya göre seçilmektedir. Örneğin referans çözüm {3,1,5,2,4} ve mevcut çözüm {4,2,5,1,3}. Önce 3 işi çıkarılıyor ve muhtemel tüm sıralara ekleniyor. Sonra 1 işi çıkarılır ve liste tamamlanana kadar bu şekilde devam edilir. Temel mantık sökme işleminin rastgele değil de referans bir çözüm tarafından yönetilmesidir. Bu yaklaşım birçok çalışmada kullanılmıştır (Deng and Gu 2012). Ancak Tasgetiren et al. (2009) bu yaklaşımın rassal yaklaşımdan farksız olduğunu iddia etmiştir. Deng and Gu (2014) çalışmalarında yerel arama için kullanılan komşuluk yapıları ile ilgili literatürde yapılan değerlendirmelere yer vermiştir. İlgili çalışmada, problem tipine göre komşuluk yapılarının birbirlerine üstünlük sağladıkları ortaya koyulmuştur. García-Martínez et al. (2014) karesel çoklu sırt çantası problemi için ikisi literatürden ikisi özgün yerel arama metodu kullanmıştır.

Kabul Kriteri

Bu aşamada, yıkım-yeniden yapılandırma ve yerel arama aşamalarından sonra oluşan aday çözüm ve mevcut çözüm amaç değerlerine göre karşılaştırılır. Eğer daha iyi bir sonuç elde edilmiş ise aday çözüm yeni mevcut çözüm olarak güncellenir. Algoritmanın bu açgözlü yapısı, çözümün çabucak yerele takılmasına sebep olabilir. Bu olumsuzluğu aşmak için bazı araştırmacılar (Kim et al. 2017; Pranzo and Pacciarelli 2016; Nouri and Ladhari 2016; Pan and Ruiz 2014; Pan et al 2008; Lin et al. 2013; Li et al. 2019) tavlama benzetimine benzer sıcaklık tabanlı bir kabul kriteri kullanmışlardır. Belirli bir olasılık ile kötü sonuçların da kabul edildiği bu kriter, minimizasyon problemi için şu şekilde çalışmaktadır:

$$\Delta = F(S_i) - F(S_c)$$

Burada $F(S_i)$ ve $F(S_c)$ yeni çözüm ve mevcut çözüme ait amaç değerleridir. Δ , mevcut çözüm amaç değeri ile aday çözüm amaç değeri arası fark iken; $\Delta > 0$ durumunda aday çözüm kabul edilir. Aksi takdirde aşağıda verilen fonksiyon ile elde edilen ihtimal ile kabul edilir:

$$\text{Rnd}() \leq \exp\{\Delta / \text{Temp}\}$$

Bunun için [0,1] aralığında düzgün dağılıma göre rassal bir sayı üretilir ($\text{Rnd}()$). Yeni oluşan çözümün kabulü sıcaklık diye isimlendirilen parametre (Temp) tarafından kontrol edilir. Algoritma koşumu boyunca kabul kriterine ait değer birçok çalışmada sabittir. Ancak Riahi et al. (2020) zamana bağlı dinamik bir kabul kriteri geliştirmişlerdir. Bu yaklaşımda, koşumun ilk zamanlarında iyi olmayan (*non-improving*) çözümlerin kabulü için oran yüksek tutulmakta ve sona doğru sifıra yaklaştırılmaktadır. Temel fikir, arama alanını erken yinelemelerde daha fazla keşfetmek ve sonraki yinelemelerde iyileşmeye odaklanmaktır. Bir diğer kabul kriteri rassal yürüyüştür (Pranzo and Pacciarelli 2016). Bu kriter gere göre her uygun aday çözüm kabul edilir. Rodriguez et al. (2013) bu kriteri modifiye ederek uygun çözümlerin yarısını kabul etmektedir. Yazarlar, amaç fonksiyonundan bağımsız bir değerlendirme yapan bu yaklaşım için 0-1 aralığında düzgün dağılan bir değer (p_a) üretmiştir. Eğer bu değer 0,5'ten küçük ise ($p_a < 0,5$) yeni çözüm olarak kabul etmiş aksi takdirde mevcut çözümlerle iterasyona devam etmiştir.

Pinheiro and Arroyo (2020) ve Pan et al (2008) kabul kriteri olarak açgözlü bir yaklaşım benimsemiştir. Bu yaklaşım aday çözümler arasından en iyi sonucun seçilmesi (*greedily made*) esasına dayanmaktadır.

