A Simple Genetic Algorithm using Sequential Constructive Crossover for the Quadratic Assignment Problem

Z H Ahmed
Department of Computer Science, Al Imam Mohammad Ibn Saud Islamic University (IMSIU), P.O. Box No. 5701, Riyadh 11432, Kingdom of Saudi Arabia

Received 30 August 2012; revised 20 December 2013; accepted 30 June 2014

Since crossover operator plays a vital role in genetic algorithms (GAs), several crossover operators have been proposed for the travelling salesman problem, which are then modified for the quadratic assignment problem (QAP). In this paper, we modify the sequential constructive crossover (SCX) operator for a simple GA to find heuristic solution to the QAP. Efficiency of the proposed GA using SCX is tested on some benchmark QAPLIB instances and then compared with GAs using other existing crossover operators.

Keywords: Quadratic assignment problem, NP-hard, Genetic algorithm, Sequential constructive crossover.

Introduction
The quadratic assignment problem (QAP) is a NP-hard1 combinatorial optimization problem that has many real-life applications. It was first introduced by Koopmans and Beckmann2 that can be defined as follow: There are a set of \( n \) facilities and a set of \( n \) locations. For each pair of facilities, a weight or flow matrix, \( F=[f_{ij}] \), is specified and for each pair of locations, a distance matrix, \( D=[d_{ij}] \), is specified. Also, let \( a = \{a(1),a(2),...,a(n)\} \) be an assignment, where \( a(i) \) represents the location of the facility \( i \). The problem is to assign to each location exactly one facility so as to minimize the cost (objective function)

\[
Z_a = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{a(i)a(j)} \quad \text{(1)}
\]

Several exact and heuristic algorithms have been proposed for solving the QAP. Exact algorithms for the QAP include branch-and-bound3, branch-and-cut4, and lexisearch5, etc. Heuristic methods for the QAP are constructive methods6, enumerative methods7, improvement methods8, greedy algorithms9, tabu search10, genetic algorithms11, simulated annealing12, and ant colony optimization13, etc. Amongst the heuristics genetic algorithms (GAs) are found to be the best algorithms. Genetic algorithms (GAs) are based essentially on imitating the process of evolution14. In GAs, there are (possibly) three genetic operators – selection, crossover and mutation. Among the operators, crossover plays a vital role, and hence, several crossover operators have been proposed for solving different optimization problems. Most of the crossover operators that are used for the QAP are the modification of the crossover operators proposed for the travelling salesman problem (TSP). Goldberg and Lingle15 proposed partially mapped crossover (PMX) for the TSP that has been modified by Migkikh et al16 for the QAP. The ordered crossover (OX) operator is developed by Davis17 for the TSP that has been applied to the QAP by Vázquez and Whitley18. One point crossover14 (OPX) operators are classical methods which were widely used in early versions of GAs. Lim et al19 proposed a variation of the OPX for the QAP. The cycle crossover20 (CX) operator was proposed for the TSP, which was then applied for the QAP21. The uniform like crossover22 (ULX) was slightly improved by Misevičius23 which was then renamed as optimized crossover (OX) and used in the improved hybrid GA for the QAP. Misevičius and Kilda24 proposed many modification of the ULX and called them as randomized ULX (RULX), modified ULX (or block crossover (BX)), and extended ULX (or repair crossover (RX)). Ahuja et al25 developed a swap path crossover (SPX) for the QAP. Drezner26 proposed another crossover operator, named cohesive crossover (COHX), for the QAP. Misevičius and Rubliauskas26 proposed a multiple parent crossover (MPX) operator that creates an offspring by choosing

*Author for Correspondence
E-mail: zhahmed@gmail.com
information from multiple number of parents. Misevičius and Kilda\textsuperscript{24} made a comparative study among 10 different crossover operators for the QAP and found that MPX is better than SPX which is better than OPX which is then better than COHX. However, in this study, we do not consider multi-parent crossover operator, rather stick to traditional two-parent crossover operators. In this paper, we modify the sequential constructive crossover\textsuperscript{27} (SCX) operator that was successfully proposed for the TSP, and then a GA based on SCX is developed for solving the QAP.

The efficiency of the GA using SCX is then compared with GAs using SPX and OPX for some benchmark QAPLIB\textsuperscript{28} instances. Experimental results show that SCX operator is better than SPX which is then better than OPX.

