KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

DOKTORA TEZİ

ZAMANA BAĞLI PROSES MALİYETİ ALTINDA TEK MAKİNA ÇİZELGELEME PROBLEMİNİN İKİ AMAÇLI OPTİMİZASYONUNA YÖNELİK BİR MODEL ÖNERİSİ

MUSTAFA TACETTİN

KOCAELİ 2014

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

DOKTORA TEZİ

ZAMANA BAĞLI PROSES MALİYETİ ALTINDA TEK MAKİNA ÇİZELGELEME PROBLEMİNİN İKİ AMAÇLI OPTİMİZASYONUNA YÖNELİK BİR MODEL ÖNERİSİ

MUSTAFA TACETTİN

Prof.Dr. Alpaslan FIĞLALI
Danışman, Kocaeli Üniv. .....

Prof.Dr. Zerrin ALADAĞ
Jüri Üyesi, Kocaeli Üniv. .....

Doç.Dr. Tonguç ÜNLÜYURT
Jüri Üyesi, Sabancı Üniv. .....

Doç.Dr. Ayhan DEMİRİZ
Jüri Üyesi, Sakarya Üniv. .....

Yrd.Doç.Dr. Yıldız YULUĞKURAL
Jüri Üyesi, Kocaeli Üniv. .....

Tezin Savunulduğu Tarih: 10.02.2014

ÖNSÖZ ve TEŞEKKÜR

Globalleşen dünyada bilgi paylaşımının artmasıyla beraber en iyiyi hedefleyen yöntemler firmalar tarafından kullanılmaya başladıkça, fark yaratmak için daha orijinal fikirlere ihtiyaç duyulmakta ve birden fazla amacı bir arada yapmayı hedefleyen yöntemler ortaya koymak gerekmektedir. Bu tez çalışmasında Türkiye' deki elektrik fiyatlarının farklı zamanlarda farklı birim fiyat şeklinde uygulaması, kurumlar açısından nasıl maliyet avantajına dönüşür fikri ile yola çıkılarak, az elektrik tüketen ürünleri elektriğin pahalı olduğu zamanda üretip, çok elektrik çekenleri elektriğin ucuz olduğu zamanda üretilmesine yönelik bir model önerilmiştir. Bununla beraber, gözetilmesi gereken önemli bir husus, elektrik maliyetini azaltırken ürünlerin termin kısıtını da gözetmek ve müşterilerin taleplerini geciktirmeden üretimi devam ettirmektir. Bu iki amaç ilk olarak ayrı ayrı ele alınmış, sonrasında ise bu amaçlar birleştirilerek kullanıcıya baskın çözüm kümesi sunulmuş ve istenilen alternatifi seçmesi önerilmiştir.

Bana doktoraya başlama fırsatı verdiği için o zamanki yöneticim Fatih Tunçbilek'e, tezin oluşması sırasında beni yönlendiren ve desteğini esirgemeyen değerli hocam Prof. Dr. Alpaslan Fığlalı'ya ve bu tezi yazarken evde sabrının sınırlarını zorladığım eşim Şeyda Tacettin'e teşekkür ederim. Aynı zamanda bugüne kadar beni yetiştiren anne, babama ve tüm öğretmenlerime şükranlarımı sunarım.

Şubat - 2014 Mustafa TACETTİN

İÇİNDEKİLER

ÖNSÖZ ve TEŞEKKÜR i
İÇİNDEKİLER ii
ŞEKİLLER DİZİNİ iv
TABLOLAR DİZİNİ ix
SİMGELER DİZİNİ VE KISALTMALAR xi
ÖZET xiii
ABSTRACT xiv
GİRİŞ 1
1. ÇİZELGELEME PROBLEMLERİ 5
1.1. Çizelgeme Problemlerinin Sınıflandırılması 5
1.2. Çizelgeleme Problemleri için Komplekslik Hiyerarşisi 7
2. ENERJİ MALİYETİNİN MİNİMİZASYONU 10
2.1. GAP (Genel Atama Problemi) 12
2.1. Özel Sıra Kümeli Genel Atama Problemi (GAPS2) 15
2.2. Önerilen Yöntem 16
2.2.1. Varsayımlar 16
2.2.2. Problemin kompleksliği 16
2.2.3. Önerilen model 17
2.2.4. Model 2'nin gevşetilmesi 20
2.2.5. Matematiksel model 3 21
2.2.6. Model 4 23
2.2.7. Model 3'ün çözüm için kullanılması 24
2.2.8. Problemlerin oluşturulması 25
2.2.9. Sabitlenecek işlerin belirlenmesi 25
2.3. Problemlerin Sonuçları 26
3. TEK MAKİNA TOPLAM GECİKME ZAMANININ ENAZLANMASI ÇİZELGELEME PROBLEMİ 30
3.1. Çözüm Yöntemleri 31
3.1.1. Dinamik programlama ile çözüm yöntemi 31
3.1.2. Karmaşık tamsayılı programlama ile çözüm yöntemleri 34
3.1.2.1. Zaman endeksli formülasyon yöntemi (TIS) 34
3.1.2.2. Lineer sıralama yöntemi (LO) 34
3.1.2.3. Pozisyon formülasyonu yöntemi (POS) 35
3.1.2.4. Aralık endeksli formülasyon yöntemi (IIF) 36
3.1.3. Yaklaşık sonuçlu çözüm yöntemleri 36
3.1.3.1. Değiştirilmiş Lawler algoritması 36
3.1.3.2. Tam polinom zamanlı yaklaşık düzen yöntemi 37
3.1.4. Metasezgisel çözüm yöntemleri 37
3.2. Önerilen Yöntem 38
3.2.1. Atama problemi 39
3.2.2. Önerilen model 40
3.2.3. Problemlerin belirlenen yöntem ile çözümü 41
3.3. Sonuçların Değerlendirilmesi 52

4. AMAÇLARIN BİRLEŞTİRİLMESİ 53
4.1. Genetik Algoritma Yaklaşımı 54
4.1.1. EOX (Emmons kuralına dayalı sıralı çaprazlama) yöntemi 57
4.2. Deney Tasarımı 59
4.3. Önerilen Yöntem 62
4.4. Problemlerin Sonuçları 65
5. SONUÇLAR ve ÖNERİLER 133
KAYNAKLAR 138
EKLER 143
KİŞİSEL YAYIN VE ESERLER 161
ÖZGEÇMİŞ 162

ŞEKİLLER DİZİNİ

Şekil 1.1. Çizelgeleme problemlerinin gösterimi 5
Şekil 1.2. Komplekslik hiyerarşisi 8
Şekil 2.1. Problemin sırt çantası problemine benzetimi 19
Şekil 3.1. Problemlerin standart sapma değerleriyle iyi sonuç verdiği yöntem ilişkisi 46
Şekil 3.2. EDD başarı oranı ile varyasyon katsayısı ilişkisi 49
Şekil 4.1. Faktör ve sayılarına göre ortogonal seri eşleme tablosu 61
Şekil 4.2. e_60_1 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 67
Şekil 4.3. e_60_1 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 67
Şekil 4.4. e_60_1 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 68
Şekil 4.5. e_60_1 problemi için çaprazlama operatörlerinin kıyaslaması 68
Şekil 4.6. e_60_2 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 69
Şekil 4.7. e_60_2 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 70
Şekil 4.8. e_60_2 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 70
Şekil 4.9. e_60_2 problemi için çaprazlama operatörlerinin kıyaslaması 71
Şekil 4.10. e_60_3 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 72
Şekil 4.11. e_60_3 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 72
Şekil 4.12. e_60_3 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 72
Şekil 4.13. e_60_3 problemi için çaprazlama operatörlerinin kıyaslaması 73
Şekil 4.14. e_60_4 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 74
Şekil 4.15. e_60_4 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 74
Şekil 4.16. e_60_4 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 74
Şekil 4.17. e_60_4 problemi için çaprazlama operatörlerinin kıyaslaması 75
Şekil 4.18. e_60_5 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 76
Şekil 4.19. e_60_5 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 76
Şekil 4.20. e_60_5 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 76
Şekil 4.21. e_60_5 problemi için çaprazlama operatörlerinin kıyaslaması 77

Şekil 4.22. e_60_6 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 78
Şekil 4.23. e_60_6 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 78
Şekil 4.24. e_60_6 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 78
Şekil 4.25. e_60_6 problemi için çaprazlama operatörlerinin kıyaslaması 79
Şekil 4.26. e_60_7 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 80
Şekil 4.27. e_60_7 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 80
Şekil 4.28. e_60_7 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 80
Şekil 4.29. e_60_7 problemi için çaprazlama operatörlerinin kıyaslaması 81
Şekil 4.30. e_60_8 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 82
Şekil 4.31. e_60_8 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 82
Şekil 4.32. e_60_8 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 82
Şekil 4.33. e_60_8 problemi için çaprazlama operatörlerinin kıyaslaması 83
Şekil 4.34. e_60_9 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 84
Şekil 4.35. e_60_9 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 84
Şekil 4.36. e_60_9 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 84
Şekil 4.37. e_60_9 problemi için çaprazlama operatörlerinin kıyaslaması 85
Şekil 4.38. e_60_10 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 86
Şekil 4.39. e_60_10 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 86
Şekil 4.40. e_60_10 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 86
Şekil 4.41. e_60_10 problemi için çaprazlama operatörlerinin kıyaslaması 87
Şekil 4.42. e_90_1 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 89
Şekil 4.43. e_90_1 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 90
Şekil 4.44. e_90_1 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 90
Şekil 4.45. e_90_1 problemi için çaprazlama operatörlerinin kıyaslaması 91
Şekil 4.46. e_90_2 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 92
Şekil 4.47. e_90_2 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 92
Şekil 4.48. e_90_2 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 92
Şekil 4.49. e_90_2 problemi için çaprazlama operatörlerinin kıyaslaması 93

Şekil 4.50. e_90_3 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 94
Şekil 4.51. e_90_3 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 94
Şekil 4.52. e_90_3 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 94
Şekil 4.53. e_90_3 problemi için çaprazlama operatörlerinin kıyaslaması 95
Şekil 4.54. e_90_4 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 96
Şekil 4.55. e_90_4 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 96
Şekil 4.56. e_90_4 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 96
Şekil 4.57. e_90_4 problemi için çaprazlama operatörlerinin kıyaslaması 97
Şekil 4.58. e_90_5 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 98
Şekil 4.59. e_90_5 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 98
Şekil 4.60. e_90_5 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 98
Şekil 4.61. e_90_5 problemi için çaprazlama operatörlerinin kıyaslaması 99
Şekil 4.62. e_90_6 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 100
Şekil 4.63. e_90_6 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 100
Şekil 4.64. e_90_6 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 100
Şekil 4.65. e_90_6 problemi için çaprazlama operatörlerinin kıyaslaması 101
Şekil 4.66. e_90_7 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 102
Şekil 4.67. e_90_7 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 102
Şekil 4.68. e_90_7 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 102
Şekil 4.69. e_90_7 problemi için çaprazlama operatörlerinin kıyaslaması 103
Şekil 4.70. e_90_8 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 104
Şekil 4.71. e_90_8 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 104
Şekil 4.72. e_90_8 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 104
Şekil 4.73. e_90_8 problemi için çaprazlama operatörlerinin kıyaslaması 105
Şekil 4.74. e_90_9 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 106
Şekil 4.75. e_90_9 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 106
Şekil 4.76. e_90_9 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 106
Şekil 4.77. e_90_9 problemi için çaprazlama operatörlerinin kıyaslaması 107

Şekil 4.78. e_90_10 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 108
Şekil 4.79. e_90_10 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 108
Şekil 4.80. e_90_10 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 108
Şekil 4.81. e_90_10 problemi için çaprazlama operatörlerinin kıyaslaması 109
Şekil 4.82. e_120_1 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 111
Şekil 4.83. e_120_1 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 112
Şekil 4.84. e_120_1 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 112
Şekil 4.85. e_120_1 problemi için çaprazlama operatörlerinin kıyaslaması 113
Şekil 4.86. e_120_2 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 114
Şekil 4.87. e_120_2 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 114
Şekil 4.88. e_120_2 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 114
Şekil 4.89. e_120_2 problemi için çaprazlama operatörlerinin kıyaslaması 115
Şekil 4.90. e_120_3 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 116
Şekil 4.91. e_120_3 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 116
Şekil 4.92. e_120_3 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 116
Şekil 4.93. e_120_3 problemi için çaprazlama operatörlerinin kıyaslaması 117
Şekil 4.94. e_120_4 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 118
Şekil 4.95. e_120_4 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 118
Şekil 4.96. e_120_4 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 118
Şekil 4.97. e_120_4 problemi için çaprazlama operatörlerinin kıyaslaması 119
Şekil 4.98. e_120_5 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 120
Şekil 4.99. e_120_5 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 120
Şekil 4.100. e_120_5 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 120
Şekil 4.101. e_120_5 problemi için çaprazlama operatörlerinin kıyaslaması 121
Şekil 4.102. e_120_6 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 122
Şekil 4.103. e_120_6 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 122
Şekil 4.104. e_120_6 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 122
Şekil 4.105. e_120_6 problemi için çaprazlama operatörlerinin kıyaslaması 123

Şekil 4.106. e_120_7 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 124
Şekil 4.107. e_120_7 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 124
Şekil 4.108. e_120_7 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 124
Şekil 4.109. e_120_7 problemi için çaprazlama operatörlerinin kıyaslaması 125
Şekil 4.110. e_120_8 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 126
Şekil 4.111. e_120_8 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 126
Şekil 4.112. e_120_8 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 126
Şekil 4.113. e_120_8 problemi için çaprazlama operatörlerinin kıyaslaması 127
Şekil 4.114. e_120_9 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 128
Şekil 4.115. e_120_9 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 128
Şekil 4.116. e_120_9 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 128
Şekil 4.117. e_120_9 problemi için çaprazlama operatörlerinin kıyaslaması 129
Şekil 4.118. e_120_10 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 130
Şekil 4.119. e_120_10 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 130
Şekil 4.120. e_120_10 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi 130
Şekil 4.121. e_120_10 problemi için çaprazlama operatörlerinin kıyaslaması 131
Şekil 5.1. e_90_8 problemi için α değeri 5 arttırıldığında OX çaprazlama operatörü ile elde edilen çözüm kümesi 135
Şekil 5.2. e_90_6 problemi için değiştirilmiş uygunluk fonksiyonu değeriyle orijinal uygunluk fonksiyonu değerinin karşılaştırılması 136

TABLOLAR DİZİNİ

Tablo 2.1. X_{ij}, $X_{i\ j-1}$ ve Y_{ij}'nin alabileceği olası değerler 23
Tablo 2.2. Problemler ve çözüm süreleri 28
Tablo 3.1. Problemlerin toplam gecikme zamanı için önerilen yöntemle çözümü 43
Tablo 3.2. Gecikme faktörü 0,58 olan problemlerin önerilen yöntemle çözümü 44
Tablo 3.3. Problemlerin EDD ile kıyaslanması 45
Tablo 3.4. Standart sapma değeri değiştirilmiş problemler için EDD ile önerilen yöntemin kıyaslanması 48
Tablo 3.5. 30 işlik problemlerin optimum çözümleriyle önerilen yöntemin kıyaslanması 51
Tablo 4.1. L9 ortogonal dizisi 62
Tablo 4.2. Deney tasarımı için kullanılacak faktörler ve seviyeleri 64
Tablo 4.3. L9'a göre yapılacak deneyler 64
Tablo 4.4. Yapılan deneyler için S/G oranları 65
Tablo 4.5. Her bir faktörün her bir seviyesi için S/G oranları 65
Tablo 4.6. e_60_1 problemi için baskın çözüm kümesi 66
Tablo 4.7. e_60_2 problemi için baskın çözüm kümesi 69
Tablo 4.8. e_60_3 problemi için baskın çözüm kümesi 71
Tablo 4.9. e_60_4 problemi için baskın çözüm kümesi 73
Tablo 4.10. e_60_5 problemi için baskın çözüm kümesi 75
Tablo 4.11. e_60_6 problemi için baskın çözüm kümesi 77
Tablo 4.12. e_60_7 problemi için baskın çözüm kümesi 79
Tablo 4.13. e_60_8 problemi için baskın çözüm kümesi 81
Tablo 4.14. e_60_9 problemi için baskın çözüm kümesi 83
Tablo 4.15. e_60_10 problemi için baskın çözüm kümesi 85
Tablo 4.16. 60 işlik problemlerin en iyi enerji maliyet değerleri 88
Tablo 4.17. 60 işlik problemlerin en iyi toplam gecikme zamanı değerleri 88
Tablo 4.18. e_90_1 problemi için baskın çözüm kümesi 89
Tablo 4.19. e_90_2 problemi için baskın çözüm kümesi 91
Tablo 4.20. e_90_3 problemi için baskın çözüm kümesi 93
Tablo 4.21. e_90_4 problemi için baskın çözüm kümesi 95
Tablo 4.22. e_90_5 problemi için baskın çözüm kümesi 97
Tablo 4.23. e_90_6 problemi için baskın çözüm kümesi 99
Tablo 4.24. e_90_7 problemi için baskın çözüm kümesi 101
Tablo 4.25. e_90_8 problemi için baskın çözüm kümesi 103
Tablo 4.26. e_90_9 problemi için baskın çözüm kümesi 105
Tablo 4.27. e_90_10 problemi için baskın çözüm kümesi 107
Tablo 4.28. 90 işlik problemlerin en iyi enerji maliyet değerleri 110
Tablo 4.29. 90 işlik problemlerin en iyi toplam gecikme zamanı değerleri 110
Tablo 4.30. e_120_1 problemi için baskın çözüm kümesi 111
Tablo 4.31. e_120_2 problemi için baskın çözüm kümesi 113
Tablo 4.32. e_120_3 problemi için baskın çözüm kümesi 115

Tablo 4.33. e_120_4 problemi için baskın çözüm kümesi..... 117
Tablo 4.34. e_120_5 problemi için baskın çözüm kümesi..... 119
Tablo 4.35. e_120_6 problemi için baskın çözüm kümesi..... 121
Tablo 4.36. e_120_7 problemi için baskın çözüm kümesi..... 123
Tablo 4.37. e_120_8 problemi için baskın çözüm kümesi..... 125
Tablo 4.38. e_120_9 problemi için baskın çözüm kümesi..... 127
Tablo 4.39. e_120_10 problemi için baskın çözüm kümesi 129
Tablo 4.40. 120 işlik problemlerin enerji maliyeti amacına göre değerlendirilmesi..... 132
Tablo 4.41. 120 işlik problemlerin toplam gecikme amacına göre değerlendirilmesi..... 132
Tablo 5.1. e_90_8 problemi için α 5 arttırıldığında oluşan baskın çözüm kümesi 134
Tablo 5.2. Önerilen yöntemler ve genetik algoritma kıyaslaması 137

SİMGELER DİZİNİ VE KISALTMALAR

C_{max} : Maksimum tamamlama süresi (dk)
FF_s : s aşamalı esnek akış tipi çizelgeleme problemi
F_m : Akış tipi m makine çizelgeleme problemi
H_k : k periyodu için sabitlenen işin işlem zamanı
J_m : m makine atölye tipi çizelgeleme problemi
L_{max} : Maksimum pozitif geç kalma süresi (dk)
O_m : m makine atölye tipi çizelgeleme problemi
p_{ij} : j işinin i makinesindeki işlem zamanı (dk)
P_m : Paralel özdeş m makine çizelgeleme problemi
Q_m : m makine uniform paralel çizelgeleme problemi
r_j : j işinin hazır olma zamanı
R_m : Paralel özdeş olmayan m makine çizelgeleme problemi
s_{jk} : j işi ile k işi arası sıra bağımlı ayar zamanı
s_k : k işinin ayar zamanı
W_jC_j : Ağırlıklı tamamlama zamanı
W_jT_j : Ağırlıklı pozitif gecikme değeri
W_jU_j : Ağırlıklandırılmış gecikmiş iş sayısı
Z : Amaç fonksiyon değeri
λ_{ij} : Lagrange çarpanı
τ : Gecikme faktörü (Problemin zorluk derecesi)

Kısaltmalar

ANOVA : Varyans Analizi yöntemi
ATC : Açık Gecikme Maliyeti Yöntemi
batch(b) : Toplu İşleme
block : Bloklama
brkdwn : Arızalar
CX : Dairesel Çaprazlama
DGA : Difransiyel Gelişim Algoritması
EDD : En erken Termin Zamanlı En Önce Kuralı
EMOX : Enerji Maliyetine Göre Sıralı Çaprazlama
EOX : Emmons Kuralına Bağlı Sıralı Çaprazlama
ERD : En Erken Hazır Olma Zamanlı En Önce Kuralı
fmls : İş Aileleri
GAP : Genel Atama Problemi
GAPS2 : Özel Sıra Kümeli Genel Atama Problemi
IIF : Aralık Endeksli Formülasyon Yöntemi
KwH : Kilowatt Saat
LO : Doğrusal Sıralama Yöntemi
LOX : Doğrusal Sıralı Çaprazlama
MDD : Değiştirilmiş Termin Tarihlerine Göre Sıralama Yöntemi
MIP : Karmaşık Tamsayılı Matematiksel Programlama

M_j	: j Makinesi İçin Erişilebilirlik Kısıtı
MS	: Minimum Sapmaya göre Çizelgeleme Kuralı
NBR	: Net Kazanç Değişimi Sezgiseli
NP	: Polinom Zamanlı Olmayan
nwt	: Beklemesiz
OPT	: Optimum
OX	: Sıralı Çaprazlama
ÖSK	: Özel Sıra Kümeli
PMX	: Kısmi Planlı Çaprazlama
POS	: Pozisyon Formülasyon Yöntemi
prec	: Öncül Ardıl İlişkisi
prmp	: Bölünebilme
prmu	: Permütasyon
PSK	: Panwalkar, Smith, Koulamas Tek Makine Toplam Gecikme Zamanı Minimizasyonu Algoritması
recr	: Yeniden Dolaşım
S/G	: Sinyal/Gürültü Oranı
SPT	: En Kısa İşlem Süreli En Önce Kuralı
SXX	: Alt Değişimli Çaprazlama
TIS	: Zaman Endeksli Formülasyon Yöntemi
TL	: Türk Lirası
TMEMM	: Tek Makine Enerji Maliyeti Minimizasyonu Yöntemi
TMTGZM	: Tek Makine Toplam Gecikme Zamanı Minimizasyonu Yöntemi
TOU	: Zamana Bağlı Kullanım
WSPT	: Ağırlıklı En Kısa Süreli İşi En Önce Çizelgele Kuralı

ZAMANA BAĞLI PROSES MALİYETİ ALTINDA TEK MAKİNA ÇİZELGELEME PROBLEMİNİN İKİ AMAÇLI OPTİMİZASYONUNA YÖNELİK BİR MODEL ÖNERİSİ

ÖZET

Ülkemizde uygulanmakta olan zamana bağlı enerji fiyatlandırması nedeniyle, eğer üretilen ürünün birim zamanda enerji tüketimi farklılık gösteriyorsa, toplam enerji maliyeti aynı zamanda makinalara atanmış işlerin sıralamasına bağlıdır denilebilir.

Diğer bir konu ise, işler yapılırken, bir sonraki iç müşteriye ya da nihai müşteriye istenilen zamanda yetiştirilmesi gerekir. Maliyetler iyileştirilirken müşteri memnuniyeti göz ardı edilmemelidir. Yukarıda bahsedilen enerji maliyetlerini azaltmaya yönelik yapılacak işlerin sıra değişikliği, aynı zamanda bu işlerin asıl yapılma amacı olan müşterilerin kullanımı konusunda herhangi bir sıkıntıya yol açmamalıdır.

Bu çalışmada, zamana bağlı elektrik tarifesine göre enerji maliyeti anlamında tek makine problemi incelenmiş ve bir lastik fabrikası üretim ortamına uygun olarak oluşturulmuş toplam 60, 90 ve 120 işten oluşan 30 problem için enerji maliyetini azaltmaya yönelik karmaşık tamsayılı bir model önerilmiştir. Bu modelin çözülmesine yönelik olarak lagrange gevşetme ve kesme düzlemi teknikleri kullanılmış ve makul sürelerde çözümler elde edilebilmiştir.

Aynı zamanda, tek makine çizelge problemi için toplam gecikme zamanının enazlanması problemi, oluşturulan problemler için ele alınmış ve işlem zamanlarının birbirine yakın olduğu problem tipleri için bir model önerilmiştir.

Son olarak ise, her iki amacı tek bir amaç fonksiyonu haline dönüştürerek genetik algoritma kullanılarak pareto optimal çözümler oluşturulmuş ve her bir problem için karar vericiye farklı baskın çözüm alternatifleri sunularak bunlardan tercih yapabilmesine yönelik bir model önerilmiştir.

Anahtar Kelimeler: Çok Amaçlı Genetik Algoritma, Genel Atama Problemi, Lagrange Gevşetimi, Tek Makina Çizelgeleme, Zamana Bağlı Elektrik Tarifesi.

A MODEL PROPOSAL FOR OPTIMIZING TWO OBJECTIVES FOR THE SINGLE MACHINE SCHEDULING PROBLEM WITH TIME DEPENDENT PROCESS COST

ABSTRACT

In our country, because of using time of usage (TOU) tariff, if unit energy consumption of products differs, total electricity cost also depends on to the sequence of jobs assigned to the machines.

While doing the jobs, an important criteria is providing these products to next customer or final customer on time. While decreasing costs, customer satisfaction should not be ignored. Changing the sequence of jobs in order to decrease electricity cost should not create a problem related with customer which is main reason of doing those jobs.

In this study, single machine scheduling problem with TOU tariff is investigated and a mixed integer programming model is proposed for total 30 problems with 60, 90 and 120 jobs which are generated according to a tyre manufacturing environment. This model is solved in reasonable time by using lagrange relaxation and cutting plane techniques.

On the other hand, total tardiness minimization problem in single machine scheduling environment is considered. A new model is proposed for this problem where process times of jobs are similar to each other.

Finally, these two objectives are combined in a single objective function and pareto optimal solutions are achieved by using genetic algorithm approach. For each problem, dominant solution set is provided to decision makers to select one of them.

Keywords: Multiobjective Genetic Algorithm, Generalized Assignment Problem, Lagrange Relaxation, Single Machine Scheduling, Tou Tariff.

GİRİŞ

Çizelgeleme süreci, üretim faaliyetlerinin gerçekleştirilmesinde, hangi işin hangi miktarda hangi makinada ne kadar yapılacağına karar veren bir mekanizma olduğu için çok önemlidir. Aslında bir işletmede, üretim sahasında kimin ne iş yapacağına çizelgeleme probleminin çözümü karar vermektedir. Bu işlerin hangi makinada, ne miktarda ve ne sırada yapıldığı üretim maliyetlerini önemli ölçüde etkileyebilir. Örneğin, aynı iş farklı makinalarda farklı işçilik maliyetiyle ya da sürede yapılıyorsa, işin hangi makinada çizelgelendiği maliyet açısından önemlidir. Benzer şekilde, eğer parti büyüklüğü arttıkça fire oranı artıyorsa, işin miktarının belirlenmesi de maliyet anlamında önemli bir etken olabilir. Son olarak, işlerin hangi sırada yapıldığı, eğer sıra bağımlı ayar zamanı söz konusu ise, toplam süreyi etkileyeceğinden maliyet anlamında bir anlam ifade eder.

Zamana bağlı işlem ya da işletme maliyeti altında çizelgeleme problemi, çoklu ürün gamına sahip bir üretim tesisinde, bu değişik ürünlerin farklı zaman dilimlerinde, farklı maliyetlere sahip olması şeklinde tanımlanabilir. Amaç, işlerin sıralamasını değiştirerek, maliyeti en aza indirmektir. Örneğin, ürünlerin üretim esnasında tükettikleri elektrik miktarları farklıysa, elektrik birim fiyatı zamana bağlı değişen bir tarife için, ürünlerin enerji maliyetleri zamana bağlı değişir. Yine üretim esnasında gereken işçi sayısı ürüne bağlı ise ve vardiyalardaki ücretlendirme farklı ise, farklı ürünlerin farklı zaman diliminde farklı işletme maliyetine sahip olduğu söylenebilir.

Bu tez çalışmasında, elektrik fiyatlandırması üzerine odaklanılacaktır.

Gelişen birçok ülkede, talep profilinden dolayı, elektrik üreten şirketler TOU (Time of usage) tarifesi denilen bir tarife uygulamaktadır. TOU tarifesi farklı zaman diliminde farklı elektrik fiyatlandırması şeklindedir ve amacı sistemin toplam yük eğrisini düzleştirmektir. TOU tarifesi özellikle 2000 ve 2001 yıllarında Amerika Kaliforniya'da yaşanan santral krizlerinin ardından daha fazla ilgi çekmiştir. Kaliforniya'da uygulanmaya başlanan TOU tarifesi sayesinde 2003 Temmuz'dan 2004 Aralık ayına kadar elektriğin en fazla tüketildiği periyodda tüketim % 7,6 ile

%25 oranında azalmıştır (Faruqui ve George, 2005). Yine benzer şekilde Norveç'de TOU tarifesinin uygulanmasıyla beraber elektriğin en fazla tüketildiği periyotta tüketim %8-9 oranında azalmıştır (Faruqui ve Sergici, 2010). Havalandırma sistemlerinin tükettiği enerji miktarının yine TOU tarifesine uygun olarak yük dengelemesi yapıldığında pik periyodundaki tüketimin % 38 oranında azaltılabileceği gösterilmiştir (Ashok ve Banerjee, 2003). %38'lik azalmanın operasyonel maliyet anlamında karşılığı %5,9 olarak hesaplanmıştır. Bazı endüstriler, elektrik maliyetlerini kontrol altına almak için çizelgelerini TOU tarifesine göre yapmaktadır. Örneğin, çelik imalatında, Ashok tarafından karmaşık tam sayılı matematiksel bir model önerilmiştir. Bu model lotların belirlenen bir periyot içinde hangi makinada hangi sırada üretileceğini belirler. Ancak, önerilen model her bir zaman dilimi için bir ikili (binary) değişken önermiştir. Böyle bir model, iş sayısının çok olduğu, zaman diliminin dakika olarak belirlendiği durumlarda değişken sayısı çok fazla olacağı için uygun bir zamanda çözüm veremez. Farklı bir çalışma olarak, Lee ve Chen optimum kontrat kapasite seçimi ile ilgili bir çalışma yapmıştır. Burada önerilen modelde toplam elektrik maliyetini eniyilemek için kontrat sözleşme maliyeti ve kontratta belirtilen kapasiteyi aşma durumunda ödenen ceza maliyetleri ele alınmıştır. TOU tarifesini belirlerken, üreticiler talebin yüksek olduğu yerlerde fiyatı yüksek tutmakta, düşük olduğu yerlerde ise ucuz tutmak istemektedirler. Üretici açısından elektriğin pik noktasının azaltılması toplam enerji üretim maliyetini azaltmakta ve yüksek tüketim zamanındaki talebi karşılamak için yapılması gereken yatırım maliyetinin önüne geçmektedir. Yang ve diğ. tarafından yapılan çalışmada TOU tarifesi için periyotların başlama bitiş zamanlarının belirlenmesi ve bu periyodlardaki fiyatların tespit edilmesi ile ilgili bir yöntem önerilmiş ve böylelikle TOU tarifesi kullanılarak sabit fiyat uygulamasına göre kıyaslandığında hem tüketici için hem de üretici için maliyet avantajı sağlanabileceği gösterilmiştir. TOU tarifesine üretici açısından bakıldığında parçacık sürü optimizasyonu yöntemi kullanılarak santraller için yük dengelemesi yapıldığında oluşacak maliyet avantajının %30 civarında olabileceği söylenebilir (Tang ve diğ., 2014).

Türkiye'de uygulanan TOU tarifesine göre, gün 3 parçaya bölünmüştür. Sabah saat 06:00'dan 17:00'ye kadar olan kısma gün periyodu, 17:00'den 22:00'e kadar olan

kısma prime-time periyodu, kalan zamana ise gece periyodu denilmektedir. 2008 yılı fiyatları baz alındığında, gece periyodunda ücretlendirme 1 TL/kwH iken, prime-time için ücret 3,44 TL/kwH, gün periyodu içinse 2 TL/kwH olabilmektedir. Fiyatların periyotlar arası bu kadar farklı olmasından dolayı işlerin toplam elektrik maliyetini azaltacak şekilde sıralanması oldukça önem kazanmaktadır.

Tez çalışmasına esas olacak veriler bir lastik fabrikası baz alınarak üretilmiştir. Lastik üretiminde, ilk olarak mikserlerde hazırlanan karışımlar, daha sonrasında lastiği oluşturan bileşenleri hazırlamak üzere şekillendirilir. Bu şekillendirme ve karıştırma işlemleri çok fazla miktarda elektrik tüketen işlerdir ve tüketilen enerji miktarı karışım özelliklerine göre (sertlik, akışkanlık) değişir. Bu bileşenlerin hazırlandığı ekstruderlerde birbirinden farklı lastik bileşenleri (sırt, yanak, dolgu v.b.) üretilir. Bir bileşen sadece bir makinede tanımlıdır ve çizelgelemeye başlamadan önce hangi bileşenin hangi ekstruderde üretileceği bellidir. Kapasite kısıtından dolayı, bu makinelerde boşluk bırakılmaması hedeflenmiştir. Kurulacak olan modelde, işler arası boş zaman bırakılmasına izin verilmeyecektir. Dolayısıyla, üretilecek olan çizelgenin olurlu çizelge sınıfına girmesi gerekir. Üretilecek olan ürünün hangi makinada olması gerektiği bilindiği için, makinalar birbirleriyle etkileşimleri yokmuş gibi düşünülebilir ve her bir ekstruderin çizelgelenmesi tek makine çizelgeleme problemi olarak ele alınabilir.

Ele alınan problem için zaman penceresine bağlı olarak elektrik fiyatları değiştiğinden, zaman penceresine bağlı tek makina çizelgeleme problemi olarak düşünülebilir. Zaman pencerelerine bağlı olarak tek makina çizelgeleme problemi ile ilgili yapılan çalışmalarda, zamana bağlı işlem zamanı ya da ortak termin penceresi içeren problemler ele alınmıştır. Tek makina çizelgeleme probleminde zaman pencerelerine bağlı olarak işlem zamanı Lahlou ve Peres tarafından çalışılmıştır. Biskup ve Feldman tek makina çizelgeleme problemlerinde ortak termin penceresi problemi ile ilgili matematiksel model tabanlı çözüm önerisi getirmişlerdir.

Bu tezde önerilen yöntemde ise, öncelikle yukarıda tarif edilen problem belirlenen iki ayrı amaç için ayrı ayrı tek amaçlı problemler olarak çözülmüş ve daha sonrasında iki farklı amaç için farklı ağırlıklandırmalar kullanılarak pareto optimum çözüm kümesinin bulunması amaçlanmıştır. Bunun için çok amaçlı genetik algoritma

kullanılmıştır. Ele alınan çizelgenin toplam gecikme zamanı ve toplam elektrik maliyeti değerleri, bu amaçlar tek olarak ele alındığında elde edilen en iyi değere bölünerek normalize edilmiş, farklı ağırlıklandırma katsayıları kullanılarak pareto optimal çözümler elde edilmiştir.

Bu tez çalışması üç aşamadan oluşmaktadır. Bu aşamalar;

1- Problemin amaçlarından biri olan maliyetlerin en azlanması problemi için bir yöntem ve çözüm önerisi getirilmesi,

2- Problemin amaçlarından bir başkası olan toplam gecikmenin en azlanması problemi için yöntem ve çözüm önerisi getirilmesi,

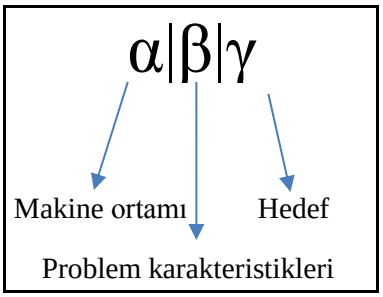
3- İlk iki aşamada elde edilen sonuçlara göre genetik algoritma kullanarak arzulanan baskın çözümlerin elde edilmesidir.

1. ÇİZELGELEME PROBLEMLERİ

Çizelgeleme problemlerinin çözümü işlerin sadece hangi makinada ve sırada üretileceği bilgisini değil, aynı zamanda hangi zamanda başlayacağı ve biteceği bilgisini de içerir. Çizelgeleme problemleri çok farklı şekilde karşımıza çıkabilir. Örneğin, havaalanlarında uçakların en az süre beklemesini amaçlayan iniş kalkış çizelgesinin oluşturulması, hastanelerdeki ameliyathaneler için hastaların hastalık derecesine göre öncelik verilerek çizelgelenmesi, bilgisayar işlemcilerinin çizelgelenmesi olarak karşımıza çıkabilir. Bu tez için çalışılan çizelgeleme probleminde, deterministik, rassal olmayan bir ortam öngörülmüştür. Bir makinada aynı anda sadece bir operasyon gerçekleştirilebilir. İşler başladıktan sonra iptal edilemez. Tüm işlerin hazır olma zamanı sıfır anıdır ve stok tutma ile ilgili herhangi bir sınırlandırma getirilmemiştir.

1.1. Çizelgeme Problemlerinin Sınıflandırılması

Çizelgeleme problemleri 3 parametre ile ifade edilir ve bu şekilde sınıflandırılır. Bu parametreler, makina ortamı, kısıtlar ya da probleme ait karakteristik özellikler ve son olarak optimize edilmesi gereken hedeftir. Böylelikle herhangi bir çizelgeleme problemi Şekil 1.1'deki gibi ifade edilebilir.



Şekil 1.1. Çizelgeleme problemlerinin gösterimi

Bu gösterim şeklinde α değeri makina ortamını, β değeri probleme ait karakteristikleri, γ değeri ise hedefi belirtir. Makina ortamı (α) ile ilgili alternatifler aşağıda sıralanmıştır.

1. Tek makine çizelgeleme problemleri, α değeri 1 ile gösterilir.
2. Paralel özdeş makine çizelgeleme problemleri, α değeri P_m ile gösterilir. m makine sayısını ifade eder.
3. Uniform paralel makine çizelgeleme problemleri, α değeri Q_m ile gösterilir. m makine sayısını ifade eder.

$p_{ij} = p_j/v_i$'dir. Yani, j işinin i makinasında yapılma süresi i makinasının hızına bağlıdır.

4. Paralel özdeş olmayan makine çizelgeleme problemleri, α değeri R_m ile gösterilir. m makine sayısını ifade eder.
5. Akış tipi çizelgeleme problemi, α değeri F_m ile gösterilir. m makine sayısını ifade eder. m adet seri makine vardır.
6. Esnek akış tipi çizelgeleme problemi, α değeri FF_s ile gösterilir. s tane aşama ve her bir aşamada paralel makineler olan bir ortamı ifade eder.
7. Açık atölye tipi çizelgeleme problemi, α değeri O_m ile gösterilir. m makine sayısını ifade eder. Üretilen ürünlerin rotalarıyla ilgili herhangi bir sınırlama yoktur.
8. Atölye tipi çizelgeleme problemi,α değeri J_m ile gösterilir. m makine sayısını ifade eder. Üretilen ürünlerin takip etmesi gereken bir rotası vardır.
9. Esnek atölye tipi çizelgeleme problemi, α değeri FJ_s ile gösterilir. s tane aşama ve her bir aşamada özdeş makinalar bulunur. Üretilen ürünlerin takip etmesi gereken bir rotası vardır (Pinedo, 2008).

Probleme ait karakteristikleri belirten β değeri için olası değerler şu şekilde sıralanabilir. Hazır olma zamanı (r_j), ayar zamanı (s_k), sıra bağımlı ayar zamanı (s_{jk}), bölünebilme (prmp), öncül-ardıl ilişkileri (prec), iş aileleri (fmls), toplu işleme (batch(b)), arızalar (brkdwn), makine erişilebilirlik kısıtları (M_j) , permütasyon (prmu), bloklama (block), beklemenin olmaması (nwt), yeniden dolaşım (recrc) (Pinedo, 2008).

Probleme ait optimize edilmek istenilen hedefleri ifade eden γ değeri için olası değerler ise şu şekilde sıralanabilir.

Maksimum tamamlanma süresi $\rightarrow C_{max}$
Maksimum pozitif geç kalma süresi $\rightarrow L_{max}$
Toplam ağırlıklı tamamlanma zamanı $\rightarrow \sum W_j C_j$

Azalan toplam ağırlıklı tamamlanma zamanı $\rightarrow(\sum W_j(1-e^{-rC_j}))$

Toplam ağırlıklı pozitif gecikme $\rightarrow\sum W_j T_j$

Ağırlandırılmış gecikmiş toplam iş sayısı $\rightarrow\sum W_j U_j$

1.2. Çizelgeleme Problemleri için Komplekslik Hiyerarşisi

Genellikle bir çizelgeleme problemi için geliştirilen bir algoritma bir başka problem için de uygulanabilir. Örneğin, $1 \parallel \sum C_j$ problemi, $1 \parallel \sum W_j C_j$ probleminin özel bir durumudur ve $1 \mid\mid \sum W_j C_j$ problemi için geliştirilen bir algoritma $1 \mid\mid \sum C_j$ problemi için uygulanabilir. Komplekslik terminolojisinde $1 \mid\mid \sum C_j$ problemi, $1 \mid\mid \sum W_j C_j$ probleminin alt kümesidir denilebilir ve şu şekilde gösterilir.

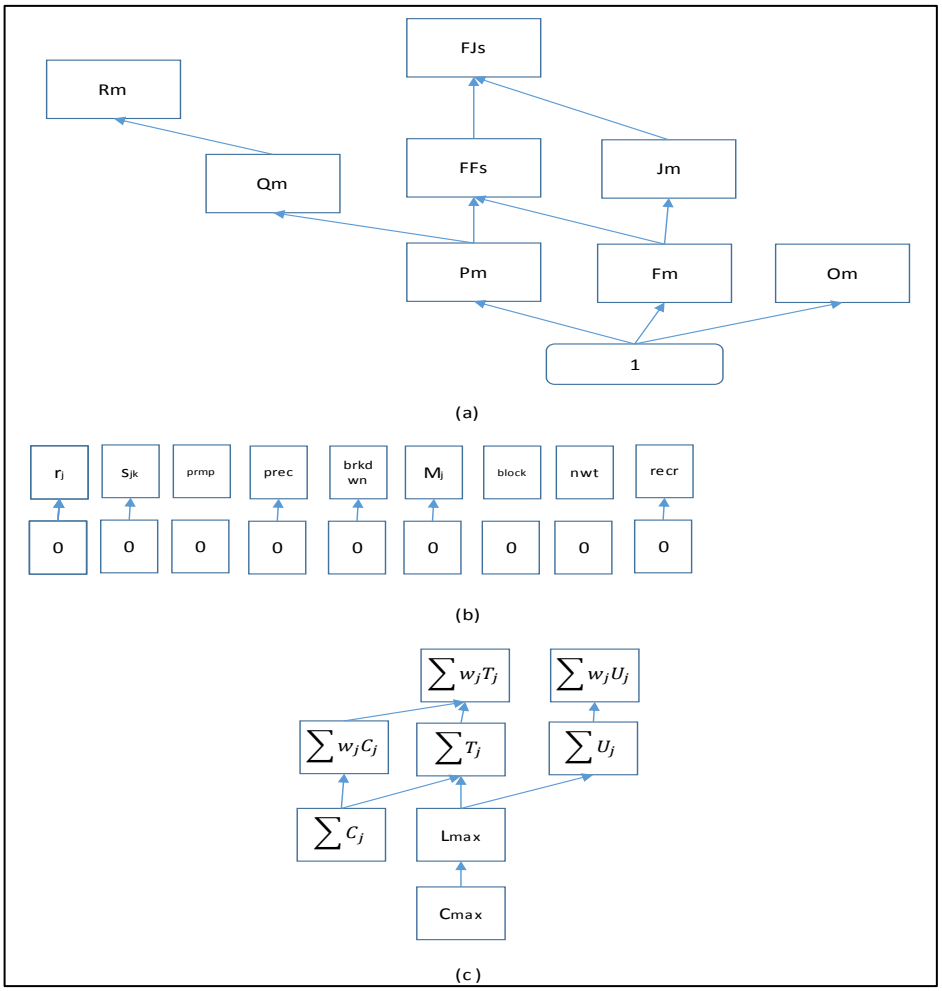
$$1 \mid\mid \sum C_j \propto 1 \mid\mid \sum W_j C_j \tag{1.1}$$

Benzer bir şekilde alttaki çıkarımlar yapılabilir.

$$1 \mid\mid \sum C_j \propto \quad 1 \mid\mid \sum W_j C_j \quad \propto \quad P_m \mid\mid \sum W_j C_j \propto \quad Q_m \mid\mid \sum W_j C_j \tag{1.2}$$

Elbette tüm problemler birbirleriyle bu şekilde bir ilişki içinde değildir. Birçok problem birbirleriyle kıyaslanamaz. Örneğin, $P_m \mid\mid \sum W_j Cj$ problemi ile $J_m \mid\mid C_{max}$ problemi birbiriyle kıyaslanamaz.

Problemler arası komplekslik hiyerarşisi Şekil 1.2'de gösterilmiştir. Bu gösterimden yola çıkılarak alttaki çıkarımlar elde edilebilir.



Şekil 1.2. Komplekslik hiyerarşisi

$$\alpha \mid \beta \mid L_{max} \propto \quad \alpha \mid \beta \mid \sum U_j \tag{1.3}$$

$$\alpha \mid \beta \mid L_{max} \propto \quad \alpha \mid \beta \mid \sum T_j \tag{1.4}$$

Dolayısıyla $\alpha \mid \beta \mid \sum T_j$ için geliştirilen herhangi bir algoritma ufak değişikliklerle $\alpha \mid \beta \mid L_{max}$ için de uygulanabilir (Pinedo, 2008). Ele alınacak olan zamana bağlı elektrik fiyatlandırmasına göre tek makine çizelgeleme problemi için hedeflerden biri toplam gecikme zamanının en azlanmasıdır. Dolayısıyla, önerilen yöntem aynı zamanda L_{max} maksimum pozitif geç kalma süresi hedefi için de kullanılabilir.

Çizelgeleme problemleri için uygulanan çözüm yöntemleri ise problemin zor ya da kolay olmasına göre değişir. Problemin girdilerine göre polinom zamanda çözülebilen problemler kolay, polinom olmayan zamanda optimum çözümün elde edilebildiği problemler zor olarak değerlendirilir. Çizelgeleme problemlerinin çözümünde uygulanan genel çözüm yöntemleri aşağıda sıralanmıştır.

1- Sıralama kurallarına göre üretilen çözümler. Bu yöntem kendi içinde ikiye ayrılır, bunlar statik ve dinamik sıralama kurallarıdır. Statik sıralama kuralları zamana bağlı değildir. İşlere ya da makinalara bağlı bir fonksiyonun değerine göre işler çizelgelenir. Örneğin, WSPT (ağırlıklı en kısa süreli işi en önce) böyle bir sıralama yöntemidir. Dinamik sıralama kuralları ise zamana bağlıdır. Örneğin, MS (minimum sapma) yönteminde işler $\max(d_j-p_j-t,0)$ değerine göre sıralanır. Böylece herhangi bir zamanda j işi k işinden önce çizelgenirken başka bir zaman sıralama değişebilir. Sıralama kuralları arasında SPT (En kısa zamanlı en önce), EDD (En erken termin zamanlı en önce), ERD (En erken hazır olma zamanı olan en önce) gibi kurallar verilebilir.

2- Karma sıralama kurallarına göre üretilen çözümler. Bu yöntem birden fazla sıralama kuralını içinde ihtiva eder. Örneğin, tek makine çizelgeleme ortamında toplam gecikme zaman minimizasyonu problemi için geliştirilen ATC (Açık Gecikme Maliyeti) yöntemi WSPT ve MS yöntemlerinin birleştirilmiş halidir. ATC yönteminde çizelgelenecek işlerden hesaplanan endeks değeri en yüksek olan alınır, çizelgelenir, kalan işler için endeks değeri tekrar hesaplanarak en yüksek çıkan iş çizelgelenmeye devam edilir.

3- Metasezgisel yöntemler. Bu yöntemlere örnek olarak tavlama benzetimi yöntemi, tabu araması yöntemi, genetik algoritma yöntemi, karınca kolonisi optimizasyonu yöntemi ve beam arama yöntemi verilebilir.

4- Ayrıştırma Metodları. Bu metod makine bazlı ya da iş bazlı ayrıştırma yöntemleridir. Örneğin, "darboğaz kaydırma" tekniği makine bazlı ayrıştırma metoduna örnek gösterilebilir.

5- Kısıt programlama yöntemi.

6- Ajan bazlı prosedürler.

7- Matematiksel modelleme yöntemi (Pinedo, 2008).

Bu tez çalışmasında yöntem olarak sıralama kurallarına göre çözüm geliştirme, matematiksel modelleme, probleme özel geliştirilmiş ayrıştırma metotları ve genetik algoritma yöntemleri kullanılmıştır.

2. ENERJİ MALİYETİNİN MİNİMİZASYONU

Yukarıda da bahsedildiği üzere problem öncelikli olarak enerji maliyetinin minimizasyonu problemi olarak ele alınmıştır. Problem termin kısıtı düşünülmeden incelendiğinde, ele alınan problemin literatürde tanımlanan GAPS2 (Özel Sıra Kümeli Genel Atama Problemi) problemine oldukça çok benzediği görülecektir. GAPS2 problemi ilk olarak Farias ve diğ. tarafından ele alınmıştır. GAPS2 problemi üzerinde az çalışılmış bir problemdir. Farias ve diğ. polihedral yaklaşımıyla küçük ölçekli problemleri çözerken, French ve Wilson lineer programlama tabanlı bir sezgisel algoritma geliştirmiştir. Herhangi bir işin başlama ve bitiş zamanları farklı periyotta ise ve tüketilen enerji miktarı işin farklı periyotlardaki harcadığı zamanın oranına göre ise problem GAPS2 problemi olarak ele alınabilir. Ancak, harcanan enerjinin büyük bir bölümü işe başlarken harcandığından ele alınan problem için tüketilen enerjinin tamamının işin başlama zamanında olduğu varsayılmıştır. Dolayısıyla, ele alınacak problem GAPS2 probleminden farklıdır. Aşağıda, incelenen problem, toplam gecikme zamanı kısıtını içeren bir matematiksel model üzerinden olarak anlatılacaktır.

S_i : i işinin başlama zamanı

P_i : i işinin işlem zamanı

X_{ik} : İkili değişken, eğer i işi k işinden önce çizelgelenmişse 1 değilse 0'dır.

C_j : j periyodunda birim tüketim fiyatı

V_{ji} : İkili değişken, eğer i işi j periyodunda çizelgelenmişse 1 değilse 0'dır.

A_j : j periyodunun başlama zamanı

B_j : j periyodunun bitiş zamanı

K_{ji} : İkili değişken, eğer S_i Aj'den büyükse 1 değilse 0'dır.

L_{ji} : İkili değişken, eğer S_i Aj'den küçük eşitse 1 değilse 0'dır.

R : Oldukça büyük bir sayı

E_i : i işinin enerji tüketim miktarı

D_i : i işinin termin zamanı

T_i : i işinin birim gecikme maliyet

F_i : i işinin gecikme miktarı = $\max\{0, S_i+P_i-D_i\}$

Min $\sum_{j=1}^{m}\sum_{i=1}^{n} C_j \cdot E_i \cdot V_{ji} + \sum_{i=1}^{n} T_i \cdot F_i$

Kısıtlar:

$S_i+P_i \leq S_k + R(1-X_{ik})$ $i=1,...,n-1$ $k=i+1,\ldots,n$ (2.1)

$S_k+P_k \leq Si + R \cdot X_{ik}$ $i=1,...,n-1$ $k=i+1,\ldots,n$ (2.2)

$S_i \geq 0$ $i=1,\ldots,n$ (2.3)

$X_{ik} \in \{0,1\}$ $i=1,...,n-1$ $k=i+1,\ldots,n$ (2.4)

$S_i+P_i \leq \sum_{i=1}^{n} P_i = C_{max}$ $i=1,\ldots,n$ (2.5)

$R \cdot K_{ji} > S_i - A_j$ $i=1,\ldots,n$ $j=1,\ldots,m$ (2.6)

$R \cdot L_{ji} \geq B_j - S_i$ $i=1,\ldots,n$ $j=1,\ldots,m$ (2.7)

$V_{ji} \geq K_{ji}+L_{ji}-1$ $i=1,\ldots,n$ $j=1,\ldots,m$ (2.8)

$V_{ji}, K_{ji}, L_{ji} \in \{0,1\}$ $i=1,\ldots,n$ $j=1,\ldots,m$ (2.9)

$F_i \geq S_i+P_i-D_i$ $i=1,\ldots,n$ (2.10)

Amaç fonksiyonu toplam enerji maliyetini ve toplam gecikme zamanını beraber içerir. F_i değeri tüm işler için oldukça büyük bir sayı alınarak hedef fonksiyonun birincil amacının toplam gecikme zamanının en azlanması olması istenmiştir. İlk dört kısıt Manne tarafından kullanılan sıra bağımlı ikili değişkenlerle ilişkilidir. İkili değişkenler i işinin j sırasına atanması şeklinde tutulursa n^2 kadar ikili değişken olması gerekirken, bu şekilde tanımlandığında n(n-1)/2 ikili değişken vardır. Beşinci kısıt işler arasında boşluk olmamasını amaçlar. 6, 7, 8 ve 9. kısıtlar i işinin hangi periyotta olduğunu anlamak içindir. Son kısıt ise gecikme miktarını belirler. Kurulan model ile iş sayısı 15 den büyük olan problemlerin çalıştırılması mümkün olmadığından tez çalışmasında bu modeli çözmek için farklı yöntemler denenecektir.

2.1. GAP (Genel Atama Problemi)

Literatürde GAPS2 problemi olarak anlatılan ve bizim ele aldığımız probleme benzeyen özel sıra kümeli genel atama problemini açıklamak için öncelikle GAP (genel atama problemi) anlatılacaktır. Genel atama problemi, sırt çantası (knapsack) probleminin terminolojisiyle anlaşılabilir. Elimizde, n adet parça ve m adet çanta varsa,

p_{ij}=j parçası i çantasına atanırsa elde edilecek kar,
w_{ij}=j parçası i çantasına atanırsa j parçasının ağırlığı,
c_i= i çantasının kapasitesi olsun.

Problem tüm parçaların (işlerin) maksimum karı sağlamak üzere çantalara (periyotlara) atanması şeklindedir.

Max $z= \sum_{j=1}^{m} \sum_{i=1}^{n} p_{ij}.x_{ij}$

Kısıtlar:

$$\sum_{j=1}^{n} w_{ij}.x_{ij} \leq c_i \qquad i=1,...m \qquad (2.11)$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad j=1,...n \qquad (2.12)$$

$$x_{ij} = 0 \text{ veya } 1 \qquad i=1,...m \quad j=1,\ldots n \qquad (2.13)$$

x_{ij} değeri j parçası i çantasına atanmışsa 1, değilse 0 olacaktır. Bizim ele alacağımız problemde, w_{ij} değeri işin yapılma zamanıdır. Dolayısıyla, periyoda bağlı olmadığında w_i olarak alınabilir. Yukarıda anlatılan tek makina çizelgeleme probleminde işlerin periyotlara atanması problemiyle GAP problemi benzerlik göstermesine rağmen, bu tezde ele alınacak problemde bir iş iki farklı periyoda atanabilir. Eğer işin başladığı ve bittiği periyotlar farklıysa, bu iş, iki farklı periyodun kapasitesinden kullanıyor olacaktır. Martello ve Toth GAP probleminin minimizasyon versiyonunu, c_{ij} değerinin j parçasının i çantasına atılma maliyeti olduğu durum için anlatmışlardır.

GAP probleminin çözülmesi için farklı metotlar kullanılmıştır. Bu yöntemler tam çözümü hedefleyen ya da yaklaşık çözümü hedefleyen yöntemlerdir. Dal-sınır

yöntemi içinde tabu arama kullanılarak tam çözüm elde edildiği gibi (Woodcock ve Wilson, 2010), arı algoritması kullanılarak optimuma yakın çözümler de hedeflenmiştir (Özbakır ve diğ., 2010). Problemin çözümünde lagrange gevşetimine dayanan sezgisel algoritmalar geliştirilmiş ve uygulanmıştır (Jeet ve Kutanoğlu, 2007). GAP probleminin birçok özel durumu literatürde incelenmiş ve çözülmüştür. LEGAP adı verilen problem türü GAP probleminin özel bir türüdür. Bu problem türü için $p_{ij} = p_j$ ve $w_{ij} = w_j$'dir, bu durumda kazanç ve ağırlık çantadan bağımsızdır. LEGAP problemi için de lagrange gevşetimi kullanılarak çözümler elde edilmiştir (Chalmet ve Gelders, 1977). GAP probleminin en bilinen özel durumu atama problemidir. Bu problem için n=m, $c_i=1$ ve $w_{ij}=1$'dir. Bu problem Macar algoritmasıyla $O(n^3)$ zamanda çözülebilir. Bizim ele aldığımız şekilde $w_{ij}=w_i$ olduğu durumlarda literatürde ele alınmış ve tam çözüm yöntemleri ile çözümler araştırılmıştır (Srinivasan ve Thompson, 1973). GAP probleminin denklik genel atama problemi (EGAP) adı verilen başka bir türünde ise amaç fonksiyonu farklılaştırılarak çözüm hedeflenmiştir. Bu problemin çözümünde ise genetik algoritma kullanılarak hedeflenen amaç fonksiyonu genetik algoritmanın uygunluk fonksiyonu olarak tanımlanmış ve çözülmüştür (Linzhong ve diğ., 2012).

GAP probleminin tam çözümüne yönelik yapılan çalışmalarda iyi bir gevşetme ile iyi bir üst sınır elde edilmeye çalışılmıştır. Bununla ilgili yapılan gevşetme yöntemleri aşağıda belirtilmiştir.

1. Kapasite kısıtları bazında gevşetim yöntemi:

Denklem (2.11) kısıtı gevşetilerek onun yerine;

$$w_{ij}.x_{ij} \leq c_i \qquad i \in M, j \in N \qquad (2.14)$$

kısıtı eklenir. Elde edilen problemin çözümü her bir işin en çok kar getiren çantaya girmesi şeklindedir, dolayısıyla elde edilecek üst sınır

$$U_0 = \sum_{j=1}^{n} p_{i(j),j} \qquad (2.15)$$

değeridir. Bu değer gerçek kısıta uymadığı durumlar tespit edilerek iyileştirilir ve daha iyi üst sınır değerleri elde edilir.

2. Atama kısıtları bazında gevşetim yöntemi:

Denklem (2.12) kısıtı kaldırılarak bir üst sınır elde edilebilir. Bu şekliye problem 0-1 sırt çantası problemlerine bölünebilir. Bu şekilde elde edilen m adet 0-1 sırt çantası problemi çözdürüldüğünde elde edilen optimum çözümlerin toplamı asıl problem için bir üst sınırdır.

Max $z_i=\sum_{j=1}^{n} p_{ij}x_{ij}$

Kısıtlar:

$$\sum_{j=1}^{n} w_{ij}x_{ij} \leq c_i \qquad (2.16)$$

$$x_{ij} = 0 \text{ veya } 1 \qquad j=1,\ldots n \qquad (2.17)$$

Yukarıda şekilde oluşan m problem çözümünün toplamı bir üst sınırdır.

$$U_0 = \sum_{i=1}^{m} z_i \qquad (2.18)$$

Benzer şekilde bu üst sınır için gevşetilen kısıta aykırı durumlar ele alınarak üst sınır değeri iyileştirilebilir.

3. Katsayı değiştirme yöntemi:

Lagrange gevşetimi yöntemine dayanan katsayı değiştirme yöntemi, lagrange gevşetiminde kullanılan λ değerinin değiştirilerek daha iyi üst sınırlar elde edilmesini hedefler.

4. Değişken bölme yöntemi:

Bu yöntemde ise modele ilave edilen y_{ij} ikili değişkenleri ile α ve β parametreleri sayesinde problem alttaki hale dönüştürülür.

