LDPC KODLARININ ÇÖZÜMÜ İÇİN UYARLANABİLİR DOĞRUSAL PROGRAMLAMA

ADAPTIVE LINEAR PROGRAMMING FOR DECODING LDPC CODES

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

***Özetçe* —Düşük-yoğunluklu eşlik-denetim kodların (LDPC) çözümünde kullanılan doğrusal programlama (LP) algoritması geçerli bir kod sözcüğü veya kesirli sayı içeren bir sözde kod sözcüğü üretir. Sözde kod sözcüklerini doğrusal programlama için geçersiz bir çözüm haline getirebilmek ve kod çözücünün hata başarımını artırabilmek için artık eşlik denetim denklemleri üretilip probleme eklenebilir. Bu çalışmada, tam sayı programlama eniyileme yöntemi kullanılarak sözde kod sözcüklerini eleyebilen artık eşlik denetim denklemleri aranmıştır. Üretilen artık eşlik denetim denklemleri yardımıyla uyarlanabilir doğrusal programlamanın başarımının en yüksek olabilirlikli kod çözücülerinin başarımına yakınsayabildiği gösterilmiştir.**

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

***Abstract*—When linear programming is used to decode low-density parity-check (LDPC) codes, the outcome is a codeword or a pseudocodeword that contains fractional symbol values. It is possible to make pseudocodewords infeasible and increase the performance of linear programming decoders by generating redundant parity check equations. In this paper, redundant parity check equations that can eliminate pseudocodewords are searched using an integer programming based optimization approach. We show that the generated parity check equations increase the performance of the adaptive linear programming decoder and that its performance can converge that of a maximum-likelihood decoder.**

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

I. GIRIŞ

LDPC kodlama 1962 yılında Gallager tarafından geliştirilmiştir [1]. Yüksek hata başarımları sayesinde günümüzün popüler kodlarından biri olmuştur. LDPC kodların çözümünde literatürde yaygın olarak kanı yayılımı algoritması kullanmaktadır [2]. Kanı yayılımı algoritmasında, mesajlar, LDPC kodu gösteren iki bölmeli bir çizge üzerinde düğümler arasında karşılıklı yinelenerek gönderilir. Kod çizgelerinin içerdikleri çevrimler nedeniyle bu tarz kod çözücülerin davranışlarını incelemek zordur. Ayrıca, en yüksek olabilirlikli (ML) özelliğine de sahip değildirler. Jon Feldman, 2005 yılında ML özelliğine sahip alternatif bir kod çözücü önermiştir. Doğrusal programlama (LP) kod çözücüsü olarak adlandırılan bu kod çözücü kod çözme işlemini sonlandırdığında bir ML kod sözcüğü veya tam sayı olmayan bir sözde kod sözcüğü üretir [3].

Mesaj geçirme ve doğrusal programlama kod çözücülerinin arasında bazı temel farklılıklar vardır. Bunlardan en önemlisi artık eşlik denetim denklemlerine verdikleri değişik tepkilerdir. Mevcut eşlik denetim denklemlerinin doğrusal bir kombinasyonu olarak üretilecek artık eşlik denetim denklemleri LP kod çözücüsünün hata başarımını kötü yönde etkileyemezken, mesaj geçirme tabanlı kod çözücü kullanımı durumunda ele alınan çizgenin çevrim yapısı etkilendiğinden hata başarımının düşmesi söz konusu olabilecektir. Bu nedenle artık eşlik denetim denklemlerinin üretilip kullanılması LP kod çözücünün hata başarımını yükseltmek için olası bir çözümdür. Bu amaçla uyarlanabilir doğrusal programlama (ALP) kod çözücüsü tanımlanabilir. ALP kod çözücülerinin çalışma ilkesi, LP kod çözücüsünün bulduğu geçersiz çözümü olursuz bir çözüm haline getirebilen artık eşlik denetim denklemlerinin bulunmasına ve LP kod çözücüsünün eklenen artık eşlik denetim denklemleriyle birlikte yeni bir sonuç bulmasına dayanmaktadır.

