Uyarlanabilir Doğrusal Programlama Kod Çözümü İçin Sezgisel Bir Yaklaşım
A Heuristic Method For Adaptive Linear Programming Decoding

Abdullah Sarıduman[1], Ali Emre Pusane[1] ve Z. Caner Taşkın[2]
[1] Elektrik-Elektronik Mühendisliği Bölümü, Boğaziçi Üniversitesi, İstanbul, Türkiye
[2] Endüstri Mühendisliği Bölümü, Boğaziçi Üniversitesi, İstanbul, Türkiye
{abdullah.sariduman, ali.pusane, caner.taskin}@boun.edu.tr

Özetçe **—Uyarlanabilir kod çözücüler (ALP) özellikle düşük-yoğunluklu eşlik-denetim kodların (LDPC) çözümünde kullanılmaktadır. ALP kod çözücülerinin çalışma prensibi doğrusal programlama (LP) kod çözücülerinin bulduğu kesirli çözümleri olursuz hale getirebilecek artık-eşlik denetim denklemleri üretimine dayanmaktadır. Üretilen bu artık-eşlik denetim denklemleri kod çözücünün hata başarım oranını artırmaktadır. Bu çalışmada, hata başarımı artıran artık-eşlik denetim denklemlerinin bulunmasını kolaylaştırmak için LP modeli yardımcı değişkenler ile tanımlanmış ve önerilen model üzerinden artık-eşlik denetim denklemleri sezgisel bir yöntem kullanılarak aranmıştır. Benzetim sonuçları önerilen algoritmanın artık-eşlik denetim denklemlerini diğer ALP kod çözücülerine göre daha kısa sürede bulabildiğini göstermiştir.**

Anahtar Kelimeler—düşük-yoğunluklu eşlik-denetim kodları; doğrusal programlama; uyarlanabilir kod çözme.

Abstract—**Adaptive linear programming (ALP) decoders are mainly used for decoding low-density parity-check (LDPC) codes. The principle of ALP decoders is based on generating redundant-parity check equations, which could eliminate fractional solutions of linear programming (LP) decoders. These generated redundant parity check equations increase the error rate performance of decoder. In this paper, LP model is defined with auxiliary variables to facilitate finding redundant-parity check equations, and redundant parity check equations are searched over proposed LP model with a heuristic method. Simulation results demonstrate that the proposed algorithm could find redundant-parity check equations in a shorter time than other ALP decoders.**

Keywords—low-density parity-check codes; linear programming; adaptive decoding.

I. GİRİŞ

Doğrusal programlama (LP) kod çözücü ilk defa Feldman, Wainwright, and Karger tarafından 2005 yılında geliştirilmiştir [1]. Bu kod çözücünün sağladığı en büyük kazanç, kod çözümü sonucunda tam sayı değerlerine sahip bir vektör elde ettiğinde en yüksek olabilirlikli (ML) özelliğine sahip bir kod sözcüğünü bulmayı başarmasıdır. LP kod çözücüsü kesirli değerlere sahip bir vektör bulduğu zaman ise kod çözümünün hatalı bir şekilde sonlandığı kabul edilmektedir. Daha sonraki yıllarda gerçekleştirilen çalışmalarda LP kod çözücüsü çözüm olarak kesirli değerlere sahip bir vektör elde ettiği zaman bu çözümü olursuz hale getiren kısıtlar (geçerli kısıtlar) LP modele eklenerek LP kod çözücüsünün tam sayı değerlerine sahip bir vektör bulmasına çalışılmıştır [2], [3], [4], ve [5]. Uyarlanabilir doğrusal programlama (ALP) kod çözücüsü olarak adlandırılan bu tür kod çözücülerin hata başarımları eklenen kısıtlarla ML kod çözücülerinin hata başarımlarına yaklaşmaya başlamıştır.

Literatürde önerilen ALP kod çözücülerinin çalışma prensipleri temel olarak benzerlik göstermektedir. Bu çalışmaların farklı olduğu kısımlar ise geçerli kısıtları bulma yöntemlerinde görünmektedir. [2] çalışmasında geçerli kısıtları arama yöntemi çizge içerisinde çevrim oluşturabilen kesirli denetim düğümlerini toplanmasına dayanmaktadır. [4] çalışmasında ise kesirli sonucu olursuz çözüm haline getirebilecek artık eşlik denetim denklemlerinin bazı koşulları sağlaması gerektiği gösterilmiş ve eşlik denetim matrisi üzerinde temel satır-sütün işlemleriyle bu artık eşlik denetim denklemleri aranmıştır. Bu çalışmadaki yazarlar tarafından geliştirilen [5] çalışmasında ise geçerli kısıtlar tam sayı programlama eniyileme yöntemi kullanılarak aranmıştır.