Max $\alpha\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} + \beta\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}y_{ij}$

Kısıtlar:

$$\sum_{j=1}^{n} w_{ij}x_{ij} \leq c_i \qquad i=1,\ldots m \qquad (2.19)$$

$\sum_{i=1}^{m} y_{ij}=1$ j=1,...n (2.20)

$x_{ij} = y_{ij}$ i=1,…m j=1,…n (2.21)

x_{ij} =0 veya 1 i=1,…m j=1,…n (2.22)

y_{ij} =0 veya 1 i=1,…m j=1,…n (2.23)

Bu şekilde bir modele Denklem (2.21) kısıtı üzerinde lagrange gevşetimi uygulandığında problem iki ayrı problem haline dönüştürülür. Bu problemlerden bir tanesi m tane 0-1 knapsack problemidir, diğeri ise birinci yöntemde anlatılan kapasite kısıtı gevşetilmiş problemle aynı yapıdadır ve ikinci kısmın çözümü oldukça kolaydır. Bu yöntemle elde edilen üst sınır değeri diğer yöntemlere göre daha iyi sonuç vermektedir (Martello ve Toth, 1990).

2.1. Özel Sıra Kümeli Genel Atama Problemi (GAPS2)

Yukarıda da belirtildiği üzere, GAPS2 problemi literatüre yeni kazandırılmış ve üzerinde çok fazla çalışma yapılmamış bir problemdir. Özel sıra kümeleri (ÖSK) Beale ve Tomlin tarafından ortaya atılmıştır ve birçok dal-sınır algoritmasında kullanılmaktadır. Diyelim ki, x_{ij} değeri j işinin i periyodundaki gerçekleşen oranı olsun. Tüm işler için $\sum_{i=1}^{m} x_{ij} = 1$ olmalıdır. İşler yekpare yapılması gerektiğinden ve periyotların süresi işlerin yapım süresine göre oldukça büyük olduğu varsayımıyla, bir iş ikiden fazla periyoda atanamaz. Eğer bir iş iki periyoda atanmışsa, bu iki periyot komşu olmalıdır. Bu tanımlama $\{x_{1j},\ldots,x_{mj}\}$ kümesinin 2.tip bir özel sıra kümesi olduğunu gösterir. Böylelikle problem aşağıdaki gibi tanımlanabilir.

Max $z=\sum_{j=1}^{m} \sum_{i=1}^{n} p_{ij}x_{ij}$

Kısıtlar:

$\sum_{j=1}^{n} w_{ij}x_{ij} \leq c_i$ i=1,...,m (2.24)

$\sum_{i=1}^{m} x_{ij} = 1$ j=1,...,n (2.25)

$x_{ij} \geq 0$ i=1,...,m j=1,…,n (2.26)

$\{x_{1j},\ldots,x_{mj}\}$ 2. tip ÖSK $j=1,\ldots,n$ (2.27)

GAP probleminde olduğu gibi GAPS2 problemi de minimizasyon problemi olarak ele alınabilir.

2.2. Önerilen Yöntem

Problemle ilgili önerilen yöntemin anlaşılabilmesi için öncelikle problemle ilgili varsayımlar ve problemin kompleksliği anlatılacaktır.

2.2.1. Varsayımlar

Tüm işlerin 0 anında yapılmaya hazır olduğu varsayılacaktır. Çizelgenin sıfır anı, elektrik fiyatlandırmasına uygun olarak 17:00 olarak alınacaktır. Başlangıç ve bitiş zamanı farklı olan bir iş için birim elektrik tüketim fiyatı olarak, işin başlangıcına ait periyottaki birim tüketim fiyatı kullanılacaktır. Bu varsayım, ele alınan lastik fabrikası ortamında elektrik tüketiminin büyük bir kısmının işin başlama zamanında olmasından kaynaklanmaktadır. İşlerin işlem zamanı periyot uzunluğuna göre çok kısa olduğundan, bir iş başladığı periyotta biter veya başladığı periyottan bir sonraki periyotta sona erer. Yukarıda da bahsedildiği üzere işler arası boşluğa izin verilmediği için C_{max} değeri tüm işlerin işlem zamanlarının toplamına eşit olacaktır. İşler arası ayar zamanı vardır. Ancak, bu süreler işlem zamanının içine katılmış ve işler arası ayar zamanı yokmuş gibi çizelgeleme yapılmıştır.

2.2.2. Problemin kompleksliği

Termin kısıtını düşünmeden bile problem Np-Zor bir problemdir. Problemin Np-Zor olduğu şu şekilde gösterilebilir. 1'den n'e kadar olan işlerin tek makine çizelgeleme problemi için işlerin toplam işlem zamanlarının iki farklı elektrik periyodunu içerdiği durum dikkate alınmıştır. Bu periyodlardan birincisinin 1 kwH elektrik için fiyatının 1 birim, ikinci periyodun 1 kwH elektrik için birim fiyatının 0 birim olduğu varsayılırsa ve ikinci periyodun uzunluğu t ise, hedef elektrik faturasındaki değeri minimuma düşürmek için işleri mümkün olduğunca ikinci periyotta çizelgelemek olmalıdır. Bu problem klasik sırt çantası problemidir (knapsack problem). Anlatıldığı şekliyle problem aşağıdaki gibi özetlenebilir.

Max $\sum_{j=1}^{n} X_j.E_j$

Kısıtlar:

$$\sum_{j=1}^{n} P_j X_j \leq t \quad (2.28)$$

$$X_j \in \{0,1\} \quad (2.29)$$

Burada X_j 1 değerini almışsa, ikinci periyotta çizelgelenmiş demektir. Karp sırt çantası probleminin Np-Zor olduğunu göstermiştir. Bizim problemimizin küçük bir hali bile Np-Zor olduğundan problem termin kısıtı olmasa dahi Np-Zordur.

2.2.3. Önerilen model

Giriş bölümünde anlatılan zamana bağlı değişen elektrik fiyatlanmasına bağlı olarak tek makina çizelgeleme problemi literatürde daha önceden ele alınmamış bir problem türüdür. Her ne kadar, başka amaçlarla benzer problemler ele alınmış, optimum ve yaklaşık çözümler için yöntemler geliştirilmiş olsa da ele alacağımız problem yapısı itibariyle diğer problemlerden farklıdır. Elektrik tüketimleri arasında fark bulunan işlerin çizelgelendiği sektörlerde (çelik, çimento, lastik v.b) TOU tarifesine uygun olarak çizelgeleme yapmak getirisi oldukça yüksek bir yöntem olacaktır. Böyle bir yöntemle çizelgeleme yapmak yaklaşık %10-%15 arası bir maliyet iyileştirmesi sağlayabilecektir. Rekabetin oldukça yüksek olduğu günümüzde, böyle bir yaklaşımla herhangi bir yatırım yapmadan, kaliteyi etkilemeden ve en önemlisi işleri geciktirmeden maliyeti azaltıyor olmak oldukça önemlidir.

Yukarıda giriş bölümünde anlatılan problem, ikinci bölümde matematiksel model olarak ifade edilmiştir. Ancak, bu matematiksel modelle iş sayısının 15'den büyük olan bir problem için, herhangi bir çözücü program yardımıyla çözülmesi mümkün olmamıştır. Gerçek hayattaki problemlerin daha büyük olduğu düşünülürse problemi farklı bir yöntemle ele almak gerektiği açıktır. GAP modelinin çözümü için literatürde gevşetme algoritmaları türetilmiştir. Bu gevşetmeler, problemi daha basit hale getirerek, dal-sınır algoritmalarında kullanılmaktadır. Tezde yukarıda model 1 olarak anlatılan model, aşağıdaki gibi farklı bir şekilde modellenerek çözülecektir.

Aşağıda probleme ait bazı özellikler belirtilmiştir.

Özellik 1:

Eğer tüm işlerin toplam işlem zamanı ilk periyodun süresinden kısaysa, herhangi bir sıralama eniyi sonucu verir.

İşler arası boş zamana izin verilmediğinden,

$C_{max}<B_{1,}$

Toplam enerji maliyeti,

$\sum_{j=1}^{1} \sum_{i=1}^{n} C_j.E_i = C_1 . \sum_{i=1}^{n} E_i$'dir.

Dolayısıyla toplam enerji maliyeti sıraya bağlı değildir ve sabittir.

Özellik 2:

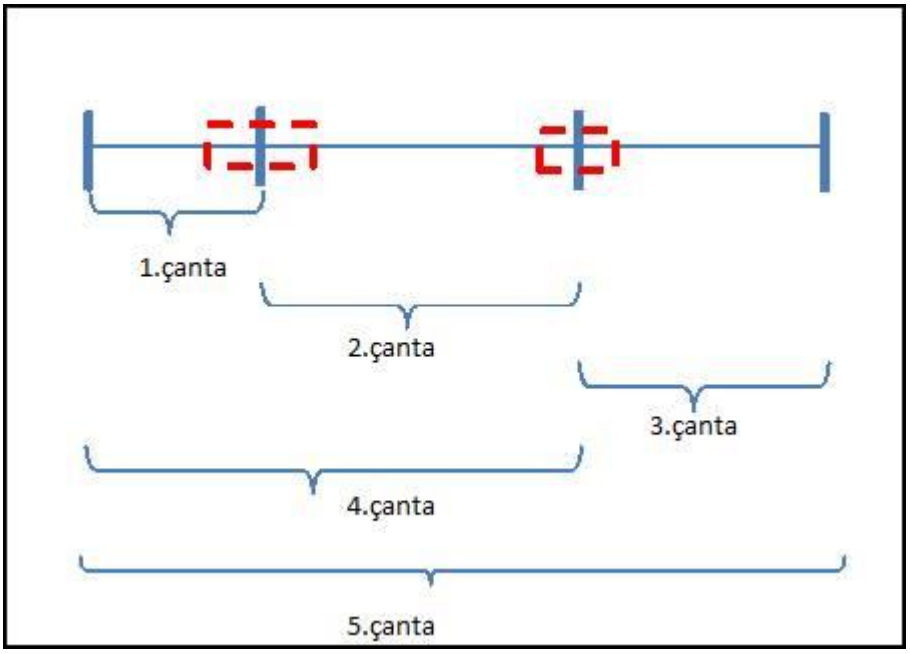
Ele alınan problem için S minimum maliyetli sıralamayı veriyor olsun. Bu sıralama içinde başlangıç ve bitiş zamanları aynı periyod içinde olan herhangi iki işi yer değiştirdiğimizde elde edilen sıralama S′ ise, S ve S′ maliyeti aynıdır. Dolayısıyla S′ için de optimum çözümü veriyor denilebilir.

Özellik 3:

Optimum sıralamadaki son iş çıkarıldığında, kalan işler için eldeki sıralama yine optimum çözümü verir.

Özellik 2 kullanılarak, problemin başlama zamanı yerine hangi periyotta başladığı bir karar değişkeni olarak belirlenirse, problem daha farklı bir şekilde modellenebilir. Periyotlar arası geçişlerdeki işlerin ne olacağı, modeli çözdürmeden önce belirlenecektir. Bu şekilde bir sabitleme Yeung ve diğ. tarafından geliştirdikleri 1 nolu algoritmada kullanılmıştır. Eğer toplam n tane iş ve m periyod varsa toplam m-1 işin sabitlenmesi gerekir. Kalan n-m+1 işin çizelgelenmesi problemi genelleştirilmiş atama probleminin (GAP) özel bir durumu olarak düşünülebilir. Burada çantalar (knapsack) periyodların kendisi ve aşağıda açıklandığı şekilde kombinasyonlarıdır. Şekil 2.1'de gösterildiği şekliyle 3 periyoddan oluşan bir çizelgeleme dönemi sözkonusu ise, mevcut 3 periyoda ilave olarak 4. ve 5. çantalar ilave edilmiştir. 4.

çantanın kapasitesi, 1.çantanın kapasitesi ve 2.çantanın kapasitesinin toplamından ilk sabitlenen işin uzunluğu çıkartılmış kadardır. Bir iş 1.çantaya atanmışsa aynı zamanda 4.çantaya ve 5.çantaya da atanmalıdır.



Şekil 2.1. Problemin sırt çantası problemine benzetimi

GAP probleminde, çantalara konulacak eşyaların ağırlıkları ve kazançları konulacağı çantaya göre değişir. Bizim problemimizde ise, ağırlık işlerin işlem zamanına denk gelir ve atandığı periyottan bağımsızdır. Alttaki modelde (model 2) , m-1 iş sabitlendiği için, n sayısı aslında kalan iş sayısı olan n-m+1 değeridir. Model aşağıdaki gibidir:

$$\text{Max } Z = \sum_{i=2}^{m} \sum_{i=1}^{n} G\text{-}C_j \; E_i \,.\, (X_{ij} - X_{i\,j-1}) + \sum_{j=1}^{n} X_{i1} E_i \,(G\text{-}C_1)$$

Kısıtlar:

$$\sum_{i=1}^{n} P_i X_{ij} \quad < B_j - \sum_{k=1}^{j} H_k \quad j \text{ in } \{1,2,\ldots, m\text{-}1\} \qquad (2.30)$$

$$\sum_{i=1}^{n} P_i X_{im} \quad \leq C_{max} - \sum_{k=1}^{m} H_k \qquad (2.31)$$

$$X_{ij} - X_{i\,j+1} \quad \leq 0 \quad i=1\ldots n \quad j \text{ in } \{1,2,\ldots, m\text{-}1\} \qquad (2.32)$$

$$Xik \in \{0,1\} \quad i=1,\ldots,n \quad k=1,\ldots,m \qquad (2.33)$$

$$\sum_{i=1}^{n} P_i (X_{ij}-X_{i\,j-1}) \quad < B_j - A_j \quad j \text{ in } \{2, \ldots, m-1\} \quad (2.34)$$

$$\sum_{i=1}^{n} P_i (X_{ij}-X_{i\,j-1}) \quad < C_{max} - A_j \quad j = m \quad (2.35)$$

Yukarıdaki modele ait opl kodu EK-A'da incelenebilir. Burada X_{ij} değişkeni 1 değerini eğer i işi j periyodunda veya j periyodundan önce herhangi bir periyotta çizelgelenmişse alır. X_{ij} 0 değerini eğer i işi j periyodundan sonra çizelgelenmişse alır. Denklem (2.34) ve (2.35) kısıtları sırt çantası kapasitelerini gösterir. Her bir periyot için, kendisinden önce gelen periyotlarlarla beraber yeni bir sırt çantası tanımlanmıştır. Bu tanımlama Denklem (2.30) ve (2.31) kısıtlarıyla ifade edilmiştir. Burada H_k değeri k periyodu için sabitlenen işin işlem zamanıdır. H_1 sıfırdır çünkü ilk periyot için sabitlenen bir iş yoktur. Problem bir maksimizasyon problemidir. Bunun anlamı amaç fonksiyonundaki değer, her bir işin pahalı bir periyot yerine ucuz periyotta çizelgelenmesinin getirisi olarak düşünülebilir. G değeri en pahalı periyodun elektrik birim fiyatı olarak düşünülebilir.

Toplam elektrik maliyeti $= W = \sum_{i=1}^{n} G.E_i - Z +$ (sabit işlerden gelen sabit maliyet)

Bu durumda iki farklı problem ortaya çıkmaktadır.

1- Sabit işlerin belirlenmesi
2- Modelin tam çözümü

Birinci problem için toplam n!/ (n-m+1)! farklı seçenek bulunmaktadır. Her bir seçimde, sabitlenen işlerden gelen sabit maliyet çizelgeleme yapmaya başlamadan önce hesaplanabilir. İkinci problemin gevşetilip çözülmesi bize problemin bütünü için bir alt sınır verecektir.

2.2.4. Model 2'nin gevşetilmesi

Bölüm 2.1'de bahsedildiği üzere GAP probleminin gevşetilmesi ile ilgili bir çok yöntem önerilmiştir. Bunlardan biri de değişken bölme metodudur. Bu metot Jörnsten ve Nasberg tarafından önerilmiştir. Bu yöntemde ilave değişkenler lagrange çarpanlarının kullanılmasını sağlamıştır. Benzer bir yaklaşımla ilave ikili değişkenler kullanarak Z değeri için lagrange çarpanları kullanılabilir. Bu sayede Z değeri için iyi bir üst sınır bulunabilir. Bunun için model 2, model 3'e dönüştürülmüştür.

2.2.5. Matematiksel model 3

$$\text{Max } Z = \sum_{j=2}^{m}\sum_{i=1}^{n}[(G - C_j).E_i.Y_{ij} + \sum_{j=1}^{n} X_{i1}E_i(G-C_1)$$

Kısıtlar:

$$\sum_{i=1}^{n} P_i X_{ij} < B_j - \sum_{k=1}^{j} H_k \quad j = 1,2,\ldots, m-1 \qquad (2.36)$$

$$\sum_{i=1}^{n} P_i X_{im} \leq C_{max} - \sum_{k=1}^{m} H_k \qquad (2.37)$$

$$Y_{ij} = X_{ij} - X_{i\,j-1} \quad i=1\ldots n \quad j=2,\ldots, m \qquad (2.38)$$

$$Y_{ik}, X_{ik} \in \{0,1\} \quad i=1,...,n \quad k=1,\ldots,m \qquad (2.39)$$

$$\sum_{i=1}^{n} P_i Y_{ij} < B_j - A_j \quad j=2,\ldots,m-1 \qquad (2.40)$$

$$\sum_{i=1}^{n} P_i Y_{ij} < C_{max} - A_j \quad j = m \qquad (2.41)$$

Denklem (2.38) kısıtı çıkarılıp, amaç fonksiyonuna eklenerek model alttaki hale dönüştürülür.

$$\text{Max } Z_1 = \sum_{j=2}^{m}\sum_{i=1}^{n}[(G - C_j).E_i.Y_{ij} + \sum_{j=1}^{n} X_{i1}E_i(G-C_1) - \sum_{j=2}^{m}\sum_{i=1}^{n}\lambda_{ij}(Y_{ij}-X_{ij}+X_{ij-1}))$$

Kısıtlar:

Denklem (2.36), (2.37), (2.39), (2.40), (2.41).

Yukarıdaki L (Model 3, λ) problemi iki alt probleme dönüştürülebilir.

Birinci bölüm:

$$\text{Max } Z_a = \sum_{j=2}^{m}\sum_{i=1}^{n}[(G - C_j).E_i - \lambda_{ij}].Y_{ij}$$

Kısıtlar:

$$\sum_{i=1}^{n} P_i Y_{ij} < B_j - A_j \quad j=2,\ldots,m-1 \qquad (2.42)$$

$$\sum_{i=1}^{n} P_i Y_{ij} < C_{max} - A_j \quad j = m \qquad (2.43)$$

$$Y_{ik} \in \{0,1\} \quad i=1,...,n \quad k=1,\ldots,m \qquad (2.44)$$

İkinci bölüm:

$$\text{Max } Z_b = \sum_{j=1}^{m} \sum_{i=1}^{n} B_{ij} \cdot X_{ij}$$

Kısıtlar:

$$\sum_{i=1}^{n} P_i X_{ij} < B_j - \sum_{k=1}^{j} H_k \qquad j = 1,2, \ldots, m-1 \qquad (2.45)$$

$$\sum_{i=1}^{n} P_i X_{im} \leq C_{max} - \sum_{k=1}^{m} H_k \qquad (2.46)$$

$$X_{ik} \in \{0,1\} \qquad i=1,...,n \qquad k=1,\ldots,m$$

Birinci bölüm yine parçalanıp (m-1) tane 0-1 sırt çantası problemine dönüştürülebilir. İkinci kısım yine aynı şekilde m tane 0-1 sırt çantası problemine parçalanabilir. Böylelikle, 2m-1 adet sırt çantası problemi pseudo polinom zamanda çözülerek lagrange gevşetmesinin çözümü bulunabilir. Bu modelin verdiği çözüm Denklem (2.38) kısıtını da sağlıyorsa, elde edilen çözüm ana problem için optimum çözümü veriyor denilebilir.

Tablo 2.1 X_{ij}, $X_{i\ j-1}$ ve Y_{ij} nin alabileceği değerleri ve amaç fonksiyonuna etkisini göstermektedir. (Y_{ij}-X_{ij}+X_{ij-1}) nin negatif olduğu tek kombinasyon X_{ij}=1, X_{ij-1} ve Y_{ij}'nin 0 olduğu durumdur. L (Model 3, λ) modeline yeni bir kısıt ekleyip (kesme) bu kombinasyon engellenirse, bu gevşetmenin sonucu her zaman optimum çözümü verir.

Böylelikle, yeni ikili değişkenler tanıtılmış ve L (Model 3, λ) modeline ilave edilmiştir.

Tablo 2.1. X_{ij}, $X_{i\ j-1}$ ve Y_{ij} nin alabileceği olası değerler

X_{ij}	$X_{i\ j-1}$	Y_{ij}	$(Y_{ij}-X_{ij}+X_{ij-1})$
0	1	0	1
0	0	0	0
0	0	1	1
0	1	1	2
1	1	0	0
1	1	1	1
1	0	0	-1
1	0	1	0

2.2.6. Model 4

$$\text{Max } Z_1 = \sum_{j=2}^{m}\sum_{i=1}^{n}(G-C_j).E_i.Y_{ij} + \sum_{j=1}^{n}X_{i1}E_i(G-C_1) - \sum_{j=2}^{m}\sum_{i=1}^{n}\lambda_{ij}(Y_{ij}-X_{ij}+X_{ij-1}))$$

Kısıtlar:

$$\sum_{i=1}^{n}P_iX_{ij} < B_j - \sum_{k=1}^{j}H_k \qquad j = 1,2,\ldots,m-1 \qquad (2.47)$$

$$\sum_{i=1}^{n}P_iX_{im} \leq C_{max} - \sum_{k=1}^{m}H_k \qquad (2.48)$$

$$Y_{ik}, X_{ik}\ Z_{ik} \in \{0,1\} \qquad i = 1,\ldots,n \qquad k=1,\ldots,m$$

$$P_i\ Y_{ij} < B_j - A_j \qquad j = 2,\ldots,n-1 \qquad (2.49)$$

$$\sum_{i=1}^{n}P_iY_{ij} < C_{max} - A_j \qquad j = n \qquad (2.50)$$

$$X_{ij}+X_{i\ j-1} + Y_{ij} = 2.Z_{ij} \qquad i= 1,\ldots,n \qquad k=1,\ldots,m$$

Son kısıt olan Denklem (2.50) kısıtı, çözüm uzayındaki X_{ij}=1, X_{ij}-1 ve Y_{ij} nin 0 olduğu durumları engeller. Aynı zamanda ($Y_{ij}-X_{ij}+X_{ij-1}$ = 0) olma durumu ile ilgili herhangi bir sınırlama getirmez. Böylelikle model 4'ün optimum çözümü aynı zamanda model 2 ve 3 içinde optimum çözümü verir. ($Y_{ij}-X_{ij}+X_{ij-1}$) nin negatif olduğu tek kombinasyon X_{ij}=1, X_{ij-1} ve Y_{ij} 0 olduğu durumdur. L(Model 3, λ) modeline yeni bir kısıt ekleyip (kesme) bu kombinasyon engellenirse, bu gevşetmenin sonucu her zaman optimum çözümü verir.

2.2.7. Model 3'ün çözüm için kullanılması

Model 2'nin lagrange gevşetimi olan model 3'ü kullanarak optimum çözümün elde edilmesi içinse aşağıda bahsedilen yöntem kullanılmıştır. Yukarıda bahsedilen modeller ilog-opl versiyon 6.3 kullanarak kodlanmış ve cplex 12.1.0 ile çözdürülmüştür.

Lagrange gevşetiminde çıkarılan Denklem (2.38) kısıtı;

$Y_{ij} = X_{ij} - X_{i\,j-1}$ $\quad i=1\ldots n \quad j = 2,\ldots, m$

Yerine;

$Y_{ij} = X_{ij} - X_{i\,j-1}$ $\quad (i,j) \in S$

Kısıtı ilave edilmiştir.

Yöntem 1:

1. $S = Ø$,
2. Model 3,
3. Model 3'ü çözdür,
4. Elde edilen çözümdeki $(Y_{ij}-X_{ij}+X_{ij-1})$ değerlerini kontrol et,
5. $S' = S$,
6. $S = S \cup \{(i,j) \mid Y_{ij}-X_{ij}+X_{ij-1} \neq 0\}$,
7. Eğer $S'=S$ => optimum bulundu.

Değilse, 2'ye dön

Bu yönteme ait opl kodu EK-B'de verilmiştir. Bu yöntem kullanılarak problemler çözdürüldüğünde özellikle iş sayısı büyük olan problemler için ilave edilen kısıtlarla beraber mevcut yöntemin model 2 ile kıyaslandığında bir avantajının olmadığı gözlemlenmiştir. Bu sorunu ortadan kaldırabilmek için şöyle bir yöntem geliştirilmiştir.

Yöntem 2:

1. Model 2'yi belirlenen t zamanı boyunca çözdür,
2. Elde ettiğin en iyi çözüm vektörünü model 3'ün başlangıç çözümü yap,

3. Model 3‘ü tüm kısıtlarıyla beraber çözdür.

Bu yönteme ait opl kodu EK-C’de verilmiştir.

2.2.8. Problemlerin oluşturulması

Çözümün gerçek hayatı yansıtması için bir lastik fabrikasındaki koşullar incelenmiş ve bu ortamı yansıtabilecek değişik büyüklükte problemler oluşturulmuştur. Bu problemler iş sayısının 60,90 ve 120 olduğu, her bir işin elektrik tüketim miktarının [25,35] arası uniform dağıldığı, işlem zamanlarının ise [20, 30] arası uniform dağıldığı varsayılarak yaratılmıştır. Termin zamanı ise her bir iş için [$C_{max}/2$, $2.C_{max}$] arasında uniform dağıtılmıştır. Her bir iş sayısı için toplam 10 problem üretilmiştir. Matematiksel model 1 diye adlandırdığımız modelle 60 işlik bir problemi çözmek donanımsal problemlerden dolayı mümkün olmamıştır. Modelin çözebileceği en alt sınırı belirlemek için, 10 işlik, 15 işlik, 20 işlik problem seti hazırlanmış ve en küçük problemden başlayarak model çözdürülmeye çalışılmıştır. 15’lik problem çözülebilmesine rağmen, 20 lik problem çözülememiştir.

2.2.9. Sabitlenecek işlerin belirlenmesi

Sabitlenecek işlerin belirlenmesi için aşağıda belirtilen yöntem uygulanmıştır.

Yöntem:

1. P = Periyot listesi
2. I = İş listesi
3. P’ den sıradaki periyod olan p yi seç.
4. Seçilen periyod birinci periyotsa (en pahalı periyot) I’dan en az elektrik çekeni iş i yi seç ve bu periyoda ata.

Seçilen periyot ikinci periyotsa (en ucuz periyod) I’dan en çok elektrik çekeni iş i yi seç ve bu periyoda ata.

Seçilen periyod üçüncü periyodsa bir şey yapma

5. P = P – {p}
6. I = I – {i}

7. P’de seçilecek periyod kalmışsa 3’e geri dön, yoksa 8’e git.
8. P’nin ilk elemanına git.
9. P’den sıradaki periyod olan p’yi seç.
10. I’dan en az elektrik çekeni iş i yi seç ve bu periyoda ata.
11. P = P – {p}
12. I = I – {i}
13. Eğer P = ∅ bitir, yoksa 9’a dön.

Yukarıda bahsedilen problemlerin oluşturulması ve sabit işlerin belirlenmesi yöntemi delphi ile kodlanmış ve sabit işler seçilmiştir.

2.3. Problemlerin Sonuçları

Yukarıda bahsedilen 30 problemin model 2 kullanılarak çözdürüldüğünde 90 işlik ve 120 işlik problemler için çözüm süresinin kabul edilebilir seviyelerde olmadığı söylenebilir. Model 3 ise, yukarıda bahsedildiği üzere gevşetilmiş hali ile problemi çok basit bir hale getirdiği için aynı problemi çok kısa bir sürede çözebilmektedir. 120 işlik problemler için model 3 kullanılarak yöntem 1 diye adlandırılan metot uygulandığında 10 problemin beşi makul sürelerde çözdürülmüştür. Tablo 2.2’de model 3 (yöntem 1) olarak adlandırılan metot için başarısız olarak işaretlenenler, bu yöntemin probleme ilave ettiği kısıtlarla beraber elde edilen ara problemin model 2’nin aynısı veya çok yakın bir hale dönüştüğünü gösterir ve problemin model 2’de olduğu gibi makul sürelerde çözülemez hale geldiği anlamına gelir. Tabloda Model 3 (yöntem 2) olarak adlandırılan metod ise 120 işlik problemlerde yöntem 1’e ilave olarak bir problemi çözdürebilmiştir.

Model 2’nin iş sayısı 90 veya 120 olan problemlerde çözüm süresi çok uzun sürdüğü için zaman sınırlaması verilmiş ve çözüm süresi 2000 saniye ile sınırlandırılmıştır. Elde edilen çözüm değeri bu süre içindeki elde edilen en iyi çözüm değeridir. Tüm yöntemlerin optimumu garanti edemediği dört problem vardır. Bunlar 120_2, 120_4, 120_5, 120_10 problemleridir.

Özellikle 120 işlik problemlerin çözümünde yukarıda bahsedildiği gibi mevcut yöntemlerin yetersiz kalmasından dolayı başka bir yöntem daha geliştirilme ihtiyacı doğmuştur. Model 2 için olurlu herhangi bir çözüm, model 3 için de olurlu bir çözüm

olacağından, model 2'nin kısa bir sürede elde edeceği iyi bir çözüm üzerinden model 3'ün çalıştırılması yöntemi denenmiştir. Model 3'ün, model 2'de verilen zaman kısıtından dolayı elde edilen en iyi çözüm değerinden daha iyi sonuç verdiği dört problem vardır. Bunlar 90_3, 90_6, 90_7 ve 120_7 probemleridir. Model 3 yöntem 2 olarak bahsedilen yöntem bundan sonra "tek makine enerji maliyeti minimizasyonu" TMEMM yöntemi olarak adlandırılacaktır.

Tablo 2.2. Problemler ve çözüm süreleri

	SONUÇLAR				
Problem Adı	Model 2 Çözüm Değeri	En İyi Çözüm Değeri	Model 2 Süre (sn)	Model 3 Yöntem 1 Süre (sn)	Model 3 Yöntem 2 Süre (sn)
60_1	3251,12	3251,12	0,2	0,03	0,03
60_2	3202,60	3202,60	0,09	0,09	0,11
60_3	3251,44	3251,44	0,05	0,09	0,06
60_4	3207,28	3207,28	58,47	51,64	99,22
60_5	3246,08	3246,08	0,14	0,59	0,36
60_6	3284,04	3284,04	48,47	43,13	110,54
60_7	3319,20	3319,20	0,05	0,03	0,03
60_8	3248	3248	0,51	0,78	0,34
60_9	3312	3312	0,02	0,01	0,02
60_10	3324,44	3324,44	0,09	0,09	0,09
90_1	4734,12	4734,12	1782	200,52	133,58
90_2	4706,96	4706,96	2000	5,82	9,93
90_3	4792,2	4768,8	2000	200,3	142,1
90_4	4801,36	4801,36	2000	103,13	7,02
90_5	4693,92	4693,92	2000	100,48	0,76
90_6	4719,32	4716,64	2000	101,32	129,41
90_7	4570,12	4569,12	2000	200,39	138,35
90_8	4789,28	4789,28	2000	6,14	2,3
90_9	4853,36	4853,36	2000	121,24	73,96
90_10	4732,88	4732,88	2000	4,78	3,61
120_1	6551,32	6551,32	2000	başarısız	246,12
120_2	6444,36	6444,36	2000	başarısız	başarısız
120_3	6512,04	6512,04	2000	400,46	500,3
120_4	6471,84	6471,84	2000	başarısız	başarısız
120_5	6463,08	6463,08	2000	başarısız	başarısız
120_6	6597	6597	2000	400,46	502,9
120_7	6468,36	6462,32	2000	400,48	446,09
120_8	6465,16	6465,16	101,74	9,38	410,07
120_9	6413,28	6413,28	2000	74,37	51,9
120_10	6461,52	6461,52	2000	başarısız	başarısız

Model 3 için her iki yöntemle ilgili bir diğer husus, λ değerinin belirlenmesidir. Tüm problemler için aynı λ değeri kullanılmıştır. Bu değerin belirlenmesi için örnek bir problem oluşturulmuş ve bu problemi kısa sürede çözebilecek alfa değeri elde edilmiştir. Burada elde edilen parametre değeri diğer problemlerde de kullanılmıştır. λ değerinin belirlenmesi ile ilgili olarak "sub-gradient optimization" yöntemi

uygulanabilir. Bazaraa ve diğ. yöntemin detaylarını anlatmıştır. Ancak, sadece λ değerinin belirlenmesi için harcanacak zamanın fazla olabileceği göz önüne alınarak böyle bir uygulamaya gidilmemiştir.

3. TEK MAKİNA TOPLAM GECİKME ZAMANININ ENAZLANMASI ÇİZELGELEME PROBLEMİ

Tek makine çizelgeleme problemi için toplam gecikme zamanının enazlanması problemi literatürde oldukça sıkça rastlanan bir problemdir.

Tek makine çizelgeleme problemi için toplam gecikme zamanını enazlama problemi Np-Zordur (Due ve Leung, 1990). Lawler bu problem için $O(n^4P)$ zamanda çalışan ayrıştırma bazlı pseudo-polinom bir algoritma geliştirmiştir. Literatürde oldukça sık çalışılan bu problemle ilgili en önemli teorik geliştirmeler Emmons'un baskınlık kurallarına ve Lawler'ın ayrıştırma prensibine dayanır. Yine bu problemle ilgili geliştirilmiş tam ve yakın çözümlerle ilgili araştırma Koulamas tarafından yapılmıştır. Bir başka tarama çalışması Sen ve diğ. tarafından gerçekleştirilmiştir. Son olarak yine Koulamas tarafından bu probleme özel yapılmış çalışmalar derlenmiş ve anlatılmıştır. Problemin zorluk derecesini ifade etmek için gecikme faktörü değeri kullanılır. Bu değer τ ile gösterilir ve tek makine çizelgeleme problemi için ortalama termin zamanı d, toplam proses zamanı P ise $\tau = 1 - d/P$'dir (Sirinivasan, 1971). Problemin birçok özel durumu incelenmiştir. İşlerin süresinin harcanacak adam sayısına bağlı olarak değişebildiği kontrol edilebilir işlem süresi altında tek makine çizelgeleme problemi için toplam gecikme zamanını minimize etmek için karmaşık tamsayılı modeller geliştirilmiş ve uygulanmıştır (Tseng ve diğ., 2009). Yine sıra bağımlı ayar zamanlarının olduğu durumlar için tek makine çizelgeleme probleminde toplam gecikme zamanının en aza indirilmesi ile ilgili yapılan çalışmalarda lagrange gevşetimi ile beraber dinamik programlama adımları uygulanmıştır (Tanaka ve Araki, 2013). İşlerin eşit olmayan hazır olma zamanına sahip olduğu durumlar içinse dal-sınır yöntemleri ve arı algoritması uygulanmıştır (Wu ve diğ., 2013). İşlerin sürelerinin öğrenme etkisiyle zamanla kısaldığı durumlar ise yine modellenmiş ve dal-sınır yöntemiyle ele alınarak çözülmüştür (Yin ve diğ., 2012). Termin zamanlarının eşit olduğu durumlar ise ortak termin problemleri olarak ele alınmıştır, tek makine çizelgeleme problemi için ortak termin zamanı olduğu durumda problemin optimum çözümü SPT kuralıyla elde edilmektedir.

3.1. Çözüm Yöntemleri

Tek makine çizelgeleme problemi için toplam gecikme zamanının enazlanması problemi ile ilgili yöntemler tam çözümü garanti edenler ve garanti etmeyenler şeklinde ikiye ayrılır. Tam çözümü garanti eden dinamik programlama ve matematiksel programlama yöntemleri varken, optimum çözümü garanti etmeyen yöntemler sezgiseller ve metasezgisellerdir. Aşağıda bu probleme özel geliştirilmiş yöntemler anlatılacaktır.

3.1.1. Dinamik programlama ile çözüm yöntemi

$\sum T_j$ nin enazlanması problemi oldukça sık çalışılmış bir problemdir. Geciken işlerin sayısının enazlanması problemi $\sum U_j$ her ne kadar bir anlam ifade etse de pratikte terminlerin nasıl karşılandığı ile ilgili tek bir ölçü olamaz. Geciken işlerin sayısını minimuma indirebilmek için bazı işler kabul edilemeyecek seviyede gecikebilir. Eğer toplam gecikme zamanları minimize edilirse, herhangi bir işin kabul edilemeyecek kadar fazla gecikmesi daha az olur (Pinedo, 2008).

$1 \parallel \sum T_j$ modelinin uzun yıllar kompleksliği açık bir konu olarak kalmıştır. 1990 yılında problemin NP-Zor olduğu ispatlanmıştır. Problem NP-Zor olduğu için dinamik programlama ile optimum çözüm elde edilebilir. Bunun için kullanılan iki temel kural (Emmons'un kuralları) vardır.

Kural 1:

Eğer $p_j \leq p_k$ ve $d_j \leq d_k$ ise, j işinin k işinden önce çizelgelendiği en az bir optimum sıralama vardır.

Böyle bir sonuç Np-Zor olan bir problem için algoritma geliştirebilmek adına oldukça kullanışlıdır. Böylece, birçok sıralamanın elenmesi sözkonusudur. Böyle bir baskınlık kuralı işlerin öncelik kısıtlarının belirlenmesi için kullanılabilir.

Kural 2:

İki problem kümesi düşünelim. İkisinin de n tane işi olsun ve işlem süreleri $p_1, \ldots, p_n$ olsun. Birinci problem kümesi için termin süreleri $d_1,\ldots,d_n$ olsun. C'_k k işinin herhangi bir optimum sıralama S′ için, en geç bitebileceği zaman olsun. İkinci

problemin termin zamanları $d_1,\ldots,d_{k-1}$, $\max(d_k, C_k^{'})$, $d_{k+1},\ldots,d_n$ olsun ve S″ bu ikinci problem için optimum sıralama olsun, $C_j^{''}$ j işi için bitim zamanını göstersin.

İkinci problem için optimum sonucu veren herhangi bir sıralama aynı zamanda ilk problem için de optimum sıralamayı verir.

İspat:

V′ (S) herhangi bir S sıralaması için birinci problemin toplam gecikmesi olsun.

V″ (S) herhangi bir S sıralaması için ikinci problemin toplam gecikmesi olsun.

Eğer $C_k^{'} \geq d_k$ ise

$V'(S') = V''(S') + A_k$

ve

$V'(S'') = V''(S'') + B_k$ dir.

$A_k = C_k^{'} - d_k$

$B_k = \max(0, \min(C_k^{''}, C_k^{'}) - d_k)$

Görüleceği üzere $A_k \geq B_k$ dır. S″ ikinci problem için optimum olduğuna göre $V''(S') \geq V''(S'')$ 'dir . Böylece $V'(S') \geq V'(S'')$ ispatlanmış olur.

Eğer $C_k^{'} < d_k$ ise her iki problem içinde terminler aynı olacağından ikinci problem için optimum sıralama, birincisi için de optimumdur (Pinedo, 2008).

Dinamik programlama için Kural 2'den yararlanılarak, şöyle bir sonuç çıkarılabilir.

$d_1 \leq d_2 \leq d_3 \leq \ldots \leq d_n$ olsun ve k işi işlem zamanı en uzun iş olsun.

Öyle bir δ tam sayısı vardır ki, $0 \leq \delta \leq n-k$ dir ve S optimum sıralaması için k işinden önce gelen j işleri için $j \leq k + \delta$ 'dır. k işinden sonra gelen j işleri içinse $j > k + \delta$'dır.

Dinamik programlama algoritmasında 1,…,l işleri için t zamanından itibaren başlayan optimum çözümü veren bir alt algoritma gerekir. Diyelim ki, k bu l iş

arasında en uzun işlem süresine sahip iş olsun. Yukarıda bahsedilen çıkarıma dayanarak bir δ değeri için ($0 \leq \delta \leq l-k$) t zamanında başlayan bir optimum sıralama vardır ve bu sıralama alttaki gibi üç bölüme ayrılabilir.

1. 1,2,…,k-1, k+1,..,k+ δ işleri için herhangi bir sıralama
2. k işi
3. k+ δ+1, k+ δ+2,…,l işleri için herhangi bir sıralama

k işinin tamamlanma zamanı, $C_k(\delta) = \sum_{j \leq k+\delta} p_j$'dir.

Toplam sıralamanın optimum olabilmesi için ilk ve üçüncü gruptaki işlerin de optimum sıralanması gerekir. Böylece herhangi bir iş kümesi için, alt kümelerinin optimum sıralamasından yola çıkarak büyük kümelerin optimum sıralamasını bulabileceğimiz bir dinamik programlama prosedürü düşünülebilir.

V(J, t) J alt kümesindeki işlerin t zamanından itibaren çizelgelendiğinde durumdaki minimum gecikme zamanı olsun.

Dinamik Programlama Prosedürü

Başlangıç şartları:

$V(\emptyset,t)=0,$

$V(\{j\},t) = \max(0,t+p_j-d_j).$

Döngü:

$V(J(j,l,k),t) = \min_{\delta}(V(J(j,k'+\delta,k'),t)+ \max(0,C_{k'}(\delta)-d_{k'}+ V(J(k' + \delta + 1, l, k'), C_{k'}(\delta)))$

k' değeri için

$p_{k'} = \max (p_{j'} \mid j' \in J(j,1,k)).$

Optimum Değer Fonksiyonu:

$V(\{1,\ldots,n\},0).$

$\sum T_j$ değeri V({1,…,n},0) değeri olarak hesaplanabilir. Algoritmanın komplekslik değeri hesaplanırken, J(j,l,k) alt kümeleri için $O(n^3)$dir. t zamanı için $\sum p_j$ nokta olduğundan her bir döngünün kompleksliği $O(n^3 \sum p_j)$ dir. Toplam n kere bu işlem yapılacağından, dinamik programlama algoritmasının toplam kompleksliği $O(n^4 \sum p_j)$ dir. $\sum p_j$ ifadesinden dolayı algoritmanın pseudo-polinom zamanlı olduğu söylenebilir.

3.1.2. Karmaşık tamsayılı programlama ile çözüm yöntemleri

Tek makine çizelgeleme problemi için toplam gecikme zamanını enazlama problemi için karmaşık tamsayılı programlama ile çözüm yöntemleri geliştirilmiştir. Bu yöntemler 4 ana gruba ayrılır. Bunlar aralık endeksli formülasyon yöntemi (IIF), zaman endeksli formulasyon yöntemi (TIS), linear sıralama yöntemi (LO) ve pozisyon formülasyonu (POS) yöntemidir (Sadykov,2006).

3.1.2.1. Zaman endeksli formülasyon yöntemi (TIS)

$\min \sum_{t=1}^{P} \max\left\{0, t+p_j-1-d_j\right\} x_{jt}$

Kısıtlar:

$$\sum_{t=1}^{P-p_j+1} x_{jt}=1, \qquad j \in N \tag{3.1}$$

$$\sum_{j=1}^{n} x_{j1}=1, \tag{3.2}$$

$$\sum_{j=t}^{n} x_{jt} - \sum_{j\in N: p_j<t} x_{j,t-p_j} =0 \qquad t \in [2,P] \tag{3.3}$$

x_{jt} ikili değişkeninin anlamı eğer j işi t zamanında başladıysa 1 değilse 0'dır. P zamanı t değerinin alabileceği en büyük değerdir. Denklem (3.3) kısıtı işlerin çakışmasını engellemek içindir.

3.1.2.2. Lineer sıralama yöntemi (LO)

$\min \sum_{j=1}^{n} w_j T_j$

Kısıtlar:

$$\delta_{ij}+\delta_{ji}\leq 1 \qquad i, j \in N_i < j \qquad (3.4)$$

$$\delta_{ij}+\delta_{jk}+\delta_{ki}\leq 2 \qquad i,j,k\in N i\neq j\neq k \qquad (3.5)$$

$$T_j \geq \sum_{i\in N, i\neq j} p_i \delta_{ij}+p_j-d_j \qquad j \in [N] \qquad (3.6)$$

$$\delta_{ij}\in \{0,1\} \qquad i,j\in N \quad i \neq j \qquad (3.7)$$

Her bir ikili iş için biri diğerinden önce çizelgelenmelidir. Bu şekliyle δ_{ij} ikili değişkeni i işi j işinden önce çizelgelendiyse 1 değilse 0'dır. Denklem (3.5) kısıtı, i işi j işinden önce çizelgelenmiş ve j işi de k işinden önce çizelgelenmişse i işi k işinden önce çizelgeleneceği bilindiği için eklenmiştir.

3.1.2.3. Pozisyon formülasyonu yöntemi (POS)

$$\min \sum_{j=1}^{N} w_j T_j$$

Kısıtlar:

$$\sum_{j\in N} x_j^k = 1 \qquad k\in N \qquad (3.8)$$

$$\sum_{k\in N} x_j^k = 1 \qquad j\in N \qquad (3.9)$$

$$\gamma_k = \gamma_{k-1} + \sum_{j\in N} p_j x_j^k \qquad k\in N\backslash\{1\} \qquad (3.10)$$

$$\gamma_1 = \sum_{j\in N} p_j x_j^1 \qquad (3.11)$$

$$T_j \geq \gamma_k - U.\left(1-x_j^k\right)-d_j \qquad j,k\in N \qquad (3.12)$$

$$T_j \geq 0 \qquad j\in N \qquad (3.13)$$

$$x_i^k \in \{0,1\} \qquad i,k\in N \qquad (3.14)$$

Bu yöntemde ise i işi k pozisyonunda çizelgelenmişse x_i^k değeri 1, değilse 0'dır. Denklem (3.8) ve (3.9) kısıtları aynı pozisyona birden fazla iş atanmaması içindir. U büyük bir sayıyı temsil eder.

3.1.2.4. Aralık endeksli formülasyon yöntemi (IIF)

Bu yöntem için zaman dilimlere bölünmüştür ve ikili değişkenler işlerin bu zaman dilimlerine atanması şeklinde tanımlanmıştır. Zaman dilimleri oluşturulurken oluşturulan dilim içinde biten işlerin optimum sıralaması bilinecek şekilde minimum sayıda dilim oluşturulur. Zaman endeksli formülasyon yöntemine benzer şekilde işlerin zamanı yerine dilimi belirlenir ve böylece optimum sıralama elde edilir. TIS yöntemine göre IIF yöntemi daha az ikili değişken içerdiğinden performansı daha iyidir.

Bu dört yöntem birbiriyle kıyaslandığında IIF ve LO yöntemlerinin daha başarılı olduğu söylenebilir (Sadykov, 2006). Bu tez çalışmasında toplam gecikme zamanının minimizasyonu problemi için, önerilen yöntemin optimumla kıyaslanması için tam sonucu garanti eden bir yöntem kullanılması gerektiğinde kullanılan modelleme, LO yöntemi baz alınarak geliştirilmiştir.

3.1.3. Yaklaşık sonuçlu çözüm yöntemleri

Tek makine çizelgeleme problemi için toplam gecikme zamanı minimizasyonu problemi NP-Zor oluduğu için probleme özel birçok sezgisel yöntem geliştirilmiştir. Bunlardan bazıları basit sıralama yöntemleri iken, bazıları iterasyon gerektiren yöntemlerdir. Basit sıralama yöntemlerine örnek olarak EDD (En erken termin tarihli en önce) ya da Baker ve Bertrand tarafından önerilen MDD (Değiştirilmiş termin tarihlerine göre sıralama) verilebilir. MDD çizelgeleme yönteminde t 0'dan başlar ve mevcut çizelgelenmemiş işler içinden $d_j' = \max\{pj, dj-t\}$ değeri en küçük olan seçilir ve çizelgenin sonuna yerleştirilir, t değeri çizelgelenen işlerin bitim zamanı olarak güncellenir ve bu işlem tüm işler çizelgelenene kadar devam ettirilir. İterasyona dayanan yöntemlere örnek olarak PSK (Panwalkar ve diğ. 1993) veya NBR (Holsenback ve Russell, 1992) yöntemleri örnek gösterilebilir. Yaklaşık sonuçlu çözüm yöntemlerine örnek olarak aşağıda iki tanesi anlatılmıştır.

3.1.3.1. Değiştirilmiş Lawler algoritması

1. Termin tarihlerini değiştir. $d_i^{'}=\max\{p_i,d_i\}$ $\quad 1\leq i \leq n$
2. $k = n$, $S=\{J_1,J_2,J_3,..,J_n\}$ $\quad P =\sum p_i$

3. max(0,P-d_i') değeri minimum olan j işini bul ve k sırasına j işini yerleştir.
4. k = k-1; S= S \ {J_j} ; P= P - p_j
5. k=0'sa dur. Değilse adım 3'e dön.

Bu algoritmanın kompleksliği O(n^2)'dir. Bu algoritmanın çözümü en kötü ihtimalle optimumdan (n-1) katı uzaklıktadır (Cheng ve diğ., 2005)

3.1.3.2. Tam polinom zamanlı yaklaşık düzen yöntemi

Yaklaşık düzen yöntemiyle elde edilen sıralama A ise

$\sum T_j(A) \leq (1+\epsilon) \sum T_j(OPT)$ 'dir. Yöntemin adımları aşağıda anlatılmıştır.

1. İşleri termin zamanına göre sırala,

Eğer $T_{max}= 0$, EDD optimum'dur. DUR

Değilse, $K= \left(\frac{2\epsilon}{n(n+1)}\right) T_{max}(EDD)$

2. $p_j^{'}= \left\lfloor p_j/K \right\rfloor$,

$d_j^{'}= d_j/K$

Bölüm 3.1.1'de anlatılan dinamik programlama prosedürünü uygula.

Bu şekilde elde edilen S sıralaması için elde edilen toplam gecikme zamanı optimum sıralamanın toplam gecikmesinin (1+ϵ) katından daha küçüktür.

3.1.4. Metasezgisel çözüm yöntemleri

Problemin çözümünde kullanılan birçok metasezgisel yöntem geliştirilmiştir. Bunlardan tavlama benzetimi kullanılarak yapılan çalışmalarda mevcut sezgisellerden elde edilen iyi bir çözümle başlanarak daha iyi çözümler hedeflenmiştir (Potts ve Wassenhove, 1991). Tabu arama, tavlama benzetimi, genetik algoritma yöntemlerinin kıyaslamasının yapıldığı bir çalışmada ise tabu arama yönteminin daha başarılı olduğu söylenmiştir (Crauwels ve diğ., 1998). Lokal arama tekniklerinin dinamik programlama ile beraber kullanıldığı bir çalışmada literatürde

incelenen problemler için daha önce elde edilen metasezgisel yöntemlere kıyasla daha iyi sonuçlar elde edilmiştir (Congram ve diğ., 2002). Tabu arama yönteminin dinamik programlama ile beraber kullanıldığı bir çalışmada ise elde edilen sonuçların daha öncekilere göre daha iyi sonuç verdiği görülmüştür (Bilge ve diğ., 2007).

3.2. Önerilen Yöntem

Bizim ele aldığımız ana problem için her bir operasyonun işlem zamanı birbirine oldukça yakındır. Eğer, tüm işlem zamanları birbirine eşit olsaydı, toplam gecikme zamanını enazlama probleminin aslında atama problemine dönüşeceği görülebilir.

Teorem 3.1:

Tek makine çizelgeleme problemi için, eğer tüm işlem zamanları birbirine eşitse, toplam gecikme zamanını enazlama problemi atama problemine döner.

İspat:

Tanımlamalar:

X_{ij} : i işi jsırasında çizelgelendiyse 1, değilse 0'dır
P_i=P : i işinin işlem zamanı
C_i : i işinin bitim zamanı
D_i : i işinin termin zamanı
T_i : i işinin gecikme zamanı =max(0,C_i-D_i)

Dolayısıyla;

$C_i = P.j$

A_{ij} =i işi j sırasında çizelgelendiyse toplam gecikme zamanı = X_{ij}.max(0, P.j-D_i)

Denklemin ikinci tarafındaki max(0, P.j-D_i) kısmı i'ye j'ye bağlı olduğundan B_{ij} olarak adlandırılacaktır.

Min $\sum_{j=1}^{n} \sum_{i=1}^{n} X_{ij} B_{ij}$

Kısıtlar:

$\sum_{j=1}^{n} x_{ij}=1$

$\sum_{i=1}^{n} x_{ij}=1$

$X_{ij}=0$ veya 1

Hedef fonksiyon kısmında belirtilen ifade tek makine çizelgeleme problemi için toplam gecikme zamanını ifade eder. Bu şekliyle bu problem atama problemidir ve polinom zamanda çözülebilir.

Aynı zamanda yukarıda bahsedilen kural 1 gereğince "Eğer $p_j \leq p_k$ ve $d_j \leq d_k$ ise, j işinin k işinden önce çizelgelendiği en az bir optimum sıralama vardır.". Eğer tüm işlem zamanları birbirine eşitse en erken termin zamanına göre sıralama (EDD) optimum çözümü verir.

3.2.1. Atama problemi

Atama problemi taşıma problemlerinin özel bir durumudur. Taşıma problemlerini çözmek için kullanılan yöntemler aynı zamanda atama problemleri için de kullanılabilir. Ancak atama probleminin özel durumundan dolayı çok daha özel yapıda algoritmalar geliştirilebilir. Matris yapıda atama problemi aşağıdaki şekilde gösterilebilir.

Min **c**x

s.t

Ax = 1

$x_{ij}=0$ veya 1 i,j=1,….,m

Burada A matrisinin toplam unimodular özelliği sayesinde en iyi temel olurlu çözüm için $x_{ij} = 0$ veya 1 yerine $x_{ij} \geq 0$ konulursa sonucun tamsayı olacağı söylenebilir (Bazaraa ve Jarvis, 1977). Dolayısıyla problem bir MIP (karmaşık tamsayılı matematiksel model) değil, bir doğrusal programlama modelidir. Doğrusal modellerin polinom zamanda çözülebildiği bilinmektedir. Bu probleme özel, "Macar

yöntemi" olarak adlandırılan yöntemle problemin polinom zamanda çözümü mümkündür.

3.2.2. Önerilen model

Yapılan çalışmalarda, tek makina çizelgeleme problemi için toplam gecikme zamanının enazlanması ile ilgili özel durumlar incelenmiştir. Genel olarak problem NP-Zor olmasına rağmen bazı özel durumlar için polinom zamanda çözülebilen durumlar olabilir. Lazarev ve Werner böyle bir çalışma gerçekleştirmiş ve problemin özel bir durumu ile ilgilenmiştir. $p_1 \geq p_2 \ldots \geq p_n$ ve $d_1 \leq d_2 \ldots \leq p_n$ olan problemler için bile problemin Np-Zor olduğunu ispatladıktan sonra bu tip özel durumdaki problemler için polinom zamanda iyi sonuç verecek bir yöntem geliştirmişlerdir. Gafarov ve diğ. ise yaptıkları çalışmada yine problemin özel bir durumu için pseudo-polinom zamanda en iyi çözümü sağlayan bir yöntem geliştirmişlerdir.

Benzer şekilde, sürelerin eşit olması durumunda problemin atama problemine dönmesi fikrinden yola çıkarak, sürelerin birbirlerine yakın olduğu durumlarda yukarıda bahsedilen özellikten (Teorem 3.1) yararlanarak toplam gecikme zamanını enazlama problemi atama problemi olarak ele alınabilir. Bir işin atandığı sıradaki atama maliyetinin doğruya yakın bir şekilde hesaplanması gerekmektedir. Bu atama maliyeti, beklenen atama maliyeti olarak altta bahsi geçen yöntemle hesaplanmıştır.

Örneğin a işi 5. sıraya atanacaksa, bu atamanın maliyeti;

max(0, a işi dışındaki işlerin ortalama işlem zamanı . 4 + a işinin proses zamanı– a işinin termin zamanı) olacak şekilde hesaplanabilir. Bu değer a işinin 5. sıraya atanması durumundaki gerçek maliyetin beklenen değeridir.

$$B_{ij} = \max\left(0, \frac{\sum_{k \in J-\{i\}} p_k}{s[J]-1}.(j-1)+p_i-d_i\right)$$

Bu şekilde tüm maliyetler hesaplanıp, atama probleminin çözümünden elde edilen sıralama ana problem için kullanılmıştır.

Bu şekilde hesaplanan beklenen değeri daha da iyileştirmek için şöyle bir yöntem izlenebilir. Dinamik programlamada kullanılan Kural 1'de bahsedilen, "eğer $p_j \leq p_k$

ve $d_j \leq d_k$ ise, j işinin k işinden önce çizelgelendiği en az bir optimum sıralama vardır." özelliği sayesinde beklenen değer alttaki gibi hesaplanabilir.

Herhangi bir i işi için işlem zamanı i işinden daha uzun ve termin zamanı i işinden daha büyük işlerin kümesi D ise:

$D = D \cup \{i\}$

$B_{ij} = \max(0, \frac{\sum_{k \in J \setminus D} p_k}{s[J \setminus D]}.(j-1)+p_i-d_i)$ olarak alınacaktır.

B_{ij} değeri i işi j sırasında çizelgelenirse oluşan beklenen gecikme değeridir. Atama maliyetlerini yukarıdaki şekilde hesaplayarak çalışan modele ait Opl kodu Ek-D'de gösterilmiştir. Tek makine çizelgeleme problemi için toplam gecikme zamanının minimizasyonu için önerilen bu model bundan sonra "tek makine toplam gecikme zamanı minimizasyonu" TMTGZM olarak adlandırılacaktır.

3.2.3. Problemlerin belirlenen yöntem ile çözümü

Yukarıda da bahsedildiği gibi tüm işlem zamanlarının eşit olduğu bir problem için problem atama problemine benzetilebilir. Ele alınan problemler için termin zamanı 2.2.8 bölümünde de bahsedildiği gibi gerçek hayatta yaşanan problemlerin benzerini oluşturmak amacıyla her bir iş için $[C_{max}/2, 2.C_{max}]$ arasında uniform dağıtılmıştır. Bölüm 2'de çözdürülen 30 problem için, toplam gecikme zamanını enazlama hedefi ele alındığında elde edilen sonuçlar Tablo 3.1'de verilmiştir. Burada elde edilen sonuçlara göre termin kısıtı çok sıkı olmadığı için tüm problemler için optimum değerin sıfır olduğu söylenebilir. Optimum değer sıfırsa EDD yönteminin uygulanması durumunda sonucun sıfır olacağı söylenebilir (Pinedo, 2008). Optimum değer toplam gecikmenin sıfır olduğu durumda sıfırdır. Ancak, çözüm süresinin getireceği avantajı görmek adına toplam gecikmeyi minimize eden model altta belirtildiği şekilde geliştirilmiş ve kodlanmıştır (Model 5).

S_i : i işinin başlama zamanı

P_i : i işinin proses zamanı

X_{ik} : ikili değişken, eğer i işi k işinden önce çizelgelenmişse 1 değilse 0'dır.

