A Group Theoretic Comparison of Different Mutation Operators on Evolution Algorithm to Travel Salesman Problem

Zhao Zhixiang 11711621
Computer Science and Technology
11711621@mail.sustech.edu.cn

***Abstract*—The traveling salesman problem(TSP) is a combinatorial optimization problem that can be mathematically modeled as a problem of searching the optimal permutation. This research describes a model for mutation in the evolution algorithm(EA) and adopts group theory to the demonstration of the global optimal promise for mutation. Also, this research uses a simple framework of EA to compare the performances of different mutation operators. The experimental results show the degeneration under this simple framework and all the mutation operators perform the same as each other.**

I. Introduction

A. Background

Evalution computing is a useful tool for optimization, especially when the objective function treated to be a blackbox. One vast and important area in combinatorial optimization is traveling salesman problems. If the solutions of this problem are codified as permutations, then it is tricky to design a mutation operatror for evalution algorithm. All possible directions of mutation form a group called permutation group or symmetric group in group theory. This kind of group has a specific structure that brings challenges for mutation.

Meanwhile, most of the real world problems of combinatorial optimization are large and complex, but they require timely solutions. For mutation operators, it is important to know whether it can reached the optimal solution or how fast it will reach. The structure of the solution space, permutation group, may give us a heuristic methods to design a mutator ensuring the global optimal solution and efficiency.

This research studies the mutation operator in evalution algorithm to solve a specific combinatorial optimization problem known as the traveling salesman problem(TSP).

B. Problem Discription

Traveling salesman problem, or TSP for short, is to find the cheapest circuit of given collection of visiting cities and the cost of travel between each pair of them.

There are two types of TSP. One is symmetric, which means the cost of traveling from city x to city y is the same as the cost of traveling from y to x. The other is not symmetric, which means the costs are not the same.

Given a starting city, we can select the second city in $n-1$ left cities. Likewise, there are $n-2$ choices for the third city. By simple inductive argument we have $(n-1)\times(n-2)\cdots\times 1=(n-1)!$ tours to be the potential solution. For symmetric TSP, the possible tours divided by 2 and we get $(n-1)!/2$ choices.

Formally speaking, the question is divided into two parts:

1) S_n containing all the possible solutions. $|S_n|=(n-1)!$
2) $\forall\sigma\in S_n$ there is a map M s.t. $M(s):S_n\to\Re$

we want to minimize M to find:

$$\arg\min_{\sigma} M(\sigma)$$

In computational mathematics, there is not yet a effective solution for the general case.

II. Literature Review

A. Evolution Computation [5]

Evolutionary computation is the study of computational systems which use ideas and get inspirations from natural evolution and adaptation. It aims at understanding such computational systems and developing more robust and efficient ones for solving complex real-world problems.

All evolutionary algorithms have two prominent features which distinguish themselves from other search algorithms. First, they are all population-based. Second, there is communications and information exchange among individuals in a population. Such communications and information exchange are the result of selection and/or recombination in evolutionary algorithms.

Algorithm 1 A general framwork of evolutionary algorithm

1: Set i = 0
2: Generate the initial population P(i) at random with size of 10.
3: **while** the population do not converge or the maximum time is not reached. **do**
4: Evaluate the fitness of each individual in P(i);
5: Select parents from P(i) based on their fitness;
6: Apply **search operator** to the parents and produce gerneration P(i+1);
7: **end while**

B. Group Theory [4]

A **group** is an axiomatic algebraic structure that ensures unique solutions to simple algebraic equations.

Definition 1. *A **binary operation** is a rule that assigns each ordered pair(a, b), $a, b \in G$,to a unique element also in G, which implies * is closed on G.*

1) *a binary operation * on a set G is a function *: G × G → G. For any a,b ∈ G we shall write a*b for *(a,b)*
2) *A binary operation * on a set G is **associative** if for all a,b,c ∈ G we have $a * (b * c) = (a * b) * c$*
3) *If * is a binary operation on a set G we say elements a and b of G commute if a*b = b*a. We say * (or G) is **commutative** if for all a,b ∈ G, a*b = b*a*

Definition 2. *A **group** is an ordered pair (G,*) where G is a set and * is a binary operation on G satisfying the following axioms:*

