67

Binary Quadratic Programming Dengan Algoritma *Branch and Bound*

Arrival Rince Putri

Jurusan Matematika, FMIPA, Universitas Andalas
e-mail : arrival@fmipa.unand.ac.id

ABSTRAK

***Quadratic Integer Programming* (QIP) adalah pemrograman nonlinier yang fungsi tujuan nonlinier dengan variabel-variabel integer. Sekilas, *quadratic integer programming* kelihatan sama dengan pemrograman kuadratik biasa. Akan tetapi, satu perbedaan yang penting adalah variabel-variabel optimisasi dari QIP harus integer. Ketika variabel-variabel integer dibatasi oleh 0 dan 1, maka masalah ini disebut dengan *Binary Quadratic Programming* (BQP). Dalam tulisan ini untuk menyelesaikan masalah BQP digunakan algoritma *Branch and Bound*.**

Kata kunci : *quadratic integer programming, binary quadratic programming,* **algoritma** *Branch and Bound*

PENDAHULUAN

Program kuadratik [1] adalah suatu masalah nonlinear fungsi tujuan kuadratik dan kendala-kendala linear. Program kuadratik secara umum dapat ditulis sebagai :

$$\begin{aligned} \min \quad & f(x) = \frac{1}{2}x^T Qx + cx \\ \text{s.t} \quad & Ax \le b \\ & x \ge 0 \end{aligned} \qquad (1)$$

dimana x adalah vektor nx1, Q adalah matriks simetris nxn yang menggambarkan koefisien-koefisien dari suku-suku kuadratik, dan c adalah vektor baris berdimensi-n yang menggambarkan koefisien-koefisien dari suku-suku liner dalam fungsi tujuan. A adalah matriks konstan mxn dan b adalah vektor kolom berdimensi-m .

Pemrograman kuadratik integer [2] adalah suatu program nonlinear dimana fungsi tujuannya nonlinear dan variabel-variabelnya integer. Sekilas, pemrograman kuadratik integer kelihatan sama dengan masalah pemrograman kuadratik yang digambarkan dalam (1). Akan tetapi, satu perbedaan yang penting adalah variabel-variabel optimisasi dari pemrograman kuadratik integer harus integer. Ketika variabel-variabel integer dibatasi oleh 0 dan 1, maka masalah ini disebut dengan masalah *Binary Quadratic Programming* (BQP).

BINARY QUADRATIC PROGRAMMING (BQP)

Secara umum masalah *Binary Quadratic Programming* (BQP) tanpa kendala-kendala dapat ditulis sebagai :

$$\begin{array}{ll} \min & f(x) = \frac{1}{2} x^T Q x + cx \\ \text{(BQP)} & x \in \{0,1\} \end{array} \quad (2)$$

dimana x adalah vektor nx1. Q adalah matriks simetris nxn yang menggambarkan koefisien-koefisien dari suku-suku kuadratik, dan c adalah vektor baris berdimensi-n yang menggambarkan koefisien-koefisien dari suku-suku liner dalam fungsi tujuan.

Terdapat suatu hubungan program kuadratik yang disebut dengan *quadratic relaxation* (QR) yang dibuat dengan merelaksasikan kendala-kendala integer terhadap kendala-kendala interval :

$$\begin{array}{ll} \min & f(x) = \frac{1}{2} x^T Q x + cx \\ \text{(QR)} & x \in [0,1] \end{array} \quad (3)$$

Hubungan antara BQP dengan QR dapat dilihat dari sifat-sifat berikut :

1. Nilai tujuan optimal QR lebih kecil atau sama dengan nilai tujuan optimal BQP.
2. Jika QR tidak feasibel, maka BQP juga tidak feasibel.
3. Jika QR dioptimisasikan dengan variabel-variabel integer, maka solusi tersebut adalah feasibel dan optimal untuk BQP.

