

IMPLEMENTASI TWO POINT CROSSOVER PADA KNAPSACK PROBLEM

Rijois Ibov Erwin Saragih

Komputerisasi Akuntansi, Universitas Methodist Indonesia

Email:erwin_saragih@yahoo.com

Jl. Hang Tuah No. 8 Medan, Sumatera Utara

ABSTRACT

Genetic algorithm is heuristic searching algorithm which based on nature selection of mechanism and nature genetic. The basic concept that inspires the genetic algorithm is that evolution theory. One of crossover operator in genetic algorithm is two-point crossover. This operator can make better improvement in solving combinatorial problem. Previous research has done with one-point crossover and it is compared with tow-point crossover in this research. Knapsack is a combinatorial problem which is to find good solution with constraint. Evaluation is done 10 times execution on genetic algorithm (GA), and experimental results show that two-point crossover can gives a quite good result in solving optimization problem.

Keywords: Genetic Algorithm, Two-Point Crossover, Knapsack Problem

1. PENDAHULUAN

Algoritma genetika adalah kelas populasi berdasarkan teknik pencarian acak yang semakin banyak digunakan di sejumlah aplikasi praktis. Biasanya algoritma ini mempertahankan sejumlah solusi potensial untuk masalah yang sedang ditangani, yang dapat dilihat sebagai bentuk memori kerja ini dikenal sebagai populasi. Poin iteratif baru dalam ruang pencarian yang dihasilkan untuk evaluasi dan opsional dimasukkan ke dalam populasi (Smith, 2002).

Permasalahan umum pada algoritma genetika yang sering terjadi adalah lokal optima dan hal ini terjadi karena hilangnya perbedaan populasi (population diversity) awal dengan populasi selanjutnya (Zhu & Liu, 2004). Jika perbedaan populasi terlalu kecil akan memungkinkan terjadinya lokal optima, dan jika terlalu besar akan mengakibatkan lamanya waktu yang dibutuhkan algoritma genetika dalam menghasilkan solusi terbaik. Banyak penelitian yang telah dilakukan terkait dengan permasalahan diatas untuk menghindari lokal optima serta meningkatkan performansi algoritma genetika. Salah satu pendekatan yang dilakukan melalui perbaikan kinerja dari operator genetika itu sendiri; seperti operator seleksi, Persilangan dan mutasi.

Menurut Varnamkhasti etal (2012), performansi algoritma genetika dipengaruhi oleh operator genetika; operator seleksi, persilangan dan mutasi secara umum.

Menurut Singh (2011), permasalahan *knapsack* adalah suatu permasalahan optimasi kombinatorial. Sebagai contoh diberikan satu set item dengan berat dan nilai, kemudian dilakukan pemilihan dari item-item tersebut untuk dimasukan kedalam ransel (*knapsack*) dengan kapasitas terbatas. Jadi item-item yang dimasukan beratnya harus lebih kecil atau sama dengan kapasitas dari ransel tersebut, tetapi total nilai sebesar mungkin.

Penelitian terdahulu telah dilakukan menggunakan one-point *crossover*, pada penelitian sekarang akan dilakukan menggunakan two-point crossover dengan harapan dapat memberikan hasil yang lebih baik terhadap permasalahan *knapsack problem*. Berdasarkan penelitian di atas maka penulis tertarik untuk melakukan penelitian implementasi two-point *crossover* dalam menyelesaikan permasalahan *knapsack*.

2. METODE PENELITIAN

Data *knapsack problem* adalah suatu permasalahan optimasi kombinatorial. Data tersebut telah diuji oleh

pelbagai algoritma yang berbeda untuk mencari solusi terbaik. Sebagai perbandingan juga digunakan data penelitian terkait dengan 50 item serta nilai weight dan value. Data diambil dari penelitian sebelumnya [12].