1) *(a*b)*c = a*(b*c) for all a,b,c ∈ G, i.e., * is **associative**.*
2) *there exists an element e in G, called an **identity** of G such that for all a ∈ G we have a*e = e*a = a*
3) *for each a ∈ G there is an element a^{-1} of G, called an **inverse** of a such that $a * a^{-1} = a^{-1} * a = e$*

Definition 3. *Let S be a nonempty set and $A(S)$ the set of all bijections $S \to S$. Under the operation of composition of functions, $f \circ g$, $A(S)$ is a group, since composition is associative, composition of bijections is a bijection. The elements of $A(S)$ are called permutations and $A(S)$ is called the group of permutations on the set S. If $S = 1, 2, 3, ..., n$, then $A(S)$ is called the **symmetric group on n letters** and denoted S_n.*

Definition 4. *Let $i_1, i_2, ..., i_r, (r \leq n)$ be distinct elements of $I_n = 1, 2, ...n$. Then $(i_1, i_2, i_3 \dots i_r)$ denotes the permutation that maps $i_1 \mapsto i_2, i_2 \mapsto i_3, i_3 \mapsto i_4, ..., i_{r-1} \mapsto ir, and i_r \mapsto i_1$, and maps every other element of I_n onto itself. $(i_1, i_2, \dots, i_r)$ is called a cycle of length r or an r-cycle; a 2-cycle is called a **transposition**.*

Definition 5. *The permutations $\sigma_1, \sigma_2, ..., \sigma_r$ of S_n are said to be disjoint provided that for each $1 < i < r$, and every $k \in I_n$, $\sigma_i(k) \notin k$ implies $\sigma_j(k) = k$ for all $j \notin i$.*

Theorem 1. *Every permutation in S_n can be wrirren as a product of(not necessarily disjoint) transpositions.*

Corollary 1.

1) *S_n is generated by the $n - 1$ transpositions $(12), (13), (14), ..., (1n)$.*
2) *S_n is generated by the $n - 1$ transpositions $(12), (23), (34), ..., (n - 1\, n)$.*
3) *S_n is generated by $\sigma = (12)$ and $\tau = (123 \dots n)$.*
4) *S_n is generated by $\sigma = (12)$ and $\tau = (23 \dots n)$.*

III. Related Work

A. Tabu Search [3]

Group theoretic tabu search (GTTS) studies TSP as a permutation problem rather than an integer program by applying the principles of group theory to define the search move and neighborhood structure. This research demonstrates an implementation of group theory to metaheuristic continuously finds near-optimal solutions to TSP. Especially, the concept of conjugation in group theory simplifies the move definition as well as the intensification and diversification strategies.

B. Random Walk [1]

Definition 6. *In group theory, an element of a group is an **involution** if it has order 2; i.e. an involution is an element a such that $a \neq e$ and $a^2 = e$, where e is the identity element.*

The number $a(n)$ of involution in the set containing n elements are given by a recursive relation:

$$a(0) = a(1) = 1$$

$$a(n) = a(n-1) + (n-1) \times a(n-2), for \quad n > 1$$

The first few items of this sequence are 1, 1, 2, 4, 10, 26, 76, 232 (See in OEIS-A000085).

Hence swap and reverse are involution mutation operators.

This study examines how the natural concept "parallelization" affects the convergence rate of random walking in the symmetric group to the stationary of convergence.

The base walk in this paper parallelizes is the p lazy random transposition walk. It has as generators the identity with probability p and a uniformly random transposition with probability 1-p. This is equivalent to putting n cards on the table and with probability 1-p swapping a random pair.

The transposition walk for $p = \frac{1}{n}$ takes order $\frac{1}{2} n \log(n) + cn$ steps to converge to its uniform stationary distribution.

C. Generating Permutation [2]

This research answer the following question:

how many transposions are needed until the permutation is close to random ?

More formally, a random transposition (lazy) is modeled by a probability measure T on the symmetric group S_n

$$T(e) = 1/n \quad \text{If e is the identity} \tag{1}$$

$$T(j) = 2/n^2 \quad \text{If j is a transposition} \tag{2}$$

$$T(k) = 0 \quad \text{otherwise} \tag{3}$$

Then after $k > \frac{1}{2} n \log n$ times "swap" of this kind, the probability of permutation generated, denoted T^{*k}, will close to uniform.