Bu çalışmamızda çevrim oluşturabilen düğümlerin daha verimli bulunabilmesi için LP kod çözücüsü [6] çalışmasında önerilen LP kod çözücüsüne benzer bir şekilde tanımlanmış ve geçerli kısıtları ürettiği ispatlanan artık eşlik denetim denklemleri sezgisel yöntemlere dayalı algoritma (HALP) ile hızlı bir şekilde bulunmuştur.

Bildirinin geri kalanı şu şekilde düzenlenmiştir: Bölüm 2'de LP kod çözücüsü anlatılmıştır. Bölüm 3'te bu çalışmada önerilen ALP kod çözücüsünden bahsedilmiştir. Bölüm 4'te sayısal sonuçlar gösterilmiş ve Bölüm 5'de sonuçlar ve gerçekleştirilebilecek gelecek çalışmalardan bahsedilmiştir.

II. LP KOD ÇÖZÜCÜLERİ VE ALP KOD ÇÖZÜCÜLERİ

(n, k) boyutlu LDPC kodu n bit düğümü ve $m = n - k$ denetim düğümünden oluşmaktadır. Eşlik denetim denklemleri denetim düğümlerine, kod çözücünün çözmeye

Bu bildirideki çalışmalar TÜBİTAK 113M499 nolu proje kapsamında desteklenmiştir.

978-1-5090-1679-2/16/$31.00 ©2016 IEEE

çalıştığı kod vektörü de bit düğümlerine karşılık gelmektedir. Denetim düğümleri eşlik denetim denklemleri sayesinde yerel kod sözcüklerini oluşturmaktadır ve yerel kod sözcüklerinin dış bükey zarflarıyla yerel bir politop elde edilebilir. LP kod çözücüsü ise her bir denetim düğümüne ait olan yerel politopların kesişimi ile elde edilen temel politop üzerinden eniyileme yapmaktadır:

$$\text{enküçükle} \sum_{j=1}^{n} \gamma_j x_j \tag{1}$$

$$\text{kısıtlar: } \mathbf{x} \in poly(\zeta_1) \cap poly(\zeta_2) ... \cap poly(\zeta_m) \tag{2}$$

Burada, $poly(\zeta_i)$, i. denetim düğümüne ait olan yerel politopu göstermektedir. x_j parametresi j. bit düğümünü ve γ_j parametresi j. bit düğümünün amaç fonksiyonun katsayısını göstermektedir. Yerel politopu oluşturan kısıtlar

$$g_i(x) = \sum_{x_j \in S} (1 - x_j) + \sum_{x_j \in (N(c_i) \setminus S)} x_j \geq 1 \tag{3}$$

ile ifade edilebilir [1], [2]. S, bir denetim düğümüne ait olan tek ağırlıklı tüm vektörlerin oluşturduğu kümeyi ifade etmektedir. Kısıt (3) ile tek ağırlıklı vektörler yasaklanarak yerel kod sözcüklerin dış bükey zarfı oluşturulmaktadır.

LP kod çözücüsü kesirli bir çözüm bulduğu zaman bit düğümleri 0 ile 1 arasında farklı değerler alacaktır. Mevcut denetim düğümlerinin toplanması ile elde edilen artık denetim düğümünün yerel politopu Kısıt (3) ifadesini sağlamayabilir. Bu kısıtın LP modele eklenmesi ile beraber LP kod çözücüsü tekrar çalıştırıldığında elde edilecek çözümün bulunan son çözümden farklı olması garantilenmiş olunur. Bu yöntemle de ALP kod çözücüleri tasarlanmaktadır.

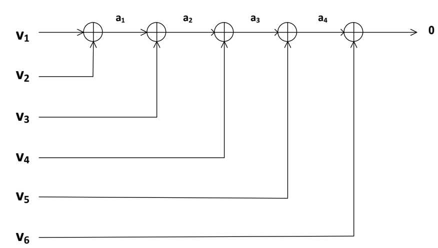
III. HALP Kod Çözücüsü

Artık denetim düğümünün geçerli bir kısıt oluşturabilmesi için toplanan denetim düğümlerinin en az bir çevrim içermesi gerekmektedir [2]. Fakat, bu çevrimlerin tamamının bulunması zor bir problemdir. Ayrıca çevrimdeki denetim düğümlerinin toplanması ile elde edilecek artık denetim düğümünün geçerli bir kısıdı oluşturabilmesi garanti edilemez. HALP kod çözücüsünde denetim düğümleri, temel politopun yapısını değiştirmeyecek şekilde yardımcı değişkenler kullanılarak yeniden tanımlanmıştır. Yeni tanımlama ile çevrimler hızlı ve verimli bir şekilde bulunmuştur. Ayrıca elde edilen artık denetim düğümünün bazı koşullar altından geçerli bir kısıt oluşturabildiği ispatlanmıştır.

