Penggunaan Strategi Algoritma Branch and Bound Dalam Bidang Industri dan Bisnis dengan Integer Programming

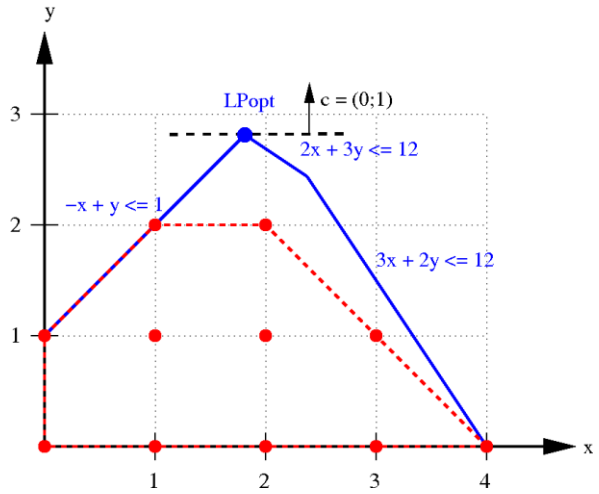
Muhamad Nobel Fauzan / 13517042
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
mn.fauzan07@gmail.com

***Abstract*—Integer programming merupakan salah satu metode penyelesaian yang sangat dibutuhkkan dalam berbagai bidang, dalam bidang Industri dan Bisnis, Integer Programming sangat sering kita gunakan untuk mendapatkan keputusan yang sangat terbaik.**

Kata Kunci—Branch and Bound; Integer Programming

I. PENDAHULUAN

Di dalam kehidupan, banyak sekali skenario yang bisa bermunculan di masa depan. Tentunya kita sebagai makhluk normal tidak bisa mengetahui kejadian apa yang akan terjadi, namun tentu saja itu bukanlah alasan manusia untuk pasrah saja dalam kehidupannya, manusia memiliki kemampuan berpikir yang sangat hebat dibandingkan dengan makhluk hidup di bumi, dengan kemampuan tersebut manusia memang tidak bisa mengetahui masa depan, tetapi tentu saja manusia bisa mengira-ngira apa yang akan terjadi di masa depan. Perkiraan manusia beragam jenisnya, bisa buruk dan juga baik, oleh karena itu manusia membutuhkan kemampuan untuk menghindari kemungkinan buruk tersebut untuk terjadi, hal tersebut dilakukan manusia dengan cara mengambil langkah yang benar dalam mengambil keputusan, sehingga manusia akan mendekat ke arah yang kemungkinannya lebih baik. Berbagai hasil pemikiran manusia hasilkan untuk dapat mengambil keputusan yang benar, dua topik yang sangat terkenal adalah mengenai *Branch and Bound* dan juga *Integer Programming.*



Gambar 1 : Ilustrasi Integer Programming

Sumber :
https://upload.wikimedia.org/wikipedia/commons/f/fd/IP_polytope_with_LP_relaxation.png

Branch and Bound (B&B) merupakan suatu algoritma memecahkan masalah tahap pertahap dengan memanfaatkan pohon pencarian, pohon pencarian tersebut melambangkan alur skenario/keadaan yang mungkin terjadi, simpul pada pohon tersebut melambangkan keadaan yang dimaksud. Seseorang akan memilih simpul-simpul untuk mencapai tujuan dengan menggunakan skema yang mirip dengan skema BFS, yaitu mengutamakan pencarian secara untuk semua simpul pada kedalaman yang sama terlebih dahulu (pencarian menyamping), namun bedanya adalah urutan simpul yang dipilih/dibangkitkan ditentukan oleh suatu atribut *cost* yang kita berikan untuk setiap simpul, atribut *cost* tidak selalu sama untuk setiap pihak yang menyelesaikan soal, karena nilai *cost* didefinisikan oleh pihak itu sendiri

Contoh sederhana untuk penggunaan algoritma *Branch and Bound* adalah saat kita ingin memilih beberapa pilihan barang yang akan kita beli. Misalnya kita ingin membeli laptop A dan laptop B, laptop A memiliki harga 2 kali lipat dari harga laptop B, dan juga memiliki kemampuan 2 kali lipat lebih besar dari

pada laptop B (anggap semua aspek laptop bisa dinilai dengan atribut ‘kemampuan’), maka terdapat dua faktor yang bisa menentukan *cost* dalam memilih kedua barang tersebut, yaitu harga dan kemampuan. Kita bisa saja menganggap *cost* akan lebih besar jika kita memilih laptop B karena pada saat itu kita diharuskan memiliki laptop yang memiliki kemampuan besar, dan bisa saja kita anggap laptop A memiliki *cost* yang lebih besar, karena harga lebih kita anggap penting dibanding kualitas, hal itu yang dimaksudkan pada paragraf sebelumnya.

