

Bilişsel Fabrikalarda Birden Fazla Robot Takımı için Eniyileştirilmiş Ayrıştırılabilir Plan Hesaplanması

Finding Optimal Decoupled Plans for Multiple Teams of Robots in Cognitive Factories

Zeynep G. Sarıbatır, Peter Schüller, Volkan Patoglu, Esra Erdem
Mühendislik ve Doğa Bilimleri Fakültesi
Sabancı Üniversitesi
İstanbul, Türkiye
Email: {zgsaribatır,peterschueller,vpatoglu,esraerdem}@sabanciuniv.edu

Özetçe —Bu makalede robotlardan oluşan birden fazla takım için, bir aracı yardımı ile, eniyileştirilmiş planları bulmak amacıyla geliştirdiğimiz yöntemler açıklanmaktadır. Bu problemde, her takımın kendine ait çalışma alanında tamamlaması gereken görevleri vardır ve takımlar arası robot değişimi yapılabilmektedir. Ayrıca takımlar şu kısıtları göz önünde bulundurmaktadırlar: 1) takımlar (ve aracı) birbirlerinin çalışma alanı ya da görevleri hakkında bilgi sahibi değildirler (örn. gizlilik nedeniyle); 2) her takım ya robot ödünç verebilir ya da ödünç alabilir, ama ikisini birden yapamaz (örn. robotların çalışma alanları arasında taşınmaları ve ayarlanmaları genelde zor ve masraflıdır). Bu problemi çözmek için önerdiğimiz yöntem, güncel SAT çözücülerini ve çözüm kümesi programlama (ASP) çözücülerini kullanmaktadır. Yaklaşımımızın uygulanabilirliği ve kullanılabilirliği bilişsel fabrikalar üzerinde çeşitli senaryolarla yaptığımız deneylerle gösterilmiştir.

Anahtar Kelimeler—Yapay Zeka Planlama, Bilgi Gösterimi ve Akıl Yürütme

Abstract—We introduce a novel method to find optimal plans for multiple teams of robots through a mediator, where each team is given a task to complete in its workspace on its own and where teams are allowed to transfer robots between each other, subject to the following constraints: 1) teams (and the mediator) do not know about each other's workspace or tasks (e.g., for privacy purposes); 2) every team can lend or borrow robots, but not both (e.g., transportation/calibration of robots between/for different workspaces is usually costly). We introduce a novel method to solve this problem using state-of-the-art SAT solvers and ASP solvers. We show applicability and usefulness of our approach by experiments on various scenarios of cognitive factories.

Keywords—AI Planning, Knowledge Representation and Reasoning

I. GİRİŞ

Geleneksel imalat ve montaj sistemleri, özelleştirilebilir ve çok çeşitli ürünler için sürekli artan pazar talebine, uygun maliyetli ve kısa teslimat zamanlı cevap vermede yetersiz kalmaktadır. Bu nedenle, özelleştirilebilir ürünlerin otomatik

üretimine yeni yaklaşımlar oluşturmak üretim sektöründe verimliliği arttırmak, rekabeti ve ekonomik sürdürülebilirliği sağlamak için çok önemlidir.

Bilişsel Fabrika kavramı [1] [2] [3], geleneksel üretime alternatif olup, bir yandan otomasyondaki esnekliği, uyumluluğu ve verimlilik talebini dengelerken, bir yandan yüksek düzeyde güvenilirliği gerçekleştirerek, kayda değer faydalar vadeden yeni bir paradigmadır. Bilişsel fabrikalar, üretim sistemini yüksek seviyede akıl yürütme yetisi ile bilişsel robotik yaklaşımıyla geliştirmeyi amaçlar. Dolayısıyla, bu sistemler kendi eylemlerini planlama yetisine sahiptirler. Bilgiye dayalı planlama ve karar verme algoritmalarını kullanarak, bilişsel fabrikalardaki birden fazla robot takımı, ortak kaynakları verimli olarak kullanabilirler ve özelleştirilebilir ürünlerin üretimini teslim tarihlerine yetişmeyi garantileyecek şekilde sağlayabilirler. Bilişsel fabrikalardaki en önemli zorluklardan biri, birden fazla robotun takımlar halinde çalışarak, bir üretim siparişini en yakın teslim tarihine yetiştirmesidir. Özelleştirilebilir ürünler için teslim tarihini öne almak, fabrika giderlerini azaltarak daha uygun maliyet sağlamaktadır. Ayrıca, ısınma ve ışıklandırma gibi tesis altyapısını daha verimli kullanarak enerji kaynaklarını koruyup olumsuz çevresel etkileri azaltmaktadır.