R : oldukça büyük bir sayı

D_i : i işinin termin zamanı

F_i : i işinin gecikme miktarı = $\max\{0, S_i+P_i-D_i\}$

Min $\sum_{i=1}^{n} F_i$

Kısıtlar:

$S_i+P_i \leq S_k + R(1-X_{ik})$ i=1,...,n-1 k=i+1,…,n (3.15)

$S_k+P_k \leq Si + R \cdot X_{ik}$ i=1,...,n-1 k=i+1,…,n (3.16)

$S_i \geq 0$ i=1,…,n (3.17)

$X_{ik} \in \{0,1\}$ i=1,...,n-1 k=i+1,…,n (3.18)

$S_i+P_i \leq \sum_{i=1}^{n} P_i=C_{max}$ i=1,…,n (3.19)

$F_i \geq S_i+P_i-D_i$ i=1,…,n (3.20)

Her iki yöntem kullanılarak elde edilen çözümlerin kıyaslaması Tablo 3.1'de görülebilir. Atama problemi olarak çözdürülen problemlerin amaç fonksiyonundaki değer oluşan çizelgenin gerçek toplam gecikme zamanı değeri değildir. Toplam gecikme zamanı değeri, problemin çözümü ile oluşan sıralamanın gerçek başlama zamanlarına göre oluşan gerçek gecikmelerine göre hesaplanmıştır. Çözdürülen 30 problem için de elde edilen amaç fonksiyonu değeri ve gerçek toplam gecikme değeri sıfırdır. Problemler OPL 6.3 kullanılarak modellenmiş ve cplex çözüm motoru kullanılarak 2.1 GHz işlemcili bir bilgisayarda çözdürülmüştür. Tabloda da görüleceği üzere en iyi gecikme değerini garanti eden yöntemle 1 saniyeden daha kısa bir sürede en iyi çözüm elde edilmiştir. 90 işlik problemler içinse önerilen yöntemle yine 1 saniyeden daha kısa bir sürede çözülen problemler, optimum değeri garantileyen modelle çözülmek istenildiğinde çözün süresi minumum 22 saniye sürmektedir. Bazı problemler içinse verilen 5 dakika zaman limiti aşıldığı için çözüm elde edilememiştir. 120 işlik problemlerde ise, optimum çözümü garantileyen model hiçbir problem için istenilen süre zarfında çözüm sağlayamamış, ancak önerilen yöntemle yine 1 saniyeden kısa bir sürede optimum çözüm elde edilmiştir. Ele alınan problemler için gecikme faktörü olan τ değeri -0.25'dir. Bu değer oldukça düşüktür

ve önerilen yöntemin performansını değerlendirmek için yeterli değildir. Önerilen yöntemin performansını daha iyi değerlendirebilmek için problemin termin tarihleri değiştirilmiştir.

Tablo 3.1. Problemlerin toplam gecikme zamanı için önerilen yöntemle çözümü

	Önerilen Yöntem			Model 5	
	Amaç fonksiyon değeri (Toplam beklenen gecikme değeri)	Toplam Gecikme Zamanı	Süre (sn)	Toplam Gecikme Zamanı	Süre (sn)
e_60_1	0	0	0,21	0	30
e_60_2	0	0	0,23	0	30
e_60_3	0	0	0,23	0	101
e_60_4	0	0	0,29	0	3
e_60_5	0	0	0,29	0	22,3
e_60_6	0	0	0,23	0	23,27
e_60_7	0	0	0,29	0	48,98
e_60_8	0	0	0,28	0	42,35
e_60_9	0	0	0,21	0	8,65
e_60_10	0	0	0,28	0	9,5
e_90_1	0	0	0,48	0	115,62
e_90_2	0	0	0,49	0	27,12
e_90_3	0	0	0,4	0	22,68
e_90_4	0	0	0,42	NA	>5 dakika
e_90_5	0	0	0,50	0	106,04
e_90_6	0	0	0,43	0	112,32
e_90_7	0	0	0,42	0	107,12
e_90_8	0	0	0,53	0	146,53
e_90_9	0	0	0,43	NA	>5 dakika
e_90_10	0	0	0,45	0	129,18
e_120_1	0	0	0,70	NA	>5 dakika
e_120_2	0	0	0,74	NA	>5 dakika
e_120_3	0	0	0,81	NA	>5 dakika
e_120_4	0	0	0,64	NA	>5 dakika
e_120_5	0	0	0,76	NA	>5 dakika
e_120_6	0	0	0,68	NA	>5 dakika
e_120_7	0	0	0,81	NA	>5 dakika
e_120_8	0	0	0,70	NA	>5 dakika
e_120_9	0	0	0,87	NA	>5 dakika
e_120_10	0	0	0,76	NA	>5 dakika

Önerilen yöntemin daha sıkı terminli problemler için de iyi sonuçlar verdiğinin doğrulanması için eldeki 30 problem için termin tarihleri sıkılaştırılarak,

$d_i^{'}$=yuvarla(d_i/3) olacak şekilde değiştirilmiştir. Bu şekilde elde edilen problemler için gecikme faktörü değeri 0,58'dir. Bu problemler yine optimumu garanti eden modelle kıyaslanmaya çalışılmıştır. Ancak, Tablo 3.2'de de görüleceği üzere model hiçbir problemi istenilen zaman kısıtında çözememiştir.

Tablo 3.2. Gecikme faktörü 0,58 olan problemlerin önerilen yöntemle çözümü

	Önerilen Yöntem				Model 5	
	Amaç fonksiyon değeri (Toplam beklenen gecikme değeri)(A)	Toplam Gecikme Zamanı (B)	(B-A)/A	Süre (sn)	Toplam Gecikme Zamanı	Süre (sn)
e_60_1 /3	7627,63	8116	%6,4	0,21	NA	>5 dakika
e_60_2 /3	11488	12198	%6,2	0,23	NA	>5 dakika
e_60_3 /3	10235	11232	%9,7	0,27	NA	>5 dakika
e_60_4 /3	9032,14	9209	%1,9	0,31	NA	>5 dakika
e_60_5 /3	9684,59	10679	%10,2	0,31	NA	>5 dakika
e_60_6 /3	12000,09	12701	%5,8	0,35	NA	>5 dakika
e_60_7 /3	9220,62	9560	%3,7	0,26	NA	>5 dakika
e_60_8 /3	10853,35	11890	%9,5	0,23	NA	>5 dakika
e_60_9 /3	8512,02	9259	%8,8	0,28	NA	>5 dakika
e_60_10 /3	11254,48	12414	%10,3	0,23	NA	>5 dakika
e_90_1 /3	20429,11	22121	%8,3	0,43	NA	>5 dakika
e_90_2 /3	19345,69	20567	%6,3	0,54	NA	>5 dakika
e_90_3 /3	22910,44	24362	%6,3	0,48	NA	>5 dakika
e_90_4 /3	18939,48	19783	%4,4	0,43	NA	>5 dakika
e_90_5 /3	21696,35	23291	%7,3	0,59	NA	>5 dakika
e_90_6 /3	21655,87	22819	%5,3	0,57	NA	>5 dakika
e_90_7 /3	19294,89	19366	%0,4	0,49	NA	>5 dakika
e_90_8 /3	21310,06	22775	%6,8	0,43	NA	>5 dakika
e_90_9 /3	22438	24238	%8,0	0,43	NA	>5 dakika
e_90_10 /3	22222,05	24155	%8,7	0,54	NA	>5 dakika
e_120_1 /3	37688,18	40280	%6,9	0,76	NA	>5 dakika
e_120_2 /3	35696,20	37466	%5,0	0,90	NA	>5 dakika
e_120_3 /3	36069,22	39857	%10,5	0,84	NA	>5 dakika
e_120_4 /3	40893,96	44303	%8,3	0,78	NA	>5 dakika
e_120_5 /3	43789,72	45739	%4,4	0,78	NA	>5 dakika
e_120_6 /3	36196,35	39391	%8,8	0,78	NA	>5 dakika
e_120_7 /3	35857,87	37553	%4,7	0,78	NA	>5 dakika
e_120_8 /3	34819,86	37911	%8,9	0,78	NA	>5 dakika
e_120_9 /3	36165,89	38684	%6,9	0,81	NA	>5 dakika
e_120_10/3	41700,08	45374	%8,8	0,96	NA	>5 dakika

Tablo 3.2'de de görüleceği üzere önerilen yöntemle hesaplanan "beklenen toplam gecikme zamanı" değeriyle oluşan sıralamanın "gerçek toplam gecikme zamanı" değeri kıyaslandığında beklenen değerin gerçekten en fazla yaklaşık %10 saptığı, bazı durumlarda ise oldukça gerçeğe yakın bir gecikme zamanı hesapladığı görülmektedir.

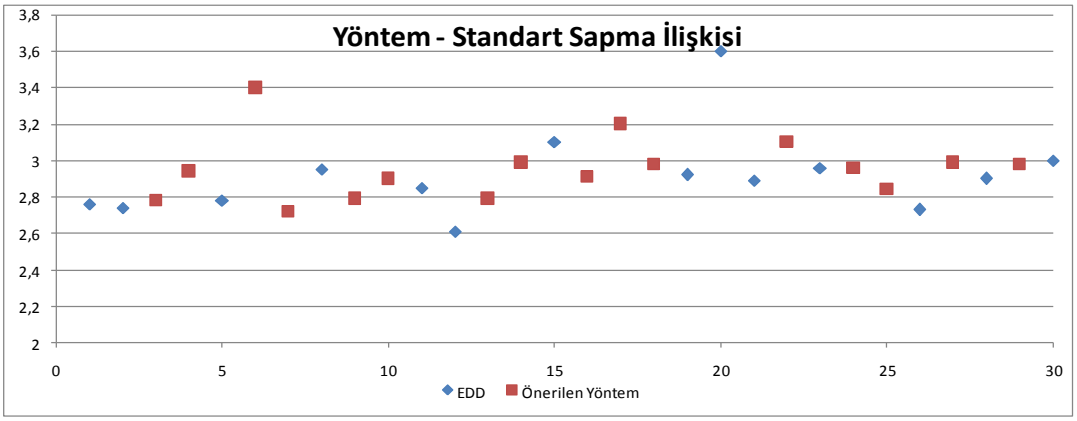
Tablo 3.3. Problemlerin EDD ile kıyaslanması

Problem Adı	Önerilen Yöntem (A)	EDD (B)	% Fark (B-A) /B
e_60_1 /3	8116	7721	5,1%
e_60_2 /3	12198	11877	2,7%
e_60_3 /3	11232	11428	-1,7%
e_60_4 /3	9209	9499	-3,1%
e_60_5 /3	10679	10321	3,5%
e_60_6 /3	12701	13102	-3,1%
e_60_7 /3	9560	9842	-2,9%
e_60_8 /3	11890	11444	3,9%
e_60_9 /3	9259	9264	-0,1%
e_60_10 /3	12414	12501	-0,7%
e_90_1 /3	22121	21638	2,2%
e_90_2 /3	20567	20354	1,0%
e_90_3 /3	24362	24607	-1,0%
e_90_4 /3	19783	20727	-4,6%
e_90_5 /3	23291	23195	0,4%
e_90_6 /3	22819	23367	-2,3%
e_90_7 /3	19366	20385	-5,0%
e_90_8 /3	22775	22978	-0,9%
e_90_9 /3	24238	23468	3,3%
e_90_10 /3	24155	23194	4,1%
e_120_1 /3	40280	40001	0,7%
e_120_2 /3	37466	38414	-2,5%
e_120_3 /3	39857	39540	0,8%
e_120_4 /3	44303	45093	-1,8%
e_120_5 /3	45739	46522	-1,7%
e_120_6 /3	39391	37959	3,8%
e_120_7 /3	37553	38643	-2,8%
e_120_8 /3	37911	36872	2,8%
e_120_9 /3	38684	38840	-0,4%
e_120_10/3	45374	44266	2,5%

Elde edilen çözümlerin referans bir modelle kıyaslanabilmesi için yukarıda bahsedilen problemler en erken termin süresine göre sıralanarak EDD yöntemiyle çözdürülmüştür. Elde edilen sonuçlar Tablo 3.3'de gösterilmiştir.

Tablo 3.3'de de görüleceği üzere elde edilen sonuçlara göre toplam 30 problemin 16'sında önerilen yöntem EDD'den daha iyi sonuç vermiştir. Tabloda ayrıca önerilen yöntemin EDD'den ne kadar daha iyi sonuç verdiği yüzdesel olarak da verilmiştir. % fark değeri negatif olanlar önerilen yöntemin daha iyi sonuç verdiği problemlerdir. Çok kısa zamanda EDD'den daha iyi sonuç elde edilebilmesi, metasezgisel yöntemlerde ilk çözüm elde etme ve kıyaslama yapılacak referans çözüm elde etme konusunda ve daha kısa zamanda daha iyi çözümler elde etme konusunda yardımcı olabilir.

EDD'ye göre daha iyi sonuç elde edilen problemlerin karakteristikleri de incelenmiştir. Yukarıda da bahsedildiği üzere önerilen yöntemin işlem zamanları birbirine yakın olan problemler için daha iyi sonuç verebileceği belirtilmiştir. EDD'nin daha iyi sonuç verdiği ve önerilen yöntemin daha iyi sonuç verdiği problemlere bakıldığında problem kümesindeki işlerin işlem zamanlarının standart sapmaları kıyaslanabilir. Bu şekilde yapılan bir kıyaslama Şekil 3.1'de gösterilmiştir. Grafikten çıkan sonuç standart sapmanın birbirine çok yakın olması nedeniyle yapılan kıyaslamanın anlamlı olmadığı yönündedir. İleride yapılacak çalışmalarda standart sapmaları arasında belirgin farklar olan problemler için kıyaslama yapılabilir.



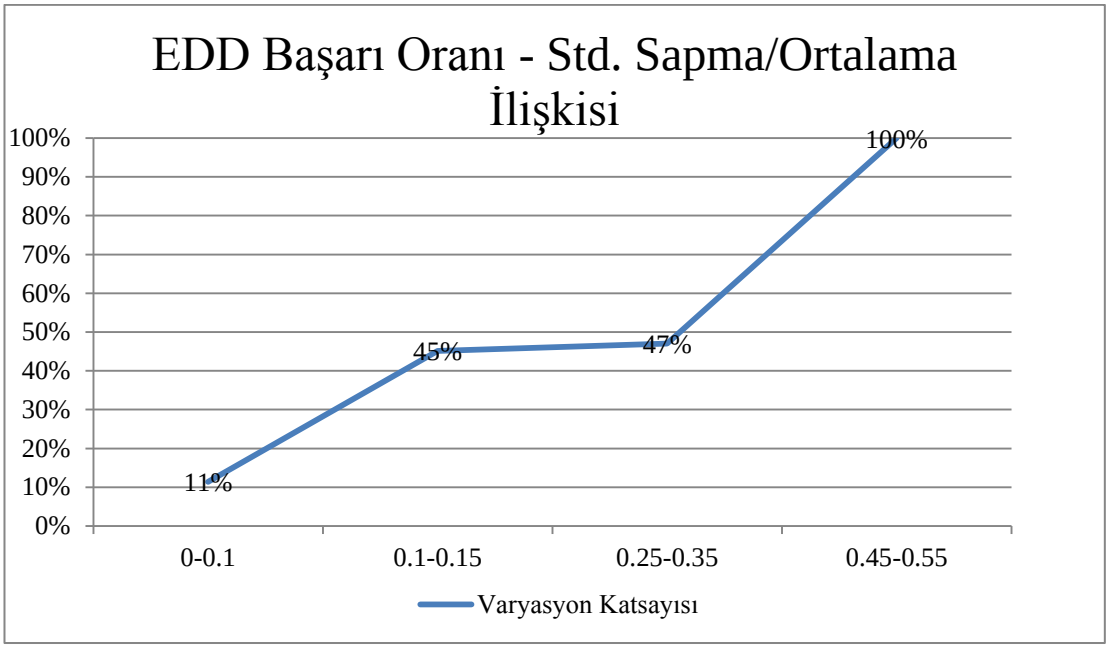
Şekil 3.1. Problemlerin standart sapma değerleriyle iyi sonuç verdiği yöntem ilişkisi

Şekil 3.1'deki grafikte her bir problem için daha iyi sonuç veren yöntem işaretlenmiştir. Örneğin ilk problem için EDD daha iyi sonuç verirken, üçüncü problem için önerilen yeni yöntem daha iyi sonuç vermiştir. Bu problemlere ait standart sapma değerleri grafiğin y ekseninde gösterilmiştir. Görüleceği üzere standart sapmalar birbirine çok yakın olduğu için bu grafik üzerinden bir yorum yapmak hatalı olabilir.

Önerilen yöntemin hangi durumlarda iyi sonuç verebildiği anlamak için eldeki termin zamanları kısaltılmış 30 problem için ortalama işlem zamanı ve dolayısıyla toplam işlem zamanı aynı kalmak koşuluyla standart sapma değerlerini artırarak ya da azaltarak 30 problem daha elde edilmiştir. Standart sapmayı artırmak için eldeki işlerden işlem süresi diğerlerine göre kısa olan yarısı alınarak süreleri daha da kısaltılırken, diğer yarısının süreleri aynı miktarda uzatılarak, ortalama değişmeden standart sapmanın artması sağlanmıştır. Benzer şekilde standart sapmanın azaltılması için de eldeki işlerin yarısının süresi artı yönde ortalamaya yaklaştırılırken, diğer yarısının da eksi yönde ortalamaya yaklaşması sağlanmıştır. Elde edilen bu 30 problemle beraber toplam çözdürülen 60 problem için standart sapma ile önerilen yöntemin başarısı arasında anlamlı bir ilişki vardır. Standart sapmanın ortalamaya oranına varyasyon katsayısı denilmektedir. Çözdürülen problemler için ortalama zaman birbirine yakın olduğu için standart sapma değeri önerilen metodun iyi sonuç verdiği durumları gösterse bile aslında etkili olan varyasyon katsayıdır. Çözdürülen problemlerin sonuçları Tablo 3.4'de görülebilir. Tabloda A/B olarak gösterilen varyasyon katsayısı değerinin 0,35'den büyük olduğu durumlarda, % fark alanındaki değerlerin pozitif yönde arttığı, dolayısıyla önerilen yöntemin kötü sonuçlar verdiği söylenebilir. Şekil 3.2'deki grafiksel olarak gösterimden de anlaşılacağı üzere varyasyon katsayısı arttığında EDD daha iyi çözüm vermekte, varyasyon katsayısının 0,1-0,35 arasında olduğu geniş bir aralık için önerilen yöntem çözülen problemlerin yaklaşık yarısında EDD'den daha iyi sonuç vermektedir. Varyasyon katsayısının 0,1'den küçük olduğu durumlarda ise önerilen yöntem %89 oranında daha başarılıdır.

Tablo 3.4. Standart sapma değeri değiştirilmiş problemler için EDD ile önerilen yöntemin kıyaslanması

			Önerilen Yöntem		EDD		
Problem	Std. Sapma (A)	Ortalama İşlem Süresi (B)	Amaç fonksiyon değeri (Toplam beklenen gecikme değeri)	Toplam Gecikme (D)	Toplam Gecikme (F)	% Fark (D-F)/F	A/B
e_60_1 /3	12,76	24,97	6054,72	8977	6069	47,9%	0,51
e_60_2 /3	12,48	24,68	7876,18	14323	10882	31,6%	0,51
e_60_3 /3	12,77	24,96	7048,18	16729	10157	64,7%	0,51
e_60_4 /3	12,94	25,68	6574,91	11366	10179	11,7%	0,50
e_60_5 /3	12,5	25,16	6877,68	12475	11179	11,6%	0,50
e_60_6 /3	12,77	25,57	10109,21	16170	15164	6,6%	0,50
e_60_7 /3	12,46	25,1	8250,25	14233	13222	7,6%	0,50
e_60_8 /3	12,62	24,9	7897,45	14145	10711	32,1%	0,51
e_60_9 /3	12,57	24,85	5550,98	8819	9163	-3,8%	0,51
e_60_10 /3	12,62	25,22	7007,08	14823	13911	6,6%	0,50
e_90_1 /3	7,61	25,39	18113,4	23513	21903	7,4%	0,30
e_90_2 /3	7,32	24,99	16614,26	21750	21903	-0,7%	0,29
e_90_3 /3	7,61	25,2	19887,7	26477	25205	5,0%	0,30
e_90_4 /3	7,79	24,88	16515,24	22796	22840	-0,2%	0,31
e_90_5 /3	7,76	25,41	18913,04	25766	23777	8,4%	0,31
e_90_6 /3	7,66	24,91	19355,77	25219	24090	4,7%	0,31
e_90_7 /3	7,83	24,84	16698,28	21401	20792	2,9%	0,32
e_90_8 /3	7,72	25,31	18780,98	23804	23406	1,7%	0,31
e_90_9 /3	7,62	25,2	20236,09	23583	22762	3,6%	0,30
e_90_10 /3	7,87	25,15	20219,05	23014	21853	5,3%	0,31
e_120_1 /3	1,6	24,99	39061,49	39879	39993	-0,3%	0,06
e_120_2 /3	1,54	24,85	36301,75	37341	37792	-1,2%	0,06
e_120_3 /3	1,61	25,1	35010,58	39677	39115	1,4%	0,06
e_120_4 /3	1,66	24,93	40946,34	43779	43937	-0,4%	0,07
e_120_5 /3	2,11	24,5	43959,05	45358	46313	-2,1%	0,09
e_120_6 /3	1,42	24,94	38052,83	39496	38920	1,5%	0,06
e_120_7 /3	1,56	25,12	36.593	38118	38240	-0,3%	0,06
e_120_8 /3	1,53	24,55	37028,06	37529	37420	0,3%	0,06
e_120_9 /3	1,6	25,04	37337,14	39188	38902	0,7%	0,06
e_120_10 /3	1,64	25,4	42964,42	44551	44609	-0,1%	0,06



Şekil 3.2. EDD başarı oranı ile varyasyon katsayısı ilişkisi

Sonuç olarak, işlem zamanları birbirine yakın olan problemler için, probleme ait işlerin işlem zamanlarına ait varyasyon katsayısı hesaplanarak, yukarıda bahsi geçen yöntemin uygulanıp uygulanmayacağına karar verilebilir.

Yukarıda önerilen yöntemin en iyi çözümü garanti eden yöntemlerle kıyaslabilmesi için daha küçük ölçekli problemler üretilerek benzer şekilde MIP'in ürettiği sonuç, önerilen yöntem ve EDD'nin sonuçları kıyaslanmıştır. Ronconi ve Kawamura tarafından yapılan çalışmada toplam gecikme ve erken başlama ile ilgili tek makine çizelgeleme probleminin karmaşık tamsayılı matematiksel model ile çözülmesinde önerdikleri dal sınır yönteminin 30 problemden daha büyük modellerin çözümünde başarılı olamadığı belirtilmiştir. Yukarıda Tablo 3.3'deki problemlere benzeyen 30 işlik 10 adet problem üretilmiş ve bu problemlerin model 5 ile çözülmesine çalışılmıştır. Ancak, kabul edilebilir sürelerde iyi çözümler elde edilemediği için matematiksel modele ilave kısıtlar eklenerek en iyi çözüme daha kısa sürede erişilmesi hedeflenmiştir. Sadykov tarafından yapılan bir çalışmada 3 işin birbirleriyle ilişkisi kullanılarak kesme kısıtı elde edilmiştir. Belirtilen yöntemden farklı bir şekilde, benzer bir yöntemle alttaki gibi iki kesme kısıtı oluşturulmuştur.

Bu kısıtların amacı eğer i işi j den önce ve j işi de k işinden önce çizelgelenmişse i işinin k işinden önce çizelgeleneceğinin önceden bilinmesinin modelde kullanılarak çözüm uzayındaki olurlu olmayan bölgenin filtrelenmesidir.

$$X_{i,j} + X_{j,k} \leq X_{i,k}+1 \qquad i=1,...,n-1 \quad j=i+1,\ldots,n \quad k=j+1,\ldots,n \qquad (3.21)$$

$$X_{i,j} + X_{j,k} \geq X_{i,k} \qquad i=1,...,n-1 \quad j=i+1,\ldots,n \quad k=j+1,\ldots,n \qquad (3.22)$$

Ayrıca, Emmons'un kurallarından yukarıda bahsedilen birinci kuralın baskınlık özelliği kullanılarak ilave kesme kısıtı elde edilebilir. Bu şekilde elde edilen kısıtlar için modele aşağıdaki karar değişkenleri eklenmiştir.

A_{ik} : İkili değişken, eğer i işinin işlem süresi k işinden kısaysa 1 değilse 0'dır.
B_{ik} : İkili değişken, eğer i işinin termin zamanı k işinden önceyse 1 değilse 0'dır.

Bu değişkenlerle beraber 3 kısıt eklenmiştir.

$$P_j - P_i \leq M \cdot A_{i,,j} \qquad i=1,...,n-1 \quad j=i+1,\ldots,n \qquad (3.23)$$

$$D_j - D_i \leq M \cdot B_{i,,j} \qquad i=1,...,n-1 \quad j=i+1,\ldots,n \qquad (3.24)$$

$$A_{i,j} + B_{i,j} - 1 \leq X_{i,,j} \qquad i=1,...,n-1 \quad j=i+1,\ldots,n \qquad (3.25)$$

Aktürk ve Yıldırım'ın önerdiği baskınlık kuralları da benzer şekilde ilave kesme kısıtları olarak eklenebilir. Bunun için gerekli olan karar değişkenleri ve ilave kısıtlar aşağıda belirtilmiştir. Ancak, kısıtlar eklendiğinde çözdürülen problemlerin belirlenen zaman dilimi içinde önerilen modelin elde ettiği sonuçlara bile ulaşamadığı görüldüğünden Denklem (3.26), (3.27), (3.28), (3.29) kısıtları modele eklenmemiştir. İlk olarak eklenebilecek karar değişkeni;

$C_{i,j}$: İkili değişken, eğer $D_j - D_i \leq P_j$ ise 1 değilse 0'dır.

Bu karar değişkeniyle beraber eklenecek kısıtlar aşağıda belirtilmiştir.

$$D_i - D_j - P_i \leq M \cdot C_{i,,j} \qquad i=1,...,n-1 \quad j=i+1,\ldots,n \qquad (3.26)$$

$$A_{i,j} + C_{i,j} - 1 \leq X_{i,,j} \qquad i=1,...,n-1 \quad j=i+1,\ldots,n \qquad (3.27)$$

İkinci olarak eklenebilecek değişkenler;

$T_{i,j} = D_j - P_i$

$F_{i,j}$: İkili değişken, eğer S_i $T_{i,j}$ den küçükse 1 değilse 0'dır.

Yukarıda eklenebilecek değişkenlerle beraber eklenecek kesme kısıtları altta belirtilmiştir.

$$S_i - D_j + P_i \geq -M \cdot F_{i,,j} \quad i=1,...,n-1 \quad j=i+1,\ldots,n \qquad (3.28)$$

$$F_{i,j} + C_{i,j} \leq 1 + X_{i,,j} \quad i=1,...,n-1 \quad j=i+1,\ldots,n \qquad (3.29)$$

Eklenen kısıtlarla beraber 10 dakika zaman limiti verilerek elde edilen modelle problemlerin çözümü Tablo 3.5'de gösterilmiştir. EDD'den daha iyi sonuç elde edildiği durumlarda önerilen yöntemin en iyi çözümden en fazla %13 uzak olduğu, ortalamada ise %5,2 uzak olduğu görülmektedir. Bir problem içinse önerilen yöntem optimum sonucu vermiştir. Bu durumda hızlı çözüm elde edilmek istenildiğinde ve probleme ait işlerin sürelerinin varyasyon katsayısının 0,35'den küçük olduğu durumlarda önerilen yöntemin kullanılabileceği sonucu ortaya çıkmaktadır.

Tablo 3.5. 30 işlik problemlerin optimum çözümleriyle önerilen yöntemin kıyaslanması

Problem Adı	Toplam Gecikme (Önerilen Yöntem)	Süre (sn)	Toplam Gecikme (EDD)	Toplam Gecikme (MIP)	Ortalama İşlem Zamanı	Std. Sapma	Varyasyon Katsayısı	En iyi çözüm den % Fark
e_30_1	2260	0,25	2158	2004	24,73	3,09	0,125	12,8
e_30_2	3242	0,15	3321	3072	24,77	3,03	0,122	5,5
e_30_3	3020	0,25	3129	2844	24,9	2,72	0,109	6,2
e_30_4	2983	0,42	3037	2983	25,7	2,94	0,114	0,0
e_30_5	2749	0,28	2711	2582	24,97	2,53	0,101	6,5
e_30_6	2398	0,21	2553	2372	24,92	2,98	0,119	1,1
e_30_7	2660	0,12	2798	2621	25,9	2,7	0,104	1,5
e_30_8	2358	0,10	2378	2242	24,07	2,78	0,115	5,2
e_30_9	2656	0,15	2658	2497	24,77	2,97	0,119	6,4
e_30_10	3658	0,15	3870	3468	24,87	2,97	0,119	5,5

3.3. Sonuçların Değerlendirilmesi

Yukarıda da bahsedildiği şekilde aslında Np-Zor olan bir problem tipi için, ele aldığımız lastik fabrikasına uygun olarak geliştirilmiş problemlerin optimum çözümleri polinom zamanda üretilebilmektedir. Tacettin ve diğ. yaptığı çalışmada EDD ile elde edilen sonuçların enerji maliyeti anlamında ideal çözüme göre %10 daha kötü sonuç verdiği söylenebilir. Ele aldığımız problemler için gecikme zamanının çok önemli olmasına rağmen, termin tarihlerinin çok fazla zorlayıcı olmamasından dolayı çok fazla sayıda alternatif çözüm olduğu düşünülebilir. Bu durumda ikinci bölümde ele alınan enerji maliyetlerinin azaltılması problemi, sıfır gecikme sağlayan çözümler arasından enerji maliyetinin enazlanması şeklinde ele alınabilir.

Çok fazla üzerinde çalışılan bir problem tipi olmasına rağmen, işlem zamanının eşit ya da birbirine yakın olan problem türleri için çözüm yöntemleri araştırılmamış bir konudur. Bu bölümde, problem eğer bahsedildiği şekilde ele alınırsa, çözüm adına değişik kapılar açılabileceği gösterilmiştir. İki farklı amacı olan ve bu iki amaç tek başına ele alındığında bile problemin zor olarak sınıflandırıldığı bir problem için bu şekilde problemi basitleştirici bir yöntem oldukça yararlı olmaktadır.

4. AMAÇLARIN BİRLEŞTİRİLMESİ

Şu ana kadar ele alınan çizelgeleme problemi için elektrik maliyetinin minimizasyonu ve toplam gecikme zamanının minimizasyonu ayrı ayrı ele alınmıştır. Son olarak başlangıç popülasyonu rassal olarak üretilecek çözümlerle başlayan ve uygunluk fonksiyonu değerinin iki ayrı amaç için elde edilmiş daha önceki çözümlere uzaklığına göre hesaplanacağı metasezgisel bir yöntem geliştirilmiştir. Genetik algoritma kullanılarak yeni nesiller üretilmiş ve yeni nesildeki çözümlerin kalitesi her iki amaç için elde edilen en iyiye yakın çözüm değerlerine uzaklığa göre hesaplanmıştır.

Çizelgeleme problemlerinin birçoğunda birden fazla amaç güdülmektedir. Örneğin, herhangi bir çizelgeleme problemi için terminlerin karşılanması ve envanterin minimum seviyede tutulması beraber gözetilmesi gereken amaçlar olabilir. Amaçların birincil amaç ve ikincil amaç olarak tanımlandığı durumlarda, birincil amaç için optimum çözümü veren çizelgeler içinden ikinci amaç için en iyi çözüm değerini veren alternatif aranır. Örneğin, birincil amaç $\sum C_j$ iken, ikincil amaç L_{max} ise bu problem $1||\sum C_j^{(1)}, L_{max}^{(2)}$ şeklinde gösterilir. Böyle bir problem için aralarında boşluğa izin verilmeyen SPT kuralına göre oluşturulan bir çizelge birincil amaç için optimum çözümü verir. Eğer işlem süresi birbirinin aynısı olan işler varsa, alternatif çözümler vardır ve işlem süresi aynı olan işler için EDD kuralı işletilir. Bu şekilde işletilen sıralama kuralına SPT/EDD denir (Pinedo, 2008).

İki amacı olan tek makine çizelgeleme problemi için amaçlar γ_1 ve γ_2 ise ve toplam amaç değeri $\theta_1\ \gamma_1 + \theta_2\ \gamma_2$ şeklinde ifade ediliyorsa bu problem $1|\beta|\ \theta_1\ \gamma_1 + \theta_2\ \gamma_2$ şeklinde gösterilir (Pinedo, 2008). Genellikle $\theta_1 + \theta_2 = 1$ olacak şekilde kullanılır. İki amacı olan bir çizelgeleme probleminde herhangi bir çözüm için bir amacının değeri arttırılmadan diğer amacın değeri iyileştirilemiyorsa bu çözüm pareto optimaldir. Pareto optimal çözümlerin kümesi bir düzlem üzerinde gösterilebilir. Farklı θ_1 ve θ2

değerleri kullanılarak farklı pareto optimal çözümler elde edilir. Eğer $\theta_1 \rightarrow 0$ ve $\theta_2 \rightarrow 1$ ise $1|\beta|\ \theta_1\ \gamma_1 + \theta_2\ \gamma_2$ problemi $1|\ \beta\ |\sum \gamma_2^{(1)}, \gamma_1^{(2)}$ problemine döner.

Karmaşık problemlerin çözümünde optimum çözümün elde edilmesi çok uzun zaman almakta yada mevcut teknolojik imkanlarla çözülememektedir. Matematiksel modelleme, dinamik programlama ya da tüm alternatifleri değerlendirerek en iyi çözümü elde etmek zaman ve kaynak açısından pahalı olabilir. Bunun için her bir probleme özel bazı sezgisel yöntemler geliştirilmiştir. Bu sezgisel yöntemler o problemin karakteristiğine göre optimuma belirli bir yakınlıkta olan çözümlerdir. Metasezgisel yöntemler ise genellikle doğada bulunan bazı arama tekniklerini baz alarak genelleştirilmiş, yine problemin yapısına göre özelleştirilebilen ve optimuma yakın çözümler üretebilen yöntemlerdir. Bu yöntemlere örnek olarak, genetik algoritma, tavlama benzetimi, karınca kolonisi optimizasyonu, tabu arama, yapay sinir ağları vb. örnek verilebilir.

Uygun pareto optimal çözümlerin elde edilmesinde genetik algoritma yöntemleri kullanılmıştır. Çok amaçlı problemler için pareto optimal çözümlerin elde edilmesinde farklı θ_1 değerleriyle genetik algoritmayla çözüm aranmaktadır (Bentley ve Wakefield, 1998). Bu çalışmada da benzer şekilde genetik algoritmanın uygunluk fonksiyonu iki amaç fonksiyonunu içerecek şekilde tasarlanmış ve pareto optimal çözüm kümesi elde edilmiştir.

4.1. Genetik Algoritma Yaklaşımı

Bu tez çalışmasında kullanılacak olan genetik algoritma yöntemi ilk olarak John Holland tarafından 1976 yılında geliştirilmiştir. Genetik algoritma ile ilgili ayrıntılar Goldberg ve Michalewicz tarafından anlatılmıştır. Genetik algoritmalar, çözüm uzayı içerisinde global optimum değerine ulaşabilmek için, ilk olarak belirli bir olurlu çözüm kümesiyle başlayarak iteratif olarak daha iyi çözümlerin konrunduğu, kötü çözümlerin elendiği bir arama yöntemidir. Bu arama yöntemi Darwin'in "Doğal Seleksiyon" prensibine dayanır.

Eldeki problem için bir çözümü ifade eden diziye kromozom adı verilmektedir. Kromozomu oluşturan her bir elemana gen denir. Her bir kromozom, çözüm uzayında bir noktayı ya da bölgeyi temsil eder.

Bir problemin genetik algoritma ile çözümünde uygulanacak adımlar aşağıda sıralanmıştır.

1. Başlangıç popülasyonunu oluştur.

Rassal bir çözüm seti oluşturulur. Bu n sayıda kromozomdan oluşan bir kümedir.

2. Her bir kromozom için uygunluk fonksiyonuna göre bir uygunluk değeri hesapla.

Hesaplanan uygunluk değeri kromozomun çözüm kalitesini gösterir.

3. Bir grup kromozom belirli bir olasılık değerine göre rassal olarak seçilip üreme işlemi gerçekleştirilir.

Üreme işlemi için çeşitli çaprazlama ve mutasyon işlemleri yapılabilir.

4. Yeni bireylerin uygunluk değerleri hesaplanır.
5. Önceden belirlenen nesil sayısı boyunca 3. ve 4. adımlar devam ettirilir.
6. Uygunluk değeri en yüksek olan kromozom optimuma en yakın çözüm olarak seçilir.

Bir genetik algoritmanın performansı doğru parametreleri kullanılarak geliştirilebilir. Popülasyon büyüklüğü, çaprazlama oranı, mutasyon oranı ve iterasyon sayısı ayarlanması gereken faktörler olarak ele alınabilir. Her bir parametrenin genetik algoritma performansı üzerine etkisi vardır. Örneğin popülasyon sayısı çok küçük olduğunda genetik algoritma çözüm uzayı içinde yanlış bir yere doğru gidebilir. Tersi bir durumda ise hesaplama için kullanılan araçlardaki kaynakların israfı söz konusu olabilir. Çaprazlama oranının yüksek olması çözüm uzayında daha fazla yerin keşfedilmesine ve hatalı bir lokal optimuma yerleşilme şansının artmasına sebep olabilir. Mutasyonun amacı premature çözümleri engellemektir. Eğer mutasyon katsayısı çok büyük olursa, rassal karışıklık fazla olacağı için üreyen yeni nesillerin bir önceki nesile benzeme ihtimali ve algoritmanın bir önceki nesilden öğrenme kabiliyeti azalır.

Üreme yöntemi bir nesilden bir sonrakine geçişte kopyalanma sürecidir. Burada uygunluk fonksiyonu yüksek olan kromozomların daha yüksek ihtimalle hayatını

devam ettirilmesi hedeflenmektedir. Literatürdeki üreme yöntemleri aşağıda sıralanmıştır.

1. Rulet çemberi yöntemi; Bu yöntemde her bir bireyin yaşamını devam ettirme şansı uygunluk fonksiyonu değerine bağlıdır. Daha iyi uygunluk fonksiyonuna sahip kromozomlar rulette daha fazla yer kaplar ve seçilme şanslarını artırır.
2. Turnuva yöntemi; Her nesil içinden her seferinde herhangi 2 kromozom seçilir. Bunlar turnuvaya katılmış olur. Daha iyi uygunluk fonksiyonuna sahip olan turnuvayı kazanır.
3. Sabit durum yöntemi; Rastgele seçilen n tane kromozomun çaprazlanmasıyla elde edilen yeni kromozomlar popülasyona katılır, aynı şekilde popülasyon içinden en kötü n tanesi çıkartılır.
4. Sıralama yöntemi; Tüm kromozomlar uygunluk fonksiyonuna göre sıralanır En iyiler bir sonraki nesile daha fazla kopya bırakır.

Bu yöntemlerin arasında anlamlı bir fark olmadığı ve gezgin satıcı problemi için sıralama yönteminin daha iyi olduğu sonucu ortaya çıkmıştır (Kumar ve Jyotishree, 2012). Bu çalışmada da üreme yöntemi olarak sıralama yöntemi tercih edilmiştir.

Çaprazlama operatörü iki ebeveynin dizisinden yeni nesiller oluşturmak için kullanılan çaprazlama metodunu belirler. Genetik algoritmada kullanılan çaprazlama operatörleri aşağıda sıralanmıştır.

1- Pozisyona dayalı çaprazlama
2- Sıraya dayalı çaprazlama
3- Kısmi planlı çaprazlama (PMX)
4- Dairesel çaprazlama (CX)
5- Doğrusal sıralı çaprazlama (LOX)
6- Sıralı çaprazlama (OX)
7- Alt değişimli çaprazlama (SXX)
8- İş tabanlı çaprazlama

Bu çaprazlama operatörlerinden Engin ve Fığlalı tarafından akış tipi çizelgeleme problemleri için yapılan çalışmada makine sayısı iki olduğunda sıralı çaprazlama operatörünün daha başarılı olduğu gösterilmiştir. Chantaravarapan ve Jatinder

tarafından tek makina için grupsal değişme zamanların bulunduğu ortamda toplam gecikme zamanının enazlanması problemi için yapılan çalışmada genetik algoritma çaprazlama operatörü olarak Emmons kuralına bağlı olarak sıralı çaprazlama (EOX) metodu kullanılmıştır. Avcı ve diğ. tarafından yapılan bir çalışmada tek makinede ağırlıklı toplam gecikmeyi enazlama çizelgeleme problemi için çaprazlama operatörünün bir etkisinin olmadığı gösterilmiştir. Bu durumda performansının iyi olduğu bilindiğinden çaprazlama yöntemi olarak OX (Sıralı çaprazlama), EOX (Emmons kuralına dayalı sıralı çaprazlama) ve enerji maliyetini azaltmaya yönelik olarak oluşturulan, baskınlık kuralına bağlı ortaya çıkan yeni bir yöntem uygulanacaktır.

Mutasyon operatörü ise var olan bir kromozom üzerinde rastgele değişiklikler yapar.

Bununla ilgili literatürde var olan yöntemler aşağıda sıralanmıştır.

1. Değer değiştirme yöntemi; Kromozomdaki değerin 1 iken 0 ya da tersi yapılması şeklinde gerçekleştirilen bir mutasyondur.
2. Kaydırma yöntemi; Kromozom içinde rassal olarak belirlenen bir bloğun yerinin yine rastgele başka bir yere kaydırılması şeklinde gerçekleştirilen bir mutasyondur.
3. Yerleştirme yöntemi; Kromozom içinde rastgele bir genin yerinin değiştirilmesi şeklinde gerçekleştirilen bir mutasyondur.
4. Karşılıklı değişim yöntemi; Kromozom içinde seçilen iki genin yerlerinin değiştirilmesi şeklinde gerçekleştirilen bir mutasyondur.

Bu çalışmada mutasyon operatörü olarak karşılıklı değişim yöntemi kullanılmıştır.

4.1.1. EOX (Emmons kuralına dayalı sıralı çaprazlama) yöntemi

Çaprazlama operatörü iki kromozomu alarak bunlardan yeni nesiller üretir. Sezgisel bazlı çaprazlama operatörleri daha iyi uygunluk fonksiyonu değeri gösteren nesiller üremesini sağlar. Bu operatör iki aşamadan oluşmaktadır. İlki sıralı çaprazlama yöntemi, ikinci ise Emmons kuralınan dayanan sezgisel yöntemdir. Sıralı çaprazlama yöntemi işlerin sırasını nesiller boyunca aynı ya da benzer konumunda tutabildiği için çizelgeleme problemlerinde tercih edilmektedir. Bu yöntem bir örnekle anlatılacaktır. Sıralı çaprazlama yöntemi (OX) bazı işlerin çıkarılması ve kalan işlerin ileri ya da geri kaydırılması ile oluşur.

Alltaki örnekte kromozom rastgele iki yerinden kesilmiştir. Bunlar 3. ve 4. işler arası, diğeri ise 6. ve 7. işler arasıdır.

Ebevevn A = 4 7 8 | 1 3 2 | 5 6

Ebeveyn B = 2 3 4 | 5 1 6 | 8 7

Üreyecek iki kromozom ebeveynlerden kopyalanır. Ancak, diğer ebeveyn'e ait kesilmiş kısımdaki işler elenir. Ebeveyn B için kesim yerleri arasında kalan işler 5, 1 ve 6'dır. Bu yüzden, A'dan üreyen kromozomda 5, 1 ve 6 işleri çıkarılır. Çıkarılan işlerin yerine x konulur. B'den üreyen kromozom için de benzer şekilde üretilir.

A' = 4 7 8 | x 3 2 | x x

B' = x x 4 | 5 x 6 | 8 7

Sonrasında kesilen yer boşaltılarak, diğer ebeveyn'den kopartılan kısmın yapıştırılması sağlanır. Bu işlem kesim yerindeki işlerin ileri ya da geri kaydırılmasıyla yapılır. A' kromozomunu düşünecek olursak, kesimden önceki kısım doludur. Dolayısıyla, 3 ve 2 nolu işlerin geriye doğru kaydırılması gerekir. B' kromozomu içinse 4, 5 ve 6 nolu işler öne kaydırılarak boş yerlere getirilir. Sonrasında açılan yere diğer ebeveyn'den işler taşınır.

A' = 4 7 8 | 5 1 6 | 3 2

B' = 4 5 6 | 1 3 2 | 8 7

EOX'un ikinci kısmında ise Emmons kuralına uygun bir şekilde oluşan kromozom üzerinde oynama yapılarak sıralama değiştirilir. Yukarıdaki örnekte Emmons kuralından dolayı 3 işi 4 işinden önce gelmesi gerekiyorsa ve 5 işi 6 işinden önce çizelgelenmeliyse, üreyen yeni nesildeki kromozomlar alttaki gibi değiştirilir.

A'' = 3 7 8 | 5 1 6 | 4 2

B'' = 3 5 6 | 1 4 2 | 8 7

Benzer bir şekilde EOX'un son adımında Emmons kuralından yararlanarak toplam gecikme zamanını iyileştirmesi gibi enerji maliyetinin azaltılmasına yönelik olarak

aşağıda bahsedilen baskınlık kuralına göre iyileştirme yapılabilir (Chantaravarapan ve Jatinder, 2003).

Eğer i işinin ve j işinin süreleri eşitse ve i işinin enerji maliyeti j işinden çoksa ve i işi j işinden daha pahalı bir periyodda çizelgelenmişse, i işiyle j işinin yerini değiştirmek toplam enerji maliyetini azaltır.

Bu kural kullanılarak A' sıralaması için eğer 2 ve 7 nolu işlerin süreleri aynı, 2 nolu işin enerji maliyeti 7 nolu işten yüksek ve 2 nolu işin çizelgelendiği periyottaki birim maliyet 7 nolu işin çizelgelendiği periyoddan yüksekse bunların yerleri değiştirilebilir. Aynı şekilde B' sıralaması içinde tüm işler ikili olarak ele alınıp benzer şekilde değiştirme işlemi yapılır. Bu yöntem "Enerji maliyetine göre OX çaprazlama operatörü" (EMOX) olarak adlandırılacaktır.

4.2. Deney Tasarımı

Deney tasarımı verimliliği artırmak, metotları, ürünleri ya da süreçleri geliştirmek için kullanılan bir yaklaşımdır (Jiju ve Preece, 2002). Deney tasarımı bağımlı faktördeki değişikliğin nedeninini ele alınan bağımsız faktörlerin ne derece etkilediğini ortaya çıkarmak üzere kullanılan bir tekniktir (Çömlekçi, 2003). Taguchi deney tasarımı yöntemi ise optimum faktör seviyelerinin belirlenmesinde kullanılan bir tekniktir.

Taguchi metodu deney tasarımı ile proseslerdeki ya da ürünlerdeki varyasyonların azaltılmasını kapsar. Metodun ana amacı yüksek kaliteli ürünlerin düşük maliyetle üretilmesidir. Taguchi yöntemi Dr. Genichi Taguchi tarafından geliştirilmiştir. Taguchi prosesin ne kadar düzgün işlediğini gösteren performans parametrelerinin ortalamalarının ve varyasyonlarının değişik parametreler tarafından nasıl etkilendiğini ortaya koyan bir metot geliştirmiştir. Taguchi tarafından önerilen metot ortogonal serilerin kullanılarak prosesi etkileyen parametrelerin farklı seviyeler için sonucu nasıl etkilediğini inceler. Faktöriyel tasarımda olduğu gibi tüm olası kombinasyonların denenmesi yerine Taguchi yöntemi kombinasyon çiftlerini test eder. Bu sayede hangi faktörlerin ürün kalitesine etki ettiğini görmek için yeterli veriyi minimum sayıda deney yaparak elde eder. Böylece kaynak ve zaman tasarrufu sağlar. Değişken sayısının 3 ila 50 arasında değiştiği durumlarda, parametreler arası

etkileşimin az olduğu ve sadece birkaç parametrenin sonuca etkilediği durumlarda taguchi yöntemi oldukça etkilidir.

Taguchi dizileri, küçük diziler için manuel olarak, büyük diziler içinse deterministik algoritmalarla hesaplanabilir. Diziler faktör sayısına ve seviye sayısına göre belirlenir. Taguchi deney tasarımı ile elde edilen verilere ait varyans analizi ile performans karakteristiğini optimize eden parametre değerleri tespit edilebilir. Dizilerden elde edilen veri görsel olarak, ANOVA yöntemiyle, Fisher tam olasılık testiyle ve Chi-kare testiyle analiz edilebilir.

Taguchi metodunda kalite ürünün içinde tasarlanmalıdır felsefesi vardır. Kalite hedefe göre minimum varyasyonla elde edilir. Ürün çevresel faktörlerden minimum etkilenecek şekilde tasarlanmalıdır. Kalitenin maliyeti standarttan sapma ve sistemsel kayıpların toplamıdır.

Taguchi Yönteminin adımları aşağıda sıralanmıştır.

1. Faktörleri belirle
2. Test ortamını belirle
3. Kontrol ve gürültü faktörlerini belirle
4. Deney matrisini tasarla
5. Data analiz prosedürünü tanımla

Buraya kadarki kısım birinci fazdır.

6. Deneyleri gerçekleştir
7. Datayı analiz et
8. Performansı tahmin et
9. ANOVA ve S/G analizi
10. Deneyi doğrula

Taguchi yönteminin amacı bahsedildiği üzere maliyetleri azaltmak ve prosesteki varyasyonu azaltmaktır. Taguchi'ye göre kayıp fonksiyonu performans parametresini gösteren hedef değerle (τ) ölçülen değeri (y) arasındaki farka bağlıdır. Kayıp fonksiyonu aşağıda belirtilmiştir.

$$l(y)=k_c(y-\tau)^2$$

Buradaki k_c sabit değeri tanım limitleri ve kabul edilebilir aralık değerine göre hesaplanır.

$$k_c = \frac{C}{\Delta^2}$$

Bu sabit değerin hesaplanmasında zorluk C değerinin tanımlanmasında oluşmaktadır. Proses çıktısının minimum olmasının en iyi olduğu durumda, $\tau = 0$'dır ve kayıp fonksiyonu alttaki gibidir.

$$l(y) = k_c y^2$$

Proses çıktısının maksimum olmasının en iyi olduğu durumda ise, kayıp fonksiyonu alttaki gibidir.

$$l(y) = \frac{k_c}{y^2}$$

Farklı parametrelerin proses çıktısı üzerindeki etkileri incelenirken, Taguchi tarafından önerilen ortogonal diziler kullanılarak deney tasarımı yapılır. Prosesi etkileyen faktörlerin sayısı ve her bir faktöre ait seviyelerin sayısı tespit edildikten sonra hangi ortogonal dizinin kullanılacağı tespit edilebilir. Bunun için Şekil 4.1'deki şablon kullanılabilir. Bu tabloya göre örneğin eğer 3 faktör (gerilim, sıcaklık, basınç) ve her bir faktör için 2 seviye (yüksek, alçak) varsa kullanılması gereken dizi L4'dür.

Seviye Sayısı	Faktör Sayısı																												
	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	L4	L4	L8	L8	L8	L8	L12	L12	L12	L12	L16	L16	L16	L16	L32	L32	L32	L32	L32	L32	L32	L32	L32	L32	L32	L32	L32	L32	L32	L32
3	L9	L9	L9	L18	L18	L18	L18	L27	L27	L27	L27	L27	L36	L36	L36	L36	L36	L36	L36	L36	L36	L36								
4	L16	L'16	L16	L'16	L'32	L'32	L'32	L'32	L'32																					
5	L25	L25	L25	L25	L25	L50	L50	L50	L50	L50	L50																			

Şekil 4.1. Faktör ve sayılarına göre ortogonal seri eşleme tablosu

Yukarıdaki şekil her bir faktör için eşit sayıda seviye olduğu durumda geçerlidir. Böyle olmadığı durumlarda en yüksek seviye sayısına sahip faktöre göre işlem yapılabilir. Ele alınan problem için 3 faktör ve her bir faktör için 3 seviye olduğundan L9 ortogonal dizisi kullanılacaktır. Faktör sayısının 4 olduğu durumda da L9 ortogonal dizisi kullanılır. L9 ortogonal dizisine ait deney tasarımı modeli Tablo 4.1'de gösterilmiştir.

Tablo 4.1. L9 ortogonal dizisi

	Faktör			
Deney	F1	F2	F3	F4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Deneyler yapıldıktan sonra her bir faktörün çıktı üzerindeki etkisini anlamak için her bir deney için sinyal gürültü oranı hesaplanır. Bu değer aşağıda belirtildiği şekilde hesaplanır.

$$SN_i=10 \log \frac{\bar{y}_i^2}{s_i^2} \tag{4.1}$$

$$\bar{y}_i^2=\frac{1}{N_i}\sum_{u=1}^{N_i} y_{i,u} \tag{4.2}$$

$$s_i^2=\frac{1}{N_i-1}\sum_{u=1}^{N_i} (y_{i,u}-\bar{y}_i) \tag{4.3}$$

Burada i deney sayısını, u deneydeki tekrar sayısını, N_i ise i deneyindeki tekrar sayısını gösterir.

4.3. Önerilen Yöntem

Ele aldığımız problemin birden fazla amacı bulunmaktadır. Şimdiye kadarki kısımda her bir problem tek amaçlı olarak ele alınmış ve bu amaca yönelik olarak problem incelenmiş ve en iyi ya da eniyiye yakın çözümler üretilmiştir. Bu kısımda ise her bir problem için herhangi bir olurlu çözümün bulunan eniyiye yakın çözümlere uzaklığına göre kullanıcıya baskın çözüm kümesi sunulacaktır. Genetik algoritma için, çözümlerin değerlendirilmesi amacıyla kullanılan uygunluk fonksiyonu aşağıdaki Denklem (4.4) formülüyle belirtilmiştir.

F_k : k kromozomuna ait uygunluk değeri

T_{ik} : k kromozumunda i işinin gecikme süresi

E_{ik} : k kromozumunda i işinin enerji maliyeti

T : Ele alınan problem için bulunan en iyi toplam gecikme zamanı

E : Ele alınan problem için bulunan en iyi enerji maliyeti toplamı değeri

α : Katsayı

$$F_k = \sum_{i=1}^{n} \frac{T_{ik}}{T} \alpha + \sum_{i=1}^{n} \frac{E_{ik}}{E} (100-\alpha) \quad (4.4)$$

α değeri 0'dan başlayarak 100'e kadar 10 artılarak elde edilen farklı uygunluk fonksiyonları için genetik algoritma çalıştırılmış ve her bir α değeri için elde edilen en iyi 50 çözüm saklanmıştır. Toplamda oluşan 550 çözüm içinde baskın olmayanlar elenerek elde edilen küme grafiksel olarak elde edilmiş ve karar vericinin seçimine sunulmuştur. Bu bölümde sunulan genetik algoritma yaklaşımı toplam gecikme zamanının minimizasyonuna ilişkin amaçta belirtilen işlem zamanlarının birbirine yakın olması koşulunu da gerektirmemektedir.

Baskın çözüm kümesinin elde edilmesinde ise uygulanan yöntem altta belirtilmiştir.

1. Elde edilen en iyi uygunluk fonksiyonu değerine sahip 550 çözüm listesi içinden iki farklı sıralama S_a ve S_b'yi al.
2. Eğer S_a sıralamasına ait enerji maliyeti S_b sıralamasına ait enerji maliyetinden küçükse ve S_a sıralamasına ait toplam gecikme zamanı S_b sıralamasına ait toplam gecikme zamanından küçükse S_b sıralamasını listeden sil.
3. İkinci maddedeki prosedürü listede kalan tüm ikili sıralamalar için gerçekleştir.

Genetik algoritmanın çalıştırılması sırasında kullanılacak olan diğer faktörlerin belirlenmesi amacıyla deney tasarımı yapılmış ve en iyi çözümü veren uygun parametre değerleri tespit edilmiştir.

Yukarıda bahsedildiği şekilde taguchi yöntemiyle deney tasarımının uygulanması için öncelikli olarak faktörler tespit edilmiştir. Bu faktörler ve bu faktörlere ait seviyeler Tablo 4.2'de belirtilmiştir. Geliştirilen genetik algoritmanın belirlenen faktörleri "çaprazlama seçme oranı", "mutasyon oranı" ve "başlangıç popülasyonu sayısı" dır. İterasyon sayısı arttıkça daha iyi sonuçlar vereceği bilindiği için deney tasarımına bir faktör olarak eklenmemiştir.

Tablo 4.2. Deney tasarımı için kullanılacak faktörler ve seviyeleri

	Seviyeler		
	1	2	3
Çaprazlama Seçme Oranı	0,4	0,6	0,8
Mutasyon Oranı	0,05	0,1	0,15
Başlangıç Popülasyonu Sayısı	100	250	500

Yukarıda bahsedilen 3 faktör ve her biri 3 seviyeden oluşan bir deney seti için uygun olan ortogonal seri L9'dur. Bu diziye göre tasarlanan deneyler Tablo 4.3'de belirtilmiştir.

Tablo 4.3. L9'a göre yapılacak deneyler

	Faktör		
Deney	1	2	3
1	0,4	0,05	100
2	0,4	0,1	250
3	0,4	0,15	500
4	0,6	0,05	250
5	0,6	0,1	500
6	0,6	0,15	100
7	0,8	0,05	500
8	0,8	0,1	100
9	0,8	0,15	250

Deneyler gerçekleştirilirken örnek problem olarak e_60_1 problemi gecikme zamanları orijinal probleme göre 2/3 oranında azaltılmış olarak ele alınmıştır. Çaprazlama yöntemi olarak yukarıda bahsedilen sıralı çaprazlama yöntemi uygulanacaktır. α değeri ise her iki amacın etkisini eşit görmek için 50 olarak belirlenmiştir. Bu 9 deney her biri 5 tekrardan oluşan deneyler şeklinde gerçeklenmiştir. Yukarıda bahsedilen uygunluk fonksiyonu için en küçük en iyi denilebilir. Dolayısıyla yapılan deneyler için Sinyal/Gürültü (S/G) oranı alttaki formüle göre hesaplanmıştır.

$$S/G_i = -10.\log\left(\sum_{u=1}^{N_i} \frac{y_u^2}{N_i}\right)$$

Yapılan deneylerin sonuçlarına göre elde edilen sinyal/gürültü (S/N) oranları Tablo 4.4'de belirtilmiştir.

Tablo 4.4. Yapılan deneyler için S/G oranları

Deney	S/G
1	-41,504
2	-41,791
3	-41,908
4	-41,824
5	-41,963
6	-41,739
7	-41,911
8	-41,696
9	-41,932

Bu değerlere göre hesaplanan her bir faktörün her bir seviyesi için ortalama S/G oranları ise Tablo 4.5'de gösterilmiştir. S/G(i,j) i faktörünün j seviyesi için olan sinyal/gürültü oranıdır.

Tablo 4.5. Her bir faktörün her bir seviyesi için S/G oranları

S/G	Değer
S/G (1,1)	-41,73
S/G (1,2)	-41,84
S/G(2,1)	-41,75
S/G(3,3)	-41,93
S/G(2,3)	-41,82
S/G(3,1)	-41,65
S/G(1,3)	-41,85
S/G(2,3)	-41,86
S/G(3,2)	-41,85

Yukarıdaki tablodan da görüleceği üzere tüm faktörler için birinci seviye değerleri en iyi sonucu vermiştir. Dolayısıyla genetik algoritma parametrelerinden çaprazlama seçme oranı 0,4, mutasyon oranı 0,05, başlangıç popülasyonu sayısı 100 olarak problemler çözdürülmüştür. Genetik algoritma MATLAB programı kullanılarak kodlanmış ve çözdürülmüştür. Genetik programlamaya ait MATLAB kodu EK-E'de gösterilmiştir.