Durdurma Kriteri

TAA'da yapım ve yıkım aşamaları, önceden tanımlanmış durdurma kriteri sağlanana kadar devam eder. Bu bölümde, literatürde geliştirilmiş çeşitli durdurma kriterlerine yer verilmiştir. Durdurma kriterlerinin en genel hali, algoritmanın belirlenen iterasyon sayısına (Kim et al. 2017; Lin et al. 2011) veya belirlenen CPU zamanına ulaşınca (Fanjul-Peyro and Ruiz 2010; Lin et al. 2013; Pranzo and Pacciarelli 2016) durdurulması şeklindedir. Pinheiro and Arroyo (2020), belirli bir iterasyon sayısınca ilerleme kaydedilmediği zaman koşumu durdurmuştur. Buna karşın Lin et al. (2011) iyileşme oluşmadığı zaman mevcut çözümü (S_i) tamamıyla yeniden oluşturmakta ve maksimum iterasyon sayısına veya ulaşılmak istenen amaca ulaşınca durdurmaktadır. Çeşitliliği artırmak ve böylece algoritmanın yerele takılmasına engel olmak için geliştirilen bu yaklaşım Naderi et al. (2014) tarafından da kullanılmıştır. Ancak bu prosedürü uygulamada dikkat edilecek kritik nokta tamamıyla yeni bir çözümün ne zaman oluşturulacağına karar verilmesidir. Bunun için genelde Minella et al. (2011) tarafından önerilen yukarıda da bahsedilen yaklaşım kullanılır. Yani kullanıcı tarafından tanımlanan sayıda iterasyon sonrasında iyileşme olmaz ise bu mekanizma tetiklenir. Çok amaçlı problemlerin çözümünde (pareto temelli çözüm önerisi geliştirilen çalışmalarda) daha önceden belirlenen baskın çözümler kümesi kapasitesine ulaşıncaya durulmaktadır (Lin et al. 2016).

Eleme ve Hızlandırma

Aynı hareketleri veya amaç açısından olumlu katkı yapmayacak hareketleri arama süresinden arındırarak hesaplama karmaşıklığını azaltması yönünde yapılan çalışmalardır. Bu tarzda çalışmalar karmaşıklığı hatırı sayılır bir şekilde düşürdüğü gözlemlenmiştir. Örneğin Tasgetiren et al. (2013a) geliştirdiği eliminasyon yapısı ile tüm ekleme komşuluk karmaşıklığını $O(n^3m)$ 'den $O(n^2m)$ 'e düşürmüştür (n iş, m makine). Deng and Gu (2014) ele aldıkları çizelgeleme problemi için geliştirdikleri algoritmanın yerel arama operatöründe aynı tamamlanma zamanına (C_i) ve aynı ağırlığa (w_i) sahip elemanların değişimini engellemişlerdir. Benzer şekilde Tasgetiren et al. (2009) ekleme tipi komşuluk arama hareketi için hızlandırıcı geliştirmiştir. Li et al (2009) toplam akış süresi minimizasyonu için önerdikleri algoritma içerisinde hızlandırıcı bir mekanizma geliştirmişlerdir. Karabulut et al. (2019), Li et al (2009) tarafından geliştirilen hızlandırma mekanizmasını toplam gecikme minimizasyonu için modifiye etmiştir. Yerel arama operatörünün yanısıra hızlandırma yaklaşımları diğer algoritma bileşenleri için de kullanılmıştır. Örneğin Taillard (1990), iki amaçlı akış tipi çizelgeleme problemi için geliştirdiği TAA'nın yapım aşamasını hızlandırmak üzere bir yaklaşım ortaya koymuştur. Karmaşıklığın $O(mn^2)$ 'ye düşürüldüğü bu yaklaşım birçok çalışmada (Tasgetiren et al. 2013; Lin et al. 2013) kullanılmıştır. Fernandez-Viagas and Framinan (2014) sınırlandırılmış arama mekanizması ismini verdiği bir yaklaşım geliştirmiştir. Belirli bir iterasyona kadar ulaşılan en iyi sonuç değeri alt sınır olarak belirlenmiştir. Yapım ve yeniden yapılandırma aşamasında her bileşen eklenirken ulaşılan sonuç bu alt sınır ile karşılaştırılmış ve eğer daha kötü sonuç ortaya çıkıyor ise ekleme işlemine devam edilmemiştir.