By using variation distance to measure two probability distribution, there is a upper bound,

$$||T^{*k} - U|| \leq be^{-2c}$$

when $n \leq 10$ and b is a positive constant,

$$c = \frac{k - \frac{1}{2}n \log n}{n}$$

IV. Methodology

A. selection schemes

Definition 7. *rank(x) = x's rank by M(x) in population. Rank 0 indicates highest M(x) and rank $\mu - 1$ indicates the lowest M(x) for population size μ.*

This research uses two selection schemes:

- Simple Rank-based Selection:

$$Prob(x) = \frac{rank(x) + 1}{\Sigma_{x' \in P}(rank(x' + 1))}$$

- Exponential Ranking Selection:

$$Prob(x) = \frac{1 - e^{-rank(x)}}{C}$$

where $C = \mu - \frac{e - e^{-\mu+1}}{e-1}$

B. Mutation operator (Search operator)

Definition 8. *Mutation Model:*
a mutation operator is a combination of $\sigma_i \in S_n$ with probabilty distribution.
Consider a pair $(\sigma_i, P(\sigma_i)), \sigma_i \in N$ for a set $N \subset Sn$.

This research compare the designed mutation operator as showed following:

1) Swap: (ij) with uniform probability $U((ij)) = \frac{1}{(n-1)\times n}$
2) k-swap: Doing the k times of swapping in each generation, k is a positive integer.
3) reverse-L: select a pattern of length L and proceed the reverse. i.e. swap the items with index $i, i+1, ..., floor(i+j/2)$ and the items with index $j, j-1, ..., ceil(i+j/2)$.
4) shuffle: generate a random permutation σ with the probability of $U(\sigma) = 1/n!$
5) Mutator001: $(12), (13) \dots (1n)$ with uniform probability $U(\sigma_i) = \frac{1}{n}$
6) Mutator002: $(12), (13) \dots (1n)$ with uniform probability $U(\sigma_i) = \frac{1}{n}$
7) Mutator003: $\sigma = (12)$ and $\tau = (23 \dots n)$ with uniform probability $P(\sigma) = \frac{1}{n}$ and $P(\sigma) = \frac{n-1}{n}$

C. Experimental Algorithm

D. Experimental Setup

The experiment is set up as followings:

- Three TSPs' maps: Western Sahara with 29 Cities, Djibouti with 38 Cities and Qatar with 194 Cities.
- 50000 generations
- expected tour length: 29
- Population size: 10

This research runs the algorithm 50 times for each case and compute the average of each generation.

Algorithm 2 Experimental evolutionary algorithm

```
Set i = 0
Generate the initial population P(i) at random;
while genneration number is less than 50,000 do
    Evaluate the fitness of each individual in P(i) by selection schemes;
    Select parents from P(i) based on their fitness;
    Apply mutation operator to the parents and produce gerneration P(i+1);
    Generation number += 1
end while
```

V. Results

A. Experiment Results

The result are shown in the figure 1, 2 and 3.

B. Analytical Conclusion

1) What is the goal of mutation? The goal of the mutation in nature is not to produce good cases but to produce a random case. There is not invariant goodness or badness. But the randomization contains all the possible goodnesses and badnesses. Hence it will always lead to a good result if the time is enough. However, if we want to mutate in a single direction to the best result of our desire, we will break the balance of mutation down. Who breaks this down? The selection scheme.
   But in this case, the selection scheme was supposed to work but it did not. And I do not know why.
   Define the search path as a squence of f in Sn such that $\sigma = f_1 f_2 \dots f_r$ for σ is the best solution. The length of search path is r.
   The selection scheme would select the individuals in the population with a relatively shorter search path. But it seems like the selected individuals could not continue to keep "excellent" after the next mutation, which means the search operation (mutate and then select) is not transitive.
   I think the reason why it is not transitive is that randomness of mutation breaks the transitive chain on one hand, and on the other hand, we cannot conclude that the solution is close to the best solution by the distance of map M. For example, We find a solution with a lower fitness, but it does not mean the solution is closer to the best solution in S_n. The distance is different between S_n and Re.
   In short, the reason why this algorithm degenerates to the simple generating and test method is like the above said.
2) The swap mutator might be better than a reverse mutator if we control the variable. That is to say, let the times of "swap" in reverse mutator to be the same with swap mutator. The behind reason is that it is possible to swap and reverse back (involution). Therefore which will have higher performance is determined by which has a lower probability of going back as a cycle.