4.4. Problemlerin Sonuçları

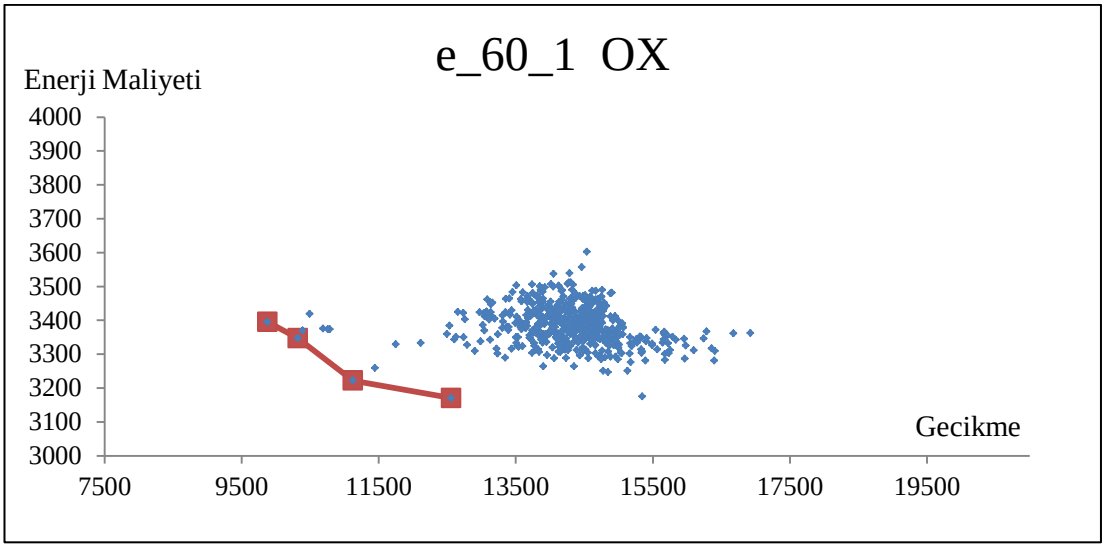
Yukarıda bahsedilen EMOX, EOX ve OX çaprazlama operatörlerinin her üçü de kullanılarak eldeki 30 problem çözdürülmüştür. Termin kısıtının zorlayıcı olabilmesi

için orijinal problemdeki termin zamanları yerine yine bu termin zamanlarının 1/3'ü kadar olan değerle işlem yapılmıştır. Tablo 4.6'da e_60_1 problemi için OX, EOX ve EMOX çaprazlama yöntemi ile birleştirilmiş amaç fonksiyonundaki çeşitli α değerleri için elde edilen baskın çözüm kümesine ait sonuçlar gösterilmiştir.

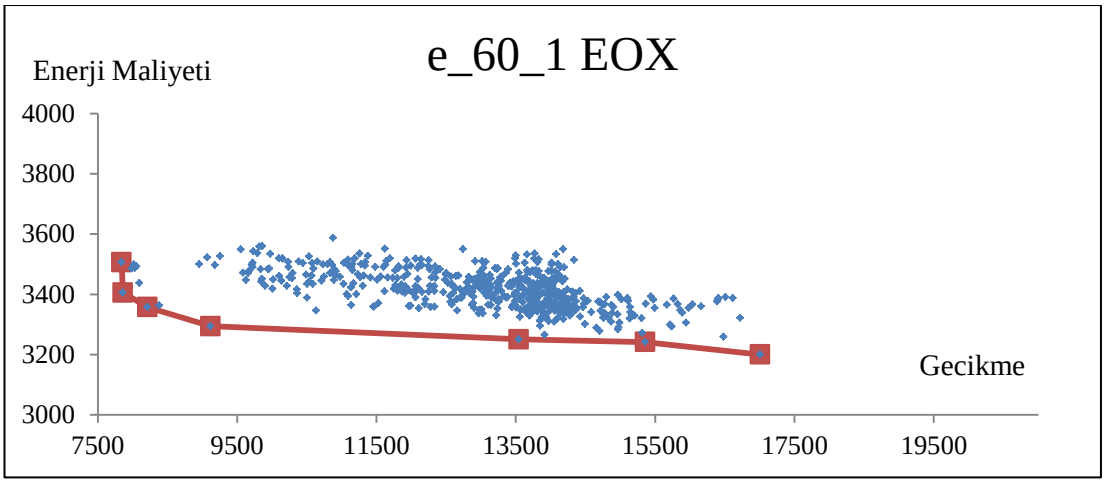
Tablo 4.6. e_60_1 problemi için baskın çözüm kümesi

	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	9875	3395,4	80	
	10319	3347,12	50	
	11124	3221,8	40	-
	12556	3170,44	10	
EOX	17008	3200,16	0	
	15358	3241,2	30	
	13543	3250,28	50	
	9118	3294,12	10	-
	8212	3357,08	30	-
	7858	3405,76	80	-
	7840	3506,24	50	-
EMOX	11198	3309,4	60	
	11544	3230,4	30	
	11664	3177,36	50	-
	12072	3162,8	40	-
	13015	3131,28	10	-
	15614	3103,08	0	-

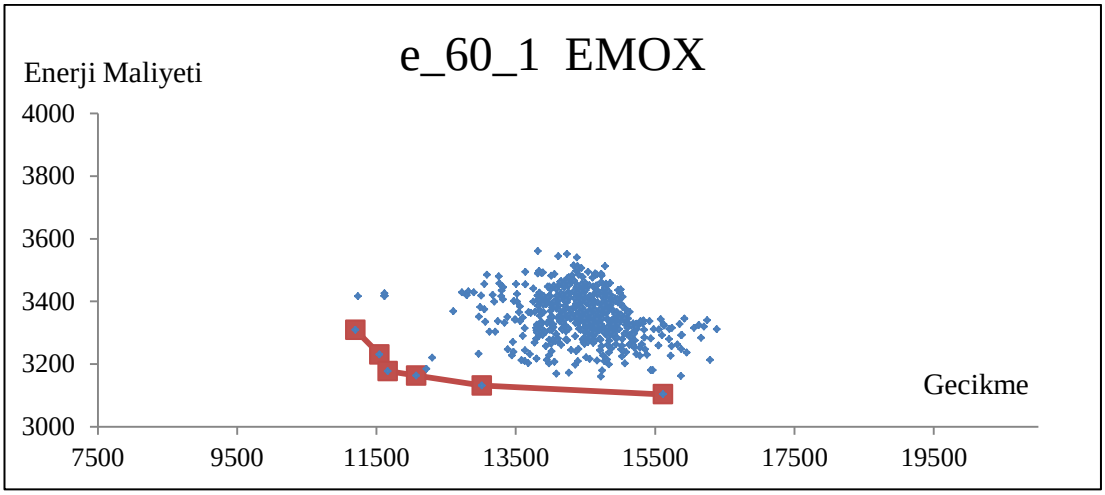
Tabloda tüm çaprazlama yöntemleri beraber düşünüldüğünde baskın olan değerler işaretlenmiştir. Tablo 4.6'dan görüleceği üzere EOX ile gecikme zamanında iyi çözümler elde edilebilirken, EMOX ile de enerji maliyeti anlamında daha iyi çözümler elde edilmiştir. EOX ile elde edilen gecikme zamanı iyi olan çözümlerin enerji maliyeti yüksek çıkabilmektedir. Bu problem için elde edilen EOX ile toplam gecikme zamanı TMTGZM ile elde edilen değerden daha iyidir. Aynı şekilde EMOX ile elde edilen enerji maliyeti değeri TMEMM ile elde edilen değerden daha iyi çıkmıştır. Şekil 4.2, 4.3 ve 4.4'de e_60_1 problemi için elde edilen değerlerin dağılımı grafiksel olarak gösterilmiştir.



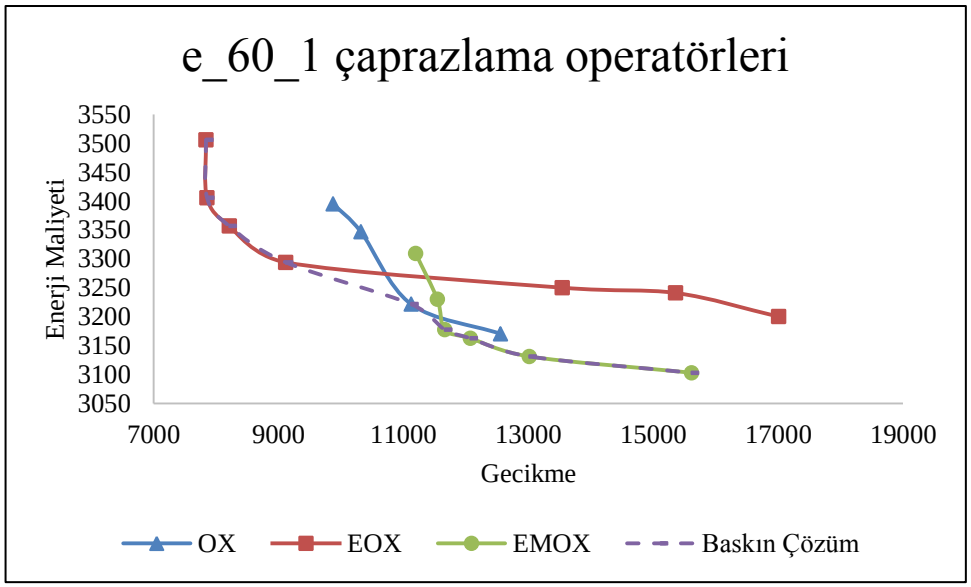
Şekil 4.2. e_60_1 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.3. e_60_1 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.4. e_60_1 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



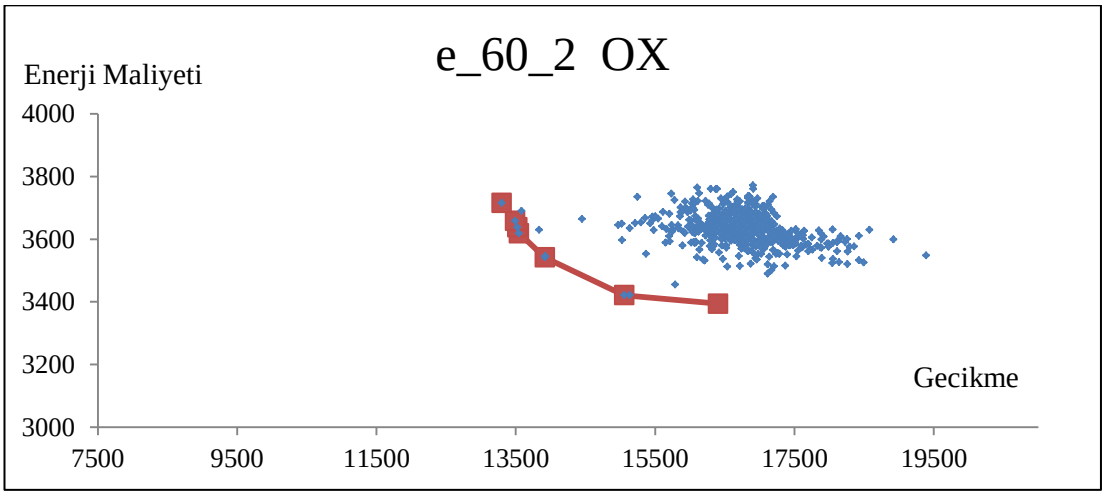
Şekil 4.5. e_60_1 problemi için çaprazlama operatörlerinin kıyaslaması

Şekil 4.5'de her bir çaprazlama operatörünün baskın çözüm kümeleri ve genel baskın çözüm kümesi tek bir grafik altında gösterilmiştir. Tüm sonuçlar değerlendirildiğinde EMOX enerji maliyeti olarak, EOX ise toplam gecikme zamanı olarak iyi sonuç verdiği için elde edilen baskın çözüm kümesinde EOX ile elde edilmiş 4 çözüm, EMOX'la elde edilmiş 4 çözüm varken OX ile elde edilmiş sadece bir çözüm vardır.

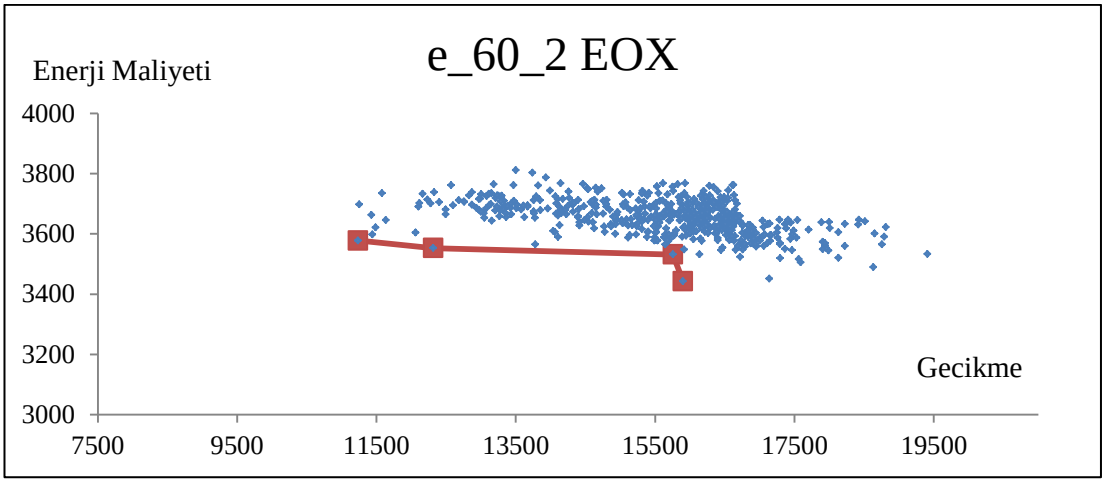
Diğer problemlerin çözümünden elde edilen baskın çözüm kümesi değerleri ve bunların grafiksel olarak gösterimi aşağıda belirtilmiştir.

Tablo 4.7. e_60_2 problemi için baskın çözüm kümesi

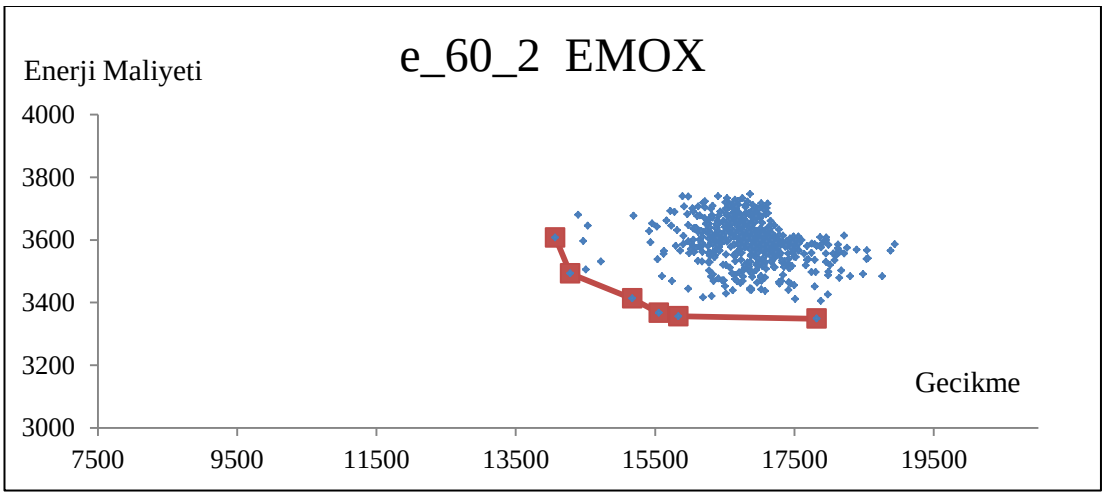
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	16404	3394,04	0	
	15058	3420,88	20	-
	13917	3541,52	30	-
	13547	3618,84	40	
	13525	3637,64	60	
	13489	3658,44	70	
	13297	3715,8	80	
EOX	15900	3442,76	10	
	12314	3553,04	20	-
	11236	3577,76	70	-
	15756	3531,84	100	
EMOX	14066	3607,48	50	
	14282	3492,44	40	-
	15172	3413,08	30	-
	15555	3366,68	20	-
	15833	3355,92	10	-
	17819	3348,4	0	-



Şekil 4.6. e_60_2 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi

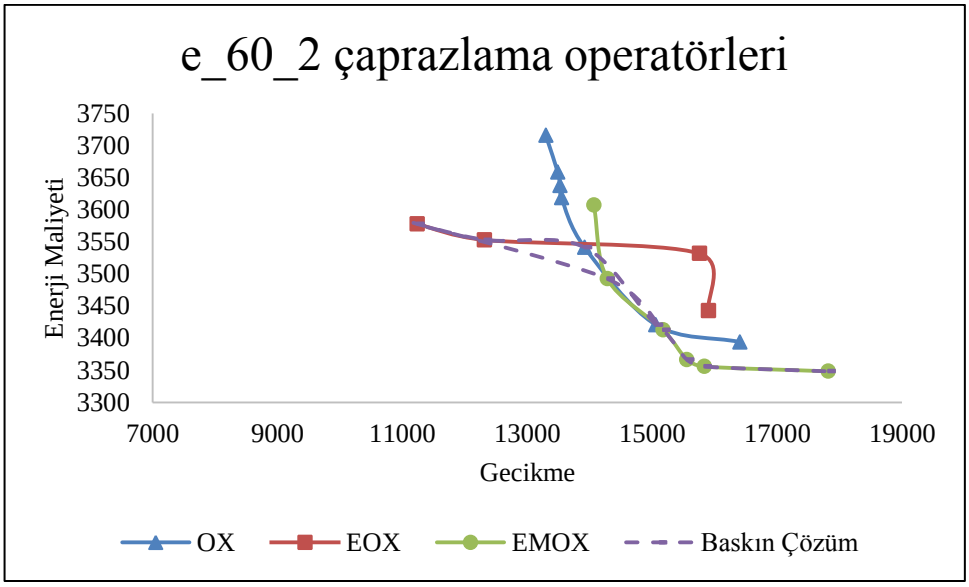


Şekil 4.7. e_60_2 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.8. e_60_2 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi

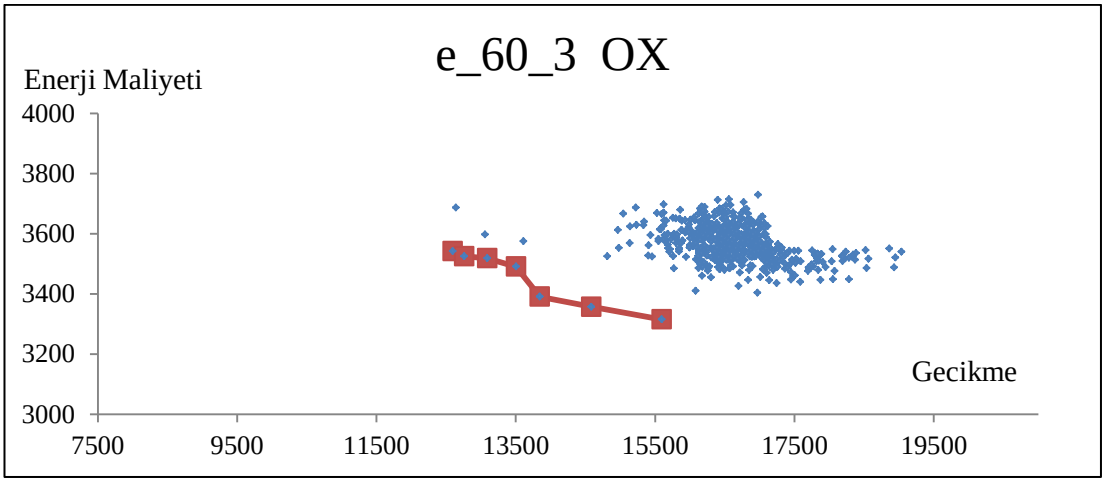
Bu problem için de OX ile elde edilemeyen toplam gecikme zamanı değerleri EOX ile yine OX ile elde edilemeyen enerji maliyetleri değerleri EMOX ile elde edilmiştir. Şekil 4.9'da de görüleceği üzere tüm çözümler içinde elde edilen baskın çözüm kümesi içerisinde EMOX çaprazlama operatörü ile çözülen 5 seçenek varken, EOX ile 2 çözüm, OX ile 2 çözüm baskın çözüm kümesine girebilmiştir. Bu problem içinde EOX'un gecikme zamanı olarak, EMOX'un ise elektrik maliyeti olarak iyi sonuçlar verdiği söylenebilir.



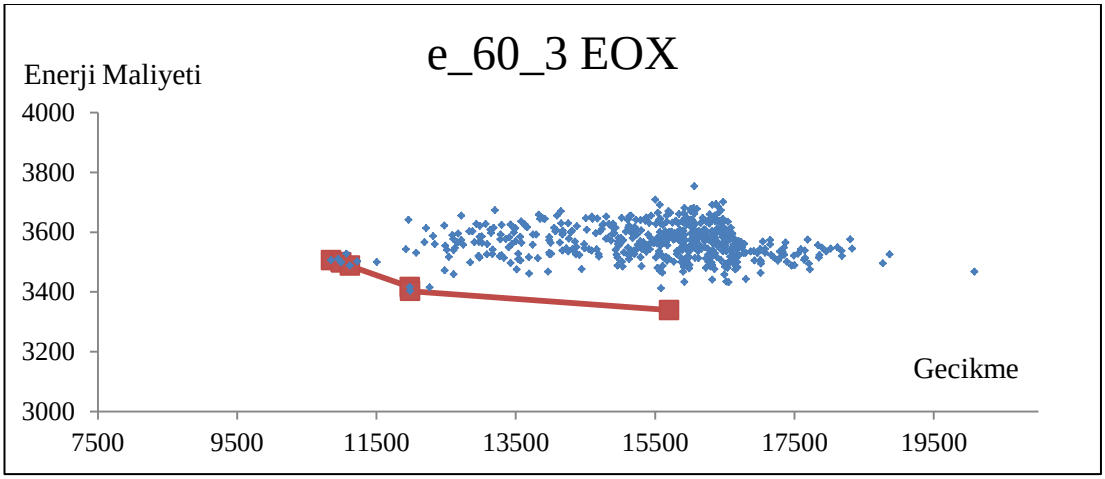
Şekil 4.9. e_60_2 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.8. e_60_3 problemi için baskın çözüm kümesi

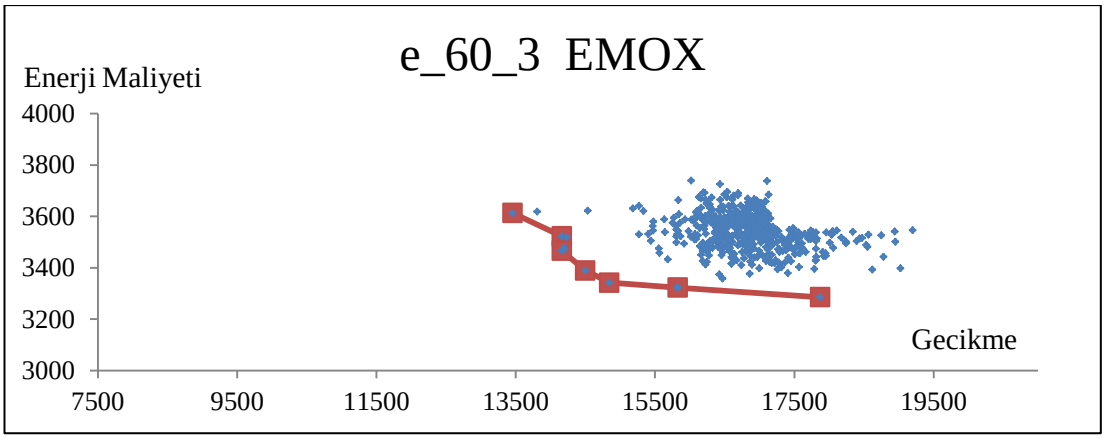
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	12595	3542,04	80	
	12762	3525,12	60	
	13092	3518,72	50	
	13504	3490,76	40	
	13845	3390,92	30	-
	14585	3357,44	20	-
	15597	3315,24	10	-
EOX	10848	3506	40	-
	10986	3499,88	50	-
	10995	3498,56	80	-
	11119	3487,44	60	-
	11981	3416,08	30	-
	11984	3402,64	20	-
	15703	3338,88	0	
EMOX	13457	3613,04	50	
	14161	3522,36	60	
	14162	3465,8	30	
	14500	3389,52	40	-
	14845	3341,64	20	-
	15827	3322,64	10	
	17874	3284,92	0	-



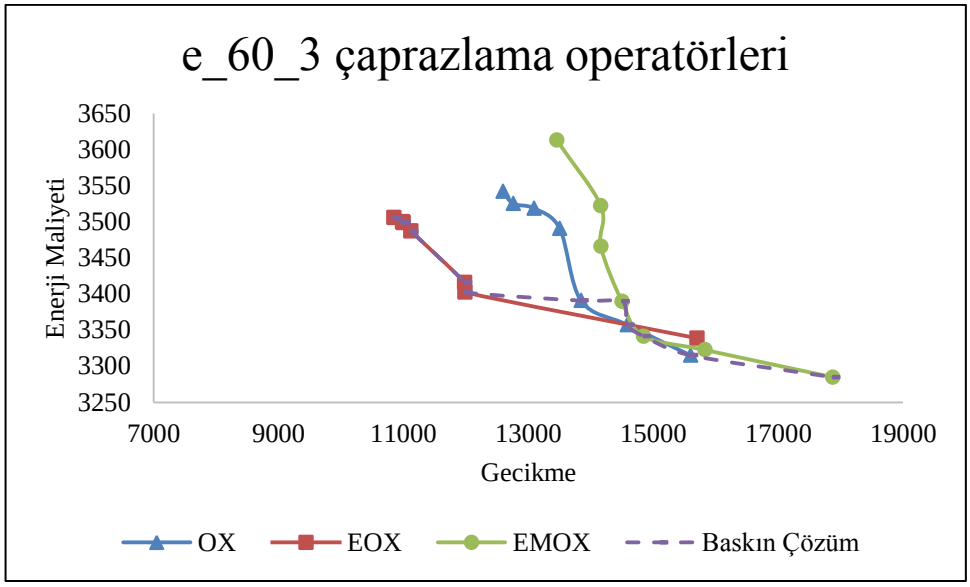
Şekil 4.10. e_60_3 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.11. e_60_3 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



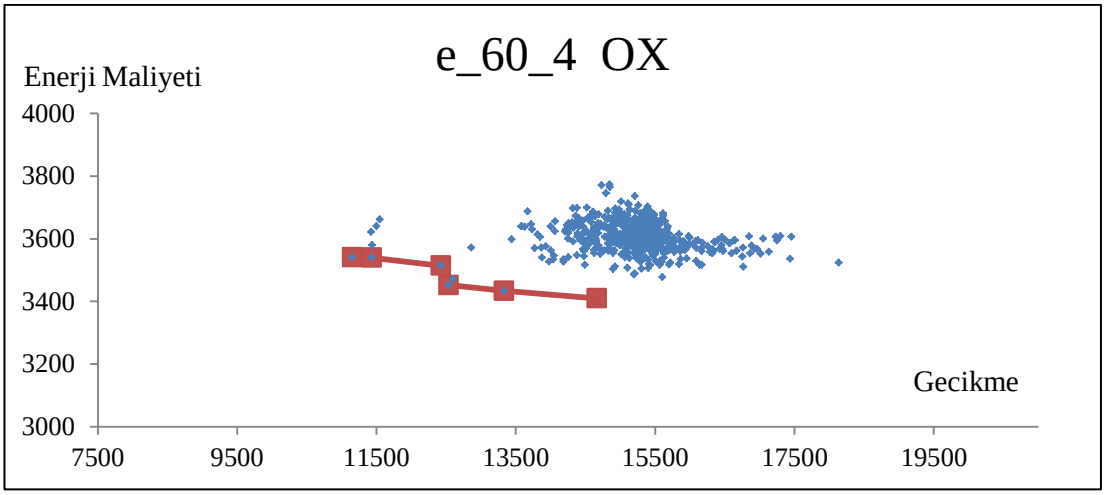
Şekil 4.12. e_60_3 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



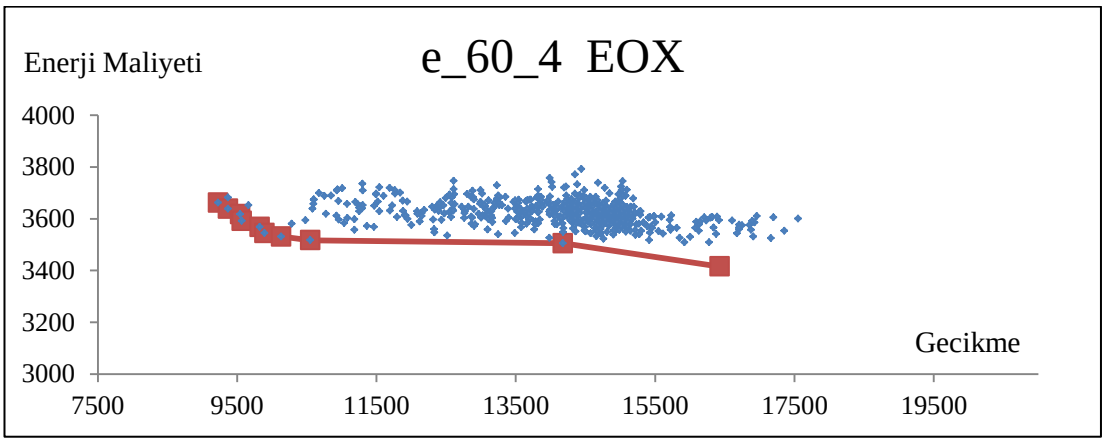
Şekil 4.13. e_60_3 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.9. e_60_4 problemi için baskın çözüm kümesi

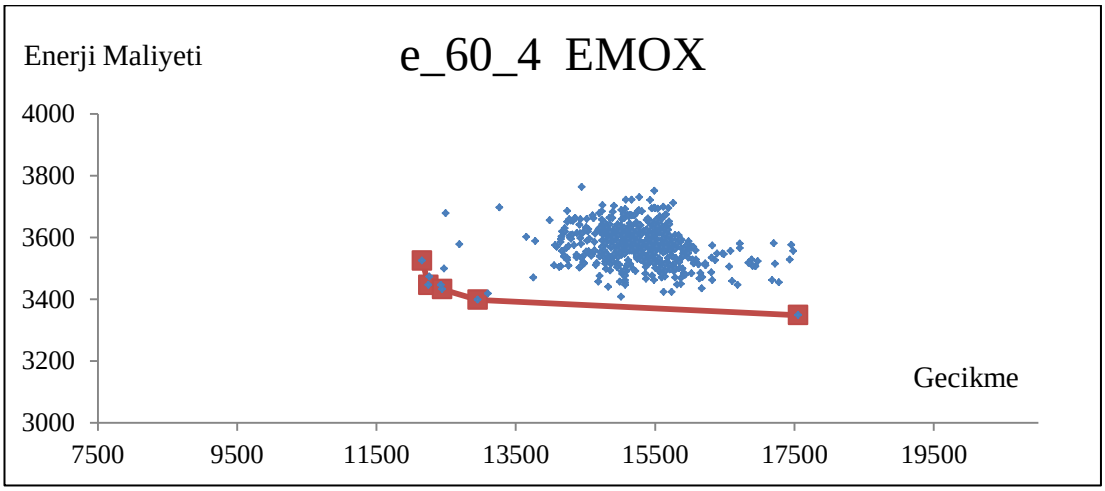
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	11150	3540,48	40	
	11432	3539,48	80	
	12425	3513,96	60	
	12535	3452,08	20	
	13330	3433,6	10	
	14662	3409,96	0	
EOX	9227	3662,48	90	-
	9370	3638,72	70	-
	9543	3617,16	60	-
	9568	3592,04	50	-
	9823	3567,88	40	-
	9894	3544,04	30	-
	10131	3531,32	20	-
	10551	3516,8	10	-
	14174	3505,36	90	
	16426	3415,44	0	
EMOX	12154	3525	80	
	12246	3445,76	60	-
	12442	3432,36	20	-
	12954	3398,32	30	-
	17554	3348,28	0	-



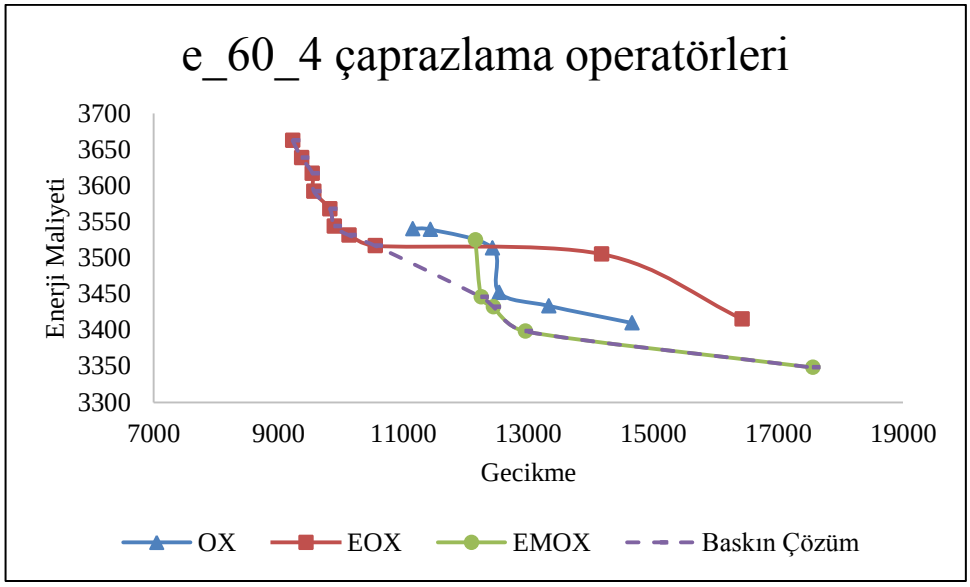
Şekil 4.14. e_60_4 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.15. e_60_4 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.16. e_60_4 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi

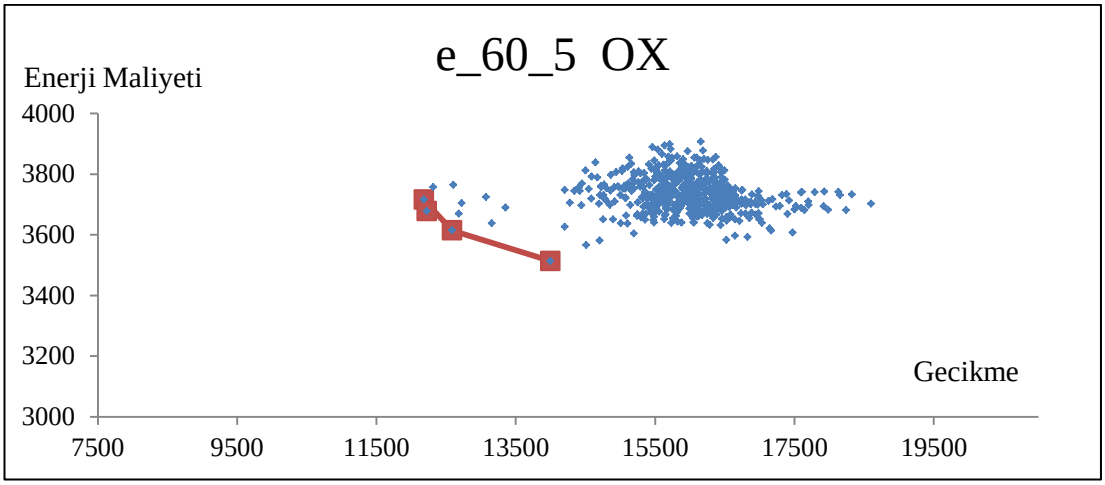


Şekil 4.17. e_60_4 problemi için çaprazlama operatörlerinin kıyaslaması

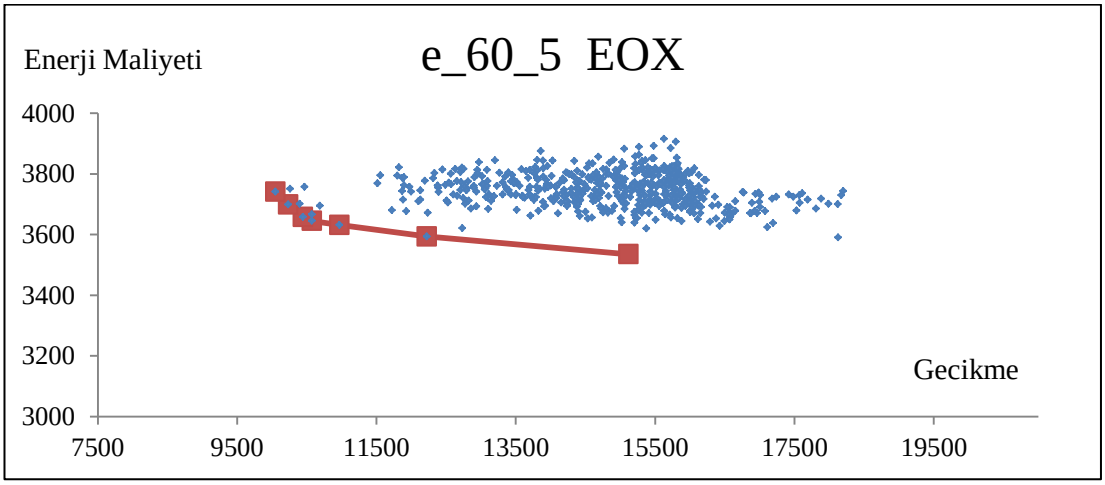
Şekil 4.17'de görüleceği üzere tüm çaprazlama operatörlerinin sonuçlarına göre oluşturulan baskın çözüm kümesine OX ile çözdürülmüş hiçbir çözüm girememiştir.

Tablo 4.10. e_60_5 problemi için baskın çözüm kümesi

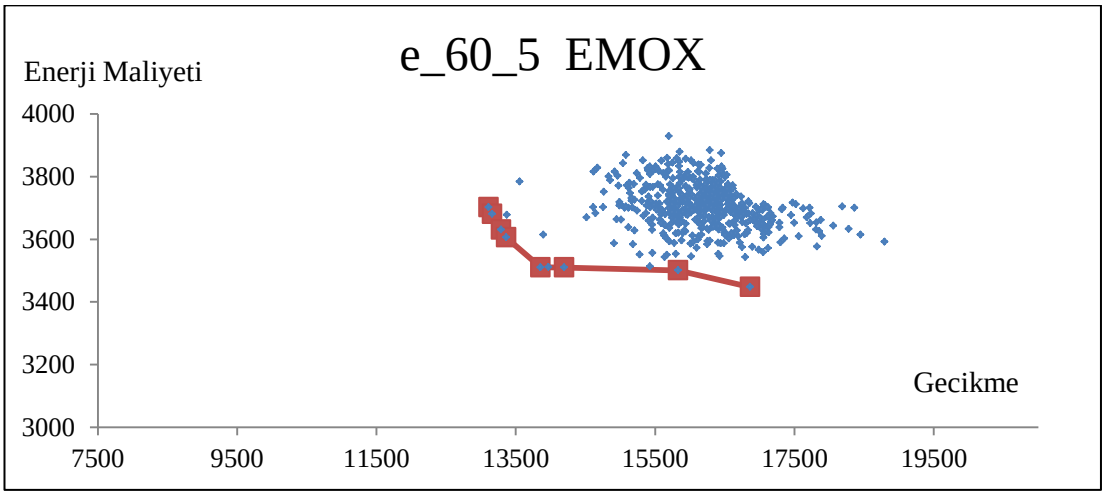
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	13997	3513,6	10	
	12588	3614,36	20	
	12181	3715,4	30	
	12223	3677,84	90	
EOX	15120	3534,92	0	
	12221	3593,76	10	-
	10967	3631,92	20	-
	10572	3644,96	30	-
	10445	3657,92	40	-
	10231	3699,4	50	-
	10049	3740,8	100	-
EMOX	13108	3701,92	70	
	13159	3680,92	100	
	13290	3631,16	90	
	13362	3605,8	50	
	13856	3510,48	30	-
	14196	3509,88	20	-
	15830	3500,64	30	-
	16865	3447,64	0	-



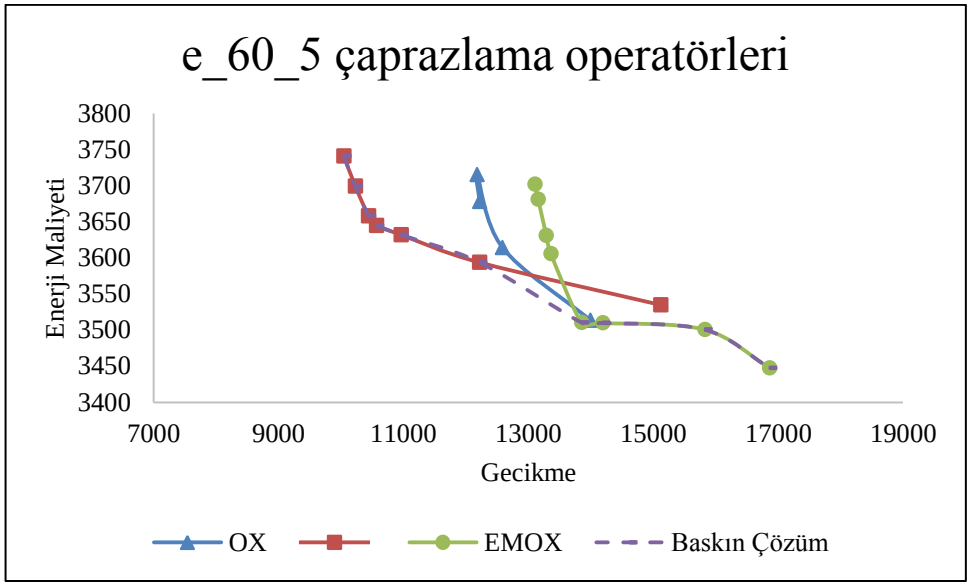
Şekil 4.18. e_60_5 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.19. e_60_5 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



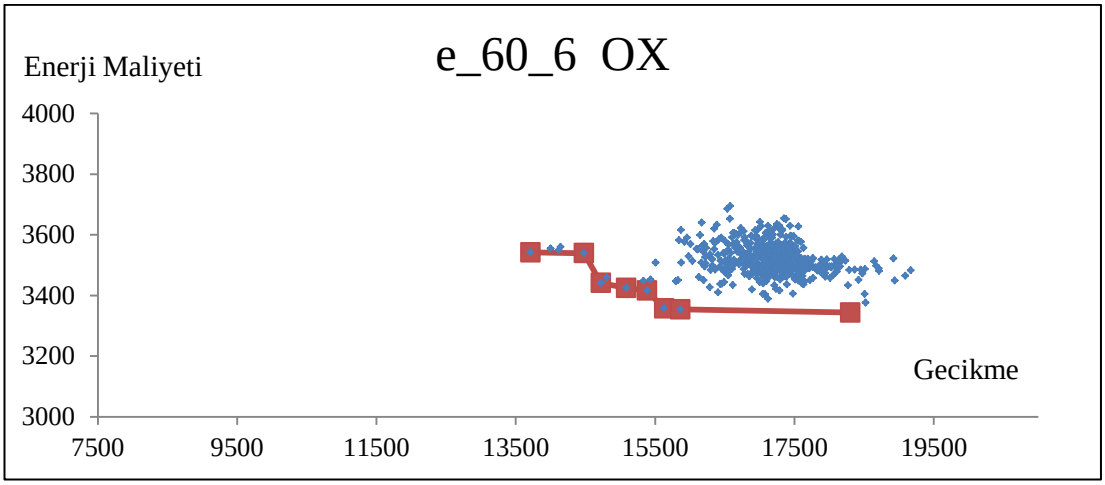
Şekil 4.20. e_60_5 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



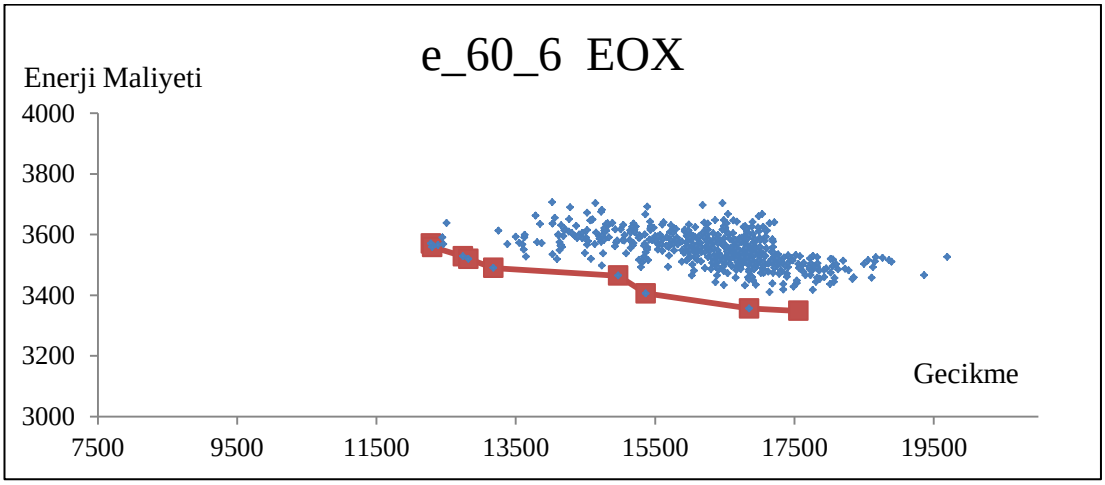
Şekil 4.21. e_60_5 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.11. e_60_6 problemi için baskın çözüm kümesi

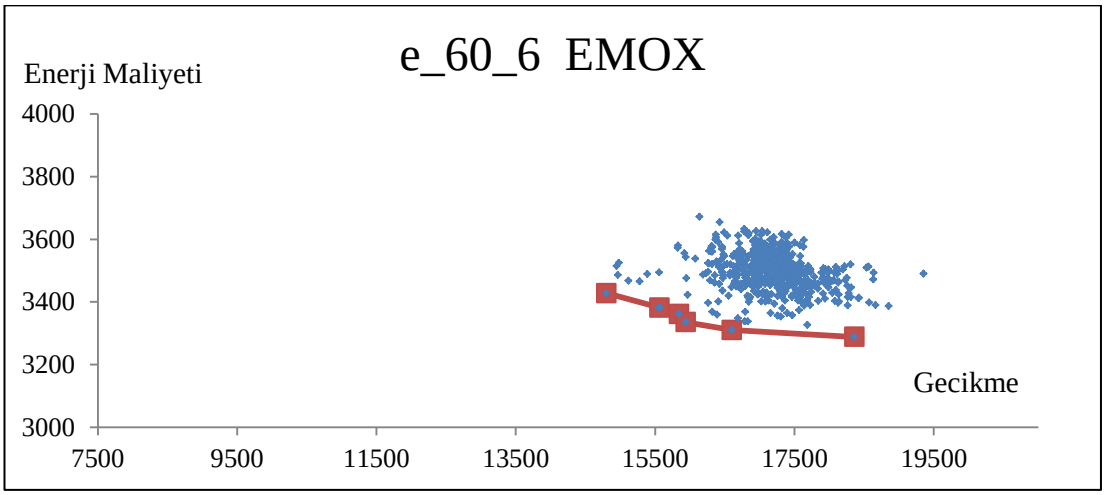
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	13710	3542,24	100	
	14481	3539,56	90	
	14721	3441,2	40	-
	15085	3424,88	30	-
	15385	3415,8	40	
	15632	3356,76	20	-
	15862	3354,04	10	-
	18302	3343,68	0	
EOX	12282	3570,64	70	-
	12302	3558,76	50	-
	12741	3528,08	30	-
	12820	3519,92	40	-
	13180	3489,52	20	-
	14970	3464,36	80	
	15366	3405,88	80	-
	16852	3356,44	10	
	17558	3348,2	0	
EMOX	14802	3427,68	80	-
	15564	3381,84	40	-
	15844	3360,6	20	
	15942	3334,96	30	-
	16604	3310,24	10	-
	18364	3288,04	0	-



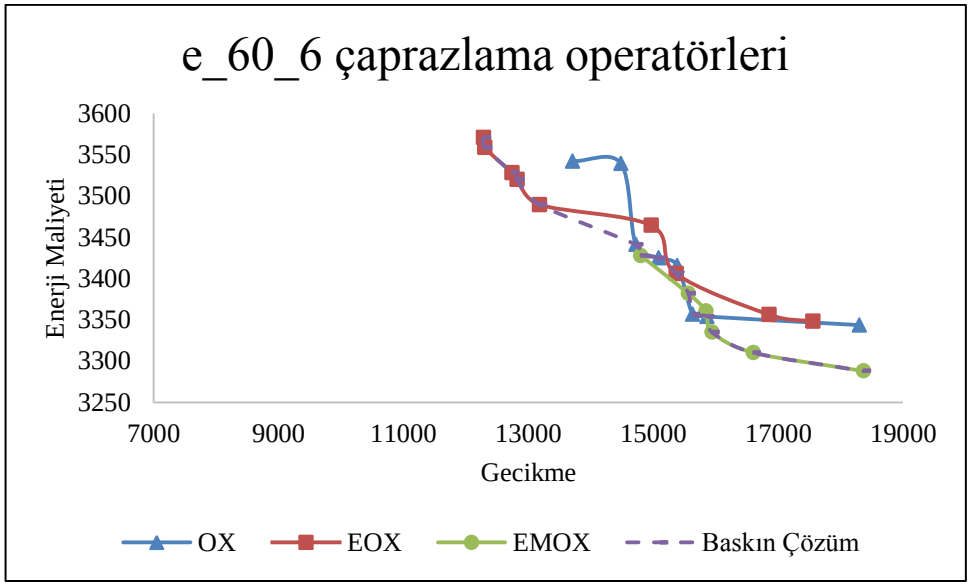
Şekil 4.22. e_60_6 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.23. e_60_6 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



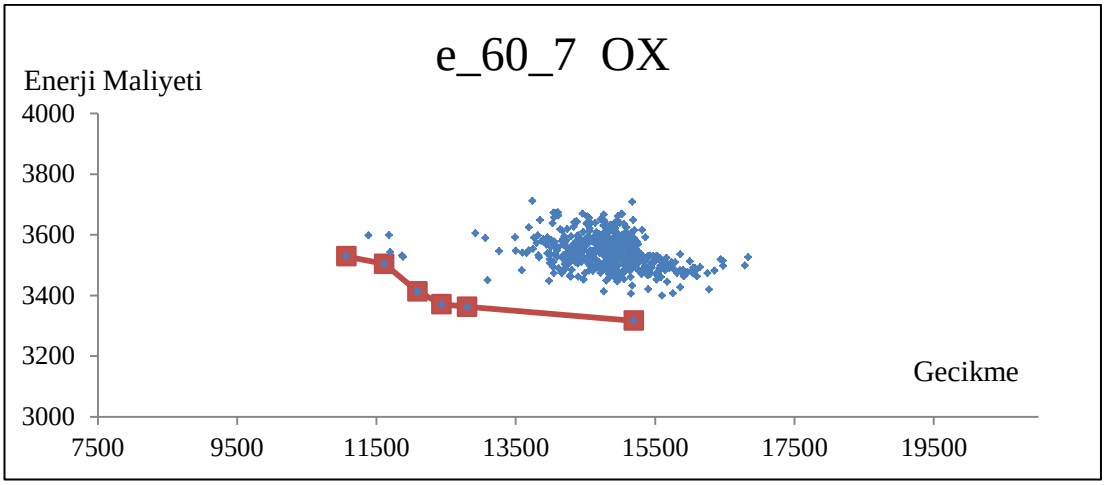
Şekil 4.24. e_60_6 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



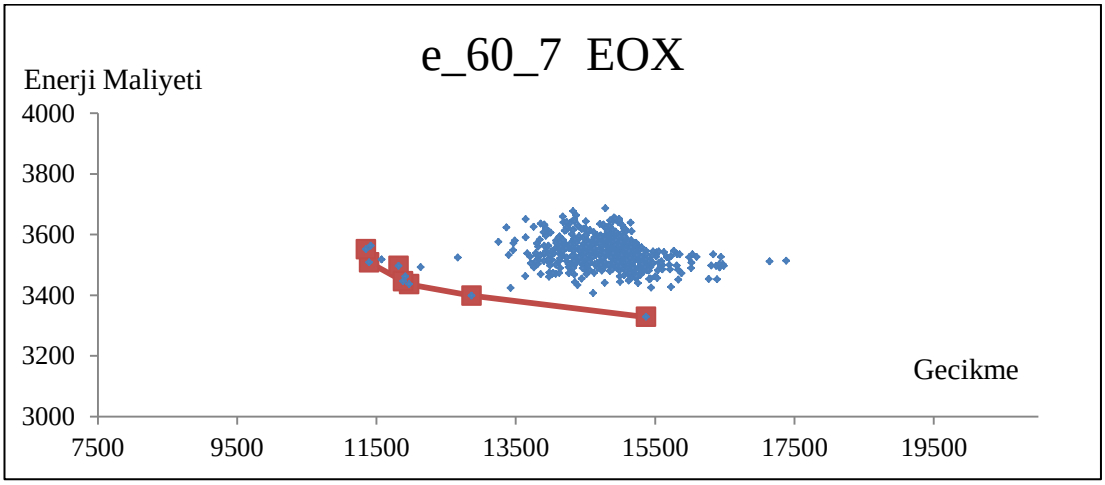
Şekil 4.25. e_60_6 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.12. e_60_7 problemi için baskın çözüm kümesi

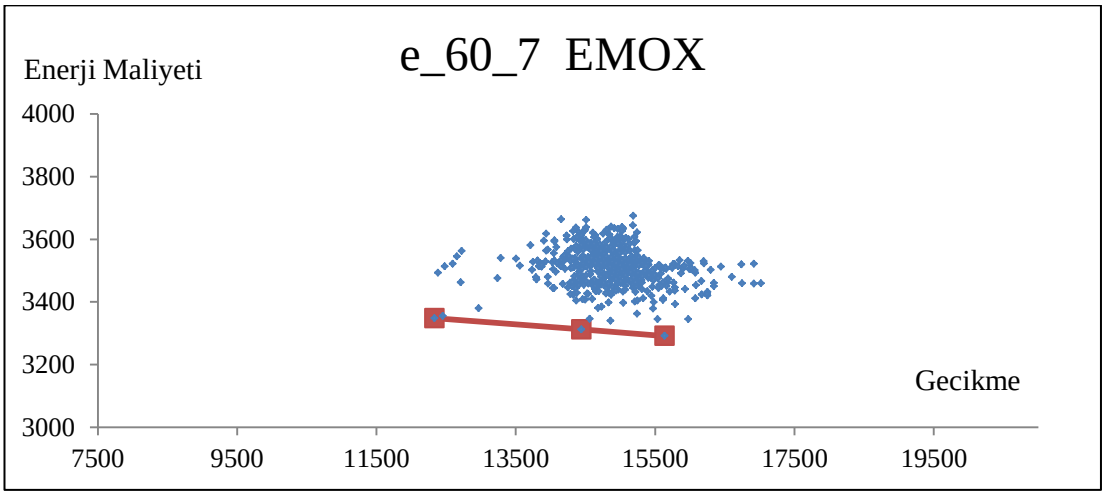
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	11068	3528,68	100	-
	11612	3503,8	30	-
	12088	3413,16	40	-
	12435	3371,2	20	
	12803	3362,44	10	
	15198	3316,64	0	
EOX	15370	3328,52	0	
	12864	3398,52	10	
	11969	3436,16	20	-
	11817	3496,72	40	-
	11884	3446,2	60	-
	11394	3508,28	70	-
	11350	3550,88	80	
EMOX	12331	3347,96	40	-
	14443	3312,2	10	-
	15636	3291,48	0	-



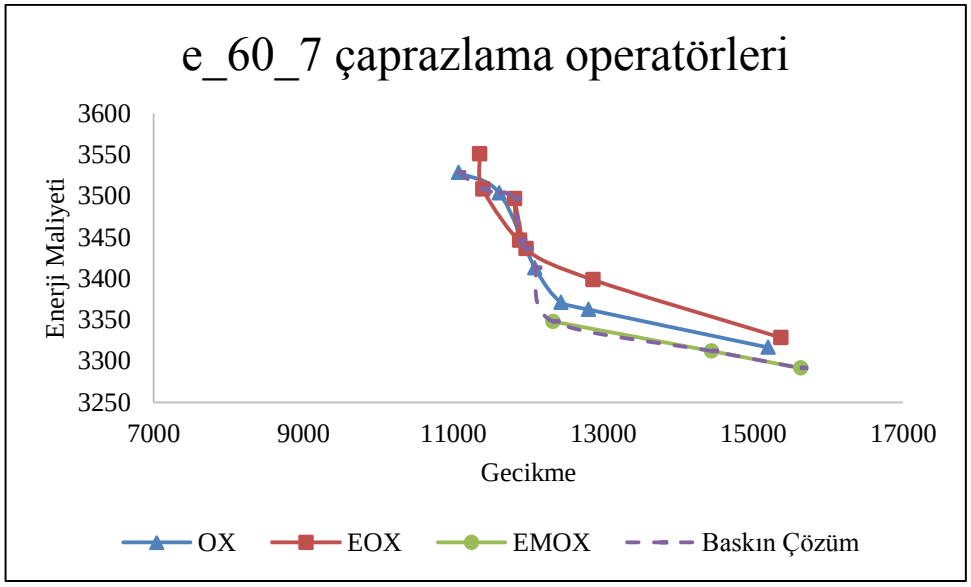
Şekil 4.26. e_60_7 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.27. e_60_7 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



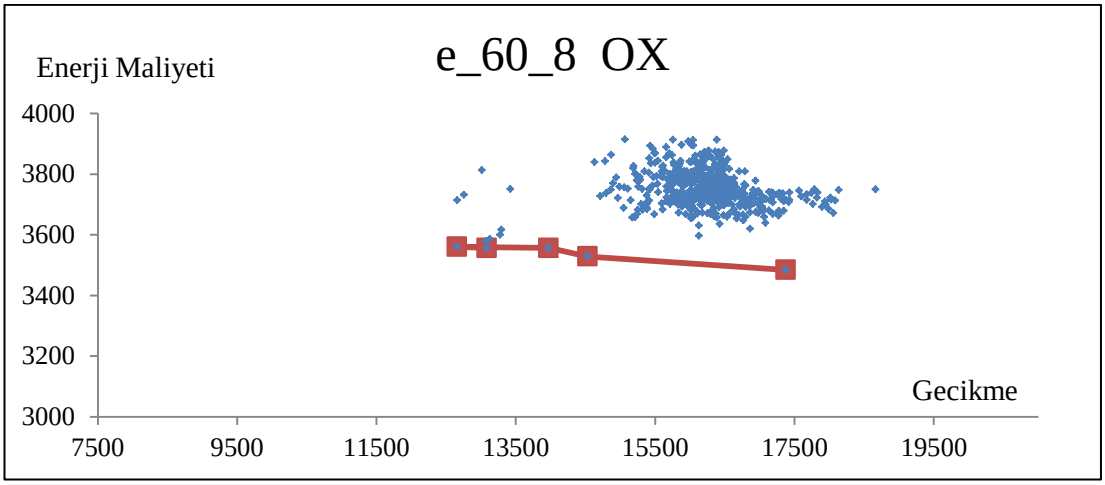
Şekil 4.28. e_60_7 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



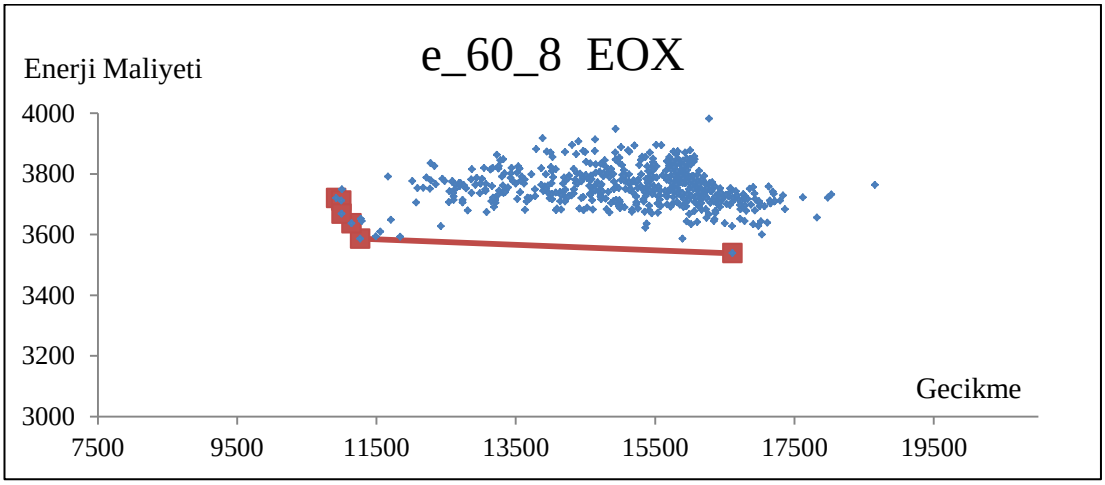
Şekil 4.29. e_60_7 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.13. e_60_8 problemi için baskın çözüm kümesi