Standart LP kod çözücüsüne göre daha az karmaşıklık içeren ve artık eşlik denetim denklemleri kullanarak standart LP kod çözücüsüne göre daha başarılı olan bir ALP kod çözücüsü [4] çalışmasında tasarlamıştır. Çizge içerisinde çevrim oluşturabilen kesirli denetim düğümlerini toplayarak artık eşlik denetim denklemlerini elde etmeyi amaçlayan bu çalışmada çevrim oluşturabilen kesirli denetim düğümleri rastgele bir yöntem ile aranmış ve bu işlem sonunda elde edilen artık eşlik denetim eşitliğinin en son bulunan çözümü olursuz çözüm hale getirebileceği garantisi verilememiştir. 2012 yılında, aynı yazarlar

Bu bildirideki çalışmalar Avrupa Birligi PIRG07-6A-2010-268264 nolu ve TÜBİTAK 113M499 nolu projeler kapsamında desteklenmiştir

tarafından, bulunan kesirli sonucu olursuz çözüm haline getirebilecek artık eşlik denetim denklemlerinin bazı koşulları sağlaması gerektiği gösterilmiş ve buna dayalı olarak tasarlanan ALP kod çözücünün yüksek başarıma sahip olacağı gösterilmiştir [5]. Ayırma algoritması ve eşlik denetim matrisi üzerinde temel satır-sütün işlemleri yaparak standart LP kod çözücülerin hata başarımlarını geliştiren bir teknik ise [6] çalışmasında önerilmiştir.

Bu çalışmamızda, [4] ve [5] çalışmalarında ele alınan rastgele artık eşlik denetim denklemlerini oluşturma aşamaları ele alınmış ve tam sayı programlama eniyileme yöntemi kullanılarak geliştirilmiştir. ALP kod çözücünün artık eşlik denetim denklemleri sayesinde ML kod çözücülerin hata başarımına yaklaşabileceği gösterilmiştir. Diğer çalışmaların aksine, hiçbir buluşsal yöntem kullanılmadan artık eşlik denetim denklemleri oluşturulmuş ve sadece aktif olan çözümü olursuz çözüm haline getiren kısıt eklenmiştir. Tam sayı programlama kullanıldığı için zaman başarımı olarak diğer ALP çözücülerin gerisinde kalmasına rağmen, aktif çözümü olursuz çözüm haline getirebilecek kısıtın (eğer böyle bir kısıt varsa) bulunması garantiye alınmıştır.

Bildirinin geri kalanı şu şekilde düzenlenmiştir: Bölüm 2'de LDPC kodlara ait tanımlamalar verilmiş ve kod çözücüleri anlatılmıştır. Bölüm 3'te uyarlanabilir kod çözücüsünden söz edilmiş ve artık eşlik denetim eşitliği arayan tam sayı programlama modeli verilmiştir. Bölüm 4'de sayısal sonuçlar gösterilmiş ve Bölüm 5'de sonuçlar ve gerçekleştirilebilecek gelecek çalışmalardan bahsedilmiştir.

II. DOĞRUSAL PROGRAMLAMA ILE KOD ÇÖZME

(n, k) boyutlu LDPC kodu ζ, n bit düğümü ve $m = n - k$ denetim düğümünden oluşmaktadır. Bit düğümleri çözülecek kod sözcüğüne, denetim düğümleri ise eşlik denetim matrisinin oluşturduğu eşlik denetim denklemlerine karşılık gelmektedir. Bir denetim düğümünde bağlı olan bit düğümü onun komşusu olarak adlandırılır ve bu tanımın tersi de geçerlidir.

Bir LDPC kodu için ML kod çözücüsü, gelen $\mathbf{r}$ vektörü için

$$\text{enazla } \mu^T \mathbf{x} \tag{1}$$

$$\text{kısıtlar: } \mathbf{x} \in \zeta \tag{2}$$

biçiminde ifade edilebilir. Burada kullanılacak amaç fonksiyonun katsayıları

$$\mu_i = log(\tfrac{Pr(r_i|x_i=0)}{Pr(r_i|x_i=1)}) \tag{3}$$

ile verilir.

