

Gerçek zamanlı çoklu-robot çoklu-hedef problemi için artımlı atama yöntemi

Sanem SARIEL^{1*}, Nadia ERDOĞAN¹, Tucker BALCH²

¹ İTÜ Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Programı, 34469, Ayazağa, İstanbul
² Georgia Institute of Technology, College of Computing, Atlanta, GA, 30332

Özet

Bu çalışmada, Çoklu-Robot Çoklu-Hedef Ataması problemi için bir gerçek zamanlı görev seçme ve atama yöntemi önerilmektedir. Çoklu-Robot Çoklu-Hedef Ataması problemi, literatürde iyi bilinen ve tüm veri kümeleri için en iyi çözümlerin polinomsal algoritmalarla bulunamadığı MTSP (Multiple Traveling Salesman Problem) probleminin her bir hedefin en az bir robot tarafından ziyaret edilmesini gerektiren değişik bir uyarlamasıdır. Bu problem için farklı eniyileme amaç fonksiyonları tanımlanabilir (Örn. yürütme zamanını iyileme, robotların toplam yollarını iyileme, vb.). Bu makalede, bu problemin gerçek dünya versiyonu ele alınıp, yürütme zamanında ortaya çıkabilecek problemler irdelenmiştir. Çoklu-robot sistemlerinin ortaklaşa çalışmalarında karşılaşılan en büyük güçlükler, görevlerin yürütme zamanının önceden kesin olarak tahmin edilememesi, yürütme süresince değişebilen maliyet değerleri ile belirsizlik ve tutarsızlıklardan kaynaklanır. Bu çalışmada, yürütme zamanı kısıtlamalarını da aşmak üzere etkin bir dinamik görev seçim ve atama yöntemi önerilmekte ve önerilen yöntemin, hem benzetim ortamlarında hem de gerçek robotlar üzerinde yapılan testlerle başarımlı analizi yapılmaktadır. Önerilen yöntem, yürütme zamanında oluşabilecek birçok hataya karşı dayanıklı olarak sistemin güncel durumuna göre artımlı ve dağıtılmış bir görev ataması gerçekler. Gerçek zamanlı yürütmeye uygun şekilde iletişim gereksinimleri minimumda tutulmaya çalışılmıştır. Gerçeklenen testler yöntemin bilgi-işlemsel açıdan etkin ve düşük maliyetli, sistemin hatalara karşı dayanıklı olmasını sağlayacak şekilde çalıştığını gösterir niteliktedir.

Anahtar Kelimeler: Çoklu-robot sistemleri, gerçek zamanlı çoklu-robot çoklu-hedef atama, artımlı görev seçimi, dinamik görev atama, hataya dayanıklılık.

*Yazışmaların yapılacağı yazar: Sanem SARIEL. sariel@itu.edu.tr; Tel: (212) 285 38 52.

Bu makale, birinci yazar tarafından İTÜ Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Programında tamamlanmış olan "An integrated planning, scheduling and execution framework for multi-robot coordination and cooperation" adlı doktora tezinin bir bölümünden hazırlanmıştır. Makale metni 22.06.2007 tarihinde dergiye ulaşmış, 29.08.2007 tarihinde basım kararı alınmıştır. Makale ile ilgili tartışmalar 31.03.2009 tarihine kadar dergiye gönderilmelidir.

Incremental task selection and allocation for the real-world multiple travelling robot problem

Extended abstract

In this study, the real-world Multiple Travelling Robot Problem (MTRP) is analyzed and an integrated approach is proposed to solve this problem in real time. MTRP is a generalization of the well-known Multiple Traveling Salesman Problems and is solved by a multi-robot team. In MTRP, a different version of the well known NP-hard MTSP (Multiple Traveling Salesman Problem), each target must be visited by at least one robot in its open tour. Depending on the selected application domain, various objectives may be defined for this problem (e.g. minimization of total path length, time, makespan etc.). Although this problem has emerged from Operations Research, it has become one of the main problems in multi-robot research for different missions (e.g., search and rescue, space, or surveillance operations, etc.). In this article, besides the target allocation issue, the real challenges of this problem are presented. Unpredictability of the exact processing times of tasks, unstable cost values during runtime and inconsistencies due to uncertain information form the main difficulties of the task allocation problem for robot systems. Since the real world is beyond the control of robots, in most cases, the Operations Research solutions are not directly applicable due to either robot hardware/software limitations or environmental dynamics. These approaches may become impractical when the size of the mission is even moderate or the cost values change frequently because of the uncertain knowledge, changes in the environment (including failures) or the changing structure of the mission (e.g. online tasks). Furthermore, robots have continuous path planning burdens for target sets in dynamic environments. Expensive computational efforts for initial allocations may become redundant.

We propose a solution as a generalized framework – DEMiR-CF (Distributed and Efficient Multi-Robot Cooperation Framework) –which includes dynamic task selection, distributed task allocation and contingency handling mechanisms. These mechanisms and lower level robot controllers, the motor and sensory modules are integrated together to solve the real-world MTRP.

DEMiR-CF, while capable of handling diverse contingencies, performs an incremental task allocation

method based on the current information about the environment. Globally efficient solutions are obtained by the proposed mechanisms that form priority based rough schedules and select the most suitable tasks from these schedules. Rough schedules are formed by using information regarding the beliefs about the other robots. Since DEMiR-CF is for real-world task execution, communication requirements are kept at minimum as much as possible. The approach is distributed and computationally efficient. Target allocation and route construction is integrated into each other by an incremental assignment approach. Real time conditions and contingencies that change the problem instance are handled at the same time by the designed contingency handling mechanisms. Robots keep system models to correct models of their own or warn other robots to maintain system consistency.

Empirical evaluations of the system performed on the Webots simulator and on Khepera II robots reveal the efficiency of the integrated components of the approach. Experiments are designed in three sets. In the first set of the experiments, evaluations of the proposed cost functions to be used with DEMiR-CF are performed. Comparisons are made both with the optimal results taken from the IP solver CPLEX and that of the Prim Allocation approach, one of the efficient methods to solve this problem. As the results illustrate, allocating all targets from scratch and generating routes of robots may result in suboptimal solutions. Therefore the target allocation and the route construction should be integrated for efficient heuristic approaches. This integration and incremental allocation is also useful for eliminating redundant calculations especially in highly dynamic or unknown environments. In the second set of experiments, scalability of the framework and the response efficiency of the contingency handling mechanism integrated into the framework are evaluated. The scalability of the approach is validated and the efficiency of using the contingency handling mechanisms is observed in the results. The third set of experiments is performed on real robots. As results illustrate, both incremental task selection and allocation approaches produce efficient results and the contingency handling mechanisms make the system robust and allow the handling of real-time online situations.

Keywords: Multi-robot systems, multi-robot multi-target assignment in real-time, incremental task selection, dynamic task allocation, robustness.

Giriş

İyi tasarlanmış ortaklaşa çalışan çoklu-robot sistemleri, arama kurtarma çalışmaları, uzay araştırmaları ve korumaya yönelik askeri uygulamalarda etkin şekilde kullanılmak üzere geniş bir kullanım alanı sunar. Bu uygulama alanlarında, “çoklu-robot çoklu-görev atama problemi” en temel problemlerden biridir. Her ne kadar aynı temel problemi içerseler de, uygulamaların eniyileme amaçları farklı olabilir. Örneğin, arama kurtarma çalışmaları için yürütme zamanını kısaltmak önemli iken, uzay araştırmaları esnasında sınırlı yakıt mevcudu nedeniyle robotların toplam yolu kısaltılmaya çalışılabilir.

