Exploring human genome feature for improving genetic algorithm performance

Febri Zukhruf[1,*], *Irma* Susan Kurnia[2], *Russ* Bona Frazila[1], *Gaga* Irawan Nugraha[2], and *Mas* Rizky A.A Syamsunarno[2]

[1]Transportation Engineering Research Group, Faculty of Civil and Environmental Engineering, Institut Teknologi Bandung, Bandung, Indonesia
[2]Faculty of Medicine, Universitas Padjajaran, Bandung, Indonesia

Abstract. Genetic algorithm (i.e., GA) has longtermly obtained an extensive recognition for solving the optimization problem. Its pipelines process, which involves several operations, has been applied in many NP-hard problems, including the transportation network design problem (i.e., TNDP). As part of evolutionary computation methods, GA is inspired by Darwinian evolution, which is relied on the genetic operators (i.e., recombination, and mutation). On other side, the considerably achievement has been acquired by the genome researches, which offers an opportunity to deeply explore the recombination and mutation processes. This paper then presents variants of GA, which are inspired by the recent genome evidence of genetic operators. This exploration expectantly extends the benefit of evolution-based algorithm, which has been shown by the previous finding of GA. For examining the performance of proposed GA, the numerical experiment is involved for solving the TNDP. The performance comparisons show that the variation of crossover rate within a certain group of population provide better result than the standard GA.

1 Introduction

The considerable growth on traffic has delivered a numerous problem, which forces the decision maker to develop the transportation network (TN). Though, the budget constraint has brought the more pressure to its development. Hence, model for designing TN is required for efficiently allocating the budgets, which is practically proposed within the framework of TN design problem (TNDP). Literatures show that TNDP can be constructed in three different types, namely, the continuous network design problem (CNDP), the discrete network design problem (DNDP), and the mixed network design problem (MNDP). However, due to the real-world expansion of TN lanes is not implemented in fractional amounts, DNDP may be more appropriate [1], which is applied in this paper.

To solve TNDP, several metaheuristic-based approaches have been proposed. Though the exact solution cannot be guaranteed, they possibly provide an adequate solution in the reasonable time. Genetic Algorithm (GA) is one of popular metaheuristic, which is successfully employed to handle TNDP. As part of evolutionary algorithms, GA is inspired by the natural process coined by Darwin [2], which is represented by the pipeline processes of operators. Although the component and structure of GA have been expansively developed, recombination and mutation operators have been positioned as a basic operator assembled in GA processes. Recombination operator mimics the mating process to produce offspring, while the mutation operators represent the possibility of random change in the chromosome.

In line with the GA development, the availability of new molecular technology has permitted to provide the more information relating to the genome natural process (e.g., recombination and mutation), which may be different from that previously available. For instance, the recombination rate, which is commonly set similar for all individual in GA, is reported to behave varied in the certain characteristics [3]. Since GA is firstly motivated by the natural process, the current information may be useful for improving GA performance. This paper thus attempts to explore the recent finding of human genome, which may be used to revise the standard GA.

The rest of paper is organised as follows, in the following section, the overall modelling framework of network design is described. 3rd section demonstrates the basic procedure of GA and its adaption from recent fact in genome research. We continue with discussion relating to the application of GA-based approach for solving TNDP and the GA performance investigation. Finally, in last section, there is a summary of the methodologies, results, and analyses used in this paper.

[*] Corresponding author: febri.zukhruf@ftsl.itb.ac.id

2 Network design framework

Modelling is implemented based on the bilevel programming framework, in which the lower level represents the multiuser equilibrium flow of TN, and the upper level decides the suitable combination for improving the TN performance. This problem can be included as the TNDP, which deals with the modification of TN in order to improve the efficiency of TN. As the network modification relates to the addition of new links or expansion of current links, which is probably represented using 0–1 integer decision variables, the problem is practically recognized as DNDP. The problem is then simply expressed as follows:

$$\max_{\mathbf{u}} P(\mathbf{u},\mathbf{b},\mathbf{x}_0^*,\mathbf{x}^*) \tag{1}$$

