

Model Pendekatan Metaheuristik Dalam Penyelesaian optimisasi Kombinatorial

Muhammad Iqbal¹, Muhammad Zarlis², Tulus³, Herman Mawengkang³

¹ Fakultas Ilmu Komputer dan Teknologi Informasi, Prodi Doktor Ilmu Komputer, Universitas Sumatera Utara, Medan, Indonesia

² Fakultas Ilmu Komputer dan Teknologi Informasi, Prodi Ilmu Komputer, Universitas Sumatera Utara, Medan, Indonesia

³ Fakultas Matematika dan Ilmu Pengetahuan Alam, Prodi Matematika, Universitas Sumatera Utara, Medan, Indonesia

Email: ¹muhammadiqbal@dosen.pancabudi.ac.id, ²m.zarlis@yahoo.com, ³tulus_jp@yahoo.com, hmawengkang@yahoo.com

Abstrak—Secara praktis pada umumnya pengguna kurang memerlukan penyelesaian eksak, yang mereka inginkan adalah suatu penyelesaian pendekatan dalam waktu yang relative cepat. Sehingga terkait dengan ini berarti metode heuristic atau metaheuristik adalah sesuai. Beberapa pendekatan pendekatan ini telah memberikan jaminan kualitas tertentu penyelesaian dan dalam waktu polynomial. Sayangnya, sifat teoritis demikian ini biasanya berakibat performance yang secara relative buruk. Dengan kata lain heuristic sederhana adakalanya lebih cepat dan menghasilkan penyelesaian lebih baik dari pada algoritma pendekatan, walaupun heuristic sederhana tidak menjamin kualitas dan dalam banyak kasus ia menghasilkan penyelesaian buruk. Artinya pendekatan seperti ini “problem sensitive”. Untuk itu perlu diajukan metode metaheuristik dapat mengatasi kelemahan dalam hal performancenya, oleh karena itu untuk permasalahan kombinatorial adalah masalah yang mempunyai himpunan solusi layak (feasible) yang terhingga. Meskipun secara prinsip solusi dari masalah ini bisa didapatkan dengan enumerasi lengkap, pada masalah kompleks dibutuhkan waktu yang tidak bisa diterima secara praktis.

Kata Kunci: Optimisasi, Kombinatorial, Metaheuristik

1. PENDAHULUAN

Pendekatan Metaheuristik untuk menyelesaikan persoalan *Combinatorial Optimization* mencakup Variabel Neighborhood Search [Liberti, L. Nannicini G.,2009], strategi penetapan variable secara otomatis [Berthold T and A. Gleixner.,2010], Percabangan Lokal [Mawengkang H. and B. A Murtagh.,1986], pencarian persekitaran layak [Mawengkang H.and Murtagh. B. A.,1986], Pompa Kelayakan [D’Ambrosio C. et.al.,2010; Frangioni A. et.al.,2010; M. Fischetti, F. Glover, A. Lodi.,2005], heuristic berdasarkan pembulatan iteratif [Nannicini G. and Belotti. P.,2009]. Beberapa waktu lalu [Liberti. L.,et.al.,2011] mengusulkan sebuah heuristic CO yang disebut algoritma Relaxed-Exact-Continuous-Integer Problem Exploration (RECIPE). Algoritma ini menggabungkan fase pencarian global berdasarkan Variabel Neighborhood Search [L. Liberti.,et.al.,2019] dan fase pencarian lokal berdasarkan heuristic. Namun, dalam pendekatan heuristic, salah satu kesulitan algoritmik utama yang terkait dengan CO adalah menemukan solusi yang layak. Dari sudut pandang kompleksitas kasus terburuk, menemukan solusi CO yang layak sama sulitnya dengan menemukan solusi Pemrograman Nonlinier yang layak, yaitu NP-hard [Mawengkang H. and Murtagh B. A.,1986].