Bu makalede, birden fazla robot takımının kendi başlarına, kendi çalışma alanlarında tamamlamaları gereken görevlerinin olduğu ve takımlar arasında robot değişimine izin verilen bir bilişsel fabrika senaryosu ele alıyoruz. Hedefimiz bütün takımlar için eniyileştirilmiş ortak bir plan bulmaktır. Bu sayede bütün görevler en kısa zamanda tamamlanabilir. Bu senaryoda her takımın uyması gereken kısıtlar vardır:

- [C1] Takımlar birbirlerinin çalışma alanları ya da görevleri hakkında (örn. gizlilik nedeniyle) bilgi sahibi değildirler.
- [C2] Robotların farklı çalışma alanları arasında sık sık ödünç verilmesi/alınması, maliyeti yüksek olduğundan tercih edilmemektedir.

Bu nedenle, her takım ya robot ödünç verebilir ya da ödünç alabilir. Öte yandan, her takım birden fazla takıma robot ödünç

verebilir; birden fazla takımdan robot ödünç alabilir.

II. YÖNTEM

Yöntemimiz, bir aracı yardımı ile, [C1] ve [C2] kısıtlarına uyan, bütün takımlar için eniyileştirilmiş, en fazla l adım uzunluğunda ortak bir plan bulmaya yöneliktir. Aracı, hiçbir takıma ait değildir ve takımların çalışma alanları, görevleri ve amaçları hakkında bilgisi bulunmamaktadır. Yaklaşımımız iki safhadan oluşmaktadır: takımlar arasında bir koordinasyon bulmak ve sonrasında eniyileştirilmiş ortak bir plan hesaplamak.

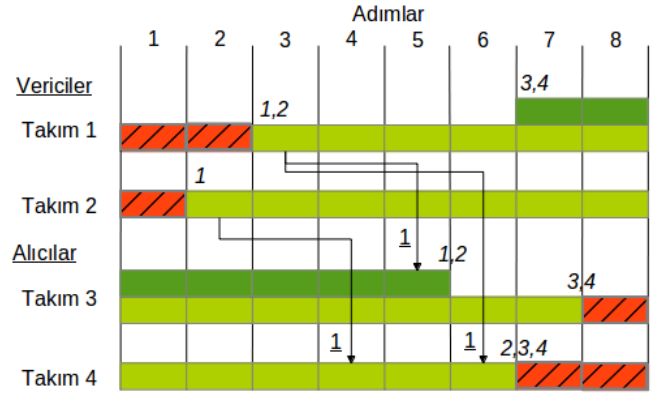
İlk safhada, ortak planın uzunluğunu belirten negatif olmayan bir tamsayı $k \leq l$ için: 1) Takımların kaç adet robotu ne zaman ödünç vererek/alarak görevlerini k adımda tamamlayıp tamamlayamayacaklarını teşhis etmek amacıyla, aracı, her takıma (herhangi bir sırayla) yanıtı evet/hayır olan sorular sorar. 2) Bütün cevaplar toplandıktan sonra, aracı, yukarıdaki [C1] ve [C2] kısıtlarına uyan, takımlar arasında bir koordinasyon (yani hangi takımın ne zaman hangi takıma kaç tane robot vermesi gerektiğini) bulmaya çalışır. Eniyileştirilmiş ortak bir plan için, k 'nin değeri ikili arama ile bulunabilir.

İkinci safhada, eniyileştirilmiş k değeri için takımlar arasında bir koordinasyon bulduktan sonra, aracı, her takıma ne kadar sayıda robotu ne zaman hangi takıma vermesi (ya da hangi takımdan alması) gerektiğini bildirir. Bu bilgiyle beraber her takım görevini tamamlamak için eniyileştirilmiş yerel planını (uzunluğu k 'den küçük ya da eşit olacak şekilde) hesaplar. Bütün takımlar için eniyileştirilmiş ortak plan, yerel planların birleşimidir.