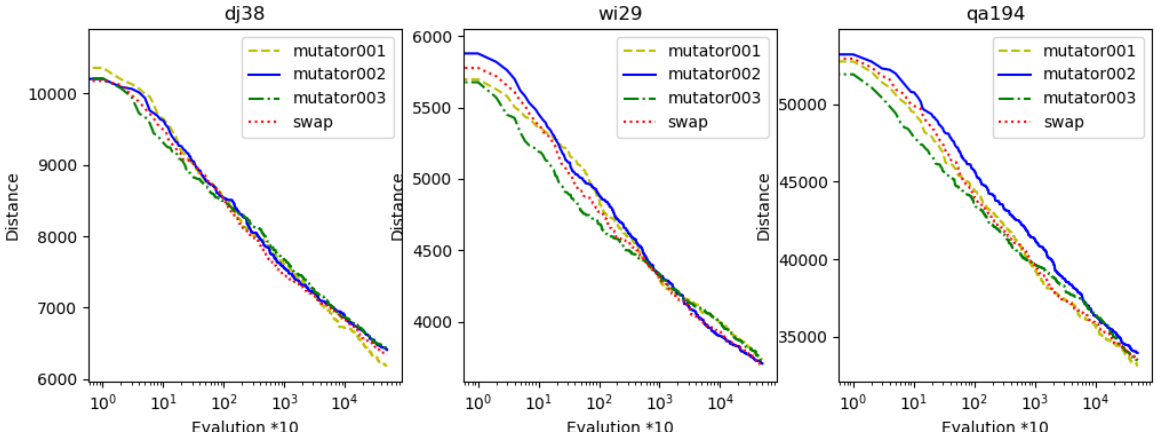


Fig. 1. EA-simple rank selection

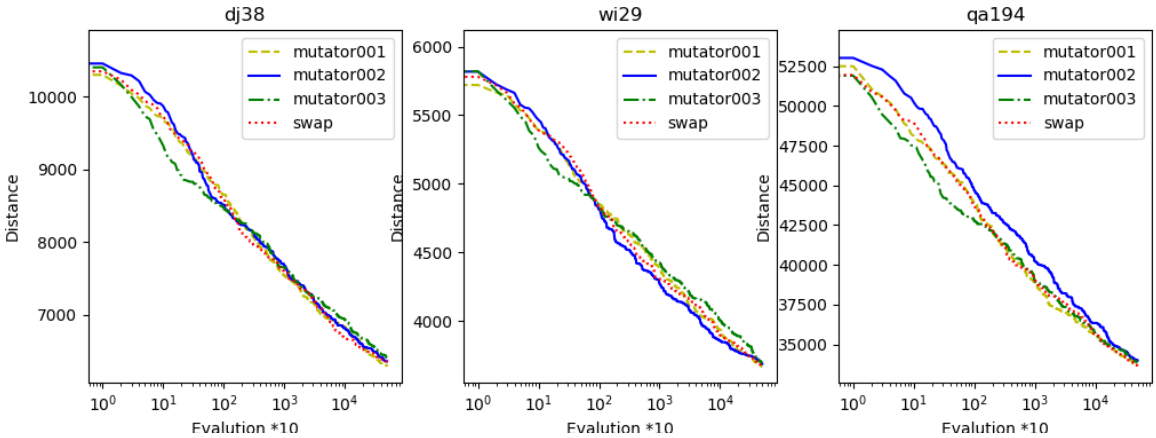


Fig. 2. EA-Exponential rank selection

3) Diversity of combination mutators might be lead to a better performance in converge since it can reduce the circulation (going back to origin).

VI. Appendix

A. Proof for Theorem 1 [4]

Let $\sigma \in Sn, \sigma \neq (1)$. Verify that the following is an equivalence relation on I_n: for $x, y \in I_n$, x y if and only if $y = \sigma^m(x)$ for some $m \in Z$. The equivalence classes $B|1 < i < s$ of this equivalence relation are called the orbits of $sigma$ and form a partition of I_n. Let $B_1, B_2, ..., B_r(I < r < s)$ be those orbits that contain more than one element each $(r \geq 1 since \sigma \neq (1))$. For each $i < r$ define $\sigma_i S_n$ by:

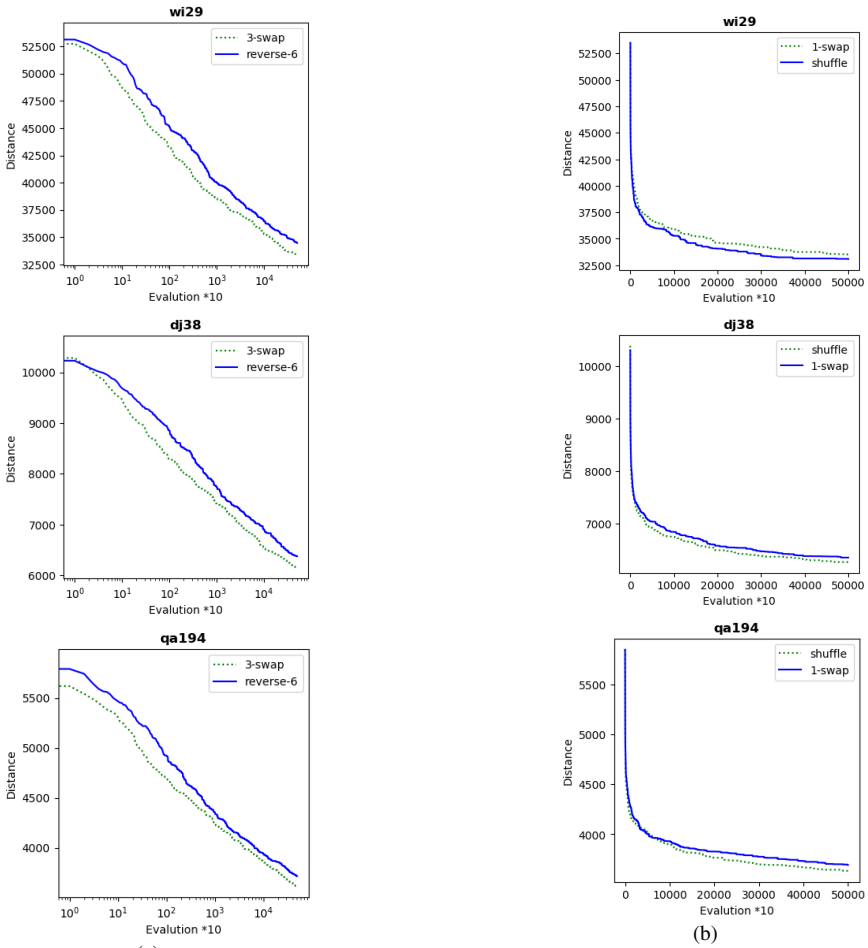
$$\sigma_i(x) = \begin{cases} \sigma(x), & \text{if } x \in B_i \\ x, & \text{if } x \notin B \end{cases} \tag{4}$$

Each σ_i is a well-defined nonidentity permutation of I_n since $\sigma|B_i$ is a bijection $B_i \to B_i$, σ_1, σ_1, σ_2,...,σ_r are disjoint permutations since the sets $B_1, ..., B_r$ are mutually disjoint. Finally verify that $\sigma = \sigma_1\sigma_2 \ldots \sigma_r$; (note that $x \in B_i$ implies $\sigma(x) = \sigma_i(x)$ if $i < r$ and $\sigma(x) = x$ if $i > r$; use disjointness). We must show that each σ_i is a cycle.

If $x \in B_i(i < r)$, then since B_i is finite there is a least positive integer d such that $\sigma^d(x) = \sigma^i(x)$ for some $j(O < j < d)$. Since $\sigma^{d-i}(x) = x$ and $0 < d-j < d$, we must have j = 0 and $\sigma^d(x) = x$. Hence $(x\sigma(x)\sigma^2(x) \ldots \sigma^{d-1}(x))$ is a well-defined cycle of length at least 2. If $\sigma^m(x) \in B_i$, then $m = ad + b$ for some $a, b \in Z$ such that $a \leq b < d$. Hence $\sigma^m(x) = \sigma^{b+ad}(x) = \sigma^b\sigma^{ad}(x) = \sigma^b(x) \in x, \sigma(x), \sigma^2(x), ..., \sigma^{d-1}(x)$. Therefore $B_i = x, \sigma(x), \sigma^2(x), ..., \sigma^{d-1}(x)$ and it follows that σ_i is the cycle