Jadi, penyelesaian QR memberikan beberapa informasi, yaitu batas bawah untuk nilai optimal dari masalah BQP. Sifat-sifat ini digunakan untuk menyelesaikan masalah BQP dengan algoritma *Branch and Bound*.

ALGORITMA BRANCH AND BOUND

Algoritma *Branch and Bound* adalah suatu pendekatan yang dibangun untuk menyelesaikan masalah optimisasi diskrit dan kombinatorik. Masalah optimisasi diskrit adalah masalah dimana variabel-variabel keputusan dianggap bernilai diskrit dari suatu himpunan tertentu, ketika himpunan ini adalah himpunan bilangan integer maka dapat dikatakan sebagai masalah pemrograman integer. Masalah optimisasi kombinatorik adalah memilih kombinasi terbaik dari semua kombinasi yang mungkin. Sebagian besar masalah kombinatorik dapat diformulasikan sebagai program integer.

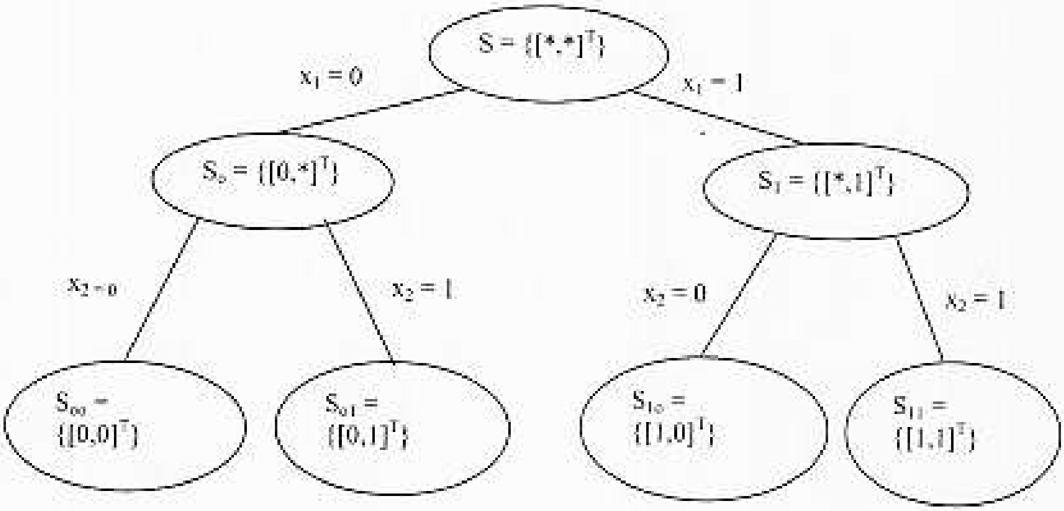
Kesulitan utama dari masalah-masalah tersebut adalah bahwa kita tidak mempunyai suatu kondisi yang optimal untuk memeriksa apakah solusi feasibel yang diberikan optimal atau tidak. Tidak ada semacam kondisi keoptimalan global untuk masalah optimisasi diskrit dan kombinarik. Sehingga untuk menjamin keoptimalan solusi feasibel yang diberikan, kita bandingkan dengan setiap solusi feasibel yang lainnya.

Kebanyakan yang dilakukan untuk menghitung solusi optimal adalah dengan mengenumerasi semua konbinasi yang mungkin dari variabel-variabel.Akan tetapi beban perhitungan untuk pendekatan ini akan menjadi besar sekali. Kesimpulannya adalah diperlukan suatu algoritma yang dapat menentukan solusi optimal tanpa mengenumerasi semua kombinasi yang mungkin dari variabel-variabel. Algoritma ini dinamakan *branch and bound* yang cukup mengenumerasi secara eksplisit hanya beberapa dari kombinasi-kombinasi yang mungkin dengan menggunakan *branching* dan *bounding* [3].