Integer programming merupakan suatu persoalan optimasi atau kelayakan dalam memilih nilai berbagai variabel yang batasi sebagai bilangan bulat atau *integer*, beberapa contoh penggunaan *Integer Programming* adlah menentukan jumlah barang yang dijual sehingga menghasilkan keuntungan maksimal, dengan memerhatikan berbagai faktor penjualan, knapsack problem, menentukan kelayakan dalam melakukan pengeluaran, dan lain-lain.

Dengan kesesuaian yang ada, makalah ini menuliskan bagaimana hubungan antara kedua topik tersebut dengan cara menggambarkan bagaimana penggunaan algoritma *Branch and Bound* dalam menyelesaikan persoalan *integer programming.*

II. Dasar Teori

A. *Branch and Bound*

Branch and Bound (B&B) memang merupakan metode yang digunakan untuk menyelesaikan masalah optimasi. B&B menggunakan pohon yang setiap simpulnya menyatakan status, simpul itulah yang akan ditelusuri sampai didapatkan status yang dituju, pada saat itu, maka simpul tujuan telah ditemukan setelah dilakukan penelusuran.

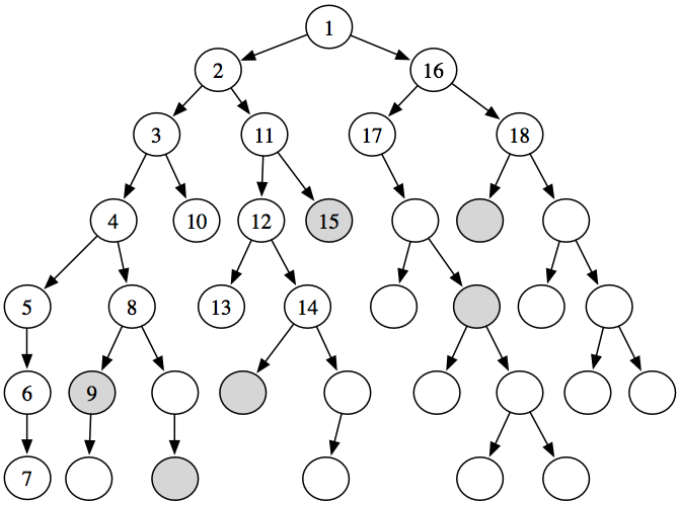
B&B memiliki kemiripan dengan algoritma runut balik yang juga menggunakan pohon dalam pencariannya, namun pada algoritma runut balik, pencarian dilakukan dengan mengutamakan simpul yang lebih dalam dahulu dari pada simpul dengan ketinggian yang sama, atau biasa disebut dengan *Depth First Search* (DFS). Sedangkan pada algoritma B&B, penelusuran simpul akan lebih mengutamakan simpul yang memiliki ketinggian yang sama terlebih dahulu, atau biasa disebut dengan Breadth First Search (BFS).

B&B memiliki perbedaan dengan dengan algoritma BFS biasa, pada BFS biasa, simpul yang memiliki tingkat ketinggian yang sama akan dikunjungi secara berurutan, namun pada B&B, setiap simpul akan diberikan atribut *cost* c(i), *cost* ini akan dijadikan penentu untuk simpul berikutnya yang akan ditelusuri/dibangkitkan, sehingga dengan cara ini kita bisa mengurangi pekerjaan untuk menelusuri simpul-simpul yang tidak relevan atau memiliki *cost* yang tidak sesuai.

Pada algoritma BFS, digunakan sebuah antrian atau *queue* untuk menyimpan simpul yang baru dibangkitkan, antrian itulah yang akan dijadikan acuan dalam menelusuri simpul. Simpul yang baru ditelusuri akan dimasukkan pada akhir antrian, dan simpul yang berada di kepala antrian adalah simpul yang akan ditelusuri anak-anaknya secara berurutan berdasarkan indeks, sehingga BFS yang murni menggunakan skema First In First Out (FIFO) dalam melakukan penelusuran simpul.

Berbeda dengan BFS murni, algoritma B&B tidak menggunakan skema FIFO, melainkan menggunakan skema *priority queue* yang prioritasnya diambil berdasarkan *cost* setiap simpul.

