LINEAR PROGRAMMING CALCULATOR APP

★ ★ ★ ★ ★

Bu uygulama ile Lineer Programlama Problemlerini Adım Adım Çöz, Zamanından Tasarruf Et!

Uygulama içerisinde Simpleks ile BigM Metotlarının çözüm yöntemleri mevcuttur. Çözmek istediğiniz eşitsizlik yapısındaki bir lineer programlama modelini bu metotlar ile çözebilirsiniz. Özel algoritmalar ile oluşturulan bu uygulama sayesinde 100x100 değişkenli ve kısıtlı problemleri bile çözebilirsiniz.

Simpleks Metot: Genel amaç, en iyi çözüme ulaşmak için ardışık tekrarlama yapmaktır. Problemin kısıtları ve pozitif olma şartıyla beraber başlangıç simpleks tablosu oluşturulur. Oluşturulan tablı ile çözüm bölgesinin bir uç noktasından başlayarak iterasyonlarla belirli bir düzen içerisinde amaç fonksiyonu (maksimum ya da minimum olabilir) yapan değişken değerlerine ulaşıncaya kadar iterasyon devam eder. Bu metotla sadece "≤" yapısındaki denklem sistemleri çözülür.

BigM Metot: Bu çözüm yöntemini eğer kısıtlar "=" veya "≥" durumlarını içeriyorsa, bu durumda M tekniğini kullanırız. Kısıtlar eşitlik durumunda olduğunda dolgu değişkeni eklenmez. Eğer büyük eşittir durumunda ise fazla değişken çıkarılır. Ancak bu durumda bir başlangıç tabanı oluşmaz. Bu durumda M tekniği devreye girer. Eşitlik ve büyük eşittir içeren kısıtlara ilave olarak yapay değişkenler eklenir. Bu yapay değişkenlerin değeri çok büyük bir sayıdır. Amaç fonksiyonunda ise değerleri eğer minimum fonksiyon ise katsayısı 1, maksimum fonksiyon ise katsayısı -1 olarak belirlenir. Temeli simpleks metot ile aynıdır. Aynı adımlar izlenir. İterasyon sonuna kadar yapay değişkenlerin tabandan çıkması gerekir. Aksi takdirde problemin çözümü yoktur.

ÜCRETSİZ İNDİR

İNDİR
GOOGLE PLAY

$$C^T x$$
$$Ax \leq b$$
$$x \geq 0$$

Bir lineer amaç fonksiyonunun eşitsizlik veya eşitlik kısıtlamalarını sağlayacak şekilde optimizasyon (maksimum veya minimum) yapılmasıdır. Bir lineer programlama modeli matris formunda Yukarıdaki gibidir.

Mevcut algoritmalar üzerinde bazı işlemlerin fazladan olduğu tespit edilmiştir. Örneğin, çözüm aşamasında tabana girecek vektörü belirlemeden önce tabandan çıkacak olan bir vektör var mı yok mu bunun kontrolü yapılmamaktadır. Dolayısıyla, C_j-Z_j değerlerinin kontrolü fazladan yapılmaktadır. Öncelikle tabandan çıkacak vektör yoksa tabana girecek olan vektörün önemi yoktur. Dolayısıyla bu durum uygulamanın zaman karmaşıklığının yükselmesine yol açacaktır. Bu sebepten ötürü, uygulamada tabandan çıkacak vektörün var olup olmadığına öncelik verilmiştir.

Bir diğer önemli nokta ise C_j-Z_j değerleri ve Z değeri hesaplanırken, vektörler sırayla birbirleri ile çarpılıp çıkartılmaktadır. Bu durum zaman karmaşıklığını pek etkilemese bile işlemci üzerindeki işlem hacmini arttıracaktır. Bu sorunu çözmek için ise, amaç fonksiyonundaki Z değerini değişkenlerle birlikte yazıp 0'a eşitlemektir. Böylece amaç fonksiyonundaki Z değerini de taban içerisinde kabul edip, gerçekleştirdiğimiz satır sütun işlemlerini aynı şekilde tabanda olan Z değeri içinde gerçekleştirirsek bu fazla işlem yükünden kurtulmuş oluruz. Ancak, burada unutulmamalıdır ki Z değeri hiçbir zaman tabandan çıkamaz. Z'nin bulunduğu satır tabana girecek olan vektörleri belirler.

- Adım Adım Çözüm
- Kullanıcı Dostu Tasarım
- Simpleks Method ve BigM Method
- Maksimum 100x100 Değişkenli ve Kısıtlı Problemler
  Maksimum sayıda değişken ve kısıtla hesaplama cihazınızın özelliklerine göre değişebilir.

BARIŞ USLUCAN - 17025501
TEZ DANIŞMANI: DOÇ. DR. MUTLU AKAR