**Proposed genetic algorithm**

**Fitness function and selection operator**

The very first step (or step) of GAs is to encode solutions as feasible chromosomes (or individuals) such that the crossovers of the chromosomes result in feasible chromosomes. A simple GA starts by randomly generated set (initial population) of chromosomes and then applies genetic operators to create new, and hopefully, better populations as successive generations. For the QAP, solutions are typically represented by permutation of natural numbers of length equal to the total number of locations (or facilities), \( n \), where each chromosome position (gene) can take any number in \( \{1, 2, 3, \ldots, n\} \). The \( j \)th number in the permutation denotes the location of the facility \( j \). For a 7-location instance, \( (5, 7, 3, 2, 1, 4, 6) \) may be a chromosome, where 1\(^{st} \) facility is located at 5\(^{th} \) site, 2\(^{nd} \) facility is at 7\(^{th} \) site, and so on. Accordingly, we randomly generate an initial population of chromosomes of prefixed population size. The ‘fitness function’ is defined as multiplicative inverse of the objective function. As regards the selection method; we have considered the stochastic remainder selection method\textsuperscript{29} to create a mating pool for our simple GA.

**Sequential constructive crossover operator**

The GA search of the solution space is done by creating new chromosomes from old ones. As mentioned above, crossover is the most important operator in GAs. In crossover operator, a pair of chromosomes is selected randomly from the mating pool. Then a point, called crossover site, along their common length is randomly selected, and the information after the crossover site of the two parent chromosomes are swapped, thus creating two new offspring. However, this basic crossover method does not support for the TSP as well as the QAP. Hence, various crossover operators have been proposed for the TSP as well as the QAP. The sequential constructive crossover\textsuperscript{27} (SCX) operator is successfully applied to the TSP as well as the TSP with some variations\textsuperscript{30-32}. We are going to modify the SCX (we do not change its name) for the QAP as follows.

**Step 1:** - Start from the location (suppose, \( p \)) of the first facility of any randomly chosen parent, and go to Step 2.

**Step 2:** - Sequentially search both of the parent chromosomes and consider the first ‘legitimate location’ (the location that is not yet assigned) appeared after 'location \( p \)' in each parent. If no any ‘legitimate location’, after ‘location \( p \)’, is available in any of the parent, search sequentially from the beginning of the parent and consider the first ‘legitimate location’, and go to Step 3.

**Step 3:** Suppose 'location \( \alpha \)' and 'location \( \beta \)' are found in 1\(^{st} \) and 2\(^{nd} \) parent respectively, then for selecting the next location go to Step 4.

**Step 4:** Compute the cost of one incomplete offspring chromosome by incorporating 'location \( \alpha \)' as the next location (suppose, \( c_\alpha \)). Similarly, compute the cost of other incomplete offspring by incorporating 'location \( \beta \)' as the next location (suppose, \( c_\beta \)). Then go to Step 5. Suppose \( (\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_k, i) \) be a partially constructed offspring and ‘location \( \delta \)’ is selected for concatenation, then the cost (value) of assigning this location for the facility \( k \) is calculated as follows:

\[
C_\delta = \sum_{i=1}^{k-1} (f_{ik} d_{\alpha_i \delta} + f_{ik} d_{\delta \alpha_i}) \quad \ldots \quad (2)
\]

**Step 5:** If \( c_\delta \leq c_\delta \), then select 'location \( \alpha \)', otherwise, ‘location \( \beta \)’ as the next location to be assigned for facility \( k \) and concatenate it to the partially constructed offspring chromosome. If the offspring is a complete chromosome, go to Step 6; otherwise, rename the present location as ‘location \( p \)’ and go to Step 2.

**Step 6:** Evaluate the first parent and the offspring. If value of the offspring is less than the value of the parent, replace the parent by the offspring, otherwise skip the offspring.
**Mutation operator**

Mutation is a process of increasing diversity in the population by introducing random variations in the population. It randomly selects a position in a chromosome and changes the corresponding gene, thereby modifying the information. For the QAP, the classical mutation operator does not work. For this investigation, we have considered the reciprocal exchange mutation\(^{14}\) that selects two locations (genes) randomly and swaps them. The probability of applying mutation is set very low, whereas the probability of crossover is set very high. A simple GA repeatedly applies these (three) operators until either the population converges or until reaches the maximum number of generations (iterations).