Sering kali persoalan optimisasi baik yang dilihat dari segi teoritis maupun praktis terdiri dari pencarian konfigurasi “terbaik” himpunan *variable* sedemikian hingga tercapainya tujuan (objektif). Persoalan optimisasi yang dikenal sebagai persoalan optimisasi kombinatorial. Yang selanjutnya dalam paper ini disingkat dengan *CO*. Maka persoalan $CO P=(S, f)$ dapat didefinisikan sebagai persoalan yang mengandung

- himpunan variabel $X = \{x_1, \dots, x_n\}$;
- domain variabel D_1, \dots, D_n ;
- kendala terhadap variabel; dan
- suatu fungsi objektif untuk diminimumkan f .

Himpunan dari semua penugasan yang mungkin adalah

$$S = \{s = \{(x_1, v_1), \dots, (x_n, v_n)\} | v_i \in D_i, s \text{ memenuhi semua kendala} \}.$$

S biasanya disebut ruang pencarian (atau penyelesaian) karena setiap elemen dapat menjadi calon penyelesaian. Untuk menyelesaikan persoalan CO perlu ditentukan $s^* \in S$ dengan nilai fungsi objektif minimum, yaitu, $f(s^*) \leq f(s) \forall s \in S$. s^* disebut penyelesaian optimal global dari (S, f) dan himpunan $S^* \subseteq S$ disebut himpunan penyelesaian global.

Dari rumitnya permasalahan ini maka diperlukan adanya algoritma yang mampu menyelesaikan persoalan CO secara efisien dan pendekatan algoritmik terhadap persoalan dilakukan dengan pendekatan (*approximate*), Persoalan kombinatorial banyak muncul dalam aplikasi praktis. Beberapa pemakaian mencakup sektor-sektor, seperti, optimisasi portofolio investasi (Bienstock, 1996; Jobst et al., 2001), rancangan layout dalam sektor servis dan manufaktur (Danna et al., 2005; Sawaya, 2006), rancangan terintegrasi dan terkendali proses kimia (Flores-Tlacuahuac dan Biegler, 2007), system distribusi air minum (Laird et al., 2006), rantai suplai multiperioda (Lejeune, 2009; Kaya dan Urek, 2016), perancangan system energy (Kim dan Edgar, 2014; Yang et al., 2015), Perencanaan layout lokasi konstruksi (Hammad et al., 2016).

Semua algoritma meta-heuristik memiliki dua komponen inti: intensifikasi (pencarian lokal) dan diversifikasi (pencarian global). Intensifikasi mengeksplorasi daerah tetangga yang menjanjikan dengan harapan menemukan solusi yang lebih baik. Di sisi lain, diversifikasi memastikan bahwa semua wilayah ruang pencarian telah dikunjungi, yang memungkinkan algoritma untuk melompat keluar dari optimum lokal. Menyeimbangkan interaksi antara dua komponen dapat secara signifikan mempengaruhi efisiensi algoritma meta-heuristik (Yang et al., 2013a), kinerja algoritma meta-heuristik sangat tergantung pada keseimbangan yang baik antara kedua komponen ini. Terlalu banyak intensifikasi dapat

mengakibatkan hilangnya keragaman cepat dalam populasi yang meningkatkan kemungkinan untuk membuat algoritma terperangkap dalam optimum lokal. Diversifikasi agresif dapat menyebabkan pencarian tidak efisien dan memperlambat kinerja pencarian keseluruhan (Yang et al., 2014).

Algoritma pendekatan yang paling banyak terpakai dan sukses adalah algoritma penjelajahan local. Algoritma ini dimulai dari penyelesaian yang diberikan dan mencoba untuk menentukan penyelesaian lebih baik dari daerah sekitar yang sesuai (melalui definisi) dari penyelesaian yang telah diperoleh. Maka, jika suatu titik penyelesaian lebih baik ditemukan, maka titik ini menggantikan titik penyelesaian sebelumnya dan penjelajahan local terus berlanjut.

