Genetic Algorithms with Variable Length Chromosomes for High Constraint Problems in Spatial Data

Rahmadya Trias Handayanto
Computer Engineering
Universitas Islam 45 Bekasi
Bekasi, Indonesia
rahmadya.trias@gmail.com

Nitin Kumar Tripathi
Remote Sensing and Geographic Information System
Asian Institute of Technology
Pathumthani, Thailand
nitinkt@ait.asia

Sumanta Guha
Computer Science
Asian Institute of Technology
Pathumthani, Thailand
guha@ait.asia

Herlawati
Information System
STMIK Bina Insani
Bekasi, Indonesia
herlawati@binainsani.ac.id

***Abstract*—Constraint handling is the main task in constrained optimization problems. Variable length chromosomes in the genetic algorithm has been used widely for faster computation, but in this study it was used to handle the constraint as well. This method uses the characteristic of the genetic algorithm with bit-strings conversion from real numbers. By the bit-strings format, the population of the candidates can be limited only in the study area where it is impossible when the real number format is used. Therefore, it will reduce the searching area and make the optimization process faster. Variable length chromosomes method can also be integrated with another constraint handling, i.e. the death penalty method. The results showed that the proposed method was able to optimize land use in Bekasi City, Indonesia, as a case study.**

Keywords—constrained-optimization, death penalty, constraint handling, Bekasi City

I. Introduction

As a robust optimization method, genetic algorithm (GA) has been widely used in many areas. One interesting area is optimizing the geographical data. City planners need an appropriate method to allocated land use properly through a land-use optimization. One challenging problem in land-use optimization is the wide area that involves large number of land uses. For example in Bekasi City, West Java, Indonesia, with its 210.49 km^2 area, 90 percent of its land use are built-up with more than 3000 building that need to be optimized [1]. The current study proposed a modification in GA to make computation faster than the original one.

GA is a stochastic optimization that mimic the evolution of nature [2]. Some terms are similar to biology such as gen, chromosome, allele, mutation, reproduction, selection, etc. The main part of GA is the chromosome that is represented by bit-strings from encoding the real number to binary. After crossover, mutation, and selection, new generations are created that more adaptive than before. This cycle makes GA as the member of evolutionary algorithms with the other methods such as particle swarm optimization (PSO) [3], simulated annealing (SA) [4], [5], and other hybrid methods [6]–[9].

GA involves many algorithms that handle encoding and decoding, crossover, and selection methods that make this algorithms flexible to be modified. Searching area is the main focus of GA optimization since it influence the accuracy (local or global optimum) and speed of optimization process. Therefore, many studies on adding and reducing the searching area have been done by manipulating the length of chromosomes that compatible with the problems, e.g. structural topology design [10], manipulators motion [11], node placement problems [12], path optimization [13], and other implementation of variable length of chromosomes [14].

The rest of the paper is organized as follows. After discussing the optimization problem and the study area, section 2 discusses the optimization methods and constraint handlings. The results are discussed and concluded regarding the benefit of variable length of chromosomes.

II. Methods

There are two kinds of GA based on the length of chromosomes, i.e. the fixed length and variable length of chromosomes (VLC). This study uses variable length of chromosomes method.

A. Related Works

Many studies on GA with a VLC method. This method changes the length of chromosomes based on its purpose [14]. For example to avoid the local optimum, VLC adds the length of chromosomes. Adding the length of chromosomes will add the searching area and need more computation resources, but ensure the global optimum. In the other hand, lowering the length of chromosomes will reduce the searching area. For example, in the combination design problem, a k-size building block with l size chromosome will require a population size, population size= $2^k \binom{l}{k}$. There are a total of $\binom{l}{k}$ gene combinations of size k, and for each gene combination there are 2^k different allele combinations.

Insertion and deletion are usually implemented to add chromosome length as well as crossover and mutation [15]. For example two parents with different length of chromosomes, e.g. P_1=10001 01101 00111 111|11 11100 and P_2=00110 000|00, by crossover operation at "|" location will give two children, i.e. Q_1=10001 01101 00111 111|00 and Q_2=00110 000|11 11100.