Contoh:



Gambar 2. Contoh pencarian menggunakan B&B
sumber : https://artint.info/figures/ch03/sgraph_bb.png

Pada BFS murni, urutan kebangkitan 10 simpul pertama adalah 1, 2, 16, 3, 11, 17, 18, 4, 10, 12, 15. Sedangkan pada B&B urutan simpul yang ditelusuri bergantung kepada cost untuk simpul-simpul yang sudah bisa dibangkitkan, bisa saja simpul 12 akan dibangkitkan terlebih dahulu daripada simpul 10 jika simpul 12 memiliki *cost* yang lebih kecil, jika kita menginginkan *cost* yang kecil. *Least Cost Search* atau penelusuran berdasarkan nilai yang terkecil ini biasanya dipakai dalam pencarian nilai terkecil, seperti biaya yang dikeluarkan, energi, dan lain-lain.

Secara umum, fungsi yang digunakan dalam menghitung nilai *cost* suatu simpul adalah sebagai berikut :

c(i) = f(i) + g(i)

yang mana :

c(i) adalah *cost* untuk simpul i

f(i) adalah *cost* yang besarnya berbanding lurus dengan jarak dari simpul i ke akar

g(i) adalah *cost* yang besarnya berbanding lurus dengan jarak dari simpul i ke tujuan

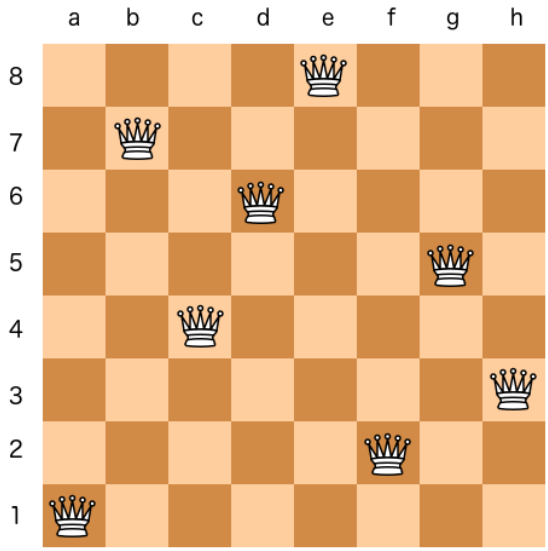
Hal yang menarik disini adalah kita biasanya tidak mengetahui jarak dari simpul i ke simpul tujuan, sehingga g(i) biasa dibuat sebagai fungsi heuristik yang nilainya berdasarkan perkiraan yang kita tentukan sendiri.

Nilai c(i) ini akan digunakan untuk menentukan simpul mana yang akan ditelusuri selanjutnya, sehingga pada B&B kita menggunakan *priority queue* dibandingkan *queue* yang menggunakan skema FIFO.

Secara umum Algoritma B&B adalah sebagai berikut :

1. Masukkan simpul akar ke dalam antrian Q.
2. Jika simpul akar adalah simpul solusi (goal node), maka solusi telah ditemukan. Berhenti.
3. Jika Q kosong, maka tidak ada solusi, Berhenti.
4. Jika Q tidak kosong, pilih dari antrian Q simpul i yang mempunyai nilai *cost* yang paling kecil. Jika simpul i bukan simpul solusi, maka bangkitkan semua anak-anaknya. Jika i tidak mempunyai anak, kembali ke langkah 2.
5. Untuk setiap anak j dari simpul i, hitung c(j), dan masukkan semua anak-anak tersebut ke dalam Q.
6. Kembali ke langkah 2.

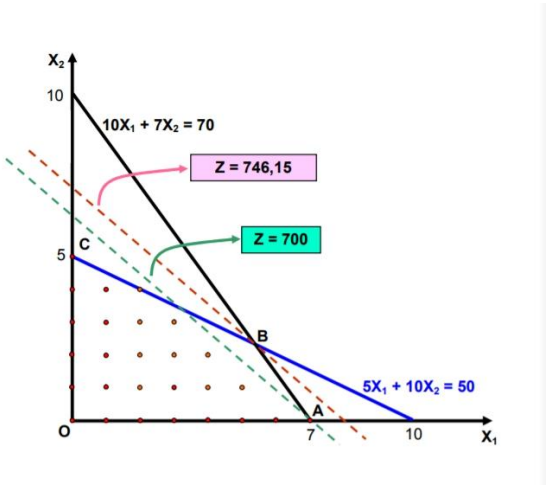
sumber :
http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Branch-&-Bound-(2018).pdf



Gambar 3 : Persoaalan N-Queens Problem yang terkenal bisa diselesaikan menggunakan B&B
sumber : https://cdn-images-1.medium.com/max/1000/1*Zm2pbDR5CS2w2xeUbTBxQQ.png