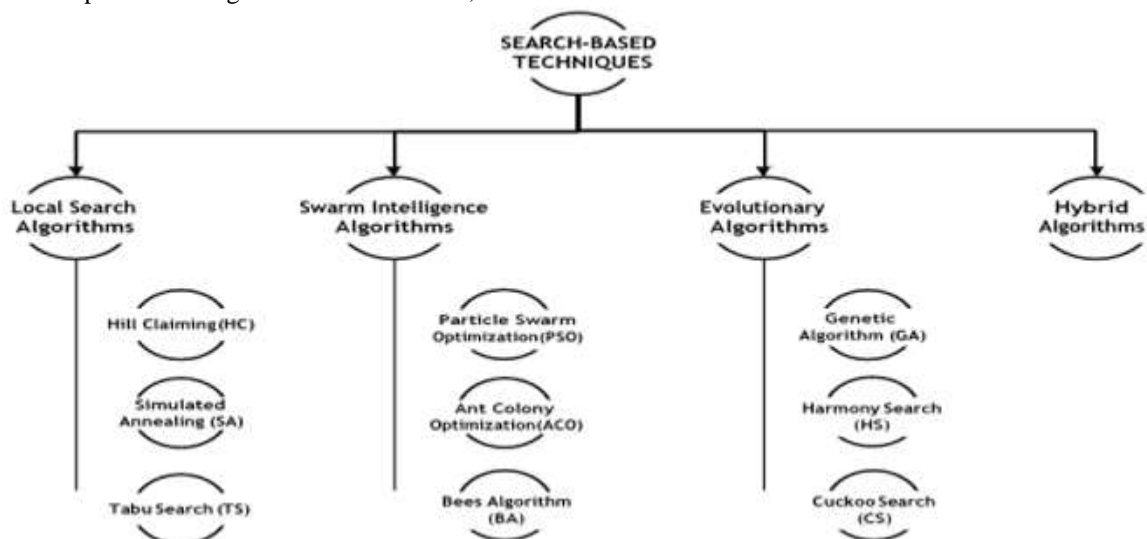
2. METODOLOGI PENELITIAN

Pada dekade 19 tahun terakhir, sebuah jenis baru dari algoritma aproksimasi telah muncul dan mencoba menggabungkan metode heuristik dasar dalam metode pencarian atau optimisasi untuk menjelajahi ruang pencarian, dan metode ini biasa disebut dengan metode metaheuristik (Alazzam and Iii 2013). Metode metaheuristik pertama sekali diperkenalkan pada tahun 1986, dan digambarkan sebagai alternatif strategi pencarian yang sangat baik, diatas ruang pencarian dengan harapan bisa menemukan hasil yang optimal. Banyak algoritma yang sudah diusulkan oleh para peneliti dalam literatur, misalnya metode *ant colony optimization*, (Blum and Roli 2003). namun sampai saat ini belum ada defenisi yang diterima secara umum dan menjadi rujukan standar untuk istilah *metaheuristic*, akan tetapi metaheuristik biasanya dijadikan sebagai strategi tingkat tinggi yang mengarahkan masalah yang mendasarinya. Beberapa ahli telah mecoba memberikan defenisi yang diusulkan antara lain “Metaheuristik secara formal didefinisikan sebagai proses pembangkit berulang yang memandu heuristik bawahan dengan menggabungkan konsep yang berbeda secara cerdas untuk menjelaskan dan mengeksplorasi ruang pencarian, strategi pembelajaran digunakan untuk menyusun informasi agar dapat ditemukan secara efisien di dekat Solusi yang optimal.” [Osman dan Laporte 1996].

Dalam 40 tahun terakhir, banyak algoritma meta-heuristik yang berguna telah dikembangkan dalam literatur (misalnya TS (Glover, 1989), SA (Kirkpatrick, 1984), GA (Holland, 1992), ACA (Dorigo et al., 2008) , PSO (Kennedy, 2011), Differential Evolution (DE) (Feoktistov, 2006), HS (Lee & Geem, 2005), FPA (Yang, 2012), Sine Cosine Algorithm (SCA) (Mirjalili, 2016), BA (Pham et al., 2011a), CS (Yang & Deb, 2009) dan Firefly Algorithm (FA) (Yang, 2010), untuk beberapa nama).