Table 1 Data Knapsack Problem

Item	Weight	Value
1	9	150
2	13	35
3	153	200
4	50	160
5	15	60
6	68	45
7	27	60
8	39	40
9	23	30
10	52	10
11	11	70
12	32	30
13	24	15
14	48	10
15	73	40
16	42	70
17	43	75
18	22	80
19	7	20
20	18	12
21	4	50
22	30	10

Inisialisasi Populasi

Pada populasi awal terdiri dari 6 kromosom yang dibentuk dari 10 item barang data *knapsack problem* pada tabel 3.1. Kesepuluh item barang tersebut membentuk sebuah

kromosom atau individu, dimana kromosom tersebut disusun dari beberapa gen yang berisi nilai. Nilai gen ditentukan berdasarkan item barang yang dipilih atau tidak. Item barang yang dipilih diberi tanda 1 sedangkan item barang yang tidak dipilih diberi tanda 0. Kedua tanda tersebut dapat direpresentasikan seperti pada gambar 3.2

0	1	1	1	1	0	0	1	1	0
1	0	1	0	1	0	0	1	1	0
0	0	1	0	1	0	0	1	1	0
0	1	1	0	1	0	0	1	1	0
1	1	0	0	0	1	0	1	0	0
0	0	1	0	1	0	0	1	1	0

Gambar 1 Representasi Kromosom Pada Knapsack Problem

Nilai Fitness

Nilai fitness dapat dihitung dengan menjumlahkan nilai semua barang yang dipilih, tetapi dibatasi berat maksimalnya. Nilai fitness dan berat total dapat dihitung dengan menggunakan rumus:

$$F = \sum_{i=1}^n b_i v_i \tag{2.1}$$

$$W_{tot} = \sum_{i=1}^n b_i w_i \tag{2.2}$$

3. HASIL DAN PEMBAHASAN

Pada penelitian ini ditampilkan hasil pengujian *two-point crossover* pada algoritma genetika.

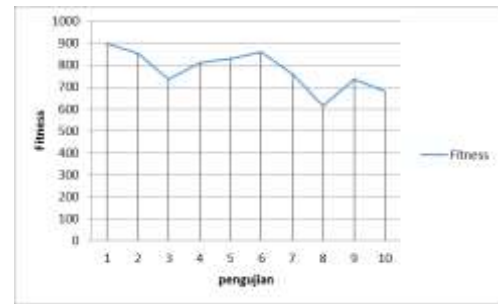
Penilaian dilakukan terhadap nilai maksimum yang didapat dan kecepatan menemukan solusi terbaik pada generasi keberapa. Adapun hasil pengujian akan ditampilkan dalam bentuk tabel dan grafik.

Parameter yang digunakan untuk penerapan *two-point crossover* pada algoritma genetika adalah sebagai berikut[12]:

1. Jumlah Individu = 10, 20, dan 50
2. Batas Generasi = 50, 80 dan 100
3. Probabilitas persilangan = 0.5

Tabel 2 Hasil pengujian pada 10 Individu dan 50 generasi

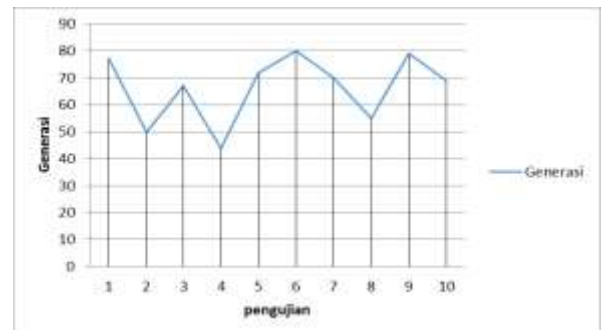
Pengujian	Generasi	Fitness
1	77	900
2	50	855
3	67	737
4	44	810
5	72	830
6	80	860
7	70	760
8	55	615
9	79	735
10	69	680



Gambar 2 Grafik pengujian untuk fitness

Pada tabel 3.1 terlihat bahwa nilai fitness dari setiap pengujian hasilnya berbeda, dan pengujian dilakukan sebanyak sepuluh kali. Nilai fitness

tersebut mewakili jumlah item barang yang dapat dipilih untuk dimasukkan ke dalam *knapsack*. Pada pengujian diatas nilai fitness tertinggi sebesar 900 pada generasi ke 77, sedangkan nilai fitness terendah terdapat pada generasi ke 55 sebesar 615.