ζ koduna ait olan kod sözcüklerinin dışbükey zarfı kod sözcüğü politopunu oluşturur. Kod sözcüğü politopu

$$poly(\zeta) = \left\{ \sum_{y \in \zeta} \lambda_y y : \lambda_y \geq 0, \sum_{y \in \zeta} \lambda_y = 1 \right\} \tag{4}$$

biçiminde ifade edilebilir. Kod sözcüğü politopunun köşe noktalarının ζ koduna ait olan bütün kod sözcüklerini oluşturması doğrusal programlamanın kullanılabilmesini sağlar. Doğrusal programlamanın bulacağı eniyi çözüm, kod sözcüğü politopunun bir köşe noktasını verecektir [7]. Bundan dolayı, doğrusal programlama ile ML kod çözücüsü tanımlanabilir. Kod sözcüğü politopundaki her bir noktayı $\mathbf{g}$ vektörü ile gösterirsek, vektörün bileşenleri $g_i = \sum_y \lambda_y y_i$ ile ifade edilebilir. Doğrusal programlama eniyileme yöntemli ile kod sözcüğü politopunu kullanılarak ML kod sözcüğü

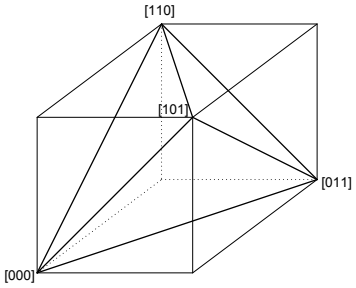
$$\text{enazla } \sum_{i=1}^{n} \mu_i g_i \tag{5}$$

$$\text{kısıtlar: } \mathbf{g} \in poly(\zeta) \tag{6}$$

ile bulunabilir. Bu doğrusal programlama modelinin en büyük sorunu kod sözcüğü sayısının pratikte kullanılan kodlar için genellikle çok yüksek sayılarda olmasıdır. Kod sözcüğü politopunu doğrusal kısıtlar ile ifade etmek kısıt sayısının çok fazla olması sebebiyle genellikle mümkün olmamaktadır. Alternatif bir çözüm yolu olarak Feldman gevşetilmiş bir politop önermiştir [3]. Her bir denetim düğümü için yerel kod sözcüklerinin dışbükey zarfı olan yerel bir politop oluşturabilir. LDPC kodların tanımından dolayı yerel kod sözcüklerinin boyutu küçük ve sayısı az olacaktır. Bunun neticesi olarak ise yerel politopları az sayıdaki doğrusal kısıtlarla göstermek mümkündür. Feldman, yerel politopların kesişimi ile temel politop ismini verdiği bir politop üretmiştir. Kod sözcükleri yerel kod sözcüklerinin kesişiminde bulunduğu için elde edilen temel politop kod sözcüklerinin tamamını kapsayacaktır. Bir denetim düğümü için oluşturulacak politop, yerel kod sözcüklerin dışbükey zarfı, yerel kod sözcüğü oluşturmayan tek ağırlıklı vektörleri yasaklayarak da oluşturulabilir. Yasaklanan bu vektörler

$$\sum_{x_i \in S} x_i - \sum_{x_i \in (N(c_j) \setminus S)} x_i \leq |S| - 1 \;\; \forall S \subseteq N(c_j), |S| \, tek \tag{7}$$

biçiminde tanımlanmıştır. Burada S, bir denetim düğümüne ait olan tek ağırlıklı tüm vektörlerin oluşturduğu kümeyi, $N(.)$ operatörü ise verilen bir düğümün komşularını içeren kümeyi göstermektedir. Üç komşulu bir denetim düğümü için elde edilecek politop Şekil 1'de verilmiştir. Temel politop, kod sözcüklerinin dışbükey zarfı olan politopu da kapsayan gevşetilmiş bir politopdur. Temel politopun köşeleri kod sözcüklerini içerdiği gibi kesirli sayılar içeren sözde kod sözcüklerini de içerir. Bu nedenle, temel politopu kullanarak modellenen doğrusal programlama, eniyi çözüm olarak tam sayılar içeren bir sonuç bulduğu zaman sonuç ML kod çözücüsü ile aynı sonucu vermektedir. Diğer yandan, eğer sonuç kesirli bir sayı içerirse LP kod çözücüsü hatalı bir sonuca (sözde kod sözcüğü) ulaştığını kabul eder. Sonuç olarak, temel politopu kullanan LP kod çözücüsü, ML kod çözücüsü başarımına sahip değildir, fakat ML özelliğine sahiptir. Sadece temel politopu kullanan LP kod çözücüsünün