Eşlik-denetim derecesi 6 olan bir denetim düğümü c_1 için (mod 2)'deki gösterimi aşağıdaki gibi tanımlanmıştır:

$$[v_1 + v_2 + v_3 + v_4 + v_5 + v_6]_2 = 0 \tag{4}$$

c_1 denetim düğümlerinin doğrusal toplamı yardımcı değişkenler (a_1, a_2, a_3, a_4) kullanılarak Şekil 1'deki gibi gösterilebilir. İkili tabanda çalışıldığı için Eşitlik (4) aşağıdaki gibi 4 denklemle gösterilebilir.



Şekil 1: Denetim düğümlerinin yardımcı değişkenlerle gösterimi

$$[v_1 + v_2 + a_1]_2 = 0 \tag{5}$$

$$[a_1 + v_3 + a_2]_2 = 0 \tag{6}$$

$$[a_2 + v_4 + a_3]_2 = 0 \tag{7}$$

$$[a_3 + v_5 + v_6]_2 = 0 \tag{8}$$

Bu işlemler neticesinde eşlik-denetim derecesi 6 olan bir denetim düğümü yerine eşlik-denetim derecesi 3 olan 4 farklı denetim düğümlerinin kullanılabileceği gösterilmektedir.

Önerme 1: Temel politopları oluşturan kesitlerin sayısı standart LP kod çözücüsü için denetim düğümünün derecesi $d(c)$ ile üstel orantılı olarak değişirken, yardımcı değişkenler kullanılarak üretilen temel politop $d(c)$ ile doğrusal orantılı sayıda kesitlerle ifade edilebilir.

İspat: Bir LDPC kodun j. denetim düğümünün yerel politopunu tanımlamak için bütün tek ağırlık vektörleri yasaklamak gerekmektedir. Bu nedenle tam olarak $(2^{d(c_j)-1})$ tane kısıt gerekmektedir. Diğer yandan, j. denetim düğümünü derecesi 3 olan $(d(c_j) - 2)$ tane denetim düğümü ile gösterebiliriz. Bu durumda her bir denetim düğümü için $2^{3-1} = 4$ tane kısıt gerekmektedir. Toplamda $(d(c_j) - 2)$ denetim düğümü yaratıldığı için bu yöntemle j. denetim düğümünün yerel politopu $(4 * (d(c_j) - 2))$ kısıt ile gösterilebilir. □

Temel politopu yardımcı değişkenlerle belirtildiği gibi tanımlamak sadece kısıt sayısını azaltmayacak, aynı zamanda geçerli eşitsizlikler üretmek için aranan artık denetim düğümlerinin bulunması kolaylaştıracaktır. Derecesi 3 olan denetim düğümünün yerel politopu Eşitlik (5) örneği için

$$v_1 + v_2 \geq a_1 \tag{9}$$

$$v_1 + a_1 \geq v_2 \tag{10}$$

$$v_2 + a_1 \geq v_1 \tag{11}$$

$$v_1 + v_2 + a_1 \leq 2 \tag{12}$$

kısıtları ile tanımlanmaktadır.

Önerme 2: Standart LP kod çözücüsü yardımcı değişkenler kullanılarak tanımlanan temel politopu çözdüğü zaman elde edilen sonuç kesirli değerler içeriyorsa herhangi bir denetim düğümü üç farklı durumda olabilir:

1) 0F Sınıfı (0F-fractional)

Denetim düğümü kesirli değişkenler içermemekte, sadece tam sayılı değişkenler ile tanımlanmaktadır.

2) 2F Sınıfı (2F-fractional)
Denetim düğümü iki tane kesirli değişken ve bir tane tam sayılı değişken içermektedir.

3) 3F Sınıfı (3F-fractional)
Denetim düğümü sadece kesirli değişkenler içermektedir.

