This advance publication article will be replaced by the finalized version after proofreading.
LETTER

Edge Assembly Crossover Incorporating Tabu Search for the Traveling Salesman Problem

Maaki SAKAI†, Kanon HOKAZONO††, and Yoshiko HANADA†††, Nonmembers

SUMMARY In this letter, we propose a method to introduce tabu search into Edge Assembly Crossover (EAX), which is an effective crossover method in solving the traveling salesman problem (TSP) using genetic algorithms. The proposed method, called EAX-tabu, archives the edges that have been exchanged over the past few generations into the tabu list for each individual and excludes them from the candidate edges to be exchanged when generating offspring by the crossover, thereby increasing the diversity of edges in the offspring. The effectiveness of the proposed method is demonstrated through numerical experiments on medium-sized instances of TSPLIB and VLSI TSP.

key words: traveling salesman problem, genetic algorithm, tabu search

1. Introduction

For tackling combinatorial optimization problems using a Genetic Algorithm (GA) [1], it is important to design a crossover method that effectively passes on good partial solutions (traits) to offspring. In the past, a number of crossover methods that focus on the inheritance of edge traits have been proposed for solving Traveling Salesman Problem (TSP) [2]–[6]. Edge Assembly Crossover (EAX) has proven to be the most successful crossover method among them [3], [6].

EAX maintains preferable traits by generating offspring with only edges of both parents to the extent possible while obtaining new traits by limited recombination of the edges between the two. EAX can search the solution space with high accuracy by increasing the diversity of individuals in the population, i.e., the diversity of edges included in the population. In [6], Nagata et al. have focused on the diversity of trait inheritance and introduced a survival selection strategy for offspring that form the next generation in order to ease the loss of edge diversity in the population. This strategy has high search performance for very large instances of TSP, outperforming other promising heuristic methods [6]. In addition to the trait inheritance, the diversity of trait acquisition is also important. Increasing the diversity of offspring generated throughout the search is expected to improve the search performance. However, a direct approach that enhances the diversity of offspring in EAX has not yet been proposed.

In this letter, to improve the diversity of exchanged edges, i.e., the diversity of offspring, in EAX, we propose EAX-tabu [7], which incorporates a tabu search algorithm [8] into the EAX process. In the proposed method, the edges exchanged by EAX over the previous several generations are recursively recorded for each individual as tabu edges. The proposed method prohibits to exchange the tabu edges when generating offspring, which prevents bias in the exchanged edges while ensuring the acquisition of diverse traits throughout the search. In this letter, we verify the effectiveness of EAX-tabu for medium-scale instances of TSPLIB [9] and VLSI TSP [10]. Through a series of experiments, we show that incorporating the tabu search algorithm into EAX improves the search performance.

2. Traveling Salesman Problem

TSP can be described as follows: Given a set of n cities (vertices) \( V = \{v_1, \ldots, v_n\} \) and distances between cities \( w(v_i, v_j) \) \((1 \leq i, j \leq n)\), we need to find the shortest tour forming Hamilton cycle \( H = (v_1, \ldots, v_n) \), where \( v(i) \) denotes the \( i \)th visited city starting from an arbitrary city in \( V \). If we let \( v_i \) and \( v_j \) be the edge between \( v_i \) and \( v_j \), then a tour is represented as a set \( E = \{(v_i, v_{i+1}) \mid 1 \leq i \leq n, v_{i+1} = v_1\} \) of edges. The objective function value is defined by the tour length calculated by \( \sum_{i=1}^{n} w(v_i, v_{i+1}) \).

3. Edge Assembly Crossover

3.1 Flow of EAX

By incorporating edges of one parent into the tour of another parent, EAX acquires new traits without destroying favorable traits of both the parents with relatively limited change. Let \( E_A \) and \( E_B \) be sets of edges constituting the tours of parent individuals \( p_A \) and \( p_B \), respectively. The flow of EAX that generates offspring \( c_A \) and \( c_B \) from \( p_A \) and \( p_B \) is described as follows.