SONUÇ

Bu çalışmada tekrarlı açgözlü arama algoritmasına yönelik kapsamlı bir analiz yapılmıştır. TAA uygulama kolaylığı, etkili sonuç vermesi ve birçok karar problemi için rahat uyarlanabilir yapısı sebebiyle son yıllarda tercih edilen bir meta-sezgisel tekniktir. Çalışmada algoritmaya ait operatörler ayrı ayrı alt başlıklar halinde incelenmiş ve literatürde kullanılan çeşitli yöntemler tanıtılmıştır. Türkçe

literatür bağlamında eksik olduğu düşünülen meta-sezgiseller alanına katkı yapacağı düşünülen bu çalışmanın hem akademik hem de pratik hayattaki araştırmacılara bir rehber olması amaçlanmıştır.

Çıkar Çatışması

Makale yazarları aralarında herhangi bir çıkar çatışması olmadığını beyan ederler.

Yazar Katkısı

Yazarlar makaleye eşit oranda katkı sağlamış olduklarını beyan ederler.

KAYNAKLAR

- Al Aqel, G., Li, X., Gao, L., Gong, W., Wang, R., Ren, T., Wu, G. 2018. Using Iterated Greedy with a New Population Approach for the Flexible Jobshop Scheduling Problem. In 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (pp. 1235-1239). IEEE.
- Al Aqel, G., Li, X., Gao, L. 2019. A modified iterated greedy algorithm for flexible job shop scheduling problem. Chinese Journal of Mechanical Engineering, 32(1), 21.
- Al-Behadili, M., Ouelhadj, D., Jones, D. 2020. Multi-objective biased randomised iterated greedy for robust permutation flow shop scheduling problem under disturbances. Journal of the Operational Research Society, 71(11), 1847-1859.
- Bouamama, S., Blum, C., Boukerram, A. 2012. A population-based iterated greedy algorithm for the minimum weight vertex cover problem. Applied Soft Computing, 12(6), 1632-1639.
- Campbell HG, Dudek RA, Smith ML. Heuristic algorithm for N-job, M-machine sequencing problem. Management Science Series B—Application 1970;16(10):B630–7.
- Cesta, A., Oddi, A., Smith, S.F.: Iterative flattening: a scalable method for solving multi-capacity scheduling problems. In: Proceedings of the National Conference on Artificial Intelligence, pp. 742–747 (2000)
- Ciavotta, M., G. Minella, and R. Ruiz. 2013. “Multi-objective Sequence Dependent Setup times Permutation Flowshop: A New Algorithm and a Comprehensive Study.” European Journal of Operational Research 227 (2): 301–313.
- Choi, I. C., and O. Korkmaz. 1997. “Job Shop Scheduling with Separable Sequence-Dependent Setups.” Annals of Operations Research 70 (1): 155–170.
- Deng G, Gu X. A hybrid discrete differential evolution algorithm for the no-idle permutation flow shop scheduling problem with makespan criterion. Computers and Operations Research 2012; 39(9): 2152–2160.
- Deng, G., Su, Q., Zhang, Z., Liu, H., Zhang, S., Jiang, T. 2020. A population-based iterated greedy algorithm for no-wait job shop scheduling with total flow time criterion. Engineering Applications of Artificial Intelligence, 88, 103369.
- Fanjul-Peyro, L., Ruiz, R. 2010. Iterated greedy local search methods for unrelated parallel machine scheduling. European Journal of Operational Research, 207(1), 55-69.
- Fernandez-Viagas, V., Framinan, J. M. 2015. A bounded-search iterated greedy algorithm for the distributed permutation flowshop scheduling problem. International Journal of Production Research, 53(4), 1111-1123.
- Framinan, J. M., Leisten, R. 2008. Total tardiness minimization in permutation flow shops: a simple approach based on a variable greedy algorithm. International Journal of Production Research, 46(22), 6479-6498.