Çoklu-hedef atama probleminin tek bir robot için tanımı, yöneylem araştırmalarında iyi bilinen TSP (Traveling Salesman Problem) probleminin (Lawler vd., 1985) açık çevrim versiyonu olup, polinomsal karmaşıklıkta yöntemlerle tüm veri kümeleri için en iyi sonucu oluşturabilecek şekilde çözülemez. Problemin çoklu-robot versiyonunda, robot yol planlamasının yanında, hangi hedefin hangi robota atanacağı problemi de göz önüne alınacağından problemin karmaşıklığı artar. En iyi sonuçlar IP (Integer Programming) formülasyonları ile bulunabilir. Ancak, bu yöntemler, hedef sayısının artması, hedef atamasında kullanılan maliyet değerlerinin belirsizliklerden dolayı sıkça değişim göstermesi, robot bozulmaları nedeniyle hatalar oluşması veya sisteme yeni görevler eklenmesi durumlarında yeniden atamaların yapılması söz konusu olduğundan, sıkça bilgi işlemsel açıdan yüklü hesaplamalar yapmayı gerektirebilir. Bunların ötesinde, robotların yürütme zamanında hedefleri ziyaret etmek üzere hem yol planlaması hem de ortam haritalaması için hesaplamalar yapmaları gerekir. Bu ek hesaplamalar sistem yükünü daha da artıracaktır. Dolayısıyla, ilk atamaların yapılması için yapılan maliyetli işlemsel hesaplamaların, yürütme zamanının sonraki aşamasında belirtilen nedenlerle tümüyle gözardı edilip, yeniden yapılması gerekebilir. Özellikle gerçek dünya üzerinde çalışan robot sistemlerinde bu çok sık rastlanan bir durumdur.

Bu çalışmada, tüm bu problemlerin çözümü olarak genel bir çoklu-robot sistem mimarisi olan

DEMiR-CF (Distributed and Efficient Multi-Robot-Cooperation Framework) önerilmekte (Sariel vd., 2006, Sariel ve Balch, 2006a) ve gerçek zamanlı çoklu atama problemine uygulaması gerçekleştirilerek, başarımlı analizi verilmektedir. Genel olarak, DEMiR-CF mimarisi heterojen bir robot takımının farklı kaynak gereksinimleri içeren karmaşık görevleri, dağıtılmış bir yönetim anlayışı içinde başarıyla tamamlamasını sağlayacak şekilde tasarlanmıştır. Farklı iyileme amaç fonksiyonları ile tamamlanmak istenen görev, robotların ortaklaşa ve koordineli şekilde çalışması ile hataya dayanıklı şekilde yürütülür.

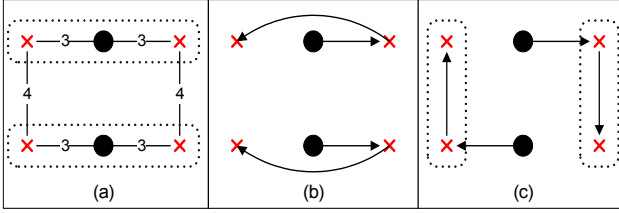
Çoklu-hedef atama problemi için maliyet fonksiyonları tasarlanmış ve DEMiR-CF mimarisi ile birleştirilerek başarılı sonuçlara ulaşılmıştır. DEMiR-CF gereksiz hesaplama yükünü artımlı atama yaklaşımı ile indirir. Artımlı atamalar sistemin güncel durumu göz önüne alınarak yapılır. Artımlı ve dinamik bir yapının kurulmasının yanında, sistemin bütününe başarımlı da göz önüne alınır. Robotlar kendi hedeflerini seçerken, ileride ziyaret etmeleri gereken hedefleri de göz önüne alarak planlama yaparlar fakat bu hedefleri belirlerken de etkin yöntemler kullanırlar. Bu hedefler daha sonra başka robotlar tarafından da ziyaret edilebilir. Önerilen yöntemde, robotlar, hedef ataması ve ziyareti esnasında oluşabilecek çeşitli hatalara karşı sistemi dayanıklı kılacak şekilde çalışmalarını sürdürürler, aynı zamanda hem ortaya çıkabilecek yeni fırsatları değerlendirir, hem de maliyet artımı durumunda daha iyi çözümler üretmeye çalışırlar.

Sonuç olarak bu çalışmada, çoklu-robot çoklu-hedef atama konusunun gerçek ortam yürütme problemleri ele alınmış, etkin maliyet fonksiyonları önerilmiş ve genel bir mimari ile bu problemin etkin ve polinomsal bir karmaşıklıkla çözülmesi sağlanmıştır. Deneysel sonuçlardan da görüleceği gibi, görev ataması ve robot yol planlamasının entegrasyonunu öneren yöntem oldukça iyi sonuçlar vermiştir.

Çoklu-robot çoklu-hedef problemi

Çoklu-robot çoklu-hedef atama problemi, çok sayıda hedefin en az bir robot tarafından etkin şekilde ziyaret edilmesi problemidir. Bu problem

eniyeleme için farklı amaç fonksiyonları ile ifade edilebilir. Robotların göreve başlangıç noktaları hakkında bir kısıtlama yoktur. Eniyilemede robotların hedef ziyaretlerini tamamladıktan sonra başlangıç noktasına geri dönüşleri göz önüne alınmaz. Problemin çözüm kalitesinde, hem hedeflerin robotlara ne şekilde atandığı hem de robotların hangi yolla kendine atanan hedefleri ziyaret ettiğinin yüksek ölçüde önemi vardır.



Şekil 1. Robot-hedef çiftleri uzaklıkları ile yapılan sınıflandırma

Sadece çiftler arası uzaklık değerlendirildiğinde, robot yolu oluşturulurken katedilmesi gereken yollar dikkate alınmaz (Şekil 1). Şekil 1’de, robot konumları nokta, hedefler çarpı işareti ile gösterilmiş ve uzaklıklar belirtilmiştir. Sadece uzaklıkları değerlendirerek yapılan atama sonucu oluşan çözüm, 18 birim maliyetli olup, en iyi çözüm 14 birim maliyetlidir. Demetleme yöntemleri kullanılarak da hedef atamaları yapılabilir. Fakat bu yöntemde de ek dolaşım maliyetleri gözardı edilmektedir (Şekil 2).

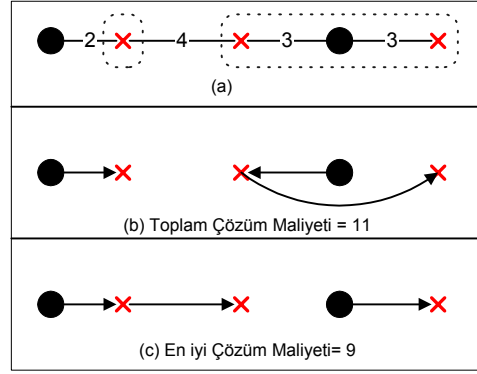
Lagoudakis ve diğerleri (2005) bu problemi hedef atama açısından hem ardışıl hedef atama ağaçlarının ve robot yollarının oluşturulması, hem de yolların doğrudan oluşturulması yönünde incelemiştir.

Problemin çözüm yöntemleri

Çözüm yöntemleri yöneylem araştırmalarındaki yöntemler ve robotlar üzerinde gerçekleştirilen yöntemler başlığı altında iki bölümde incelenecektir.

Yöneylem araştırmaları yöntemleri

En iyi çözümler, etkin bir Integer Programming (IP) formülasyonunun CPLEX gibi IP çözücü programlarla çözülmesi sonucu oluşturulabilir.



Şekil 2. Demetleme ile elde edilen çözüm

Bir başka çözüm yöntemi de problemi Branch-and-Bound (BB) algoritmaları ile çözmektir. Bu algortmada çözümleri içeren bir arama ağacı daha küçük ağaçlara bölünerek arama sürdürülür. Anlık incelenen ağaçta en küçük ve en büyük çözümler değerlendirilerek arama yönlendirilir. Aramaya tüm düğümler silinip tek bir çözüme ulaşılan dek devam edilir (Toth ve Vigo, 2001).

En iyi sonuçları üreten IP ve BB yöntemleri problemi çözmek için kullanılabilir fakat çok büyük olmayan veri kümeleri için bile çözüm süresi için bir garanti verilememektedir. Belirsizliklerden dolayı, hedefler arasındaki tahmini uzaklıkların değişmesi durumuyla sıkça karşılaşılabilir. Ortamda oluşabilecek ufak değişiklikler bile çözümü tümüyle değiştirebilir. Bu durumda problem yeniden çözülmelidir. Bu durumlara karşı hızlı çözümler üretmenin önemi özellikle gerçek zamanlı çalışan sistemler için artmaktadır.

