MASALAH PENJADUALAN JADUAL WAKTU PEPERIKSAAN UNIVERSITI DENGAN PENDEKATAN KOMBINATORIK

Oleh

TEOH CHING CHING

Tesis yang diserahkan untuk memenuhi
keperluan bagi Ijazah Sarjana Sains

Februari 2000

PENGHARGAAN

Buat permulaan, saya ingin mengambil kesempatan ini untuk merakam sepatah dua kata ucapan terima kasih kepada mereka yang telah memberi pertolongan kepada saya dari segi tunjuk ajar, nasihat, kerjasama bahkan dari segi sokongan mental.

Terutama sekali, ucapan terima kasih inginlah saya sampaikan kepada Prof. Quah Soon Hoe iaitu penyelia tesis saya yang memberi sokongan sepanjang tempoh pencalonan saya. Tidak ketinggalan juga, Dr. Tan Kok Chye yang telah banyak memberi nasihat dan tunjuk ajar dalam penyediaan tesis ini. Dengan ini, saya berasa amat bertuah dan bersyukur kerana mendapat pertolongan dan nasihat yang amat saya perlukan daripada Prof. Quah dan Dr. Tan.

Tidak lupa juga, ribuan terima kasih dikirimkan kepada para kakitangan Pusat Pengajian Sains Matematik, Institut Pengajian Siswazah dan perpustakaan Universiti Sains Malaysia yang telah memberi kerjasama dan pertolongan dalam mengolah penyediaan tesis, pengutipan maklumat dan juga bahan-bahan rujukan.

Akhir sekali, saya ingin menyalurkan dedikasi saya kepada keluarga dan rakan-rakan seperjuangan yang telah banyak memberi sokongan.

Salam daripada,

Teoh Ching Ching

ISI KANDUNGAN

ABSTRAK

Kebanyakan universiti mengalami masalah penjadualan peperiksaan pada setiap semester. Secara amnya, penjadualan peperiksaan boleh diringkaskan sebagai satu proses untuk menyusunkan kertas-kertas peperiksaan ke slot-slot masa yang tertentu.

Dalam kajian ini, objektif utama adalah untuk memastikan tiada pelajar yang perlu menghadiri lebih daripada satu peperiksaan dalam satu slot masa yang sama (*konflik langsung*); dan menjadualkan peperiksaan supaya bilangan pelajar yang perlu menduduki peperiksaan yang berturutan adalah diminimumkan (*konflik bersebelahan*).

Di sini, masalah penjadualan diselesaikan dalam tiga peringkat. Dalam peringkat pertama, kertas-kertas peperiksaan dibahagikan ke dalam blok-blok kecil supaya setiap kertas peperiksaan dalam satu blok boleh dijadualkan dalam satu slot masa dan memastikan tiada pelajar yang perlu menghadiri lebih daripada satu peperiksaan dalam satu slot masa yang sama. Seterusnya, blok-blok kecil ini akan disusun dalam satu jujukan linear supaya pelajar yang menduduki peperiksaan berturutan akan diminimumkan. Akhir sekali, blok-blok dalam jujukan akan ditetapkan ke slot-slot masa yang tertentu.

Berdasarkan teknik pewarnaan graf (*graph coloring techniques*) dan algoritma '*additive*' kami mencadangkan satu pendekatan kombinatorik untuk menyelesaikan masalah penjadualan peperiksaan yang senang dimodelkan dan jimat diperlaksanakan.

A Combinatorial Approach To the University Examination Timetable Scheduling Problem

ABSTRACT

Examination timetable scheduling is one of the problems that many universities will have to solve during each semester or term. The problem typically involves assigning each examination paper to a specific time slot.

In this research the primary objective of the examination scheduling problem is to ensure that no student will have to sit for more than one examination in the same time slot (*direct conflict*); and to schedule the examination in such a way that the number of students who have to sit for consecutive examinations is minimized (*back-to-back conflict*).

Here, we solve the problem in three phases. First, examination papers are grouped into blocks where all the papers in a block can be assigned to one time slot and ensure that no student needs to attend more than one paper in one time slot. Second, arrange the blocks of papers in linear sequence and make sure that the number of students who have to sit for consecutive examinations is minimized. Finally, the blocks are then assigned to a particular time slot according to the linear sequence.

Based on the theory of graph coloring techniques and additive algorithm, we propose a combinatorial approach to solve the university examination scheduling problem which is easy to model and is economical to implement.

Bab 1 PENGENALAN

Masalah penjadualan peperiksaan adalah masalah yang sering dihadapi oleh kebanyakan universiti pada setiap semester. Secara amnya, masalah ini melibatkan beribu-ribu pelajar dan beratus-ratus jenis kertas peperiksaan untuk berbagai jenis subjek. Pendaftar peperiksaan dikehendaki untuk menyediakan satu jadual waktu yang membolehkan kesemua kertas peperiksaan bagi setiap subjek dapat dikendalikan dalam jangka masa yang telah ditentukan oleh lembaga universiti.

Masalah penjadualan peperiksaan menjadi semakin rumit apabila setiap pelajar diberi peluang untuk membuat pilihan subjek secara bebas. Ekoran daripada kebebasan pemilihan subjek, maka jelaslah bahawa setiap pelajar perlu menghadiri pelbagai kertas peperiksaan pada hujung semester itu. Untuk memastikan setiap pelajar dapat menghadiri kesemua peperiksaan yang dipilihnya, jadi objektif utama penjadualan jadual waktu peperiksaan adalah untuk memastikan tiada pelajar yang perlu menduduki lebih daripada satu peperiksaan pada slot masa yang sama (*tiada konflik langsung*).

Bagi seseorang pelajar, jika satu peperiksaan adalah terlalu dekat dengan peperiksaan yang lain maka pelajar tersebut mungkin tidak dapat menumpukan perhatian sepenuhnya dalam setiap peperiksaan yang bakal didudukinya. Maka, objektif yang juga dipentingkan dalam penjadualan peperiksaan adalah meminimumkan jumlah pelajar yang perlu menduduki peperiksaan secara berturutan (*konflik bersebelahan*).

Selain daripada itu, setiap universiti mempunyai objektif-objektif yang tersendiri dalam proses penyediaan jadual waktu peperiksaan. Misalnya, peperiksaan untuk subjek A perlu dijadualkan pada satu slot masa tertentu supaya ia didahului oleh peperiksaan subjek B. Sementara ini, sesebuah universiti mungkin mempunyai dewan peperiksaan yang terhad maka bilangan dewan peperiksaan yang digunakan untuk setiap subjek menjadi satu persoalan yang penting.