1. Branching

Branching digunakan untuk membagi beberapa daerah feasibel ke dalam beberapa subdaerah feasibel yang lebih kecil. Jika terdapat suatu himpunan feasibel S, maka kita dapat mencabangkan S ke dalam k himpunan-himpunan yang lebih kecil, sehingga : $S = \bigcup_{i=1}^{k} S_i$ (4)

Karena pembagian adalah rekursif berulang dalam masing-masing subdaerah, dan semua subdaerah membentuk suatu struktur pohon, maka hal ini disebut *search* *tree* atau *branch and bound tree*. Contoh dari *binary search tree* dapat dilihat pada gambar dibawah ini [3].



Gambar 1. Binary Search Tree

2. Bounding

Bounding digunakan untuk menghitung batas bawah dan batas atas dari fungsi tujuan optimal untuk submasalah dalam cabang. Jika solusi parsial dari subdaerah lebih besar atau sama dengan batas atas, maka pencarian solusi dari masalah dihentikan. Seringkali, batas-batas dapat digunakan untuk menghentikan subpohon-subpohon berikutnya. Terdapat tiga kondisi yang berbeda untuk proses pencabangan subpohon [3]:

- Tidak feasibel (penyelesaian dari subpersoalan tidak feasibel).
- Keoptimalan (penyelesaian dari subpersoalan memberikan penyelesaian optimal).
- Dominan (nilai Z optimal untuk subpersoalan tidak lebih baik dari nilai Z optimal subpersoalan lain.

Dalam metode *Branch and Bound*, terdapat beberapa parameter dan pilihan yang mungkin berpengaruh secara drastic terhadap performansi. Dua parameter yang penting adalah memilih cabang berikutnya dan memilih variabel pencabangan.

ALGORITMA BRANCH AND BOUND

Algoritma branch and bound untuk masalah *binary quadratic programming*, yaitu :

Langkah 1 : Inisialisasi

a. Modelkan persoalan ke dalam masalah BQP dan misalkan L adalah himpunan solusinya.
b. Tetapkan batas atas dan batas bawah sebagai $\underline{z} := -\infty$ dan $\bar{z} := +\infty$.

Langkah 2 : Uji penghentian

Jika L = { }, maka hentikan algoritma. Solusi x yang mempunyai nilai optimal $z = \frac{1}{2}x^T Qx + cx$ sudah optimal.

Langkah 3 : Menyeleksi dan merelaksasi masalah

Pilih dan hapus masalah BQP dari L.

Selesaikan relaksasinya.

Misalkan z^i menjadi nilai dari relaksasi dan misalkan x^i menjadi solusi dari relaksasi.

Jika $\underline{z} := -\infty$, maka $\underline{z}$ harus diperbaharui terhadap z^i.

Langkah 4 : Penghentian

a. Jika $z^i \geq \bar{z}$, kembali ke langkah 2.
b. Jika $x^i \in \{0,1\}$ dan $z^i < \bar{z}$, maka $\bar{z}$ harus diperbaharui terhadap z^i dan kembali ke langkah 2.

Selainnya lanjutkan ke langkah 5.

Langkah 5 : Pencabangan

Misalkan $\{S_{ij}\}_{j=1}^{k}$ adalah bagian dari S_i.

Tambahkan masalah ke L, kemudian kembali ke langkah 2.

dimana : z^i = nilai fungsi tujuan dari sub masalah

x^i = solusi feasibel dari z^i

CONTOH PERMASALAHAN

Diberikan contoh masalah BQP sebagai berikut :