Step 1 Construct a multigraph \( G_{A,B} = (V, E_A \cup E_B) \).

Step 2 Apply \( AB \)-cycle decomposition (discussed in Section 3.2) to \( G_{A,B} \) and obtain a set \( D \) of \( AB \)-cycles.

Step 3 Generate a subset \( E_{set} \) of \( D \) according to some criterion. Let \( g_{ab}^{(i)} \) and \( E^{(i)} \) \((1 \leq i \leq |E_{set}|)\) be \( i \)th \( AB \)-cycle constituting \( E_{set} \) and the set of edges of each \( g_{ab}^{(i)} \), respectively.

Step 4 Generate intermediate individuals \( I_A \) and \( I_B \) by applying \( E_A \leftarrow E_A \oplus E^{(i)} \) and \( E_B \leftarrow E_B \oplus E^{(i)} \) \((1 \leq i \leq |E_{set}|)\) to the tours of \( p_A \) and \( p_B \), respectively.

Step 5 Apply Repairing (discussed in Section 3.3) to \( I_A \) and \( I_B \) and obtain offspring \( c_A \) and \( c_B \).

†††MicroAd, Inc.
††Kansai University
a) E-mail: hanada@kansai-u.ac.jp

Copyright © 200x The Institute of Electronics, Information and Communication Engineers
3.2 AB-cycle Decomposition

AB-cycle, represented by \( g_{AB} \) in the flow of EAX, is a closed path consisting of an even number of edges, which can be obtained by alternately tracing edges in \( E_A \) and \( E_B \) on the multigraph \( G_{AB} \). AB-cycle is the most important element of EAX, which is used in extracting candidate edges to be exchanged between parents. The procedure of the AB-cycle decomposition in Step 2 of the flow is shown in Algorithm 1. In this algorithm, \( d(v) \) denotes the degree of each city \( v \) in \( G_{AB} \); all initial values are 4. Lines 4 to 12 are the operations to extract one AB-cycle from \( G_{AB} \). Here, each AB-cycle is treated as a graph \((V', E')\), where \( V' \) and \( E' \) are subsets of cities \( V \) and edges \( E_A \cup E_B \) in \( G_{AB} \), respectively. All cities and edges in \( G_{AB} \) are decomposed so that they belong to one of the AB-cycles, yielding a set \( D \) of AB-cycles.

Each AB-cycle constituting \( D \) represents a set of candidate edges to be exchanged. In actuality, EAX generates \( E\) set which is a subset of \( D \). Thus, the exchanged edges between parents correspond to edges of the AB-cycles included in \( E\set \). There are two simple ways to construct \( E\set \): EAX-RAND [3] and \( k \)-multiple strategy [6]. EAX-RAND incorporates AB-cycles included in \( D \) into \( E\set \) with a probability of 0.5. The \( k \)-multiple strategy selects a small number of AB-cycles randomly from \( D \) to generate offspring locally. EAX-RAND can generate a wide variety of offspring. However, the size of \( E\set \) follows a binomial distribution, so that an extremely small \( E\set \) is hard to construct.

We here consider a method to construct \( E\set \) of arbitrary size with equal probability to enhance the local search property while keeping the diversity of offspring. The size of \( E\set \) is uniformly, randomly selected from among \([1, |D|]\) for each pair of parents. Then AB-cycles are randomly chosen from \( D \) according to the selected number to make up \( E\set \). Hereafter, this method is referred to as EAX-UNIFORM.

3.3 Intermediate Individual and Repairing

In Step 4 of the flow, for all edges included in \( E\set \), EAX deletes edges belonging to \( E_A \) from the tour of \( p_A \) and adds edges belonging to \( E_B \), thereby generating a set of multi-decomposed subtours, or partial cycles. The set thus obtained is called intermediate individual. Let intermediate individual generated from \( p_A \) by this operation be \( I_A \). This operation of exclusive superposition of edges in \( E\set \) into the tour of one parent is also used to generate the intermediate individual \( I_B \) from \( p_B \).