Dengan kerumitan yang tinggi dan objektif yang berbilang, masalah penjadualan jadual waktu peperiksaan boleh diselesaikan dengan menggunakan prosedur peringkat berbilang. Di mana, masalah yang rumit dipecahkan menjadi masalah-masalah yang mudah dan diselesaikan satu demi satu dalam setiap peringkat berturutan. Biasanya dalam setiap peringkat, hanya sesuatu objektif yang tertentu diambil kira. Keputusan yang diperolehi dari peringkat awal, adalah data input (*raw data*) untuk peringkat yang seterusnya. Prosedur peringkat berbilang akan dijelaskan dengan lebih lanjut dalam bahagian yang seterusnya.

1.1 Latarbelakang Kajian

Objektif-objektif utama dalam kajian ini adalah seperti berikut:

1) Memastikan tiada pelajar yang perlu menduduki lebih daripada satu peperiksaan pada slot masa yang sama (*konflik langsung* = 0).

2) Meminimumkan jumlah pelajar yang perlu menduduki peperiksaan yang berturutan (*minimumkan konflik bersebelahan*).

Dua peperiksaan dikatakan berkonflik jikalau terdapat, sekurang-kurangnya seorang pelajar yang mengambil kedua-dua peperiksaan yang dijadualkan dalam slot masa yang sama. Sekiranya peperiksaan- peperiksaan yang berkonflik dijadualkan pada slot masa yang sama, maka *konflik langsung* adalah jumlah pelajar yang menduduki lebih daripada satu peperiksaan pada satu slot masa yang sama. Contohnya, jika terdapat 5 orang pelajar yang mengambil kedua-dua peperiksaan subjek A dan B, di mana kedua-dua peperiksaan ini dijadualkan pada slot masa yang sama, maka *konflik langsung* ialah 5.

Dalam kajian ini, andaian dibuat dengan hanya terdapat dua slot masa peperiksaan pada setiap hari. Slot-slot masa dalam hari yang sama dianggapkan berturutan. Slot masa yang kedua pada sesuatu hari dan slot masa pertama pada hari berikutannya juga dianggapkan 'berturutan'. Untuk dua kertas peperiksaan yang dijadualkan pada slot-slot masa yang berturutan, kita boleh mendefinasikan *konflik bersebelahan* ini sebagai jumlah pelajar yang mengambil kertas-kertas peperiksaan tersebut. Katakan terdapat 9 orang pelajar yang mengambil kedua-dua kertas peperiksaan A dan B, di mana A dan B dijadualkan pada slot masa yang berturutan, maka *konflik bersebelahan* antara kertas A dan kertas B adalah 9.

1.2 Rancangan Tesis

Kami mencadangkan supaya masalah penjadualan jadual waktu peperiksaan diselesaikan dengan prosedur peringkat berbilang. Dalam kajian ini masalah penjadualan telah diselesaikan dalam tiga peringkat. Dalam peringkat pertama, hanya objektif pertama kajian ini diambil kira. Di sini, kertas-kertas peperiksaan dikumpul dalam blok tertentu dengan menggunakan kaedah pewarnaan graf (*graph coloring method*) yang telah diubahsuai. Ini bagi memastikan jumlah *konflik langsung* adalah sifar dalam setiap blok iaitu tiada pelajar yang perlu menduduki lebih daripada satu peperiksaan dalam satu blok yang sama. Maka, kesemua kertas peperiksaan dalam satu blok boleh dijadualkan pada satu slot masa yang sama. Selepas pengumpulan kertas peperiksaan ke dalam blok, istilah *konflik bersebelahan* perlu dikaji semula. Sekarang, *konflik bersebelahan* merujuk kepada *konflik bersebelahan* antara dua blok. Maka *konflik bersebelahan* antara blok *i* dan *j* adalah jumlah pelajar yang mengambil kedua-dua kertas peperiksaan daripada blok i dan blok j. Dengan kata lain, *konflik bersebelahan* adalah jumlah pelajar yang menduduki dua peperiksaan yang berturutan. Manakala, *konflik bersebelahan antara tiga blok* ialah jumlah pelajar yang menduduki tiga peperiksaan berturutan.

Pertimbangkan kes di mana, blok i, j dan k telah dijadualkan berturutan iaitu blok i mendahului blok j dan blok j mendahului blok k. Jika $Q(i)$ mewakili set pelajar yang mengambil kertas dari blok i, maka $\|Q(i)\|$ adalah jumlah bilangan pelajar yang mengambil kertas dari blok i. Dalam kes ini, *konflik bersebelahan* antara blok i dan j, C_{ij} boleh ditakrifkan sebagai $\|Q(i) \cap Q(j)\|$. Biar C_{ijk} mewakili *konflik bersebelahan* antara blok i, j dan k, maka

$C_{ijk} = \| Q(i) \cap Q(j) \cap Q(k) \|$. Oleh itu,

$$C_{ijk} \leq \min (\|Q(i) \cap Q(j)\|, \|Q(j) \cap Q(k)\|)$$

$$\text{iaitu,} \quad C_{ijk} \leq \min (C_{ij}, C_{jk}) \qquad \text{———————————} \qquad (1)$$

Persamaan (1) menunjukkan bahawa dengan meminimumkan C_{ij} dan C_{jk} secara tidak langsung, C_{ijk} juga diminimumkan dan *konflik bersebelahan* menyeluruh juga turut diminimumkan. Dengan ini, objektif kedua dalam kajian ini boleh dipecahkan kepada dua bahagian, iaitu

Objective a). Meminimumkan bilangan pelajar yang mempunyai dua peperiksaan dalam satu hari.

Objective b). Meminimumkan bilangan pelajar yang menduduki peperiksaan pada slot masa kedua sesuatu hari dan pada slot masa pertama hari berikutnya.

Dalam peringkat kedua, blok-blok peperiksaan yang diperolehi daripada peringkat pertama akan dikumpulkan menjadi pasangan-pasangan dengan tujuan mencapai Objektif a).

Dalam peringkat ketiga, blok-blok dikumpulkan menjadi pasangan-pasangan supaya objektif b) dicapai. Iaitu, *konflik bersebelahan* antara blok-blok dalam setiap pasangan telah diminimumkan. Dengan keputusan yang diperolehi dari peringkat kedua dan ketiga, blok-blok akan disusun menjadi satu jujukan linear di mana *konflik bersebelahan* menyeluruh telah diminimumkan. Dengan merujuk kepada jujukan linear ini, pasangan-pasangan disusun satu demi satu ke hari-hari yang berturutan. Pada masa yang sama, blok-blok dalam setiap pasangan juga diatur mengikut susunan.

Satu program dibina menggunakan Microsoft Visual Basic, Versi 5.0. Dengan program ini, kami telah menguji keberkesanan algoritma ini dengan set-set data yang dijana secara rawak dan beberapa set data yang diperolehi daripada Universiti Sains Malaysia.