subject to

$$\left\langle \mathbf{c}\left(\mathbf{x}^*\right), \mathbf{x}-\mathbf{x}^* \right\rangle \geq 0,\ \forall \mathbf{x} \in S \tag{2}$$

$$\left\langle \mathbf{c}\left(\mathbf{x}_0^*\right), \mathbf{x}_0-\mathbf{x}_0^* \right\rangle \geq 0,\ \forall \mathbf{x}_0 \in S_0 \tag{3}$$

where,

$\mathbf{u}$: the vector set of TN improvement actions,
$\mathbf{b}$: investment cost vector,
$\mathbf{x}$: link flows vector,
x* : equilibrium link flows vector,
$\mathbf{x}_0$: link flows vector at existing condition,
$\mathbf{x}_0^*$: equilibrium link flows vector at existing condition,
$\mathbf{c}$: link user costs vector,
S_0, S : non-empty feasible space.

Eq. (1) denotes an objective function in the upper level, which comprises an optimisation problem. Constraints (2) and (3) construct the lower level by involving the user equilibrium conditions (see Eq. (4)). Let W represent the set of O/D pairs, where the set of paths connecting *w* is symbolized by P_w. The link traffic flow of class *k* (i.e., f^k) is then indicated by the summing class *k* flow of all paths using link-*j* (i.e., x_p^k).

$$\eta_p^k\left(f^*\right)\begin{cases} = \lambda_w^k\ ,\text{if}\, x_p^k > 0 \\ \geq \lambda_w^k\ ,\text{if}\, x_p^k = 0 \end{cases} \quad \forall p \in P, k \in K, w \in W \tag{4}$$

For each class *k*, for all O/D pair *w*, and for all paths $p \in P_w$, there is a travel disutility as function of path-flow pattern (i.e., η_p^k). λ_w^k is an indicator, whose value is not previously known. As can be inferred in Eq. (4), if the path travel cost is larger than the travel disutility, the flow on that path is zero. On other side, the path flow is greater than or equal to zero, when the travel cost is equal to the travel disutility. The Frank Wolfe (FW) algorithm is then invoked for solving the traffic assignment problem, which is traditionally applied (e.g., [4]).

In the upper level, the combination of TN improvement actions is selected by maximising the ratio of reduced total travel cost and investment cost due to the implementation of action (see Eq. (5)). Let A be defined as $A = A_1 \cup A_2 \cup A_3$, where A_1 is the set of existing transport links without any modifications, and A_2 is the set of transport links with possible actions to be implemented, and A_3 is the updated set of A_2 after action implemented. Here u_a and u are defined as $u_a \in \{0,1\}$ and $u \in \{u_a | a \in A_2\}$, respectively, so that, u_a is equal to 1 if transport link a is modified, otherwise u_a is 0. The objective function can be represented as follows:

$$P(\mathbf{u},\mathbf{b},\mathbf{x}_0^*,\mathbf{x}^*) = \frac{\displaystyle\sum_{k \in K}\left[\sum_{a \in A_1 \cup A_2} x_{0a}^{k*} c_a^k\left(x_{0a}^{k*}\right) - \sum_{a \in A_1} x_a^{k*} c_a^k\left(x_a^{k*}\right) - \sum_{a \in A_2} x_a^{k*} c_a^k\left(x_a^{k*}\right)\right]}{\displaystyle\sum_{a \in A_2} u_a b_a}$$

(5)

subject to

$$c_a^k\left(x_a^k, u_a\right) = \partial^k\left[t_a\left(1+\alpha\left(\frac{x_a^k}{r_a\left(u_a\right)}\right)^{\beta}\right)\right] \tag{6}$$

where,

b_a : investment cost for modifying link-*a,*
r_a : capacity of link-*a*,
∂_a^k : time value for user – *k*,
α, β : calibrated parameters.

Eq. (6) illustrates that the travel cost is purely constructed by multiplying the time value and travel time. For handling the upper level problem, this paper then utilises the GA-based procedures as the solution techniques.