Amaçlanan hızı sağlayabilmek üzere, çözüm kalitesinden belli bir oranda fedakarlık ederek yine de iyi sonuçların üretildiği sezgisel yöntemler uygulanabilir. Bu tür yöntemlerde, önce standart hedef ataması ve sonra atamalar üzerinde iyileştirme çalışmaları yürütülebilir. Arama, uzayın sadece belirli bir bölgesinde gerçekleştirilmektedir. Bir başka sezgisel yöntem olan yaklaşımsal (Metaheuristic) yöntemde, arama uzayı daha da geniş tutulabilir. Bu yöntemlere örnek olarak Evrimsel Algoritmalar, Tabu Araması (Tabu Search) ve Benzetimli Tavlama (Simulated Annealing) yöntemleri sayılabilir.

Yaklaşımsal yöntemlerde iyi sonuçlar alınabileceği gibi zaman karmaşıklığı da bağıl olarak artmaktadır. Ayrıca uygulanan işlemler genellikle bağlama dayalı olup, çoğunlukla iyi sonuçlar, ancak iyi belirlenmiş parametrelerle alınabilir (Toth ve Vigo, 2001).

Robotlar için tasarlanan yöntemler

GRAMMPS (Brumitt ve Stentz, 1998) bu problemi ele alan ve robotlar üzerinde gerçekleştirilen ilk çalışmalardan biridir. Bu çalışmada, atamalar merkezi olarak yapılır. İlk atamalar rastgele olarak yapılır, zaman içinde bu atamalar Benzetimli Tavlama algoritması ile iyileştirilir.

Prim Allocation yöntemi (Lagoudakis vd., 2004), Prim Algoritması üzerine kurulmuş bir çoklu-robot çoklu-hedef atama yöntemidir. Robotlar için ayrı ayrı Minimum Kapsayan Ağaç (Minimum Spanning Tree) ve tümünde Minimum Kapsayan Orman (Minimum Spanning Forest) oluşturulur. Hedef ağaçları, algoritmanın adım adım yürütülmesi ve her bir adımda atanmamış bir hedefin kendine en yakın yola sahip robota atanması ile gerçekleşir. İşlem, tüm hedefler atanana dek devam eder. Dolayısıyla, her bir yeni hedef atanması, ilgili hedefin robotların o ana dek oluşturdukları hedef ağaçlarındaki hedeflerine uzaklıkları dikkate alınarak yapılır. Robotun son konumu değerlendirilmez. Algoritmanın dağıtılmış versiyonunda, her bir robot kendisi için en uygun hedefle ilgili bir açık artırma yapar ve en düşük istekte bulunulan hedef ilgili robot ağacına eklenir. Hedef atamalarının tümü robotların görevi yürütmesinden önce gerçekleşir. Ortam durumunun değişmesi halinde, kalan tüm hedeflerin atanması yeniden aynı yöntemle yapılır. Prim algoritması gibi Prim Allocation yöntemi çözümü de en iyi çözümün en fazla 2 katı ile sınırlıdır. Bu algoritmanın tüm gerçekleştirme ayrıntıları, ilgili makalede verilmiş olduğundan, deneyler için bu çalışma kapsamında gerçekleştirilmiş çözüm ile karşılaştırmalar yapılmıştır. Prim Allocation algoritmasının yer aldığı makalede robot benzetim ortamı üzerinde elde edilen sonuçlar bulunmakta, ancak algoritmanın gerçek robotlar üzerinde gerçekleştirilmiş bir versiyonu yer almamaktadır.

Literatürde bir anda çok-hedefli atamaların yapıldığı yöntemler de bulunmaktadır. Bu yöntemler belirli hedef kümelerinin robotlara atanması ile ilgilenir ve robotlar arasında kombinezonsal olarak bu hedef kümelerinin değişimini gerçekleştirirler. Bu yöntemi uygulayan robot sistemlerinin başında Dias ve Stentz (2002) ve Berhault ve diğerleri (2003) çalışmaları gelmektedir. Dias'ın çalışmasında ekonomi modeli ele alınıp robot grupları ve liderler ile kombinezonsal olarak hedef alış-verişi ile sonuca ulaşılr. Fakat bu yöntem, kombinezonsal olarak hedef kümeleri üzerinde çalıştığından büyük hedef kümeleri için veya dinamik ortamlarda sık değişen maliyet değerlerine karşılık karmaşık bir sistem önermektedir ve bilgi işlemsel gereksinimleri oldukça büyük oranda artmaktadır.

Gerçek zamanlı yürütme sırasında oluşabilecek durumlar

Robot sistemleri için önceden tanımlanmış plan ve stratejiler bulunsa da, fiziksel ortamda görev yürütme önceden planlanmamış durumları da kotarmayı gerektirir. Gerçek dünya, robotların kontrolü dışında kalıp, sürekli değiştiğinden çoklu-robot görev yürütme problemi görev atama probleminin de ötesine geçer. Özellikle sensör gürültüleri, beklenmeyen davranışlar ve etkileri, iletişimden kaynaklanan ortam kısıtlamaları ve donanım bozulmaları çoklu-robot görev yürütme problemini zorlaştırıcı unsurlardır. Tüm bu faktörler çözüm üzerinde etkilidir. Yürütme esnasında oluşabilecek durumlar aşağıda özetlenmiştir:

- Robotun bozulduğunu farketmesi
- Robotun bir diğer robotun bozulduğunu farketmesi
- Görev yürütme zamanının tahmin edilenden daha uzun/kısa sürmesi
- Görev tanımlarında olabilecek değişiklikler
- Yeni görevlerin eklenmesi
- Yeni robotların sisteme eklenmesi
- Operatörlerin görev dağılımında değişiklik yapması

Bu durumlar robotun kendi belirlemeleri veya dış etkenler sonucunda oluşabilir. Bu plan dış durumlara karşın, en iyi çözümün bile kalitesi

deđişebilir. Çözüm kalitesini bu durumlara karşın ölçmek robot sistemleri için zor olmaktadır. Şimdiye dek yapılan çalışmalarda, çözüm kalitesi ile ilgili sınırlar, gerçek zaman durumlar gözardı edilerek ve bilginin tümüyle erişilebilir olduğu kabul edilerek verilmiştir (Lagoudakis vd., 2004).

Önerilen yöntem

Yürütme sırasında oluşabilecek durumlar nedeniyle, en iyi çözümleri elde etmek her zaman gerekli olmayabilir (Reinelt, 1994). Problemin tam olarak doğru şekilde modellenmemesi veya en iyi çözümü bulmak için gereken sürenin gerçekleştirilmede sağlanamaması, bu durumları destekler ve çoğunlukla da robot sistemlerinde bu durumlar olasıdır. Bu çalışmada, tüm gerçek zaman kısıtlamalarına karşın iyi sonuçlar üretebilen bir görev seçim ve atama yöntemi önerilmektedir. Tasarlanan ve gerçekleştirilen altyapı, robotların bir görevi yerine getirmek üzere görevin alt parçalarını üstlenip, koordinasyon içinde çalışmalarına olanak sağlayan dinamik ve artımlı bir görev yürütme mekanizması içermektedir (Sariel ve Balch, 2006b). Sistem, robotların, merkezi bir otorite olmadan, dağıtılmış bir şekilde kendileri için uygun görevleri seçmelerine izin vermektedir. Çalışmanın özgün katkılarında biri olan geçici görev listesi oluşturma ve artımlı görev ataması yaklaşımı, dinamik ortamlarda görev atama ve yol planlama adımlarının, yürütme anında ve işlemsel yükü az olacak şekilde, bir arada gerçekleştirilmesine olanak sağlamaktadır. Böylelikle, olađan dışı durumlara karşın sistemi dayanıklı kılan davranış biçimleri üretilerek, çözüm kalitesinin yürütme ile eşzamanlı olarak korunması amacına ulaşılır.