Bab 2 PENINJAUAN KAJIAN-KAJIAN LEPAS

2.1 Pengenalan

Dalam bab ini, kita akan membincangkan pelbagai teknik yang telah digunakan dalam penyelesaian masalah penjadualan jadual waktu. Masalah penjadualan sering dihadapi oleh organisasi seperti sekolah, universiti, kilang, syarikat pengangkutan dan lain-lain.

Pada tahun 60-an hingga 70-an, perkembangan bidang penjadualan jadual waktu dihalang oleh teknologi komputer yang kurang maju. Masalah yang berkerumitan tinggi dan kompleks tidak dapat diselesai. Kebanyakan usaha-usaha kajian adalah bersifat konsep (*conceptual*) dan teori (*theoretical*) sahaja dengan perlaksanaan atau ujian yang terhad. Sebahagian besar ujian hanya dikendalikan dengan menggunakan set data kecil yang kuranga realitinya. Akan tetapi, usaha penyelesaian penjadualan jadual waktu telah berkembang sejajar dengan kemajuan teknologi komputer.

Boleh dikatakan pelbagai kaedah dan pendekatan telah digunakan untuk mengatasi masalah penjadualan. Terdapat beberapa percubaan untuk menyelesaikan masalah penjadualan dengan penggunaan teknik-teknik matematik seperti pengaturcaraan linear integer, (*integer linear programming*), kaedah pewarnaan graf (*graph coloring method*) dan kaedah rangkaian (*network flow method*). Sejak tahun 90-an, kaedah-kaedah seperti '*simulated annealing*' dan pendekatan '*knowledge-based*' turut digunakan.

2.2 *Kaedah Pewarnaan Graf*

Kaedah pewarnaan graf telah digunakan dengan meluas untuk menyelesaikan pelbagai bentuk masalah penjadualan waktu. Secara amnya, setiap nod atas graf melambangkan satu kertas peperiksaan. Jikalau terdapat *konflik langsung* antara dua kertas perperiksaan maka satu lengkung (*arc*) akan dilukis untuk menyambungkan dua nod yang melambangkan dua kertas peperiksaan tersebut. Dengan ini, kita boleh mengatakan bahawa nod-nod bersambungan (*adjacent*) tersebut melambangkan kertas-kertas ber*konflik langsung*.

Welsh dan Powell (1967) dan Wood (1968) masing-masing telah menunjukkan bahawa masalah penjadualan jadual waktu boleh diibaratkan sebagai masalah pewarnaan graf (*graph coloring problem*). Mereka telah menggunakan teknik pewarnaan graf dalam masalah penjadualan jadual waktu pada slot-slot masa yang mempunyai tempoh masa yang sama panjang. Sementara itu, Mehta (1981) telah meneliti kajian masalah penjadualan jadual waktu peperiksaan. Beliau telah memperkenalkan satu kaedah yang dapat memampat dan menyusun-semulakan jadual waktu yang telah diperolehi supaya ia dapat diselitkan dalam jangka waktu yang terhad.

Walaupun kaedah yang disebut awal tadi bukan untuk tujuan penjadualan peperiksaan universiti, tetapi algoritma baru boleh diperkenalkan berasaskannya. Dengan ini, dalam bahagian yang berikut kita akan menyentuh secara umum kaedah-kaedah yang dibangkitkan awal tadi.

2.2.1 Pewarnaan Graf dengan Penyusunan Nod

Welsh dan Powell (1967) telah memperkenalkan kaedah Pewarnaan dengan Penyusunan Nod (*Coloring by Ordering Vertices*) dengan nod-nod yang diatur mengikut sususnan menurun berdasarkan darjahnya (*degree*). Nod-nod dalam senarai ini akan diperiksa satu demi satu. Seterusnya, nod-nod yang tidak bersambungan (*adjacent*) akan diwarnakan dengan warna yang sama di mana ia akan membentuk satu blok yang dikenali sebagai blok warna (*coloring block*).

Satu blok warna yang baru akan dimulakan sekiranay terdapat satu nod yang tidak dapat diwarnai dengan sebarang warna yang telah digunakan. Process ini akan berterusan sehingga semua nod telah diwarnai. Dengan kajian tersebut, mereka juga mencadangkan batasan atas untuk numbor kromatik (*chromatic number*) di mana:

$$\gamma(G) = \max_i [\min(i, d_i + 1)],$$

dengan d_i sebagai darjah nod ke-*i* dalam senarai.

2.2.2 Pewarnaan Graf dengan Penjanaan Matriks Kesamaan

Dalam tahun 1968, Wood membuat perbincangan lanjut berkenaan dengan kaedah yang diperkenalkan oleh Welsh dan Powell. Berdasarkan kaedah tersebut, beliau memperkenalkan satu kaedah pewarnaan yang dikenali sebagai Pewarnaan dengan Penjanaan Matriks Kesamaan. Mengikut beliau, nombor kromatik (*chromatic number*) sesebuah graf sangat bergantung kepada pilihan nod-nod yang akan diwarnakan dengan warna yang sama. Teknik ini menunjukkan nod-nod yang mempunyai kesamaan tinggi perlu digolongkan ke dalam blok warna yang sama supaya nombor kromatik dapat dikurangkan.

2.2.3 Penggunaan Kaedah Pewarnaan Graf dalam Penyelesaian Masalah Penjadualan Jadual Waktu Peperiksaan

Seperti kajian-kajian yang lepas, Mehta pada tahun 1981, juga menggunakan kaedah pewarnaan graf dalam penyelesaian masalah penjadualan jadual waktu peperiksaan. Dengan pendekatan ini, beliau telah berjaya memperolehi satu jadual peperiksaan yang tidak mempunyai konflik.

Di samping itu, beliau juga mengkaji aspek-aspek lain dalam masalah penjadualan jadual waktu peperiksaan. Iaitu bagi kes di mana sesetengah kertas peperiksaan tidak boleh dijadualkan bersama di dalam suatu slot masa sama walaupun tiada *konflik-konflik langsung* antara kertas-kertas peperiksaan ini; beliau akan melukis lengkungan (*arc*) antara nod-nod yang melambangkan kertas-kertas peperiksaan ini sebelum pewarnaan graf. Sebaliknya, bagi kes di mana sesetengah kertas peperiksaan perlu dijadualkan bersama pada slot masa yang sama, maka nod-nod yang melambangkan kertas-kertas peperiksaan ini akan dicantumkan menjadi satu nod sebelum pewarnaan graf.