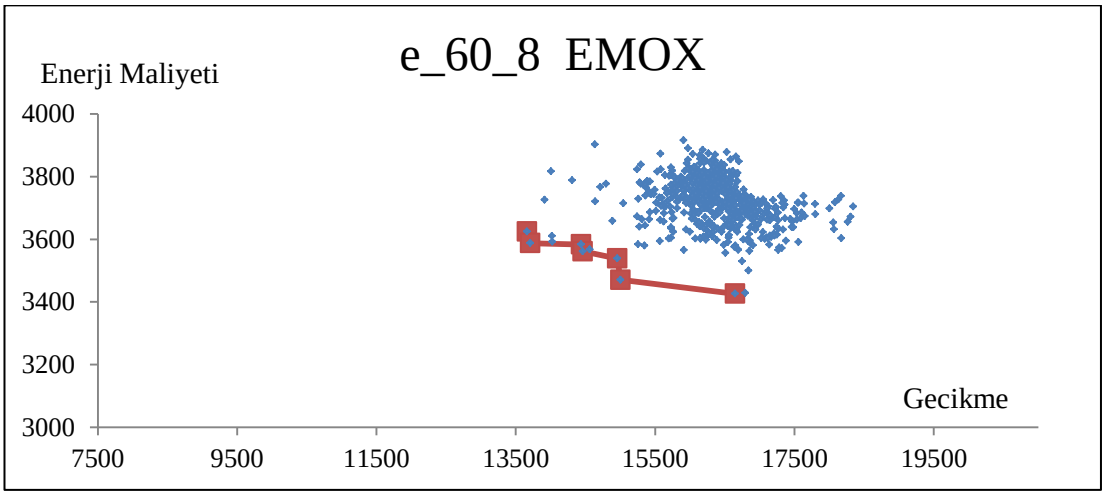
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	17376	3484,52	0	
	14529	3528,88	10	-
	13970	3556,52	20	-
	12654	3560,64	40	-
	13082	3556,92	50	-
EOX	16612	3538,52	0	
	11269	3586,4	40	-
	11144	3636,84	60	-
	10922	3719,96	80	-
	10994	3712,04	90	-
	11000	3668,52	100	-
EMOX	13662	3624,48	70	
	13708	3587,64	90	
	14438	3583,52	40	
	14460	3561,44	30	
	14958	3538,64	60	
	15002	3470,56	20	-
	16647	3425,96	10	-



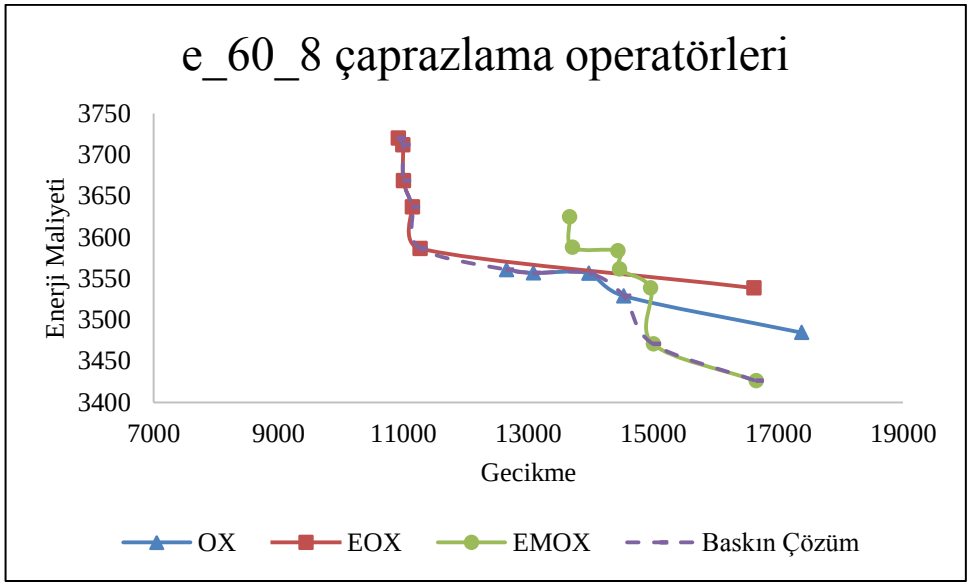
Şekil 4.30. e_60_8 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.31. e_60_8 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



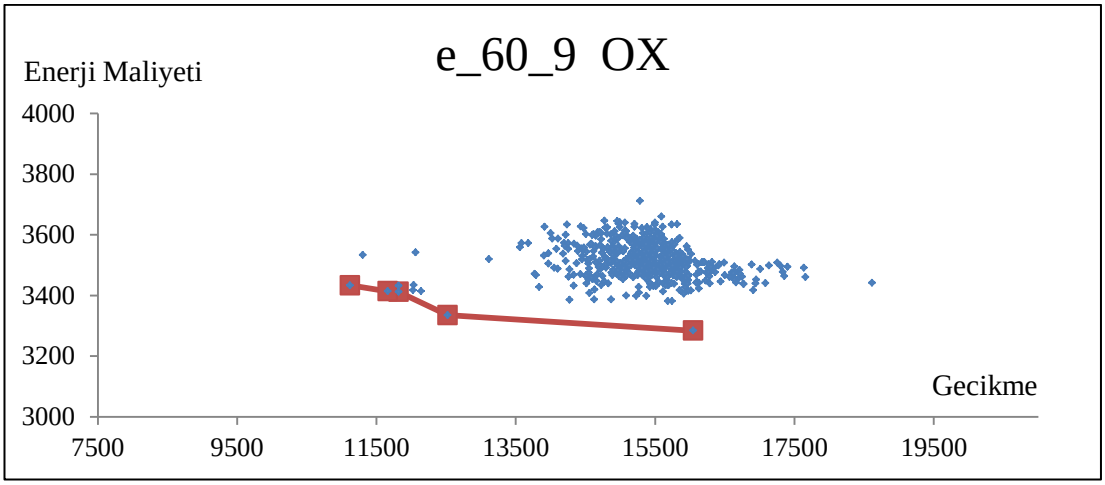
Şekil 4.32. e_60_8 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



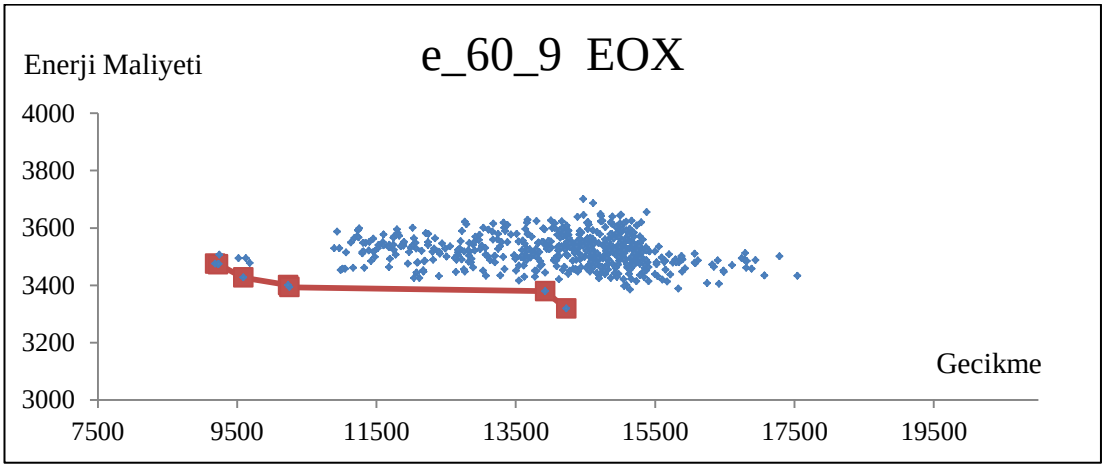
Şekil 4.33. e_60_8 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.14. e_60_9 problemi için baskın çözüm kümesi

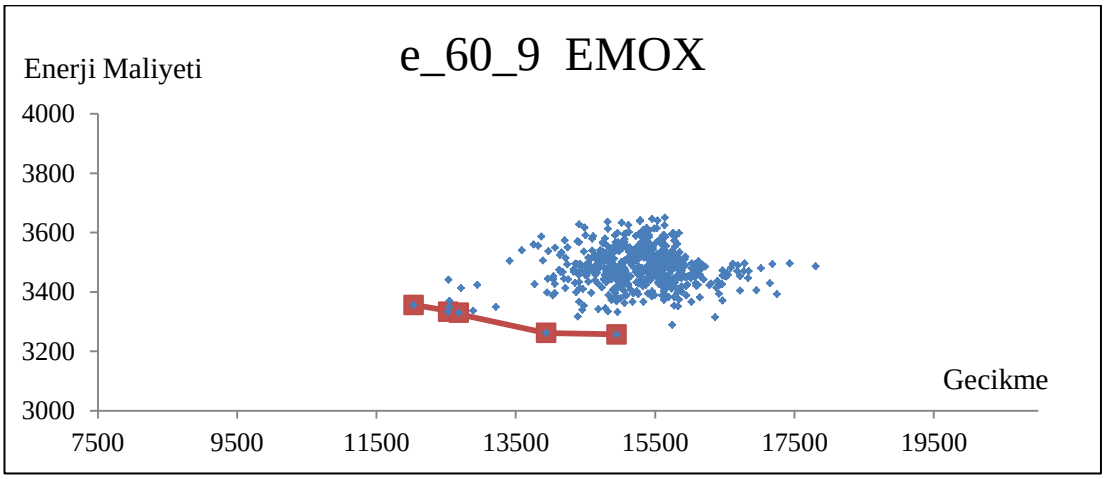
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	16046	3283,84	0	
	12520	3335,28	10	-
	11819	3411,88	20	
	11663	3413,92	40	
	11122	3433,12	90	
EOX	9186	3475,56	60	-
	9229	3473,44	50	-
	9588	3426,92	30	-
	10232	3400,4	20	-
	10251	3393,2	10	-
	13922	3379,52	100	
	14226	3318,4	0	
EMOX	14947	3256,96	0	-
	13935	3261,64	10	-
	12530	3332,8	30	-
	12683	3330,28	80	-
	12034	3356,12	100	-



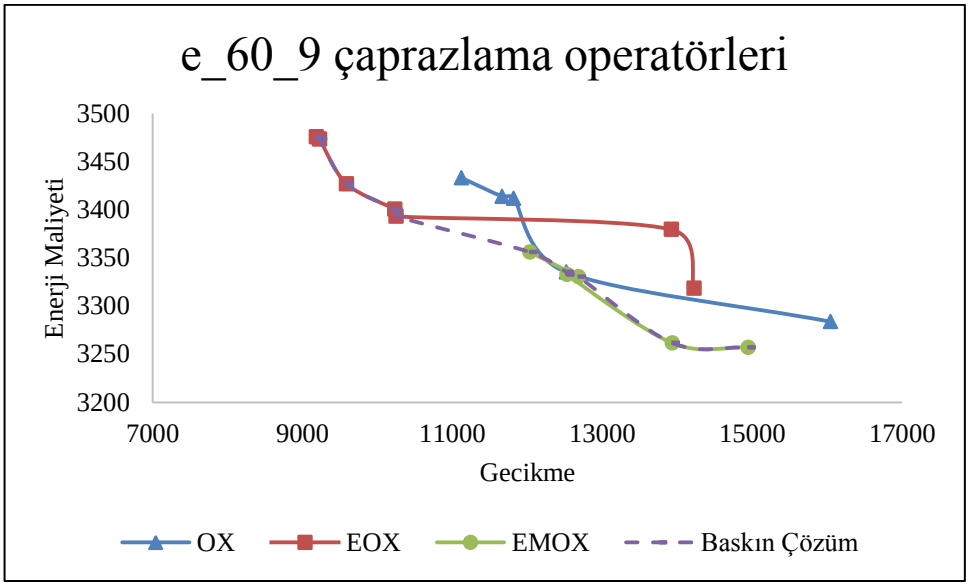
Şekil 4.34. e_60_9 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.35. e_60_9 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



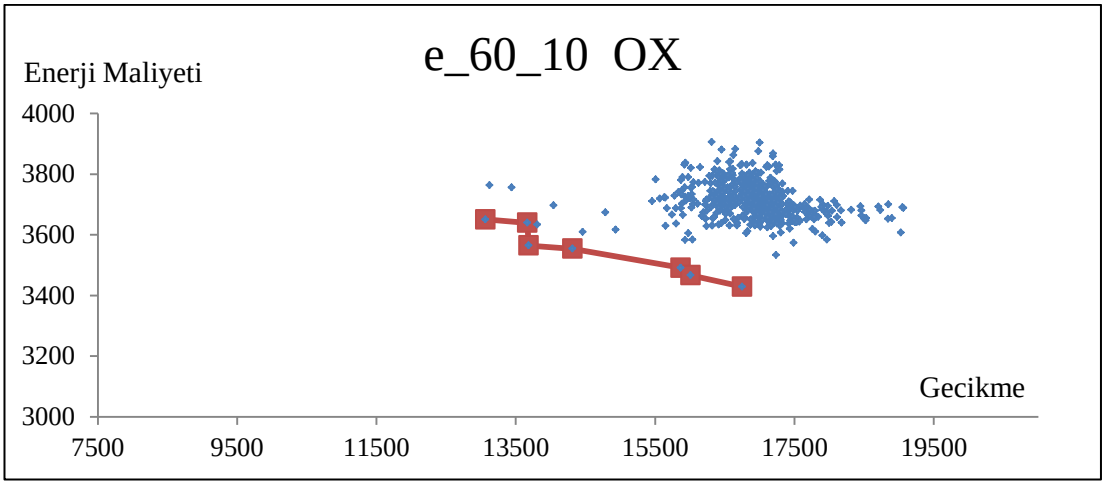
Şekil 4.36. e_60_9 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



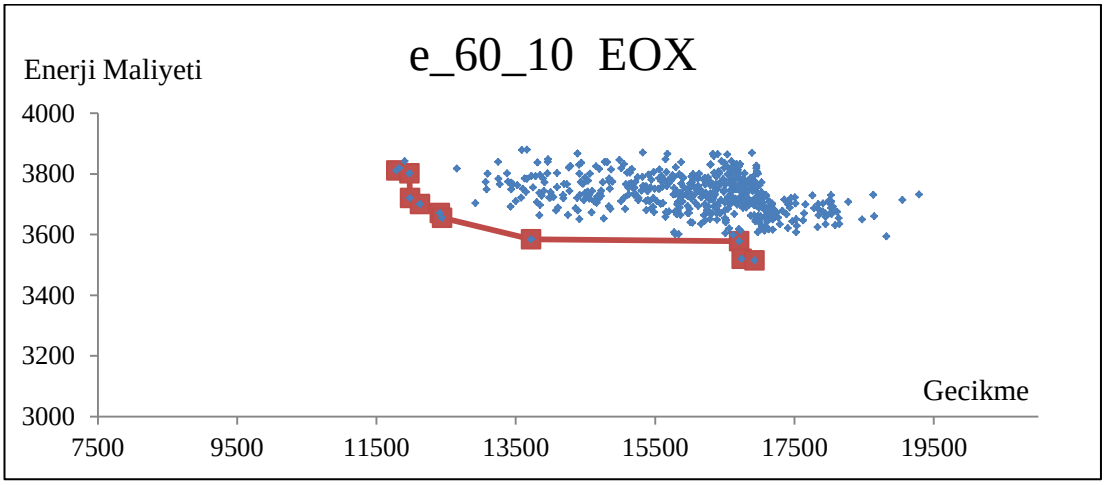
Şekil 4.37. e_60_9 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.15. e_60_10 problemi için baskın çözüm kümesi

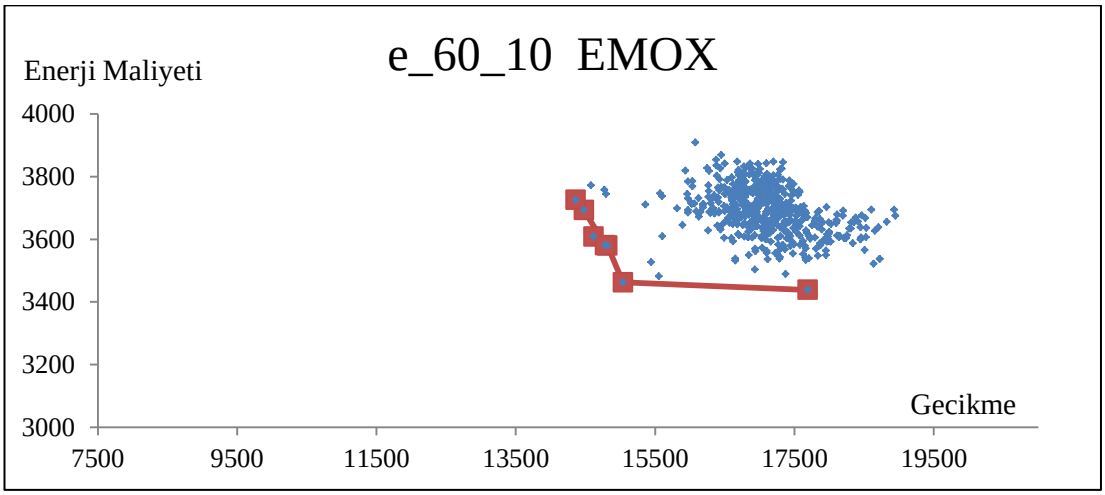
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	13064	3650,64	90	-
	13665	3639,68	70	-
	13686	3564,56	30	-
	14313	3554,36	40	-
	15867	3491,6	10	
	16011	3466,68	20	
	16749	3429,36	0	-
EOX	11785	3810,64	70	-
	11974	3801	100	-
	11982	3719,72	60	-
	12127	3700,44	50	-
	12411	3670,88	40	-
	12444	3654,76	30	-
	13723	3584,12	20	
	16709	3578	10	
	16746	3519,28	10	
	16929	3514,4	0	
EMOX	17691	3438,16	0	
	15039	3462,36	10	-
	14816	3579,64	40	
	14617	3608,36	50	
	14784	3581,2	60	
	14478	3693,2	70	
	14360	3726	90	



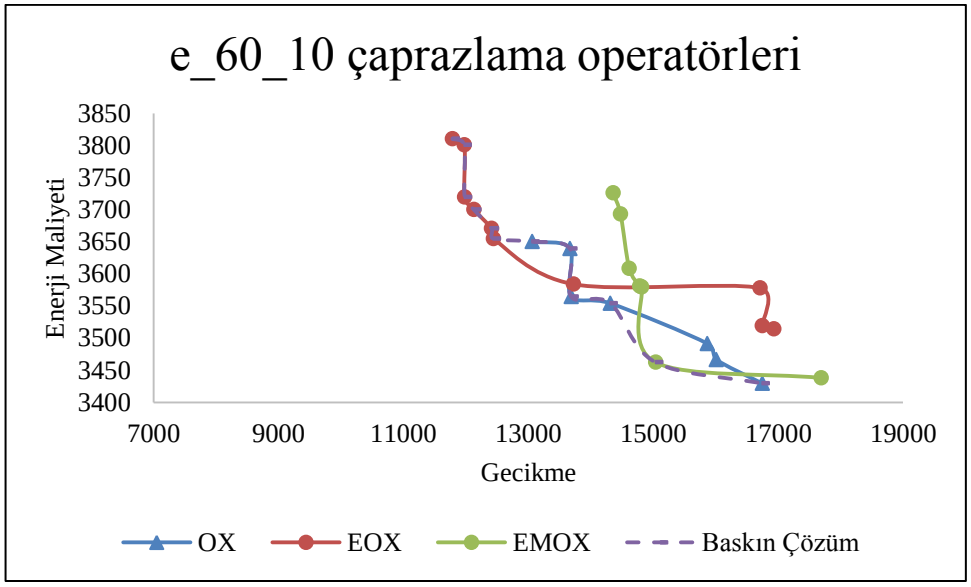
Şekil 4.38. e_60_10 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.39. e_60_10 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.40. e_60_10 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.41. e_60_10 problemi için çaprazlama operatörlerinin kıyaslaması

60 işlik problemlerin çözümleri ile ilgili genel durum Tablo 4.16'da ve 4.17'de en iyi ve en kötü sonucu veren yöntemler işaretlenerek gösterilmiştir. Tacettin tarafından enerji maliyetinin enazlanmasına yönelik Diferansiyel Gelişim Algoritması (DGA) kullanılarak yapılan çalışmada aynı problemler çok amaçlı olarak çözüdürülmüştür. Ancak, DGA ile çözülen problemler için kullanılan termin zamanları orijinal değerleridir. Orijinal termin zamanı değerleriyle problemlerin zorluk katsayısı -0,25'dir, dolayısıyla DGA'nın sonuçları ile yapılacak kıyaslama sadece enerji maliyetlerinin minimizasyonu amacı için olacaktır. TMEMM yönteminin 60 işlik problemlerin yarısında enerji maliyeti olarak DGA'dan daha iyi sonuç verdiği görülmektedir. Enerji maliyetine göre en iyi çözüm değerleri Tablo 4.16'da gösterilmiştir. Enerji maliyeti olarak genetik algoritma operatörleri arasında en iyi sonuçları EMOX çaprazlama operatörü verirken, bu yöntemle elde edilen en iyi toplam gecikme zamanı değerleri diğer yöntemlerin hepsinden daha kötüdür. Toplam gecikme zamanına göre en iyi çözüm değerleri ise Tablo 4.18'de gösterilmiştir. Toplam gecikme zamanı için elde edilen en iyi sonuçlar değerlendirildiğinde 10 problemin 7'sinde EOX çaprazlama operatörünün en iyi sonucu verdiği görülebilir. Dolayısıyla karar vericiye, alternatifleri değerlendirirken toplam gecikme zamanının daha kritik olduğu zamanlarda EOX çaprazlama operatörünü kullanması, maliyetlerin ön plana çıktığı durumlarda ise EMOX çaprazlama operatörünü kullanması önerilebilir. OX çaprazlama operatörü ise herhangi bir amaç için şekillendirilmediğinden ortada çözümler çıkarmıştır. Her bir çaprazlama

operatörünün baskın çözümleri toplamda değerlendirildiğinde EOX'un ve EMOX'un baskın çözümlerinin OX'in ürettiği baskın çözümlerden daha da baskın olduğu gözlemlenmiştir. Tüm çözümler içindeki baskın çözüm kümelerinde OX operatörü 22 baskın çözüm üretebilirken, EOX 53, EMOX ise 36 baskın çözüm üretmiştir.

Tablo 4.16. 60 işlik problemlerin en iyi enerji maliyet değerleri

Problem Adı	TMEMM	DGA	ga-ox	ga-eox	ga-emox
e_60_1	3251.12	3035	3170,4	3200,2	3103,1
e_60_2	3202,6	3298	3394,04	3442,76	3348,8
e_60_3	3251,44	3212	3315,24	3338,88	3284,92
e_60_4	3207,82	3306	3409,96	3415,44	3348,28
e_60_5	3246,08	3403	3513,6	3534,92	3447,64
e_60_6	3284,04	3236	3343,68	3348,2	3288,04
e_60_7	3319,2	3244	3316,64	3328,52	3291,48
e_60_8	3248	3373	3484,52	3538,52	3425,96
e_60_9	3312	3189	3283,84	3318,4	3256,96
e_60_10	3324,44	3356	3429,36	3514,4	3418,16

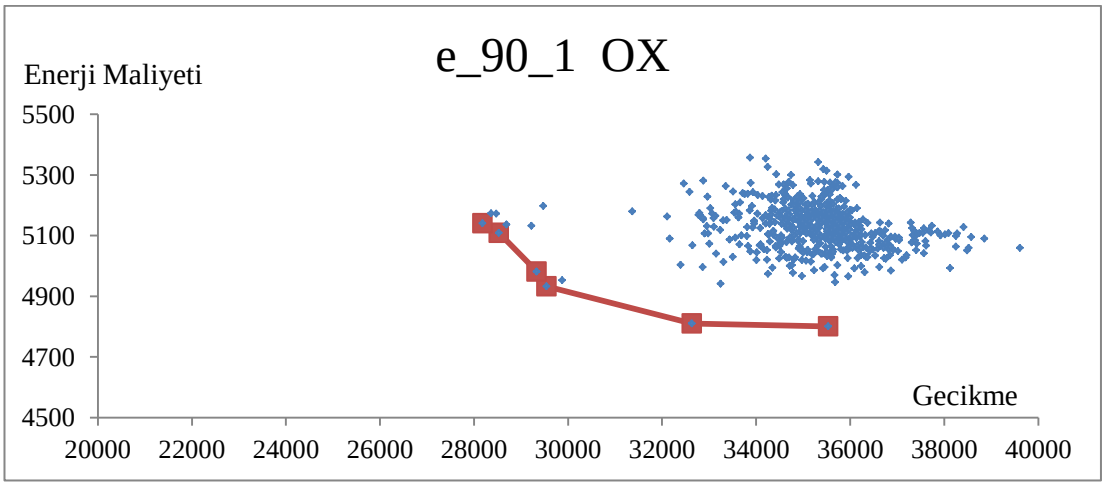
Tablo 4.17. 60 işlik problemlerin en iyi toplam gecikme zamanı değerleri

Problem Adı	TMTGZM	ga-ox	ga-eox	ga-emox
e_60_1	7721	9875	7840	11198
e_60_2	11877	13297	11236	14066
e_60_3	11232	12595	10848	13457
e_60_4	9209	11150	9227	12154
e_60_5	10321	12181	10049	13108
e_60_6	12701	13710	12282	14802
e_60_7	9560	11068	11350	12331
e_60_8	11444	12654	10922	13662
e_60_9	9259	11122	9186	12034
e_60_10	12414	13064	11785	14360

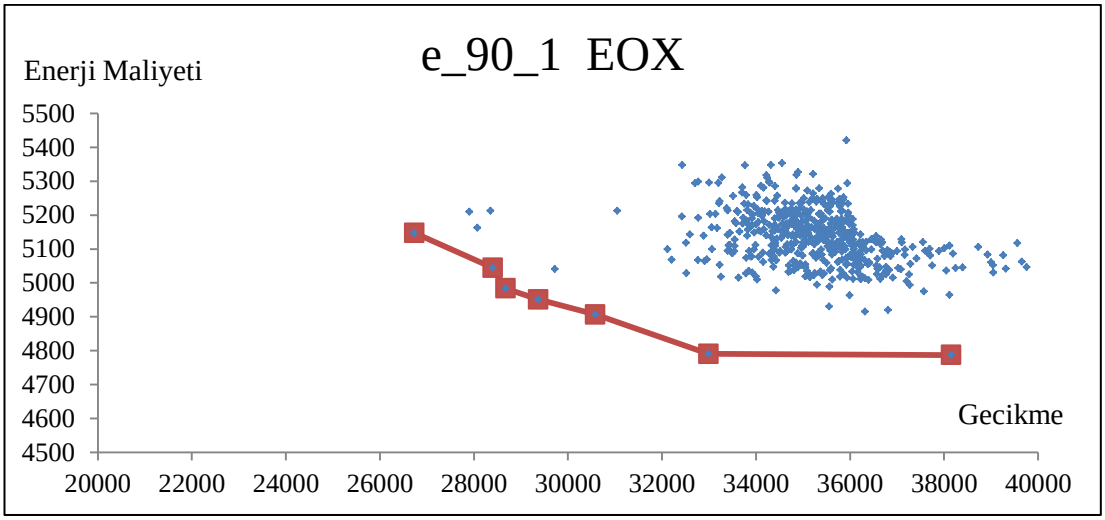
Alttaki grafiklerde ve tablolarda ise 90 işlik problemlerin baskın çözüm kümeleri ve her α değeri için en iyi 50, toplamda ise tüm α değerleri için en iyi 550 çözümün dağılımı gösterilmiştir. Tablolarda tüm çaprazlama yöntemleri beraber düşünüldüğünde baskın olan çözümler işaretlenmiştir.

Tablo 4.18. e_90_1 problemi için baskın çözüm kümesi

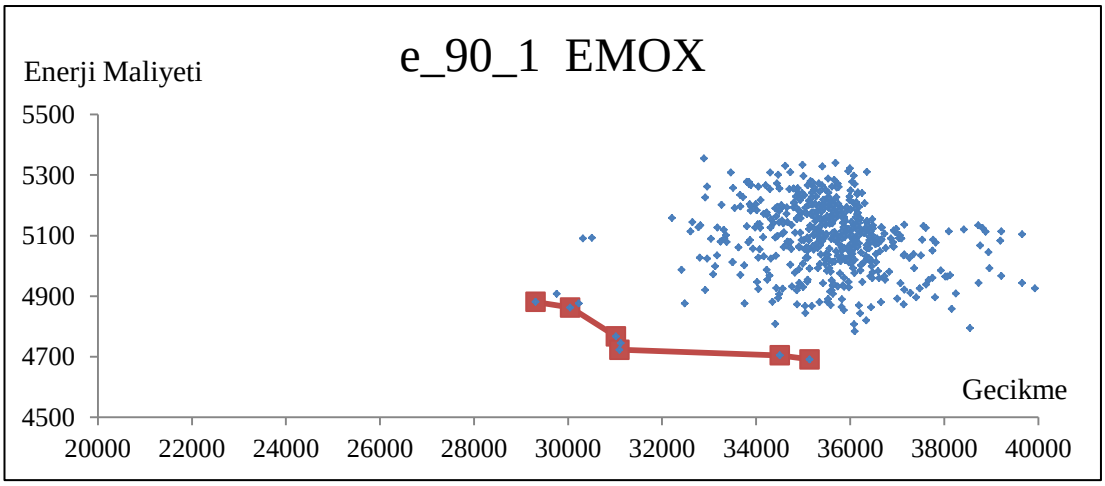
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	35534	4800,72	0	
	32636	4810,52	10	
	29546	4932,92	20	
	29333	4981,8	30	
	28530	5108,96	70	
	28183	5140,96	100	-
EOX	26729	5147,4	50	-
	28399	5044,68	70	-
	28671	4983,72	30	-
	29368	4951,24	60	
	30582	4906,72	20	
	32989	4790,52	10	
	38150	4787,36	0	
EMOX	29310	4881,32	50	-
	30049	4862,36	70	-
	31017	4767,48	40	-
	31096	4723	30	-
	34508	4704,72	10	-
	35139	4691,16	0	-



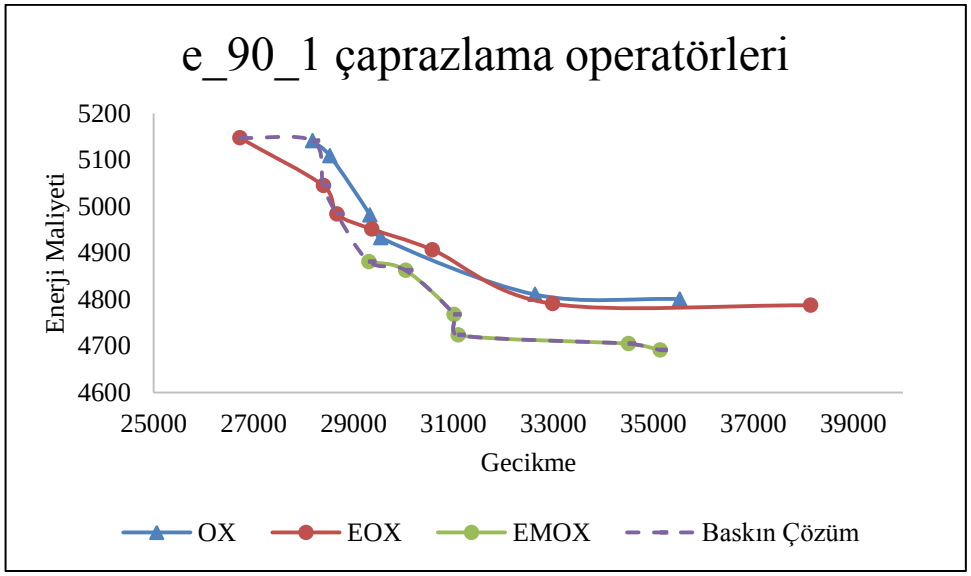
Şekil 4.42. e_90_1 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.43. e_90_1 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



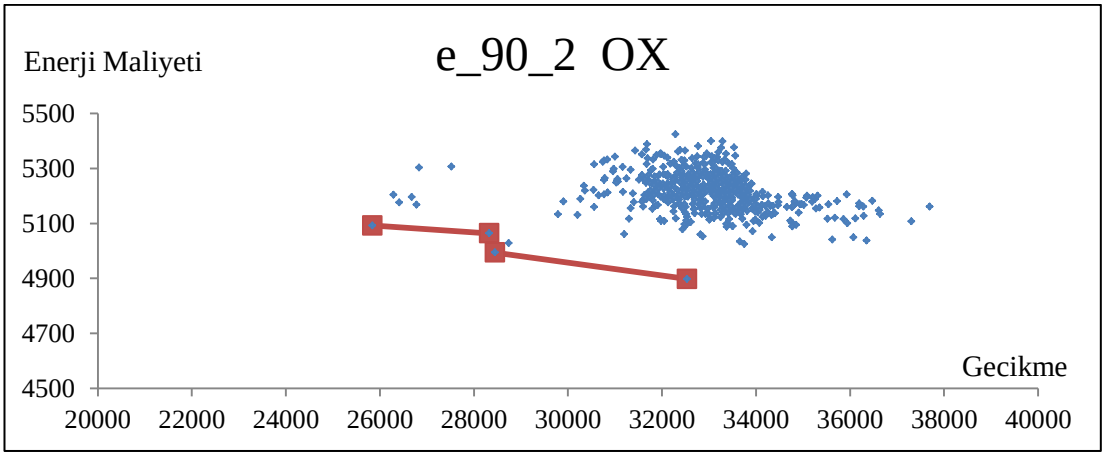
Şekil 4.44. e_90_1 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



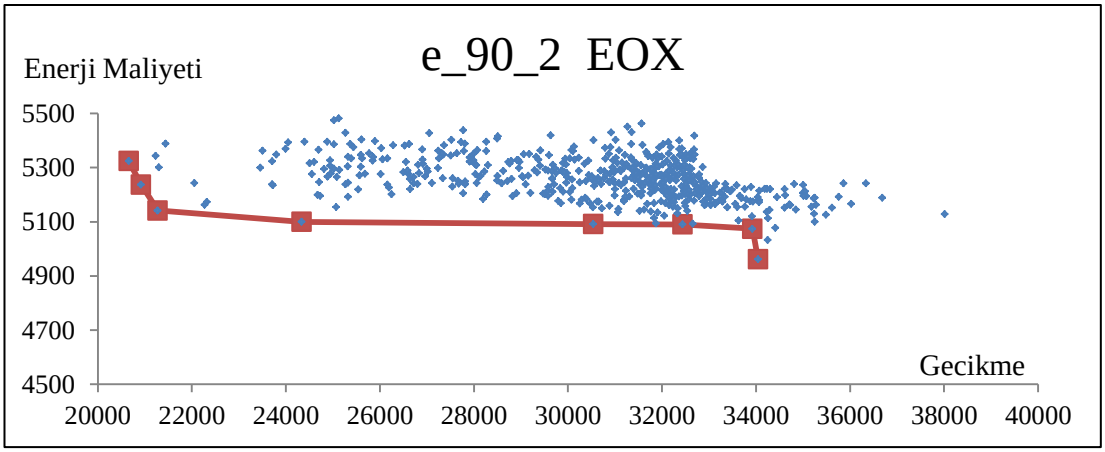
Şekil 4.45. e_90_1 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.19. e_90_2 problemi için baskın çözüm kümesi

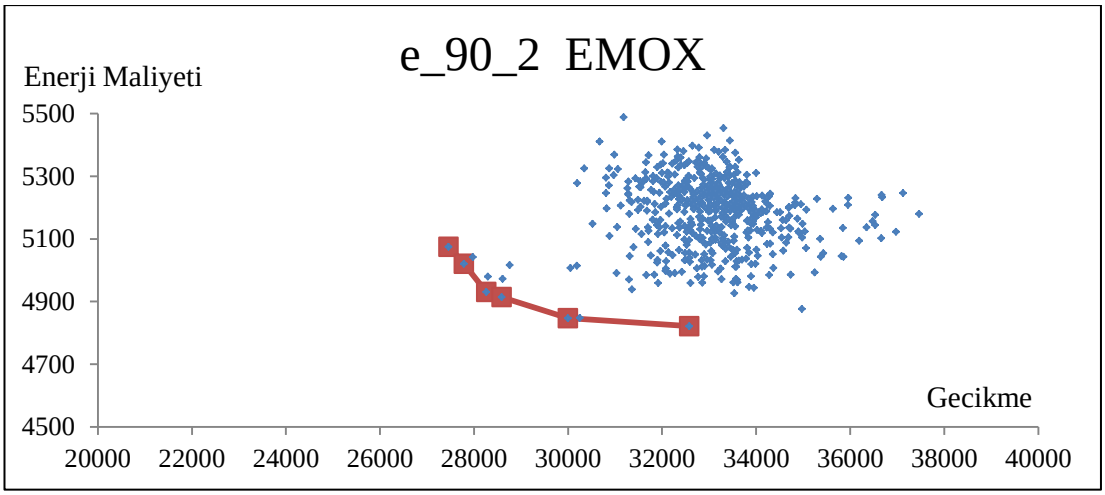
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	25838	5092,36	50	-
	28326	5063,72	30	
	28448	4993,92	10	
	32530	4897,52	0	
EOX	20656	5324,16	60	-
	20918	5237,72	80	-
	21275	5141,84	70	-
	24336	5099,88	10	-
	30541	5091,68	80	
	32438	5090,56	90	
	33925	5074,64	40	
	34042	4962,32	0	
EMOX	27461	5074,64	90	-
	27788	5020,32	60	-
	28262	4930,56	40	-
	28592	4913,96	20	-
	29996	4846,76	30	-
	32579	4821,56	0	-



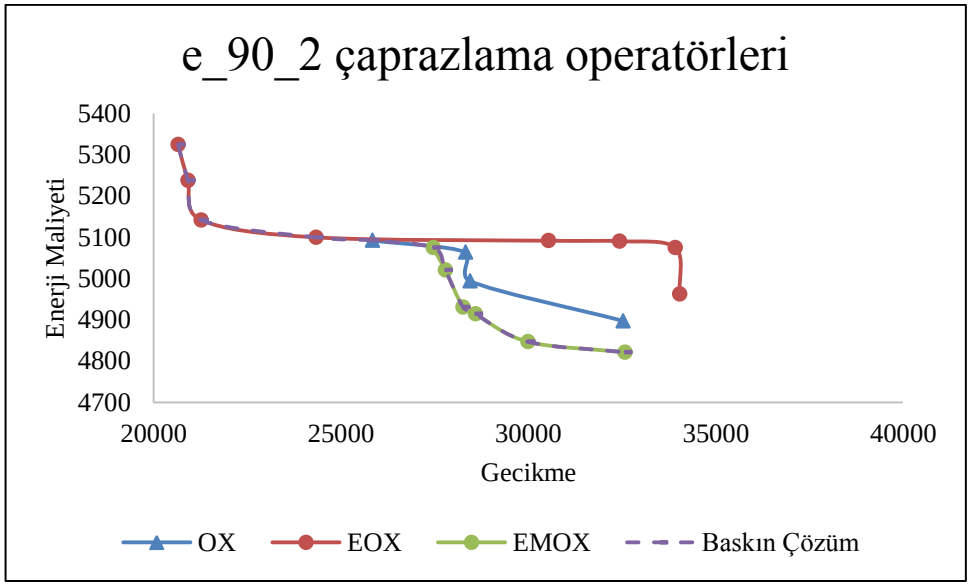
Şekil 4.46. e_90_2 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.47. e_90_2 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



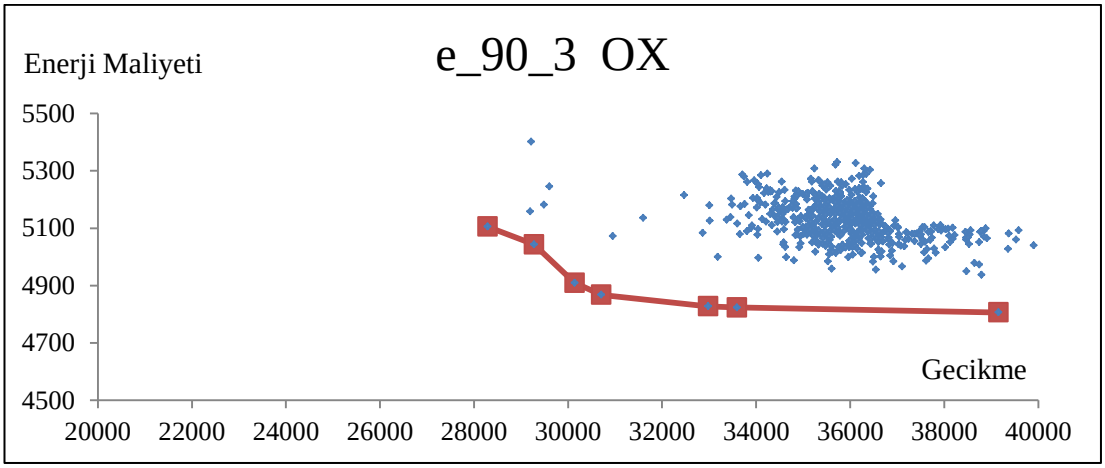
Şekil 4.48. e_90_2 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



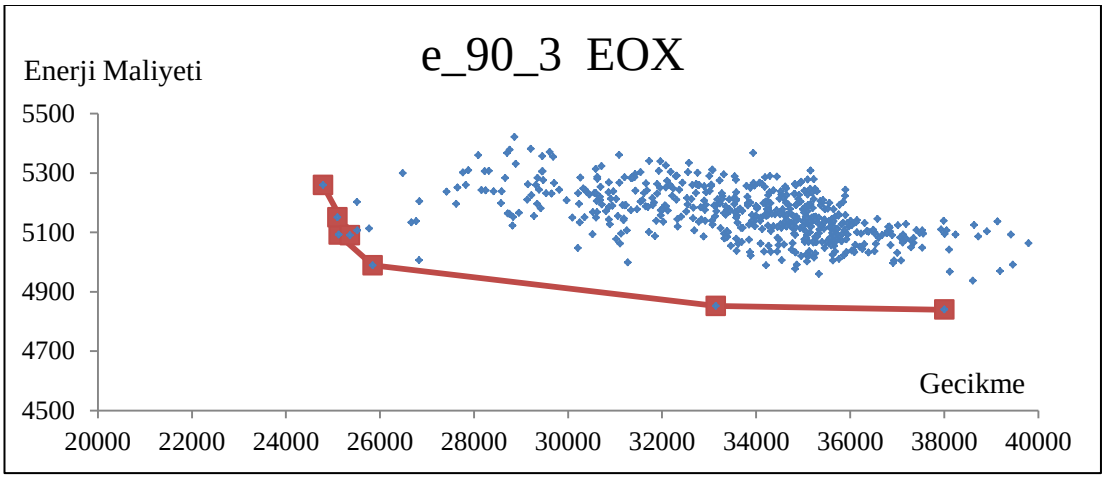
Şekil 4.49. e_90_2 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.20. e_90_3 problemi için baskın çözüm kümesi

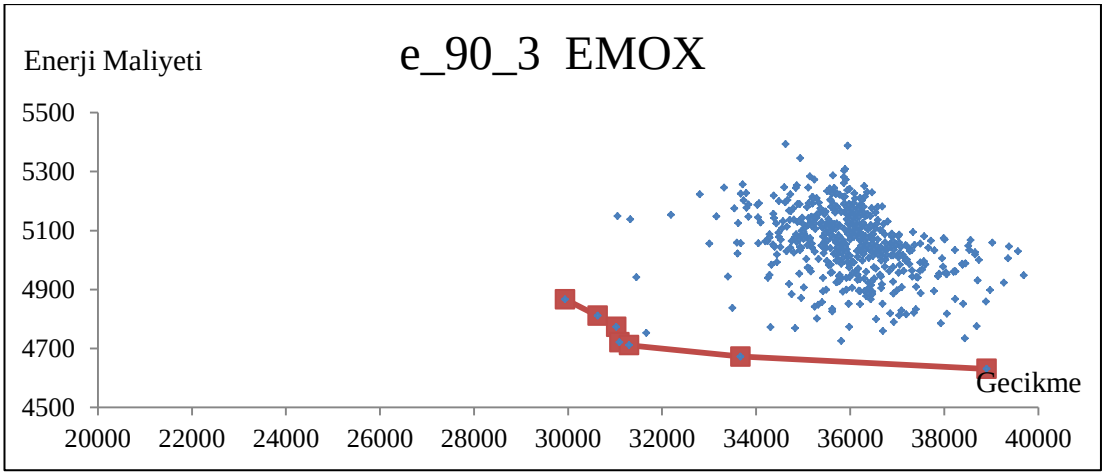
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	28288	5105,4	60	
	29274	5043,72	100	
	30141	4909,64	40	
	30708	4867,72	30	
	32982	4827,8	20	
	33593	4823,52	10	
	39158	4806,52	0	
EOX	38008	4839,44	0	
	33145	4852,04	10	
	25848	4989	30	-
	25124	5091,6	50	-
	25095	5150,52	80	-
	25366	5089,64	90	-
	24789	5258,76	100	-
EMOX	29936	4867,12	60	-
	30631	4811,2	80	-
	31026	4773,84	40	-
	31096	4720,76	50	-
	31296	4711,12	20	-
	33668	4672,52	10	-
	38906	4631,12	0	-



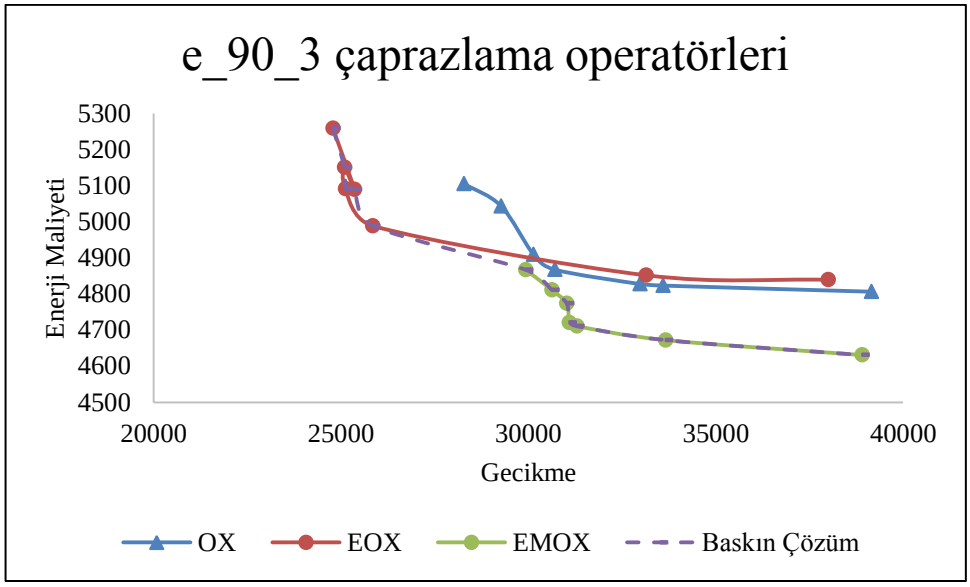
Şekil 4.50. e_90_3 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.51. e_90_3 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



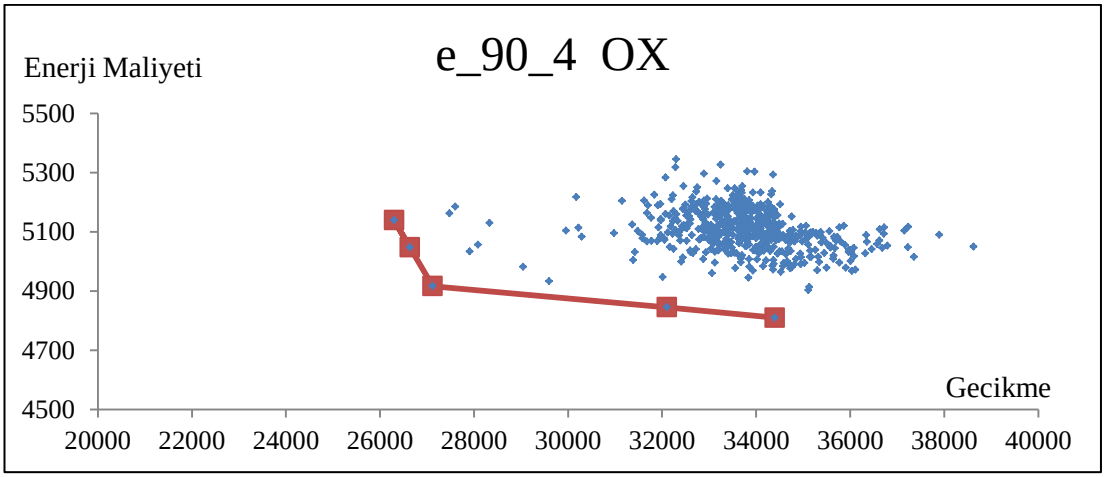
Şekil 4.52. e_90_3 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



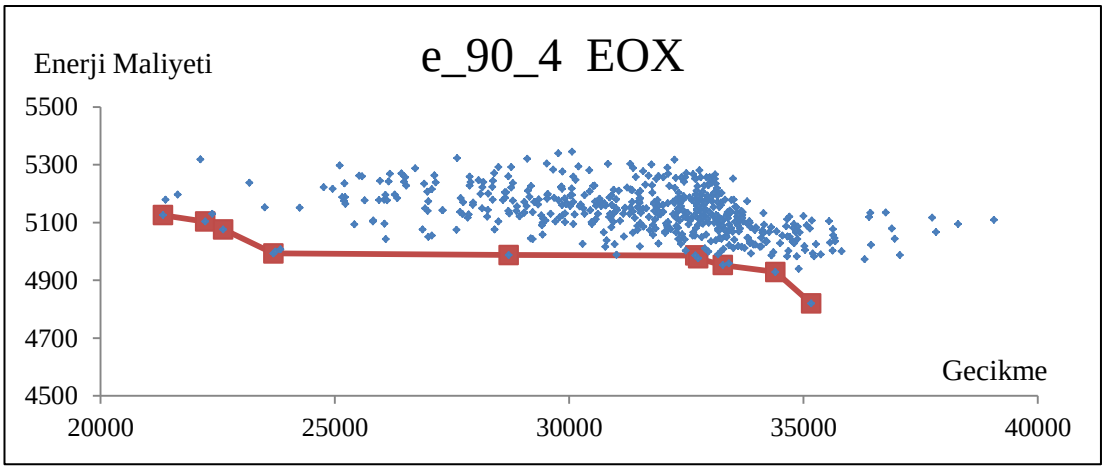
Şekil 4.53. e_90_3 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.21. e_90_4 problemi için baskın çözüm kümesi

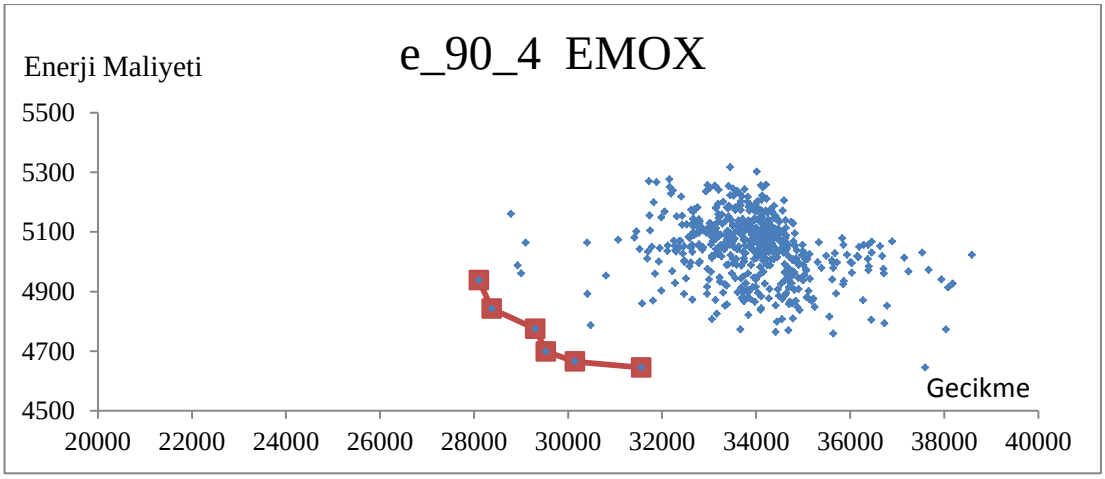
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	34394	4810	0	
	32103	4845,4	10	
	27122	4916,76	60	-
	26301	5139,2	80	
	26637	5047,44	90	
EOX	21331	5125,48	80	-
	22231	5103,36	50	-
	22619	5076,24	60	-
	23687	4993,36	10	-
	28713	4987,72	100	
	32692	4985,88	60	
	32752	4976,24	10	
	33279	4952,84	100	
	34402	4928,84	20	
	35166	4820,12	0	
EMOX	28103	4937,32	70	
	28381	4841,8	100	-
	29305	4774,4	40	-
	29530	4698,36	20	-
	30150	4664,36	30	-
	31555	4644,6	10	-



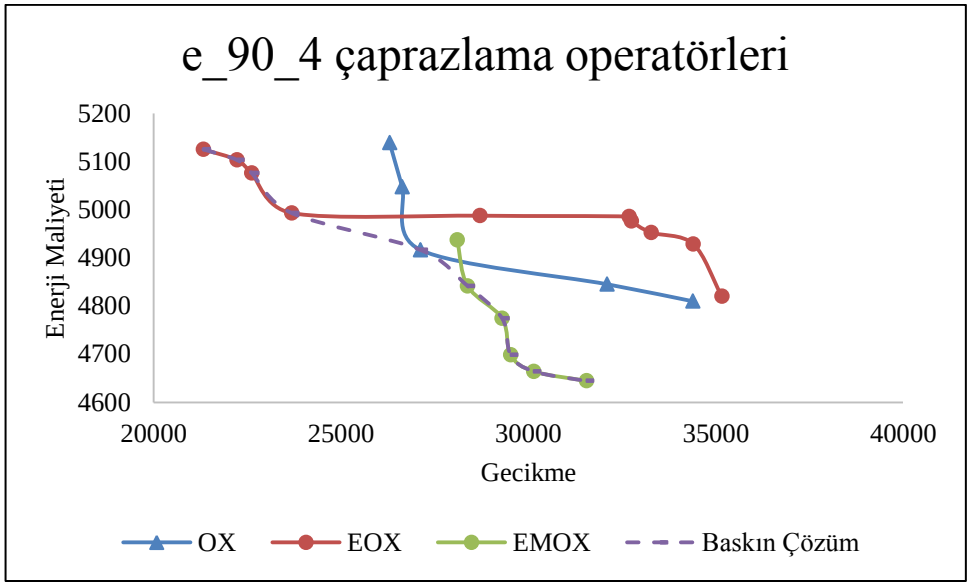
Şekil 4.54. e_90_4 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.55. e_90_4 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



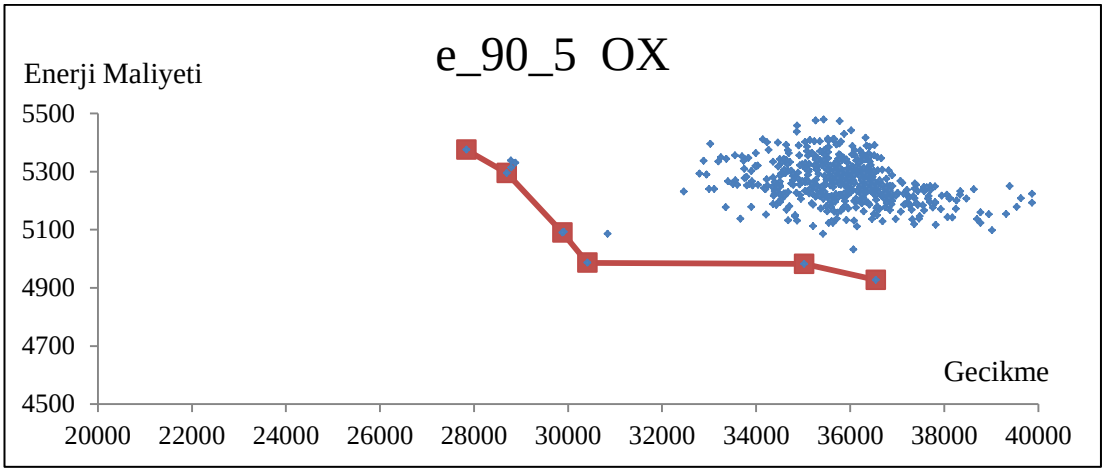
Şekil 4.56. e_90_4 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



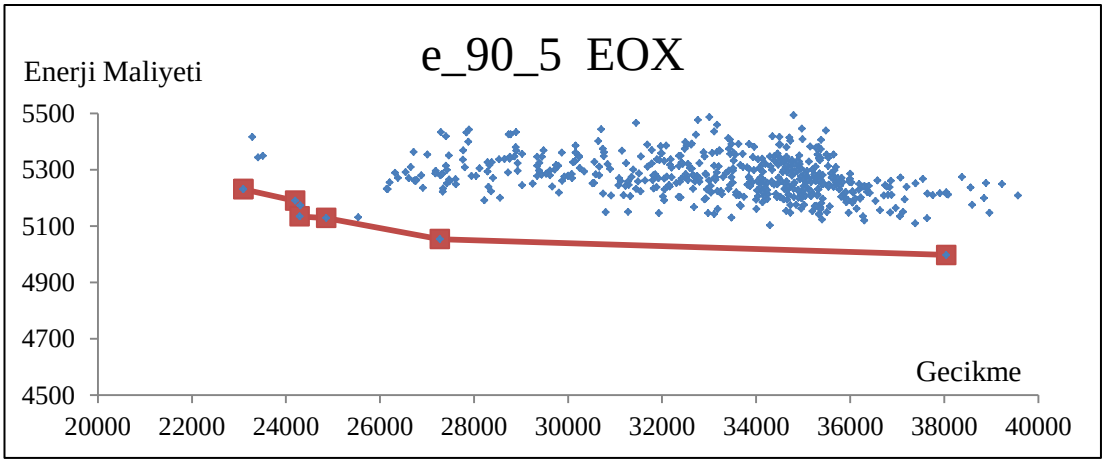
Şekil 4.57. e_90_4 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.22. e_90_5 problemi için baskın çözüm kümesi

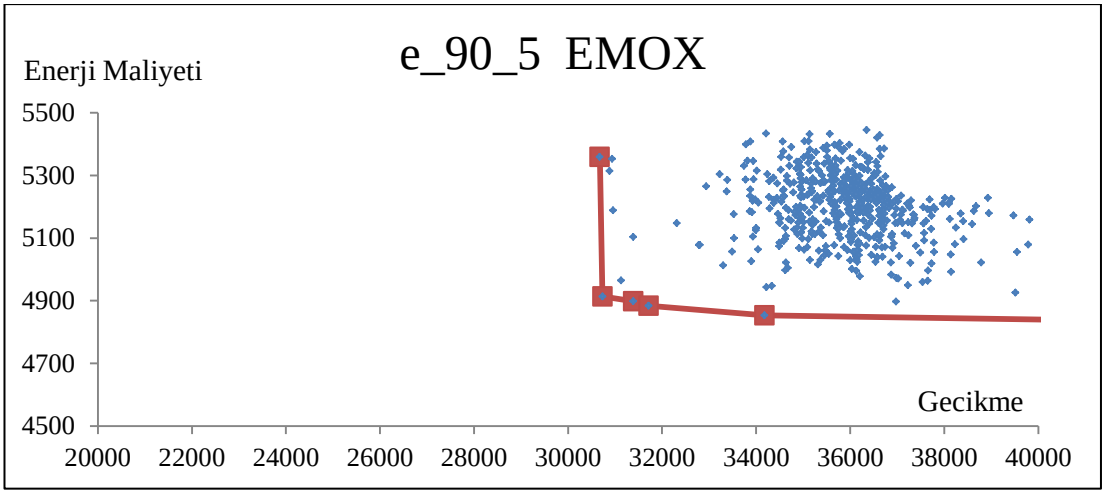
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	27839	5375,04	80	
	28697	5295,16	90	
	29881	5090,24	50	
	30414	4986,08	20	-
	35021	4982,44	10	
	36547	4927,2	0	
EOX	23094	5230,88	80	-
	23094	5230,88	80	-
	24197	5189,92	50	-
	24292	5134,68	40	-
	24859	5128,64	30	-
	27275	5053,6	10	-
	38049	4997,84	0	
EMOX	30675	5358,88	70	
	30732	4913,6	30	-
	31390	4898,24	40	-
	31712	4883,6	20	-
	34179	4853,2	10	-
	40449	4839,12	0	-



