

HIBRIDISASI ALGORITMA GENETIKA DENGAN *VARIABLE NEIGHBORHOOD SEARCH* (VNS) PADA OPTIMASI BIAYA DISTRIBUSI

Asyrofa Rahmi¹, Wayan Firdaus Mahmudy², Syaiful Anam³

^{1,2}Fakultas Ilmu Komputer, Universitas Brawijaya

³Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Brawijaya

¹asyrofarahmi@gmail.com, ²wayanfm@ub.ac.id, ³syaiful@ub.ac.id

(Naskah masuk: 25 Januari 2017, diterima untuk diterbitkan: 7 Mei 2017)

Abstrak

Proses distribusi dianggap sangat penting bagi perusahaan karena menjadi salah satu faktor yang mempengaruhi perolehan keuntungan. Besarnya biaya yang dikeluarkan serta kompleksnya permasalahan dalam proses distribusi menjadikan permasalahan distribusi sebagai topik yang perlu diteliti lebih mendalam lagi. Karena algoritma genetika (AG) sudah terbukti mampu memberikan solusi terbaik pada berbagai macam permasalahan optimasi dan kombinatorial, maka algoritma ini digunakan untuk menyelesaikan permasalahan distribusi pada penelitian ini. Namun, penerapan GA klasik memiliki kekurangan yaitu belum mencapai titik optimum global sehingga perlu dihibridisasi menggunakan algoritma *variable neighborhood search* (VNS). Algoritma ini dipilih karena selain mencari solusi secara global, algoritma ini juga mencari solusi secara lokal sehingga mampu menutupi kekurangan dari GA. Dengan menggunakan hibridisasi GA dengan VNS maka biaya yang diperoleh adalah 32392960 yang dibuktikan dengan penghematan biaya sebesar 323190 jika dibandingkan dengan GA klasik yaitu 32716150. Namun, dilihat dari waktu komputasi, GA-VNS membutuhkan waktu yang relatif sama dengan GA klasik yaitu 279332 ms (*milisecond*) dan 265091 ms.

Kata kunci: *distribusi, algoritma genetika, variable neighborhood search*

Abstract

The distribution process is considered importantly for the company as one of the factors that affects profitability. The costs incurred as well as the complexity of the distribution problems makes the distribution problems as a topic that need to be examined more deeply. Since the wide range of combinatorial and optimization problems have been ever solved by using genetic algorithm (GA) well then it is used to resolve the distribution problems in this study. However, the implementation of classical GA has the disadvantage that has not yet reached the global optimum so that needs to be hybridized by using variable neighborhood search (VNS) algorithm. The VNS algorithm has been chosen because its ability either to search the global solutions or local solutions. The local search of VNS algorithm is able to cover the shortage of the GA. By using hybridization of GA with VNS, the cost accrued is 32392960 as evidenced by cost savings of 323190 in comparison with the classical GA is 32716150. However, the computational time of GA-VNS is equal to its classical GA relatively.

Keywords: *distribution, genetic algorithm, variable neighborhood search*

1. PENDAHULUAN

Kejelian para pengusaha sangat diuji dalam mengolah strategi perusahaan di tengah maraknya pasar global pada dekade ini. Hal ini ditujukan agar perusahaan mereka tetap memperoleh keuntungan dan masih mampu bersaing dengan perusahaan-perusahaan pesaingnya. Sebagai bagian dari *supply chain*, proses meminimalkan biaya dari proses distribusi merupakan salah satu faktor yang tidak bisa dihindarkan karena proses distribusi dengan cakupan wilayah yang luas membutuhkan biaya yang tidak sedikit. Proses distribusi atau pengiriman barang dari pabrik produksi agar sampai ke konsumen harus melalui beberapa distributor lokal di beberapa wilayah (*multi-level*) seperti *distributor centre, retailer, agen, etc.* (Guo dkk. 2015). Banyaknya kemungkinan jaringan distribusi yang

terbentuk untuk mendapatkan biaya minimal menjadi permasalahan yang perlu diperhitungkan. Penentuan jaringan distribusi akan lebih kompleks ketika suatu perusahaan memproduksi lebih dari satu jenis produk (*multi-produk*) (Gicquel & Minoux 2015; Kim & Lee 2016) selain itu, pemesanan produk oleh konsumen memiliki jumlah yang tidak pasti dengan jumlah jenis produk yang berbeda-beda (Langroodi & Amiri 2016).

Beberapa cara sudah pernah dicoba untuk menyelesaikan permasalahan distribusi salah satunya adalah menggunakan pemrograman matematika dengan memodelkan permasalahan ke dalam integer (Sitek & Wikarek 2012). Pada penelitian tersebut, model distribusi yang diperhitungkan adalah dua level yaitu menggunakan perantara pusat distributor untuk mengirim barang ke konsumen. Pencarian biaya yang minimal dan

optimal diselesaikan menggunakan aplikasi LINGO dan CPLEX sehingga aliran distribusi yang dicari dapat ditemukan yang ditunjukkan oleh beberapa variabel keputusan keluaran dari aplikasi. Keunggulan lainnya adalah menyediakan beberapa kombinasi yang memungkinkan dari analisis kapasitas yang tersedia pada distributor dan jumlah unit transportasi sehingga mendukung keputusan yang tepat dalam permasalahan distribusi. Namun, kelemahannya adalah tidak bisa dilakukan dalam proses nyata karena bentuk standar dari *linear programming* dan ukuran model yang diimplementasikan dalam penelitian ini. Kelemahan lainnya adalah pendekatan ini tidak mendukung proses nalar yang rasional.

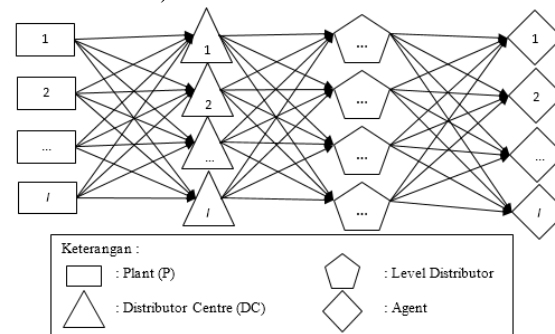
Berbagai permasalahan kombinatorial dengan kompleksnya beberapa *constraint* yang digunakan (Gupta dkk. 2012; Chen & Wang 2011) sudah teruji mampu diselesaikan menggunakan algoritma genetika termasuk permasalahan distribusi (Rahmi dkk. 2016). Permasalahan yang diangkat adalah distribusi banyak (multi) level dengan jenis produk berjumlah satu. Untuk mendapatkan jaringan distribusi yang tepat dengan biaya yang minimal maka beberapa *constraint* utama diperhitungkan. Dibandingkan dengan hasil keluaran *random search*, hasil solusi algoritma genetika yang diperoleh adalah mendekati optimal. Dalam penelitian awal tersebut, algoritma genetika (AG-awal) diimplementasikan dengan menggunakan *extended intermediates* sebagai model *crossover*, *insertion* sebagai model *mutation* dan *roulette-wheel* sebagai model *selection*.