Dengan penyelesaian yang diperolehi daripada pewarnaan graf, Mehta telah menyemak jumlah bilangan blok warna dalam penyelesaian ini. Jika bilangan blok adalah melebihi jumlah slot masa sedia ada, maka sebahagian blok akan diceraikan. Jika sesuatu blok diceraikan, maka kertas-kertas peperiksaan di dalam blok-blok ini akan diagihkan kepada blok-blok peperiksaan yang lain. Dengan ini, jumlah konflik yang muncul akibat penceraian blok dan pengagihan kertas peperiksaan dapat dirumuskan seperti berikut:

$$C_X = \sum_{i \in X} \underset{Y}{Min}(W_{iY}),$$

$$\text{di mana } W_{iY} = \sum_{j \in Y} w(i, j)$$

Dalam persamaan ini, i adalah kertas peperiksaan dalam blok X yang diagihkan ke blok Y. Jika $w(i, j)$ mewakili bilangan pelajar yang mengambil kedua-dua kertas peperiksaan i dan j maka blok peperiksaan yang akan dipilih untuk diceraikan ialah blok yang mempunyai C_X yang minimum..

Akhirnya, penyusunan-semula blok-blok peperiksaan dijalankan oleh Mehta agar bilangan pelajar yang akan menghadiri tiga peperiksaan berturutan akan diminimumkan. Di sini, jumlah bilangan pelajar yang akan menghadiri tiga peperiksaan berturutan dikenali sebagai *konflik bersebelahan tiga blok* (*3 block back-to-back conflict*). Apabila dua blok peperiksaan telah ditetapkan pada satu jujukan, maka blok peperiksaan ketiga yang akan disusun berturutan harus dipilih supaya *konflik bersebelahan tiga blok* diminimumkan. Merujuk kepada kajian Mehta, apabila menjadualkan satu blok peperiksaan kepada satu slot masa, *konflik bersebelahan tiga blok* hanya bergantung kepada dua blok yang disusun dihadapannya. Oleh yang demikian, hubungan rekursi (*recursive relation*) berikut digunakan untuk mencapai satu penyusunan yang mempunyai *konflik bersebelahan tiga blok* yang minimum:

$$f_i(j; S_{i-1}, k) = \underset{m \in S_{i-2}}{Min}[f_{i-1}(k; S_{i-2}, m) + d_{kjm}],$$

di mana

$f_i(j; S_{i-1}, k)$ = *Konflik bersebelahan tiga blok* yang minimum apabila blok-i hingga blok-j telah siap disusun dan blok-j telah ditetapkan pada slot-masa ke-i. Di sini, blok k adalah di hadapan blok-j.

S_{i-1} = Satu set yang mengandungi blok-blok peperiksaan yang telah disusunkan pada setiap slot masa iaitu dari slot masa pertama sehingga slot masa ke-($i-1$).

d_{kjm} = *konflik bersebelahan tiga blok* yang muncul apabila blok k, j, m disusun dalam satu jujukan.

2.3 Pengaturcaraan Matematik

Masalah penjadualan boleh dirumuskan sebagai satu model pengaturcaraan matematik dengan kekangan dan fungsi pengoptimuman (*optimization*). Algoritma-algoritma yang akan boleh digunakan untuk menyelesai mmodel matematik ini adalah pengaturcaraan integer linear, pengaturcaraan tidak-linear, dan algoritma rangkaian (*network flow algorithm*).

Seterusnya dalam bab ini, kita akan melihat beberapa kajian yang telah dilaksanakan dengan memgguna kaedah pengaturcaraan matematik. Dalam kaedah pengaturcaraan matematik, kesulitan pengiraan semakin meningkat apabila bilangan pembolehubah bertambah. Ini menyebabkan sesetengah model tidak dapat diselesaikan secara terus. Biasanya, sebelum penyelesaian masalah dilaksanakan, percubaan untuk mengurangkan bilangan pembolehubah akan dijalankan terlebih dahulu. Ini membolehkan model masalah yang lebih besar dan rumit dihuraikan menjadi model-model yang lebih senang untuk diuruskan. Satu cara untuk mencapai tujuan ini adalah melalui '*Lagrangian Relaxation*'.

2.3.1 Pengaturcaraan Integer Linear dengan 'Lagrangian Relaxation'

Sebenarnya, '*Lagrangian relaxation*' ialah satu prosedur untuk menyertakan kekangan-kekangan ketidaksamaan (*inequality constraints*) yang terpilih ke dalam fungsi objektif dengan mendarab satu pendarab Lagrange (*Lagrangian multiplyer*). Dalam tahun 1988, Arani, Karwan dan Lotfi telah memperkenalkan satu model pengaturcaraan integer yang dapat meminimumkan *konflik bersebelahan* dalam satu hari. Dalam kajian mereka, setiap hari mempunyai dua atau lebih slot masa. Kaedah cabang dan batas telah menjadi pilihan untuk menyelesai masalah ini. '*Lagragian relaxation*' pula digunakan untuk mencapai satu batasan yang lebih baik pada setiap cabang.

2.3.2 Pengaturcaraan Integer Tidak-linear

Pengaturcaraan integer tidak-linear ialah salah satu kaedah yang kerap digunakan dalam penyelesaian masalah penjadualan jadual waktu peperiksaan. Objektif untuk meminimumkan bilangan pelajar yang menduduki dua atau lebih peperiksaan dalam satu slot masa yang sama boleh dirumuskan sebagai fungsi:

$$\min \quad z = \sum_i \sum_j \sum_k c_{ij} x_{ik} x_{jk} \, ,$$

Dalam persamaan ini, c_{ij} ialah jumlah bilangan pelajar yang mengambil kedua-dua kertas peperiksaan i dan j; jika $x_{ik} = 1$, maka ia menunjukkan kertas peperiksaan i dijadualkan pada slot masa k; sama juga dengan $x_{jk} = 1$, ia menunjukkan kertas peperiksaan j dijadualkan pada slot masa k. Sekiranya $x_{rk} = 0$, maka kertas peperiksaan r tidak dijadualkan pada slot masa k.

Dalam tahun 1964, Broder telah mencadangkan satu pendekatan heuristik untuk menyelesaikan model pengaturcaraan integer tidak-linear.

2.3.3 Masalah Perjalanan Jurujual (*Traveling Salesman*)

Dalam masalah perjalanan jurujual, terdapat sebilangan bandar yang perlu dikunjungi oleh jurujual tersebut. Oleh yang demikian, objektif utama jurujual ini adalah untuk mengunjungi kesemua bandar dengan menggunakan satu lintasan yang terpendek.