The Repairing is then applied to these generated intermediate individuals in Step 5. The procedure of Repairing is shown in Algorithm 2, where intermediate individual \( I \) is represented as a set of subtours \([S_1, \ldots, S_m]\) and the distance of edge \( e_{ij} = (v_i, v_j) \) is abbreviated as \( w(e_{ij}) \). This operation iteratively connects subtours of an intermediate individual so that a single feasible tour is formed as an offspring. Each iteration generates edges that connect two close subtours with the shortest distance between them.

4. Proposed Method

As the population moves toward convergence, the exchange edges in EAX become biased and the ability to generate new traits decreases. To ease the bias, we propose EAX-tabu by incorporating a tabu search scheme into EAX [7]. Tabu search is a local search that works effectively on combinatorial optimization problems, including TSP [8]. EAX is highly compatible with tabu search since it does not significantly perturb the tours of parents, and changes in edges are almost exclusively limited to the edges possessed by the parents. Tabu search focuses on the change of a single search point. When generating neighborhood solutions, it refers to a tabu list that records changes over a certain period in the past called tabu tenure in order to prohibit the repetition of the same change. To incorporate the concept of tabu search into GA using the multiset approach in this letter, past changes for each individual are archived in the tabu list.

Edge changes in individuals in EAX are primarily caused by the acquisition and loss of edges due to the application of AB-cycles to the tours. Therefore, by archiving the set of AB-cycles, i.e., \( E\set \), applied in the past for each individual, EAX can refer to the past \( E\set \) as a tabu list. The flow of the proposed EAX-tabu is as follows, where the archives of parents \( p_A \) and \( p_B \) are denoted as \( E^A_{\text{pre}} \) and \( E^B_{\text{pre}} \). The archive of each individual in the initial population is initialized with \( \emptyset \). Since each offspring obtains more edges from one of its parents, it inherits the archive of that parent.
Step 1-2 Same processes as Step 1-2 of EAX (section 3.1)

Step 3 $E_{pre} \leftarrow E^A_{pre} \cup E^B_{pre}$.

Step 4 Add edges included in $E_{pre}$ as tabu edges to the tabu list $E_T$ according to some criterion.

Step 5 Remove $AB$-cycles whose edges are contained in $E_T$ from $D$.

Step 6-8 Same processes as Step 3-5 of EAX

Step 9 $E^A_{pre} \leftarrow E^A_{pre} \cup E_{set}$, $E^B_{pre} \leftarrow E^B_{pre} \cup E_{set}$, and set $E^A_{pre}$ and $E^B_{pre}$ as the archives of $c_A$ and $c_B$, respectively.

Step 10 Delete $AB$-cycles that have expired after a given retention period (i.e., tabu tenure) from $E_{pre}$ and $E^B_{pre}$.

In Step 5, all edges contained in $E_{pre}$ are selected as tabu edges with a probability of 0.5 to constitute a tabu list $E_T$. Since $E_{pre}$ keeps each $AB$-cycle as a graph, some edges could be redundantly archived in the tabu list. This means that edges that exist in the archive more frequently, i.e., edges that have been exchanged more frequently in the past, are more likely to be selected as tabu edges.

5. Numerical Experiments

We evaluated the search performance of EAX-tabu by comparing it with that of EAX-UNIFORM. In EAX-tabu, after removing $AB$-cycles from $D$ according to the tabu list $E_T$, $E_{set}$ is constructed in the same way as EAX-UNIFORM. In the experiments, we used 10 medium-scale instances from TSPLIB and VLSI TSP, respectively. For both EAXs, the generation alternation model in the GA was the same as that used in [4], [6], where the population size was set to 300 and the number of offspring generated by one pair of parents in the crossover was set to 200. In the case where the best solution in the population was not updated for 30 generations, the search was terminated as having achieved convergence.