Penelitian lanjutan dari AG-awal adalah menguji permasalahan distribusi yang sama dengan mengimplementasikan model operator algoritma genetika (AG-lanjut) yang berbeda yaitu menggunakan *crossover one-cut-point*, *swapmutation* dan *elitism selection* (Sarwani dkk. 2016). Hasil dapat dilihat dari eksekusi yang dilakukan sebanyak 10 kali untuk mengetahui seberapa jauh pengacakan yang dilakukan algoritma genetika baik AG-awal maupun AG-lanjut dengan hasil *random search*. Berdasarkan nilai *fitness* dari 10 eksekusi, didapatkan bahwa AG-awal dibandingkan *random search* memiliki hasil akhir yang tidak terlalu jauh perbedaannya. Hal ini dikarenakan model operator yang digunakan dalam algoritma genetika memberikan hasil yang konvergensi dini sehingga kurang bereksplorasi dalam mencari solusi sedangkan hasil AG-lanjut dibandingkan *random search* menunjukkan hasil perbedaan yang cukup signifikan. Namun, secara keseluruhan hasil AG-lanjut lebih berkualitas dibandingkan dengan AG-awal sehingga dapat disimpulkan bahwa berdasarkan permasalahan distribusi pada kedua penelitian tersebut (AG-awal dan AG-lanjut), algoritma genetika dengan operator *crossover one-cut-point*, *swapmutation* dan *elitism selection* sangat direkomendasikan.

Meskipun menggunakan model operator yang berbeda, penggunaan algoritma genetika yang masih dasar (Boudjelaba dkk. 2014) memberikan perbedaan yang cukup signifikan pada 10 hasil eksekusi AG-lanjut (Rahmi dkk. 2016). Hal ini menunjukkan bahwa dia tidak secara keseluruhan terjebak dalam konvergensi dini karena masih bisa bereksplorasi lebih jauh dalam mencari solusi. Disamping itu, perolehan hasil tersebut dapat diperbaiki sampai hasil yang diberikan masing-masing eksekusi memiliki perbedaan yang tidak terlalu signifikan sehingga perlu dihibridisasi dengan algoritma pencari global dan pencari lokal seperti *variable neighborhood search* agar bisa memberikan hasil yang lebih baik (Mahmudy dkk. 2014).

2. METODOLOGI

Proses distribusi atau pengiriman barang dari pabrik produksi agar sampai ke konsumen harus melalui beberapa distributor lokal di beberapa wilayah (multi-level) seperti *distributor centre, retailer, agen*, etc. (Guo dkk. 2015). Banyaknya kemungkinan jaringan distribusi yang terbentuk untuk mendapatkan biaya minimal menjadi permasalahan yang perlu diperhitungkan. Penentuan jaringan distribusi akan lebih kompleks ketika suatu perusahaan memproduksi lebih dari satu jenis produk (multi-produk) (Gicquel & Minoux 2015; Kim & Lee 2016) selain itu, pemesanan produk oleh konsumen memiliki jumlah yang tidak pasti dengan jumlah jenis produk yang berbeda-beda (Langroodi & Amiri 2016)



Gambar 1. Jaringan Distribusi

Berdasarkan Gambar 1, model distribusi multi-level memiliki banyak kemungkinan solusi untuk membentuk jaringan distribusi yang tepat yang ditandai dengan minimalnya biaya yang dikeluarkan (Sitek & Wikarek 2012; Rahmi dkk. 2016; Sarwani dkk. 2016). Untuk meminimalkan biaya distribusi maka perlu disusun formulasi matematis dalam beberapa variabel untuk pencarian biaya yang minimal beserta beberapa *constraint* utama yang akan mempengaruhi jaringan distribusi (Sitek & Wikarek 2012).

Data pada penelitian ini merupakan data rancangan yang diperoleh proses wawancara dengan beberapa pakar distribusi yang menjadi ahli dan bekerja di perusahaannya masing-masing. Data

disesuaikan dengan permasalahan pada penelitian ini sehingga data yang digunakan adalah data kapasitas stok setiap produk untuk setiap level, kapasitas kendaraan, jumlah kendaraan, biaya tetap kendaraan (Sarwani dkk. 2016; Rahmi dkk. 2016), serta biayadari beberapa produk untuk setiap level distributor.

Agar permasalahan distribusi mampu diselesaikan dengan baik maka perlu dibentuk formulasi matematika baik untuk fungsi tujuan maupun *constraint* yang mengikutinya. Pertama adalah membentuk formulasi untuk menjabarkan fungsi tujuan dari distribusi yaitu biaya yang paling minimal sehingga untuk pencarian biaya, fungsi digunakan ada pada persamaan (1) yang terdiri dari beberapa variabel.

$$z = \sum_{i=0}^I \left[\sum_{j=0}^J \sum_{k=0}^{K_j} \sum_{r=0}^R \sum_{p=0}^{P_j} ((X_{ijkp} Cvb_{ijp}) + Cf_{ijk}) St_{ijk} \right] \quad (1)$$

dimana $i \in \{1, 2, \dots, I\}$ adalah level pengiriman distribusi, $j \in \{1, 2, \dots, J\}$ adalah unit distributor yang berperan sebagai pengirim sedangkan $r \in \{1, 2, \dots, R\}$ sebagai pemesan. $p \in \{1, 2, \dots, P\}$ adalah jenis produk yang dipesan dan $k \in \{1, 2, \dots, K\}$ adalah kendaraan yang dimiliki oleh unit distributor pengirim j untuk mendistribusikan produknya. X_{ijkp} merepresentasikan jumlah unit barang yang akan dikirim, Cvb_{ijp} merupakan biayavariabel untuk setiap produk, Cf_{ijk} merupakan biaya tetap pengiriman, dan St_{ijk} merupakan status bahwa distributor pada level i melakukan pengiriman atau tidak dengan kendaraan k nya $\{0,1\}$.