Another study on intrusion detection with VLC [16] used variable length where the chromosome represented as the rule. The crossover process was similar to [15]. As an example, there are two parents, P_1=*{rule a1, rule a2, rule a3}* and P_2=*{rule b1, rule b2, rule b3, rule b4, rule b5}* with the crossover point at the *rule a3* and *rule b3* will have two children, C_1=*{rule a1, rule a2, rule a3, rule b4, rule b5}* and C_2=*{rule b1, rule b2, rule b3}*.

Previous studies showed that adding and reducing length of chromosomes have improved performance according to the problem (increasing accuracy or reducing computation time). In this study VLC method was used in spatial-data optimization with many regions as constraints. The chromosomes length will be reduced to cover only the study area and reduce the computation time.

B. *Proposed Method*

Since the land-use candidates in city level is very wide, lowering the length of chromosomes will make the computation faster. In addition, by reducing the chromosomes inside the limit of latitude and longitude will help GA handling the constraint, especially the candidates that have to be inside the study area. Following are the procedures to create VLC of GA, following [2], [17] and also from previous work [18] for spatial data optimization:

1. N Samples of population are created from the search space. Decision variables are chosen.
2. The series of binary strings (0s and 1s series) are created as chromosomes. This is the encoding step from real number to binary with variable length as follow:
   2a. Found the minimum x and y as basis of phenotype
   2b. Encode the number outside the minimum x and y into binary (genotype)
3. Cost function values are calculated and assign fitness to each individual.
4. Individuals are sorted based on the fitness value and the reproduction operator is applied.
5. Crossover operation is carried on. This operation take two chromosomes at a time.
6. Based on the probability number, some chromosomes are mutated.
   6a. Decode the binary into real number of x and y
   6b. Add to the base of phenotype to convert back to real number
7. Weakest member (based on fitness value) of new population is replaced with the best member of old population.
8. New population replaces the old population.
9. Step 3 to 8 are repeated for a given number of generations.

The modification from the original GA are shown in step 2a, 2b, 6a, and 6b that encode and decode part of their real number of candidates. In Matlab, the procedure for encoding the real number into binary are as follows (step 2a and 2b):

```
1: basex=min(landuse(:,1))
2: basey=min(landuse(:,2))
3: binx=dec2bin(x(:,1)-basex);
4: biny=dec2bin(x(:,2)-basey);
```

Only the real number in the range of study area that have to be convert into binary. Two variables were chosen as base of the real number (line 1 and 2) that excluded in encoding to binary (line 3 and 4). After crossover and mutation process, the two binaries (latitude and longitude) were merged with their base and decode to real number, similar to phenotype in biology (Fig1).

```
1: x=bin2dec(binx)+basex
2: y=bin2dec(biny)+basey
```

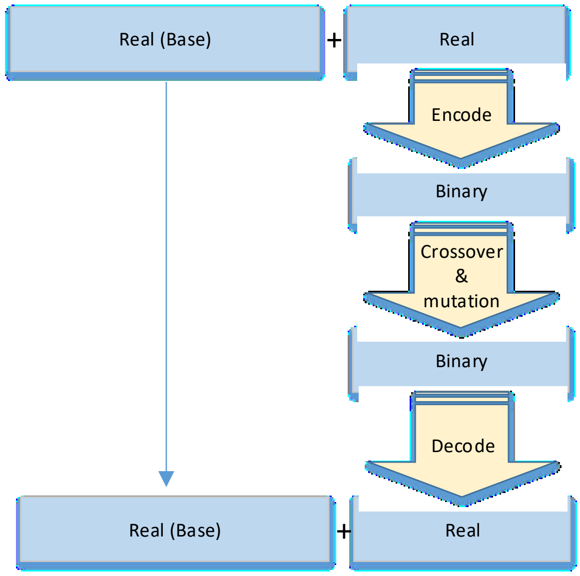


Fig. 1. Encode and decode of a candidate

C. *Case Study*

As case study, the proposed method was implemented in land use optimization in Bekasi City, West Java, Indonesia (Fig2). This optimization used objective functions that follow the sustainable development concept maximizing compactness (F_1), compatibility (F_2), dependency (F_3), and suitability (F_4) as follows:

$$F_1\text{: Maximize}\left(\frac{1}{n}\left(\sum_{i=1}^{n}\text{Compactness}\right)\right) \quad (1)$$

$$F_2\text{: Maximize}\left(\frac{1}{n}\left(\sum_{i=1}^{n}\frac{1}{n_i}\sum_{j=1}^{n_i}(\text{Comp}_{ij})\right)\right) \quad (2)$$

$$F_3\text{: Maximize}\left(\frac{1}{n}\left(\sum_{i=1}^{n}\frac{1}{n_i}\sum_{j=1}^{n_i}(\text{Dep}_{ij})\right)\right) \quad (3)$$

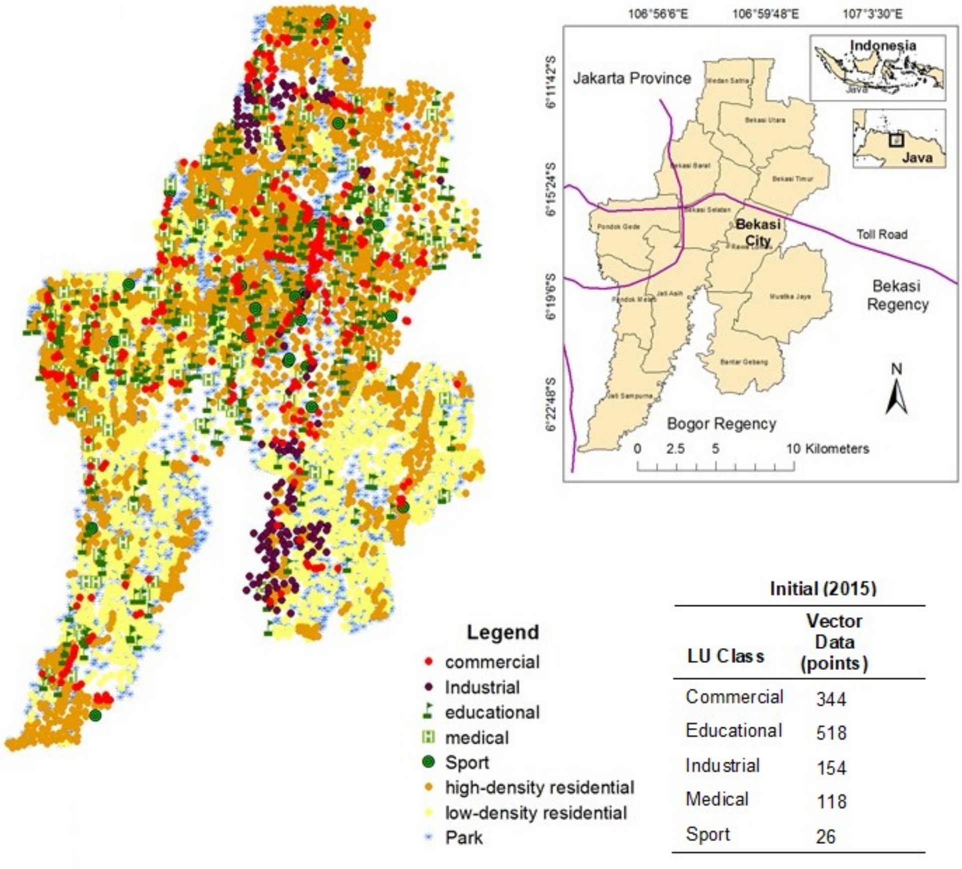
$$F_4\text{: Maximize}\left(\frac{1}{n}\left(\sum_{i=1}^{n}\text{Suitability Score}\right)\right) \quad (4)$$

where n is the number of land use in the study area, i and j are the current land use and its neighbor respectively. Compactness, Comp_{ij}, Dep_{ij}, and Suitability Score are objective values based on based on previous work (suitability analysis) [6]. Aggregating function method [19] was chosen to handle multi-objective problem of maximization:

$$F = max \sum_{i=1}^{k} w_i F_i(x) \quad (5)$$

$$H = \text{Inside Allowable Location} \quad (6)$$

where wi ≥ 0 are the weighting coefficients representing the relative importance of the k objective functions; Fi is the objective function of criterion i from Equations (1) to (4) and H is constraint handling based on land use class and scenario. In this study, the sustainable-development based constraint was chosen, i.e. inside the study area and outside roads, rivers, lakes, and other restricted areas.