Pertimbangkan kes yang terdapat dua slot masa pada setiap hari dan slot masa yang kedua pada suatu hari dan slot masa pertama pada hari berikutannya adalah dianggap slot masa berturutan. Maka, objektif untuk meminimumkan *konflik bersebelahan antara blok* boleh dimodelkan seperti masalah perjalanan jurujual (*traveling salesman*). Sekiranya, kertas-kertas peperiksaan telah dikumpulkan menjadi blok, maka setiap blok ini boleh diwakili oleh satu bandar seperti dalam masalah perjalanan jurujual. Jarak di antara dua bandar itu menunjukkan *konflik bersebelahan* antara dua blok peperiksaan. Dengan meminimumkan jarak perjalanan, maka *konflik bersebelahan* turut diminimumkan. Oleh yang demikian, jujukan blok peperiksaan yang mempunyai *konflik bersebelahan* yang telah diminimumkan boleh diwakili oleh laluan optimum yang diperolehi daripada masalah perjalanan jurujual.

Dalam tahun 1979, Coljin telah merumuskan masalah penjadualan jadual waktu peperiksaan sebagai masalah perjalanan jurujual. Satu pendekatan heuristik telah dilaksanakan untuk menyelesai masalah ini. Sementara itu, White dan Chan (1979) telah menggunakan pendekatan perjalanan jurujual untuk menjadualkan blok ke slot masa.

2.3.4 Perkembangan-perkembangan Lain

Satu kaedah baru, simulasi '*annealing*' adalah sesuai digunakan untuk menyelesai masalah penjadualan jadual waktu peperiksaan yang mempunyai berbilang objektif. Sejak kebelakangan ini, beberapa kajian telah dilakukan dengan menggunakan simulasi '*annealing*' dalam penyelesaian masalah penjadualan jadual waktu peperiksaan. Salah satu daripada kajian tersebut telah dilakukan oleh Dige, Lund dan Ravn pada tahun 1993. Dalam kajian mereka, masalah penjadualan disimulasikan sebagai tindakbalas antara bahan-bahan kimia. Peningkatan suhu daripada tindakbalas adalah diandaikan sebagai konflik antara kerta-kertas peperiksaan.

Sementara itu, pada tahun 1998 Thompson dan Dowsland juga menggunakan kaedah ini dalam penyelesaian masalah penjadualan mereka. Thompson dan Dowsland juga menegaskan bahawa di dalam prosedur peringkat berbilang (*multi-phased*), keputusan yang dibuat pada peringkat awal mungkin akan memburukkan penyelesaian pada peringkat seterusnya. Dalam kajian mereka, penggunaan simulasi '*annealing*' boleh mengatasi masalah ini kerana keputusan yang dibuat pada peringkat awal boleh diubahsuai di peringkat seterusnya. Walau bagaimanapun, menurut mereka kaedah ini masih perlu dimajukan lagi.

Kadangkala terdapat kes di mana keperluan sesebuah universiti tidak dapat dirumuskan melalui model matematik. Maka satu sistem yang lebih '*flexible*' diperlukan. Oleh yang demikian, banyak kajian telah dibuat seperti di dalam bidang sistem pakar (*expert*), sistem interaktif dan sistem berasas pengetahuan (*knowledge-based*). Misalnya, White dan Wong (1988) telah memperkenalkan satu perisian interaktif untuk masalah penjadualan. Khader (1993) pula telah memperkenalkan satu kaedah penyelesaian dengan menggunakan sistem berasas pengetahuan.

2.4 Kesimpulan

Pada keseluruhannya, kita boleh membuat kesimpulan bahawa masalah penjadualan jadual waktu peperiksaan dengan kekangan berbilang perlu diselesaikan dengan menggunakan prosedur berbilang peringkat (*multi phase*). Dengan prosedur ini, setiap peringkat adalah saling bergantung.

Daripada kajian-kajian yang lepas, kaedah pewarnaan graf adalah kaedah yang biasa dipraktikkan pada peringkat awal penyelesaian masalah penjadualan jadual waktu peperiksaan. Dalam peringkat ini, objektif utamanya adalah untuk mencapai satu penyelesaian yang tidak ada *konflik langsung* antara kertas-kertas peperiksaan di dalam setiap blok. Oleh kerana masalah penjadualan senang dimodelkan sebagai satu masalah pewarnaan graf maka kaedah ini selalu menjadi pilihan untuk digunakan dalam prosedur penyelesaian.

Pengaturcaraan matematik adalah satu lagi pendekatan yang telah digunakan secara meluas dalam penyelesaian masalah penjadualan. Walapun kaedah ini melibatkan proses pengiraan yang rumit, ia masih boleh diringkaskan dan dihuraikan menjadi model yang lebih mudah dengan menggunakan teknik-teknik tertentu.

Di dalam kajian ini, kita telah merumuskan masalah penjadualan jadual waktu peperiksaan sebagai satu masalah pewarnaan graf dan seterusnya satu model pengaturcaraan integer dirumuskan. Teknik-teknik pewarnaan graf dan pengaturcaraan '*additive*' turut digunakan untuk menyelesaikan model-model masalah ini. Dalam Bab 3, teknik-teknik ini akan dibincangkan.

Bab 3 METODOLOGI

3.1 *Kaedah Pewarnaan Graf*

3.1.1 Pengenalan

Secara am, satu graf adalah satu gambarajah yang mengandungi lengkungan-lengkungan (*arcs*) dan nod-nod (*nodes*), di mana setiap lengkungan menyambungkan dua nod. Di dalam kajian matematik, graf boleh dikelaskan sebagai graf berarah (*directed graf*) dan graf tidak berarah (*undirected graf*). Bagi graf berarah, lengkungan yang menyambungkan dua nod mempunyai satu anak panah yang berpunca daripada satu nod dan mengarah menuju ke arah nod yang kedua. Di sebaliknya, dalam graf tidak berarah, lengkungan menyambungkan dua nod tanpa mengambil kira arahnya. Di sini, hanya graf tidak berarah sahaja yang dikaji.

Dalam bab ini, satu kaedah piawai pewarnaan graf akan dibincangkan dan satu contoh ringkas dibentangkan di akhir bab ini.

3.1.2 Perumusan Masalah

Penjadualan adalah satu proses untuk menetapkan perlaksanaan tugas-tugasan (*jobs*) ke slot-slot masa yang tertentu. Setiap tugasan ini memerlukan satu atau lebih daripada satu sumber (*resources*) untuk pelaksanaannya. Sumber-sumber tersebut mungkin dikongsi antara tugas-tugasan ini. Pada sebarang slot masa, satu sumber hanya boleh digunakan untuk perlaksanaan satu tugasan.

Katakan, dua tugasan adalah ber*konflik* jika mereka berkongsi satu sumber yang sama. Maka, masalah yang dihadapi dapat diringkaskan sebagai penjadualan tugasan supaya tugasan yang ber*konflik* tidak dijadualkan bersama di dalam satu slot masa yang sama.