Setelah mengetahui formulasi matematika dari fungsi tujuan permasalahan distribusi maka selanjutnya adalah membentuk formulasi matematika untuk *constraint* yang harus dipenuhi sehingga solusi yang dihasilkan tidak hanya biaya yang minimal saja namun juga memenuhi *constraint* yang ada sehingga permasalahan distribusi bisa terselesaikan dengan benar.

Fungsi constraint permintaan order

Constraint pertama ini berhubungan dengan batasan jumlah order dari pemesan sehingga jumlah barang yang dipesan oleh pemesan (Or_{irp}) harus sama dengan barang yang dikirimkan nantinya kepada pemesan. Fungsi *constraint* untuk jumlah order ditunjukkan pada persamaan (2).

$$\sum_{j=0}^J \sum_{k=0}^{K_j} \sum_{r=0}^R \sum_{p=0}^{P_j} X_{ijkp} = Or_{irp} \quad (2)$$

Fungsi constraint kapasitas kendaraan untuk setiap unit distributor

Constraint yang kedua adalah pada kapasitas kendaraan yang digunakan. Suatu barang dikirim ke pemesan maupun level dibawahnya kemudian barang akan diangkut oleh kendaraan. Permasalahannya

adalah kapasitas kendaraan dari setiap distributor pengirim (Cp_{ijp}) memiliki batasan kapasitas yang tidak boleh dilanggar. Fungsi *constraint* untuk kapasitas kendaraan ditunjukkan pada persamaan (3).

$$\sum_{j=0}^J \sum_{k=0}^{K_j} \sum_{r=0}^R \sum_{p=0}^{P_j} X_{ijkp} \leq Cp_{ijp} \quad (3)$$

Fungsi constraint persediaan barang untuk setiap produk pada unit distributor pengirim

Constraint yang terakhir adalah *constraint* untuk ketersediaan stok pada setiap unit distributor pengirim (Kcp_{ijkp}). Setiap distribusi selalu memiliki stok ketersediaan barang. Ketika suatu distributor berperan sebagai pengirim, maka dilakukan pengecekan stok bahwa jumlah barang yang akan dikirim tidak boleh melebihi stok. Fungsi *constraint* untuk stok unit distribusi pengirim ditunjukkan pada persamaan (4)

$$\sum_{j=0}^J \sum_{k=0}^{K_j} \sum_{r=0}^R \sum_{p=0}^{P_j} X_{ijkp} \leq Kcp_{ijkp} \quad (4)$$

3. VARIABLE NEIGHBORHOOD SEARCH (VNS)

VNS merupakan algoritma yang menerapkan proses pencarian solusi ketetanggaan yang jauh dan dekat sehingga memberikan kemungkinan besar menghasilkan solusi yang unggul. Pencarian solusi pada ketetanggaan yang jauh diartikan sebagai pencarian dengan sistem *random* sehingga bisa menjangkau cakupan masalah dengan skala yang besar. Setelah dilakukan pengacakan dan didapatkan solusi pada suatu area tertentu maka dilanjutkan dengan pencarian ketetanggaan yang dekat yang biasa dikenal sebagai pencarian lokal. Dengan adanya pencarian lokal tersebut, maka solusi paling unggulnya akan ditemukan (Mladenovic dkk. 2016).

Dengan kemampuan yang dimiliki VNS dalam mencari solusi tersebut maka tidak terjebak dalam solusi optimum lokal dan bisa mencapai solusi optimum global menjadi keunggulannya (Castelli & Vanneschi 2014). Gambar 2 merupakan tahapan umum dalam proses VNS (Cheikh dkk. 2015). Algoritma ini dipertimbangkan karena pernah diterapkan untuk menyelesaikan masalah transportasi (Ziebuhr & Kopfer 2016).

4. HIBRIDISASI ALGORITMA GENETIKA DAN VNS (GA-VNS)

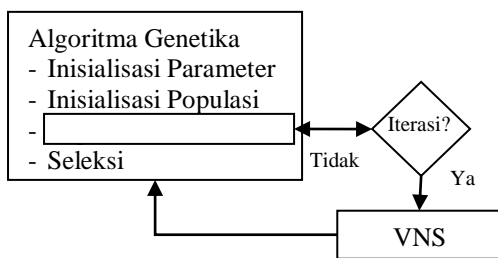
Pada penelitian ini, algoritma genetika (GA) digunakan untuk menyelesaikan permasalahan distribusi multi-level multi-produk. GA merupakan metode yang mengadopsi sifat alami biologi individu pada seleksi alam (Qiongbing 2016) Terdapat beberapa langkah dalam menyelesaikan permasalahan menggunakan GA yaitu dimulai dari melakukan representasi kromosom, reproduksi,

evaluasi dan seleksi. Proses evaluasi pada GA digunakan untuk mengukur seberapa baik sebuah individu untuk dapat memberikan solusi dengan memperhatikan nilai *fitness* sehingga tingginya nilai *fitness* yang dihasilkan oleh sebuah individu maka pada generasi selanjutnya semakin tinggi kemungkinannya terpilih sebagai sebuah solusi (Thakur & Kumar 2016). Arsitektur hibridisasi GA-VNS ditunjukkan pada Gambar 3.

```

PROCEDURE VNS
INPUT :
    xtemp, lgmax
OUTPUT :
    x
x sebagai solusi awal;
Atur ketetanggan dengan lg={1,2,...,lgmax}
lg = 1;
WHILE lg < lgmax do
    xtemp = Acak x
    xlokal = Local(xtemp)
    IF (f(xlokal) > f(x)) then
        x = xlokal
        lg = 1
    ELSE
        lg = lg+1;
    ENDIF
END WHILE
Mengembalikan nilai x sebagai solusi terbaik
    
```

Gambar 2. Pseudocode algoritma VNS



Gambar 3. Arsitektur hibridisasi GA-VNS

Kuantitas dilakukannya proses hibridisasi algoritma genetika adalah dengan persentase 50 berdasarkan iterasi/generasi maksimal yang sudah diinisialisasi sehingga hibridisasi VNS dalam memperbaiki hasil reproduksi dilakukan ketika di pertengahan generasi dan di akhir generasi.

4.1. Representasi Kromosom dan Inisialisasi Populasi

Representasi kromosom merupakan suatu proses untuk mengajukan solusi penyelesaian masalah, dimana suatu solusi permasalahan dapat dikodekan kedalam pengkodean yang terdapat pada algoritma genetika (Mahmudy dkk. 2013). Oleh sebab itu, proses representasi kromosom adalah proses awal dan yang paling penting untuk memberikan sebuah solusi permasalahan. Terdapat berbagai macam pengkodean pada algoritma genetika dan pada penelitian ini menggunakan pengkodean berjenis riil (*real-coded*). Struktur

kromosom sebagai solusi dalam proses algoritma genetika memiliki struktur solusi seperti Gambar 4.