5.1 Performance of EAX-tabu

Table 1 shows the number of trials that produced the optimal solution and the average number of generations required for convergence in EAX-tabu and EAX-UNIFORM (“EAX” in the table) for each of the 10 instances. These results are from 30 trials. For EAX-tabu, a tabu tenure of 5 was employed across the experiment.

From Table 1, we observe that the proposed method finds optimal solutions with a high probability compared to EAX-UNIFORM in most instances. In addition, the proposed method increases the number of generations required for the convergence of the search, which means the ability of the population to generate new solutions is maintained longer by the tabu scheme.

Among the instances, no improvement in performance was observed in fnl4461 (TSPLIB). Figure 1 shows the transition of the number of $AB$-cycles generated in one pair of parents in solving fnl4461 by EAX-tabu. For comparison, it shows the transition in bgb4335 (VLSI TSP), which has the same scale with respect to the number of cities as fnl4461 and

In general, it is known that restricting the number of $AB$-cycles applied to offspring improves search performance in EAX [6]. To demonstrate the superiority of the proposed method over a straightforward approach that just limits $AB$-cycles selection, we conducted a comparative analysis between EAX-tabu and EAX reducing the selection probability of $AB$-cycles named EAX-limit. In the 20 instances used in the experiments, we observed that $AB$-cycles are removed in each pair of parents in EAX-tabu at an average rate of 0.3 to 0.5 (TSPLIB) and 0.3 to 0.4 (VLSI TSP) throughout all generations. Moreover, they have been removed at a maximum rate of 0.6 to 0.7. From these trends, EAX-limit randomly selected $AB$-cycles in the proportions of 0.3, 0.5, and 0.7 to the entire set and then some of those are randomly incorporated into $E_{set}$ as in EAX-UNIFORM.

<table>
<thead>
<tr>
<th>Instance</th>
<th>TSPLIB</th>
<th>VLSI TSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rat575</td>
<td>22 (49.8)</td>
<td>27 (71.0)</td>
</tr>
<tr>
<td>Vm1084</td>
<td>25 (48.2)</td>
<td>28 (65.6)</td>
</tr>
<tr>
<td>Di2190</td>
<td>23 (44.7)</td>
<td>28 (61.7)</td>
</tr>
<tr>
<td>Ul1432</td>
<td>14 (52.9)</td>
<td>17 (109.9)</td>
</tr>
<tr>
<td>Vm1748</td>
<td>7 (56.0)</td>
<td>29 (91.0)</td>
</tr>
<tr>
<td>Ul2152</td>
<td>7 (56.0)</td>
<td>29 (110.7)</td>
</tr>
<tr>
<td>Pr2392</td>
<td>22 (61.5)</td>
<td>29 (120.7)</td>
</tr>
<tr>
<td>Pch3038</td>
<td>0 (73.8)</td>
<td>12 (159.9)</td>
</tr>
<tr>
<td>Fln4461</td>
<td>0 (102.0)</td>
<td>0 (212.1)</td>
</tr>
<tr>
<td>Rl5914</td>
<td>2 (66.0)</td>
<td>4 (136.9)</td>
</tr>
</tbody>
</table>

Table 1 Comparisons of the number of trials producing the optimum and the number of generations for convergence

![Fig. 1](image-url) Transition of the number of $AB$-cycles

where the tabu scheme worked well. In the figure, the dashed line represents the average number of all the $AB$-cycles generated by the $AB$-cycle decomposition process, and the solid line represents the average number of $AB$-cycles after being removed by the tabu list. Both are the results of a typical trial. Figure 1 shows that although the total number of generated $AB$-cycles decreases with convergence in bgb4335, the number of $AB$-cycles is kept high even after removing some $AB$-cycles that contain tabu edges. In contrast, in fnl4461, the number of $AB$-cycles decreases as the search progresses, and is finally close to 0. Thus, in such a case, the tabu scheme does not work well.