- García-Martínez, C., Rodríguez, F. J., Lozano, M. 2014. Tabu-enhanced iterated greedy algorithm: a case study in the quadratic multiple knapsack problem. *European Journal of Operational Research*, 232(3), 454-463.
- Graham, R. L., Lawler, E. L., Lenstra, J. K., Kan, A. R. 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. In *Annals of discrete mathematics* (Vol. 5, pp. 287-326). Elsevier.
- Hartigan, J. A., Wong, M. A. 1979. A K -means clustering algorithm. *Journal of the Royal Statistical Society, Series C: Applied Statistics*, 28 (1), 100–108
- Hiley, A., Julstrom, B. 2006. The quadratic multiple knapsack problem and three heuristic approaches to it. In *Proc. of the genetic and evolutionary computation conference (GECCO)* (pp. 547–552).
- Hoos, H.H., Stutzle, T.: *Stochastic Local Search—Foundations and Applications*. Morgan Kaufmann Publishers/Elsevier, San Francisco (2004)
- Huerta-Muñoz, D. L., Ríos-Mercado, R. Z., Ruiz, R. 2017. An iterated greedy heuristic for a market segmentation problem with multiple attributes. *European Journal of Operational Research*, 261(1), 75-87.
- Jacobs, L. W., Brusco, M. J., 1995. A local search heuristic for large set-covering problems. *Naval Research Logistics Quarterly*, 42(7), 1129–1140
- Johnson SM. Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly* 1954;1(1):61–8.
- Juan, A. A., Lourenc,o, H. R., Mateo, M., Luo, R., Castella, Q., 2014. Using iterated local search for solving the flow-shop problem: Parallelization, parametrization, and randomization issues. *International Transactions in Operational Research*, 21(1), 103–126. doi:10.1111/itor.12028
- Kang, Q., He, H., Wei, J. 2013. An effective iterated greedy algorithm for reliability-oriented task allocation in distributed computing systems. *Journal of parallel and distributed computing*, 73(8), 1106-1115.
- Karabulut, K. 2016. A hybrid iterated greedy algorithm for total tardiness minimization in permutation flowshops. *Computers & Industrial Engineering*, 98, 300-307.
- Kim, J. S., Park, J. H., Lee, D. H. 2017. Iterated greedy algorithms to minimize the total family flow time for job-shop scheduling with job families and sequence-dependent set-ups. *Engineering Optimization*, 49(10), 1719-1732.
- Lee, C. H., 2018. A dispatching rule and a random iterated greedy metaheuristic for identical parallel machine scheduling to minimize total tardiness. *International Journal of Production Research*, 56(6), 2292-2308.
- Li, X., Wang, Q., Wu, C. 2009. Efficient composite heuristics for total flowtime minimization in permutation flow shops. *OMEGA, The International Journal of Management Science*, 37, 155–164
- Li, W., Li, J., Gao, K., Han, Y., Niu, B., Liu, Z., Sun, Q. 2019. Solving robotic distributed flowshop problem using an improved iterated greedy algorithm. *International Journal of Advanced Robotic Systems*, 16(5), 1729881419879819.
- Lin, S. W., Lu, C. C., Ying, K. C. 2011. Minimization of total tardiness on unrelated parallel machines with sequence-and machine-dependent setup times under due date constraints. *The International Journal of Advanced Manufacturing Technology*, 53(1-4), 353-361.
- Lin, G. 2013. An iterative greedy algorithm for hardware/software partitioning. In *2013 ninth international conference on natural computation (ICNC)* (pp. 777-781). IEEE.