Şekil 1: Yerel politop örneği.

pratikte uygulanabilirliği olduğu için bu kod çözücüsüne standart LP kod çözücüsü denmektedir.

III. Uyarlanabilir doğrusal programlama ile kod çözme

Standart LP kod çözücü kesirli bir çözüm bulduğu zaman, bu çözümü olursuz çözüm haline getirebilecek bir kısıt aranabilir. Kesirli çözümleri eleyerek standart LP kod çözücüsünün hata başarımı artırılabilir ve ML kod çözcülerin hata başarımına yakınsanabilir.

LDPC kodların doğrusal bir kod türü olmasından dolayı denetim düğümlerinin toplanması ile oluşturulan yeni bir denetim düğümü, bütün kod sözcükleri tarafından sağlanacaktır. Diğer yandan, bu denetim düğümünün oluşturduğu yerel politop standart LP kod çözücüsünün bulmuş olduğu kesirli çözümü içermeyebilir (sözde kod sözcükleri bu yeni eşlik denetim denklemini sağlamak zorunda değillerdir). Bu durumda, elde edilen denetim düğümü, çözümü olursuz hale getirebilecek bir kısıt içermektedir. Yerel politopu oluşturan ve (7) ile ifade edilen kısıt

$$g_j(x) = \sum_{x_i \in S} (1 - x_i) + \sum_{x_i \in (N(c_j) \setminus S)} x_i \geq 1 \qquad (8)$$

biçiminde de ifade edilebilir. Eğer herhangi bir tek küme için (8) sağlanmıyorsa, bu denklem, aktif olan çözümü olursuz çözüm haline getirecektir. Diğer bir ifade ile, $g_j(x)$ değeri 1'in altında olduğu zaman aranan kısıt bulunmuş olacaktır.

Önerme 1: Herhangi bir denetim düğümü için $g(x)$ değerinin alabileceği en küçük değerin bir alt sınırı kolaylıkla bulunabilir. İlgili denetim düğümünün komşuluğundaki 0.5 değerinden büyük olan bütün bit düğümleri, S kümesine ait olduğunda ve 0.5 değerinden küçük olan bütün bit düğümleri S kümesine ait olmadığında bir alt sınır bulunmuş olur. Eğer elde edilen S kümesi tek ise bulunan sonuç $g(x)$'in en küçük değeridir.

İspat 1: x_i bit düğümünün değeri 0.5'den büyük olduğu zaman $1 - x_i$ değeri x_i değerinden küçük olacaktır. Bu durumda g(x) değerinin enazlanması için (8)'e göre x_i bit düğümünün S kümesine ait olması gerekmektedir. x_i bit düğümünün değeri 0.5'den küçük olduğunda ise x_i değeri $1 - x_i$ değerinden küçük olacaktır. Bu şartlar altında da $g(x)$ değerinin enazlanması için x_i bit düğümünün $N(c_j) \setminus S$ kümesine ait olması gerektiği gözükmektedir.