Biar $U = \{u_1, u_2, \ldots, u_m\}$ mewakili satu set yang mengandungi m sumber dan $J = \{j_1, j_2, \ldots, j_n\}$ mewakili satu set yang mengandungi n tugasan yang akan dijadualkan. Jika satu nod atas graf melambangkan satu tugasan. Maka, terdapat n nod di atas graf melambangkan tugasan dalam set J. Lengkungan dilukis antara dua nod jika tugasan-tugasan yang diwakilinya adalah ber*konflik*. Di sini, nod-nod yang disambungkan oleh satu lengkungan dikatakan bersambungan (*adjacent*).

Dalam proses pewarnaan graf, nod-nod yang bersambungan diwarnakan dengan warna yang berlainan. Sekiranya satu warna mewakili satu slot masa, maka pewarnaan graf adalah menyerupai penyelesaian masalah penjadualan. Iaitu, nod-nod yang diwarnakan dengan warna yang sama boleh dijadualkan pada slot masa yang sama. Oleh yang demikian, bilangan warna yang digunakan atau nombor kromatik adalah bilangan slot masa yang diperlukan untuk penjadualan tugasan-tugasan. Dengan ini, tiada tugasan yang ber*konflik* akan dijadualkan bersama dalam satu slot masa.

3.1.3 Prosedur

Sebelum penjelasan prosedur pewarnaan graf, tatatanda-tatatanda disenaraikan dahulu.

Tatatanda	Keterangan
$G(V, E)$	Graf G di mana set $V = \{v_1, v_2, \ldots, v_n\}$ mewakili n nod atas graf dan set $E = \{e_1, e_2, \ldots, e_m\}$ ialah set m lengkungan. Di atas graf G, setiap lengkungan menyambungkan dua nod yang merupakan ahli set V.
$G_{V_t}(V_t, E_t)$	Sub-graf bagi graf $G(V, E)$ pada lelaran (*iteration*) ke-t. $V_t \subset V$ dan $E_t \subset E$. Tatatanda bagi $G_{V_t}(V_t, E_t)$ selalunya diringkas sebagai G_t.
$d_{G_t}(x_k)$	Darjah bagi node $x_k \in V_t$ dalam sub-graf $G_{V_t}(V_t, E_t)$. Perhatian: darjah bagi sesebuah nod x, ialah bilangan lengkungan yang menyambungkan x dengan nod yang lain.
$R = \{r_1, r_2, \ldots, r_n\}$,	Set yang mengandungi nod-nod yang telah disusun mengikut susunan meningkat (*increasing*) berdasarkan darjahnya. Dalam graf $G(V, E)$, $r_i = x_{r_i}$ di mana $r_i = x_{r_i}$ dan $x_{r_i} \in V$ bagi setiap $i \in \{1, 2, 3, \ldots, n\}$.
$C = \{1, 2, 3, \ldots\}$	Set yang mengandungi warna yang telah digunakan. Setiap warna dilambang oleh satu nombor integer. Jumlah bilangan unsur C yang diwakili oleh $\|C\|$ adalah lebih besar atau sama dengan nombor kromatik (*chromatic number*) graf $G(V, E)$ iaitu $\gamma(G)$.
$F(x_i)$	Satu set warna yang tidak boleh digunakan untuk mewarnai nod x_i di mana $x_i \in R$ dan $F(x_i) \subset C$.
$A(n \times n)$	Matriks konflik, di mana, $a_{ij} = \begin{cases} 1; & \text{jika } x_i \text{ dan } x_j \text{ berkonflik dan } i \neq j \\ 0; & \text{jika } x_i \text{ dan } x_j \text{ tidak berkonflik} \end{cases}$ dengan $x_i \in V$, $x_j \in V$.

Jadual 3-1 Jadual Tatatanda bagi Kaedah Mewarnakan Graf

Kita membahagikan kaedah ini kepada dua peringkat. Peringkat pertama adalah proses penyusunan nod dan peringkat kedua adalah proses pewarnaan graf.

3.1.4 Proses Penyusunan Nod

Proses penyusunan nod bermula dengan pengiraan darjah bagi setiap nod. Katakan pada lelaran $t=0$, dalam sub-graf G_0, x_k adalah nod yang mempunyai darjah minimum. Maka, x_k dipilih untuk menjadi unsur pertama dalam set R. Dengan ini, pada lelaran $t = 1$, singkirkan nod x_k daripada graf G_0 maka sub-graf G_1 dihasilkan. Kira darjah bagi setiap nod dalam G_1; kemudian, daripada graf G_1, pilih satu nod dengan darjah yang minimum untuk menjadi unsur yang ke-dua dalam set R. Seterusnya, proses ini diulangi sehingga semua nod telah disusun dalam set R. Langkah-langkah yang terperinci telah disenaraikan di bawah:

1. $t = 0$
2. Set nod permulaan dalam graf $G_o(V_0, E_0)$ ialah $V_0 = V = \{x_1, x_2, \ldots, x_n\}$.
3. Tentukan darjah $d_{G_t}(x_k)$ bagi setiap nod $x_k \in V_t$.
4. Pilih nod x_k, di mana $d_{G_t}(x_k) = \min_{x_j \in V_t} d_{G_t}(x_j)$.
5. $t \leftarrow t + 1$
6. Masukkan x_k ke set R. Biar $r_t = x_k$.
7. Jika $t = n$, maka TAMAT prosedur ini.
8. Jana graf $G_{V_t}(V_t, E_t)$ dengan menyingkirkan nod x_k dan kesemua lengkungan yang berhubung dengannya daripada graf G_{t-1}.
9. Balik ke langkah yang ke-3.

3.1.5 Pewarnaan Graf

Dalam proses pewarnaan graf, terdapat n lelaran yang perlu dilaksanakan; iaitu lelaran $t = \{0, 1, 2, \ldots n-1\}$. Sekiranya, warna dinomborkan dengan integer seperti, 1, 2, 3... , maka proses pewarnaan bermula dengan mewarna nod yang disusun pada akhir set R dengan warna yang paling kecil. Kemudian, warna ini disenaraikan dalam set $F(x_i)$ bagi semua nod x_i yang bersambungan dengannya. Proses ini berulang sehingga semua nod dalam R telah diwarna dengan satu warna. Langkah-langkah terperinci disenaraikan di bawah:

1. $t = 0$
2. Permulaan: $G_0(V_0, E_0)$, di mana $V_0 = \{x_1, x_2, .., x_n\}$ dan $E_0 = \{e_1, e_2, \ldots e_m\}$, $C = \{1\}$, dan $F(x_i) = \phi$ untuk semua $x_i \in V$; Laksanakan penyusunan nod dengan mengikut langkah-langkah

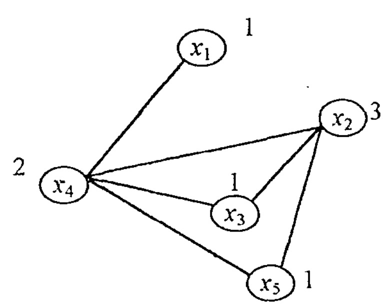
yang telah dibincangkan sebelum ini. Set R yang diperolehi ialah $\{r_1, r_2, \ldots, r_n\}$, di mana $r_i = x_{r_i}$ dan $x_{r_i} \in V$ bagi setiap $i \in \{1, 2, 3, \ldots, n\}$.