B. *Integer Programming*

Integer Programming adalah suatu jenis pemrograman linear dengan tambahan syarat yaitu menggunakan variabel yang merupakan bilangan bulat (integer). Pemrograman linear itu sendiri adalah metode matematik untuk menentukan jumlah alokasi terhadap sumber daya yang terbatas untuk mencapai tujuan tertentu seperti mencari nilai optimum, yaitu memaksimumkan keuntungan dan meminimumkan biaya. PL ini sangat sering digunakan dalam berbagai bidang, seperti ekonomi, industri, militer, sosial, dan lain-lain. Dalam kenyataannya, sangat sering bahwa nilai variabel yang berperan sebagai sumber daya merupakan benda diskrit, pada dasarnya memang di dunia ini semuanya diskrit, termasuk gelombang cahaya yang ditemukan membentuk kuanta yang sangat amat kecil, sehingga sering kali variabel yang digunanakan direpresetasikan dengan bilangan bulat, seperti manusia, jumlah benda, dan lain-lain.



Gambar 4, Penyelesaian *Linear Programming* dengan metode grafik
sumber :

eko_hartanto.staff.gunadarma.ac.id/Downloads/files/28374/Integer.pdf

Misalnya, suatu solusi yang menyatakan bahwa kita membutuhkan 3,54 pesawat terbang tidak memliki makna yang berarti, karena kita tidak mungkin menyediakan peawat terbang yang badannya hanya setengah, pada titik ini, cara yang baik kita lakukan adalah menyediakan 3 atau 4 pesawat, yaitu 2 bilangan bulat terdekat dengan 3.54, namun ternyata solusi tersebut juga sering merupakan penyelesaian yang tidak optimal, Sehingga topik mengenai *Integer Programming* yang dikhususkan untuk integer menjadi penting.

Contoh:

Sebuah soal pemrograman linear dengan Z = 100X1 + 90X2;

Dengan syarat

10 X1 + 7 X2 <= 70

5 X1 + 10 X2 <= 50

X1 ; X2 >= 0

Sehingga jika dicari solusinya dengan metode simpleks:

X1 = 5.38

X2 = 2.31

Z = 746.15

Dengan pembulatan terdekat:

X1 = 5

X2 = 2

Z = 680

Bulat optimum sesungguhnya

X1 = 7

X2 = 0

Z = 700

Soal tersebut merupakan contoh dimana pencarian solusi pemrograman linear dengan pembulatan tidak selalu menghasilkan nilai yang optimal.

Dalam pengambilan keputusan yang sempat dibicarakan pada bab pendahuluan, sering jumlah variabel yang bisa dipilih hanyalah 1 atau 0, seperti contohnya saat kita ingin mengadakan suatu kegiatan yang menggunakan modal yang cukup banyak, dua jenis keputusan yang bisa kita ambil adalah "kegiatan dilaksanakan" atau biasa dilambangkan dengan variabel bernilai 1, atau "kegiatan dibatalkan" atau biasa dilambangkan dengan variabel bernilai 0. Sehingga topik ini sangatlah penting dalam kegiatan manajemen.

III. Pembahasan

Seperti yang telah disebutkan sebelumnya, penerapan *integer programming* sangatlah banyak, pada bab ini kita akan membahas secara spesifik beberapa aplikasi B&B dalam industri dan bisnis.

A. Binary Integer Programming

Binary integer programming adalah suatu topik integer programming dimana variabel yang diperhitungkan adalah variabel yang sifatnya binary, yaitu hanya 0 dan 1, atau biasa juga di interpretasikan sebagai iya dan tidak, permasalahan seperti ini contohnya seperti apabila kita sedang menyusun rangkaian acara dengan modal yang terbatas, maka penyelesaian *binary Integer Programming* bisa diterapkan, secara umum, terdapat tiga langkah utama dalam menyelesaikan permasalah Binary Integer Programming ini.

1. Branching
   Misalkan kita sedang mengurusi sebuah masalah dengan variable yang binary, maka branching yang dimaksud adalah dengan membuat berbagai kemungkinan solusi dari kumpulan binary variables tersebut, contohnya jika terdapat 3 variabel binary, maka akan terdapat 8 kemungkinan dalam menyelesaikan permasalahan tersebut, setiap kemungkinan permasalahan tersebut itulah yang akan dianggap sebagai suatu simpul keadaan/state.