Dinamik görev seçimi yordamı (DGSY)

Her bir robot (r_j), problemi bütün olarak ele alabilmek için ileride yürütebileceđi hedefleri içeren geçici bir görev listesi (T_{R_j}) oluşturur. Bir sonraki adımda da, robotlar listelerinden kendileri için ilk olarak yürütülmesi en uygun olan görevi (t_S) seçerler. Geçici görev listeleri, Denklem 1'e göre ziyaret edilmemiş hedefler (T_U) arasından seçimle oluşturulur ve robota en yakın olan ziyaret edilmemiş hedefleri içerir. Bu denk-

lemdeki $dist$ fonksiyonu iki nokta arasındaki Euclidean uzaklığını hesaplar.

$$\begin{aligned} reldist(r_j, t_i) &= dist(r_j, t_i) \\ &\quad - \min(dist(r_k, t_i)), \{\forall k \neq j\} \end{aligned} \quad (1)$$

$$T_{R_j} = \cup_{t_i, reldist(r_j, t_i) < 0, \forall t_i \in T_U}$$

Dolayısıyla, en uygun hedef seçilmeden önce her bir robot, geçici görev listesini oluşturur; ancak bu, listede yer alan görevlerin ilgili robot tarafından mutlaka yürütüleceđi anlamına gelmez. Aksine, geçici görev listeleri zaman içinde deđişebilir. Bu listelerin oluşturulmasındaki amaç, görev ataması öncesinde probleme global olarak bakabilmek ve daha iyi atama planı oluşturabilmektir. Geçici görev listesi Algoritma 1 uygulanarak oluşturulur. Robot, daha sonra kendine ait görev listesi içinden en uygun görevi (t_S) yürütmek üzere seçer.

Algoritma-1. Geçici görev listesi oluşturma, r_j

algoritma ismi: MTRP-FormRoughSchedule

giriş parametreleri: T_U

çıkış parametreleri: T_{R_j} ve t_S

$T_{R_j} = \phi$ (maliyet fonksiyonuna göre oluşan heap)

$t_S = \phi$

while ($T_U \neq \phi$)

if t_i geçici görev listesi (Denklem (1)) içindeyse

c_{ji} maliyetini t_i için hesapla

t_i 'yi T_{R_j} 'ye yerleştir

if $\|T_{R_j}\| > 0$

$t_S = top(T_{R_j})$

Algoritma 2, artımlı görev seçimini gerçekleyen temel döngüyü gösterir. Bu algoritma, görev yürütme başında ve robotun ortam bilgisi her deđiştiğinde çağrılır. Her bir robot, ana görevin ziyaret edilmemiş bütün hedefleri ziyaret edilene kadar dağıtılmış bir şekilde aynı algoritmayı gerçekleştirir ve ilgili görevleri yürütür. Önerilen algoritma tüm hedefleri tümüyle yeniden atamak için de kullanılabilir fakat bu çalışma, dinamik ortamlar için hedeflerin tümüyle deđil, artımlı olarak atanmasını öngörmektedir. Bu sayede, yinelenen gereksiz atamalar ve bu adımların neden olacağı fazladan işlem yükü önlen-

miş olur. Hedef seçimi için kullanılacak maliyet fonksiyonu karmaşıklığı, üzerinde gerçekleştirilecek robotların kapasitelerine göre belirlenebilir.

Algoritma-2. Artımlı görev seçimi döngüsü, r_j

algoritma ismi: MTRP-DPTSS
giriş parametreleri: T_U
çıkış parametreleri: Robot davranışı
 $[T_{R_j}, t_S] = \text{MTRP-FormRoughSchedule}(T_U)$
if $t_S \neq \emptyset$
 if $t_S =$ yürütülmekte olan görev
 görevi yürütmeye devam et
 else
 if t_S yürütülüyor ise
 yürütme isteği belirt
 else görev değişimi yap
else boşta bekle

Maliyet fonksiyonu

TSP için önerilmiş olan maliyet fonksiyonları çoğunlukla üçgen eşitsizliği özelliğini kullanır. Bu çalışmada önerilen fonksiyonlar da aynı prensibe dayanır. En Yakın Hedef (EYH) sezgisel maliyet hesabı basit şekilde, robot ve hedef arasındaki uzaklığı değerlendirir (Denklem (2)).

$$c_{ji} = \text{dist}(r_j, t_i), t_i \in T_{R_j} \quad (2)$$

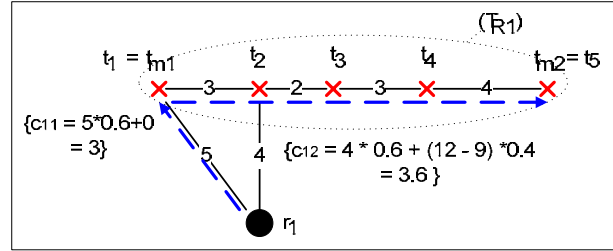
Uzak Hedef Ekleme Maliyeti (UEM) bir başka maliyet hesap fonksiyonu olup (Denklem 3), sınırda olan hedeflerin seçilmemesi durumu için ceza uygulayarak geçici hedef listesi içinde uygun bir rota çizmeye yarar (Sariel ve Balch, 2005).

$$c_{ji} = \alpha * \text{dist}(r_j, t_i) + (1 - \alpha) * \left[\begin{array}{l} \text{dist}(t_{b1}, t_{b2}) \\ - \max(\text{dist}(t_i, t_{b1}), \text{dist}(t_i, t_{b2})) \end{array} \right] \quad (3)$$

$\{\text{dist}(t_{b1}, t_{b2}) = \max(\text{dist}(t_k, t_l)), t_{i,k,l} \in T_{R_j}\}$

t_{b1} ve t_{b2} , T_{R_j} içinde birbirine en uzak olan iki hedef çiftidir. Fonksiyon, robotları bir dereceye kadar bu hedefleri öncelikle ziyaret etmeye iter çünkü bu hedefler ziyaret edilmesi gereken birbirlerine en uzak hedef çiftidir. Söz konusu olan en uzun mesafe, yoldaki diğer hedefler de ziya-

ret edilerek azaltılabilmektedir. Örnek bir robot yolu Şekil 3’de verilmektedir. Şekilde t_2 , r_1 ’e daha yakın olduğu halde, yapılan maliyet hesabı ile, önce t_1 ’i ziyaret etmenin maliyetinin daha düşük olduğu belirlenmiştir; robot önce t_1 ’e gider, daha sonra da yol üzerindeki diğer hedefleri tek tek ziyaret eder. Yapılan deneysel ölçümler sonrasında α katsayısı için seçilen 0.6 değeri ile en iyi ölçümler elde edilmiştir.



Şekil 3. UEM ile örnek bir maliyet hesabı

Algoritma-3. Temel adımlar (Asenkron), r_j

if istek için son cevap yollama süresi doldu
 isteği sonlandır
 en uygun robotu seç ve bildir ya da yürütmeye başla
if t_k için r_j ’den istek mesajı geldiyse
 ilgili amaç fonksiyonuna göre
 MTRP-Response yürüt
if istek seçim mesajı geldiyse ve $\|T_{R_j}\| = 0$
 görevi yürütmeye başla
if ortam durumu T_{R_j} ’yi değiştirecek şekilde değişti ise
 yürütme veya istekleri iptal et

Algoritma-3’. Toplam yolu kısaltma amacıyla t_k için r_j ’den gelen isteğe yanıt, r_j

algoritma ismi: MTRP-Response - Path
if yürütülmekte olan görev veya istek yayını varsa ve $c_{ji} > c_{lk}$
 yürütme veya istek yayını iptal et
if $c_{ji} < c_{lk}$
 t_k için kendi maliyet bilgisini yolla

Algoritma-3’. Toplam zamanı kısaltma amacıyla t_k için r_j ’den gelen isteğe yanıt, r_j

algoritma ismi: MTRP-Response - Time
if yürütülmekte olan görev veya istek yayını varsa ve $c_{ji} > c_{jk} \parallel c_{ji} > c_{lk}$ ve t_k veya r_l T_{R_j} ’ye yakın ise
 yürütme veya istek yayını iptal et
if $c_{ji} < c_{lk}$
 t_k için kendi maliyet bilgisini yolla

Dağıtılmış hedef ataması

Çalışmada, robotlar arasında dağıtılmış görev ataması için en uygun yöntem olarak CNP (Contract Net Protocol) protokolü seçilmiştir. Protokolün önerilen sisteme uyarlanmış hali, robotların önce görev yürütme isteklerini yayınlaması, daha sonra da yayının hedefi olan görevin yürütülmesi için en uygun robotun seçilmesi adımlarını içermektedir. Önerilen yöntemde, farklı görevler için eş anlı olarak görev yürütme istekleri yayınlama ve robot seçimi mümkün olmaktadır. Eğer eniyileme fonksiyonu, toplam yolun kısaltılması ise, bir anda tek bir isteği sonlandırmak daha iyi sonuç vermektedir. Robotlar istek mesajları aldıklarında göreve ilişkin kendi maliyet değerlerini diğer robotlara yollarlar. Sistem bütünlüğünü bozan durumlar için sonraki bölümde ayrıntıları verilecek olan bir uyarı mesajı da yollayabilirler. Farklı durumlara karşılık asenkron olarak uygulanan adımlar Algoritma-3 ve içinde kullandığı altıyordamlar Algoritma-3' ve Algoritma-3'' ' te verilmiştir.