Dalam literatur, ada algoritma meta-heuristik yang berbeda yang telah digunakan dalam SBSE. Algoritma ini dapat diklasifikasikan ke dalam empat kategori: pencarian lokal, kecerdasan swarm, evolusioner, dan algoritma hibrida seperti yang ditunjukkan pada Gambar 2.1 (Malhotra et al., 2016). Seperti namanya, algoritma pencarian lokal sering bias terhadap eksploitasi. Secara khusus, algoritma pencarian lokal biasanya mengeksplorasi pengetahuan pada tetangganya untuk mendapatkan solusi yang lebih baru. Algoritma kecerdasan Swarm menggunakan perilaku kolektif dari populasi sebagai cara untuk mengeksplorasi dan mengeksplorasi ruang pencariannya. Algoritma evolusioner sering memanfaatkan evolusi biologis seperti reproduksi, mutasi, rekombinasi dan seleksi untuk mendapatkan solusi yang lebih baru. Akhirnya, algoritma hibrid mengadopsi kombinasi dari algoritma lain sehingga memiliki cara yang beragam untuk mendapatkan solusi yang lebih baru. Sejalan dengan lingkup paper ini, bagian 2.6 memberikan rincian tentang algoritma meta-heuristik tersebut yang telah digunakan untuk strategi t-way.

Metaheuristik biasanya merupakan strategi tingkat tinggi yang membimbing masalah mendasari spesifik heuristik untuk meningkatkan kinerjanya yang tujuan utamanya untuk menuju perbaikan iteratif dan mendapatkan pengembangan algoritma. Banyak pendekatan metaheuristik bergantung keputusan probabilistik yang dilakukan selama pencarian namun perbedaan utama dari pencarian dengan metode metaheuristik ialah dilakukan dengan cerdas "[Stu] tze 1999]. berikut gambar klasifikasi dan proses dari algoritma metaheuristik ;



Gambar 1. Algoritma Metaheuristik

Jadi dapat dikatakan metaheuristik adalah proses master yang berulang yang memandu dan memodifikasi operasi heuristik bawahan untuk menghasilkan solusi berkualitas tinggi secara efisien. Ini dapat memanipulasi solusi tunggal yang lengkap (atau tidak lengkap) atau kumpulan larutan pada setiap iterasi berbeda dengan metode Heuristik adalah teknik yang dirancang untuk memecahkan masalah dengan mengabaikan apakah solusi dari masalah tersebut dapat dibuktikan benar atau tidak, tetapi biasanya menghasilkan solusi yang baik atau memecahkan masalah dengan lebih mudah, cepat dan sederhana. Teknik heuristik tidak memiliki algoritma pencarian solusi optimum yang pasti tetapi memiliki kaidah yang dapat mengeksplorasi ruang pencarian yang paling menjanjikan, yaitu ruang terdapatnya solusi optimum atau mendekati optimum (Thierauf dan Klekamp: 1975).