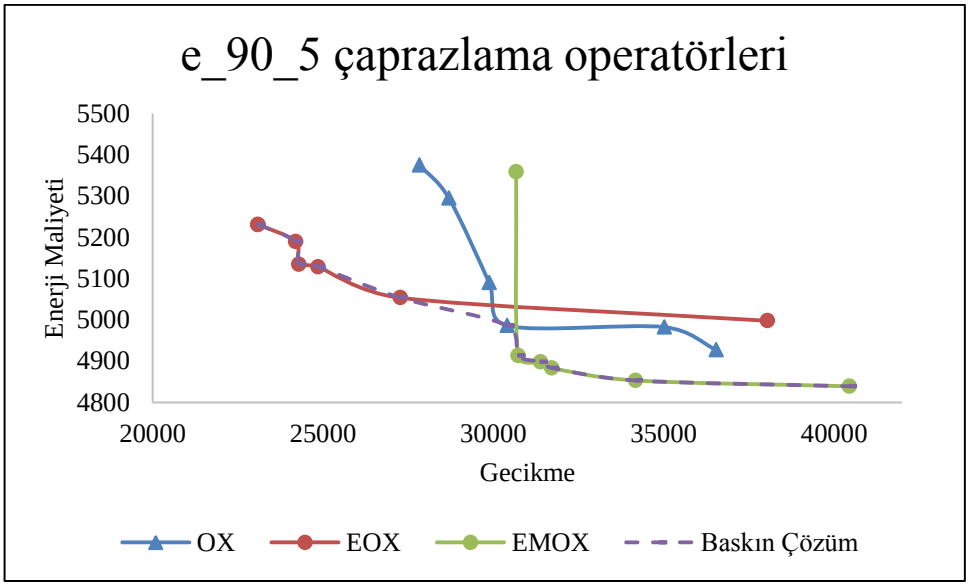
Şekil 4.58. e_90_5 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.59. e_90_5 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



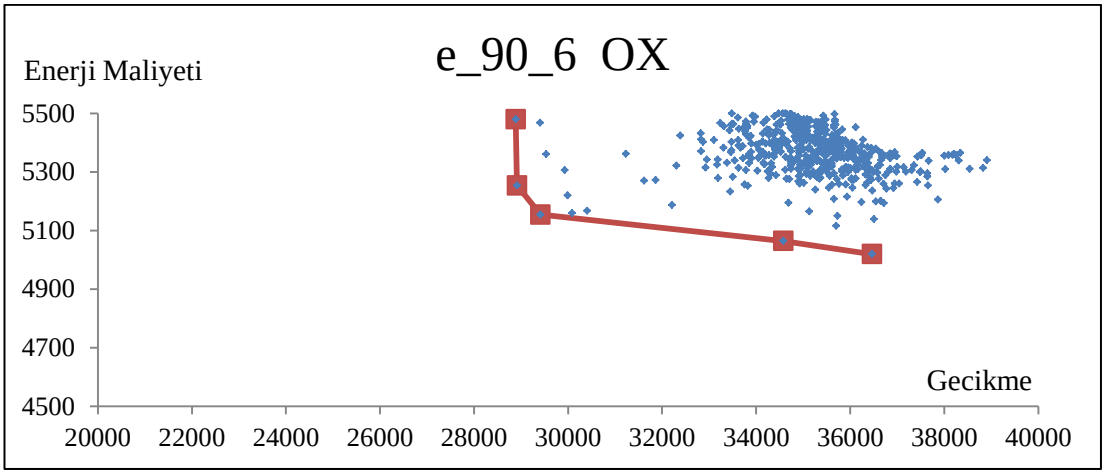
Şekil 4.60. e_90_5 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



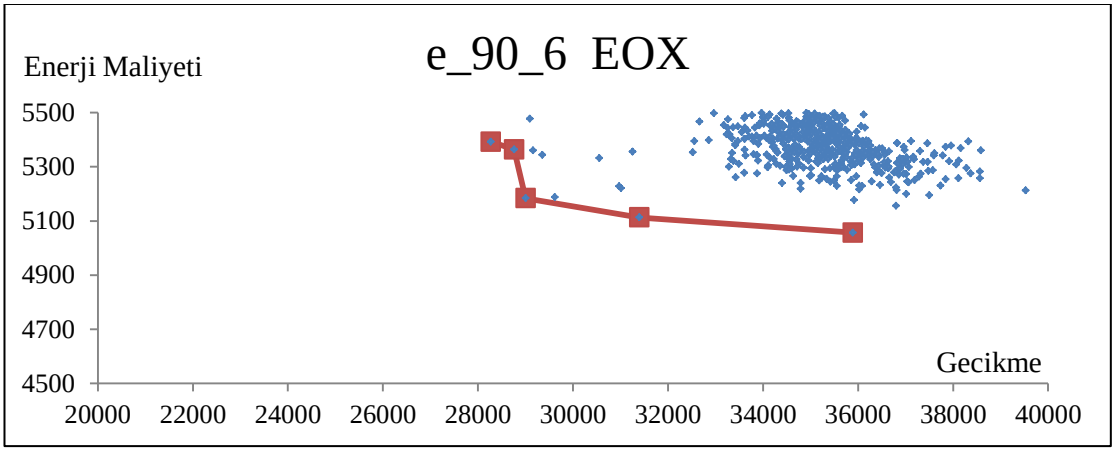
Şekil 4.61. e_90_5 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.23. e_90_6 problemi için baskın çözüm kümesi

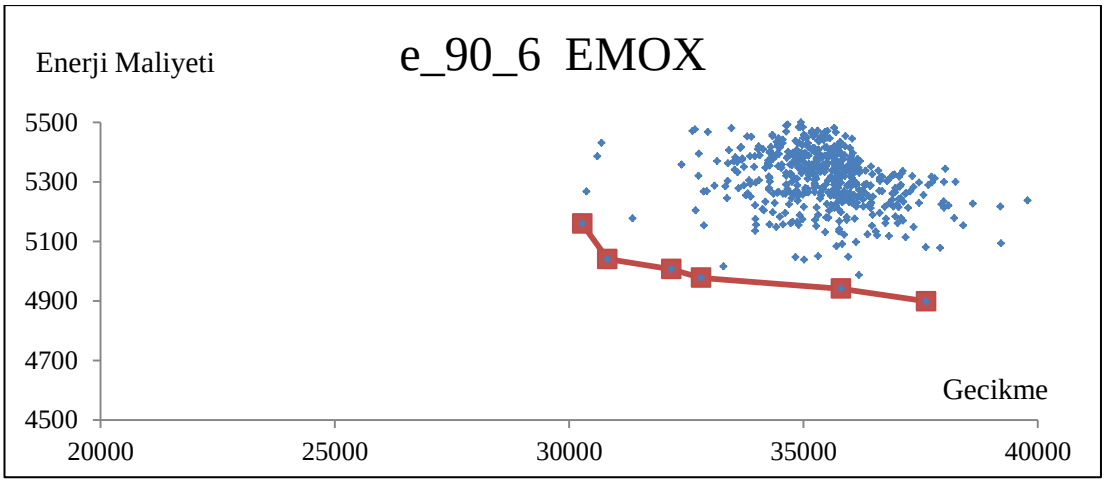
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	28887	5479,48	90	
	28916	5254,08	50	-
	29416	5154,24	80	-
	34582	5064,2	10	
	36467	5019,36	0	
EOX	35894	5056,84	0	
	31398	5113,04	10	
	29003	5183,88	30	-
	28763	5364	60	-
	28271	5392,28	90	-
EMOX	37619	4898,6	0	-
	35802	4940,84	10	-
	32813	4977,72	20	-
	32181	5006,68	50	-
	30817	5040,4	60	-
	30279	5159,52	90	



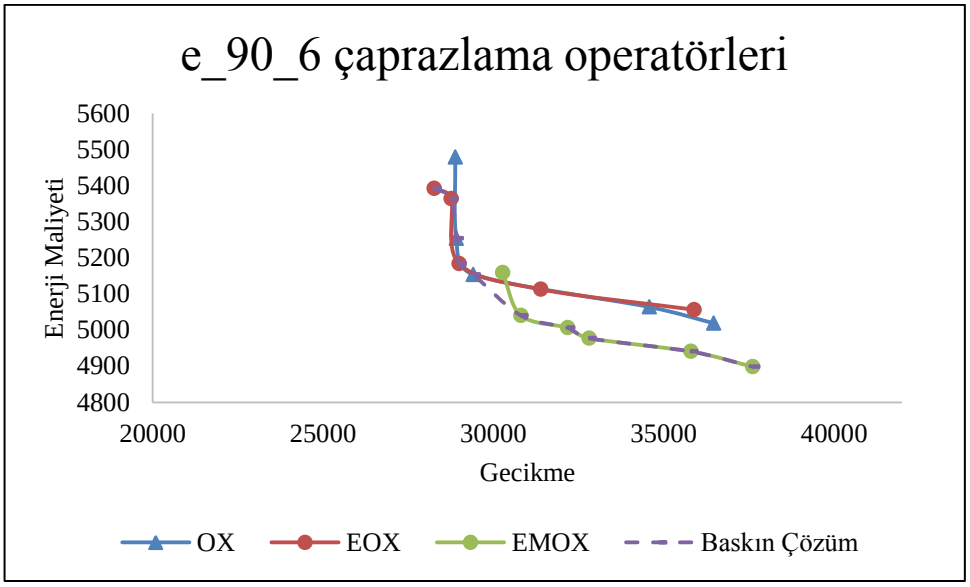
Şekil 4.62. e_90_6 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.63. e_90_6 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



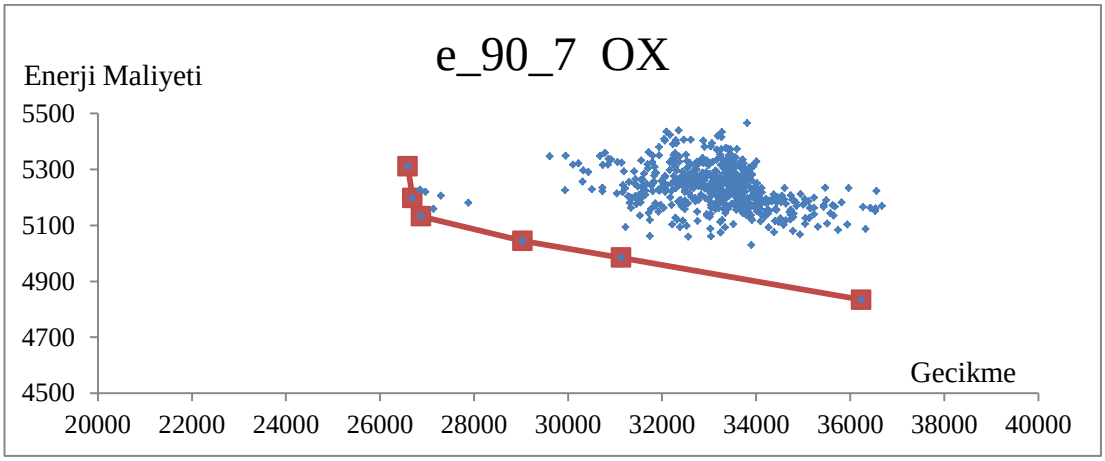
Şekil 4.64. e_90_6 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



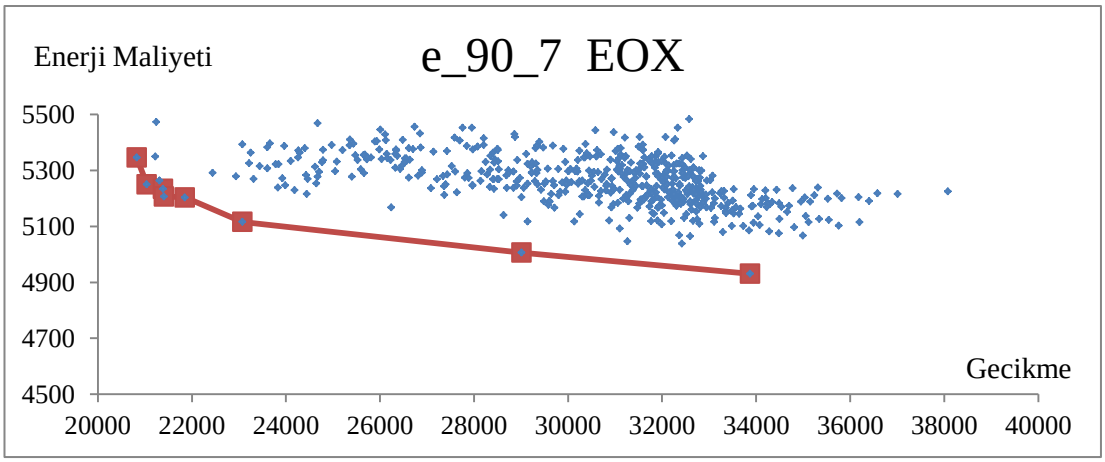
Şekil 4.65. e_90_6 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.24. e_90_7 problemi için baskın çözüm kümesi

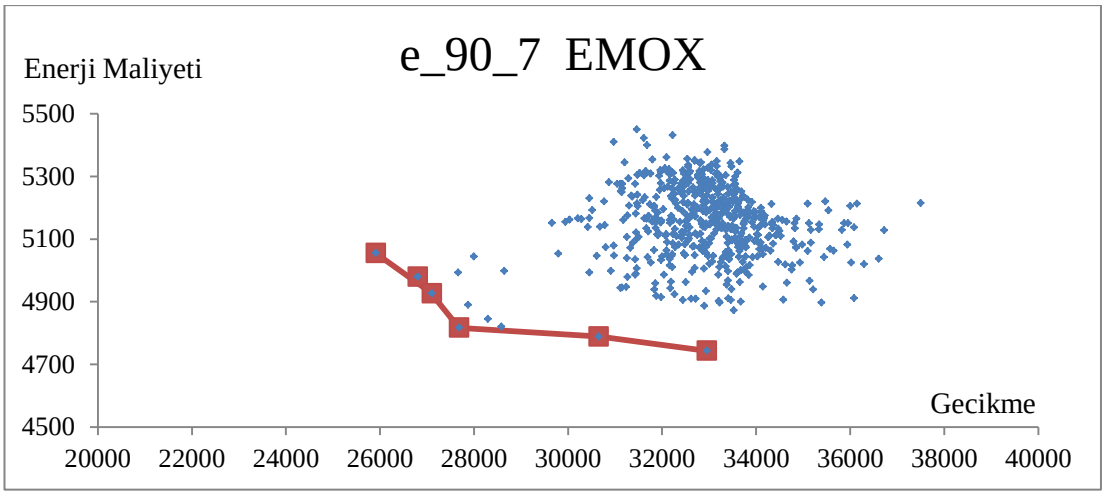
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	26591	5310,96	100	
	26693	5197,64	70	
	26875	5132,12	60	
	29030	5044,6	20	
	31126	4984,64	10	
	36238	4833,8	0	
EOX	20826	5345,92	80	-
	21041	5250,24	60	-
	21388	5234,52	40	-
	21411	5206,92	50	-
	21852	5203,72	30	-
	23073	5116,04	20	-
	29013	5006,2	10	
	33873	4930,6	0	
EMOX	32956	4743,12	0	-
	30654	4788,72	10	-
	27681	4816,92	20	-
	26804	4979,88	60	-
	27104	4926,68	70	-
	25914	5055,08	90	-



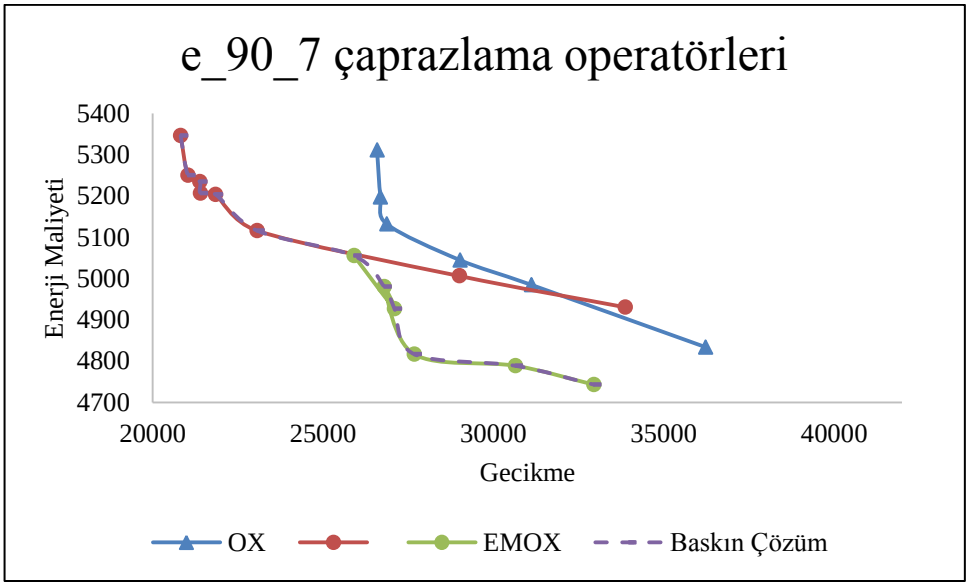
Şekil 4.66. e_90_7 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.67. e_90_7 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



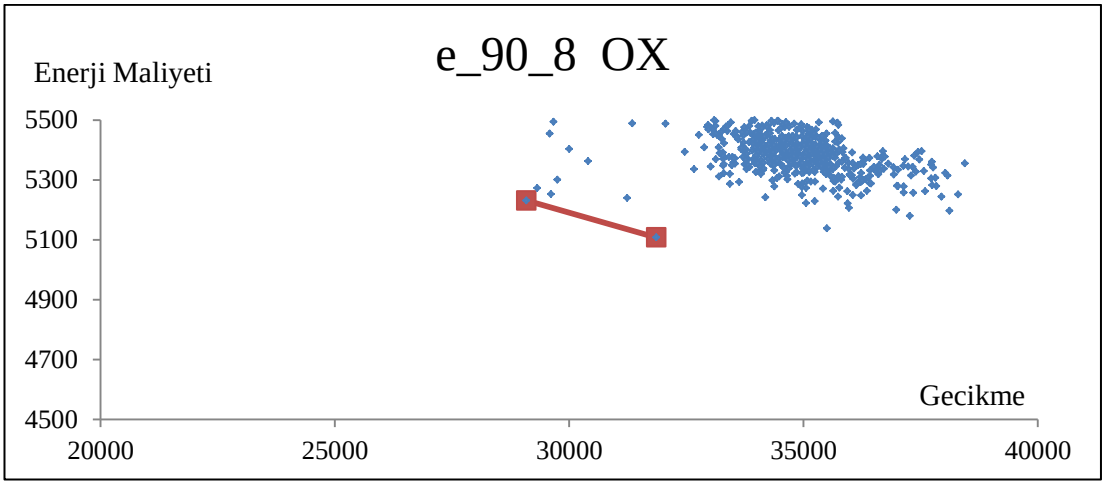
Şekil 4.68. e_90_7 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



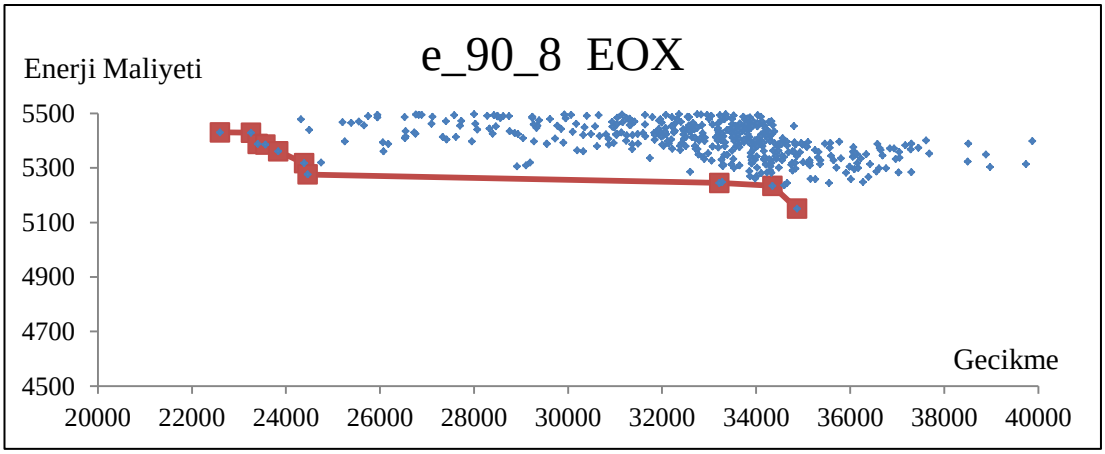
Şekil 4.69. e_90_7 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.25. e_90_8 problemi için baskın çözüm kümesi

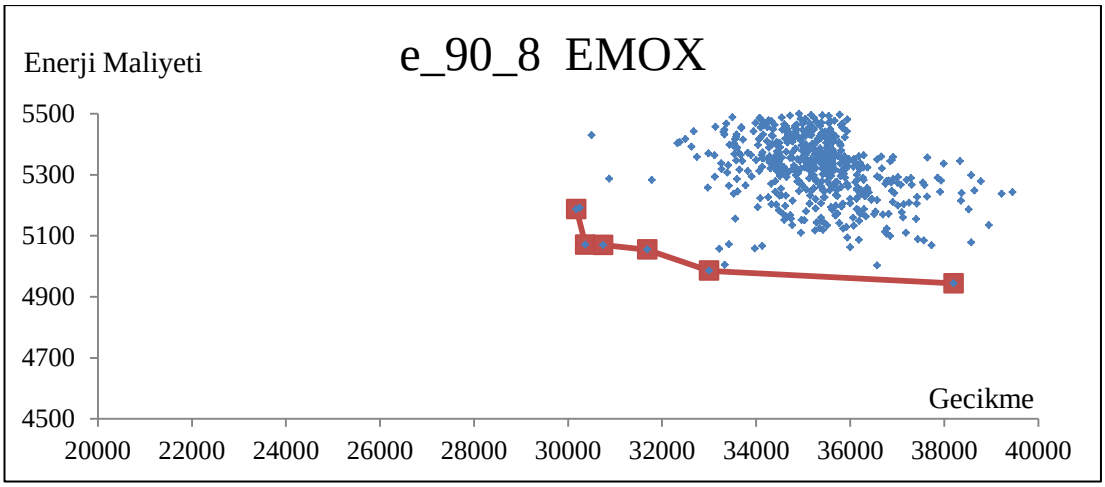
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	31860	5107,76	10	
	29089	5231,28	40	-
EOX	22601	5430,84	80	-
	23261	5429,2	70	-
	23401	5387,64	40	-
	23564	5385,76	50	-
	23838	5361,24	60	-
	24389	5317,4	30	-
	24465	5276,24	20	-
	33221	5244,92	40	
	34349	5234,6	80	
	34870	5150,64	0	
EMOX	30177	5187,32	90	-
	30369	5070,88	40	-
	30746	5069,92	70	-
	31685	5054,8	30	-
	33003	4985,52	20	-
	38205	4944,04	0	-



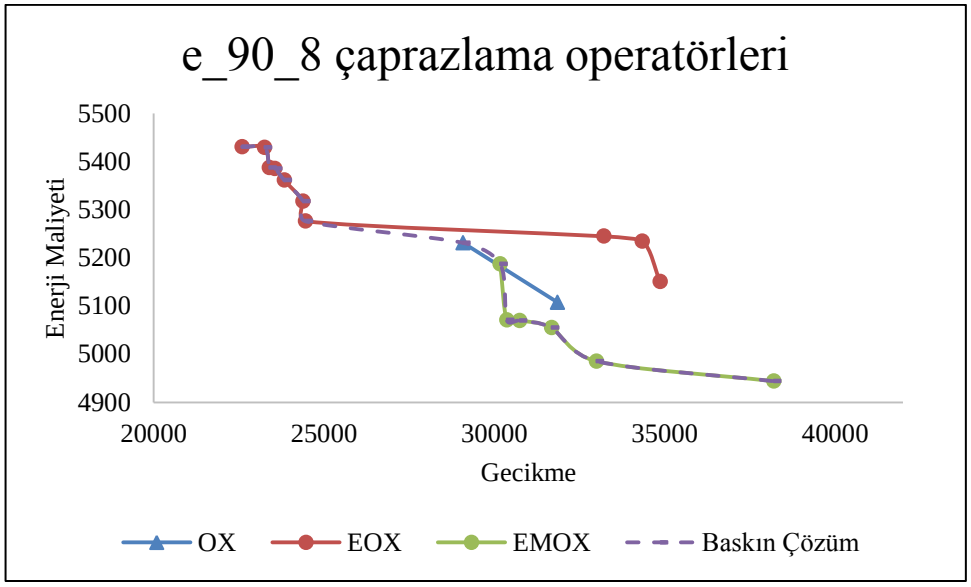
Şekil 4.70. e_90_8 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.71. e_90_8 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



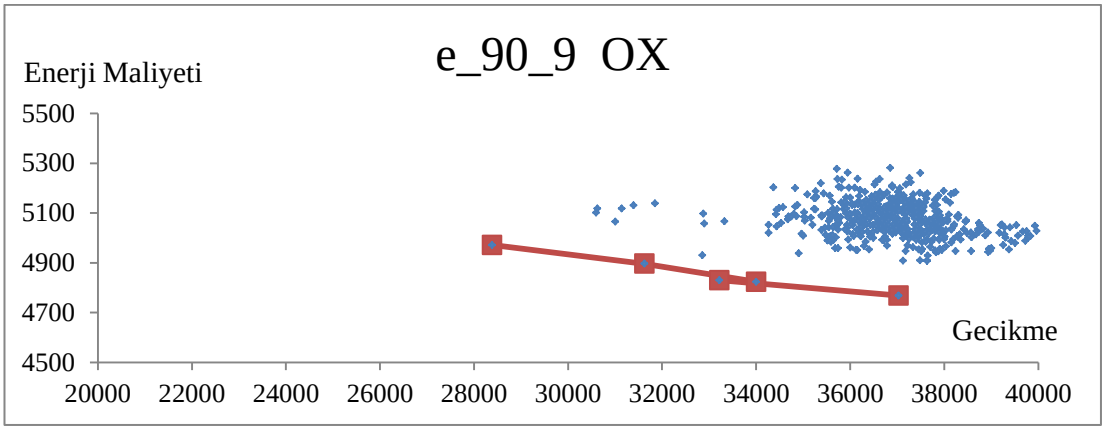
Şekil 4.72. e_90_8 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



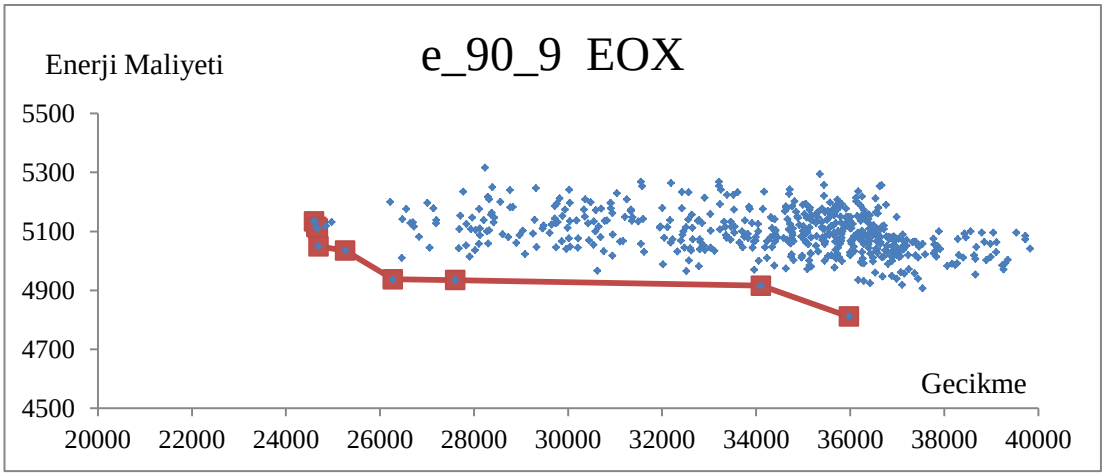
Şekil 4.73. e_90_8 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.26. e_90_9 problemi için baskın çözüm kümesi

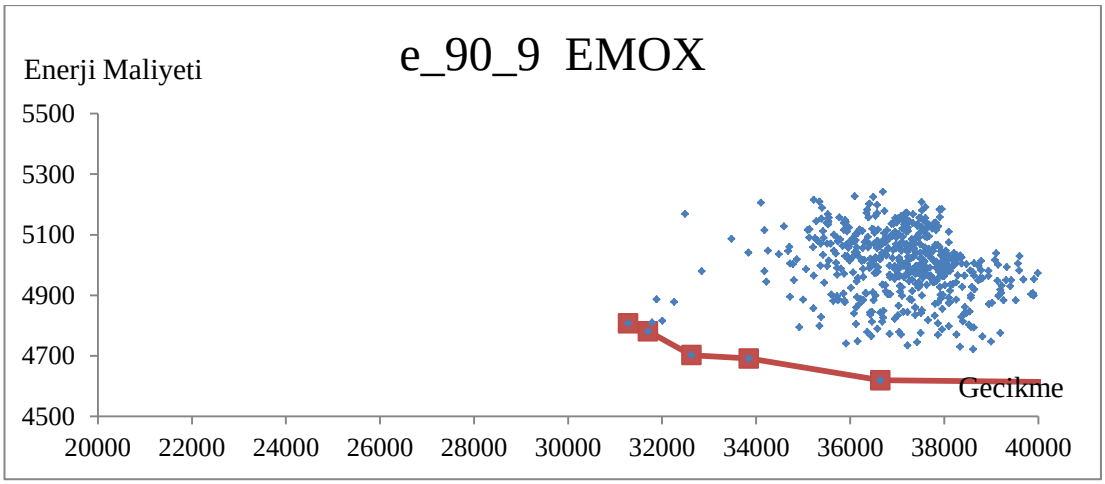
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	37032	4768,04	0	
	33221	4829,84	10	
	34005	4823,08	20	
	31628	4895,68	40	
	28388	4970,92	50	
EOX	24601	5133,36	100	-
	24650	5115	40	-
	24682	5109,68	60	-
	24695	5049,16	70	-
	25261	5034,52	30	-
	26278	4937,32	20	-
	27600	4934,96	10	-
	34104	4915,84	30	
	35976	4811,04	0	
EMOX	31276	4807,72	70	-
	31703	4781,28	30	-
	32624	4702,56	40	-
	33846	4691,6	20	-
	36643	4619,72	10	-
	41048	4612,08	0	-



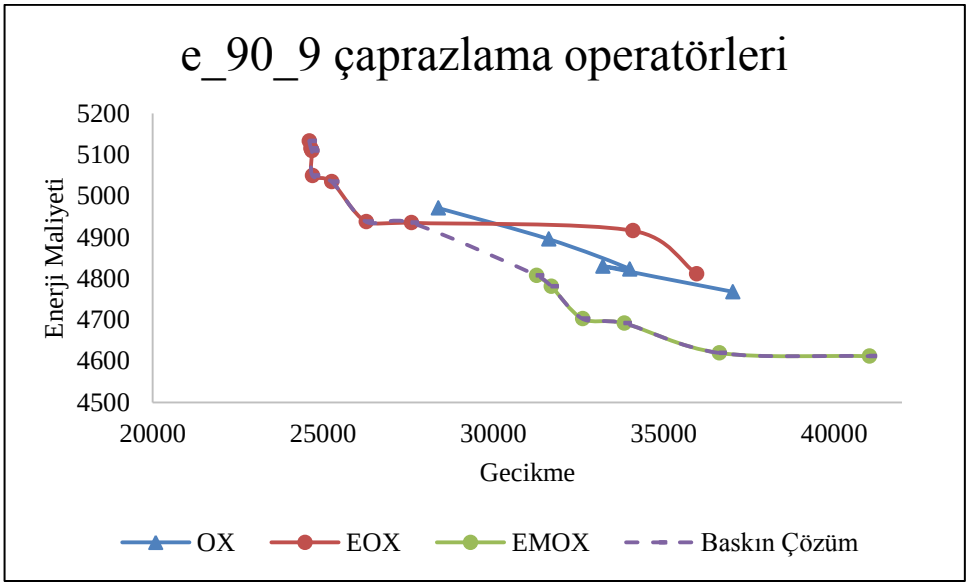
Şekil 4.74. e_90_9 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.75. e_90_9 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



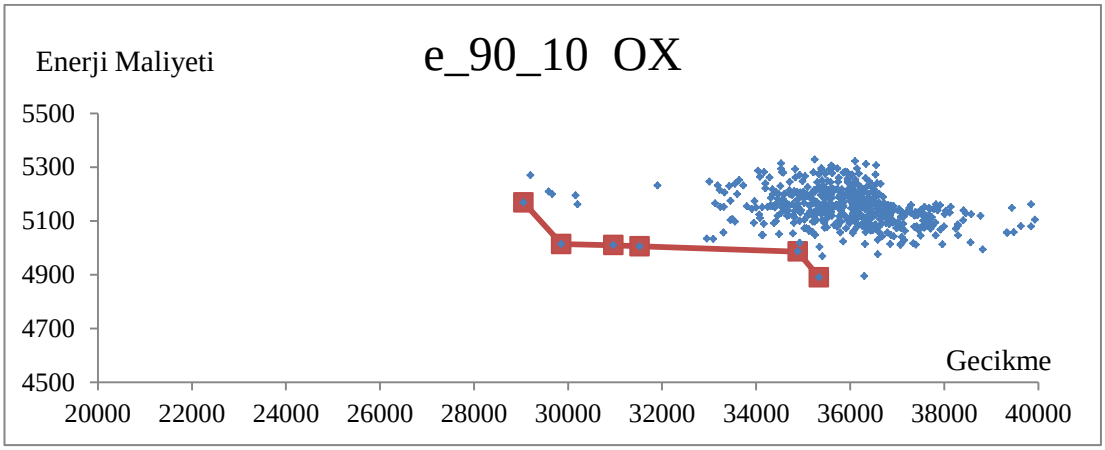
Şekil 4.76. e_90_9 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



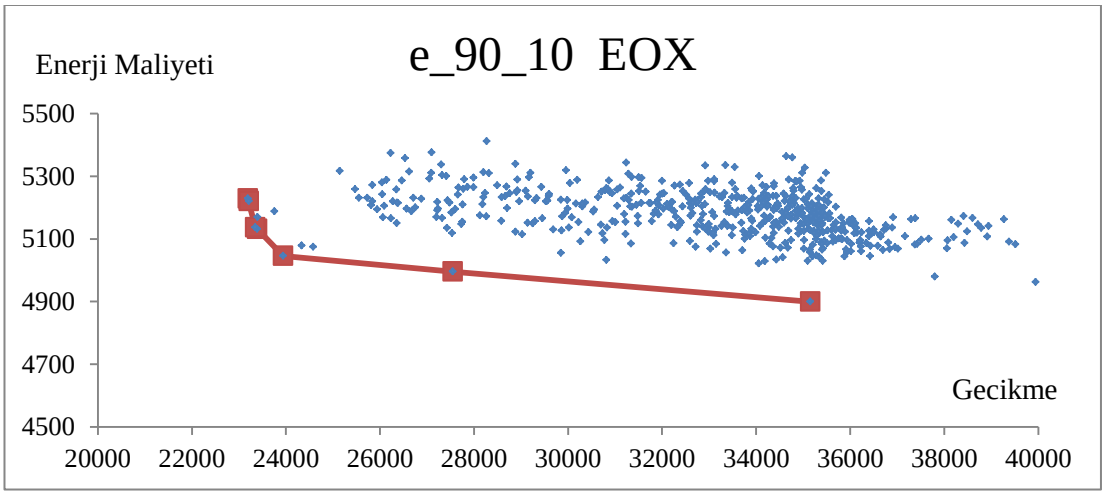
Şekil 4.77. e_90_9 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.27. e_90_10 problemi için baskın çözüm kümesi

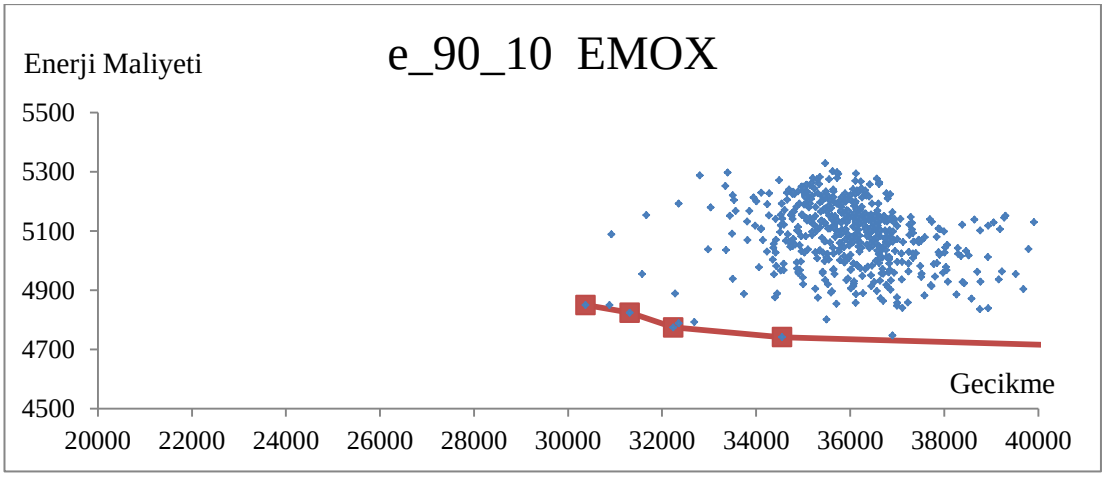
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	29056	5169,2	70	
	29857	5014,28	40	
	30967	5010,16	20	
	31523	5006,08	30	
	34886	4986,48	60	
	35339	4890,88	10	
EOX	35155	4899,68	0	
	27548	4995,48	10	-
	23941	5045,72	40	-
	23388	5131,8	60	-
	23351	5137,36	70	-
	23191	5228,56	80	-
	23213	5220,44	90	-
EMOX	30372	4850,04	50	-
	31311	4824,16	30	-
	32240	4774,56	70	-
	34553	4741,28	10	-
	40928	4711,68	0	-



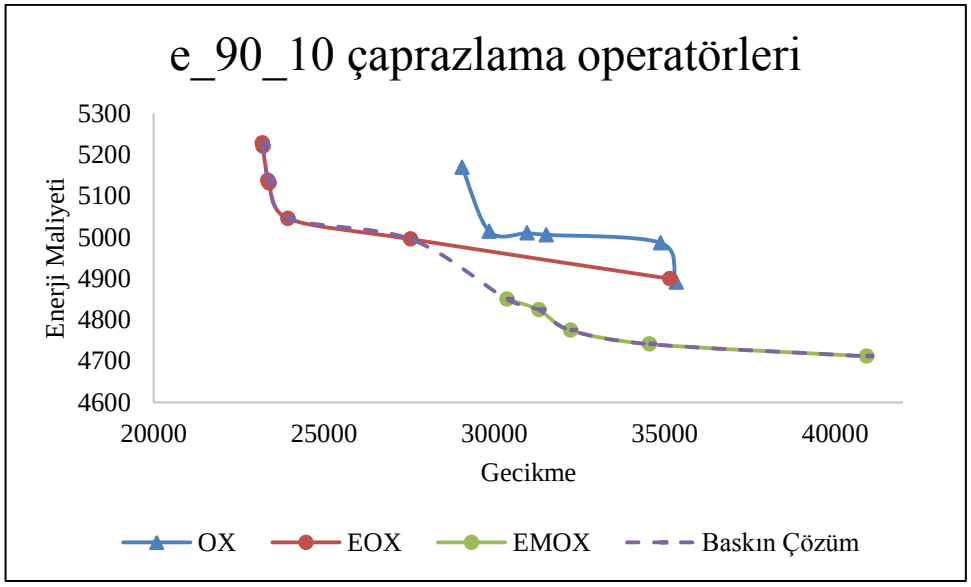
Şekil 4.78. e_90_10 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.79. e_90_10 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.80. e_90_10 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.81. e_90_10 problemi için çaprazlama operatörlerinin kıyaslaması

90 işlik problemlerin çözümleri ile ilgili genel durum Tablo 4.28'de ve 4.29'da en iyi ve en kötü sonucu veren yöntemler işaretlenerek gösterilmiştir. Diferansiyel Gelişim Algoritması (DGA) kullanılarak elde edilen çözümler için iki amaç fonksiyonu beraber tek bir amaç fonksiyonu yazılarak çözülmüştür. Bu problemler termin tarihlerinin orijinal değerlerine göre ele alındığı zorluk derecesi düşük problemlerdir. Dolayısıyla, enerji maliyeti anlamında genetik algoritma ile kıyaslanması doğru olmayabilir. Ancak, TMEMM yöntemi termin tarihlerini dikkate almadığı için kıyaslama yapıldığında 10 problemin 4'ünde DGA'dan daha iyi sonuç verdiği görülebilir. Genetik algoritma için kullanılan EOX çaprazlama operatörü toplam gecikme zamanı hedefiyle ilgili amaç fonksiyonları için uygun olduğundan tercih edilmiştir. Dolayısıyla TMTGZM yönteminden daha iyi sonuç vermesi beklenebilir. Ancak, sadece 3 problemde TMTGZM yönteminden daha iyi sonuç verdiği gözlemlenmiştir. OX, EOX ve EMOX çaprazlama yöntemleri kendi içinde kıyaslandığında 60 işlik problemlerde olduğu gibi EMOX'un enerji maliyeti anlamında diğerlerinden daha iyi sonuç verdiği, EOX'un ise diğer amaç olan toplam gecikme zamanını iyileştirmek konusunda daha başarılı olduğu söylenebilir. Karar vericiye, alternatifleri değerlendirirken toplam gecikme zamanının daha kritik olduğu zamanlarda EOX çaprazlama operatörünü kullanması, maliyetlerin ön plana çıktığı durumlarda ise EMOX çaprazlama operatörünü kullanması önerilebilir. OX çaprazlama operatörü ise herhangi bir amaç için şekillendirilmediğinden ortada çözümler çıkarmıştır. Her bir çaprazlama operatörünün baskın çözümleri toplamda

değerlendirildiğinde EOX'un ve EMOX'un baskın çözümlerinin OX'in ürettiği baskın çözümlerden daha da baskın olduğu gözlemlenmiştir. Tüm çözümler içindeki baskın çözüm kümelerinde OX operatörü 7 baskın çözüm üretebilirken, EOX 51, EMOX ise 56 baskın çözüm üretmiştir.

Tablo 4.28. 90 işlik problemlerin en iyi enerji maliyet değerleri

Problem Adı	TMEMM	DGA	ga-ox	ga-eox	ga-emox
e_90_1	4734,12	4556	4800,72	4787,36	4691,16
e_90_2	4706,96	4696	4897,52	4962,32	4821,56
e_90_3	4801,36	4533	4806,52	4838,44	4631,12
e_90_4	4768,8	4550	4810	4820,12	4644,6
e_90_5	4693,92	4699	4927,2	4997,94	4839,12
e_90_6	4716,64	4737	5019,36	5056,84	4898,6
e_90_7	4569,12	4625	4833,8	4930,6	4743,12
e_90_8	4789,28	4850	5107,76	5150,64	4944,04
e_90_9	4853,36	4509	4768,04	4811,04	4612,08
e_90_10	4732,88	4612	4986,48	4899,68	4711,68

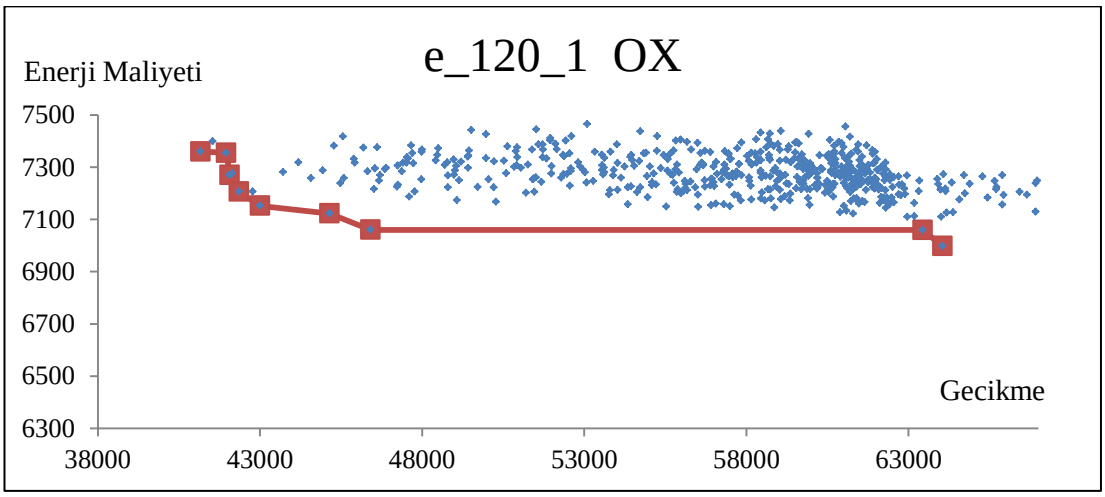
Tablo 4.29. 90 işlik problemlerin en iyi toplam gecikme zamanı değerleri

Problem Adı	TMTGZM	ga-ox	ga-eox	ga-emox
e_90_1	21638	28183	26729	29310
e_90_2	20354	25838	20656	27461
e_90_3	24362	28288	24789	29936
e_90_4	19783	26637	21331	28103
e_90_5	23195	27839	23094	30675
e_90_6	22819	28887	28271	30279
e_90_7	19366	26591	20826	25914
e_90_8	22775	29089	22601	30177
e_90_9	23468	28388	24601	31276
e_90_10	23194	29056	23191	30372

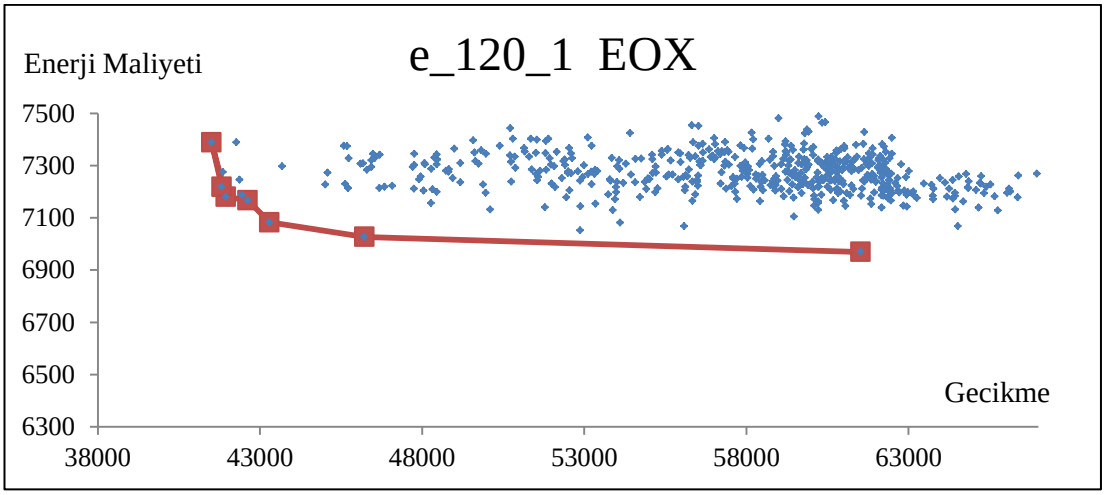
Alttaki grafiklerde ve tablolarda ise 120 işlik problemlerin baskın çözüm kümeleri ve her α değeri için en iyi 50, toplamda ise tüm α değerleri için en iyi 550 çözümün dağılımı gösterilmiştir. Tablolarda tüm çaprazlama yöntemleri beraber düşünüldüğünde baskın olan çözümler işaretlenmiştir.

Tablo 4.30. e_120_1 problemi için baskın çözüm kümesi

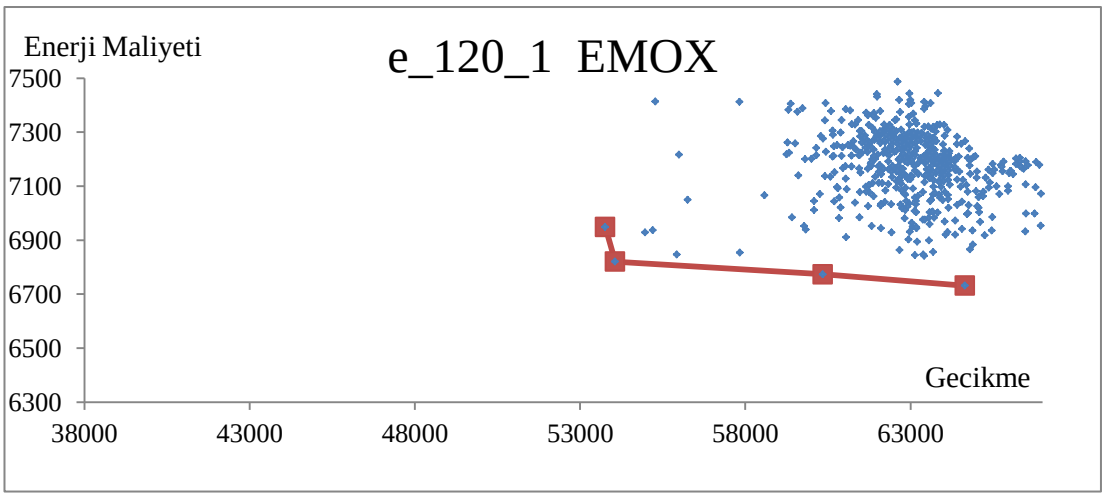
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	41163	7360,36	50	-
	41957	7355,04	70	
	42067	7270,32	100	
	42355	7206,52	80	
	43006	7152,24	30	-
	45147	7122,72	20	
	46409	7060	10	
	63431	7059,8	40	
	64046	6998,48	0	
EOX	61520	6969,56	0	
	46223	7027,24	10	-
	43290	7082,72	20	
	42620	7167,32	40	-
	41948	7180,64	70	-
	41502	7389,52	80	
	41820	7218,88	90	-
EMOX	64642	6731,28	0	-
	60343	6773,4	10	-
	54061	6820,12	40	-
	53760	6948,8	70	-



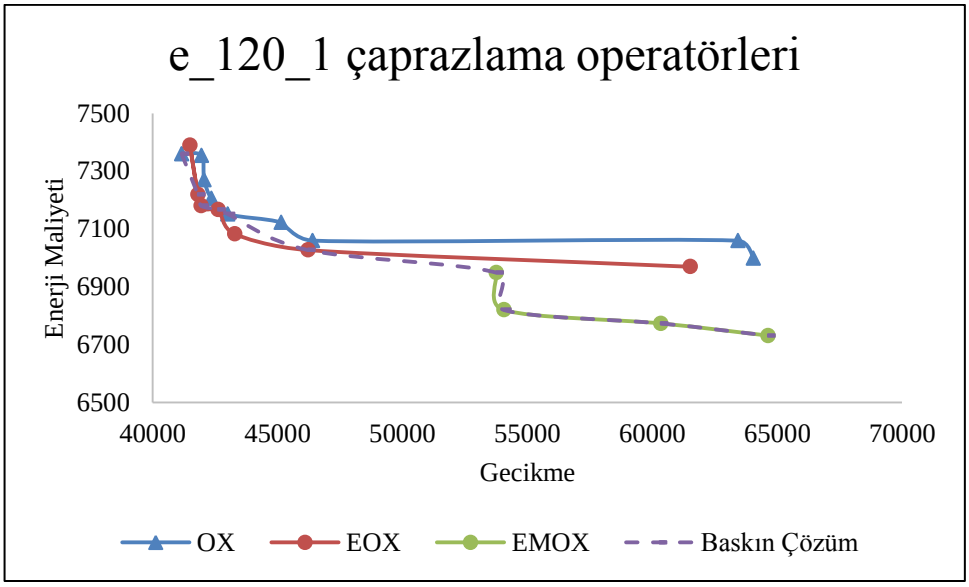
Şekil 4.82. e_120_1 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.83. e_120_1 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



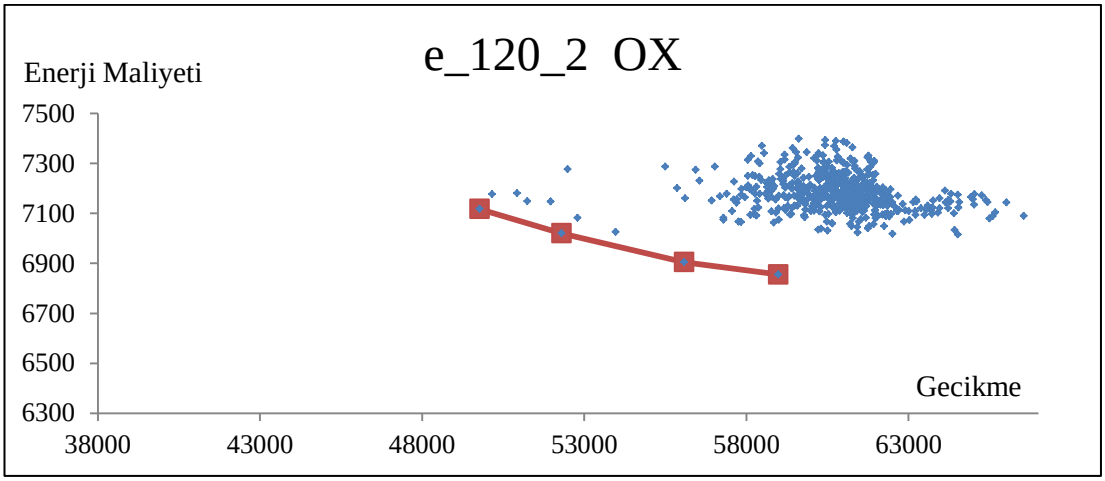
Şekil 4.84. e_120_1 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



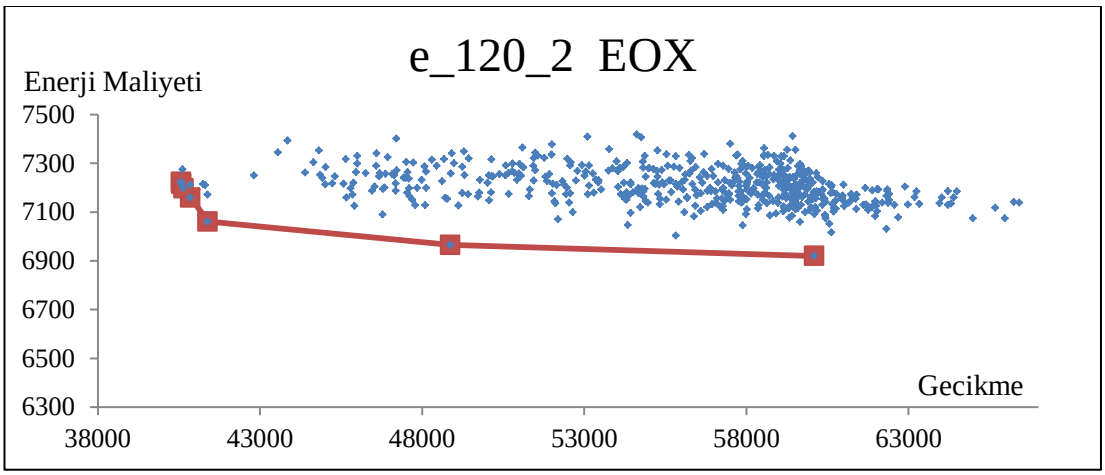
Şekil 4.85. e_120_1 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.31. e_120_2 problemi için baskın çözüm kümesi

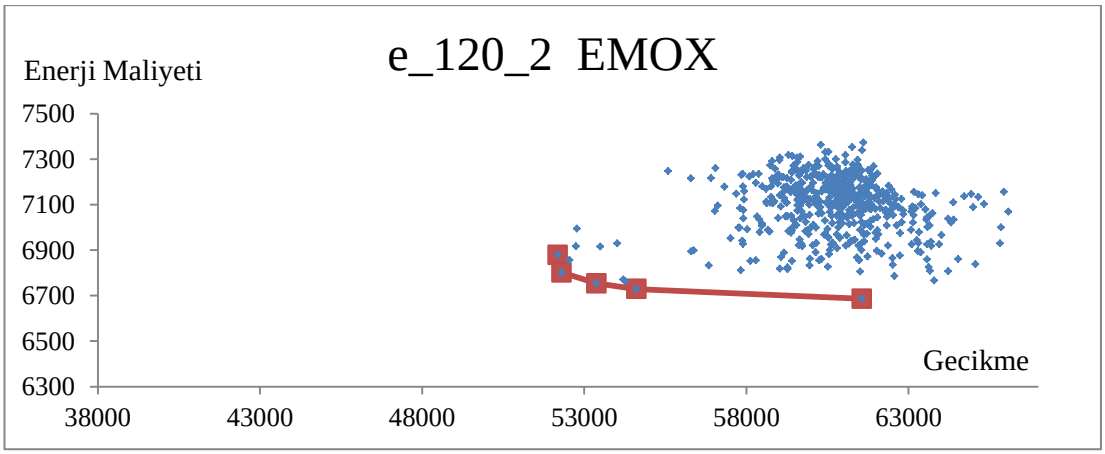
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	58982	6855,2	0	
	56083	6904,72	10	
	52303	7019,96	30	
	49773	7117,16	100	
EOX	60091	6919,92	0	
	48861	6965	10	-
	41385	7061,48	20	-
	40576	7215,36	60	-
	40853	7158,76	70	-
	40565	7223,32	80	-
	40647	7197,88	100	-
EMOX	52181	6878,76	50	-
	52302	6801,72	90	-
	53377	6753,96	30	-
	54612	6728,88	20	-
	61561	6686,2	0	-



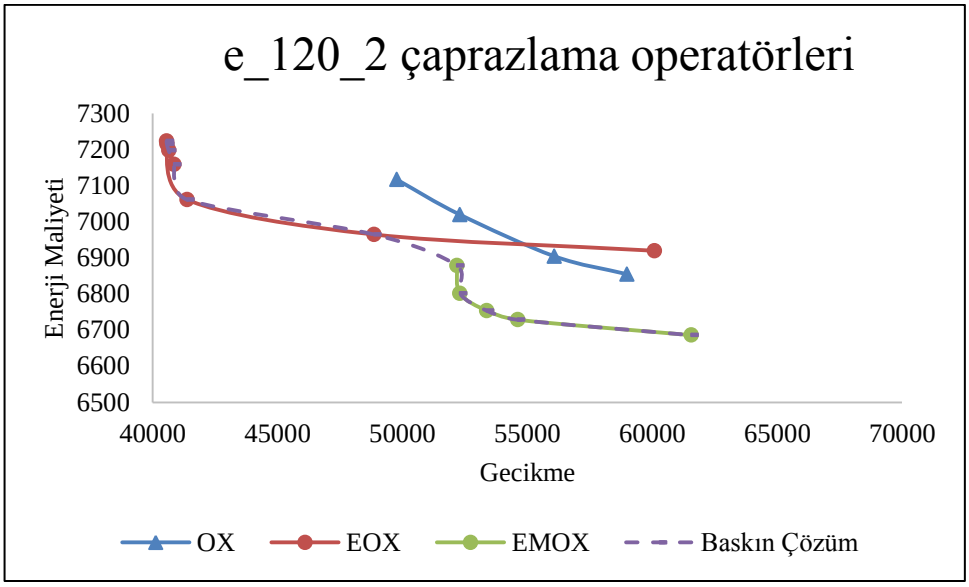
Şekil 4.86. e_120_2 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.87. e_120_2 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