Burada dikkat edilmesi gereken nokta, aracının, takımların çalışma alanlarını, görevlerini, planlarını, eylemlerini, amaçlarını, vb. bilmediği için ortak bir planı kendi başına bulamamasıdır. Aslında, bütün çalışma alanlarının ve takımların ortak gösteriminin büyük olması yüzünden, ortak bir planı hesaplamak için merkezi bir yöntem kullanmak çoğu durumda ölçeklenebilir değildir. Bir diğer önemli nokta ise, takımların birbirleri ile iletişim kurmamasıdır. Aksi takdirde, soru sayısı (ve ileti alışverişi) önemli miktarda artar ve eniyileştirilmiş ortak bir plan bulmak daha fazla zaman tüketen veya uygulanamaz bir işleme dönüşür.

Zamana dayalı kısıtlarla beraber, uzunluğu k adım olan planlar bulmak NP-Tamdır [4]. Bu nedenle, eniyileştirilmiş ortak bir plan hesabının iki safhası da zor hesaplama problemlerini çözme içerir. Birinci safhada aracı tarafından sorulan soruların her biri zamana dayalı kısıtlarla beraber bir planlama problemidir, yani NP-Tamdır. Uzunluğu en fazla k adım olan ortak bir plan için takımlar arasında bir koordinasyon bulunması da NP-Tamdır. Bu zor problemlerin çözümünde, yaklaşımımız paralelleştirmeye izin vermektedir. Birinci safhada, her takım sorulara diğer takımlardan bağımsız bir şekilde cevap verdiği için, sorular paralel olarak cevaplandırılabilir. İkinci safhada, her takım eniyileştirilmiş yerel planını kendi başına hesapladığından eniyileştirilmiş yerel planlar da paralel olarak hesaplanabilirler.

Yöntemimizde, zamana dayalı kısıtlarla beraber planlama problemlerini (ve aracı tarafından sorulan sorulara cevap verilmesini) MANYSAT [5] gibi güncel paralel SAT çözücülerini kullanarak; koordinasyon problemini de CLASP [6] gibi güncel ASP çözücülerini kullanarak çözme öneriyoruz. SAT çözücülerini CCALC [7] vasıtasıyla kullanıyoruz. CCALC dinamik



Şekil 1. Takımların sorulara olan cevaplarının bir özeti.

ortamların eylem modelleme dili olan $C+$ [8] ile gösterimine izin verip takımların zamana dayalı kısıtlarla beraber planlama problemlerini çözmelerini sağlar. ASP çözücülerini kullanmak için, koordinasyon probleminin ASP'de [9] [10] gösterimini yapmaktayız.

Yöntemimizin kullanılabilirliğini ve uygulanabilirliğini bilişsel fabrikalar üzerinde çeşitli senaryolarla yaptığımız deneylerle göstermekteyiz.

III. ÖRNEK SENARYO

Örnek senaryomuzda 4 takım bulunmaktadır. 1. ve 2. takımlar robot ödünç verebilir, 3. ve 4. takımların da robot ödünç alması gerekmektedir. Takımlar arasında bir koordinasyon bulmak istediğimiz ortak planın uzunluğu $\bar{l} = 8$ ve takımların aralarında değiştirebildiği robot sayısı en fazla $\bar{m} = 4$ olarak belirlenmiştir. Takımların, kaç tane robotu hangi adımda ödünç vererek/alarak görevlerini \bar{l} adımda tamamlayıp tamamlayamayacaklarını teşhis etmek için aracının sorduğu evet/hayır sorularına verdikleri cevaplar Şekil 1'de özetlenmiştir. Bu şekilde l . adım ($l \leq \bar{l}$) için olumlu (veya, olumsuz) cevaplar yeşil/düz (veya, kırmızı/çizgili) şeklinde; verilebilen ya da alınabilen robot sayısı, m , çizgilerin üzerinde belirtilmiştir. Bu cevaplara göre, 1. takım 3. adımdan sonra 2 robot ya da 7. adımdan sonra 4 robot, 2. takım 2. adımdan sonra 1 robot vererek, 3. takım 5. adımdan önce 1 robot ya da 7. adımdan önce 3 robot, 4. takım 6. adımdan önce 2 robot alarak görevlerini 8 adımda tamamlayabilirler. Bu bilgilere dayanarak, robot ödünç verebilen veya alması gereken takımlar için, m tane robotu en erken/geç ödünç verebilecekleri/alabilecekleri adımları göstermek amacıyla, $Lend_earliest_m/Borrow_latest_m$ kısmi fonksiyonları tanımlanmaktadır.