İspat: Her hangi bir denetim düğümünün bir tane kesirli değişken ve 2 tane tam sayılı değişken içermediğini göstermek ispat için yeterlidir. Doğrusal programlama yönteminin eniyi çözümü temel politopun bir köşe noktası olması gerekmektedir. Bu durumda eniyi çözümün bir yerel politop için en az iki yüzeyin kesişim noktasına karşılık gelmesi demektir. Bu nedenle (9), (10), (11) ve (12) ile gösterilen kısıtların en az 2 tanesinin eşitlik ile sağlanması gerekmektedir. Bu nedenle bir denetim düğümüne ait iki tane değişkenin değerinin toplamı bu denetim düğümünün üçüncü değişkenin değerine eşit olmak zorundadır. Bu nedenle değişkenlerden sadece birisinin kesirli olması mümkün değildir. □

Önerme 3: 2F sınıfına ait bir denetim düğümünün tam sayılı değişkenin değeri 0 ise diğer iki kesirli değişkenlerinin değerleri birbirine eşittir.

İspat: $[v_1 + v_2 + a_1]_2 = 0$ denetim düğümü için genelliği kaybetmeksizin a_1 değerini 0 kabul edersek, Kısıt (10) ve Kısıt (11), kolayca görüleceği gibi, v_1 ve v_2 değerlerinin eşit olmasını sağlamaktadır. □

Önerme 4: 2F sınıfına ait bir denetim düğümünün tam sayılı değişkenin değeri 1 ise diğer iki kesirli değişkenlerinin toplamı 1'dir.

İspat: $[v_1 + v_2 + a_1]_2 = 0$ denetim düğümü için genel olarak a_1 değerini 1 kabul edersek Kısıt (9) ve Kısıt (12) v_1 ve v_2 değerlerinin toplamının 1 olmasını sağlamaktadır. □

Önerme 5: Bir artık denetim düğümü toplamda 2 tane kesirli değere sahip değişken düğümü içeriyorsa ve bunların toplamı 1'den farklıysa veya değişkenlerin değerleri eşit değilse, bu denetim düğümünün yerel politopu LP kod çözücüsünün son bulduğu çözümü içermemektedir.

İspat: Kısıt 3'da verilen g_x değerini enazlamak için 1 değerine sahip değişkenlerin yasaklanan tek ağırlıklı kümeye (S), 0 değerine sahip değişkenlerin çift kümeye ait olduğu varsayılabilir. Bu durumda tam sayılı değişkenler, g_x değerine katkı vermemektedir. Geriye kalan iki kesirli değişken (v_1 ve v_2 değerlerine sahip varsayalım) 3 farklı durumda olabilir:

1) $v_1 \leq 0.5$ ve $v_2 \leq 0.5$. Eğer S kümesinin tek sayıda elemanı varsa, bu iki değişkende çift ağırlıklı kümeye dahil olur ve g_x değeri 1'den küçük olacaktır. Eğer S kümesinin çift sayıda elemanı varsa, değişkenlerden biri S kümesine ait olmak zorundadır. $v_1 \geq v_2$ varsayalım. v_1 S kümesine ait olursa $g_x = 1 - v_1 + v_2$ ifadesine eşit olacaktır. g_x değerinin 1'den büyük eşit olması için tek koşul $v_1 = v_2$ olur.

2) $v_1 \geq 0.5$ ve $v_2 \leq 0.5$. Eğer S kümesinin tek sayıda elemanı varsa, bu iki değişken ya S kümesine veya çift ağırlıklı kümeye dahil olmak zorundadır. 0.5 değerine en yakın olanı fazla katkı vereceği kümeye eklersek, (v_2 kabul edelim), kesit oluşturan denklem
$g_x = (1 - v_1) + (1 - v_2) = 1 + (0.5 - v_2) - (v_1 - 0.5)$
ile gösterilebilir. v_2 değeri v_1 değerine göre 0.5'e daha yakın olduğu için g_x 1'den küçük olacaktır. İki kesirli ifadenin değerlerinin 0.5'e eşit uzaklıkta olması durumunda g_x 1 değeri almaktadır. $v_1 - a = v_2 + a = 0.5$ ise $v_1 + v_2 = 1$.

3) $v_1 \geq 0.5$ ve $v_2 \geq 0.5$. İlk durum ile benzer ispatı vardır. □

Önerme 6: 3F sınıfına ait bir denetim düğümünün iki kesirli değişkenin değerleri toplamı diğer değişkenin değerine eşittir.