3 Genetic algorithm

3.1 Simple genetic algorithm (SGA)

GA has a long history for handling the engineering optimization problems, in which the evolution of GA has also been growth in the extensive manner. However, in this section, the simple GA (i.e., SGA) is firstly elaborated in order to identify the further improvement by considering recent finding in genome research. As the basic version of GA, SGA is equipped by standard operators, namely, recombination, mutation, and selection. GA is initialized with a group of individuals formed the population. Each individual has a varied chromosomes information, which constructs the fitness value. For creating the offspring, the recombination operator, is also known as "crossover", plays important roles by mating two individual in the population. The common approach selects two random individuals (i.e., parents) that produce two offspring, in which a single point crossover or a uniform crossover method is involved.

The mutation operator is applied for exploring the search spaces, since it possible introduces the chromosome information that is not inherited from the original population. Because the number individual in population must be kept in the consistent number, the selection process is practically invoked. The numerous methods have been proposed for selecting the successor individual, although the roulette wheel-based selection procedures are incorporated in this SGA. The general procedures for implementing the SGA in this paper are described as follows:

Step 1. Initialisation (l=1)

- Set predefined parameter values namely, number of individual (i.e., N), number of generations (i.e., pop), crossover rate (i.e., ω), and mutation rate (i.e., τ)
- Generate N individual candidate solution using a randomized algorithm.

Step 2. Randomly mate two individuals to obtain N/2 pairs of parent.

Step 3. Apply uniform crossover procedure with predetermined crossover rate in order to produce N offspring (i.e., $child_1$).

Step 4. Mutate each bit of $child_1$ offspring by considering mutation rate to yield $child_2$.

Step 5. Evaluate fitness value of each parent (i.e., N) and its offspring (i.e., $child_2$ & $child_2$).

Step 6. Select N successor individuals by conducting route wheel selection.

Step 7. Change l to l+1, if the algorithm has reached a stopping criterion (i.e., l=pop), then stop the iterations. Otherwise, return to step 2

3.2 Genome research evidence

Mutation and recombination are process in genetics field that can be regarded as the sources variability of genetics. Both processes have a different mechanism [5-6], which could change (or not change) phenotype of the living things. Taking a similar concept of both processes that play important roles in the human evolution, they are thus involved in GA for varying the solution candidate (i.e., individual) to avoid the local optimal solution.

Recombination is generally known as an exchange process of genetic material (i.e., segment of DNA molecule) that shaping the genome variation [5-7]. The process eliminates the deleterious mutation, which can be viewed as the positive effect of recombination. The recombination ability, which is affected by recombination rate, act not only as a major force of evolution, but also helps to maintain genome repair and integrity in cell division. Moreover, the current evidence found that recombination rate is different among the autosomes, sex chromosomes, mtDNA, and chloroplasts [3]. This evidence is relatively different with the practical approach of GA that set a recombination rate (i.e., crossover rate) similar for all individual.

A mutation is change in the base sequence of DNA, whether it involves a single base or many bases. Sort of mutations shows the mechanism of how the mutation convert the genome sequence. Most of mutation processes are behaved as a point mutation (i.e., single nucleotide polymorphism (SNP)) that describe a transition or replacement procedure of one purine or one pyrimidine. Other type of mutations goes through insertion or deletion mechanism of one or a few nucleotides. The transformation can produce a frameshift mutation that result in alteration reading frame group of codons. Other transformation can be named as Null mutations, which damage the gene function. Null mutation is generated by recombination, by transposition, or due to the genetic engineering. At the end it is separated the parts of a gene and inactivating the gene [8]. The mutation step of GA is commonly applied by incorporating the SNP process, in which another type of mutation may need to be tested for investigating its contribution to the GA performance, as it is elaborated in the further section.

3.3 GA adaptation

Based on the above review on the current genome evidence, it can be inferred that the exploration can be taken place in the two parts, namely:

a) Crossover rate, which is currently set in the similar number for all individual, might be arranged to be varied for each individual based on the specific characteristics (e.g., gender).
b) Mutation processes, which is relied on the SNP approach, may be investigated by applying "inserting" based approach.

Therefore, this paper proposed three revisions of SGA, in which 1st and 2nd revisions related to the crossover rate, and the last revision related to the mutation processes.