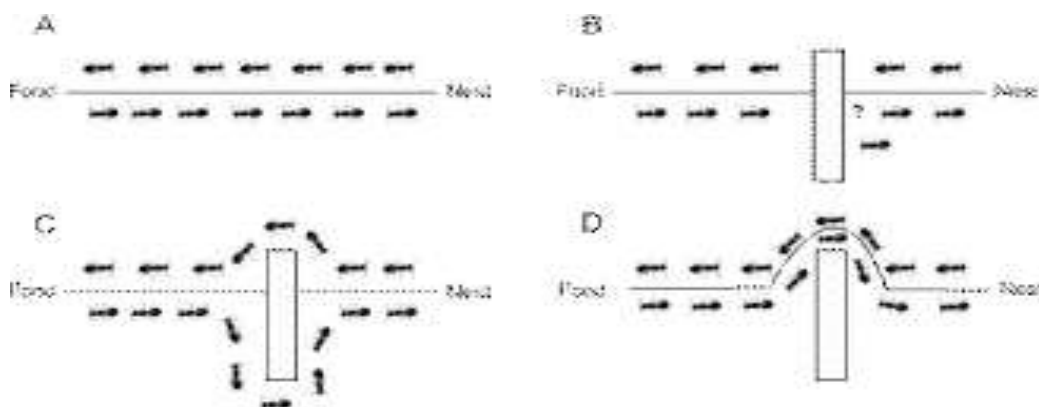
Beberapa algoritma yang hadir dengan metode metaheuristik dalam penyelesaian kombinatorial ini antara lain ialah algoritma *Ant Colony*, yang biasa disebut juga dengan perilaku koloni semut yang diperkenalkan oleh Dorigo M, dan Gambardella pada tahun 1996, secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang ke tempat-tempat sumber makanan. Koloni semut dapat menemukan rute terpendek antara sarang dan sumber makanan berdasarkan jejak kaki pada lintasan yang telah dilalui. Semakin banyak semut yang melalui suatu lintasan, maka akan semakin jelas bekas jejak kakinya. Hal ini akan menyebabkan lintasan yang dilalui semut dalam jumlah sedikit, semakin lama akan semakin berkurang kepadatan semut yang melewatinya, atau bahkan akan tidak dilewati sama sekali. Sebaliknya, lintasan yang dilalui semut dalam jumlah banyak, semakin lama akan semakin bertambah kepadatan semut yang melewatinya, atau bahkan semua semut akan melalui lintasan tersebut, setiap semut akan membuat jalur masing – masing sampai kembali ketempat semula dimana mereka ditempatkan pertama kali. Jika sudah mencapai keadaan ini, maka semut telah menyelesaikan sebuah siklus yang akhirnya menemukan jalur terpendek.

2.1 Algoritma *Ant Colony*

Algoritma ACO telah banyak diaplikasikan dalam berbagai bidang yang mencakup beberapa persoalan, yaitu :

1. Traveling Salesman Problem (TSP)
Untuk mencari rute terpendek dalam sebuah graph menggunakan rute Hamilton.
2. Quadratic Assignment Problem (QAP)
Untuk menugaskan sejumlah resources untuk ditempatkan pada sejumlah m lokasi dengan meminimalisasi biaya penugasan (assignment).
3. Job-shop Scheduling Problem (JSP) Merupakan salah satu contoh aplikasi Ant Colony Optimization, yaitu untuk mencari lintasan sejumlah pekerjaan menggunakan sejumlah m mesin demikian sehingga seluruh pekerjaan diselesaikan dalam waktu yang seminimal mungkin.
4. Vehicle Routing Problem (VRP)
5. Pengaturan rute kendaraan
6. Implementasi pada jaringan komunikasi
7. Network routing, dll.

Berikut gambar ilustrasi penyelesaian dari algoritma *ant colony* ;



Gambar 2. Contoh Optimasi *Ant Colony*

2.2 Algoritma Tabu Search

Tabu Search berasal dari Tongan, suatu bahasa Polinesia yang digunakan oleh suku Aborigin Pulau tonga untuk mengindikasikan suatu hal yang tidak boleh “disentuh” karena sakralnya. Menurut kasus Webster, Tabu berarti larangan yang dipaksakan oleh kebudayaan sosial sebagai suatu tindakan pencegahan atau sesuatu yang dilarang karena berbahaya, istilah “Bahaya” yang harus dihindari dalam *Tabu Search* adalah penjadwalan yang tidak layak, dan terjebak tanpa ada jalan keluar, dalam konteks lebih luas.

Tabu Search adalah sebuah metode optimasi yang berbasis pada *local search* yang merupakan proses pencarian bergerak dari satu solusi ke solusi berikutnya, dengan cara memilih solusi terbaik *neighbourhood* solusi sekarang (current) yang tidak tergolong solusi terlarang (tabu). Ide dasar dari algoritma tabu search adalah mencegah proses pencarian dari *local search* agar tidak melakukan pencarian ulang pada ruang solusi yang sudah pernah ditelusuri, dengan memanfaatkan suatu struktur memori yang mencatat sebagian jejak proses pencarian yang telah dilakukan. Adapun struktur memori fundamental