$$(x\sigma(x)\sigma^2(x) \ldots \sigma^{d-1}(x)).$$

Suppose $\tau_1, ..., \tau_t$ are disjoint cycles such that $\sigma = \tau_1\tau 2 \ldots \tau_t$ Let $x \in I_n$ be such that $\sigma(x)$ x. By disjointness there exists a unique $j(1 < j < t)$ with $\sigma(x) = \tau_i(x))$. Since $\sigma\tau_i = \tau_j\sigma$, we have $\sigma^k(x) = \tau_j^k(x))$ for all $k \in Z$. Therefore, the orbit of x under τ_i is precisely the orbit of x under σ, say B_i. Consequently, $\tau_i(Y) = \sigma(y)$ for every $y \in B_i$ (since $y = \sigma^n(x) = \tau_j^n(x)$ for some $n \in Z$). Since τ_i



(a) (b)

Fig. 3. Swap, Reverse and Degeneration

is a cycle it has only one nontrivial orbit, which must be B_i since $x \neq \sigma(x) = \tau_j(x)$. Therefore $\tau_i(y) = y$ for all $y \notin B_i$, whence $\tau_j = \sigma_i$. A suitable inductive argument shows that $r = t$ and (after reindexing) $\sigma_i = \tau_i$ for each $i = 1, 2, ..., r$

And we know that $(x_i) = (x_1x_2)(x_1x_2)$ and for $r > 1$, $(x_1x_2x_3 \dots x_r) = (x_1x_r)(x_rx_{r-1}) \dots (x_1x_3)(x_1x_2)$. Hence every cycle is a product of transpositions. ■

B. Proof for Collary 1

Lemma 1. *If $\sigma = (i_1i_2 \dots i_r) \in S_n$ and $\tau \in S_n$, then $\tau\sigma\tau^{-1}$ is the r-cycle $(\tau(i_1)\tau(i_2) \dots \tau(i_3))$.*

Proof:

Notice$(i_1 \dots i_r) = (i_1i_r)(i_1i_{r-1}) \dots (i_1i_r)$;therefore

$$\tau\sigma\tau^{-1} = \tau(i_1i_r)\tau^{-1}\tau(i_1i_{r-1})\tau^{-1} \dots \tau(i_1i_2)\tau^{-1}$$

Now let $\tau(i_si_t)\tau^{-1} = v$ Next $\tau(i_si_t) = v\tau$, so $\tau(i_si_t)(i_s) = v\tau(i_s)$ and therefore $\tau(i_t) = v(\tau(i_s))$; likewise $\tau(is) = v(\tau(i_t))$; and finally i_k, $k \neq s, t$, implies $\tau(i_k) = i_k = v(\tau(i_k))$. Whence $v = (\tau(i_s)\tau(i_t))$. Applying this to the whole we notice:

$$\tau\sigma\tau^{-1} = (\tau(i_1)\tau(i_r))(\tau(i_1)\tau(i_{r-1})) \dots (\tau(i_1)\tau(i_2))$$

$$= (\tau(i_1)\tau(i_2) \dots \tau(i_r))$$

■

1) Since $(1i)(1j)(1i) = (ij)$, $(12), (13), (14)$,...,$(1n)$ can generate S_n by theorem 1.
2) Since $(1\ j-1)(j-1\ j)(1\ j-1) = (1j)$, $(12), (23), (34)$,...,$(n-1\ n)$ can generate S_n by 1)
3) Notice $\tau^{-1} = (n \dots 21)$, Define $\sigma_i = \tau^{i-1}\sigma_1\tau^{-i+1} = (ii+1)$ By Lemma 1. According to (2), S_n is generated by $\sigma = (12)$ and $\tau = (123 \dots n)$.
4) Likewise (3) apply lemma 1 we get $(12), (13), (14), ..., (1n)$ Hence S_n is generated by $\sigma = (12)$ and $\tau = (23 \dots n)$ by (1).

■

REFERENCES

[1] BERNSTEIN, M. A random walk on the symmetric group generated by random involutions. *Electron. J. Probab. 23* (2018), 28 pp.

[2] DIACONIS, P., AND SHAHSHAHANI, M. Generating a random permutation with random transpositions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete 57* (1981), 159–179.

[3] HALL, S. A group theoretic tabu search approach to the traveling salesman problem. 108.

[4] HUNGERFORD, T. *Algebra*, eighth ed. Springer, 2003.

[5] SARKER, R., AND YAO, X. *Evolutionary Optimization*. Kluwer Academic Publishers, USA, 2002.