5.2 Effectiveness of Tabu

In general, it is known that restricting the number of $AB$-cycles applied to offspring improves search performance in EAX [6]. To demonstrate the superiority of the proposed method over a straightforward approach that just limits $AB$-cycles selection, we conducted a comparative analysis between EAX-tabu and EAX reducing the selection probability of $AB$-cycles named EAX-limit. In the 20 instances used in the experiments, we observed that $AB$-cycles are removed in each pair of parents in EAX-tabu at an average rate of 0.3 to 0.5 (TSPLIB) and 0.3 to 0.4 (VLSI TSP) throughout all generations. Moreover, they have been removed at a maximum rate of 0.6 to 0.7. From these trends, EAX-limit randomly selected $AB$-cycles in the proportions of 0.3, 0.5, and 0.7 to the entire set and then some of those are randomly incorporated into $E_{set}$ as in EAX-UNIFORM.
Table 2 shows the results of comparing the number of trials that produced the optimal solution. These results are from 30 trials. From the results of EAX-limit in Table 2 and those of EAX-UNIFORM (“1.0” under EAX-limit) shown in Table 1, we can confirm that restricting the number of AB-cycles improves the performance of EAX in some instances. For vm1748, u2152, and pcb3038 in TSPLIB and most instances of VLSI TSP, the simple suppression of AB-cycles did not show a significant improvement in the search performance, and EAX-tabu showed superior performance to EAX-limit. These instances have city layouts of a lattice pattern or a straight line, which have a strong edge dependence among neighborhood cities, and the diversity of exchange edges has a significant impact on the performance of the crossover. In such instances, the tabu scheme is deemed effective, as it restricts the exchange edges while preserving diversity across successive generations.

5.3 Influence of Tabu Tenure

Here, we examine the effect of tabu tenure on search performance. Figure 2 shows the proportion of trials that found optimal solutions in 8 instances of TSPLIB and VLSI TSP. These results are from 30 trials; tabu tenure was set to 1, 3, 5, 7, and 10. The other search conditions are the same as in the previous sections. Note that tenure=0 corresponds to EAX-UNIFORM. For comparison, we refer to the results of the most recent EAX that uses edge entropy-based survival selection for improving the diversity of trait inheritance (labeled with EAX-ent). The results of EAX-ent in instances of TSPLIB are from [6]; we identify these instances by attaching an asterisk “*” to their labels.

Figure 2 illustrates that augmenting the tabu tenure value in EAX-tabu can enhance its search performance, albeit with variation depending on the instance. The performance enhancement diminishes with larger tabu tenure values in VLSI TSP instances where diversity of exchange edges is required. This is attributed to a reduction in available AB-cycles within the crossover due to the expansion of the tabu list. It is also worth noting the exceptional performance of EAX-ent for the fnl4461 instance, which indicates that, in addition to the crossover, maintaining a diversity of trait inheritance in the selection is important.

6. Conclusion

In this study, we proposed EAX-tabu that introduces a tabu search scheme into EAX. Through numerical experiments conducted on medium-scale instances of TSPLIB and VLSI TSP, we showed that the tabu scheme can significantly improve the performance of EAX, regardless of the tabu tenure setting. Further preliminary experiments indicated that the tabu scheme also improves the performance of EAX-RAND and EAX-5AB which is a part of the k-multiplex strategy. Moreover, Nagata’s edge entropy-based survival selection [6], emphasizing trait inheritance diversity, has proved highly effective and can be seamlessly integrated with the tabu scheme proposed in our study. However, the evaluation of EAX-tabu’s search performance on larger instances and its compatibility with the edge entropy-based survival selection method are left to future work.

References

[9] TSPLIB, http://comopt.ifi.uni-heidelberg.de/software/ TSPLIB95/ (Last accessed on 2024/03/14)