dalam *tabu search* dinamakan *tabu list* yang menyimpan atribut dari sebagian move (transisi solusi) yang telah diterapkan pada iterasi-iterasi sebelumnya, oleh karena itu, dengan mengevaluasi solusi, solusi yang baik diperlukan untuk dapat ditemukan dalam waktu yang dapat diterima, untuk masalah seperti itu, mengevaluasi solusi dalam waktu yang dapat diterima memiliki arti yang sama dengan mengevaluasi "beberapa solusi" di seluruh ruang solusi. Pemilihan beberapa solusi tergantung pada apa dan bagaimana dipilih berubah sesuai *metaheuristik*. Tidak dapat dijamin bahwa solusi optimal termasuk dalam solusi yang terlibat dalam evaluasi, karenanya solusi diusulkan oleh *metaheuristik* metode untuk masalah optimasi, harus dianggap sebagai solusi yang baik, bukan sebagai solusi optimal (Cura;2008)

Mereka adalah kriteria atau metode komputer yang diidentifikasi untuk memutuskan yang efektif dari berbagai gerakan alternatif untuk mencapai tujuan atau sasaran apa pun. Algoritma semacam ini memiliki properti konvergensi, tetapi mereka tidak dapat menjamin solusi yang tepat, mereka hanya dapat menjamin solusi yang mendekati solusi yang tepat. mengapa algoritma *metaheuristik* diperlukan adalah karena :

- Masalah optimasi dapat memiliki struktur bahwa proses pencarian solusi yang tepat tidak dapat ditentukan.
- Algoritma *Metaheuristik* bisa jauh lebih sederhana dari titik pengambil keputusan, dalam hal komprehensibilitas.
- Algoritma *Metaheuristik* dapat digunakan sebagai bagian dari proses menemukan solusi yang tepat, dan tujuan pembelajaran.
- Umumnya, bagian yang paling sulit dari masalah dunia nyata (tujuan dan batasan mana yang harus digunakan, alternatif mana yang harus diuji, bagaimana masalah data harus dikumpulkan) diabaikan dalam definisi yang dibuat dengan rumus matematika. Kerusakan data yang digunakan dalam proses penentuan parameter model dapat menyebabkan kesalahan yang jauh lebih besar daripada solusi sub-optimal yang dihasilkan oleh pendekatan *metaheuristik* (Karabog̃a 2011).

3. ANALISA DAN PEMBAHASAN

Pada pembahasan di atas, maka dapat di analisis dari perbandingan algoritma di atas, maka didapatkan penyelesaian pendekatan model *metaheuristik* dengan perhitungan sebagai berikut ;

3.1 Model dan Pengujian Optimalisasi

Setelah titik layak yang telah diperoleh (pers. 6) perlu di uji apakah titik ini telah optimal terhadap persoalan (PL) atau belum. Pandang fungsi objektif (PL)

$$Z = C^T X$$

Vektor C dan x di partisi menurut matriks basis (B) dan tak basis (N)

$$\text{Jadi } Z = (C_B \ C_N) \begin{pmatrix} X_B \\ X_N \end{pmatrix} \quad (7)$$

Kalikan , diperoleh

$$Z = C_B X_B + C_N X_N \quad (8)$$

Substitusi pers. (5) untuk X_B , diperoleh

$$Z = C_B (B^{-1} b - B^{-1} N X_N) + C_N X_N \quad (9)$$

$$Z = C_B B^{-1} b - C_B B^{-1} N X_N + C_N X_N \quad (10)$$

$$Z = C_B B^{-1} b - (C_B B^{-1} N - C_N) X_N \quad (11)$$

Andaikan $Z_j = C_B B^{-1} N, \forall j \in N$

Dan $C_N = C_i \forall j \in N$

$$\text{Jadi } Z_j - C_j = C_B B^{-1} N - C_N \quad (12)$$

Jika $Z_j - C_j < 0$ (negatif), himpunan vektor X_N di naikkan dari batasannya 0 maka nilai Z akan bertambah untuk persoalan (PL), berarti nilai fungsi objektif dapat naik. Dengan kata lain titik X_B yang telah diperoleh belum menghasilkan penyelesaian maksimum.

Namun

Jika $Z_j - C_j \geq 0$, X_N di naikkan dari batasannya 0 akan mengakibatkan nilai Z berkurang (mengecil) atau tetap.