3. Warnakan nod r_{n-t} dengan satu warna set C iaitu warna yang dilambang oleh nombor yang terkecil dan pastikan warna ini bukan unsur set $F(r_{n-t})$ iaitu unsur yang paling kecil dalam set $C \cap \overline{F(r_{n-t})}$.

   Jika set $C \cap \overline{F(r_{n-t})}$ adalah set kosong maka satu warna baru, c_k akan diperkenalkan dan nod r_{n-t} akan diwarnakan dengan warna ini. Masukkan c_k ke dalam set C.

4. Untuk semua $i \in \{1, 2, .., n\}$, jika $a_{i\,(n-t)} = 1$ maka masukkan warna k ke dalam set $F(x_i)$.
5. $t = t + 1$
6. Jika $t < $ n balik ke langkah yang ke-3.
7. TAMAT.

3.1.6 Contoh



Rajah 3-1 Graf $G(V, E)$

j	$d_{G_0}(x_j)$	$d_{G_1}(x_j)$	$d_{G_2}(x_j)$	$d_{G_3}(x_j)$	$d_{G_4}(x_j)$
1	1	-	-	-	-
2	3	3	2	-	-
3	2	2	-	-	-
4	4	3	2	1	-
5	2	2	2	1	0

Jadual 3-2 Darjah untuk setiap nod x_j dalam sub-graf, G_t

Dengan menelitikan Rajah 3-1, darjah untuk setiap nod dalam $G(V, E)$ telah ditentu dan disenaraikan dalam Jadual 3-2. Nombor integer di sebelah setiap nod adalah warna yang digunakan untuk mewarnai nod tersebut.

Mulakan penyusunan nod dalam graf $G(V, E)$ mengikut darjahnya dan set R yang diperolehi adalah $\{x_1, x_3, x_5, x_2, x_4\}$. Seterusnya, nod dalam R diwarnakan satu demi satu bermula dari nod yang terakhir iaitu, x_4. Pada lelaran $t = 0$, set permulaan bagi $C = \{1\}$. Matriks konflik yang sepadan dengan graf $G(V, E)$ ialah A(5x5) yang telah disenaraikan pada Jadual 3-3. Set $F(x_j)$ pada permulaan adalah kosong bagi semua $x_j \in V$ yang tersenarai seperti dalam Jadual 3-4.

	1	2	3	4	5
1	0	0	0	1	0
2	0	0	1	1	1
3	0	1	0	1	0
4	1	1	1	0	1
5	0	1	0	1	0

Jadual 3-3 Matriks Konflik untuk graf Rajah 3-1

	t = 0	t = 1	t = 2	t = 3	t = 4
$F(x_j)$	C = {1}	C = {1}	C = {1, 2}	C = {1, 2,3}	C = {1, 2, 3}
$F(x_1)$	ϕ	ϕ	{2}	{2}	{2}
$F(x_2)$	ϕ	{1}	{1, 2}	{1,2}	{1,2}
$F(x_3)$	ϕ	ϕ	{2}	{2,3}	{2,3}
$F(x_4)$	ϕ	{1}	{1}	{1}	{1}
$F(x_5)$	ϕ	ϕ	ϕ	ϕ	ϕ

Jadual 3-4 Perubahan pada setiap lelaran t untuk set warna yang telah digunakan C dan set warna yang tidak boleh digunakan $F(x_j)$ untuk mewarnai nod x_j

Pada lelaran $t = 0$, bermula dengan nod yang akhir dalam set R, iaitu $r_{(n-t)} = r_{5-0} = r_5 = x_5$. Warna yang paling kecil dalam set C ialah warna 1 dan $F(x_5) = \phi$, maka $C \cap \overline{F(r_{n-t})} = C \cap \overline{F(x_5)} = \{1\}$. Oleh itu, nod x_5 diwarnakan dengan warna 1. Periksa a_{j5} dalam matriks A(5x5) untuk semua $j = \{1, 2, 3, 4, 5\}$ yang ditunjukkan dalam Jadual 3-3, kita akan perolehi:

$a_{15} = 0, a_{25} = 1, a_{35} = 0, a_{45} = 1, a_{55} = 0;$

$a_{25} = 1$ dan $a_{45} = 1$ bermakna nod x_2 dan x_4 yang berkonflik dengan x_5. Masukkan warna 1 dalam set $F(x_j)$ untuk x_2 iaitu biar $F(x_2) = \{1\}$ dan $F(x_4) = \{1\}$. Perkembangan ini dicatitkan dalam Jadual 3-4.

Pada lelaran $t = 1$, nod yang akan diwarna ialah $r_{(n-t)} = r_4$ dalam set R, iaitu nod x_4. Untuk pewarnaan nod x_4, periksa set $C \cap \overline{F(r_{n-t})} = C \cap \overline{F(x_4)}$ di mana kita akan mendapati ia adalah bersamaan dengan ϕ kerana $C = \{1\}$ dan $F(x_4) = \{1\}$. Dalam kes ini, warna baru, warna 2 diperkenalkan dan nod x_4 diwarna dengan warna 2; maka, set $C = \{1, 2\}$. Seterusnya, periksa semua a_{j4} dalam matriks A untuk $j = \{1, 2, 3, 4, 5\}$. Daripada Jadual 3-3, kita dapati a_{34}, a_{24} dan a_{14} adalah bernilai satu, maka warna 2 tidak boleh digunakan untuk mewarnai nod x_3, x_2, dan x_1 dan ia dimasukkan ke dalam set $F(x_3)$, $F(x_2)$ dan $F(x_1)$. Perubahan ini juga dicatitkan dalam Jadual 3-4.