Önerme 1 herhangi bir denetim düğümünün alt sınırının sadece denetim düğümünün komşu olduğu bit düğümlerinin değerlerinin bilinmesi ile bulunabileceğini göstermektedir. Bunun ötesinde, denetim düğümünün komşusu olan ve değeri 0.5'den büyük olan bit düğümlerinin sayısı tek ise bu alt sınırın $g(x)$'in alabileceği en küçük değere eşit olacağını göstermektedir. Böylelikle, alt sınır 1'den küçük olan ve alt sınırla ilgili olan S kümesini tek sayılı elemanlı olan bir denetim düğümü aktif olan kesirli çözümü olursuz hale getirecektir. Bunlar göz önüne alınarak, $g_j(x)$ değerini enazlayarak en küçük $g_j(x)$ değeri 1'den küçük olan bir c_j denetim düğümü arayan bir eniyileme modeli geliştirilmiştir. Geliştirilen tam sayı eniyileme modeli küçük bir örnek kullanılarak açıklanabilir. Öncelikle, standart LP kod çözücüsünün eniyi çözüm olarak kesirli değerlere sahip

$$\mathbf{x} = [0.9 \quad 1 \quad 0.12 \quad \ldots \quad 0 \quad 0.60] \qquad (9)$$

vektörünü bulduğunu varsayalım. Önerme 1 ışığında her bir bit düğümünün, x_i, g(x) fonksiyonun alt sınırı için yaptığı katkıyı

$$\widehat{x_i} = 0.5 - |(x_i - 0.5)|, \qquad i = 1, 2, ..., n \qquad (10)$$

ile bulabiliriz. Burada, x_i 0.5 değerinden büyük olduğunda $\widehat{x_i} = 1 - x_i$ ve 0.5 değerinden küçük olduğunda ise $\widehat{x_i} = x_i$ olmasını sağlar. Bu durumda örneğimiz için $\widehat{\mathbf{x}}$,

$$\widehat{\mathbf{x}} = [0.1 \quad 0 \quad 0.12 \quad \ldots \quad 0 \quad 0.40] \qquad (11)$$

olarak elde edilir. Değeri 0.5'den büyük olan bit düğümlerinin pozisyonunu gösteren bir p vektörü aşağıdaki gibi tanımlanmıştır:

$$\mathbf{p} = [1 \quad 1 \quad 0 \quad \ldots \quad 0 \quad 1] \qquad (12)$$

Elde edilen bu parametreler ışığında tam sayı programlama modeli aşağıdaki gibi geliştirilmiştir:

$$\text{enazla} \sum_{i=1}^{n} \omega_i \widehat{x_i} \qquad (13)$$

$$\text{kısıtlar:} \sum_{j=1}^{m} \alpha_j H_{j,i} + \omega_i = 2k_i \qquad i = 1, 2, ..., n. \qquad (14)$$

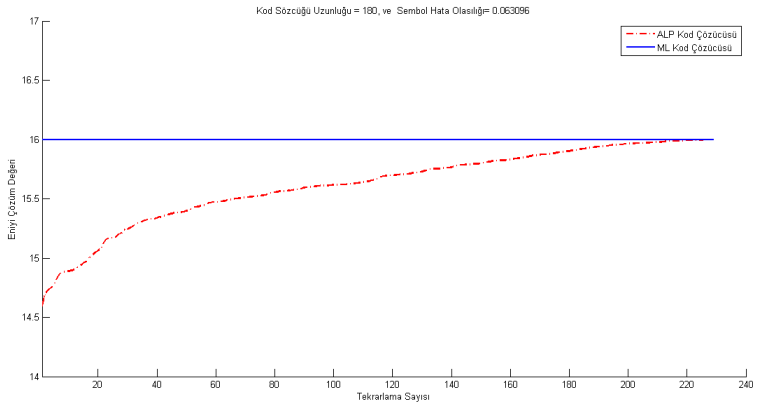
$$\sum_{i=1}^{n} \omega_i p_i + 1 = 2\eta \qquad (15)$$

$$\sum_{j=1}^{m} \alpha_j \geq 2. \qquad (16)$$

$$k_i, \eta \in \mathbb{Z} \qquad (17)$$

$$\omega_i, \alpha_j \in \{0, 1\} \qquad (18)$$

ω_i parametresi i. bit düğümünün oluşturulan artık eşlik denetim düğümünün komşusu olup olmadığını



Şekil 2: Uyarlanabilir doğrusal programlama.