Artinya titik X_B yang telah di peroleh sudah merupakan titik maksimum ,

Jadi syarat maksimum terhadap persoalan (PL) telah diperoleh yaitu :

$$Z_j - C_j \geq 0 \forall j \text{ (non basis)} \quad (13)$$

Titik layak untuk CO

Perhatikan bahwa nilai titik layak X_B yang dinyatakan oleh persamaan (6) adalah untuk persoalan (PL) .

Sekarang pandang persoalan (PO)

Kendala $x \in \{0,1\}$ dapat dinyatakan dalam bentuk

$$x \geq 0$$

$$x \leq 1$$

Jadi persoalan (PO) dapat di tulis sebagai

$$\text{Maksimumkan } Z = C^T X$$

Kendala $Ax = b$

$$x \geq 0$$

$$x \leq 1$$

seperti persoalan (PL), nilai titik layak juga mengambil bentuk seperti pers (6) yaitu :

$$X_B = B^{-1} b - B^{-1} N X_N$$

$$X_B = \beta - \alpha X_N \quad (14)$$

Dengan $P = B^{-1} b$ dan $\alpha = B^{-1} N X_N$

Jika pada persoalan (PL) X_N bernilai 0 karena seperti x , dengan kata lain nilai variabel terbesar X_N berada di titik batasannya yaitu 0.

Pada persoalan kombinatorial (POL) terdapat

$$x \geq 0$$

$$x \leq 1$$

Jadi dalam kasus ini nilai variabel tak basis X_N adalah 0 atau 1, perhatikan lagi persamaan (4) terlihat bahwa nilai variabel tersebut basis X_N sudah bernilai 0 atau 1 berarti nilai x_n telah biner.

Jika semua komponen dari vektor β telah bernilai 0 atau 1 berarti penyelesaian layak terhadap persoalan (PO) telah diperoleh. Jika komponen dari vektor β masih ada yang tidak 0 atau 1 berarti penyelesaian layak terhadap persoalan (PO) belum diperoleh.

Prosedur memperoleh nilai biner dilakukan sebagai berikut :

1. Pisahkan himpunan variabel basis I_1 menjadi himpunan I_2 , variabel tersebut basis yang berada di batasannya 0 atau 1 dan himpunan I_1 pada himpunan I_2, I_3, I_3 .
2. Lakukan pencarian pada fungsi objektif dengan mempertahankan variabel tak basis I_1 dan hanya perubahan diskrit pada nilai variabel dalam himpunan I_2 .
3. Pada penyelesaian yang diperoleh di langkah 2, periksa nilai $Z_j - C_j$ dari persamaan (12) terhadap variabel dalam himpunan I_1 Jika ada yang dapat di gerakkan dari batasannya, tambahkan ke himpunan I_2 , ulangi dari langkah langkah jika ini tidak berhenti. Hasil dari proses prosedur tadi adalah bahan semua komponen dari Vektor β telah bernilai biner, sehingga di peroleh penyelesaian layak biru untuk persoalan (PO).

4. KESIMPULAN

Adapun kesimpulan dari hasil penelitian ini adalah sebagai berikut ;

1. Penelitian ini dapat dikembangkan dalam penyelesaian optimisasi deterministik dengan pengujian dan pembuktian analitik dan eksak.
2. Dalam penyelesaian optimisasi lainnya pendekatan metode metaheuristik perlu diperhatikan kompleksitas permasalahan, *scala input*, struktur input dan waktu dalam penyelesaian masalah tersebut.
3. Model ini dihasilkan dari *trial* dan *error* sehingga membutuhkan waktu yang lama, oleh karena itu diharapkan setelah di dapat metode ini untuk penyelesaian dalam optimisasi berikutnya dapat dilakukan dengan cepat dan efektif.
4. Perlu dilakukan riset lanjutan untuk membangun sebuah pengembangan metode *Artificial Intelligence* dengan konsep dan metode ini, sehingga memudahkan dan membantu pekerjaan manusia. serta menambah khazanah kelimuan di dunia Kelimuan Komputer.
5. Model ini dapat menyelesaikan persoalan optimisasi secara efisien, dengan waktu yang cepat dan dengan penyelesaian terbaik (optimum).