İspat: Önerme 2 ile verilmiştir. □

1) 3F Sınıfından Bir Tane Denetim Düğümü İçeren Çevrimler: Bir tane kesirli değişkeni ortak olan, 2F sınıfına ait iki tane denetim düğümlerinin toplamından elde edilen denetim düğümü 2F sınıfına ait olacaktır. Tanner çizgisi üzerinden birbirine kesirli değişkenlerle komşu olan 2F sınıfına ait değişken düğümler kolaylıkla bulunabilir ($\mathcal{O}(1)$). Kesirli değişkenler üzerinden oluşan ve içerisinde 3F sınıfından bir tane denetim düğümü içeren çevrimlerde sadece bir tane kesirli değişken olacaktır. [3] ve [4] çalışmalarında üretilen artık denetim düğümünün sadece 1 tane kesirli çözüme sahip olduğunda geçerli eşitsizlik üretebildiği gösterilmiştir. Bu nedenle bu çevrimlerin bulunması geçerli eşitsizliğin bulunmasını garanti etmektedir.

2) 3F Sınıfından İki Tane Denetim Düğümü İçeren Çevrimler: İçerisinde iki tane 3F sınıfından denetim düğümü içeren çevrimlerin toplamından oluşan artık denetim düğümü sadece iki tane kesirli değişken içerecektir. Bu artık denetim düğümünün geçerli bir eşitsizlik üretebilmesi için Önerme 3 veya Önerme 4'deki kurallardan birini ihlal etmesi yeterli olacaktır.

Yukarıdaki örneklerde gösterildiği gibi çevrimlerde bulunan 3F sınıfına ait düğüm sayısının geçerli eşitsizlik bulmanın zorluğunu doğrudan etkilediği gözükmektedir. HALP Kod çözücüsü 3F sınıfına ait düğüm sayısı en az olacak şekilde sezgisel olarak çevrim aramakta ve bu yöntemle geçerli eşitsizlikleri kısa süre içerisinde bulmaktadır.

Algoritma 5: (HALP Kod Çözücüsü)

Girdi: Eşlik-denetim matrisi ($\mathbf{H}$), alınan vektör ($\mathbf{r}$), zaman limiti (t_{max})

1. Yardımcı değişkenler kullanarak eşlik-denetim matrisini yeniden tanımla
2. SDP modelini çöz , y değişkenini sıfırla
3. **Eğer** Çözüm tam sayı ise,
    ML kod sözcüğü bulundu ve DUR
4. **Değilse**
    İçinde y tane 3F ait denetim düğümü olan çevrimleri sezgisel yöntemle ara
5. y değerini bir arttır
6. **Bitir Eğer**
7. Bulunan geçerli eşitsizliği temel politopa ekle ve LP kullanarak problemi çöz.
8. **Eğer** Zaman limiti aşılmamışsa,
    3. adıma git
9. **Değilse**
    Çözüm bulunamadı ve DUR.
10. **Bitir Eğer**

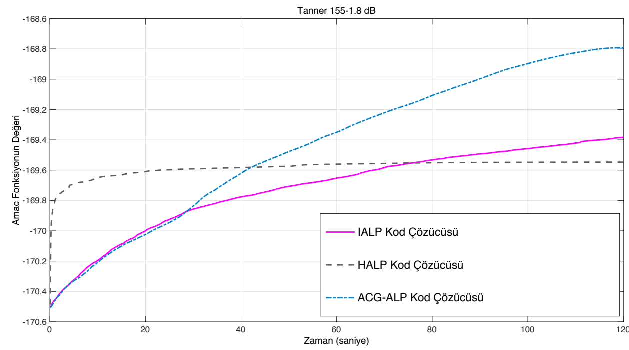
Çıktı: En yüksek olabilirlikli kod sözcüğü veya zaman limiti nedeniyle bir alt sınır bulundu