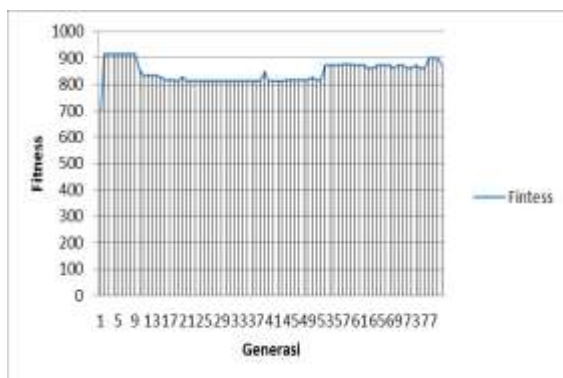
Gambar 3 Grafik pengujian untuk generasi

Berdasarkan nilai fitness terbaik pada tabel 3.1 yaitu sebesar 900 maka pada tabel 3.2 ditampilkan proses pencarian fitness terbaik per generasi serta solusi.

Tabel 3.2 Pencarian fitness terbaik per generasi

Generasi	Fitness	Solusi
1	702	10011011001010111110110 1 4 5 7 8 11 12 14 15 16 17 18 20 21 (Item barang terpilih)
2	813	
3	813	
4	813	
5	813	
6	813	
7	813	
8	813	
9	813	
10	853	
11	833	
12	833	
13	833	
14	833	
15	827	
16	817	
17	817	
18	817	
19	812	
20	827	
21	812	
22	812	
23	872	
24	872	
25	900	
50	872	

Pada tabel 2 terlihat bahwa pencarian fitness terbaik terjadi pertama di generasi ke 25, meskipun generasi berikutnya bernilai fitness sama. Solusi merupakan item-item barang yang dipilih pada knapsack. Adapun contoh hasil perhitungan ditampilkan pada lampiran 1.



Gambar 4 Grafik fitness per generasi

Tabel 4 Hasil Pengujian

Crossover	Fitness
One-point	877
Two-point	900

Pada tabel 4 terlihat bahwa hasil pengujian pada two-point crossover ada peningkatan nilai fitness atau nilai value item yang dapat dimasukkan ke dalam ransel (knapsack problem) dibandingkan dengan one-point crossover.

KESIMPULAN

Berdasarkan pembahasan serta pengujian yang dilakukan pada penelitian ini, maka kesimpulan yang dapat diambil adalah:

1. Dari hasil penelitian yang telah terlihat bahwa two-point crossover lebih baik dibandingkan dengan one-point crossover pada penelitian ini.
2. Hasil best fitness two-point crossover pada permasalahan knapsack adalah 900 sedangkan hasil best fitness one-point crossover pada penelitian sebelumnya adalah 877 dengan data yang sama [12].

REFERENSI

[1] De Falco, Della, A., Ciopa. & Tarantino, E. 2002. Mutation-based genetic algorithm: performance evaluation. *Applied Soft Computing* 1:285-299

[2] Gen, M. & Cheng, R. 1997. *Genetic Algorithms & Engineering Design*. Wiley-Interscience: New York.

[3] Haibo, Z., Liwen, C., Shenyong, G., Jianguo, C., Feng, Y., & Daqing, L. 2011. Improvements of Genetic Algorithm to the Knapsack Problem. *ICAIC*. Part I: 202-206

[4] Holland, J.H. 1975. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Ann Arbor, Mich: USA.

[5] Katayama, K. & Sakamoto, H. 2000. The efficiency of Hybrid Mutation Genetic Algorithm for the Travelling Salesmen Problem. *Mathematical and Computer Engineering* 31:197-203

[6] Mühlenbein, H. 1992. How Genetic Algorithms Really Work I. Mutation and Hillclimbing. In *Proceedings of the 2nd Conference on Parallel Problem Solving from Nature*, pp. 15-29.

[7] Passino, K.M, Yurkovich, S. 1998. *Fuzzy Control*. Edison Wesley Longman Inc: Ohio.

[8] Sakawa, M. 2002. *Genetic Algorithm and Fuzzy Multiobjective Optimizaton*. Springer: Japan.

[9] Singh, R.P. 2011. Solving 0-1 Knapsack Problem Using Genetic Algorithm. *IEEE*. 11:591-585

[10] Sivanandam, S.M, & Deepa, S.N. 2008. *Introduction to Genetic Algorithms*, Springer, Berlin:Germany.

[11] Smith, J.E. 2002. *Handbook of Global Optimization*. Volume 2, 275-362.

[12] Rijojs, I. E. S., 2016., Analisis One Point Crossover Pada Knapsack Problem Algoritma Genetika., *ISD*. Volume 1, 156-160.