- Lin, S. W., Ying, K. C., Huang, C. Y. 2013. Minimising makespan in distributed permutation flowshops using a modified iterated greedy algorithm. *International Journal of Production Research*, 51(16), 5029-5038.
- Lin, S. W., Ying, K. C., Wu, W. J., Chiang, Y. I. 2016. Multi-objective unrelated parallel machine scheduling: a Tabu-enhanced iterated Pareto greedy algorithm. *International Journal of Production Research*, 54(4), 1110-1121.
- Mascis, A., Pacciarelli, D.: Job-shop scheduling with blocking and no-wait constraints. *Eur. J. Oper. Res.* 143(3), 498–517 (2002)
- Minella, G., R. Ruiz, and M. Ciavotta, “Restarted Iterated Pareto Greedy algorithm for multi-objective flowshop scheduling problems,” *Computers and Operations Research*, vol. 38, no. 11, pp. 1521–1533, 2011
- Naderi, B., Ruiz, R., 2010. The distributed permutation flowshop scheduling problem. *Comput. Oper. Res.* 37, 754–768. <https://doi.org/10.1016/j.cor.2009.06.019>
- Naderi, B., Rahmani, S., Rahmani, S. 2014. A multiobjective iterated greedy algorithm for truck scheduling in cross-dock problems. *Journal of Industrial Engineering*, 2014.
- Nawaz M, Ensore Jr. EE, Ham I. A heuristic algorithm for the m machine, n job flowshop sequencing problem. *Omega-International Journal of Management Science* 1983;11(1):91–5.
- Nouri, N., Ladhari, T. 2016. An Efficient Iterated Greedy Algorithm for the Makespan Blocking Flow Shop Scheduling Problem. *Polibits*, (53), 91-95.
- Nowicki, E., Smutnicki, C. 1996. A fast taboo search algorithm for the job shop problem. *Management science*, 42(6), 797-813.
- Pacciarelli, D. (2002) Alternative graph formulation for solving complex factory scheduling problems, *International Journal of Production Research*, 40:15, 3641-3653, DOI:10.1080/00207540210136478
- Palmer, D.S., 1965. Sequencing jobs through a multi-stage process in the minimum total time: a quick method of obtaining a near optimum. *Operational Research Quarterly*, 16 (1), 101–107.
- Pan, Q. K., Tasgetiren, M. F., Liang, Y. C. 2008. A discrete differential evolution algorithm for the permutation flowshop scheduling problem. *Computers & Industrial Engineering*, 55(4), 795-816.
- Pan, Q.K., R Ruiz. Local search methods for the flowshop scheduling problem with flowtime minimization. *European Journal of Operational Research*, 2012, 222: 31-43.
- Pan, Q. K., Ruiz, R. 2014. An effective iterated greedy algorithm for the mixed no-idle permutation flowshop scheduling problem. *Omega*, 44, 41-50.
- Pinheiro, J. C., & Arroyo, J. E. C. 2020. Effective IG heuristics for a single-machine scheduling problem with family setups and resource constraints. *Annals of Mathematics and Artificial Intelligence*, 88(1-3), 169-185.
- Pranzo, M., Pacciarelli, D. 2016. An iterated greedy metaheuristic for the blocking job shop scheduling problem. *Journal of Heuristics*, 22(4), 587-611.
- Rad SF, Ruiz R, Boroojerdian N. New high performing heuristics for minimizing makespan in permutation flowshops. *OMEGA, The International Journal of Management Science* 2009; 37(2): 331–345.
- Rajendran, C., Ziegler, H. 2004. Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research*, 155(2), 426-438.

- Ribas, I., Companys, R., Tort-Martorell, X. 2019. An iterated greedy algorithm for solving the total tardiness parallel blocking flow shop scheduling problem. *Expert Systems with Applications*, 121, 347-361.
- Riahi, V., Chiong, R., Zhang, Y. 2020. A new iterated greedy algorithm for no-idle permutation flowshop scheduling with the total tardiness criterion. *Computers & Operations Research*, 117, 104839.
- Rodriguez, F. J., Lozano, M., Blum, C., & García-Martínez, C. (2013). An iterated greedy algorithm for the large-scale unrelated parallel machines scheduling problem. *Computers & Operations Research*, 40(7), 1829-1841.
- Ruiz, R., and Stutzle, T. (2007), 'A Simple and Effective Iterated Greedy Algorithm for the Permutation Flowshop Scheduling Problem', *European Journal of Operational Research*, 177, 2033–2049.
- Shaw, P.: A new local search algorithm providing high quality solutions to vehicle routing problems. Department of Computer Sciences, University of Strathclyde, Glasgow, Scotland, Technical Report, APES group (1997).
- Taillard E. Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, 1990; 47(1): 65–74.
- Tasgetiren, M.F., Pan, Q.K., Liang, Y.C. 2009, 'A Discrete Differential Evolution Algorithm for the Single Machine Total Weighted Tardiness Problem with Sequence Dependent Setup Times', *Computers & Operations Research*, 36, 1900–1915.
- Tasgetiren, M. F., Pan, Q. K., Suganthan, P. N., Buyukdagli, O., 2013. A variable iterated greedy algorithm with differential evolution for the no-idle permutation flowshop scheduling problem. *Computers & Operations Research*, 40(7), 1729-1743.
- Urlings, T., Ruiz, R., Stützle, T. 2010. Shifting representation search for hybrid flexible flowline problems. *European Journal of Operational Research*, 207(2), 1086-1095.
- Yuan, Z., Fügenschuh, A., Homfeld, H., Balaprakash, P., Stützle, T., Schoch, M., 2008. Iterated greedy algorithms for a real-world cyclic train scheduling problem. In *International Workshop on Hybrid Metaheuristics* (pp. 102-116). Springer, Berlin, Heidelberg.
- Ying, K. C., S. W. Lin, and S. Y. Wan., 2014. "Bi-objective Reentrant Hybrid Flowshop Scheduling: An Iterated Pareto Greedy Algorithm." *International Journal of Production Research* 52 (19): 5735–5747.