Initial (2015)	
LU Class	Vector Data (points)
Commercial	344
Educational	518
Industrial	154
Medical	118
Sport	26

Fig. 2. Study Area and Initial Land Use

Death penalty method was used to handle the constraints (unallowable location for land use allocation). The "point-in-polygon" algorithm to check the point inside or outside a polygon was used [20], [21]. The procedure of death penalty are as follows:

1. Define constraints (vector x and vector y)
2. Check number of intersection between a point of candidate and the constraints
   2a. if there is odd number of intersection, then the point is inside the constraint
   2b. if there is even number of intersection, then the point is outside the constraint
3. Change the candidates inside the constraint to their previous locations

For testing in the study area, the VLC of GA was integrated with PSO and Local Search in Hybrid Multi-Criteria Evolutionary Algorithms (HMCEA) based on previous works [6], [22]. The proposed VLC used parameters, i.e. mutation and selection probability that were set to 0.00015 and 0.5 respectively.

III. Result and Discussion

To see the effect of variable length of chromosomes, GA was run to optimize the land use in the level of "*kecamatan*" (one level below a district or city), i.e. South of Bekasi (shown in bold line in Fig. 3a). A rectangle area in Fig. 3a shows the effect of VLC that limits the optimization only in a specific study area in latitude and longitude range. Some points were outside the VLC limit (rectangle area) because of mutation process. However, some points are outside the study area and need another process, i.e. constraint handling, but most of the results of VLC are inside the study area that minimize the constraint handling's task.

To ensure the candidates inside the study area and outside of restricted areas (rivers, lakes, roads, etc.) a constraint handling method, the death penalty, was used. Fig 3b shows the optimization result after death penalty implementation that force every candidate to the initial position and search for another candidate locations.

For optimizing in a City scale, the proposed method use a VLC with the rectangle area around Bekasi City. GA with VLC run several time to optimize land use in Bekasi City until fitness score reached the stop condition.

Fig 4 shows the optimization result for every land-use class in the study area. The chart shows the saturated condition is achieved after 6 and 7 run and the fitness score increases from 2.68 to 2.83. The commercial and industrial areas tend to be located in the center of the city, high density residential in the middle and north area, and low density residential in the south and south-west of Bekasi City. The other classes (sport, medical, park, and education) are located spread around the study area.

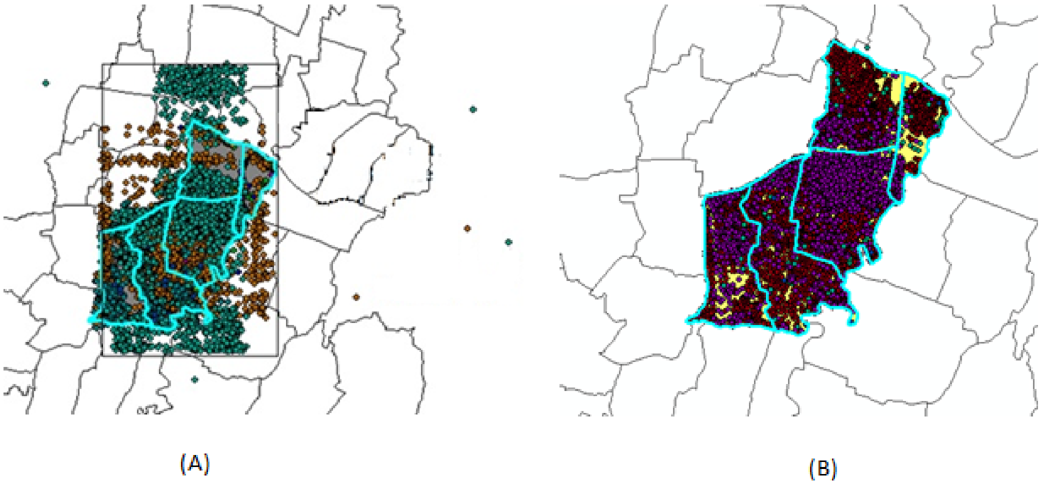


Fig. 3. Testing for (a) VLC optimization result, and (b) VLC with death-penalty constraint handling