Herhangi bir robotun i . takımın çalışma sahasından j . takımın çalışma sahasına transferi için geçecek zamanın $Delay(i, j) = |i - j|$ olduğunu varsayalım. Gecikme zamanını dikkate alarak, robot ödünç verebilen i . takım ile ödünç alınması gereken j . takımın l . adımda robot değişimi yapabileceği $coop_l(i, j) = x$ ile gösterilmektedir. Burada x , i . takımın verebileceği en fazla robot sayısıdır.

Bu senaryo için, $f(i, j) = (l, u)$ şeklinde i . takımın j . takıma u tane robotu l . adımda ödünç verdiğini belirten, takımlar arasında koordinasyonu gösteren bir fonksiyonun, f' 'in,

Tablo I. ALTI SENARYO İÇİN DENEYSSEL SONUÇLAR

Senaryo	Takımlar #	Çalışma Alanları #	İşçi Robot Sayısı #	Toplam Robot Sayısı #	Sipariş (kutu sayısı) #	Soru Sayısı (toplam) #	Soru Cevaplama (ortalama süre) sn	İşbirliği Bulma (ortalama süre) sn	Eniyileştirilmiş Ortak Plan (işbirliği ile) uzunluk	Eniyileştirilmiş Ortak Plan (işbirliği olmadan) uzunluk
1	2	15	1,2	5	6	71	1.00	< 0.1	18	20
2	3	15	1,2,3	9	9	103	1.61	< 0.1	16	20
3	4	15	1,2,3,4	15	12	228	1.22	< 0.1	15	20
4	2	24	2,4	8	8	94	4.31	< 0.1	18	21
5	3	24	2,4,6	18	12	210	4.89	< 0.1	17	21
6	4	24	2,4,6,8	30	16	334	7.10	< 0.1	17	21

bulunması istenmektedir. Bu fonksiyonun bir koordinasyon oluşturması için şu koşulları sağlaması gerekmektedir:

- Her robot ödünç alan takım için, ihtiyaç duyduğu zamanda ödünç verebilen bir takım mevcut olmalıdır.
- Robot ödünç alan hiçbir takım ihtiyacından daha az robot almamalıdır.
- Robot ödünç veren hiçbir takım verebileceğinden fazla robot vermemelidir.

Bu senaryoya, gerekli koşulları sağlayan bir f fonksiyonu bulunabilir: $f(1, 3) = (3, 1)$, $f(1, 4) = (3, 1)$, $f(2, 4) = (2, 1)$.

- Tanımlanan $coop_3(1, 3) = 2$, $coop_3(1, 4) = 2$ ve $coop_2(2, 4) = 1$ her robot ödünç alan takıma ödünç verebilen takımların varlığını sağlamaktadır.
3. takım için, $f(1, 3) = (3, 1)$, ve öyle bir $m = 1 \leq 1$ bulunuyor ki $Borrow_latest_1(3) = 5 \geq 3$. Bu demektir ki, 3. takım 5. adıma kadar 1 robot alırsa görevini 8 adımda tamamlayabilmektedir. Aynı şekilde, 4. takım için, $f(1, 4) = (3, 1)$ ve $f(2, 4) = (2, 1)$ ve öyle bir $m = 2 \leq 1 + 1$ bulunuyor ki $Borrow_latest_2(4) = 6 \geq 3 = \max\{3, 2\}$.
1. takım için, $f(1, 3) = (3, 1)$ ve $f(1, 4) = (3, 1)$, ve öyle bir $m = 2 \geq 1 + 1$ bulunuyor ki $Lend_earliest_2(1) = 3 \leq 3$. Bu demektir ki, 1. takım 3. adımdan itibaren 2 robot verebilmektedir. Aynı şekilde, 2. takım için, $f(2, 4) = (2, 1)$, ve öyle bir $m = 1 \geq 1$ bulunuyor ki $Lend_earliest_1(2) = 2 \leq 2$.

IV. DENEYSSEL DEĞERLENDİRME

Bir boyama fabrikası ortamında çeşitli deneyler gerçekleştirdik. Bu ortamda, verilen süre içerisinde her takım kutular imal etmektedir. Her kutunun, boyama, cilalama, ve damgalama gibi çeşitli aşamalardan geçmesi ve aynı zamanda belirli zaman kısıtlarına uyulması gerekmektedir. Bu ortamda, iki çeşit robot bulunmaktadır: *işçi robotlar* kutuların

üzerinde çalışırlar, kendilerini işlemin farklı aşamalarına göre ayarlayabilirler ve takımlar arasında değişimleri yapılabilir; *yükleyici robotlar* işçilerin pillerinin sürekliliğini sağlarlar ve takımın planını gözlerler, ancak bu robotların takımlar arasında değişimleri yapılamaz. İşçi robotların takımlar arasındaki değişiminin talep edilen miktardaki kutuların üretilmesi için gereken zamanı azaltabileceği senaryolar oluşturmak amacıyla, farklı boyutlardaki takımları bir araya getirdik. Bu boyama fabrikasının biçimsel gösterimi, daha önceki çalışmamızda örnek olarak verilen bilişsel fabrika gösterimine benzerdir [11].