Dağıtılmış hata tanıma ve kotarma

Sistemde yer alan hata tanıma ve kotarma mekanizması, yürütme zamanında oluşabilecek hataların tanınması ve bu hatalara karşı sistem bütünlüğü ve etkinliğinin korunmasını sağlar. Hata kotarma mekanizması, sistemin daha önce tanıtılan görev atama bileşenlerine entegre şekilde tasarlanmış olup, sistemin etkin ve dayanıklı çalışır halde tek bir bütün olmasını sağlar. Hataya dayanıklılık iki bileşen ile sağlanır: *Model Güncelleme Modülü* ve *Sistem Bütünlüğü Kontrol Modülü*. *Model Güncelleme Modülü*, robotun kendi ortam/dünya modelini, iletişim yoluyla veya sensörlerine gelen bilgilere dayanarak, kendi karar mekanizması ile güncellemesini sağlar. *Sistem Bütünlüğü Kontrol Modülü* ise sistem tutarlılığını sağlamak üzere, robotun modeline göre uyarı mesajları yollamakla yükümlüdür. Olağandışı durumlara karşılık modelleri güncellemek veya hata kotarma için uygun tutarlılık kontrol ve güncelleme adımları, dağıtılmış bir şekilde uygulanır. Her bir robot, sistemdeki tüm görevler ve diğer robotlara ilişkin dünya modelini oluşturur ve gerektiğinde günceller. Bazı durumlarda robotların modelleri

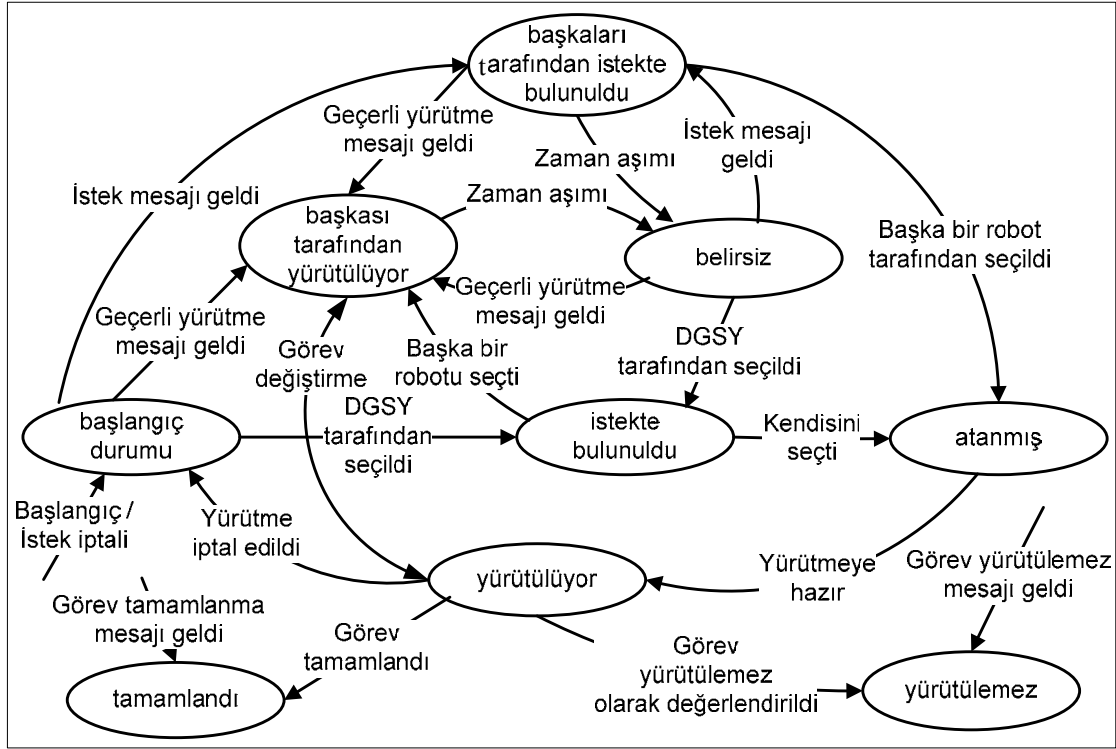
farklılaşabilir. Bunun nedeni iletişim problemlerinden kaynaklanan belirsizlikler, tümüyle tanımlanmamış bilgiler, varsayımlar, vb. olabilir. Dağıtılmış sistemlerde veya gerçek dünya sistemlerinde tüm bilginin her zaman tümüyle doğru bir şekilde tanımlı olması beklenemez.

Görev modelleri, her bir görev için farklı sonlu durumlu makineler (Şekil 4) halinde tutularak gerçekleştirilir. Görev durum geçişleri, *Model Güncelleme Modülü* tarafından etkinleştirilir. Görevler, şu durumlarla temsil edilebilir: *başlangıç durumu, belirsiz, istekte bulunuldu, başka robotlar tarafından istekte bulunuldu, atanmış, yürütülüyor, başkaları tarafından yürütülüyor, tamamlandı ve yürütülemez*. Robot modelleri de sonlu durumlu makinelerde *boşta, yürütme aşamasında, bozuldu, istekte bulunuyor* durumlarından birinde olacak şekilde tutulur.

Hata kotarma işlemleri, robotun kendi modelini güncellemesi ya da diğer robotları uyarması şeklinde olur. Robotların tamamlanmış, yürütülmekte olan veya istekte bulunulan görevlerden haberdar olmamaları durumunda tutarsızlıklar oluşur. Robotlar, sistem tutarlılığını korumak üzere birbirleri ile haberleşirler ve şu bilgileri yollarlar:

- yeni keşfedilen hedefler
- belli aralıklarla güncellenmiş maliyet bilgisi ile görev yürütme bilgisi
- görevi tamamladıklarında başarı bilgisi
- görev yürütme iptali bilgisi
- görev yürütülemez bilgisi

Sistemde bilginin mükemmel olmadığı göz önüne alınır ve robotlar görev atamasını ve hata kotarma mekanizmalarını dağıtılmış şekilde gerçekleştirirler. Bazı durumlarda robotlar, doğru olmayan yargılara ulaşabilirler. Örneğin, aslında bozulmamış olan bir robotun bozulmuş olarak modellenmesi gibi. Bu durumlarda paralel yürütme durumları (aynı görevin birden fazla robota atanması) oluşabilir. Görevin hiç yürütülmesine karşı tercih edilebilecek bir durum olduğundan, buna izin verilmektedir. Fakat yine de bu tür durumlar, uygun hata kotarma mekanizmaları ile düzeltilir.



Şekil 4. Görev modelleri sonlu durum makinesi

Deneyel sonuçlar

Deneyler, üç küme halinde gerçekleştirilmiştir. Birinci küme, önerilen yöntem için maliyet fonksiyonlarının sınanmasını hedefler. Elde edilen sonuçlar, Prim Allocation ve CPLEX ile gerçekleştirilen IP yaklaşımı ile karşılaştırılmıştır. Dinamik yürütme zamanı deneyleri Webots gerçekçi benzetim ortamı üzerinde yürütülmüştür. Üçüncü kümede yer alan deneyler ise Khepera II robotları üzerinde gerçekleştirilmiştir.