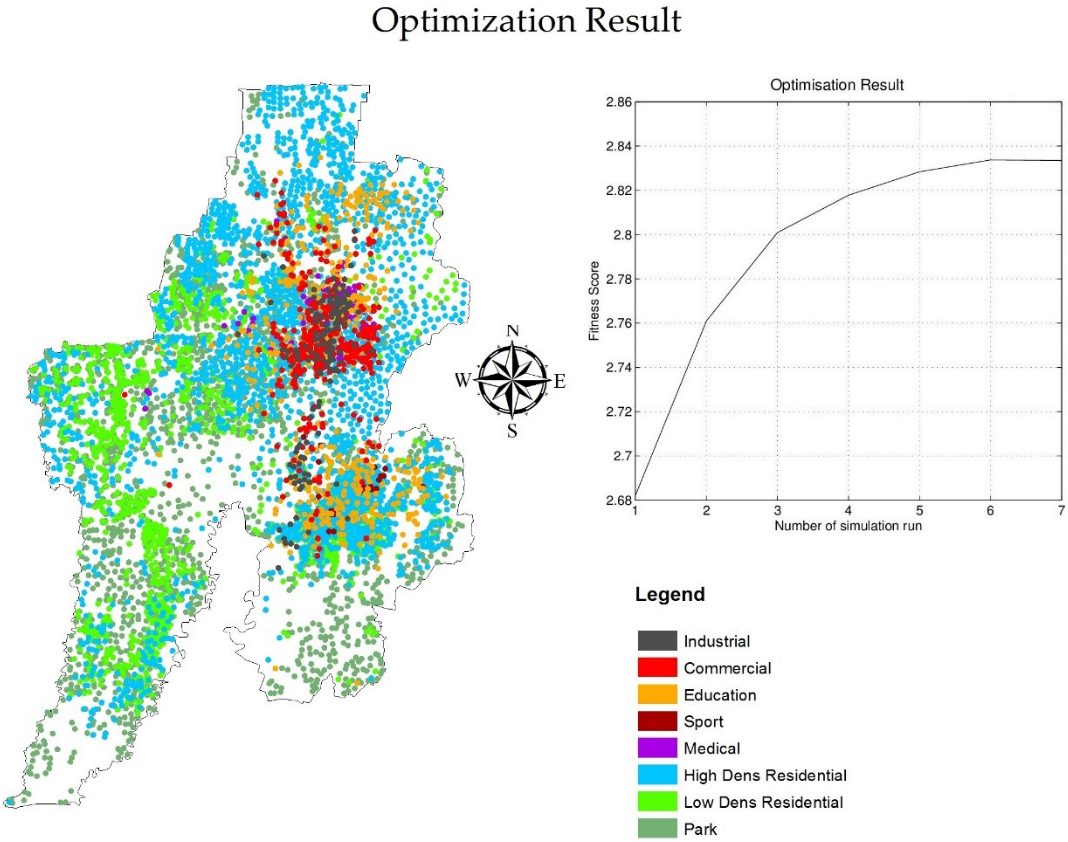


Fig. 4. Optimization Result

IV. CONCLUSIONS

To achieve the optimization objective (sustainable development goal), every land-use class need another class. Large number of land uses will need more computation resources to calculate computability, dependency, compactness and suitability objectives. The study showed the ability of VLC to reduce the searching area and minimize computation need. To handle a lot of constraints, death penalty method in this study was used successfully and showed the ability to ensure the results to follow the allowable locations. However, future research on more flexible constraint handling and a 3D spatial optimization are needed.

ACKNOWLEDGMENT

The authors thank to Research, Technology and Higher Education Department (RISTEK-DIKTI) of Indonesia, local government of Bekasi City during, Higher Education in Informatics and Computer (APTIKOM), Asian Institute of Technology (AIT) Thailand, and Universitas Islam 45 Bekasi. Also for the reviewers for the comments and valuable suggestions.