kontrol eder. Kısıt (14) yardımıyla ω_i'nin i. bit düğümü denetim düğümünün komşusu olduğu zaman 1 değerini alması, aksi takdirde 0 değerini alması sağlanmış olur. Bu durumda da amaç fonksiyonu ω değeri ile bit düğümlerinin $g(x)$ alt sınırına yaptıkları katkının çarpılması ile elde edilir. α_j değeri ise j. denetim düğümünün üretilen artık denetim düğümünün oluşumunda rol alıp almamasını belirler. Eğer α_j değeri 1 alırsa j. denetim düğümü yeni oluşturulan denetim düğümü için toplamaya girecek denetim düğümlerinden biri olduğunu göstermektedir. Kısıt (15) ile de artık denetim düğümünün alt sınırını belirleyen kümenin eleman sayısının tek olması sağlanır. Son olarak, Kısıt (16) yardımıyla en az iki denetim düğümünün toplanması sağlanır.

IV. Sayısal sonuçlar ve tartışma

Bu çalışmada, artık eşlik denetim düğümünü üreten tam sayı programlama modeli ve standart LP kod çözücüsü CPLEX 12.4 yazılımıyla çözülmüştür. Standart LP kod çözücüsünün ürettiği sözde kod sözcüklerini eleyen artık denetim düğümleri, geliştirilen tam sayı programlama modeli sayesinde bulunmuştur. Standart LP kod çözücüsü tam sayı bir sonuç (geçerli bir kod sözcüğü) bulana kadar yeni artık denetim düğümlerinin üretilmesine devam edilmiştir.

Bilgisayar benzetimlerinde (180,90) boyutlarında bir rastgele LDPC kodu üretilmiştir. İkili simetrik kanalda yapılan benzetimlerde ML kod çözücüsü ile önerilen ALP kod çözücüsünün hata başarımı rastgele bir kod sözcüğü için karşılaştırılmış ve sonuçlar Şekil 2'de sunulmuştur. ML kod çözücüsünün bulmuş olduğu eniyi çözüm değerine ALP kod çözücüsünün üretilen artık denetim düğümleri sayesinde belli miktarda bir yinelemeden sonra yakınsadığı gözlenmiştir.

Bir tam sayı programlama problemlerini çözmek NP-zordur. Bu yaklaşımda ise geçerli bir kod sözcüğü bulunana kadar belirsiz sayıda tam sayı programlama problemlerinin çözülmesi gerekmektedir. Şekil 2' deki problem için yaklaşık 230 tane tam sayı programlama modeli çözülmüştür. Fakat, amaç ML performanına yaklaşmak olduğu için işlemsel karmaşıklığı çok düşük olamamaktadır, ancak bu yöndeki çalışmalar sürmektedir. Tam sayı programlama problemleri, dal-sınır ve dal-kesi gibi algoritmalar kullanılarak pratikte büyük problemler için çözülebilmektedir [8].

V. Sonuçlar

ALP kod çözücülerin hata başarımının artık denetim düğümleri üreterek ML kod çözücülerin hata başarımına yakınsayabildiği gösterilmiştir. Artık denetim düğümlerinin üretilmesi için bir tam sayı programlama modeli geliştirilmiştir. Gelecek çalışmalarda tam sayı programlama modeli ile belirtilen problemin eniyi olmayan algoritmalar geliştirilerek daha az karmaşıklıkla bulunması amaçlanmaktadır.

Kaynakça

[1] Gallager, R., "Low-density parity-check codes," *IRE Transactions on Information Theory*, cilt 8, no. 1, ss. 21–28, Ocak 1962.

[2] McEliece, R., MacKay, D. J. C., ve Cheng, J.-F., "Turbo decoding as an instance of Pearl's ldquo;belief propagation rdquo; algorithm," *IEEE Journal on Selected Areas in Communications*, cilt 16, no. 2, ss. 140–152, 1998.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Schrijver, A., *Theory of Linear and Integer Programming*. New York, NY, USA: John Wiley & Sons, Inc., 1986.

[8] Wolsey, L. A., *Integer Programming*. New York, NY: Wiley-Interscience, 1998.