Setiap kolom yang terdapat pada solusi merupakan representasi dari jumlah unit barang yang akan dikirim ke distributor pemesan. Setiap kromosom juga dibagi menjadi beberapa segmen. Banyaknya segmen tergantung pada jumlah level yang dibutuhkan berdasarkan pemesanan. Setiap segmen memiliki beberapa subsegmen dan subsegment dari segmen level *i* adalah unit distributor pengirim *j* sesuai dengan level yang ada pada segmen level. Subsegment distributor pengirim memiliki suatu subsegment yaitu kendaraan *v* yang digunakan untuk melakukan pengiriman. Selanjutnya subsegment kendaraan adalah unit distributor yang melakukan pemesanan dan jenis produk yang dipesan *p* menjadi subsegment dari distributor pemesan.

						Level <i>i</i>			...	<i>J</i>	
						Pengirim <i>j</i>			...	<i>J</i>	
						Kendaraan <i>v</i>			...	<i>V</i>	
						Pemesan <i>r</i>			...	<i>R</i>	
<i>P</i>	...	<i>P</i>		<i>P</i>	...	<i>P</i>					
10		111		11		11					

Gambar 4. Struktur Kromosom

Representasi solusi untuk solusi permasalahan distribusi pada Gambar 4 memiliki nilai 10 pada kolom pertama. Arti dari nilai 10 tersebut adalah pada level awal, distributor awal *j* menggunakan kendaraan awal *v* menuju ke distributor pemesan awal *r* mengirimkan produk berjenis *p* awal sebanyak 10 unit produk. Nilai riil pada kolom yang lain juga memiliki makna yang sama sesuai dengan segmen masing-masing. Panjang kromosom didapatkan dari perkalian antara jumlah jenis produk yang dipesan oleh setiap distributor pemesan dan jumlah kendaraan distributor pengirim pada setiap level dan hasil perkalian setiap level dijumlahkan.

4.2. Fungsi Fitness

Sebuah kromosom/individu yang dibuat oleh algoritma genetika memiliki nilai fungsi yang digunakan untuk mengukur seberapa baik kromosom tersebut untuk memberikan solusi. nilai fungsi yang digunakan yaitu nilai *fitness*. Pada proses seleksi nilai *fitness* digunakan untuk menentukan kromosom yang dipilih. Semakin besar nilai *fitness* pada kromosom maka semakin besar pula kemungkinan untuk bisa terpilih menjadi sebuah solusi.

Pada penelitian ini fungsi *fitness* digunakan adalah untuk mendapatkan nilai minimum biaya distribusi. sehingga rumus yang digunakan untuk mencari nilai *fitness* ditunjukkan pada Persamaan (5).

$$fitness = \frac{1}{Z} \tag{5}$$

dimana Z merupakan biaya distribusi yang ditunjukkan pada Persamaan (1) yang dimiliki oleh setiap kromosom

4.3. Reproduksi

Tahap reproduksi merupakan proses unik yang dimiliki oleh algoritma genetika karena pembentukan individu/solusi baru berada pada proses ini. Proses reproduksi memiliki 2 operator yaitu pindah silang (*crossover*) dan mutasi yang bertujuan untuk menggali lebih jauh dan lebih dalam untuk mencari individu baru sehingga ditemukannya individu baru yang unggul dalam menyelesaikan permasalahan (Jiao dkk. 2014). Pada masing-masing operator terdapat *crossover rate* (cr) pada operator *crossover* dan *mutation rate* (mr) pada operator mutasi yang fungsinya adalah sebagai petunjuk berapa jumlah individu baru yang dihasilkan pada setiap proses dalam satu generasi.

4.3.1. Crossover

Salah satu operator pembentuk individu baru dalam proses reproduksi adalah *crossover*. Dikarenakan operator ini membentuk individu baru dari dua individu induk maka hasil yang diperoleh akan beragam karena mewarisi sifat dari dua induk. Jumlah individu baru didapatkan dari perkalian ukuran populasi dan *crossover rate* ($popSize * cr$). Operator *crossover* memiliki beragam model yang bisa dipertimbangkan beberapa diantaranya adalah *one-cut-point*, *two-cut-point*, dan *extended intermediate*.

4.3.1.1. One-cut-point

Model *one-cut-point* merupakan salah satu model *crossover* dengan satu titik pemotongan pada posisi gen. Selanjutnya dilakukan pemotongan posisi gen pada kedua individu induk yang sudah terpilih secara *random* dan disilangkan serangkaian gen sampai pada posisi titik pemotongan gen pada individu induk pertama ke posisi gen pada individu induk kedua (Arora & Arora 2012). Ilustrasi proses *crossover* dengan model *one-cut-point* ditunjukkan pada Gambar 5 dengan perpotongan pada posisi gen ke 5.

x_1	10	34	122	...	21	7	51
y_1	30	21	29	...	13	26	4
z_1	30	21	29	...	13	7	51
z_2	10	34	122	...	21	26	4

Gambar 5. Proses perhitungan *one-cut-point*

dimana $x \in \{x_1, \dots, x_n\}$ dan $y \in \{y_1, \dots, y_n\}$ adalah dua individu induk terpilih secara *random* dan $z \in \{z_1, \dots, z_n\}$ adalah hasil proses *crossover* yang biasa disebut dengan individu baru/anak.

4.3.1.2. Two-cut-point

Two-cut-point merupakan pengembangan dari model *one-cut-point* yang menerapkan titik pemotongan pada dua posisi gen secara *random* dalam individu/kromosom sehingga individu baru hasil model ini adalah ditukarnya nilai gen antara kedua posisi terpilih ke individu yang kedua (Arora & Arora 2012). Gambaran model *two-cut-point* dengan posisi gen ke 2 dan 6 diilustrasikan pada Gambar 6.

x_1	10	34	122	...	21	7	51
y_1	30	21	29	...	13	26	4
z_1	10	34	29	...	13	26	51
z_2	30	21	122	...	21	7	4

Gambar 6. Proses perhitungan *two-cut-point*

4.3.1.3. Extended intermediate

Sebagai salah satu model *crossover*, *extended intermediate* merupakan model yang bertujuan untuk mengubah nilai gen berdasarkan bilangan acak α dengan selisih nilai gen pada kedua individu. (Mahmudy 2015). Persamaan (6) merupakan fungsi pencarian hasil dengan model *extended intermediate*.

$$z_i = x_i + \alpha(y_i - x_i) \tag{6}$$

4.3.2. Mutasi

Operator kedua dalam proses reproduksi yang menjadi ciri khas algoritma genetika adalah mutasi. Berbeda dengan operator *crossover* yang harus menggunakan dua individu induk, operator mutasi hanya menggunakan satu induk saja dalam menghasilkan individu baru. Jumlah individu baru yang dihasilkan berpacu pada hasil perkalian antara ukuran populasi ($popsize$) dan *mutation rate*. Operator mutasi memiliki beberapa model, sebagian diantaranya adalah *swap*, *insertion*, dan *simple random*.