Pada lelaran $t = 2$, nod x_2 akan diwarnakan. Sekarang, $C \cap \overline{F(r_{n-t})} = C \cap \overline{F(x_2)} = \phi$. Maka warna 3 diperkenal dan digunakan untuk mewarnai x_2. Set-set C dan $F(x_j)$ yang telah dikemaskini itu telah disenaraikan dalam Jadual 3-4. Nod yang akan diwarnakan seterusnya ialah $r_2 = x_3$. Sekarang, $C \cap \overline{F(r_{n-t})} = C \cap \overline{F(x_3)} = \{1\}$; maka, x_3 diwarnakan dengan warna 1. Pada akhirnya, nod yang akan diwarna ialah $r_1 = x_1$, di mana $C \cap \overline{F(r_{n-t})} = C \cap \overline{F(x_1)} = \{1,3\}$, warna yang paling kecil dipilih, maka x_1 diwarna dengan warna 1 dan prosedur ini telah lengkap dan tamat. Warna yang digunakan untuk mewarnai sesuatu nod ditandakan di sebelah nod atas graf yang ditunjukkan dalam Rajah 3-1.

3.2 *Algoritma 'Additive'*

3.2.1 Pengenalan

Satu daripada algoritma yang digunakan untuk menyelesaikan masalah pengaturcaraan integer sifar-satu (*zero-one integer programming*) ialah algoritma '*additive*'. Algoritma ini mengambil kira semua penyelesaian yang mungkin (*all possible results*). Akan tetapi, ia boleh mengenalpasti penyelesaian yang tidak tersaur dan boleh mengeluarkannya daripada pertimbangan. Akibatnya, hanya sebahagian daripada penyelesaian yang mungkin diambil kira sahaja.

3.2.2 Perumusan Masalah

Algoritma '*additive*' boleh dilaksanakan atas model seperti berikut:

Minimumkan $$z = \sum_{j=1}^{n} c_j x_j$$

terhadap

$$\sum_{j=1}^{n} a_{ij} x_j + S_i = b_i, \qquad i = 1, 2, 3, \ldots, m$$

$$x_j = 0 \text{ or } 1, \qquad j = 1, 2, 3, \ldots, n$$

$$S_i \geq 0, \qquad i = 1, 2, 3, \ldots, m$$

Dalam kajian ini, model di atas dinamakan Model '*Additive*'. Ia mempunyai ciri-ciri berikut:

1. Fungsi objektif adalah sentiasa meminumumkan nilai matlamat;

   Perhatian: Model yang mempunyai fungsi objektif memaksimumkan nilai matlamat boleh diubahsuaikan dengan mendarab kedua-dua belah fungsi objektif dengan -1.

2. Semua koefisian (*coefficients*) pembolehubah-pembolehubah adalah tidak-negatif dalam fungsi objektifnya.

   Perhatian: Pembolehubah yang mempunyai koefisian negatif dalam fungsi objektifnya mesti diganti dengan satu pembolehubah x_j' di mana $x_j' = 1 - x_j$.

3. Semua kekangan mesti berjenis (≤).

   Perhatian:

   a). Kekangan yang berjenis (≥) akan didarab dengan −1 pada kedua-dua belah ungkapan kekangan tersebut supaya ia akan ditukar ke bentuk (≤).

   b). Kekangan yang berjenis (=) mesti ditukar ke jenis (≤) dan ditambahkan satu kekangan berjenis (≥) yang sepadan dengan kekangan yang ditukar tadi. Untuk kekangan yang ditambah ini, kita gunakan langkah ke-3a) untuk menukarkannya ke jenis (≤).

Pada permulaan algoritma, anggapkan nilai bagi semua pembolehubah adalah sifar dan nilai matlamat adalah sifar juga. Walau bagaimanapun, nilai bagi semua pembolehubah belum ditetapkan. Satu penyelesaian dikatakan tidak tersaur, jika terdapat sebarang pembolehubah lalai S_i adalah bernilai negatif. Dalam keadaan ini, sesetengah nilai pembolehubah mesti ditukar ke nilai satu untuk menggerakkan penyelesaian in ke arah ketersauran. Pembolehubah dipilih satu demi satu untuk pertimbangan. Pada setiap ulangan, nilai pembolehubah yang dipilih akan ditetapkan pada 1 atau 0. Apabila nilai sesuatu pembolehubah telah ditetapkan, maka ia dikatakan telah ditambah ke dalam penyelesaian semasa. Algoritma ini akan menentukan sama ada sesuatu penyelesaian semasa boleh dibataskan iaitu perhentikan penambahan pembolehubah baru ke dalam penyelesaian semasa. Di samping itu, algoritma ini akan menentukan sama ada sesuatu pembolehubah boleh diabaikan daripada pertimbangan selanjutnya. Tujuan pembatasan dan pengabaian adalah untuk menggerakkan penyelesaian ke arah ketersauran dan mencapai nilai objektif yang lebih baik.

3.2.3 Definasi

PEMBOLEHUBAH BEBAS

Pada sebarang ulangan, satu pembolehubah dikatakan bebas jika nilainya belum ditentukan. N_t ialah satu set yang terdiri daripada subskrip-subskrip bagi pembolehubah-pembolehubah bebas pada ulangan ke-t. Katakan pada ulangan ke-t, pembolehubah x_2 dan x_3 adalah bebas maka N_t={2,3}.

PENYELESAIAN SEMASA

Penyelesaian semasa ialah koleksi pembolehubah yang nilainya telah ditetapkan samada pada nilai 1 atau 0. Penyelesaian semasa boleh diwakili sebagai satu set yang teratur. Biar J_t mewakili penyelesaian semasa pada ulangan ke-t. Unsur-unsur set J_t mengandungi (+j) yang mewakili subskrip bagi pembolehubah, x_j yang nilainya ditetapkan pada 1; dan (-j) mewakili pembolehubah x_j yang nilainya telah ditetapkan pada 0. Misalnya, pada ulangan ke-t, J_t = {-7, 4} maka kita dapat tahu bahawa $x_7 = 0$ dan $x_4 = 1$. Perhatikan bahawa susunan unsur-unsur dalam J_t adalah mengikut jujukan pembolehubah dipilih untuk menetapkan nilainya; pembolehubah yang baru dipilih ditambah ke hujung kanan set J_t. Dalam contoh ini, x_4 dipilih selepas x_7 maka ia ditambah ke hujung kanan set J_t.

MEMBATAS PENYELESAIAN SEMASA

Sesuatu penyelesaian semasa akan dibatas daripada penambahan pembolehubah yang seterusnya atau dikatakan '*fathomed*', jika dan hanya jika tidak ada penyelesaian yang lebih baik yang akan dijana daripada penyelesaian semasa ini. Maknanya:

1. Ia tidak akan menggerakkan penyelesaian ke arah nilai objektif yang lebih baik.
2. Ia tidak akan menggerakkan penyelesaian ke arah ketersauran.

Di sini, kami ingin menjelaskan tatatanda-tatatanda yang akan digunakan dalam perbincangan langkah-langkah algoritma seterusnya.