GA with varied crossover rate (GACOVAR)

The first revision of SGA is applied by varying the crossover rate of each individual. Therefore, in the initial step, the crossover rate is not set using a single value, but it is termed in the range manner. Furthermore, **Step 3a** needs to be inserted after **Step 2** in the SGA procedures for determining the crossover rate of individual. The general procedures of GACOVAR is highlighted as follows:

Step 1. Similar with SGA, except the crossover rate is set in the predetermined range.

Step 2. Similar with SGA,

Step 3a. Randomly determined the crossover rate within the predefined range of crossover rate,

Step 3-7. Similar with SGA.

GA with gender classification (GAGEN)

This proposal is inspired by the genome evidence that the recombination rate of men is higher than the women. Therefore, GA with gender classification is proposed, which is further stated as GAGEN. As different with the SGA, the population is divided into two gender groups with equal number of members. The crossover processes can only be occurred by mating two individual from different group, which may illustrate the gender mating of human. Moreover, the roulette wheel selection is also applied in each group to maintain the number of member group. The offspring gender is determined by applying the randomized algorithm. The general procedures are then described as below:

Step 1. Initialisation (l=1)

- Set predefined parameter values namely, number of individual of each group (i.e., N_1&N_2=N), number of generations (i.e., *pop*), crossover factor (i.e., ψ), crossover rate for each group (i.e., $\omega_1 = \psi\omega_2$), and mutation rate (i.e., τ)
- Generate N individual candidate solution of each group using a randomized algorithm.

Step 2. Randomly mate two individuals from different group to obtain N pairs of parent.

Step 3. Apply uniform point crossover procedure with the previous crossover rate in order to produce N offspring (i.e., $child_1$).

Step 3b. Determine the offspring gender using random algorithm and set its crossover rate.

Step 4. Mutate each bit of $child_1$ offspring by considering mutation rate to yield $child_2$.

Step 5. Evaluate fitness value of each parent (i.e., 2N) and its offspring (i.e., $child_2$ & $child_2$).

Step 6. Select N successor individuals for each group by conducting route wheel selection.

Step 7. Change l to l+1, if the algorithm has reached a stopping criterion (i.e., l=*pop*), then stop the iterations. Otherwise, return to step 2

GA with inserting mutation (GAINMUT)

In SGA, the mutation is implemented by changing the bit value from the stated condition, for instance, if bits contain 1, it is then mutated to 0, and otherwise. The last proposal is corresponded to the incorporation of the inserting mutation process in SGA. The inserting process is applied by inserting the random value of bit to the allele, which significantly alternate the chromosome information. To keep the chromosome length, the last bits information is then deleted. The SGA procedure is then revised as follows:

Step 1-3. Similar with SGA.

Step 4a. Randomly determine the mutation location and bit information.

Step 4b. Insert the random bit information of $child_1$ offspring to the mutation location by considering mutation rate to yield $child_2$. Delete the last allele to keep the chromosome length.

Step 5-7. Similar with SGA.

4 Numerical experiments

4.1 Transportation network test

To elaborate the applicability of model, as well as the performance of proposed GA, the numerical experiments are conducted. The experiments involve a small transportation network (see Figure 1), which is used by passenger car (PC) and heavy vehicle (HV) users.

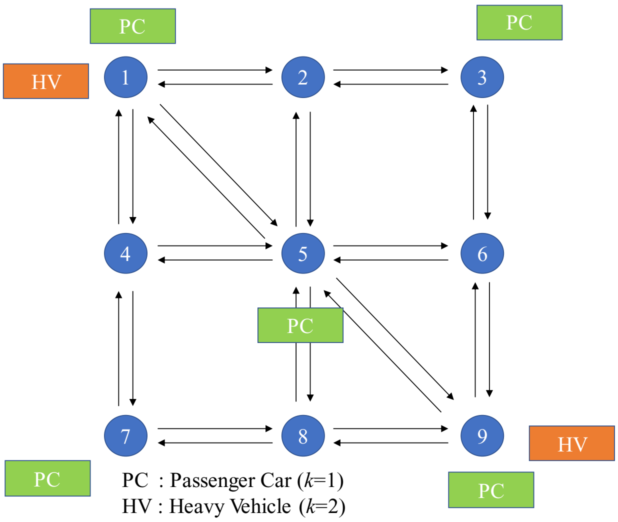

Fig. 1. Test network

The network consists 5 pair OD of PC, 2 pair OD of HV, 9 nodes, and 28 links. For the easiness interpretation, the link capacity is set in the similar value for all links, which is also applied in the similar manner for the demand of OD.