4.3.2.1. Swap

Salah satu model operator mutasi adalah *swap*. Cara kerja model operator mutasi ini adalah menukarkan nilai antar dua posisi gen yang sudah terpilih secara acak (Soni & Kumar 2014).

4.3.2.2. Insertion

Model *insertion* merupakan model yang menyisipkan nilai dari gen posisi tertentu ke posisi gen yang lain yang sudah dibangkitkan secara acak. Setelah disisipkan, posisi gen-gen yang masuk dalam *range* penyisipan akan digeser sampai menempati posisi yang pas dengan jumlah gennya (Soni & Kumar 2014).

4.3.2.3. Simple random

Simple random merupakan salah satu model operator mutasi yang membangkitkan bilangan acak r untuk memperbesar atau memperkecil nilai dari setiap gen setelah dijumlahkan dengan 1. Range r sendiri berkisar pada -0.1 sampai 0.1. Persamaan (7) merupakan formulasi matematika pada perhitungan simple random.

$$x'_i = x_i(1 + r) \quad i = 1, \dots, n \quad (7)$$

Berdasarkan algoritma yang diusulkan, setelah dilakukan proses reproduksi maka akan memasuki proses kondisi apakah hasil individu baru dari proses reproduksi perlu diperbaiki lagi ke proses hibridisasi menggunakan *variable neighborhood search* (VNS). Penentuan dilakukannya hibridisasi atau tidak, dilihat dari banyaknya iterasi/generasi yang sedang berjalan. Karena persentase dilakukannya proses hibridisasi adalah 50, maka proses hibridisasi dilakukan ketika banyaknya generasi mencapai generasi maksimal (akhir) dan setengah dari generasi yang dideklarasikan di awal. Contoh jika generasi yang dideklarasikan di awal adalah 1000, maka hibridisasi untuk memperbaiki hasil reproduksi dilakukan pada generasi akhir yaitu 1000 dan setengahnya yaitu 500. Hasil perbaikan dari hasil reproduksi inilah yang akan masuk ke dalam proses seleksi.

4.4. Seleksi

Langkah selanjutnya adalah proses seleksi untuk memilih individu/kromosom sejumlah ukuran populasi yang akan bertahan menjadi populasi baru pada proses generasi selanjutnya berdasarkan nilai *fitness* yang diperoleh. Ada beberapa jenis model seleksi dalam algoritma genetika seperti *roulette wheely* yang memperhitungkan probabilitas kumulatif nilai *fitness* setiap individu dan *elitism* yang pernah digunakan dalam menyelesaikan permasalahan distribusi multi-level (Rahmi dkk. 2016)(Sarwani dkk. 2016). Model seleksi *elitism* sendiri adalah proses seleksi dengan mengurutkan secara *descending*. Selain *roulette wheely* dan *elitism*, juga terdapat model seleksi *binary tournament*. Model seleksi ini adalah membandingkan nilai *fitness* antara 2 individu yang terpilih secara acak dan individu dengan *fitness* terbesar akan terpilih di populasi baru pada generasi selanjutnya.

5. HASIL DAN DISKUSI

Pada tahap ini dilakukan berbagai pengujian untuk menunjang keoptimalan algoritma yang diusulkan seperti yang pertama adalah pengujian parameter yang mencakup pengujian ukuran populasi, banyaknya generasi, nilai cr , dan nilai mr berdasarkan nilai rata-rata *fitness* terbaik yang diperoleh. Pengujian selanjutnya adalah pengujian tentang pencarian model operator genetika dari proses reproduksi yang terbaik termasuk didalamnya

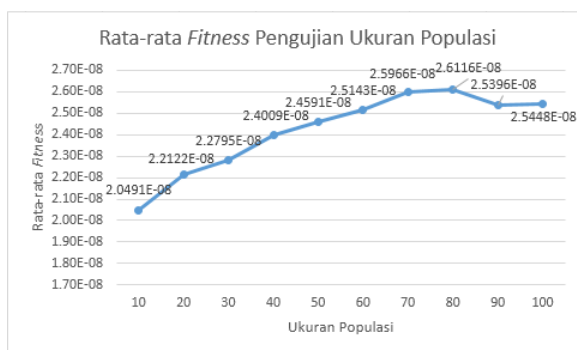
adalah model *crossover* dan mutasi. Setelah dilakukan pengujian parameter dan pengujian pencarian model operator reproduksi yang cocok maka dilakukan analisa dan diskusi seberapa baik model yang diusulkan dalam penelitian ini dalam menyelesaikan permasalahan distribusi multi-level multi-produk.

5.1. Pengujian Parameter GA

Pengujian pertama adalah pengujian parameter dari algoritma genetika. Pengujian ini dilakukan untuk memperoleh hasil algoritma genetika terbaik dan stabil dilihat dari rata-rata *fitness*. Nilai rata-rata *fitness* diperoleh dari pengujian satu ukuran dieksekusi sebanyak sepuluh kali dengan asumsi sudah mewakili solusi yang ditawarkan karena mengingat bahwa sifat algoritma genetika yang *stochastic*. Pengujian parameter algoritma genetika yang dimaksud adalah pengujian tentang ukuran populasi, banyaknya generasi, dan kombinasi dari nilai cr dan mr .

5.1.1. Ukuran Populasi (*Popsize*)

Penerapan multi-populasi dalam menawarkan beragam solusi membuat algoritma genetika mampu menyelesaikan permasalahan optimasi dalam beberapa bidang. Berdasarkan rata-rata *fitness*, pengujian ukuran populasi bertujuan untuk mengetahui ukuran populasi yang tepat dalam menawarkan solusi sehingga mendapatkan hasil yang optimal. Gambar 7 merupakan hasil pengujian ukuran populasi yang ditunjukkan dengan grafik garis dari rata-rata *fitness*. Parameter awal yang digunakan adalah banyaknya generasi 300, nilai cr 0.5, dan nilai mr 0.5 karena sudah pernah diuji pada penelitian distribusi sebelumnya, begitu juga dengan model *crossover*, *mutation*, dan *selection* yang digunakan (Sarwani dkk. 2016).