REFERENCES

- [1] Doef, V. D, and M. Cappendijk. 2006. *Veiligheid Nederland in Kaart-Modelling en Analyse Van Evacuatie Dijkkringen 7, 14 en 36*. Rijkswaterstaat
- [2] Drezner, Z. and G. Wesolowsky. 1997. Selecting an optimum configuration of one-way and two-way routes. *Transportation Science* 31(4): 386-394.
- [3] Duran M. A. and I. E Grossmann, An Outer-Approximation Algorithm For a class of Mixed-Integer Nonlinear Programs, *Mathematical Programming* 36 (1986) 307.
- [4] El-Sbayti, H. H. 2008. *Optimal Scheduling of Evacuation Operations With ContraFlow*. University of Maryland
- [5] Fernandes F P, Costa M F P, Rocha A M A C, Fernandes E M G P. 2016. Improving efficiency of a multistart with interrupted Hooke-and-Jeeves filter search for solving MINLP problems. *Int. Conf. on Computational Sciences and its Applications*, pp. 345-358.
- [6] Flores-Tlacuahuac A and L. T. Biegler, Simultaneous mixed-integer dynamic optimization for integrated design and control, *Computers and Chemical Engineering*, 31 (2007), pp. 648–656.
- [7] Ford, L.R. and D.R. Fulkerson. 2000. *Flows in Networks*. Princeton Univ. Press
- [8] Fischetti M., F. Glover, A. Lodi. The Feasibility Pump. *Mathematical Programming A* 104(1), 91-104 (2005).
- [9] G. Nannicini and P. Belotti, Local Branching for MINPs. *Technical Report workingpaper*, CMU, 2009.
- [10] G. Nannicini, P. Belotti. Rounding-based heuristics for non-convex MINLPs. In: P. Bonami, L. Liberti, A. Miller, A. Sarteneur (eds). *Proceedings of the European Workshop on MINLP. CIRM, Marseille, France* (2009).
- [11] Kaya O, Urek B. (2016). A mixed integer nonlinear model and heuristic solutions for location, inventory and pricing decisions in a closed loop supply chain. *Competers & Operations Research*, 65, 93-103.
- [12] Kim J S, and Edgar T F. (2014) Optimal scheduling of combined heat and powerplants using mixed-integer nonlinear programming, *Energy*, 77, 675-690
- [13] M. A. Duran and I. E Grossmann, An Outer-Approximation Algorithm For a class of Mixed-Integer Nonlinear Programs, *Mathematical Programming* 36 (1986) 307.
- [14] M. Fischetti, F. Glover, A. Lodi. The Feasibility Pump. *Mathematical Programming A* 104(1), 91-104 (2005).
- [15] Mawengkang H. and B. A Murtagh, Solving Nonlinear Integer Programs with Large-Scale Optimization Software. *Annals of Operations Research* 5425-437, 1986
- [16] Madireddy, M., D.J. Medeiros, and S. Kumara. 2011. An Agent Based Model for Evacuation Traffic Management. *Proceedings of the 2011 Winter Simulation Conference*: pp. 222-233
- [17] Maniezzo V., T. Stutzle, S. Voß (Eds.), *Matheuristics*, Vol. 10 of *Annals of Information Systems*, Springer Verlag, Berlin, Germany, 2010.
- [18] Nannicini G. and P. Belotti, Local Branching for MINPs. *Technical Report workingpaper*, CMU, 2009.



- [19] Nannicini, G.P. Belotti. Rounding-based heuristics for non-convex MINLPs. In: P. Bonami, L. Liberti, A. Miller, A. Sarteneau (eds). *Proceedings of the European Workshop on MINLP. CIRM, Marseille, France (2009)*.
- [20] Gupta O.K. and A. Ravindran (1985). Branch and Bound Experiments in Convex Nonlinear Integer Programming, *Manage Sci.*, 31 (12), 1533–1546