Table 1. TN Improvement action

No	Links (Node A, Node B)	Type
1	(1,2) & (2,1)	Cap. Expansion
2	(3,2) & (2,3)	Cap. Expansion
3	(3,6) & (6,3)	Cap. Expansion
4	(9,6) & (6,9)	Cap. Expansion
5	(9,8) & (8,9)	Cap. Expansion
6	(8,7) & (7,8)	Cap. Expansion
7	(4,7) & (7,4)	Cap. Expansion
8	(4,1) & (1,4)	Cap. Expansion
9	(4,5) & (5,4)	Cap. Expansion
10	(2,5) & (5,2)	Cap. Expansion
11	(8,5) & (5,8)	Cap. Expansion
12	(6,5) & (5,6)	Cap. Expansion
13	(1,5) & (5,1)	Link Addition
14	(3,5) & (5,3)	Link Addition
15	(7,5) & (5,7)	Link Addition
16	(9,5) & (5,9)	Link Addition

As the TNDP aims to enhance the performance of TN, several improvement actions are proposed, which is summarized in Table below. The action includes the link capacity improvement and the link addition. Moreover, the implementation cost for adding link is five times higher than the capacity expansion of link.

The improvement action can be executed by implementing a single action or its combination, and hence, the problem to decide the suitable action become more complex. The total available combination can be easily estimated as 65,535 (i.e., 2^16-1). This combination with a very small TN practically can be handled by the exact-based approach (e.g., complete numeration). However, if the combination and TN-sizes increases, the computation time will be extensively growth, which is not practical for decision making processes. Hence, the metaheuristics-based approach is generally invoked.

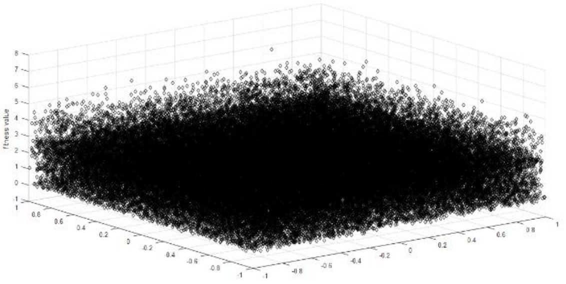

Fig. 2. Complete enumeration result

Figure 2 shows the result plot of complete enumeration procedures, in which for evaluating 65,535 combinations in a PC of core i7 with 16 GB RAM, it takes time around 6.6 hours. In addition, the enumeration method provides the exact solution for this TNDP, which is equal to 7.00.

Table 2. Performance comparison in SGA (10 runs)

Mutation Rate	**Cross Rate**				
	0.1	**0.3**	**0.5**	**0.7**	**0.9**
0.01					
Best	5.51	7.00	6.26	6.27	7.00
Average	4.74	5.21	5.23	5.68	5.63
Worst	4.15	4.12	4.14	4.51	4.52
0.05					
Best	6.12	6.26	6.27	7.00	7.00
Average	4.92	5.44	5.72	5.70	6.35
Worst	3.76	4.10	4.25	4.26	4.52
0.1					
Best	6.15	7.00	7.00	7.00	7.00
Average	5.42	5.32	5.95	5.68	5.82
Worst	3.85	4.48	4.53	4.49	4.99

4.2 GA-based performance comparisons

The result that is provided by exact-based method, is then used for evaluating the performance of SGA, and the proposed GA (i.e., GACOVAR, GAGEN, and GAINMUT). To ensure the fair comparison, the best parameter setting, which significantly influences the performance, is firstly seek by applying the sensitivity analysis.