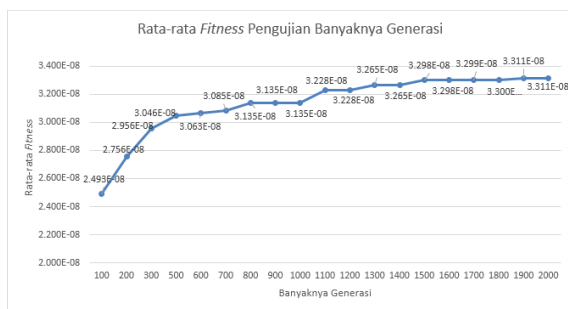
Gambar 7. Pengujian ukuran populasi

Penentuan *popsize* dilihat dari titik awal terjadinya kondisi konvergen berdasarkan grafik garis dari rata-rata *fitness* pada Gambar 7. Kondisi konvergen adalah kondisi ketika pada pengujian yang lebih besar, hasil yang ditunjukkan memiliki perbedaan yang relatif sedikit sehingga

berdasarkan Gambar 7, *popsi* 60 menjadi titik awal terjadinya konvergen. Pada beberapa *popsi* dibawah 60, hasil rata-rata *fitness* pada grafik garis menunjukkan peningkatan hasil yang signifikan. Hal ini disebabkan oleh kecilnya *popsi* memiliki arti sedikitnya solusi yang ditawarkan sehingga masih sulit untuk menemukan solusi yang baik dari sekumpulan solusi tersebut sedangkan pada beberapa *popsi* yang lebih dari 60, hasil rata-rata *fitness* juga mengalami peningkatan sampai pada ukuran 80 meskipun perbedaannya tidak terlalu banyak, namun ketika *popsi* lebih dari 90 terjadi penurunan hasil yang mengartikan bahwa semakin besar *popsi*, peluang solusi yang ditawarkan sudah memiliki solusi terbaiknya sehingga jika diolah kembali, maka tidak lebih baik hasil yang diperoleh. Meskipun *popsi* 90 dan 100 mengalami penurunan, penurunannya tidak akan berbeda jauh dari hasil *popsi* terpilih yaitu 60 (Rahmi dkk, 2016).

5.1.2. Banyaknya Generasi

Selanjutnya adalah pengujian parameter banyaknya generasi yang tujuannya adalah mendapatkan banyaknya generasi yang mendekati optimal dengan waktu komputasi yang relatif singkat. Untuk menentukan banyaknya generasi yang cocok sesuai dengan tujuan yang diinginkan adalah dengan memperhatikan banyaknya generasi mana yang menjadi titik dimulainya konvergensi. Konvergensi merupakan kondisi ketika pada banyaknya generasi yang semakin besar solusi yang ditawarkan tidak memiliki selisih yang terlalu jauh. Gambar 8 merupakan grafik hasil pengujian banyaknya generasi dengan ukuran populasi terbaik hasil dari pengujian sebelumnya 60 sedangkan kombinasi *cr* dan *mr* yang digunakan masih sama dengan pengujian sebelumnya.



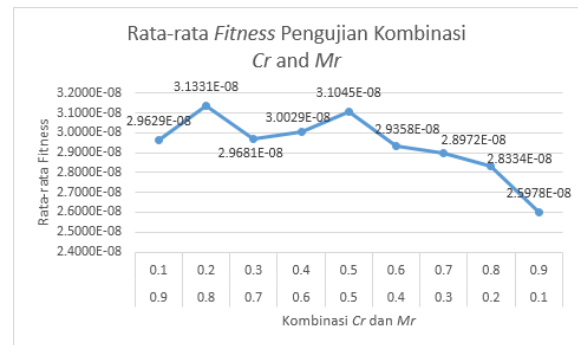
Gambar 8. Pengujian Banyaknya Generasi

Pengujian banyaknya generasi pada Gambar 8 menunjukkan bahwa rata-rata terbaik setiap percobaan mengalami ketidakstabilan meskipun pada generasi tertinggi. Hal ini disebabkan oleh tingkat kompleksitas permasalahan yang cukup tinggi sedangkan berdasarkan nilai *fitness* terbaiknya, titik konvergensi *fitness* berada pada generasi ke 1100. Titik ini dipilih karena perubahan nilai *fitness* terbaik pada generasi yang semakin besar

tidak terlalu memberikan perbedaan yang jauh.

5.1.3. Kombinasi crossover rate (cr) dan mutation rate (mr)

Yang terakhir adalah pengujian parameter kombinasi nilai *cr* dan *mr* agar keberagaman anak (individu/solusi baru) hasil proses reproduksi semakin terjaga. Gambar 9 merupakan grafik garis rata-rata *fitness* hasil pengujian nilai *cr* dan *mr* dengan ukuran populasi dan banyaknya generasi hasil pengujian sebelumnya.



Gambar 9. Pengujian kombinasi nilai *cr* dan *mr*

Hasil pengujian pada Gambar 9 menunjukkan bahwa nilai *cr* 0.8 dan *mr* 0.2 merupakan kombinasi yang tepat untuk permasalahan distribusi multi-level multi-produk. Pada kombinasi tersebut solusi yang dihasilkan menunjukkan hasil yang paling baik dengan keberagaman individu yang lebih banyak dilakukan dari proses mutasi.

5.2. Pengujian Operator Genetika

Pengujian operator genetika merupakan pengujian untuk menentukan model operator genetika mana yang sesuai dan menghasilkan solusi yang mendekati optimal. Ada dua macam pengujian yang dilakukan berdasarkan operator genetika yaitu pengujian dari model *crossover* dan pengujian model *mutation*.

5.2.1. Model Crossover

Pengujian operator genetika pertama adalah pengujian model *crossover*. Tujuannya adalah mencari model *crossover* yang sesuai. Model *one-cut-point*, *two-cut-point*, dan *extended intermediate* adalah *crossover* yang diujikan pada penelitian ini. Tabel 1 adalah proses pengujian yang dilakukan dengan menggunakan parameter algoritma genetika yang terpilih dari hasil pengujian sebelumnya yaitu ukuran populasi 80, banyaknya generasi 800, *cr* 0.3 dan *mr* 0.7.