REFERENCES

[1] BAPPEDA, "Rencana Tata Ruang Kota Bekasi," 2013.

[2] J. H. Holland, "Genetic Algorithms," *Scientific American*, pp. 66–72, 1992.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," *IEEE Int. Conf. Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[4] D. Novitasari, I. Cholissodin, and W. F. Mahmudy, "Hybridizing PSO With SA for Optimizing SVR Applied to Software Effort Estimation," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 14, no. 1, p. 245, 2016.

[5] X. Li and A. G. Yeh, "Integration of genetic algorithms and GIS for optimal location search," *Int. J. Geogr. Inf. Sci.*, vol. 19, no. 5, pp. 581–601, 2005.

[6] R. T. Handayanto, N. K. Tripathi, S. M. Kim, and S. Guha, "Achieving a Sustainable Urban Form through Land Use Optimisation : Insights from Bekasi City's Land Use Plan (2010-2030)," *Sustainability*, vol. 9, 2017.

[7] T. Goel and K. Deb, "Hybrid methods for multi-objective evolutionary algorithms," *Proc. Fourth Asia-Pacific Conf. Simulated Evol. Learn.*, pp. 188–192, 2002.

[8] K. Deb and R. Datta, "Hybrid evolutionary multi-objective optimization and analysis of machining operations," *Eng. Optim.*, vol. 44, no. 6, pp. 685–706, 2012.

[9] K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multi-objective optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, 2013, pp. 495–511.

[10] I. Y. Kim and O. De Weck, "Variable Chromosome Length Genetic Algorithm for Structural Topology Design Optimization," *45th AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf.*, no. April, pp. 1–12, 2004.

[11] J. Riquelme, M. A. Ridao, E. F. Camacho, and M. Toro, "Using Genetic Algorithm With Variable-length Individuals For Planning Two Manipulators Motion," *Artif. Neural Nets Genet. Algorithms*, no. Cd, pp. 2–5, 1998.

[12] Y. H. Zhang, Y. J. Gong, T. L. Gu, Y. Li, and J. Zhang, "Flexible genetic algorithm: A simple and generic approach to node placement problems," *Appl. Soft Comput. J.*, vol. 52, pp. 457–470, 2017.

[13] Z. Qiongbing and D. Lixin, "A new crossover mechanism for genetic algorithms with variable-length chromosomes for path optimization problems," *Expert Syst. Appl.*, vol. 60, pp. 183–189, 2016.

[14] H. Kargupta, "SEARCH, Polynomial Complexity, And The Fast Messy Genetic Algorithm," University of Illinois, 1995.

[15] R. Srikanth, R. George, N. Warsi, D. Prabhu, F. E. Petry, and B. P. Buckles, "A variable-length genetic algorithm for clustering and classification," *Pattern Recognit. Lett.*, vol. 16, no. 8, pp. 789–800, 1995.

[16] H. Wang and R. Ma, "Design of Neural Networks for Intrusion Detection," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 14, no. 3A, p. 321, 2016.

[17] J. R. Raol and A. Jalisatgi, "From Genetics to Genetic Algorithms," *Resonance*, vol. 1, no. 8, pp. 43–54, 1996.

[18] R. T. Handayanto, A. S. Srie Gunarti, S. Samsiana, and Herlawati, "A Web-GIS based integrated optimum location assessment tool for gas station using genetic algorithms," *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 3, pp. 1383–1388, 2015.

[19] C. A. C. Coello, "Evolutionary Multiobjective Optimization: Theoretical Advances and Applications," A. Abraham, L. Jain, and R. Goldberg, Eds. London: Springer London, 2005, pp. 7–32.

[20] K. Hormann and A. Agathos, "The point in polygon problem for arbitrary polygons," *Comput. Geom. Theory Appl.*, vol. 20, no. 3, pp. 131–144, 2001.

[21] S. Wise, *Gis Basics*. New York: Taylor & Francis, 2002.

[22] R. T. Handayanto, S. M. Kim, N. K. Tripathi, and Herlawati, "Land use growth simulation and optimization in the urban area," in *Proceedings of the 2nd International Conference on Informatics and Computing, ICIC 2017*, 2018, vol. 2018–Janua.