IV. Sayisal sonuçlar ve tartişma

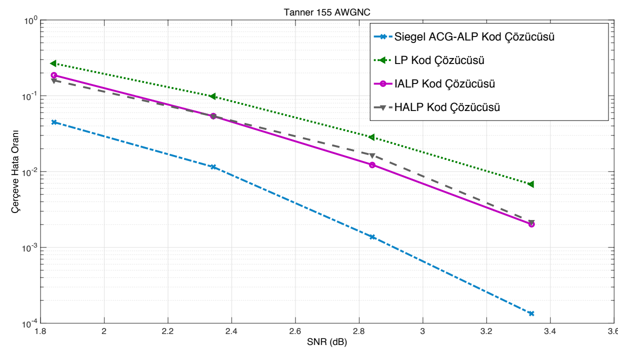
Daha önce yapılan [5] çalışmasında geçerli kısıtlar tam sayı programlama eniyileme yöntemi kullanılarak aranmasından dolayı uzun süreler alabilmekteydi. HALP kod çözücüsü ile birlikte geçerli kısıtlar çok hızlı bir şekilde bulunabilmiştir. HALP kod çözücüsünün zaman başarımını gözlemleyebilmek için literatürde bulunan en iyi ALP kod çözücülerinden biri olan Siegel ve arkadaşlarının önerdiği ACG-ALP kod çözücüsü [4] ve daha önceki çalışmada önerilen tam sayı programlama tabanlı ALP kod çözücüsünün (IALP) zaman başarımları karşılaştırılmıştır. Benzetimler için yaygın olarak kullanılan Tanner'in sözde-dolanır matrislerini temel alarak oluştuduğu (155,64) Tanner kodu kullanılmıştır, [7]. Kod çözücülerin zaman başarımlarını karşılaştırmak için 1.8 dB sinyal gürültü oranına (SNR) sahip AWGN kanalında farklı vektörler üretilmiş ve LP modellerin ortalama amaç fonksiyonu değerlerinin zamana göre değişimi Şekil 2 ile verilmiştir. Şekil 2'den gözlemlenebileceği üzere HALP kod çözücüsü LP modelin amaç fonksiyonu değerini diğer kod çözücülerine göre daha hızlı yükseltmiştir. Fakat, bir noktadan sonra HALP kod çözücüsü LP modelin amaç fonksiyonu değerini artıramamaktadır. Bu nedenle hata başarımının özellikle de ACG-ALP kod çözücüsüne göre daha düşük olması beklenmektedir. Şekil 3 ile kod çözücülerin farklı SNR değerleri için hata başarım grafikleri verilmiştir. HALP kod çözücüsü ACG-ALP kod çözücüsüne göre daha kötü hata başarımına sahip olmasına rağmen ilk geçerli kısıtları diğer algoritmalardan çok daha hızlı bulduğu gösterilmiştir.

V. Sonuçlar

HALP kod çözücüsü ile LP kod çözücüsünün ürettiği kesirli değerlere sahip çözümleri olursuz hale getirecek kısıtların özellikle de algoritmanın başlarındaki iterasyonlarda diğer ALP kod çözücülerine göre daha hızlı bulduğu gösterilmiştir. Fakat, belli sayıdaki kısıdı hızlı bulmasına rağmen bir yerden sonra kısıtları bulması zorlaşmaktadır. Bu nedenle HALP kod çözücüsünün tek başına bir kod çözücüsü yerine literatürde bulunan diğer kod çözücüleri ile birlikte etkin çalışması hedeflenmektedir.



Şekil 2: Zaman Başarımı



Şekil 3: Hata Başarımı

Kaynakça

[1] Feldman, J., Wainwright, M., ve Karger, D., "Using linear programming to decode binary linear codes," *IEEE Transactions on Information Theory*, cilt 51, no. 3, ss. 954–972, 2005.

[2] Taghavi, M.-H. ve Siegel, P., "Adaptive methods for linear programming decoding," *IEEE Transactions on Information Theory*, cilt 54, no. 12, ss. 5396–5410, 2008.

[3] Tanatmis, A., Ruzika, S., Hamacher, H., Punekar, M., Kienle, F., ve Wehn, N., "A separation algorithm for improved LP-decoding of linear block codes," *IEEE Transactions on Information Theory*, cilt 56, no. 7, ss. 3277–3289, 2010.

[4] Zhang, X. ve Siegel, P., "Adaptive cut generation algorithm for improved linear programming decoding of binary linear codes," *IEEE Transactions on Information Theory*, cilt 58, no. 10, ss. 6581–6594, 2012.

[5] Sariduman, A., Pusane, A., ve Taskin, Z., "Adaptive linear programming for decoding LDPC codes," *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, April 2014, ss. 706–709.

[6] Yang, K., Wang, X., ve Feldman, J., "A new linear programming approach to decoding linear block codes," *IEEE Transactions on Information Theory*, cilt 54, no. 3, ss. 1061–1072, March 2008.

[7] Tanner, R., Sridhara, D., Sridharan, A., Fuja, T., ve Costello Jr, D., "LDPC block and convolutional codes based on circulant matrices," *IEEE Transactions on Information Theory*, cilt 50, no. 12, ss. 2966–2984, 2004.