Berdasarkan pengujian model *crossover* pada Tabel 1 dapat disimpulkan bahwa model *crossover* yang paling baik adalah *two-cut-point*. Hal ini dibuktikan dengan *average fitness* terbesar sehingga

rata-rata biaya (Rupiah) yang dihasilkan paling minimal.

5.2.2. Model Mutasi

Pengujian operator genetika selanjutnya adalah pengujian model mutasi. Tujuannya adalah mencari model mutasi yang cocok dengan permasalahan distribusi sehingga solusi yang ditawarkan mendekati optimal. Hasil pengujian dalam memilih model mutasi yang sesuai ditunjukkan pada **Error! Not a valid bookmark self-reference.** Berdasarkan rata-rata *fitness* dan rata-rata biaya yang dihasilkan, *swap mutation* memberikan solusi yang lebih baik dibandingkan *insertionmutation*.

Tabel 2 merupakan rincian hasil rata-rata *fitness* dengan biaya dari masing-masing model yang dibandingkan. Model mutasi yang diuji dalam penelitian ini adalah *swapmutation* dan *insertionmutation*.

Tabel 1. Model Crossover

no.	model crossover	rata-rata fitness	rata-rata biaya
1	Crossoverone-cut-point	2.54088E-8	39583805
2	Crossovertwo-cut-point	3.07092E-8	32679440
3	Crossoverextended intermediate	2.49052E-8	40537165

Hasil pengujian dalam memilih model mutasi yang sesuai ditunjukkan pada **Error! Not a valid bookmark self-reference.** Berdasarkan rata-rata *fitness* dan rata-rata biaya yang dihasilkan, *swap mutation* memberikan solusi yang lebih baik dibandingkan *insertionmutation*.

Tabel 2. Model Mutasi

no.	model mutasi	rata-rata fitness	rata-rata biaya
1	Mutasi Swap	3.18331E-8	31484625
2	Mutasi Insertion	2.83421E-8	35348305
3	Mutasi Simple random	2.36457E-8	42419490

5.2.3. Model Seleksi

Pengujian operator genetika selanjutnya adalah pengujian model seleksi. Tujuannya adalah mencari model seleksi yang cocok dengan permasalahan distribusi sehingga solusi yang ditawarkan mendekati optimal. Tabel 3 merupakan rincian hasil rata-rata *fitness* dengan biaya dari masing-masing model yang dibandingkan. Hasil model seleksi yang terbaik dibandingkan model lainnya ditunjukkan pada Tabel 3

Hasil pengujian dalam memilih model seleksi yang sesuai ditunjukkan pada Tabel 3. Berdasarkan rata-rata *fitness* dan rata-rata biaya yang dihasilkan, *swapmutation* memberikan solusi yang lebih baik dibandingkan model seleksi yang lain.

Tabel 3. Model Seleksi

no.	model seleksi	rata-rata fitness	rata-rata biaya
1	Seleksi Elitism	3.13905E-8	31994380
2	Seleksi Roulette-wheel	1.15668E-8	86463140
3	Seleksi Binary Tournament	2.81680E-8	94982361

Proses selanjutnya adalah analisa seberapa baik algoritma yang diusulkan. Untuk mendapatkan hasil yang optimal, algoritma yang berbasis genetika bisa menggunakan parameter dan operator genetika hasil pengujian sebelumnya dan kemudian dibandingkan keunggulan solusinya.

Dengan menggunakan parameter dan operator genetika terbaik hasil pengujian sebelumnya maka dilakukan perbandingan hasil antara *random search* (RS), algoritma genetika klasik (GA), dan hibridisasi GA menggunakan *variable neighborhood search* (GA-VNS).

Tabel 4. Hasil Keseluruhan Algoritma

no.	algoritma	rata-rata fitness	rata-rata biaya	rata-rata waktu (ms)
1	RS	2.05077E-8	48784775	1809
2	GA	3.06647E-8	32716150	265091
3	GA-VNS	3.09724E-8	32392960	279332

Berdasarkan Tabel 7, rata-rata waktu dalam *milisecond*(ms) yang ditunjukkan oleh masing-masing algoritma berbeda-beda dan dapat dilihat bahwa waktu yang dihasilkan oleh GA dan GA-VNS relatif sama dan algoritma GA-VNS memberikan hasil yang lebih baik. Meskipun algoritma RS memiliki waktu komputasi yang cepat, hasil yang diperoleh masih jauh selisihnya dibandingkan algoritma lainnya.

6. KESIMPULAN

Permasalahan distribusi multi-level dan multi-produk pada penelitian ini diselesaikan dengan membentuk model matematis dari beberapa variabel yang tujuannya adalah mendapatkan biaya yang minimal. Variabel-variabel tersebut seperti jumlah produk, biaya variabel, kendaraan yang digunakan, biaya tetap, dan status pengiriman. Variabel-variabel yang tersusun secara matematis juga menyelesaikan beberapa *constraint* yang terjadi dalam proses distribusi seperti jumlah produk yang didistribusikan harus sama dengan jumlah order dan tidak melebihi kapasitas maksimal kendaraan maupun jumlah stok yang dimiliki.

Berdasarkan model matematis yang telah tersusun, maka suatu solusi/kromosom permasalahan distribusi dalam penelitian ini terdiri dari serangkaian kolom/gen. Jumlah gen didapatkan dari jumlah gen setiap level yang memesan dimana jumlah gen setiap level diperoleh dari jumlah jenis produk yang dipesan dikalikan dengan kendaraan

yang dimiliki distributor pengirim. Setiap gen dari suatu kromosom berisi bilangan riil yang menunjukkan jumlah produk yang akan didistribusikan. Untuk mengetahui jumlah produk dan jaringan distribusi mana yang terbaik maka diperhitungkan berdasarkan nilai *fitness* yang dihasilkan yang menunjukkan minimalnya biaya yang dikeluarkan dari keseluruhan proses distribusi.

Parameter-parameter optimal algoritma genetika memberikan pengaruh yang besar dalam memberikan solusi terbaik dalam menyelesaikan masalah distribusi. Parameter-parameter tersebut adalah populasi berukuran 60, banyaknya generasi 1100, nilai *cr* 0.8 dan *mr* 0.2. Model *two-cut-point* pada operator *crossover*, model *swapp* pada operator mutasi, dan model seleksi *elitism* menjadi model terbaik dalam memberikan hasil solusi yang lebih baik dalam menyelesaikan masalah distribusi pada penelitian ini.