Deney-1: DEMİR-CF ve maliyet fonksiyonlarının analizi

Bu deneylerde algoritmalar, rasgele üretilen farklı sayıda robot ve hedef kümeleri üzerinde çalıştırılarak başarımları analiz edilmiştir. Deneyler 100x100 boyutlu ızgara yapısında bir ortam üzerinde, robotların sayısı 1-50 arasında, hedeflerin sayısı da 10-50 arasında değiştirilerek gerçekleştirilmiştir. 100 farklı çalıştırmanın her birinde robotlar rastgele belirlenmiş farklı konumlarda olacak şekilde yerleştirilmiştir.

UEM başarımları analizi TSPLIB’den alınan, bilinen TSP veri kümeleri için gerçekleştirilmiştir. Sonuçlar

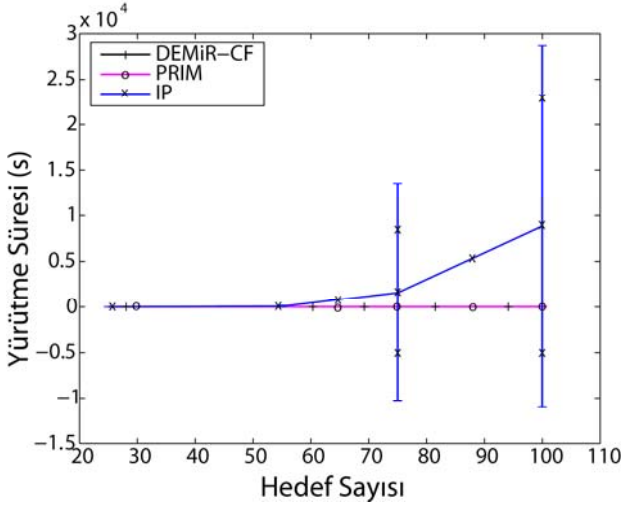
ve en iyi değerden sapmalar Tablo 1’de verilmiştir. Bu sonuçlara göre, en iyiden sapma 101 hedef içeren geniş ölçekli bir problem için en fazla %15.24 oranında olmuştur.

Tablo 1. TSP Veri setleri için UEM başarımları analizi

TSP Veri Seti	UEM	En iyi Sonuç	Sapma
ATT48	33537.83	31470.4	6.5
EIL51	444.01	413.51	7.37
BERLIN52	8104.99	7305.38	10.94
EIL101	725.31	629.38	15.24

Algoritmaların yürütme zamanlarına ilişkin karşılaştırmalar Şekil 5’de görülebilir. Şekilde yer alan grafikte, tek bir robot için atama yapmak üzere seçilen yöntemlerin çözüm süreleri karşılaştırılmaktadır. IP yönteminin yüksek orandaki standart sapması, bu yöntemin çözüm süresinin veri kümesinin yapısına bağımlılığını göstermektedir. Hatta bazı veri kümeleri için çözüm süresi, tek bir makine üzerinde günleri aşmaktadır. Bu sonuç, bu yöntemin pratik olarak uygu-

lanabilirliğini kısıtlamaktadır. Sezgisel yöntemlerle yapılan atamalar, şekilde de görüldüğü gibi oldukça kısa sürede gerçekleşmektedir.



Şekil 5. Yürütme zamanı karşılaştırması

Tüm başarımların analizi sonuçları Şekil 6’ da verilmektedir. PRIM Allocation yöntemi ile karşılaştırmalar yapılırken, ilgili makalede değinilmeyen ama çözüm kalitesini artırıcı etkisi olabileceği düşünülen bir iyileştirme aşağıdaki şekilde belirlenmiş ve bu iyileştirme de eklenerek karşılaştırmalar yapılmıştır. Bu iyileştirme eklemeleri, problemin açık çevrim için tanımlı olmasından uygun görülmüştür. İyileştirme, ağaç taraması esnasında ilk olarak en kısa ağacın seçilmesi ile gerçekleşmiştir. Deneylede PRIM-ORG sonuçları Prim Allocation algoritmasının iyileştirme olmadan orijinal versiyonun gerçekleşmesi ile oluşan sonuçlarını, PRIM-SD sonuçları ise bahsedilen iyileştirmenin gerçekleşmesi ile oluşan sonuçları göstermektedir. Çalışmanın ürünü olan DEMİR-CF yöntemi UEM ve EYH maliyet hesapları ile gerçekleştirilerek elde edilen sonuçlar ayrı grafiklerde sunulmuştur. Sonuçlar incelendiğinde, artan robot sayısına karşılık çözüm kalitesinin de görev atamasına bağlı olarak değiştiği belirlenmiştir. DEMİR-CF yönteminin, her iki maliyet fonksiyonu ile, en iyi sonuçlara yakın sonuçlar ürettiği gözlenmiştir. Hedef/robot yoğunluğundaki azalma, yöntemlerin en iyi sonuca yakınlaşmasında etkili olmaktadır. Fakat, şekilden de görülebileceği gibi, Prim Allocation yöntemi sonuçları bu durumda bile en iyi sonuçtan sapma göstermektedir. Bu sonuca, hedef atamasının robotların son konumları göz önüne alınmadan

gerçeklenmesi yol açmaktadır. Önerilen DEMİR-CF yönteminde görev listelerinin (T_{Rj}) dinamik olarak oluşturulması ile bu durum engellenmektedir.

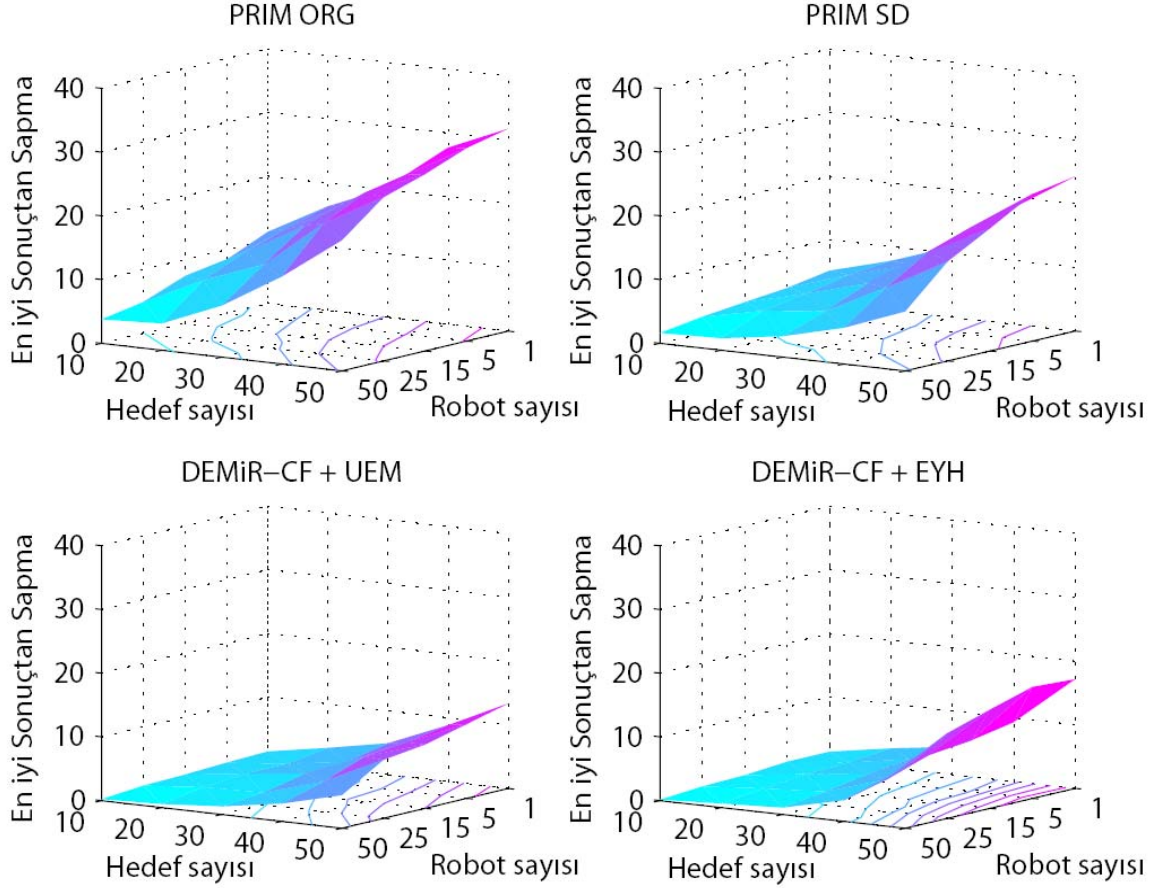
Deney-2: Gerçek zamanlı dinamik benzetim ortamı deneyleri