   Contoh :
   Maksimumkan Z = 9X1 + 5X2 + 6X3 + 4X4 dengan konstrain

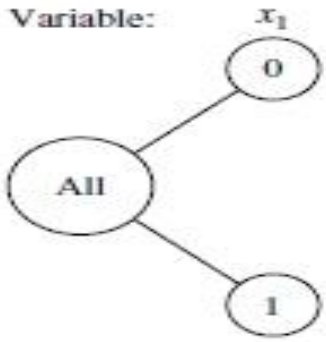
   6X1 + 3X2 + 5X3 + 2X4 <= 10

   X3 + X4 <= 1

   -X1 + X3 <= 0

   -X2 + X4 <= 0

   Maka kita bisa membagi state ini menjadi dua cabang, misalnya dengan cara mengambil X1 = 1 dan X1 = 0.



Gambar 5, ilustrasi branching pada kasus diatas
Sumber : Introduction to Operations Research, Hiller / Lieberman

2. Bounding
   Sekarang kita membutuhkan suatu jenis bound untuk menetukan yang mana "branch" yang kita akan pilih dalam menyelesaikan permasalahan. Hal ini dapat dilakukan dengan mengeliminasi kemungkinan yang tidak mungkin, dari persoalan barusan, bisa kita dapat dengan metode simplex bahwa solusi optimalnya adalah :

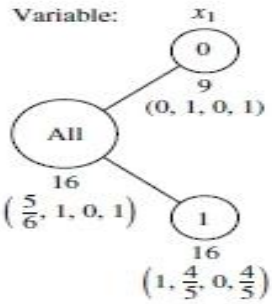
   (X1, X2, X3, X4) = (5/6,1,0,1), dengan Z = 16.5

   sehingga semua penyelesaian dari persoalan akan memenuhi Z <= 16.5 .

   Dengan cara yang sama, kita bisa mendapatkan bound untuk kedua cabang permasalahan yang telah kita buat, yaitu:

   Submasalah 1 : (X1, X2, X3, X4) = (0,1,0,1) dengan Z = 9

   Submasalah 2 : (X1, X2, X3, X4) = (1,4/5,0,4/5) dengan Z = 16/5



Gambar 6, hasil dari melakukan bound
Sumber : Introduction to Operations Research, Hiller / Lieberman

3. Fathoming
   Untuk setiap submasalah yang baru tersebut, lakukan 3 tes, yaitu:

   Test 1: bound harus <= Z*

   Yang mana :

   Z* = nilai dari solusi layak terbaik saat ini,

   Pada submasalah sebelumnya, hasil dari penyelesaian submasalah 1 menghasilkan nilai variabel yang bulat, sehingga bisa disimpulkan bahwa penyelesaian tesebut merupakan penyelesaian yang terbaik saat ini.

   Test 2: hasil relakssasi tidak memiliki solusi layak

   Test 3: Solusi optimalnya merupakan bilangan bulat

IV. KESIMPULAN

Integer Programming merupakan salah satu permasalahan yang sangat amat berguna bagi kehidupan manusia apabila kita mengetahui bagaimana cara menyelesaikannya. *Branch and Bound* juga merupakan algoritma yang baik karena tidak memriksa semua kemungkinan seperti *Brute Force*, namu tetap bisa menghasilkan hasil yang baik. Meskipun masih terdapat kelemahan seperti perkiraan dalam menentukan fungsi heuristik, setidaknya waktu yang diperlukan dalam menyelesaikan permasalahan ini jauh lebih singkat dari pada *Brute Force.*

UCAPAN TERIMAKASIH

Puji syukur saya panjatkan ke hadirat Allah SWT yang mana telah memberikan saya kekuatan, ketabahan dan inspirasi dalam menyelesaikan makalah ini. Terima kasih saya ucapkan kepada Bapak Rinaldi Munir, Bu Nur Ulfa Maulidevi, dan Bu Masayu Leylia Khodra selaku dosen pengajar mata kuliah IF2211 Strategi Algoritma yang telah membimbing saya selama satu semester ini. Tak lupa saya juga berterima kasih kepada rekan - rekan, saudara – saudara dan orang tua yang telah memberikan semangat dan bantuan selama masa pengerjaan makalah ini hingga makalah ini selesai.

REFERENSI

[1] Munir, Rinaldi, Diktat Kuliah IF 2211 : Strategi Algoritma, Bandung, 2018.

[2] https://www.geeksforgeeks.org/branch-and-bound-algorithm/K. Elissa, “Title of paper if known,” unpublished.

[3] *"Mixed-Integer Linear Programming (MILP): Model Formulation" (PDF). Retrieved April 26 2018*

[4] Erickson, J. (2015). "Integer Programming Reduction" (PDF). Archived from the original (PDF) on 18 May 2015

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 29 April 2012

Muhamad Nobel Fauzan/13517042