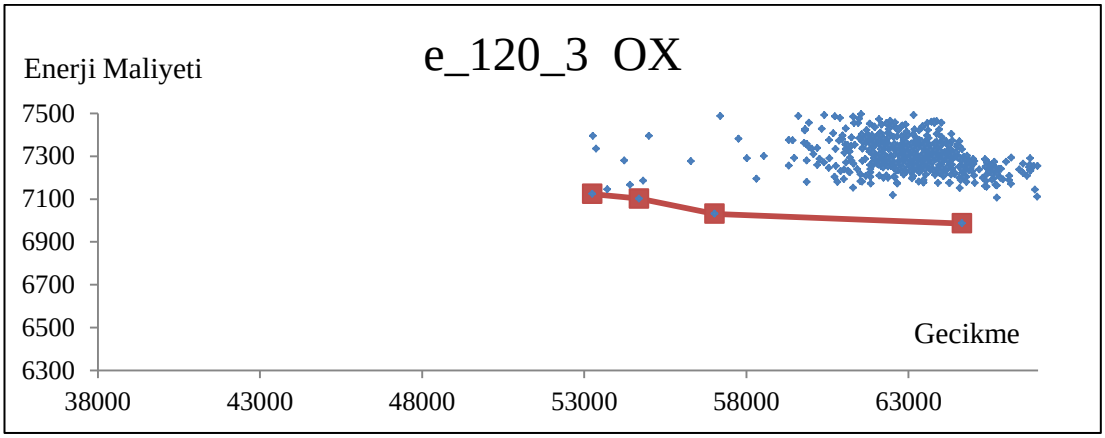
Şekil 4.88. e_120_2 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



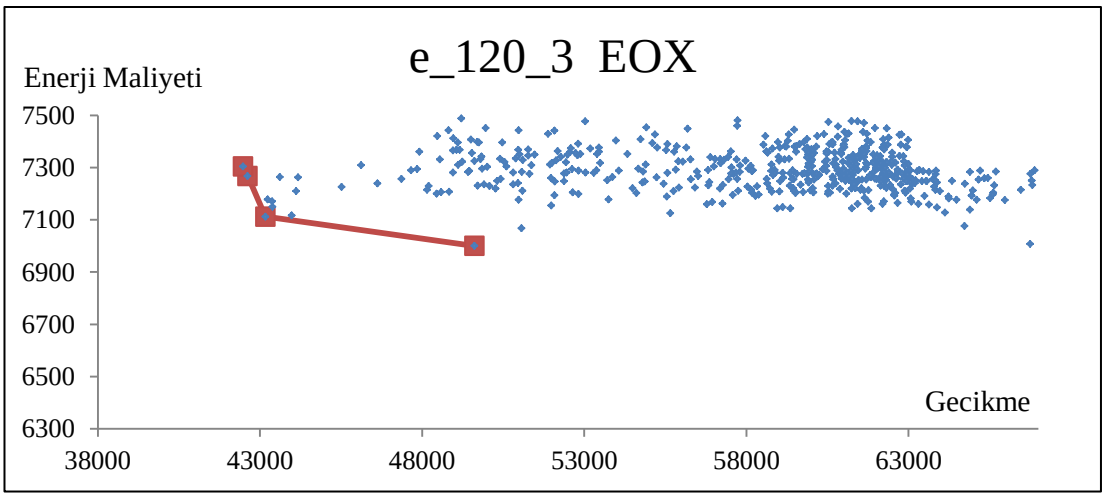
Şekil 4.89. e_120_2 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.32. e_120_3 problemi için baskın çözüm kümesi

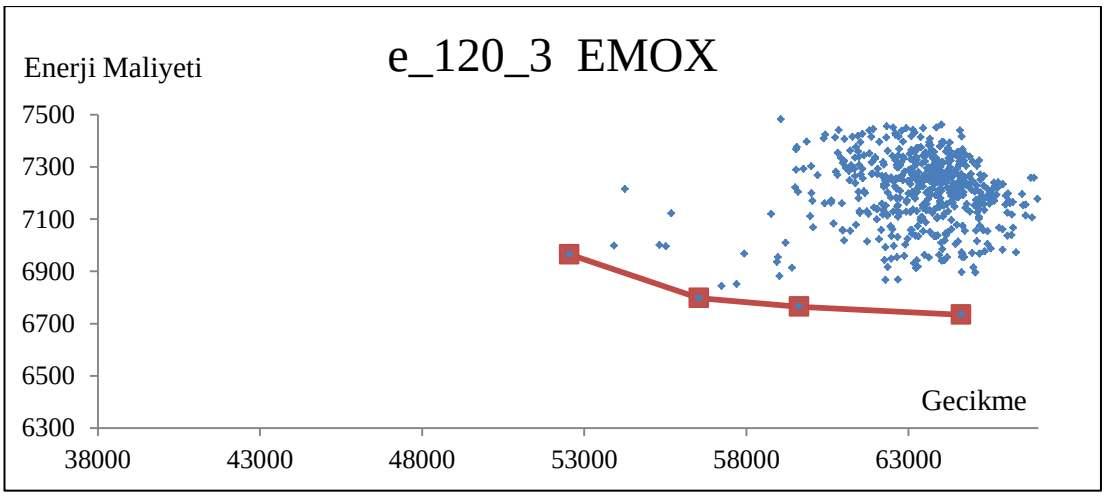
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	64653	6986,44	0	
	57023	7031	10	
	54696	7102,04	30	
	53249	7124,08	50	
EOX	49618	7000	10	-
	43169	7112,16	70	-
	42620	7268,36	80	-
	42476	7304,08	100	-
EMOX	64617	6734,24	0	-
	59623	6764,96	10	-
	56540	6797,68	30	-
	52534	6965,04	80	-



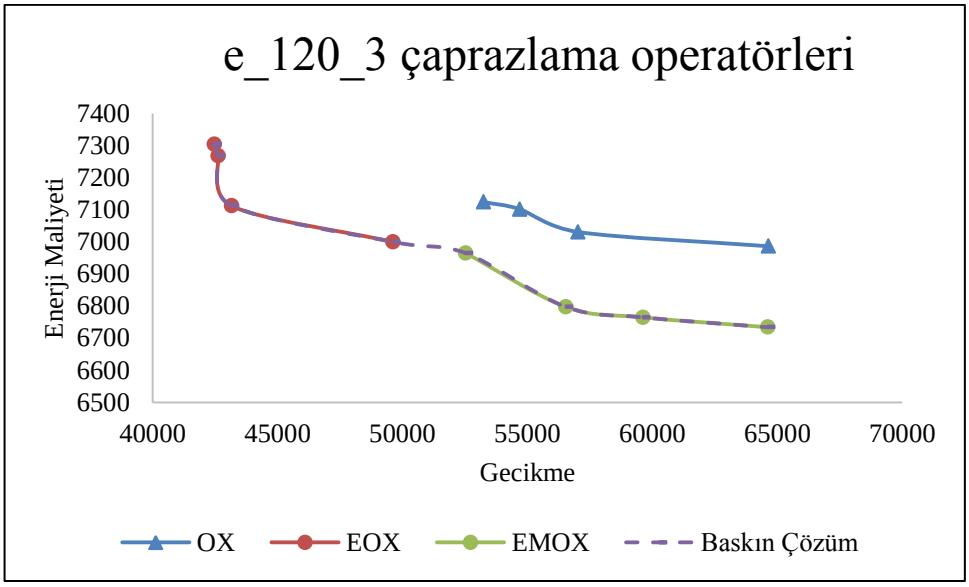
Şekil 4.90. e_120_3 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.91. e_120_3 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



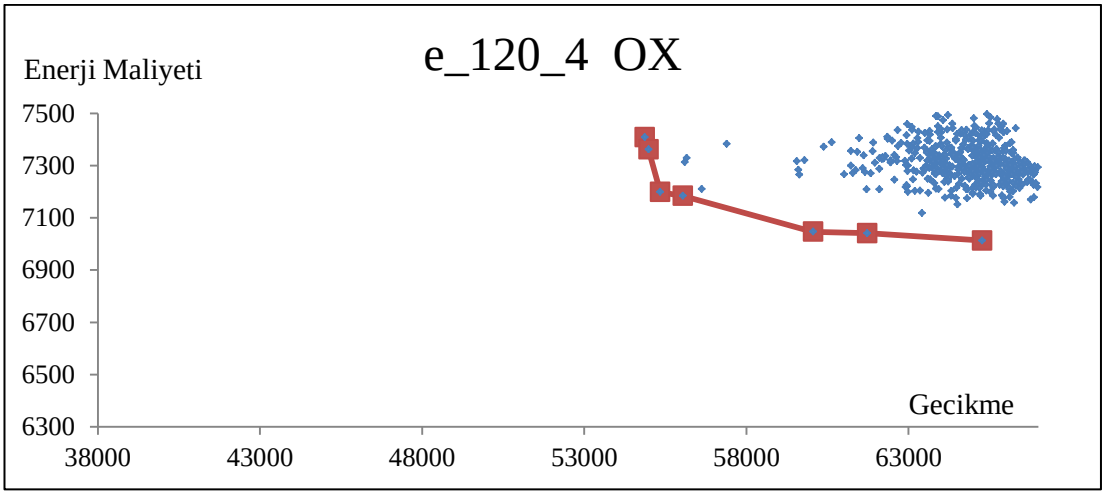
Şekil 4.92. e_120_3 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



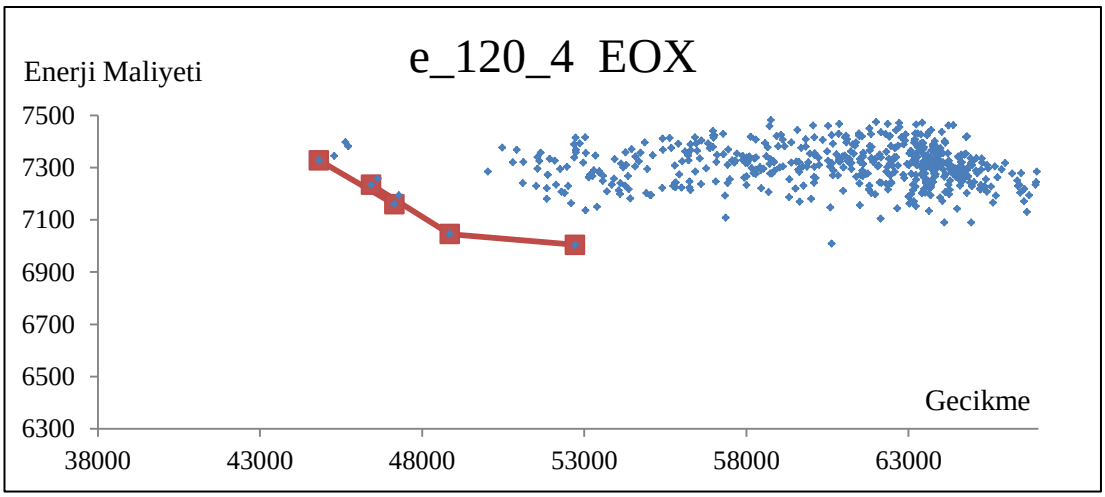
Şekil 4.93. e_120_3 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.33. e_120_4 problemi için baskın çözüm kümesi

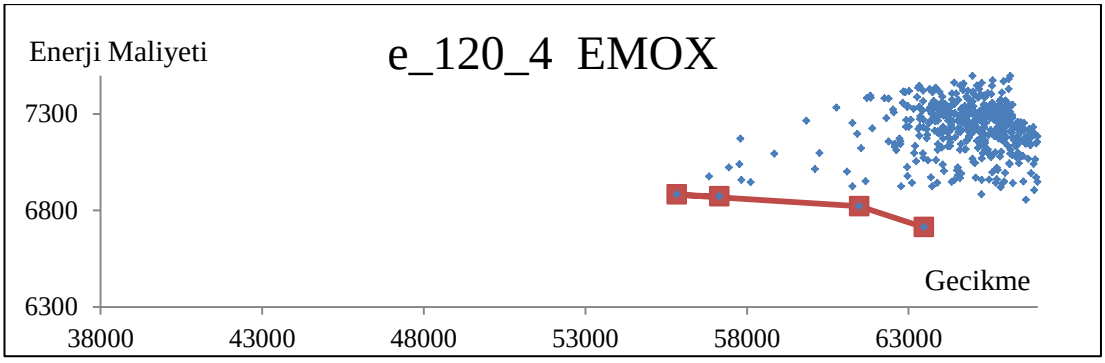
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	54866	7409,16	100	
	54984	7362,16	90	
	55343	7198,92	20	
	56044	7184,28	40	
	60057	7047	30	
	61730	7041,28	10	
	65276	7013,2	0	
EOX	52720	7004,6	10	-
	48860	7045,4	20	-
	46430	7234,28	30	-
	47152	7159,44	40	-
	44816	7328,16	100	-
EMOX	63473	6713,44	10	-
	61476	6821,56	20	-
	55830	6882,16	50	-
	57147	6872,4	80	-



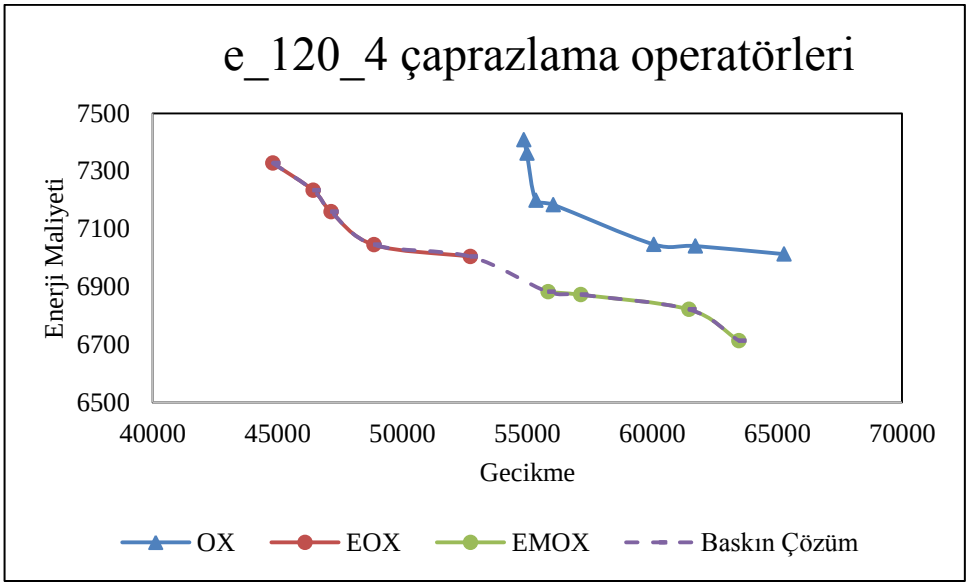
Şekil 4.94. e_120_4 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.95. e_120_4 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



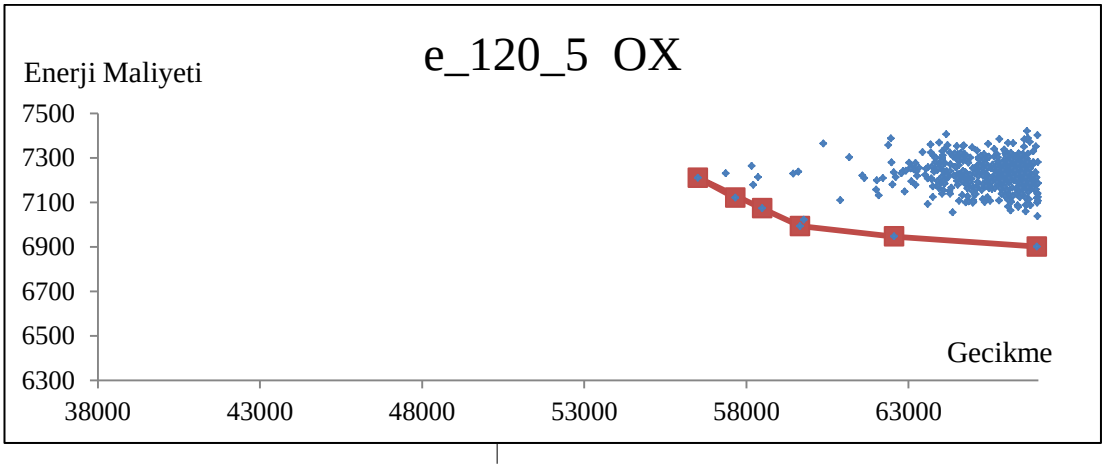
Şekil 4.96. e_120_4 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



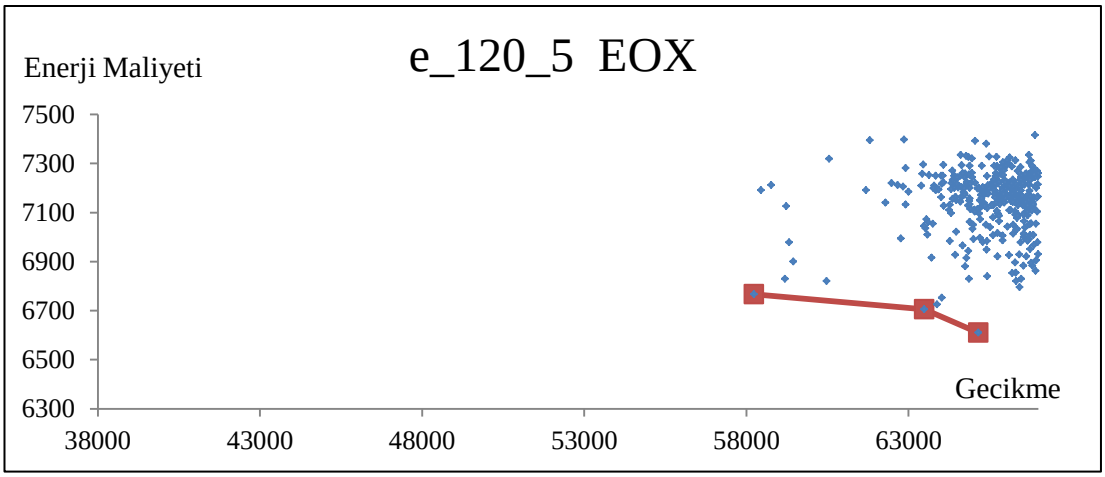
Şekil 4.97. e_120_4 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.34. e_120_5 problemi için baskın çözüm kümesi

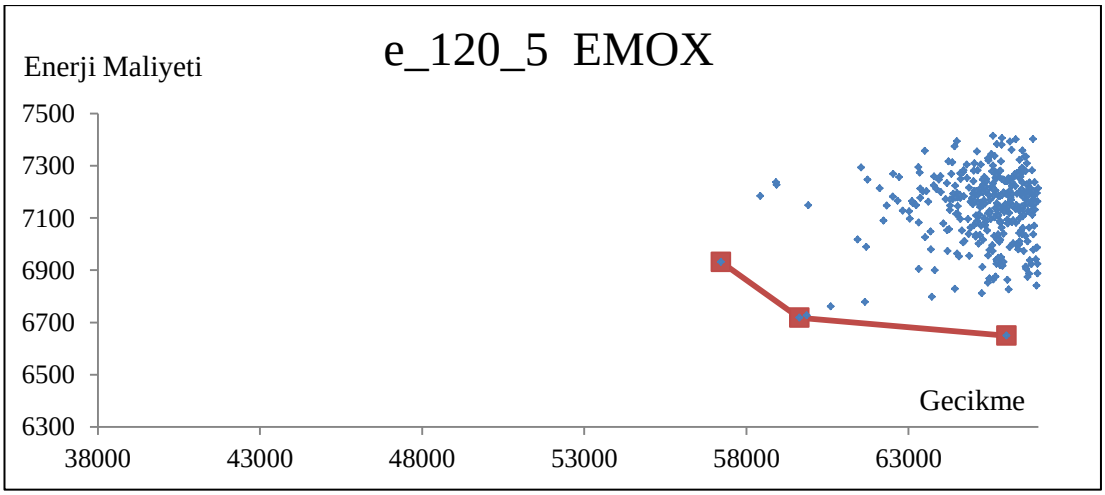
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	66961	6900,76	0	
	62561	6946,44	10	
	59657	6992,88	20	
	58490	7073,28	30	
	56510	7210,32	50	-
	57661	7120,32	60	
EOX	65150	6610,72	10	-
	63489	6705,72	20	-
	58237	6767,36	30	-
EMOX	66020	6649,24	10	
	59634	6718,4	50	-
	57218	6931,8	60	-



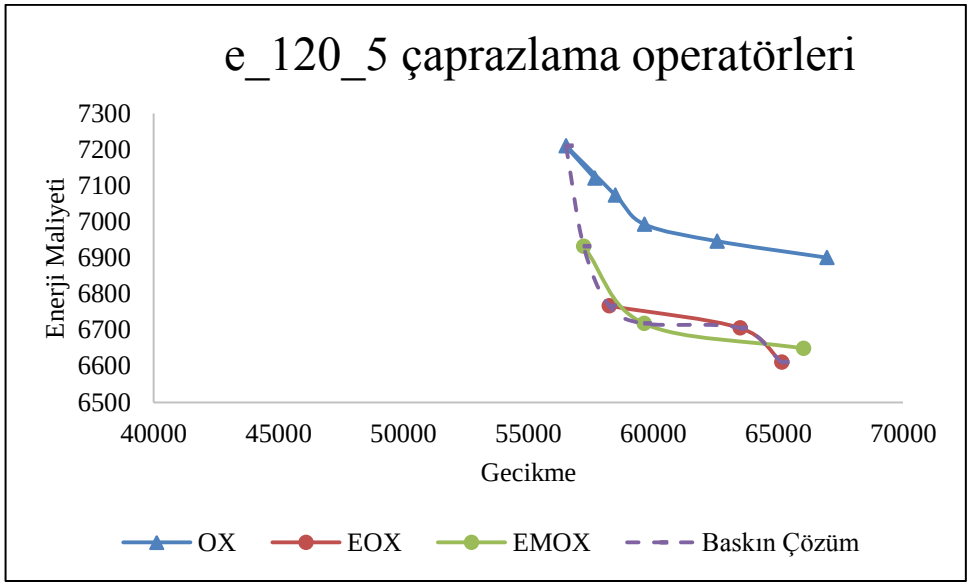
Şekil 4.98. e_120_5 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.99. e_120_5 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



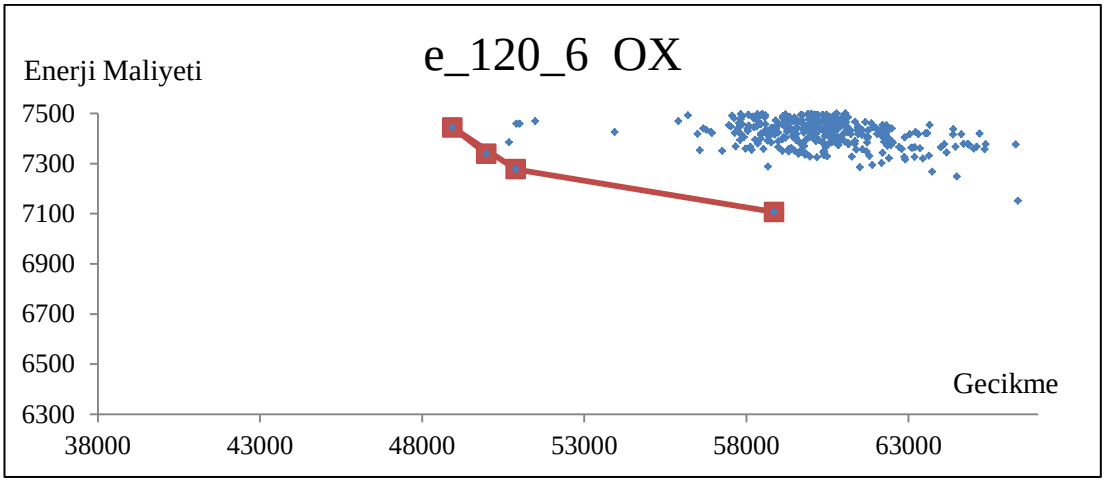
Şekil 4.100. e_120_5 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



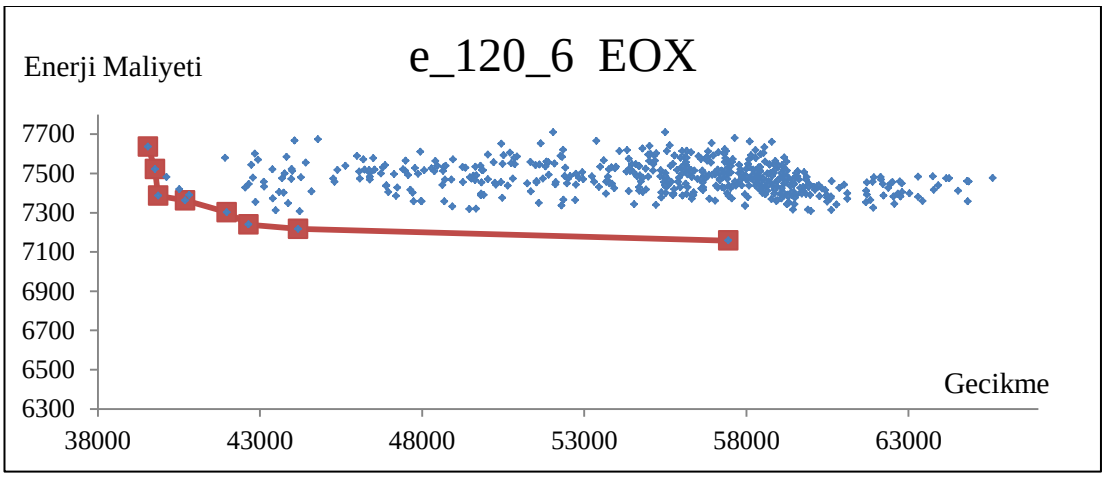
Şekil 4.101. e_120_5 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.35. e_120_6 problemi için baskın çözüm kümesi

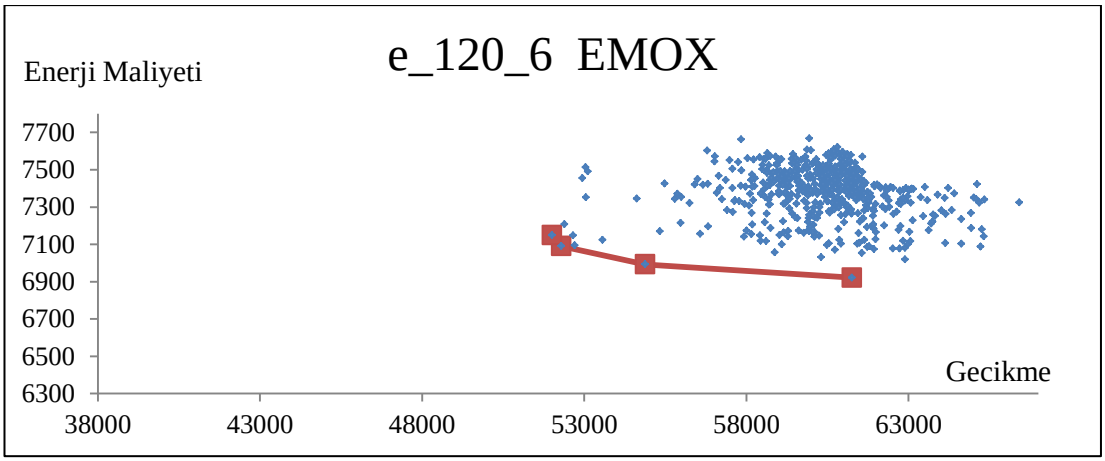
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	58861	7106,96	10	
	50896	7277,2	20	
	48939	7443,44	40	
	49984	7338,88	50	
EOX	39549	7636,16	100	-
	39761	7522,4	90	-
	39862	7386,6	70	-
	40694	7362,88	40	-
	41977	7302,12	30	-
	42648	7239,24	20	-
	44179	7217,84	10	-
	57442	7158	0	
EMOX	61251	6920,92	0	-
	54882	6992,08	10	-
	52295	7090,08	20	-
	52007	7148,64	50	-



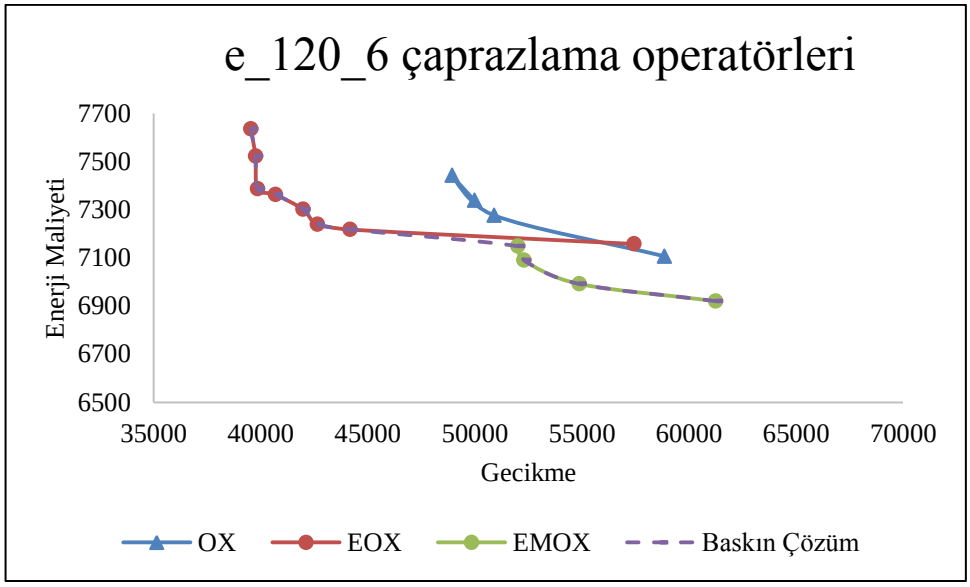
Şekil 4.102. e_120_6 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.103. e_120_6 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



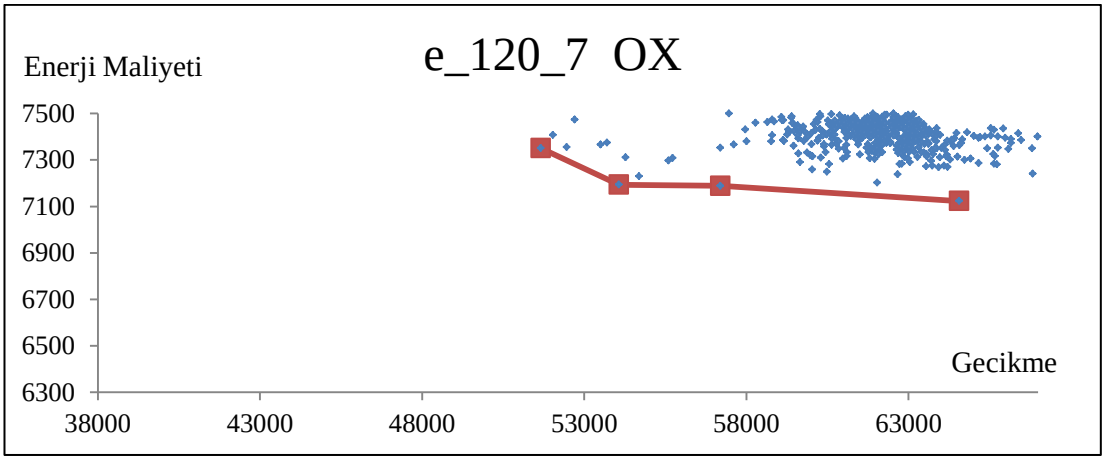
Şekil 4.104. e_120_6 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



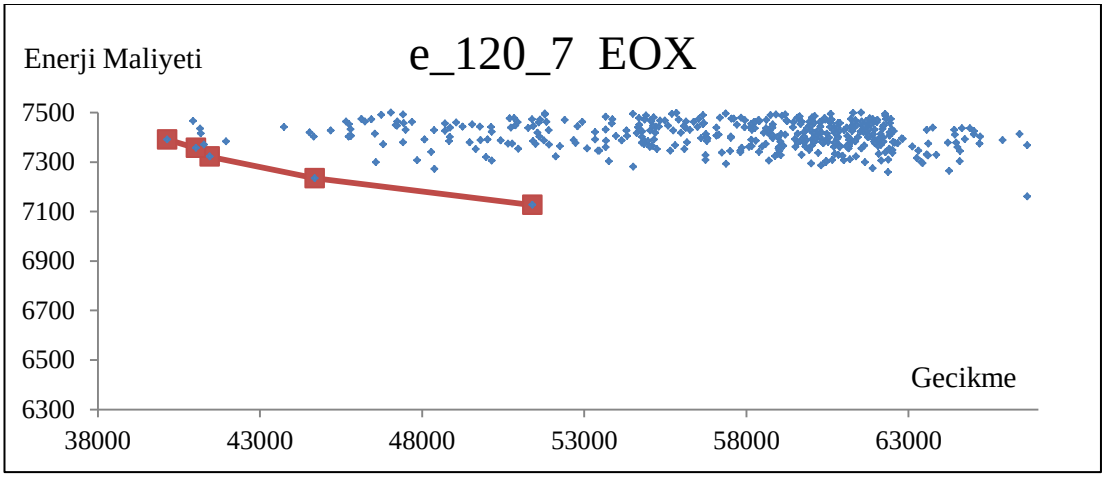
Şekil 4.105. e_120_6 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.36. e_120_7 problemi için baskın çözüm kümesi

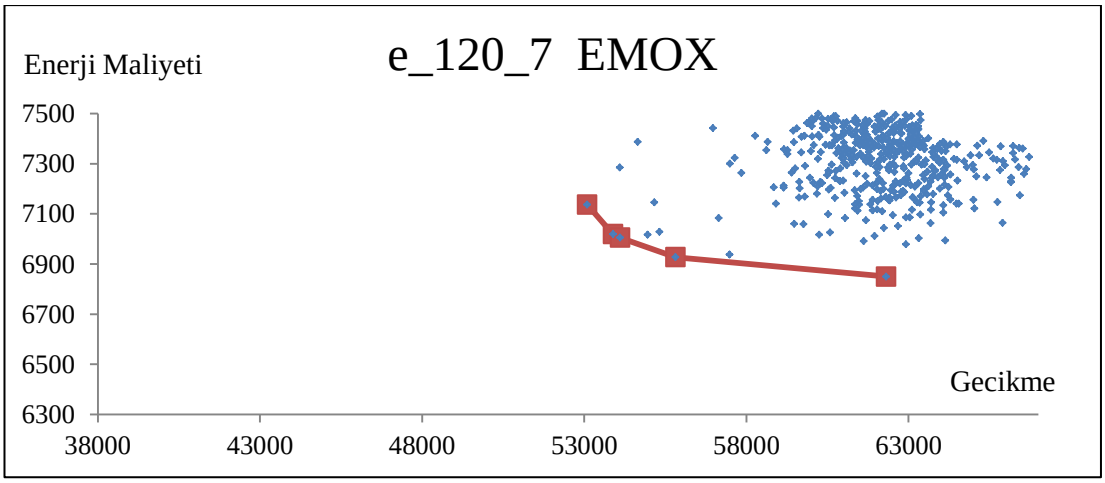
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	64562	7123,28	0	
	57198	7188,64	10	
	54072	7193,12	40	
	51666	7350,28	100	
EOX	51399	7126,64	10	-
	44687	7234,76	20	-
	41452	7322,16	30	-
	41028	7357	40	-
	40135	7391,52	100	-
EMOX	53087	7137,52	100	
	53886	7019,48	60	-
	54106	7005,44	50	-
	55820	6927,48	30	-
	62313	6850,16	0	-



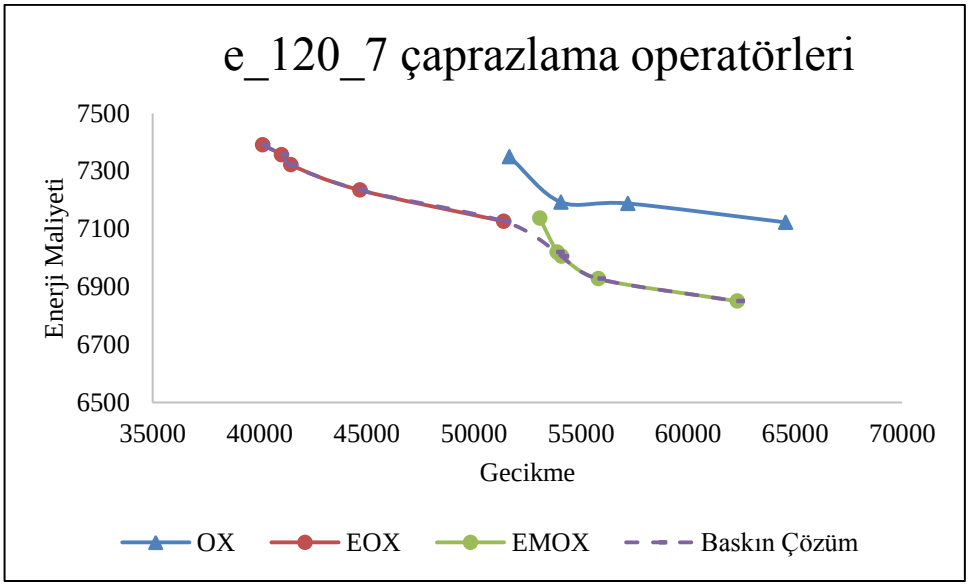
Şekil 4.106. e_120_7 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.107. e_120_7 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



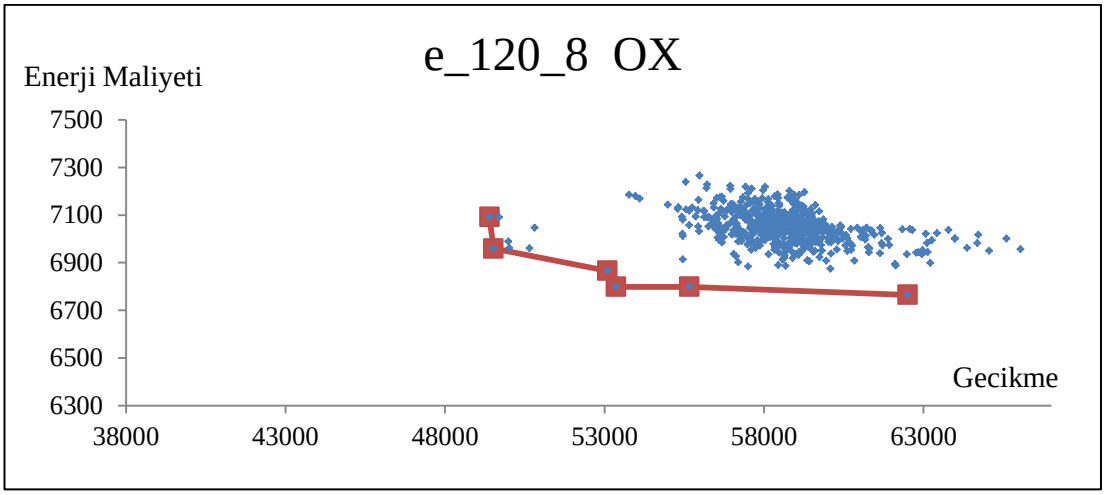
Şekil 4.108. e_120_7 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



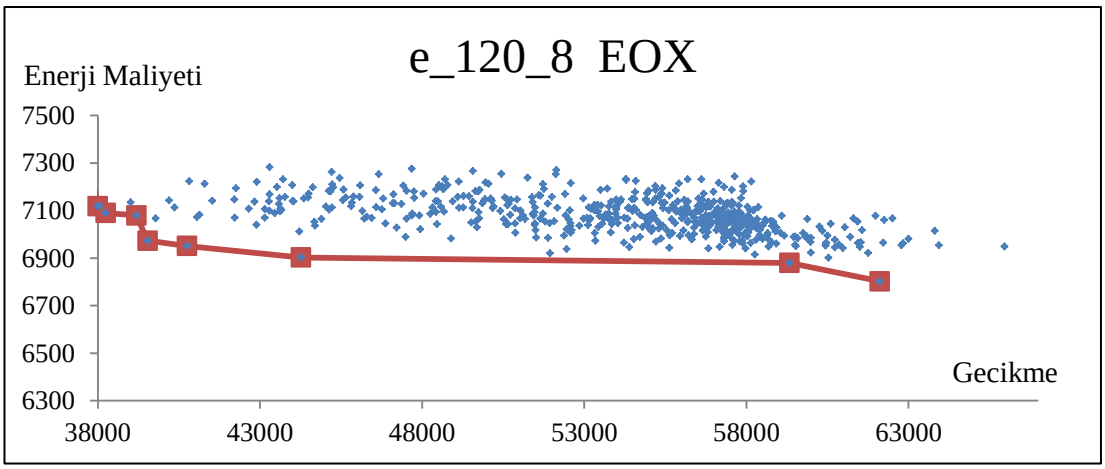
Şekil 4.109. e_120_7 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.37. e_120_8 problemi için baskın çözüm kümesi

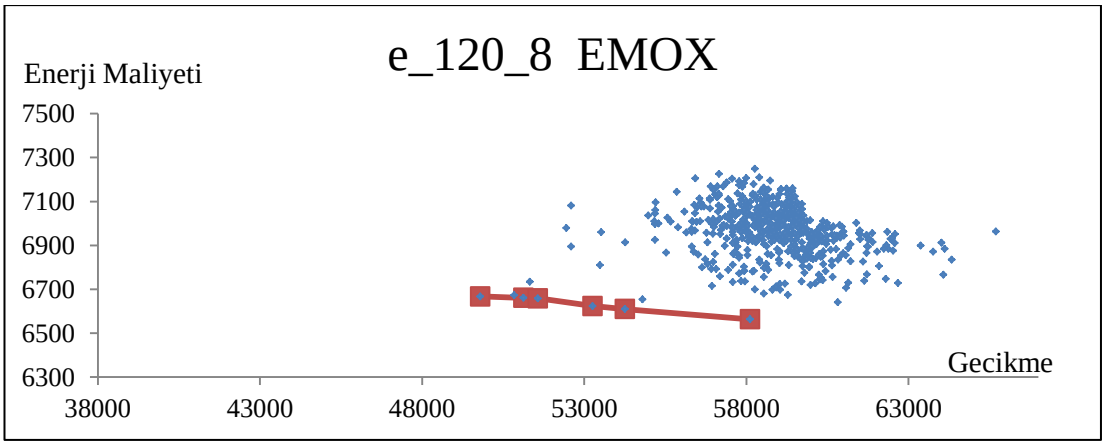
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	49395	7091,64	70	
	49519	6959,44	90	
	53091	6865,84	30	
	53361	6798,8	20	
	55662	6798,4	10	
	62499	6765,36	0	
EOX	37926	7182,32	90	-
	37950	7120,2	40	-
	38002	7118,4	50	-
	38242	7089,64	60	-
	39188	7079,96	70	-
	39537	6973,44	30	-
	40749	6951,12	20	-
	44269	6902,64	10	-
	59334	6879,92	30	
	62113	6802,08	0	
EMOX	49793	6667,52	30	-
	51126	6661,28	60	-
	51574	6658,76	100	-
	53258	6624	40	-
	54251	6610,12	10	-
	58117	6563,48	0	-



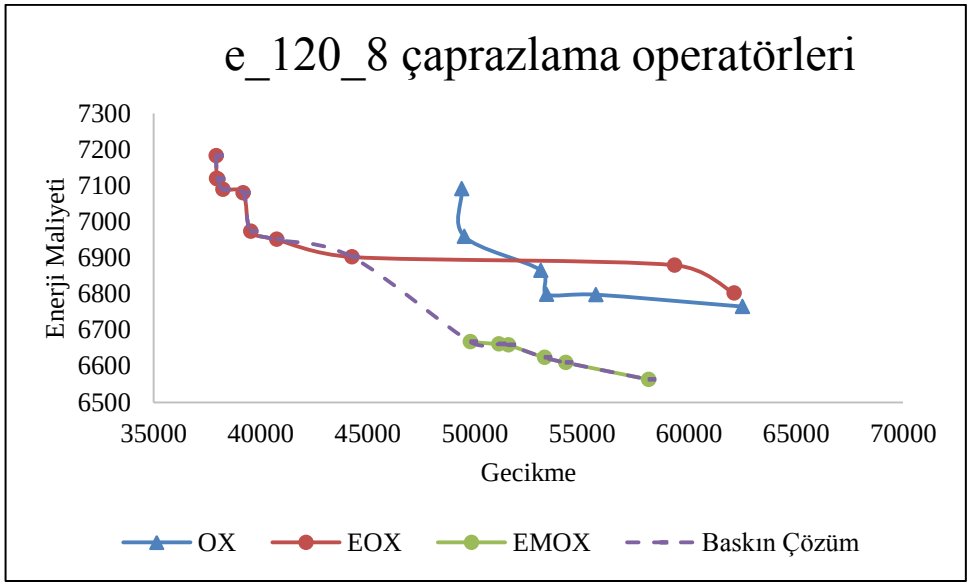
Şekil 4.110. e_120_8 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.111. e_120_8 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



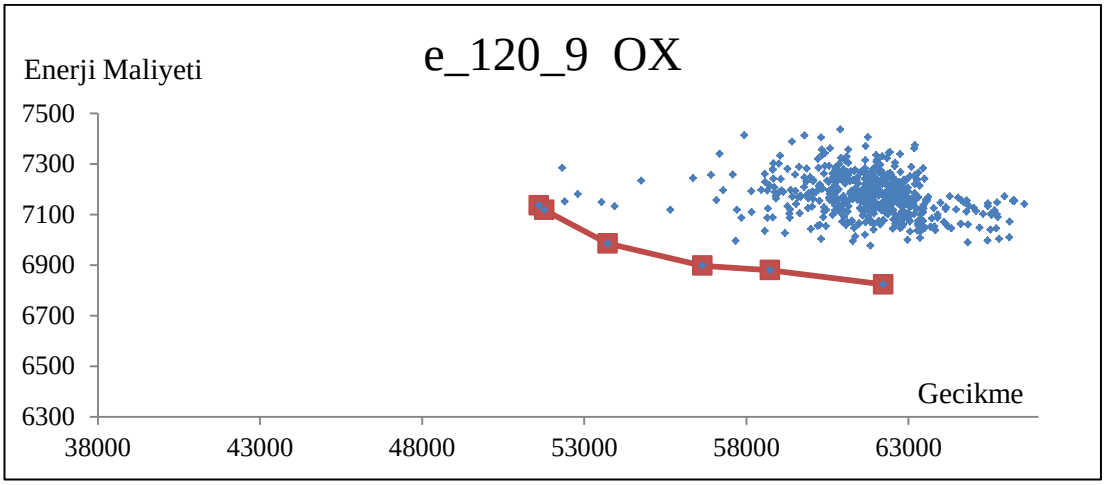
Şekil 4.112. e_120_8 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



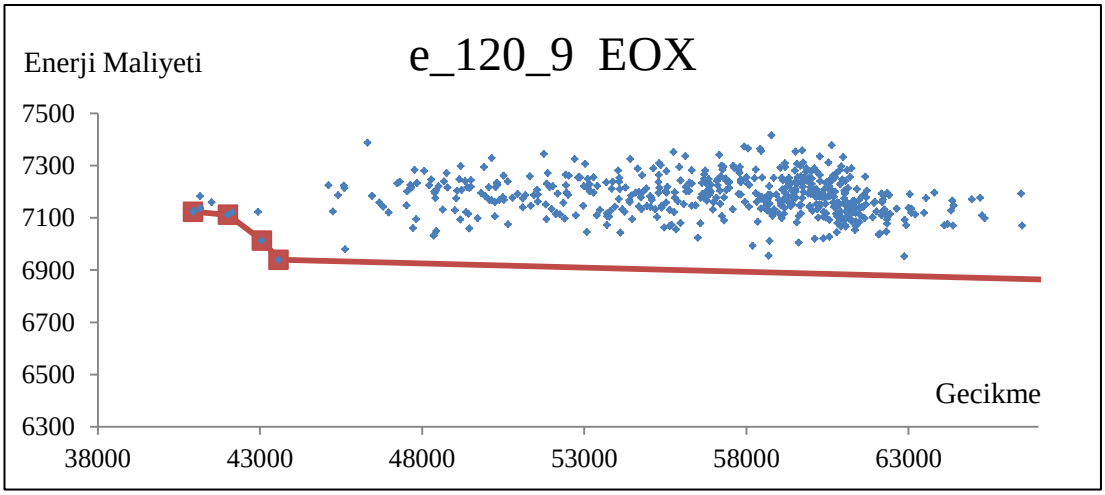
Şekil 4.113. e_120_8 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.38. e_120_9 problemi için baskın çözüm kümesi

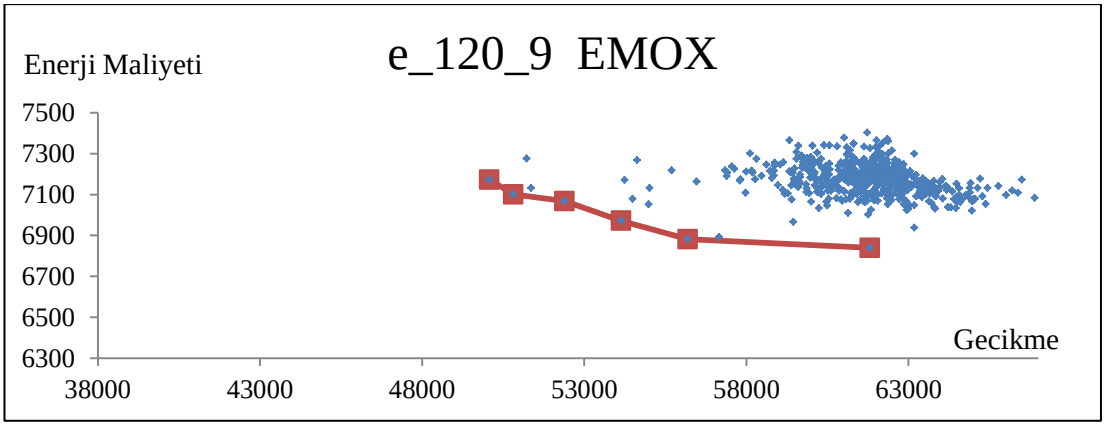
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	51597	7135,8	90	
	51769	7118,16	100	
	53718	6985,6	30	
	56649	6898,04	20	
	58730	6880,2	10	-
	62218	6823,6	0	-
EOX	67575	6862,88	0	
	43579	6939,28	20	-
	43063	7013,12	30	-
	42019	7112,08	80	-
	40941	7123,56	90	-
EMOX	61803	6838,88	0	-
	56192	6881,16	10	-
	54141	6971,68	50	
	52393	7067,96	60	
	50811	7101,04	80	
	50071	7172,6	90	



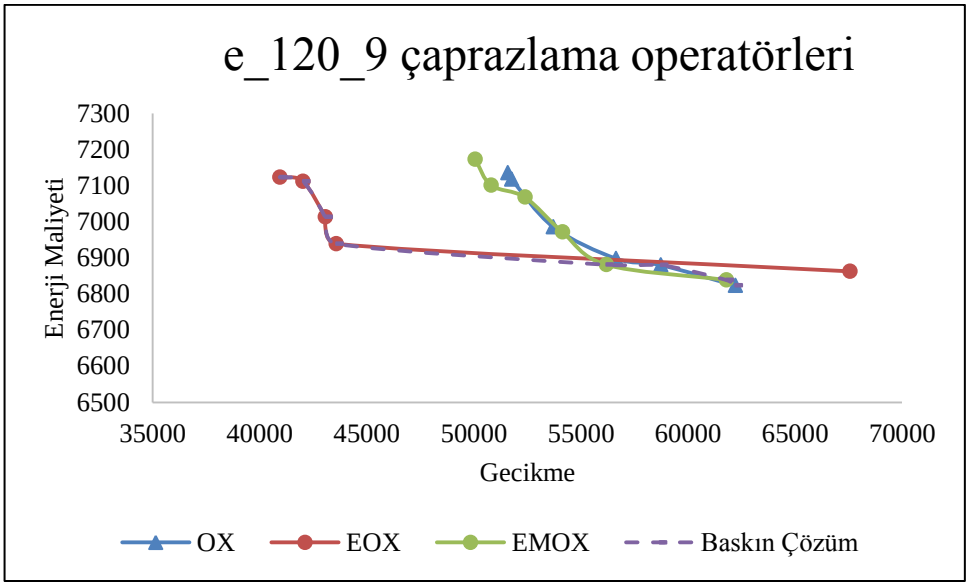
Şekil 4.114. e_120_9 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.115. e_120_9 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



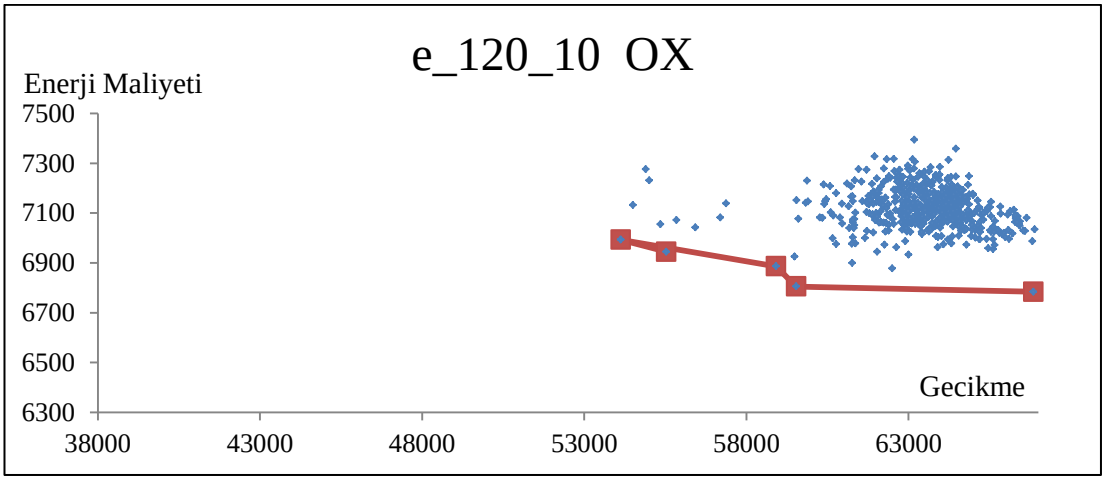
Şekil 4.116. e_120_9 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



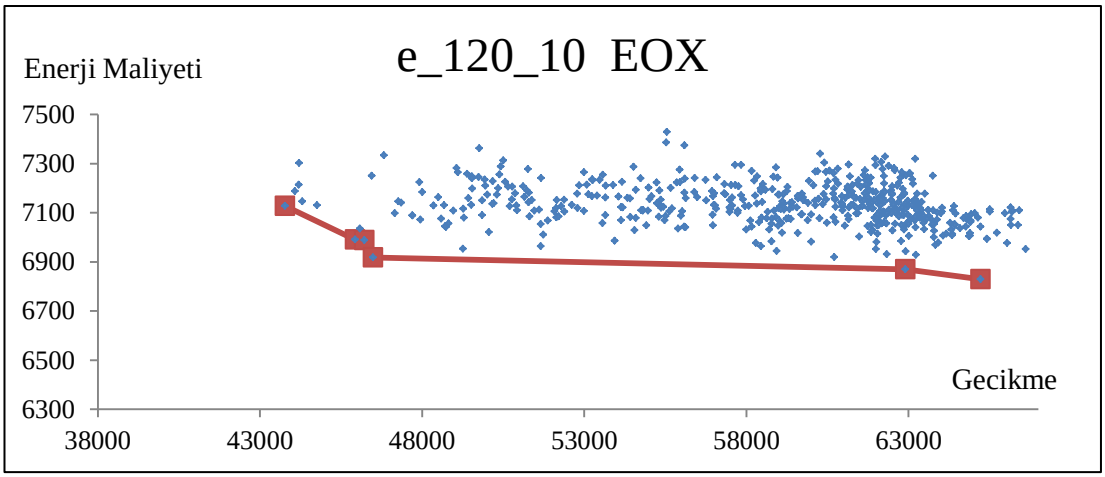
Şekil 4.117. e_120_9 problemi için çaprazlama operatörlerinin kıyaslaması

Tablo 4.39. e_120_10 problemi için baskın çözüm kümesi

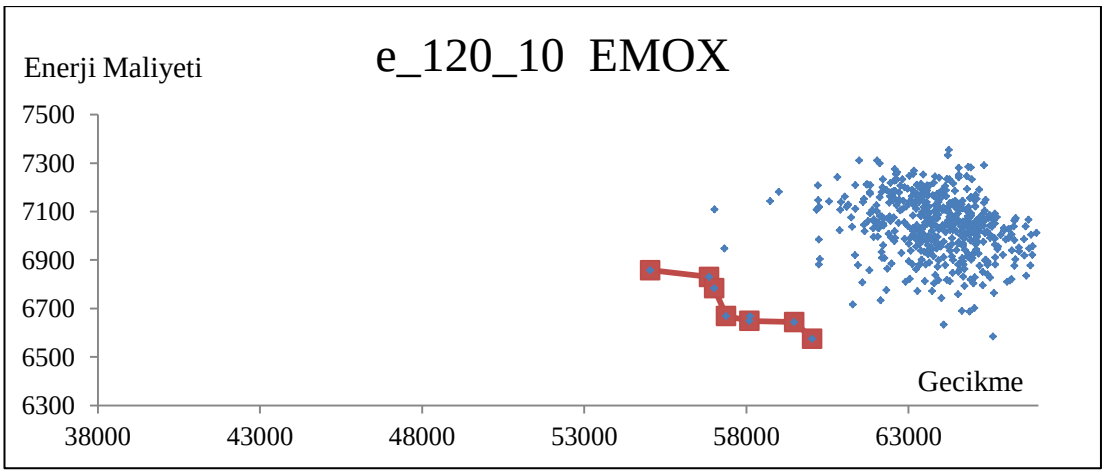
	Toplam Gecikme	Enerji Maliyeti	α	Tüm Çözümler İçinde Baskınlık
OX	66848	6784,16	0	
	59534	6804,92	10	
	58915	6886,28	20	
	54131	6992,68	30	
	55532	6944,76	40	
EOX	43769	7128,08	60	-
	45931	6991,16	30	-
	46219	6988,92	20	-
	46488	6918,04	10	-
	62904	6870,28	90	
	65219	6829,4	0	
EMOX	55037	6857,76	90	-
	56847	6830,36	70	-
	57007	6784,32	40	-
	57375	6668,88	50	-
	58097	6648,64	60	-
	59482	6644,4	10	-
	60026	6575,4	30	-



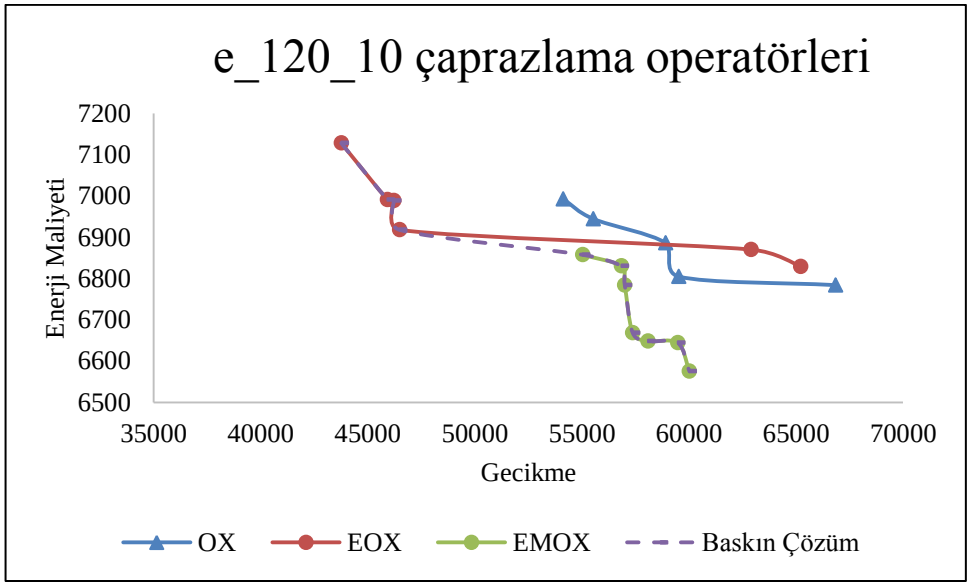
Şekil 4.118. e_120_10 problemi için OX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.119. e_120_10 problemi için EOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.120. e_120_10 problemi için EMOX çaprazlama operatörü ile elde edilen baskın çözüm kümesi



Şekil 4.121. e_120_10 problemi için çaprazlama operatörlerinin kıyaslaması

120 işlik problemler için elde edilen çözümlerden her iki amaç fonksiyonu için en iyi sonuçlara ait değerler Tablo 4.41 ve Tablo 4.42'de görülebilir. TMEMM yönteminin başarısı DGA ile kıyaslandığında problem sayısı 120 olduğunda daha iyidir. Toplam 10 problemin yedisinde TMEMM yöntemi DGA'ya göre daha başarılı sonuçlar vermiştir. Toplam gecikme zamanı içinse TMTGZM yönteminin genetik algoritmaya göre 120 işlik problemler için daha başarılı olduğu söylenebilir. 10 problemin dokuzunda TMTGZM yöntemi genetik algoritmadan daha iyi sonuç vermiştir. EOX çaprazlama yöntemi enerji maliyeti olarak 10 problemin yedisinde en kötü sonucu verirken, EMOX çaprazlama yöntemi de yine 10 problemin yedisinde en kötü sonucu vermiştir. 120 işlik problemler için de karar vericiye, alternatifleri değerlendirirken toplam gecikme zamanının daha kritik olduğu zamanlarda EOX çaprazlama operatörünü kullanması, maliyetlerin ön plana çıktığı durumlarda ise EMOX çaprazlama operatörünü kullanması önerilebilir. OX çaprazlama operatörü ise herhangi bir amaç için şekillendirilmediğinden ortada çözümler çıkarmıştır. Her bir çaprazlama operatörünün baskın çözümleri toplamda değerlendirildiğinde EOX'un ve EMOX'un baskın çözümlerinin OX'in ürettiği baskın çözümlerden daha da baskın olduğu gözlemlenmiştir. Tüm çözümler içindeki baskın çözüm kümelerinde OX operatörü 5 baskın çözüm üretebilirken, EOX 50, EMOX ise 42 baskın çözüm üretmiştir.