Table 2 provides an example of SGA sensitivity analysing for finding the suitable setting of crossover rate and mutation rate. The test is conducted in 10 runs by evaluating the best, average, and worst solution, which is resulted by SGA. The maximum possible of solution is set to 4500 as it is implemented by previous researches [9-10]. Utilizing the identical approach, the proper setting of parameters is thus decided for all GA-based methods, which can be summarized in Table 3.

Table 3. Resume of setting parameter

	SGA	GACOVAR	GAGEN	GAIMUT
N	50	60	25	100
Pop	30	25	30	15
ω_1	0.9	0.5-0.8	0.5	0.2
ω_2	-	-	0.8	-
τ	0.05	0.06	0.07	0.09

Applying the best parameter setting to handle the TNDP, it can be inferred from Table 4 that all approaches within 10 runs can offer the good solution as the exact-based approach. Although, the stability issue is remained, in which most of GA-based method experienced to be trapped in the local optimal, except GAGEN. The group division mechanism in the GAGEN seem influence its stability. The division mechanism may control the diversification of chromosome, which tend to widely explore the search spaces. Hence, it can be concluded that GAGEN provides a better result when it is applied in the relatively small TN and action combinations. In addition, the GA-based approaches significantly reduce the computation time compare to the exact-based approach.

Table 4. Performance comparisons among GA-based approaches (10 runs)

	SGA	GACOVAR	GAGEN	GAIMUT
Best	7.00	7.00	7.00	7.00
Ave.	6.35	4.14	7.00	5.63
Worst	4.52	2.82	7.00	3.04
Ave. compt. time (s)	38.01	57.80	113.82	98.21

5 Conclusion

Transportation network design problem has long been recognised as the most challenging problem in transportation field due to the TN complexity and its combination of action. Metaheuristic-based approaches are practically invoked to handle this problem, including GA with its recombination and mutation operators. Such operators are inspired by the natural processes of organism genome, which has achieved a remarkable development in recent years. This paper then presents three variants of GA, which is driven by the current evidence of human genome. These variants adjust recombination and mutation processes, which is commonly applied in the standard GA. The numerical experiment in the relatively small TN reveals that the diversion of crossover rate within a certain group of population poses a better result compare to standard GA.

References

1. D.E. Boyce and B.N. Janson, *A discrete transportation network design problem with combined trip distribution and assignment*, Transportation Research Part B, Vol. **14B**, pp. 147–154, (1980)

2. C. Darwin, *The Origin of Species: By Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life*, reissue ed., Murray, London, (1859).
3. I. Alves, A.A. Houle, J.G. Hussin, P. Awadalla, *The impact of recombination on human mutation load and disease*, Philos. Trans. R. Soc. Lond., B, Biol. Sci. **372**,1736, (2017).
4. A. Chen, J.S. Oh, D. Park, W. Recker, *Solving the bicriteria traffic equilibrium problem with variable demand and nonlinear path costs,* Applied Mathematics and Computation, **217** (7), 3020–303, (2010).
5. T. A. Brown, *Mutation, Repair and Recombination,* In: Genomes 2nd edition. Oxford: willey liss. 2002.
6. N.H. Barton*, Mutation and the evolution of recombination*, Philos Trans R Soc Lond B Biol Sci; **365** (1544): 1281–1294, (2010).
7. S. Gupta and L. Yel, *Molecular Biology and Genetic Engineering*, In : Middleton's Allergy: Principles and Practice 8th edition, **10**, 162-18. Saunders. (2014).
8. P.R. Murray, K.S. Rosenthal, and M.A. Pfaller, *Bacterial Metabolism and Genetics*, In: Medical Microbiology Eighth Edition. Philadelphia: Elsevier Inc. (2016)
9. T. Yamada, Z. Febri, *Freight transport network design using particle swarm optimisation in supply chain–transport supernetwork equilibrium,* Transportation Research Part E: Logistics and Transportation Review **75**, 164–187, (2015).
10. Z. Febri, T. Yamada, and E. Taniguchi, *Designing Cocoa Transport Networks using A Supply Chain Network Equilibrium Model with the Behaviour of Freight Carriers*, J. JSCE, Ser. D3 (Infrastructure Planning and Management), Volume **70**(5), pp. 709–722, (2014)