$$\min \quad f(x) = \frac{1}{2}x^T Qx + cx$$

$$\text{(BQP)} \quad x_i \in \{0,1\},\ i = 1, 2, 3$$

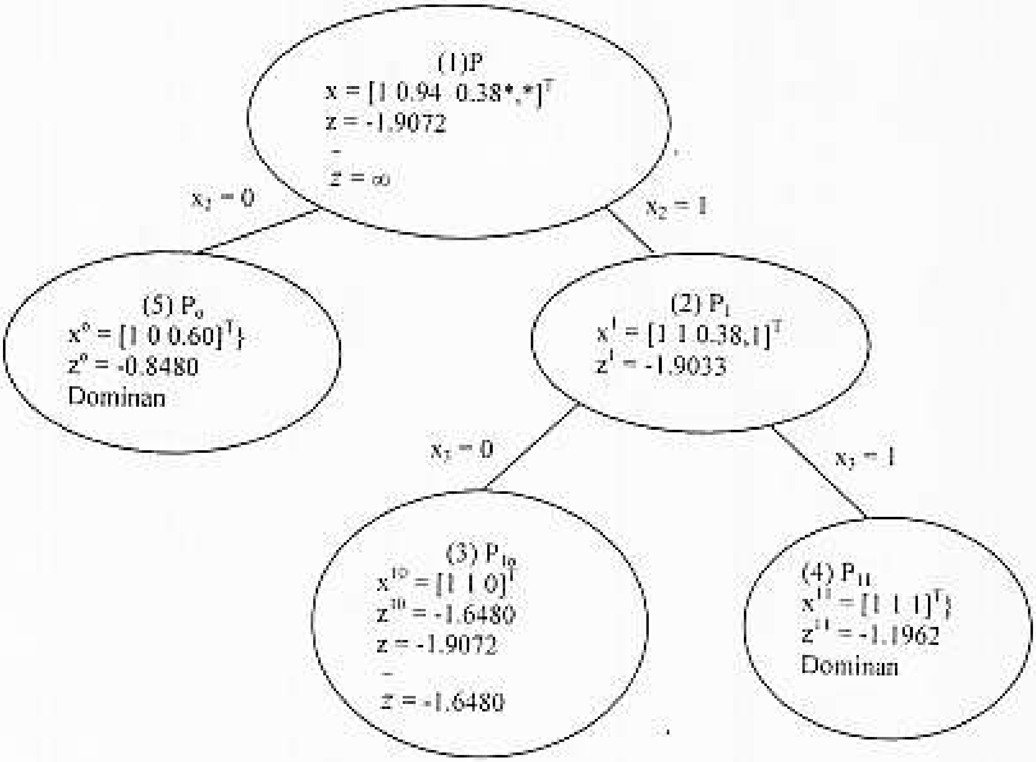
Dimana : $Q = \begin{bmatrix} 1.1507 & -0.7427 & 0.9244 \\ -0.7427 & 2.5649 & 0.8078 \\ 0.9244 & 0.8078 & 3.6248 \end{bmatrix}$ dan c = [-0,7747 -1.9884 -3.0926]

Solusi dari permasalahan di atas adalah x = $[0.9355 \quad 0.9169 \quad 0.4103]^T$.

Terdapat suatu hubungan program kuadratik yang disebut dengan *quadratic relaxation* (QR) yang dibuat dengan merelaksasikan kendala-kendala integer terhadap kendala-kendala interval :

$$\min \quad f(x) = \frac{1}{2}x^T Qx + cx$$

$$\text{(QR)} \quad x \in [0,1]$$



Gambar 2. Pohon Branch and Bound

KESIMPULAN

Algoritma *branch and bound* cukup mengenumerasi secara eksplisit hanya beberapa dari kombinasi-kombinasi yang mungkin dengan menggunakan *branching* dan *bounding*. Solusi relaksasi dari permasalahan BQP pada contoh yang dibahas adalah x = [1 1 0] dengan nilai optimal Z = -1.6480 dan dapat dilihat pada gambar 2.

TINJAUAN PUSTAKA

[1] Wayne L. Winston. 2004. Operation Research : Applications and Algorithms. Brooks/Cole.

[2] Daniel Axehill. 2005. Applications of Integer Quadratic Programming in Control and Communication. Linkoping Universitet.

[3] Laurence A. Wolsey. 1988. Integer and Combinatorial Optimization. John Wiley & Sons. Inc