Tablo 4.40. 120 işlik problemlerin enerji maliyeti amacına göre değerlendirilmesi

Problem Adı	TMEMM	DGA	ga-ox	ga-eox	ga-emox
e_120_1	6551,32	6512	6998,48	6969,56	6731,28
e_120_2	6444,36	6529	6855,2	6919,92	6686,2
e_120_3	6512,04	6526	6986,44	7000	6734,24
e_120_4	6471,84	6581	7013,2	7004,6	6713,44
e_120_5	6463,08	6482	6900,76	6610,72	6649,24
e_120_6	6597	6718	7106,96	7158	6920,92
e_120_7	6462,32	6696	7123,28	7126,64	6850,16
e_120_8	6465,16	6368	6765,36	6802,08	6563,48
e_120_9	6413,28	6430	6823,6	6862,88	6838,88
e_120_10	6461,52	6347	6784,16	6829,4	6575,4

Tablo 4.41. 120 işlik problemlerin toplam gecikme amacına göre değerlendirilmesi

Problem Adı	TMTGZM	ga-ox	ga-eox	ga-emox
e_120_1	40001	41163	41502	53760
e_120_2	37466	49773	40565	52181
e_120_3	39540	53249	42476	52534
e_120_4	44303	54866	44816	55830
e_120_5	45739	57661	58237	57218
e_120_6	37959	48939	39549	52007
e_120_7	37553	51666	40135	53087
e_120_8	36872	49395	37926	49793
e_120_9	38684	51597	40941	50071
e_120_10	44266	54131	43769	55037

5. SONUÇLAR ve ÖNERİLER

Bu tez çalışmasında TOU tarifesi altında tek makine çizelgeleme problemi daha önceden ele alınmadığı bir şekilde alınmış ve modellenmiştir. Bu modelin daha basit ve çözülebilir bir hale getirilmesi için de zorluk çıkartan işlerin sabitlenmesi yöntemi önerilmiş ve kalan problem için de yine literatürde olmayan bir şekilde, farklı periyotları birleştirerek sırt çantası problemine dönüştürerek modelleme geliştirilmiştir. Daha önce yapılan çalışmalarda kullanılan değişken bölme ve lagrange gevşetimi gibi metotlar bu model üzerine uygulandığında problemin çözülmesi çok daha kolay alt problemlere dönüştüğü gözlemlenmiştir. Modelin bu haliyle çözdürülen problemlerin, iş sayısının 120 olduğu durumlarda aynı problemler için daha önce kullanılan ve bu tez çalışmasında geliştirilen metasezgisel yöntemlere göre çok daha iyi sonuçlar verdiği görülmektedir. 120'lik problemler için elde edilen enerji maliyetinin DGA'ya göre ortalamada yüzde 1 iyileştirme yaptığı söylenebilir. DGA'nın enerji maliyeti gözetilmeden yapılan herhangi bir sıralamaya göre ise %10 civarında bir iyileştirme yapıldığı bilinmektedir (Tacettin ve diğ., 2011). 60 işlik ve 90 işlik problemler için de önerilen yöntemin problemlerin %50'sinde daha başarılı olduğu gözlemlenmiştir.

İkinci olarak tek makine çizelgeleme problemi için toplam gecikme zamanı minimizasyonu çokça çalışılan bir problemdir. İşlem zamanlarının birbirine yakın olduğu özel bir durumda işlerin sırasının yaklaşık olarak nerede başlayacağı ile ilgili fikir verebileceği düşüncesinden yola çıkarak, işlerin yapıldığı sıradaki beklenen gecikme değerleri hesaplanmış ve bu değerler üzerinden problem atama problemine dönüştürülmüştür. Bununla ilgili olarak hangi durumda önerilen yöntemin iyi sonuç verdiği araştırılmış ve işlerin işlem zamanları için varyasyon katsayısı 0,35'den büyükse, önerilen yöntemin çok başarılı olamadığı, 0,1-0,35 aralığında problemlerin yarısında EDD'den daha iyi sonuç verdiği, varyasyon katsayısının 0,1'den küçük olduğu durumlarda ise %89 oranında EDD'den daha iyi sonuç verdiği görülmüştür. Önerilen yöntemin optimumla kıyaslanabilmesi içinse 10 adet 30 işlik problem oluşturulmuş ve optimumdan %5,2 uzaklaşıldığı gözlemlenmiştir.

Son olarak genetik algoritma kullanılarak herhangi bir çözümün daha önce her iki amaç için elde edilmiş optimuma yakın çözüm değerlerine yüzdesel olarak uzaklığa göre değerlenlendirme yapılmıştır. EOX çaprazlama operatörü, OX çaprazlama operatörü ve geliştirilen EMOX çaprazlama operatörü kullanılarak problemler çözdürüldüğünde hem 60 işlik, hem 90 işlik hem de 120 işlik problemler için EOX'un toplam gecikme zamanı olarak, EMOX'un enerji maliyeti olarak daha iyi sonuç verdiği görülmüştür. Her üç yöntemle elde edilen pareto-optimal kümeler grafiksel olarak gösterilmiş ve karar vericinin bu pareto-optimal sonuçlar arasından seçim yapabileceği öngörülmüştür. Genetik algoritma için kullanılan uygunluk fonksiyonu değeri;

$$F_k = \sum_{i=1}^{n} \frac{T_{ik}}{T} \alpha + \sum_{i=1}^{n} \frac{E_{ik}}{E} (100-\alpha)$$

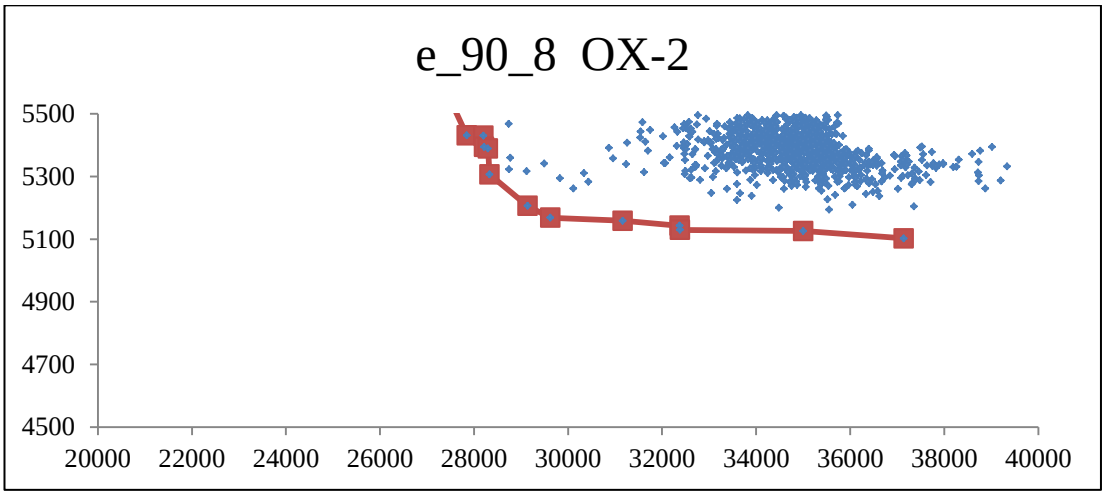
şeklindedir. α değeri 0'dan başlayarak 100'e kadar 10 artırılarak iki amacın ağırlıklandırılması farklılaştırılmıştır. Bu şekilde elde pareto optimal sonuçların sayısı eğer artırma miktarı 5 yapılırsa ne olacağı örnek bir problem üzerinde incelenmiştir.

Tablo 5.1. e_90_8 problemi için α 5 arttırıldığında oluşan baskın çözüm kümesi

	Toplam Gecikme	Enerji Maliyeti	α
OX	27586	5515,64	90
	27846	5431,64	100
	28200	5429,76	75
	28214	5394,4	85
	28299	5389,44	80
	28328	5306,8	30
	29140	5205,92	55
	29627	5168,36	20
	31162	5158,84	35
	32373	5142,6	10
	32380	5129,56	15
	35002	5125,68	5
	37137	5102,24	0

e_90_8 problemi α değerinin artırma miktarı 10 olduğunda, OX çaprazlama operatörü ile sadece 2 noktada pareto optimal sonuç vermiştir. α değerinin artırma

miktarı 5 yapıldığında ise elde edilen baskın çözüm kümesi Tablo 5.1'de görülebilir. Her bir α değeri için en iyi 50 çözümü gösteren ve toplam 1050 çözümü içeren grafiksel gösterim ise Şekil 5.1'de görülebilir. Daha öncesinde sadece 2 tane baskın çözüm varken, α değeri beşer beşer artırıldığında elde edilen baskın çözüm sayısı 13'e çıkmaktadır. Karar vericinin önüne daha fazla seçenekle çıkmak gerektiğinde α değerinin artma miktarını azaltmak, ileride çalışılması gereken bir durum olarak ortaya çıkmaktadır.



Şekil 5.1. e_90_8 problemi için α değeri 5 arttırıldığında OX çaprazlama operatörü ile elde edilen çözüm kümesi

Aynı zamanda uygunluk fonksiyonu değeri:

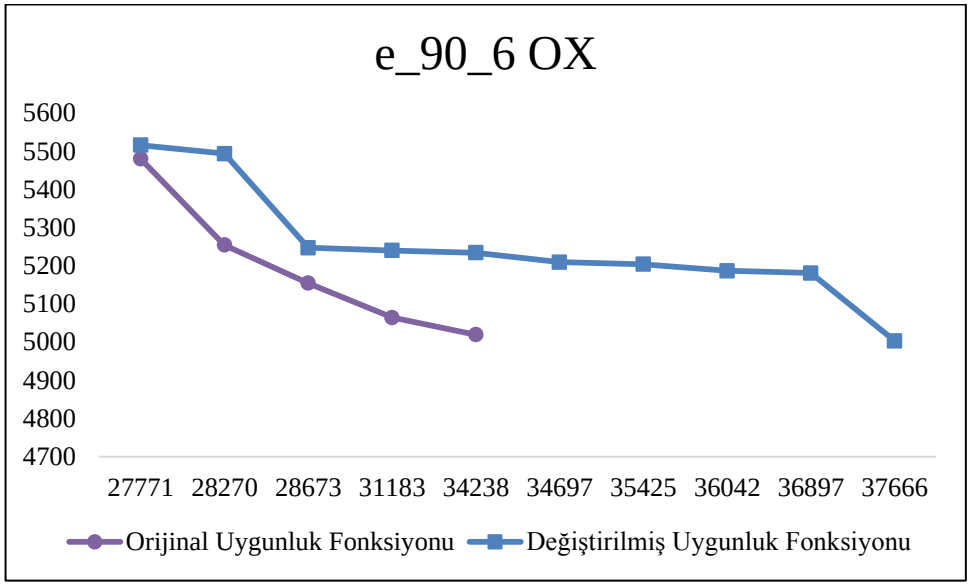
$$F_k = \sum_{i=1}^{n} \frac{T_{ik}}{T} \alpha + \sum_{i=1}^{n} \frac{E_{ik}}{E} (100-\alpha)$$

Yerine;

$$F_k = \sum_{i=1}^{n} T_{ik} \alpha + \sum_{i=1}^{n} E_{ik} (100-\alpha)$$

şeklinde hesaplanırsa ne olacağı da örnek bir problem üzerinde araştırılmıştır. e_90_6 problemi için OX çaprazlama operatörü ile elde edilen sonuç uygunluk fonksiyonu yukarıdaki gibi değiştirilirse ne olacağı incelendiğinde, değiştirilmiş uygunluk fonksiyonunun amaçların değerlerini normalize etmediğinden ve toplam gecikme zamanı değeri daha büyük olduğundan, enerji maliyeti amacı için iyi sonuçlar vermediği gözlemlenmiştir. Aynı problemin orijinal uygunluk fonksiyonu ve değiştirilmiş uygunluk fonksiyonuna göre baskın çözüm değerleri Şekil 5.2'de

gösterilmiştir. Grafikten de görüleceği üzere uygunluk fonksiyonun TMTGZM ve TMEMM ile elde edilen en iyi sonuç değerlerine bölünmesinin daha iyi sonuçlar elde etmek için daha yararlı olduğu söylenebilir.



Şekil 5.2. e_90_6 problemi için değiştirilmiş uygunluk fonksiyonu değeriyle orijinal uygunluk fonksiyonu değerinin karşılaştırılması

Tek makine çizelgeleme problemleri için önerilen yöntemler daha karmaşık problemlerin çözdürülmesinde temel teşkil etmektedir. Dolayısıyla, burada önerilen yöntemlerin başka problem tipleri için de kullanılabileceği söylenebilir.

Özetle, Tablo 5.2'de de görüleceği üzere problemlerin tek bir amacına özel geliştirilen yöntemler, genetik algoritma ilr geliştirilen çaprazlama yöntemlerine göre daha başarılıdır. TMEMM yöntemi ortalama olarak genetik algoritmada kullanılan 3 çaprazlama operatörüyle elde edilen en iyi çözümden %1,8 daha iyi sonuç vermiştir. TMTGZM yöntemi de ortalama olarak genetik algoritmada kullanılan 3 çaprazlama operatörüyle elde edilen en iyi çözümlerden %3,4 daha iyi sonuç vermiştir.

Konuyla ilgili, varsayımlarda belirtilen kısımlar kaldırılarak bir takım çalışmalar yapılabilir. Örneğin, periyotlar arası geçişteki işler için yapılan işin tamamı için başlangıçtaki periyoda ait birim fiyat yerine, periyotlarda harcadığı zamana göre yapılacak masraf dağıtımı üzerinde çalışılabilecek bir konudur. Benzer şekilde, ayar zamanlarının sıra bağımlı ayar olarak hesaplandığı durumlar göz önüne alınarak çizelgenin daha gerçekçi olması ile ilgili çalışma yapılabilir.

Tablo 5.2. Önerilen yöntemler ve genetik algoritma kıyaslaması

Problem Adı	Enerji Maliyeti TMEMM	Enerji Maliyeti GA-En iyi Çözüm	% Fark	Toplam Gecikme TMTGZM	Toplam Gecikme GA-En iyi Çözüm	% Fark
e_60_1	3251,12	3103,1	-4,8%	7721	7840	1,5%
e_60_2	3202,6	3348,8	4,4%	11877	11236	-5,7%
e_60_3	3251,44	3284,92	1,0%	11232	10848	-3,5%
e_60_4	3207,82	3348,28	4,2%	9209	9227	0,2%
e_60_5	3246,08	3447,64	5,8%	10321	10049	-2,7%
e_60_6	3284,04	3288,04	0,1%	12701	12282	-3,4%
e_60_7	3319,2	3291,48	-0,8%	9560	11068	13,6%
e_60_8	3248	3425,96	5,2%	11444	10922	-4,8%
e_60_9	3312	3256,96	-1,7%	9259	9186	-0,8%
e_60_10	3324,44	3418,16	2,7%	12414	11785	-5,3%
e_90_1	4734,12	4691,16	-0,9%	40001	41163	2,8%
e_90_2	4706,96	4821,56	2,4%	37466	40565	7,6%
e_90_3	4801,36	4631,12	-3,7%	39540	42476	6,9%
e_90_4	4768,8	4644,6	-2,7%	44303	44816	1,1%
e_90_5	4693,92	4839,12	3,0%	45739	57218	20,1%
e_90_6	4716,64	4898,6	3,7%	37959	39549	4,0%
e_90_7	4569,12	4743,12	3,7%	37553	40135	6,4%
e_90_8	4789,28	4944,04	3,1%	36872	37926	2,8%
e_90_9	4853,36	4612,08	-5,2%	38684	40941	5,5%
e_90_10	4732,88	4711,68	-0,4%	44266	43769	-1,1%
e_120_1	6551,32	6731,28	2,7%	40001	41163	2,8%
e_120_2	6444,36	6686,2	3,6%	37466	40565	7,6%
e_120_3	6512,04	6734,24	3,3%	39540	42476	6,9%
e_120_4	6471,84	6713,44	3,6%	44303	44816	1,1%
e_120_5	6463,08	6610,72	2,2%	45739	57218	20,1%
e_120_6	6597	6920,92	4,7%	37959	39549	4,0%
e_120_7	6462,32	6850,16	5,7%	37553	40135	6,4%
e_120_8	6465,16	6563,48	1,5%	36872	37926	2,8%
e_120_9	6413,28	6823,6	6,0%	38684	40941	5,5%
e_120_10	6461,52	6575,4	1,7%	44266	43769	-1,1%

KAYNAKLAR

Ashok S., Banerjee R., Optimal cool storage capacity for load management, *Energy*, 2003, **28**, 115-126.

Ashok S., Peak-load management in steel plants, *Applied Energy***,** 2006, **83**, 413-424.

Aktürk S., Yıldırım M. B., A new dominance rule for the weighted tardiness problem, *Production Planning & Control*, 1999, **10**, 138-149.

Avcı S., Aktürk M. S., Storer R. H., A problem space algorithm for single machine weighted tardiness problems, *IIE Transactions*, 2003, **35**, 479-486.

Baker K. R., Bertrand J. W., A dynamic priority rule for scheduling against due-dates, *Journal of Operational Management*, 1982, **3**, 37-42.

Bazaraa M. S., Jarvis J. J., *Linear Programming and Network Flows*, 1st ed., John Wiley & Sons, New York, 1977.

Bazaraa M. S., Sherali H. D., Shetty C. M., *Nonlinear Programming*, 3rd ed., John Wiley & Sons, New Jersey, 2006.

Beale E. M. L., Tomlin J. A., Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables, *Proceedings of the Fifth International Conference on Operations Research*, Tavistock Publications, London, 447-454, 1970.

Bentley P. J., Wakefield J. P., Finding acceptable solutions in the pareto-optimal range using multi-objective genetic algorithm, Springer Verlag Ltd. Research Report, RN/98/66, 1998.

Bilge Ü., Kurtulan M., Kıraç F., A tabu search algorithm for the single machine total weighted tardiness problem, *European Journal Of Operational Research*, 2007, **176**, 1423-1435.

Biskup D., Feldmann M., On scheduling around large restrictive common due windows, *European Journal of Operational Research,* 2005, **162**, 740–761.

Chalmet L., Gelders L., Lagrange relaxation for a generalized assignment-type problem, *Advances in Operations Research*, 1977, **1**, 103-109**.**

Chantaravarapan S., Jatinder N. D. G., A hybrid genetic algorithm for minimizing total tardiness on a single machine with family setups, *Production and Operations Management Society Conference Proceedings*, Savannah, USA, 4-7 April 2003.

Cheng T. C. E, Ng C. T., Yuan C. C., Liu Z. H., Single machine scheduling to minimize total weighted tardiness, *European Journal of Operational Research*, 2005, **165**, 423-443.

Congram R. K., Potts C. N., Van de Velde S. L., An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem, *Informs Journal on Computing*, 2002, **14**, 52-67.

Crauwels H. A. J., Potts C. N., Van Wassenhove L. N., Local search heuristics for single machine total weighted tardiness scheduling problem, *Informs Journal on Computing*, 1998, **10**, 341-350.

Çömlekçi N., *Deney tasarımı ilke ve teknikleri*, Alfa yayınları, İstanbul, 2003.

Du J., Leung J. Y. T., Minimizing total tardiness on one machine is NP-hard, *Mathematics of Operations Research*, 1990, **15**, 483–495.

Emmons H., One Machine sequencing to minimize certain functions of job tardiness, *Operations Research*, 1969, **17**, 701-715.

Engin O., Fığlalı A., Akış tipi çizelgeleme problemlerinin genetik algoritma yardımı ile çözümünde uygun çaprazlama operatörlerinin belirlenmesi, *Doğuş Üniversitesi Dergisi*, 2002, **6**, 27-35.

Farias I. R., Johnson E. L., Nemhauser G. L., A generalized assignment problem with special ordered sets: a polyhedral approach, *Mathematical Programming*, 2000, **89**, 187-203.

Faruqui A., George S., Quantifying customer response to dynamic pricing, *The Electricity Journal*, 2005, 53-63.

Faruqui A., Sergici S., Household response to dynamic pricing of electricity: A survey of 15 experiments, *Journal of Regulatory Economics*, 2010, **38**, 193-255.

French A. P., Wilson J. M., An LP-Based heuristic procedure for the generalized assignment problem with special ordered sets, *Computers&Operations Research*, 2007, **34**, 2359-2369.

Gafarov E. R., Lazarev A. A., Werner F., A note on a single machine scheduling problem with generalized total tardiness objective function, *Information Processing Letters*, 2012, **112**, 72-76.

Goldberg G. D. E., *Genetic algorithms in search, Optimization and machine learning*, 1st ed., Addison-Wesley, Boston, 1989.

Holsenback J. E., Russell R. M., A heuristic algorithm for sequencing on one machine to minimize total tardiness, *Journal of Operational Research Society*, 1992, **43**, 53-62.

Jeet V., Kutanoğlu E., Lagrangian relaxation guided problem space search heuristics for generalized assignment problems, *European Journal of Operational Research*, 2007, **182**, 1039-1056.

Jiju A., Preece D., *Understanding, managing and implementing quality frameworks, techniques and cases*, 1st ed., Routledge, London, 2002.

Jörnsten K., Nasberg M., A new lagrangian relaxation approach to the generalized assignment problem, *European Journal of Operational Research*, 1986, **27**, 313–323.

Karp R. M., Reducibility among combinatorial problems, in: Miller R. E., Thatcher J. W. (Eds), *Complexity of Computer Computations*, Plenum Pres, New York, 85-103, 1972.

Koulamas C., The total tardiness problem: review and extensions, *Operations Research*, 1994, **42**, 1025-1041.

Koulamas C., The single machine total tardiness scheduling problem: review and extensions, *European Journal of Operational Research*, 2010, **202**, 1-7.

Kumar R., Jyotishree M., Blending roulette wheel selection & rank selection in genetic algorithms, *International Journal of Machine Learning and Computing*, 2012, **2**, 365-370.

Lahlou C., Dauzère-Pérès S., Single-machine scheduling with time window-dependent processing times, *Journal of the Operational Research Society*, 2006, **57**, 133–139.

Lawler E. L., A "Pseudo Polynomial" algorithm for sequencing jobs to minimize total tardiness, *Annals of Discrete Mathematics*, 1977, **1**, 331-342.

Lazarev A. A., Werner F., Algorithms for special cases of the single machine total tardiness problem and an application to the even-odd partition problem, *Mathematical and Computer Modelling*, 2009, **49**, 2061-2072

Lee T., Chen C., Iteration particle swarm optimization for contract capacities selection of time-of-use rates industrial customers, *Energy Conservation and Management***,** 2007, **48**, 1120-1131.

Linzhong L., Mu H., Song Y., Luo H., Li X., The equilibrium generalized assignment problem and genetic algorithm, *Applied Mathematics and Computation*, 2012, **218**, 6526-6535.

Manne A. S., On the job-shop scheduling problem, *Operations Research***,** 1960, **8**, 219-223.

Martello S., Toth P., *Knapsack problems*, 1st ed., John Wiley & Sons, England, 1990.

Michalewicz Z., *Genetic algoritms + data structures = evolution programs*, 3rd ed., Springer, New York, 1996.

Özbakır L., Baykasoğlu A., Tapkan P., Bees algorithm for generalized assignment problem, *Applied Mathematics and Computation*, 2010, **215**, 3782-3795.

Panwalkar S. S., Smith M., Koulamas C., A heuristic for the single machine tardiness problem, *European Journal of Operatinoal Research*, 1993, **70**, 304-310.

Pinedo M. L., *Scheduling theory, algorithms and systems*, 3rd ed., Printice Hall, New York, 2008.

Potts C. N., Van Wassenhove L. N., Single machine tardiness sequencing heuristics, *IIE Transactions*, 1991, **23**, 346-354.

Ronconi D. P., Kawamura M. S., The single machine earliness and tardiness scheduling problem: lower bounds and a branch-and-bound algorithm, *Computational & Applied Mathematics*, 2010, **29**, 107-124.

Sadykov R., Integer programming-based decomposition approaches for solving machine scheduling problems, Dissertation, Universite Catholique De Louvain, Faculte des Sciences Appliquees, Louvain, 2006.

Sen T., Sulek J. M., Dileepan P., Static scheduling research to minimize weighted and unweighted tardiness: a state-of-art survey, *International Journal of Production Economics*, 2003, **83**, 1-12.

Srinivasan V., A hybrid algorithm for the one machine sequencing problem to minimize total tardiness, *Naval Research Logistics Quarter*, 1971, **18**, 317-327.

Srinivasan V., Thompson G., An algorithm for assigning uses to sources in a special class of transportation problems, *Operations Research*, 1973, **21**, 284-295.

Tacettin M., Terzi Ü., Fığlalı A., Zamana bağlı elektrik fiyatlandırmasına göre tek makine çizelgeleme problemi, *Endüstri Mühendisliği*, 2011, **22**, 2-13.

Tanaka S., Araki M., An exact algorithm for the single machine total weighted tardiness problem with sequence dependent setup times, *Computers & Operations Research*, 2013, **40**, 344-352.

Tang Y., Zheng G., Zhang S., Optimal control approaches of pumping stations to achieve energy efficiency and load shifting, *Electrical Power and Energy Systems*, 2014, **55**, 572-580.

Tseng C. T., Liao C. J., Huang K. L., Minimizing total tardiness on a single machine with controllable processing times, *Computers & Operations Research*, 2009, **36**, 1852-1858.

Woodcock A. C., Wilson J. M., A hybrid/tabu search/ branch & bound approach to solving generalized assigment problem, *European Journal of Operational Research*, 2010, **207**, 566-578.

Yeung W. K., Oğuz C., Cheng T. C. E., Single machine scheduling with a common due window, *Computers & Operations Research*, 2001, **28**, 157-175.

Yin Y., Wu C. C., Wu W. H., Cheng S. R., The single machine total weighted tardiness scheduling problem with position-based learning effects, *Computers& Operations Research*, 2012, **39**, 1109-1116.

Yang L., Dong C., Wan C., Ng C., Electricity time-of-use tariff with consumer behavior consideration, *International Journal of Production Economics*, 2013, **146**, 402-410.

Wu C. C., Cheng S. R., Wu W. H., Yin Y., Wu W. H., The single machine total tardiness problem with unequal release times and a linear deterioration, *Applied Mathematics and Computation*, 2013, **219**, 10401-10415.

EKLER

EK-A

Model 2

```
/*********************************************
 * OPL 6.3 Model
 * Author: MTACETTIN
 * Creation Date: Aug 10, 2010 at 11:57:55 AM
 *********************************************/
int number_of_jobs = ...;
int number_of_periods = ...;
range JOBS = 1..number_of_jobs;
range PERIODS =1..number_of_periods;
float cost[PERIODS] = ...;
int energy[JOBS] = ...;
int process_time[JOBS] = ...;
int A[PERIODS] = ...;
int B[PERIODS] = ...;
int selected_pt[PERIODS] = ...;
int selected_e[PERIODS] = ...;
int selected_duedate[PERIODS] = ...;
int Cmax = ...;
tuple subset { int job; int period;}
{subset} subsets = ...;
int due_date[JOBS] = ...;
dvar boolean X[JOBS,PERIODS];
maximize    sum (i in JOBS, j in 2..number_of_periods)
             ((X[i,j] -X[i,j-1]) * energy[i] * (5-cost[j]))+
            sum(i in JOBS)
            X[i,1] * energy[i] * (5-cost[1]);
subject to {
forall(j in 1..number_of_periods-1)
 sum(i in JOBS) (process_time[i]* X[i,j]) <= (B[j] - 1- sum(k in
1..j) selected_pt[k]);
 c2: sum(i in JOBS) process_time[i]* X[i,number_of_periods] <= Cmax
- sum(k in PERIODS) selected_pt[k];
 forall(i in JOBS, j in 1..number_of_periods-1)
  X[i,j+1] >= X[i,j];
 forall(j in 2..number_of_periods-1)
 sum(i in JOBS) process_time[i] * (X[i,j]- X[i,j-1]) <= B[j]-A[j]-1;
 c5: sum(i in JOBS) process_time[i] * (X[i,number_of_periods]-
X[i,number_of_periods-1]) <= Cmax-A[number_of_periods];
}


int toplam;
execute {
toplam = 0;
for ( var ii in JOBS) {
 for ( var jj in PERIODS) {
 if (jj != 1)
 {
  toplam =  toplam +((X[ii][jj] -X[ii][jj-1]) * energy[ii] *
cost[jj]);
  };
```

```
  };
  };
 for ( ii in JOBS) {
 toplam = toplam + X[ii][1] * energy[ii] * cost[1] };
 for ( jj in PERIODS) {
 toplam = toplam + selected_e[jj]* cost[jj] };
 writeln(toplam);
 }


main {
  thisOplModel.generate();

  var produce = thisOplModel;
  var def = produce.modelDefinition;
  var data = produce.dataElements;
  produce = new IloOplModel(def,cplex);

  data.Cmax = 0;
  for (var
m=0;m<(data.number_of_jobs);m++){data.Cmax=data.Cmax+data.process_ti
me[m+1]};
  for (var
m1=0;m1<(data.number_of_periods);m1++){data.Cmax=data.Cmax+data.sele
cted_pt[m1+1]};
  for (var m2=1;m2<=(data.number_of_periods);m2++){ if((m2%3) ==0) {
data.cost[m2]= 2} else if ((m2%3) ==1) { data.cost[m2]= 3.44}  else
data.cost[m2] = 1};
  for (var m3=1;m3<=(data.number_of_periods);m3++){ if((m3%3) ==0) {
data.A[m3]= (1440*(Opl.floor(m3/3)-1) +780)}  else if ((m3%3) ==1) {
data.A[m3]= 1440*(Opl.floor((m3-1)/3)) }  else data.A[m3] =
(1440*Opl.floor((m3-2)/3)+300)};
  for (var m4=1;m4<=(data.number_of_periods);m4++){ if((m4%3) ==0) {
data.B[m4]= (1440*(Opl.floor(m4/3)-1) +1440)} else if ((m4%3) ==1) {
data.B[m4]= (1440*Opl.floor((m4-1)/3)+300)}  else data.B[m4] =
((1440*Opl.floor((m4-2)/3))+780)};
  produce.addDataSource(data);
  produce.generate();
  cplex.tilim = 2000;
  cplex.preind = 0;
  cplex.PreslvNd = -1;
  cplex.RelaxPreInd = 0;
  cplex.RepeatPresolve = 0;
  cplex.solve();
  writeln(cplex.getObjValue());
  produce.postProcess();
 }
```

EK-B

Model 3-Yöntem 1

```
/*********************************************
 * OPL 6.3 Model
 * Author: Mustafa TACETTİN
  *********************************************/

int number_of_jobs = ...;
int number_of_periods = ...;
range JOBS = 1..number_of_jobs;
range PERIODS =1..number_of_periods;
float cost[PERIODS] = ...;
int energy[JOBS] = ...;
int process_time[JOBS] = ...;
int A[PERIODS] = ...;
int B[PERIODS] = ...;
int selected_pt[PERIODS] = ...;
int selected_e[PERIODS] = ...;
int selected_duedate[PERIODS] = ...;
int Cmax = ...;
int alfa = 38;
int due_date[JOBS] = ...;
tuple subset { int job; int period;}
{subset} subsets = ...;
dvar boolean X[JOBS,PERIODS];
dvar boolean Y[JOBS, PERIODS];
maximize     sum (i in JOBS, j in 2..number_of_periods)
              (Y[i,j] * energy[i] * (5-cost[j]))+
             sum(i in JOBS)
             X[i,1] * energy[i] * (5-cost[1])
              - sum(i in JOBS, j in 2..number_of_periods)
              (alfa*(Y[i,j]-X[i,j]+X[i,j-1]));
subject to {
forall(j in 1..number_of_periods-1)
 sum(i in JOBS) (process_time[i]* X[i,j]) <= (B[j] - 1- sum(k in
1..j) selected_pt[k]);
 c2: sum(i in JOBS) process_time[i]* X[i,number_of_periods] <= Cmax
- sum(k in PERIODS) selected_pt[k];
 forall(j in 2..number_of_periods-1)
 sum(i in JOBS) process_time[i] * (X[i,j]- X[i,j-1]) <= B[j]-A[j]-1;
 c5: sum(i in JOBS) process_time[i] * (X[i,number_of_periods]-
X[i,number_of_periods-1]) <= Cmax-A[number_of_periods];
 forall(i in JOBS, j in 1..number_of_periods-1)
 X[i,j]<=X[i,j+1];
 c6: forall (k in subsets)
 Y[k.job][k.period] - X[k.job][k.period]+X[k.job][k.period-1] >= 0;
 c7: forall (k in subsets)
 Y[k.job][k.period] - X[k.job][k.period]+X[k.job][k.period-1] <= 0;
}

int fark;
int toplamfark;
int toplam;
```

```
//subset newsubset;
 execute kisitekle{
 fark = 0;
 toplamfark = 0;
 for (var ii in JOBS) {
 for (var jj in PERIODS) {
 if (jj != 1)
 {
  fark = Y[ii][jj]-X[ii][jj]+X[ii][jj-1];
  if (fark != 0)
  {    subsets.add(ii,jj);
       toplamfark = toplamfark + fark;
  };
 };
 };
 };
 if (toplamfark == 0)
 {toplam = 0;
 for ( ii in JOBS) {
 for ( jj in PERIODS) {
 if (jj != 1)
 {
  toplam =  toplam +((X[ii][jj] -X[ii][jj-1]) * energy[ii] *
cost[jj]);
  };
  };
  };
 for ( ii in JOBS) {
 toplam = toplam + X[ii][1] * energy[ii] * cost[1] };
 for ( jj in PERIODS) {
 toplam = toplam + selected_e[jj]* cost[jj] };
 writeln(toplam);
 };
 }
main {
  //  var it = 0;
     var bestresult = 1000000;
     while (1) {
        cplex.clearModel();
        var produce = thisOplModel;
        var def = produce.modelDefinition;
        var data = produce.dataElements;
        produce = new IloOplModel(def,cplex);
        data.Cmax = 0;
        for (var
m=0;m<(data.number_of_jobs);m++){data.Cmax=data.Cmax+data.process_ti
me[m+1]};
        for (var
m1=0;m1<(data.number_of_periods);m1++){data.Cmax=data.Cmax+data.sele
cted_pt[m1+1]};
        for (var m2=1;m2<=(data.number_of_periods);m2++){ if((m2%3)
==0) { data.cost[m2]= 2} else if ((m2%3) ==1) { data.cost[m2]= 3.44}
else data.cost[m2] = 1};
        for (var m3=1;m3<=(data.number_of_periods);m3++){ if((m3%3)
==0) { data.A[m3]= (1440*(Opl.floor(m3/3)-1) +780)}  else if ((m3%3)
==1) { data.A[m3]= 1440*(Opl.floor((m3-1)/3)) }  else data.A[m3] =
(1440*Opl.floor((m3-2)/3)+300)};
        for (var m4=1;m4<=(data.number_of_periods);m4++){ if((m4%3)
==0) { data.B[m4]= (1440*(Opl.floor(m4/3)-1) +1440)} else if ((m4%3)
```

```
==1) { data.B[m4]= (1440*Opl.floor((m4-1)/3)+300)}  else data.B[m4]
= ((1440*Opl.floor((m4-2)/3))+780)};
        produce.addDataSource(data);
        produce.generate();
        cplex.preind = 0;
        cplex.PreslvNd = -1;
        cplex.RelaxPreInd = 0;
        cplex.RepeatPresolve = 0;
        cplex.tilim = 200;
    //    writeln("Iteration ",it);
        cplex.solve();
        produce.postProcess();
       writeln(cplex.getObjValue());
       if (bestresult>cplex.getObjValue())
       {bestresult = cplex.getObjValue()} else
       {writeln("best result is :", bestresult);
         writeln(produce.toplamfark);
        break;}
       writeln(produce.toplamfark);
      if (produce.toplamfark ==  0) {
      break;};
    };
    }
```

EK-C

Model 3 – Yöntem 2

```
/*********************************************
 * OPL 6.3 Model
 * Author: Mustafa TACETTİN
  *********************************************/

int number_of_jobs = ...;
int number_of_periods = ...;
range JOBS = 1..number_of_jobs;
range PERIODS =1..number_of_periods;
float cost[PERIODS] = ...;
int energy[JOBS] = ...;
int process_time[JOBS] = ...;
int A[PERIODS] = ...;
int B[PERIODS] = ...;
int selected_pt[PERIODS] = ...;
int selected_e[PERIODS] = ...;
int selected_duedate[PERIODS] = ...;
int Cmax = ...;
int alfa = 38;
int due_date[JOBS] = ...;
tuple subset { int job; int period;}
{subset} subsets = ...;
dvar boolean X[JOBS,PERIODS];
dvar boolean Y[JOBS, PERIODS];
maximize     sum (i in JOBS, j in 2..number_of_periods)
              (Y[i,j] * energy[i] * (5-cost[j]))+
             sum(i in JOBS)
             X[i,1] * energy[i] * (5-cost[1])
              - sum(i in JOBS, j in 2..number_of_periods)
              (alfa*(Y[i,j]-X[i,j]+X[i,j-1]));
subject to {
forall(j in 1..number_of_periods-1)
 sum(i in JOBS) (process_time[i]* X[i,j]) <= (B[j] - 1- sum(k in
1..j) selected_pt[k]);
 c2: sum(i in JOBS) process_time[i]* X[i,number_of_periods] <= Cmax
- sum(k in PERIODS) selected_pt[k];
 forall(j in 2..number_of_periods-1)
 sum(i in JOBS) process_time[i] * (X[i,j]- X[i,j-1]) <= B[j]-A[j]-1;
 c5: sum(i in JOBS) process_time[i] * (X[i,number_of_periods]-
X[i,number_of_periods-1]) <= Cmax-A[number_of_periods];
 forall(i in JOBS, j in 1..number_of_periods-1)
 X[i,j]<=X[i,j+1];
 c6: forall (i in JOBS, j in 2..number_of_periods)
 Y[i][j] - X[i][j]+X[i][j-1] >= 0;
 c7: forall (i in JOBS, j in 2..number_of_periods)
 Y[i][j] - X[i][j]+X[i][j-1] <= 0;
//c6: forall (k in subsets)
// Y[k.job][k.period] - X[k.job][k.period]+X[k.job][k.period-1] >=
0;
// c7: forall (k in subsets)
```

```
// Y[k.job][k.period] - X[k.job][k.period]+X[k.job][k.period-1] <=
0;
}

int fark;
int toplamfark;
int toplam;
//subset newsubset;
 execute kisitekle{
 fark = 0;
 toplamfark = 0;
 for (var ii in JOBS) {
 for (var jj in PERIODS) {
 if (jj != 1)
 {
  fark = Y[ii][jj]-X[ii][jj]+X[ii][jj-1];
  if (fark != 0)
  {    subsets.add(ii,jj);
       toplamfark = toplamfark + fark;
  };
 };
 };
 };
 if (toplamfark == 0)
 {toplam = 0;
 for ( ii in JOBS) {
 for ( jj in PERIODS) {
 if (jj != 1)
 {
  toplam =  toplam +((X[ii][jj] -X[ii][jj-1]) * energy[ii] *
cost[jj]);
  };
  };
  };
 for ( ii in JOBS) {
 toplam = toplam + X[ii][1] * energy[ii] * cost[1] };
 for ( jj in PERIODS) {
 toplam = toplam + selected_e[jj]* cost[jj] };
 writeln(toplam);
 };
 }
main {
  //  var it = 0;
     var bestresult = 1000000;
     thisOplModel.settings.mainEndEnabled = true;
     thisOplModel.generate();
     var data1 = thisOplModel.dataElements;
     var m1Source = new IloOplModelSource("model2-1.mod");
     var m1Cplex = new IloCplex();
     var m1Def = new IloOplModelDefinition(m1Source);
     var m1Opl = new IloOplModel(m1Def,m1Cplex);
     data1.Cmax = 0;
     for (var
m=0;m<(data1.number_of_jobs);m++){data1.Cmax=data1.Cmax+data1.proces
s_time[m+1]};
     for (var
m1=0;m1<(data1.number_of_periods);m1++){data1.Cmax=data1.Cmax+data1.
selected_pt[m1+1]};
```

```
     for (var m2=1;m2<=(data1.number_of_periods);m2++){ if((m2%3)
==0) { data1.cost[m2]= 2} else if ((m2%3) ==1) { data1.cost[m2]=
3.44}  else data1.cost[m2] = 1};
     for (var m3=1;m3<=(data1.number_of_periods);m3++){ if((m3%3)
==0) { data1.A[m3]= (1440*(Opl.floor(m3/3)-1) +780)}  else if
((m3%3) ==1) { data1.A[m3]= 1440*(Opl.floor((m3-1)/3)) }  else
data1.A[m3] = (1440*Opl.floor((m3-2)/3)+300)};
     for (var m4=1;m4<=(data1.number_of_periods);m4++){ if((m4%3)
==0) { data1.B[m4]= (1440*(Opl.floor(m4/3)-1) +1440)} else if
((m4%3) ==1) { data1.B[m4]= (1440*Opl.floor((m4-1)/3)+300)}  else
data1.B[m4] = ((1440*Opl.floor((m4-2)/3))+780)};
     m1Opl.addDataSource(data1);
     m1Opl.generate();
     m1Cplex.tilim = 100;
     m1Cplex.solve();
     writeln(m1Cplex.getObjValue());
     while (1) {
        cplex.clearModel();
        var produce = thisOplModel;
        var def = produce.modelDefinition;
        var data = produce.dataElements;
        produce = new IloOplModel(def,cplex);
        data.Cmax = 0;
        for (
m=0;m<(data.number_of_jobs);m++){data.Cmax=data.Cmax+data.process_ti
me[m+1]};
        for (
m1=0;m1<(data.number_of_periods);m1++){data.Cmax=data.Cmax+data.sele
cted_pt[m1+1]};
        for ( m2=1;m2<=(data.number_of_periods);m2++){ if((m2%3)
==0) { data.cost[m2]= 2} else if ((m2%3) ==1) { data.cost[m2]= 3.44}
else data.cost[m2] = 1};
        for ( m3=1;m3<=(data.number_of_periods);m3++){ if((m3%3)
==0) { data.A[m3]= (1440*(Opl.floor(m3/3)-1) +780)}  else if ((m3%3)
==1) { data.A[m3]= 1440*(Opl.floor((m3-1)/3)) }  else data.A[m3] =
(1440*Opl.floor((m3-2)/3)+300)};
        for ( m4=1;m4<=(data.number_of_periods);m4++){ if((m4%3)
==0) { data.B[m4]= (1440*(Opl.floor(m4/3)-1) +1440)} else if ((m4%3)
==1) { data.B[m4]= (1440*Opl.floor((m4-1)/3)+300)}  else data.B[m4]
= ((1440*Opl.floor((m4-2)/3))+780)};
        produce.addDataSource(data);
        produce.generate();
        var vectors = new IloOplCplexVectors();
        // We attach the values (defined as data) as starting
solution
        // for the variables x.
        //vectors.attach(produce.X,m1Opl.X);
        //vectors.attach(produce.Y,m1Opl.Y);
        vectors.getVectors(m1Cplex);
        vectors.setVectors(cplex);
        cplex.preind = 0;
        cplex.PreslvNd = -1;
        cplex.RelaxPreInd = 0;
        cplex.RepeatPresolve = 0;
        cplex.tilim = 400;
    //    writeln("Iteration ",it);
        cplex.solve();
        produce.postProcess();
       writeln(cplex.getObjValue());
       if (bestresult>cplex.getObjValue())
```

```
        {bestresult = cplex.getObjValue()} else
        {writeln("best result is :", bestresult);
          writeln(produce.toplamfark);
         break;}
        writeln(produce.toplamfark);
       if (produce.toplamfark ==  0) {
       break;};   };    }
```

EK-D

```
/*********************************************
 * OPL 6.3 Model
 * Creation Date: 29.Haz.2013 at 21:41:36
 *********************************************/
int number_of_jobs = ...;
int number_of_periods = ...;
range JOBS = 1..number_of_jobs;
range PERIODS = 1..number_of_periods;
int process_time[JOBS] = ...;
int energy[JOBS] = ...;
int A[PERIODS] = ...;
int B[PERIODS] = ...;
int selected_pt[PERIODS] = ...;
int selected_e[PERIODS] = ...;
//int selected_duedate[PERIODS] = ...;
int due_date[JOBS] = ...;
float cost[PERIODS] = ...;
float maliyet[JOBS, JOBS] = ...;
tuple subset { int job; int period;}
{subset} subsets = ...;
dvar boolean x[JOBS,JOBS];
minimize    sum( i in JOBS, j in JOBS) maliyet[i,j]*x[i,j];
subject to {
forall(i in JOBS)
 sum(j in JOBS) x[i,j] >=1;
forall(j in JOBS)
 sum(i in JOBS) x[i,j] >=1;
forall(i in JOBS)
 sum(j in JOBS) x[i,j] <=1;
forall(j in JOBS)
 sum(i in JOBS) x[i,j] <=1;
 }
 main {
     var TPrTime = 0;
     var TSayac = 0;
     var cmaliyet = 0;
     thisOplModel.generate();
     var data1 = thisOplModel.dataElements;
     var produce = thisOplModel;
     var def = produce.modelDefinition;
     produce = new IloOplModel(def,cplex);
     for (var i=0; i< (data1.number_of_jobs);i++)
     {
      TPrTime = 0;
      TSayac = 0;
      for (var m=0;m<(data1.number_of_jobs);m++){
           if ((data1.due_date[m+1]< data1.due_date[i+1]) ||
(data1.process_time[m+1]<data1.process_time[i+1]))
          { TPrTime=TPrTime+data1.process_time[m+1];
            TSayac = TSayac +1;
          }};
      for (var j=0; j< (data1.number_of_jobs);j++)
       {
```

```
      //cmaliyet = (TPrTime-
data1.process_time[i+1])/((data1.number_of_jobs)-
1)*(j)+data1.process_time[i+1]-data1.due_date[i+1];
         // if (j+1<TSayac)
         // {
          cmaliyet = (TPrTime/TSayac)*j+data1.process_time[i+1]-
data1.due_date[i+1];
         // }
          //if (j+1>=TSayac)
         // {
         // cmaliyet = 1000000;
         // }

      if (cmaliyet <0)

      {
      // writeln(cmaliyet);
      // writeln(TPrTime);
      // writeln(data1.process_time[i+1]);
      // writeln(data1.due_date[i+1]);
      // writeln(i);
      // writeln(j);
      // writeln(data1.process_time[j+1]);
      // writeln(data1.due_date[j+1]);

       cmaliyet = 0;
       }


       data1.maliyet[i+1][j+1] = cmaliyet;
      }
     };


     produce.addDataSource(data1);
     produce.generate();
     cplex.solve();
     var objfunc = 0;
     for (var k=0; k<120;k++)
     {
      for (var l=0;l<120;l++)
       { if (produce.x[k+1][l+1]>0)
          {writeln(k+1,' işi sırası ',l+1);
           objfunc = produce.maliyet[k+1][l+1]+objfunc;
           }
       }
     }
     writeln(cplex.getObjValue());
     writeln(objfunc);


        }
```

EK-E

```
clear;
n = 120;
duedate = zeros(n);
proses =zeros(n);
global iga x y
% Setup the GA
%ff='tspfun'; % objective function
npar=120; % # optimization variables
%_____________________________________________________
% Stopping criteria
maxit=2000; % max number of iterations
%_____________________________________________________
% other parameters
alfa = -10;
t_best = 44266;
e_best = 6461.52;
eox = 0;
% GA parameters
popsize=100; % set population size
mutrate=.05; % set mutation rate
selection=0.4; % fraction of population kept
keep=floor(selection*popsize); % #population members
% that survive
M=ceil((popsize-keep)/2); % number of matings
odds=1;
for ii=2:keep
odds=[odds ii*ones(1,ii)];
end
Nodds=length(odds);
while alfa<91
   alfa = alfa+10;

%_____________________________________________________
% Create the initial population
iga=0; % generation counter initialized
pop = cell(popsize,1);
for i = 1:popsize
   pop{i} = randperm(npar);
end

cost=fitness_function(pop,npar,proses,duedate,energy,alfa, t_best, e_best);
% calculates population cost
% using ff
```

```
[cost,ind]=sort(cost); % min cost in element 1
pop=pop(ind,:); % sort population with lowest
% cost first
minc(1)=min(cost); % minc contains min of
% population
meanc(1)=mean(cost); % meanc contains mean of population
%_____________________________________________________
% Iterate through generations

while iga<maxit
iga=iga+1; % increments generation counter
%_____________________________________________________

parent_a = ceil(rand*keep);
parent_b = ceil (rand*keep);
cuttingpoint1 = ceil (rand*(npar-2));
cuttingpoint2 = ceil (rand*(npar-cuttingpoint1-1));
cuttingpoint2 = cuttingpoint2+cuttingpoint1;
Cromozom_a = pop{parent_a};
Cromozom_b = pop{parent_b};
cut_a = ones(1,cuttingpoint2-cuttingpoint1);
cut_b = ones(1,cuttingpoint2-cuttingpoint1);
for i = 1:cuttingpoint2-cuttingpoint1
   cut_a(i) = Cromozom_a(cuttingpoint1+i);
end
for i = 1:cuttingpoint2-cuttingpoint1
   cut_b(i) = Cromozom_b(cuttingpoint1+i);
end

dif_a = setdiff(cut_a, cut_b);
dif_b = setdiff(cut_b, cut_a);
j=1;
for i = 1:cuttingpoint1
   bulundu = 0;
   for k = 1:cuttingpoint2-cuttingpoint1
      if (Cromozom_a(i) == cut_b(k) ) && (bulundu == 0)
         Cromozom_a(i)= dif_a(j);
         j = j+1;
         bulundu =1;

      end
   end
end

for i= cuttingpoint2+1:npar
   bulundu = 0;
   for k = 1:cuttingpoint2-cuttingpoint1
      if (Cromozom_a(i) == cut_b(k)) && (bulundu == 0)
         Cromozom_a(i)= dif_a(j);
```

```
            j = j+1;
            bulundu = 1;
        end
    end
end
for i = cuttingpoint1+1:cuttingpoint2
   Cromozom_a(i) = cut_b(i-cuttingpoint1);
end;


j=1;
for i = 1:cuttingpoint1
    bulundu = 0;
    for k = 1:cuttingpoint2-cuttingpoint1
        if (Cromozom_b(i) == cut_a(k) ) && (bulundu == 0 )
            Cromozom_b(i)=dif_b(j);
            j = j+1;
            bulundu = 1;
        end
    end
end

for i= cuttingpoint2+1:npar
    bulundu = 0;
    for k = 1:cuttingpoint2-cuttingpoint1
        if (Cromozom_b(i) == cut_a(k) ) && (bulundu == 0)
            Cromozom_b(i)=dif_b(j);
            j = j+1;
            bulundu = 1;
        end
    end
end
for i = cuttingpoint1+1:cuttingpoint2
   Cromozom_b(i) = cut_a(i-cuttingpoint1);
end;



% OX u EOX yapmak için emmons kuralı
if eox == 1

bulundu = 0;
for i = 1:npar
    for j = i:npar
        bulundu = 0;
        k =  Cromozom_a(i);
        p1 = proses(k);
        d1 = duedate(k);
        l =  Cromozom_a(j);
```

```
        p2 = proses(l);
        d2 = duedate(l);
        if p1>p2 && d1>d2 && bulundu == 0
            Cromozom_a(i) = l;
            Cromozom_a(j) = k;
            bulundu = 1;
        end
    end
end
bulundu = 0;
for i = 1:npar
    for j = i:npar
        bulundu = 0;
        k =  Cromozom_b(i);
        p1 = proses(k);
        d1 = duedate(k);
        l =  Cromozom_b(j);
        p2 = proses(l);
        d2 = duedate(l);
        if p1>p2 && d1>d2 && bulundu == 0
            Cromozom_b(i) = l;
            Cromozom_b(j) = k;
            bulundu = 1;
        end
    end
end
end
% end of EOX

% OX u ECOX yapmak için emmons kuralı
if eox == 2

bulundu = 0;
for i = 1:npar
    for j = i:npar
        bulundu = 0;
        k =  Cromozom_a(i);
        p1 = proses(k);
        e1 = energy(k);
        periyot1 =  periyot(Cromozom_a, i, n, proses);
        l =  Cromozom_a(j);
        p2 = proses(l);
        e2 = energy(l);
        periyot2 =  periyot(Cromozom_a, j, n, proses);
        if p1==p2 && e1>e2 && periyot1>periyot2
            Cromozom_a(i) = l;
            Cromozom_a(j) = k;
            bulundu = 1;
        end
```

```
   end
end
bulundu = 0;
for i = 1:npar
   for j = i:npar
      bulundu = 0;
      k =  Cromozom_b(i);
      p1 = proses(k);
      e1 = energy(k);
      periyot1 =  periyot(Cromozom_b, i, n, proses);
      l =  Cromozom_b(j);
      p2 = proses(l);
      e2 = energy(l);
      periyot2 =  periyot(Cromozom_b, j, n, proses);
      if p1==p2 && e1>e2 && periyot1>periyot2
         Cromozom_b(i) = l;
         Cromozom_b(j) = k;
         bulundu = 1;
      end
   end
end
end
% end of CEOX

cur_size = size(pop);
pop{cur_size(1)+1} = Cromozom_a;
pop{cur_size(1)+2} = Cromozom_b;

%_____________________________________________________________
% Mutate the population
   nmut=ceil(popsize*npar*mutrate);
   for ic = 1:nmut
   row1=ceil(rand*(popsize-1))+1;
   col1=ceil(rand*npar);
   col2=ceil(rand*npar);
   temp=pop{row1}(col1);
   pop{row1}(col1)=pop{row1}(col2);
   pop{row1}(col2)=temp;
   im(ic)=row1;
   end
cost= fitness_function(pop,npar,proses,duedate,energy,alfa, t_best, e_best);
%_____________________________________________________________
% Sort the costs and associated parameters
part=pop; costt=cost;
[cost,ind]=sort(cost);
pop=pop(ind,:);
%_____________________________________________________________
% Do statistics
minc(iga)=min(cost);
```

```
meanc(iga)=mean(cost);
end %iga
cost_t = fitness_tardiness(pop,npar,proses,duedate,energy,alfa, t_best, e_best);
cost_e = fitness_energy(pop,npar,proses,duedate,energy,alfa, t_best, e_best);
for i = 1:50
select_50{alfa*5+i} = [cost_t(i),cost_e(i),cost(i), alfa];
end

%_____________________________________________________
end %alfa
% Displays the output

for i = 1:550
disp([num2str(select_50{i})])
end
disp('baskin sonuclar');
v = 1;
for i = 1:550
   cik = 0;
   for j = 1:550
      if i~=j
         if select_50{i}(1)>select_50{j}(1) && select_50{i}(2)>select_50{j}(2) &&
cik==0
            cikar{v} = [i];
            v= v+1;
            cik = 1;
         end
      end
   end
end


v= v-1;
for i = 1:550
  finded = 0;
   for j = 1:v
    if cikar{j}(1) == i
       finded = 1;
    end
   end
   if finded == 0
      disp([num2str(select_50{i})]);
   end
end
```

KİŞİSEL YAYIN VE ESERLER

[1] **Tacettin M.**, Ünlüyurt T., An alternative proof that exact inference problem on bayesian belief networks is NP-hard, *Lecture Notes in Computer Sciences*, 2005, **3733**, 947-955.

[2] **Tacettin M.,** Ünal A. T., Brisa-Icron çizelgeleme sistemi ve dağıtık, açık çizelgeleme yaklaşımının bir uygulaması, *YA/EM 28. Ulusal Kongresi Sanayi Proje Yarışması*, Galatasaray Üniversitesi, İstanbul, 30 Haziran-2 Temmuz 2008.

[3] **Tacettin M.**, Fığlalı A., Yavuz M., An algorithm for banbury mixer scheduling, *Proceedings of 6th international symposium on intelligent and manufacturing systems*, IMS 2008-069, Sakarya, Turley, 2008.

[4] **Tacettin M.**, Fığlalı A., Terzi Ü., Zamana bağlı elektrik fiyatlandırmasına bağlı olarak tek makina çizelgeleme problemi, *YA/EM 29. Ulusal Kongresi*, Bilkent Üniversitesi, Ankara, 22-24 Haziran 2009.

[5] **Tacettin M.**, Fığlalı A., Terzi Ü., Zamana bağlı elektrik fiyatlandırmasına bağlı olarak tek makina çizelgeleme problemi, *Endüstri Mühendisliği*, 2011, **22**, 2-13.

ÖZGEÇMİŞ

1978'de Kayseri'de doğdu. İlkokul ve ortaokulu Kayseri'de, liseyi Ankara'da Ankara Fen Lisesi'nde okudu. 2000 yılında Bilkent Üniversitesi Endüstri Mühendisliği bölümünü bitirdi. 2002 yılında ise Sabancı Üniversitesinde Endüstri Mühendisliği bölümünde yüksek lisansını tamamladı. 2008 yılında Kocaeli Üniversitesi Endüstri Mühendisliği Ana Bilim Dalında doktora programına başladı. 2000-2001 yıllarında Sabancı Üniversitesinde Araştırma Görevlisi olarak, 2001-2003 yılları arasında IMS firmasında uygulama danışmanı olarak çalışmıştır. 2003 yılından bugüne kadar da Brisa'da görev yapmaktadır.