Deneysel senaryolarımız için (bkz. Tablo I), takım çalışma alanlarını sırasıyla $5 \times 3 = 15$ ve $8 \times 3 = 24$ karelaj olarak belirledik. Her senaryoda takım sayısını, takımlardaki robot sayısını ve her takım tarafından imal edilmesi gereken kutu sayısını değiştirdik. Takımlarda, her iki işçi robot için bir yükleyici robot bulunmaktadır. Örneğin, 5. senaryoda her takımın $8 \times 3 = 24$ karelajlık çalışma alanı vardır. Takımlar 2, 4 ve 6 işçi robot ve uygun olarak, 1, 2, ve 3 yükleyici robottan oluşmaktadır. Böylece 5. senaryonun toplam robot sayısı 18'dir.

Deneylerimizi 2.4GHz 32 Intel® E5-2665 CPU işlemcili ve 64GB bellekli (deneylerimiz 300MB bellekten fazlasını hiç kullanmamıştır) Linux sunucusu üzerinde gerçekleştirdik. Algoritma PYTHON dilinde uygulanmıştır. Sorulara cevap vermek için, SAT çözücüsü MANYSAT ile CCALC kullanılmıştır. İşbirliği bulma problemini çözmek için ASP çözücüsü CLASP kullanılmıştır.

Tablo I'de gösterilen sonuçlar, boyutları değişen altı senaryonun 5 kere uygulanmasından ortaya çıkan sonuçların ortalamalarıdır. Tablo her senaryo için, takımların cevapladığı soruların toplam sayısını, MANYSAT kullanarak bir sorunun cevaplanması için harcanan ve CLASP kullanarak takımlar arasında bir koordinasyon bulmak için harcanan ortalama CPU zamanını, takımlar işbirliği yapıyorken ve yapmıyorken ortaya çıkan eniyileştirilmiş ortak planın uzunluğunu göstermektedir. Örneğin, 5. senaryoda, toplam 210 soru takımlar tarafından cevaplanmıştır; bir soruya ortalama cevap süresi 4.89 saniyedir;

takımlar arasında bir koordinasyon bulmak bir saniyeden az zaman almaktadır. Bu koordinasyonla birlikte eniyileştirilmiş ortak plan 17 adım; hiç koordinasyon olmadan eniyileştirilmiş ortak plan 21 adım olmaktadır.

Tablodan gözlemleyebildiğimiz gibi aracı ile bir koordinasyon fonksiyonu bulmak önemsiz miktarda zaman almaktadır. Hesaplama zamanının çoğunluğu takımların sorulara cevap vermesine harcanmaktadır. Problem boyutu arttıkça, CCALC tarafından sağlanan önermeli kuram büyüdüğünden MANYSAT'ın bir cevap bulmasını zorlaştırmaktadır.

Ayrıca, takımların işbirliği varken ve yokken oluşan eniyileştirilmiş ortak planlar ve toplam hesaplama zamanları arasındaki farkı gözlemleyebiliriz. Örneğin, 5. senaryo için, eğer takımların işbirliğine izin verilirse, 17 adım uzunluğunda eniyileştirilmiş ortak plan 17 dakikada bulunabilir; eğer takımların işbirliğine izin verilmezse, 21 adım uzunluğunda eniyileştirilmiş plan 2.5 dakikada bulunabilir. Bu hesaplama maliyeti, işlem süresindeki %20 azalmanın sağladığı genel maliyet tasarrufu ve daha iyi müşteri memnuniyeti ile karşılaştırılınca önemsiz kalabilir. Bu nedenle, ekonomik sürdürülebilirlik için takımların işbirliği tercih edilebilir.