Benzetim ortamı (Webots) deneylerinde 25m²’lik 3B alan içinde robotların 70mm çapında ve nesnelerin hedef olarak modellendiği ortamlar oluşturulmuştur. Ortamlar, hedeflerin ve robot konumlarının rasgele belirlendiği VRML (Virtual Reality Modeling Language) dosyaları yaratılarak oluşturulmaktadır. Robot kinematik hesapları kodlayıcılardan gelen bilgilerle konum belirleme hesabı yapılarak gerçekleştirilmektedir. Robotlar, üst seviyede görev atamasının yanında, hem otonom olarak ortam haritasını oluşturmakta, hem de ortamda kendi konumlarını belirlemektedirler. Benzetim ortamında kayma ve kodlayıcı hataları modellenmiştir. Beklendiği gibi, diferansiyel sürürlü robotlarda kayma nedeniyle konum belirleme hataları oluşmaktadır. Bu durum da gerçek dünya probleminin bir parçasıdır.

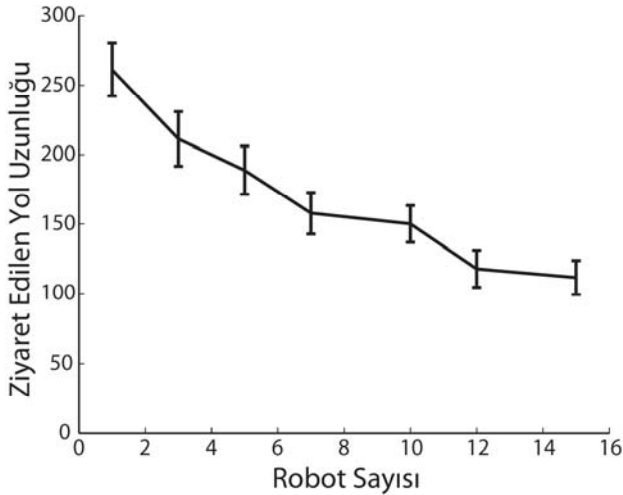
İlk deneylerde 10 hedef için farklı robot sayılarına karşılık ölçeklenebilirlik testi yapılmıştır. Şekil 7’ de görüldüğü gibi çoklu-robot sisteminden beklenen iyileştirme sağlanarak robotların toplam yolları kısalmıştır.

İkinci deneyde, hata kotarma mekanizmasının başarımların analiz edilmiştir. Şekil 8’ de verilen grafiklerde “Hata kotarmasız” etiketli sonuçlar hata kotarma mekanizması etkin olmadığı durum için oluşturulmuştur. “Hata kotarmalı” etiketli sonuçlar hata kotarma mekanizması eklenmesi durumunda oluşan sonuçları gösterir.

Hata kotarma mekanizmasının sisteme ek mesaj yükü getireceği düşünülebilirse de, deneysel sonuçlarda sistemde toplam yollanan mesaj sayısı analiz edilecek olursa, bu yöntemin ek mesajlaşma trafiğini azalttığı görülmektedir. Ayrıca, ikinci şekilde de görüldüğü gibi, hata kotarma mekanizması gereksiz ziyaret yollarının azalması da etkili olmuştur. Sonuçlar, hata kotarma mekanizmasının sistemin hataya karşı dayanıklılığını arttırmaya ek olarak, yolun kısaltılmasında ve mesaj trafiğinin azaltılmasına da katkıda bulunduğunu göstermektedir.



Şekil 6. Tüm Algoritmaların farklı sayıda hedef ve robot kümeleri için başarımların analizi



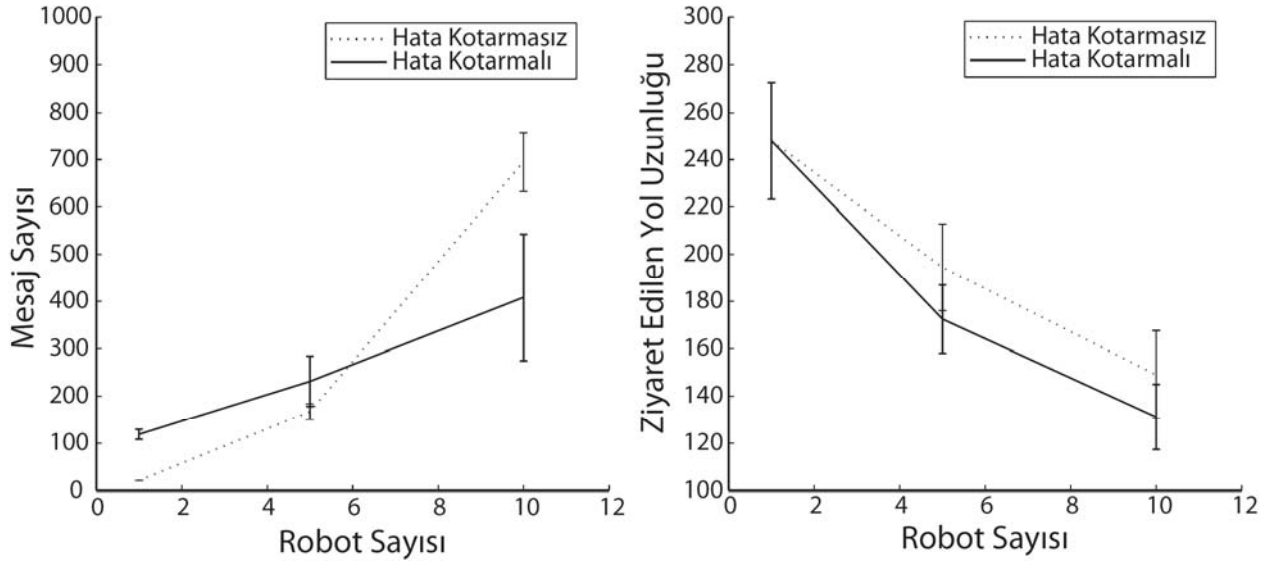
Şekil 7. Farklı robot sayıları için ölçüleme başarımların analizi

Ortam haritası bilgisine göre atamaların farklılaşmasına ilişkin bir senaryo Şekil 9’ da görülmektedir. Bu senaryoda bulunan 3 robot, yerleri bilinen dokuz hedefi ziyaret etmekle görevlidir-

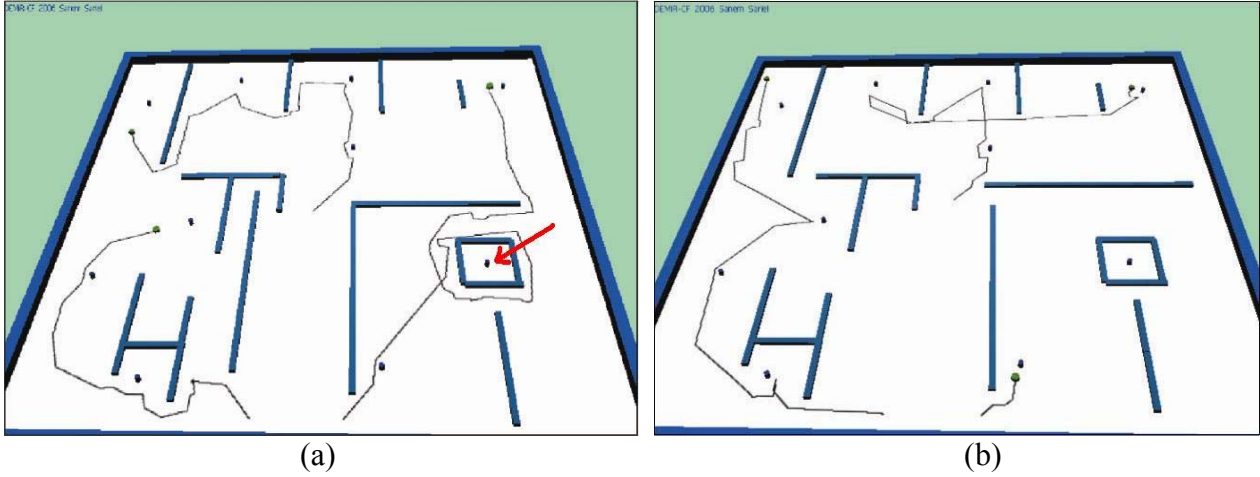
ler. Şekil 9 (a)’ da ortam haritası bilinmediğinden robotların yürütülemez olan görevleri (kırmızı ok ile işaretlenmiştir) ancak ortam haritası oluşturulduktan sonra belirlenmesi söz konusu olmuştur. Harita bilindiğinde, bu basit örnek için bile atamalar çok farklı olmaktadır (Şekil 9 (b)).