Berdasarkan parameter dan model optimal tersebut, untuk menyelesaikan masalah distribusi, algoritma GA-VNS mampu memberikan solusi dengan hasil yang mendekati optimal dengan waktu komputasi yang relatif sama dibandingkan dengan algoritma GA klasik yaitu 279332 ms dan 265091 ms. Hasil yang diperoleh dari algoritma GA-VNS juga lebih baik yaitu 32392960 yang dibuktikan dengan penghematan biaya sebesar 323190 jika dibandingkan dengan GA klasik yaitu 32716150. Dibandingkan dengan *random search* (RS), waktu yang dihabiskan oleh algoritma GA-VNS memang lebih lama namun algoritma yang diusulkan tersebut memberikan biaya yang paling minimal dengan penghematan sebesar 16391815.

7. REFERENSI

- ARORA, J.S. & ARORA, J.S., 2012. *Chapter 16 – Genetic Algorithms for Optimum Design*, Elsevier Inc.
- BOUDJELABA, K., ROS, F. & CHIKOUCHE, D., 2014. An efficient hybrid genetic algorithm to design finite impulse response filters. *Expert Systems With Applications*, 41(13), pp.5917–5937. Available at: <http://dx.doi.org/10.1016/j.eswa.2014.03.034>.
- CASTELLI, M. & VANNESCHI, L., 2014. Genetic algorithm with variable neighborhood search for the optimal allocation of goods in shop shelves. *Operations Research Letters*, 42(5), pp.355–360. Available at: <http://dx.doi.org/10.1016/j.orl.2014.06.002>.
- CHEIKH, M. DKK., 2015. A variable neighborhood search algorithm for the vehicle routing problem with multiple trips. *Electronic Notes In Discrete Mathematics*, 47, pp.277–284. Available at: <http://dx.doi.org/10.1016/j.endm.2014.11.036>.
- CHEN, Z.-Q. & WANG, R.-L., 2011. Solving the m-way graph partitioning problem using a genetic algorithm. *Ieej Transactions On Electrical And Electronic Engineering*, 6(5), pp.483–489. Available at: <http://doi.wiley.com/10.1002/tee.20685>.
- GICQUEL, C. & MINOUX, M., 2015. Multi-product valid inequalities for the discrete lot-sizing and scheduling problem. *Computers And Operation Research*, 54, pp.12–20. Available at: <http://dx.doi.org/10.1016/j.cor.2014.08.022>.
- GUO, H., WANG, X. & ZHOU, S., 2015. A Transportation Problem with Uncertain Costs and Random Supplies. *International Journal Of E-Navigation And Maritime Economy*, 2, pp.1–11. Available at: <http://www.sciencedirect.com/science/article/pii/S2405535215000558>.
- GUPTA, A. DKK., 2012. A Comparative Study of Three Echelon Inventory Optimization using Genetic Algorithm and Particle Swarm Optimization. *International Journal Of Trade, Economics And Finance*, 3(3), pp.205–208.
- JIAO, R. DKK., 2014. A Multistage Multiobjective Substation Siting and Sizing Model Based on Operator-Repair Genetic Algorithm. *Ieej Transactions On Electrical And Electronic Engineering*, 9, pp.S28–S36.
- KIM, S.H. & LEE, Y.H., 2016. Synchronized production planning and scheduling in semiconductor fabrication. *Computers & Industrial Engineering*, 96, pp.72–85. Available at: <http://dx.doi.org/10.1016/j.cie.2016.03.019>.
- LANGROODI, R.R.P. & AMIRI, M., 2016. A system dynamics modeling approach for a multi-level, multi-product, multi-region supply chain under demand uncertainty. *Expert Systems With Applications*, 51, pp.231–244. Available at: <http://dx.doi.org/10.1016/j.eswa.2015.12.043>.
- MAHMUDY, W.F., 2015. Optimization of Part Type Selection and Machine Loading Problems in Flexible Manufacturing System Using Variable Neighborhood Search. *Iaeng International Journal Of Computer Science*, 42(3).
- MAHMUDY, W.F., MARIAN, R.M. & LUONG, L.H.S., 2014. Hybrid Genetic Algorithms for Part Type Selection and Machine Loading Problems with Alternative Production Plans in Flexible Manufacturing System. *Ecti Transaction On Computer And Information Technology*, 8(1), pp.80–93.
- MAHMUDY, W.F., MARIAN, R.M. & LUONG, L.H.S., 2013. Modeling and Optimization of Part Type Selection and Loading Problem in Flexible Manufacturing System Using Real Coded Genetic Algorithms. *International*

Journal Of Electrical, Computer, Electronics And Communication Engineering, 7(4), pp.251–260.

- MLADENOVIC, N., UROSEVIC, D. & PEREZ-BRIT, D., 2016. Variable Neighborhood Search for Minimum Linear Arrangement Problem. *Yugoslav Journal Of Operations Research*, 26(1), pp.3–16.
- QIONGBING, Z., 2016. A New Crossover Mechanism for Genetic Algorithms with Variable-length Chromosomes for Path Optimization Problems. *Expert Systems With Applications*, 60, pp.183–189. Available at: <http://dx.doi.org/10.1016/j.eswa.2016.04.005>.
- RAHMI, A., SARWANI, M.Z. & MAHMUDY, W.F., 2017. Genetic Algorithms for Optimization of Multi-Level Product Distribution. *Accepted to International Journal Of Intelligent Engineering & Systems*.
- SARWANI, M.Z., RAHMI, A. & MAHMUDY, W.F., 2017. An Adaptive Genetic Algorithm for Cost Optimization of Multi-Stage Supply Chain. *Accepted to Journal Of Telecommunication, Electronic And Computer Engineering*.
- SITEK, P. & WIKAREK, J., 2012. Mathematical programming model of cost optimization for supply chain from perspective of logistics provider. *Management And Production Engineering Review*, 3(2), pp.49–61.
- SONI, N. & KUMAR, T., 2014. Study of Various Mutation Operators in Genetic Algorithms. *International Journal Of Computer Science And Information Technology*, 5(3), pp.4519–4521.
- THAKUR, M. & KUMAR, A., 2016. Electrical Power and Energy Systems Optimal coordination of directional over current relays using a modified real coded genetic algorithm : A comparative study. *International Journal Of Electrical Power And Energy Systems*, 82, pp.484–495. Available at: <http://dx.doi.org/10.1016/j.ijepes.2016.03.036>.
- ZIEBUHR, M. & KOPFER, H., 2016. Solving an integrated operational transportation planning problem with forwarding limitations. *Transportation Research Part E: Logistics And Transportation Review*, 87, pp.149–166. Available at: <http://dx.doi.org/10.1016/j.tre.2016.01.006>.