V. SONUÇ

Robotlardan oluşan birden fazla takım için, çalışma alanları, görevleri, robotları, planları, eylemleri, hedefleri, vb. gibi özel bilgilerini içermeyen evet/hayır sorularına verdikleri (SAT çözücü tarafından bulunan) cevaplarla takımlar arasında (ASP çözücü kullanarak) bir koordinasyon belirlemek suretiyle, eniyileştirilmiş ortak bir plan bulmaya yönelik yeni bir yöntem tanıttık. Koordinasyon belirleme problemini tanımladık ve zorluğunu kanıtladık. Detaylı matematiksel tanımımız ve NP-Tamlık ispatımız [12]'de yer almaktadır. Yöntemimizin kullanılabilirliğini bir bilişsel fabrika ortamında değerlendirdik, ve toplam işlem zamanında önemli bir düşüş gözlemledik. Bu düşüş, maliyet tasarrufu ve daha iyi müşteri memnuniyeti ile ekonomik sürdürülebilirlik için önemli bir sonuçtur.

Yöntemimiz birçok şekilde geliştirilebilir. Eğer aracının basit görevleri ve değiştirilebilir robotların çeşitlerini bilmesine izin verilirse, o zaman "Ağır bir kutu taşıyabilen bir robotu k' adımdan önce ödünç verirsene, görevini k adımda tamamlayabilir misin?" gibi sorular sorabilir. Takımlar böyle sorulara SAT çözücüsü kullanarak cevap verebilir, çünkü görevleri robotlar ile bağdaştırılan bilgi [13]'de olduğu gibi eylem modellerine ve sorulara yerleştirilebilir. Koordinasyon bulmaya yönelik ASP formülasyonu, ilgili kısıtlar eklenerek biraz değiştirilebilir.

Eniyileştirilmiş ortak bir plan bulmaya yönelik algoritmamız icra takibi çerçevesine yerleştirilebilir. Plan icra sırasında başarısız olursa, algoritmamız yeni bir eniyileştirilmiş ortak plan bulabilir. Sonraki planlar için önceden hesaplanmış plandan alınmış bilgiler (örn. takımların rolleri ve sınırları) yeniden kullanılabilir.

KAYNAKLAR

- [1] M. Beetz, M. Buss, and D. Wollherr, "CTS - What is the role of artificial intelligence?" in *Proc. of KI*, 2007, pp. 19–42.
- [2] M. Zaeh, M. Beetz, K. Shea, G. Reinhart, K. Bender, C. Lau, M. Ostgathe, W. Vogl, M. Wiesbeck, M. Engelhard, C. Ertelt, T. Rühr, M. Friedrich, and S. Herle, "The cognitive factory," in *Changeable and Reconfg. Manufacturing Systems*, 2009, pp. 355–371.

- [3] M. Zaeh, M. Ostgathe, F. Geiger, and G. Reinhart, "Adaptive job control in the cognitive factory," in *Enabling Manufacturing Competitiveness and Economic Sustainability*, H. A. ElMaraghy, Ed. Springer Berlin Heidelberg, 2012, pp. 10–17.
- [4] H. Turner, "Polynomial-length planning spans the polynomial hierarchy," in *Proc. of JELIA*, 2002, pp. 111–124.
- [5] Y. Hamadi, S. Jabbour, and L. Sais, "Control-based clause sharing in parallel sat solving," in *Proc. of IJCAI*, 2009, pp. 499–504.
- [6] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub, "clasp: A conflict-driven answer set solver," in *Proc. of LPNMR*, 2007, pp. 260–265.
- [7] N. McCain and H. Turner, "Causal theories of action and change," in *Proc. of AAAI/IAAI*, 1997, pp. 460–465.
- [8] E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain, and H. Turner, "Nonmonotonic causal theories," *AJL*, vol. 153, pp. 49–104, 2004.
- [9] V. Lifschitz, "What is answer set programming?" in *Proc. of AAAI*. MIT Press, 2008, pp. 1594–1597.
- [10] G. Brewka, T. Eiter, and M. Truszczynski, "Answer set programming at a glance," *Commun. ACM*, vol. 54, no. 12, pp. 92–103, 2011.
- [11] E. Erdem, K. Haspalamutgil, V. Patoglu, and T. Uras, "Causality-based planning and diagnostic reasoning for cognitive factories," in *Proc. of IEEE Int. Conf. Emerging Technologies and Factory Automation (ETFA)*, 2012.
- [12] E. Erdem, V. Patoglu, Z. G. Saribatur, and P. Schueller, "Finding optimal plans for multiple teams of robots through a mediator," in *Submitted*, 2013.
- [13] E. Erdem and V. Patoglu, "Applications of action languages in cognitive robotics," in *Correct Reasoning*, 2012, pp. 229–246.