Deney-3: Gerçek-zamanlı gerçek ortam robot deneyleri

Gerçek ortam deneylerinde kullanılan KheperaII robotları 25MHz MC68331 mikro-kontrolörü, 512K Flash, 512K RAM ve 8 kızıl ötesi sensör ile donatılmışlardır. Robotlar arası iletişim, radyo hatları üzerinden kablo-suz olarak gerçekleşmektedir. Robotlar, DEMİR-CF mimarisi ile donatılmışlardır. Görev atama modülleri, deney için gerçekleştirilen Algı modülü, Motor Arayüzü, Hareket Modeli Modülü ve Haritalama modüllerine çok iplikli (thread) bir yapı ile entegre edilmiştir.



Şekil 8. Hata Kotarma mekanizmasının ek katkıları



Şekil 9. Ortam haritasının bilinip bilinmemesine göre atamalar çok farklı şekilde yapılabilir

Şekil 10'da robotların gerçekledikleri bir senaryoya ilişkin izledikleri yollar görülmektedir. Bu senaryoda ziyaret edilmesi gereken altı hedef bulunmaktadır. Yol gösterilimi farklı zaman noktalarına ait video çerçevelerinin birbirinin üstüne bindirilmesi ile elde edilmiştir. Şekil 10a' da bir robot tarafından hedeflerin ziyaret edilmesi ve robotun bu görevi yerine getirirken izlediği yol görülmektedir. Hedefler resim üzerinde mavi bayraklarla işaretlenmiştir. Aynı görev, bir üçlü robot takımı ile zamanı kısaltmak üzere yürütüldüğünde, robotlar farklı hedefleri ziyaret ederek görevi paylaşmaktadırlar (Şekil 10 (b)).

Diğer bir senaryoda, DEMiR-CF ile çoklu-robot sisteminin oluşabilecek hata durumlarına karşı dayanıklı bir sistem oluşturup görevi başarıyla sonlandırdığı görülmektedir (Şekil 10 (c)). Bozulma anındaki konumlar işaretlenmiştir. Bozulmadan sonra robotlar, tamamlanmamış görevleri de yürüterek tüm görevi tamamlarlar.

Sonuçlar

Bu çalışmada, gerçek zamanlı çoklu-robot çoklu hedef problemini çözmek üzere artımlı bir görev atama ve yürütme zamanlı hata kotarma yöntemi

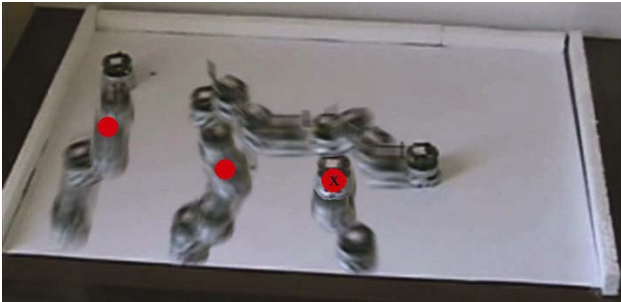
mi (DEMiR-CF) önerilmektedir. Önerilen yöntem, robot alt katman modülleriyle tüm bir sistem oluşturmak üzere başarıyla kullanılabilir. Robotlar, görev atama ve yol belirleme kararlarını dağıtılmış bir şekilde verirler. Artımlı olarak görev atama ve yol belirleme adımları birbiri ile iç içe geçmiş olarak problem çözülür. Problemin yapısını değiştiren yürütme zamanı durumları, görev atama ve yürütme aşamasında eş zamanlı olarak çözülür. Başarım ölçme deneyleri, üç küme halinde hem benzetim ortamlarında hem de gerçek robotlar üzerinde gerçekleştirilmiştir. Elde edilen deneysel sonuçlar, önerilen yöntemin ölçeklenebilir olduğunu göstermiş ve başarım değerlerinin literatürde yer alan diğer çalışmalara oranla daha yüksek olduğunu kanıtlamıştır. Ayrıca, yöntemin oldukça kısıtlı işlemsel kapasiteye sahip küçük robotlar üzerinde de gerçekleştirilebilir olduğu gösterilmiştir.



(a)



(b)



(c)

Şekil 10. Khepera II deneyleri

Kaynaklar

- Berhaut, M., Huang, H., Keskinocak, P., Elmaghraby, W., Griffin, P., ve Kleywegt, A., (2003). Robot exploration with combinatorial auctions, *Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1957-1962, Las Vegas, Nevada, USA.
- Brumitt, B.L. ve Stentz, A., (1998). Grammps: A generalized mission planner for multiple robots in unstructured environments, *Proceedings, IEEE International Conference on Robotics and Automation (ICRA)*, 1564-1571, Leuven, Belgium.
- Dias, M.B. ve Stentz, A., (2002). Opportunistic optimization for market-based multirobot control, *Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2714-2720, Switzerland.
- Lagoudakis, M. G., Keskinocak, P., Kleywegt, A., ve Koenig, S., (2004). Simple auctions with performance guarantees for multi-robot task allocation, *Proceedings, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 698-705, Sendai, Japan.
- Lagoudakis, M. G., Keskinocak, P., Kleywegt, A., ve Koenig, S., (2005). Auction-based multi-robot routing, *Proceedings, Robotics: Science and Systems (RSS)*, 343-350, Cambridge, Massachusetts, USA.
- Lawler, E. L., Lenstra, J. K., Kan, R., ve Shmoys, D. B., (1985). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. John Wiley & Sons, New York, NY, USA.
- Michel, O., (1998). Webots: Symbiosis between virtual and real mobile robots. *Proceedings of the First International Conference on Virtual Worlds*, 254-263, Paris, France.
- Reinelt, G., (1994). *The Traveling Salesman: Computational Solutions for TSP Applications*. Springer-Verlag, Heidelberg, Germany.
- Sariel S., ve Balch T., (2005). Real Time Auction Based Allocation of Tasks for Multi-Robot Exploration Problem in Dynamic Environments, *Proceedings, Integrating Planning into Scheduling AAAI Workshop*, 27-33, Pittsburgh, Pennsylvania, USA.
- Sariel S., ve Balch T., (2006a). *A Distributed Multi-Robot Cooperation Framework for Real Time Task Achievement*. in Gini, M.; Voyles, R. eds, *Distributed Autonomous Robotic Systems 7 (DARS)*, Springer-Verlag, 187-196., Heidelberg, Germany.

- Sariel S., ve Balch T., (2006b). Efficient bids on task allocation for multi-robot exploration, *Proceedings, The 19th International Florida Artificial Intelligence Society Conference*, 116-121, Melbourne Beach, Florida, USA .
- Sariel S., Balch T., ve Erdoğan N., (2006). Robust multi-robot cooperation through dynamic task allocation and precaution routines, *Proceedings, The 3rd International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 196-201, Setubal, Portugal.
- Toth, P., ve Vigo, D., (2001). *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.
-
- CPLEX: CPLEX -10.0 (ILOG) User Manual. <http://www.ilog.com/> (2007).
- TSPLIB: <http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/tsp/index.html> (2007).
- Webots: Webots User Guide 5.1.11. <http://www.cyberbotics.com/cdrom/common/doc/webots/guide/guide.pdf> (2006).