**Computational experiments**

As an experimental basis for the crossover operators, simple GAs using three different crossover operators, namely, OPX, SPX and SCX, have been encoded in Visual C++ and run on a PC with Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz and 8.00GB RAM under MS Windows 7. In the experiments, we used some benchmark QAPLIB\(^{28}\) instances. The experiments were performed 20 times for each instance. The solution quality is measured by the percentage of excess of average solution value (ASV) of 20 runs over the best known solution value (BSV) reported in QAPLIB, as given by the formula

\[
\text{Excess}(\%) = 100 \times \frac{\text{ASV} - \text{BSV}}{\text{BSV}} \quad \ldots(3)
\]

We also report average CPU time of convergence (in seconds) by the algorithms. GAs are controlled by some parameters, namely, population size, crossover probability, mutation probability and termination criterion. To have a clear picture of the crossover operators, we set 1.0 (i.e., 100%) as crossover probability. Also after many testing, we set 0.08 (i.e., 8%) as mutation probability, 75 as population size and approximately same CPU time as a termination criterion. Table 1 shows comparative study of the crossover operators for eighteen QAPLIB instances of size from 20 to 100. For the instance tai35a, OPX is found to be the best; whereas SPX is the best for tai80a, tai80b and tai100a; and SCX is the best for remaining fourteen instances. On average, SCX is found to be the best one, and SPX is better than OPX.

**Conclusion and future work**

Crossover operator is the most important process in genetic algorithms (GAs). Several crossover operators that were proposed for the travelling salesman problem (TSP) have been modified to solve quadratic assignment problem (QAP). We have modified sequential constructive crossover (SCX) for our simple GA to obtain heuristic solution to the QAP. However, the aim and objective of this study is to investigate the efficiency of the crossover operators, not to find the optimal solution to the QAP. That is why; we do not use any local search technique to improve the solution quality. We set highest crossover probability to show the exact nature of crossover operators. Mutation operator is applied just not to get stuck in local minima quickly. We presented a comparative study among one point crossover (OPX), swap path crossover (SPX) and our SCX for some benchmark QAPLIB instances. In terms of solution quality, for the examined QAPLIB instances, SCX is found to be the best one, and SPX is found to be better than OPX. However, SCX does not obtain exact optimal solution to the examined instances. So an incorporation of good local search\(^{33}\) technique to the simple GA using SCX may obtain exact optimal solution to the instances, which is under our investigation. Also, some literatures\(^{24,34}\) found that the

\[
\text{Table 1—Comparison of the crossover operators for some benchmark QAPLIB instances}
\]

<table>
<thead>
<tr>
<th>Instance</th>
<th>BSV (1185996137)</th>
<th>Excess (%)</th>
<th>Avg. CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>tai20a</td>
<td>703482</td>
<td>6.57</td>
<td>6.50</td>
</tr>
<tr>
<td>tai20b</td>
<td>122455319</td>
<td>8.73</td>
<td>6.56</td>
</tr>
<tr>
<td>tai25a</td>
<td>1167256</td>
<td>5.01</td>
<td>4.58</td>
</tr>
<tr>
<td>tai25b</td>
<td>344355646</td>
<td>12.92</td>
<td>5.24</td>
</tr>
<tr>
<td>tai30a</td>
<td>1818146</td>
<td>5.16</td>
<td>4.29</td>
</tr>
<tr>
<td>tai30b</td>
<td>637117113</td>
<td>13.46</td>
<td>8.78</td>
</tr>
<tr>
<td>tai35a</td>
<td>2422002</td>
<td>4.92</td>
<td>4.95</td>
</tr>
<tr>
<td>tai35b</td>
<td>283315445</td>
<td>7.49</td>
<td>5.00</td>
</tr>
<tr>
<td>tai40a</td>
<td>3139370</td>
<td>4.97</td>
<td>4.53</td>
</tr>
<tr>
<td>tai40b</td>
<td>637250948</td>
<td>9.44</td>
<td>7.30</td>
</tr>
<tr>
<td>tai50a</td>
<td>4938796</td>
<td>5.00</td>
<td>4.51</td>
</tr>
<tr>
<td>tai50b</td>
<td>458821517</td>
<td>7.46</td>
<td>5.60</td>
</tr>
<tr>
<td>tai60a</td>
<td>7205962</td>
<td>4.89</td>
<td>4.60</td>
</tr>
<tr>
<td>tai60b</td>
<td>608215054</td>
<td>6.95</td>
<td>5.12</td>
</tr>
<tr>
<td>tai80a</td>
<td>13499184</td>
<td>4.32</td>
<td>3.83</td>
</tr>
<tr>
<td>tai80b</td>
<td>818415043</td>
<td>6.15</td>
<td>6.63</td>
</tr>
<tr>
<td>tai100a</td>
<td>21052466</td>
<td>4.04</td>
<td>4.02</td>
</tr>
<tr>
<td>tai100b</td>
<td>1185996137</td>
<td>9.33</td>
<td>5.08</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>7.04</td>
<td>6.89</td>
</tr>
</tbody>
</table>

AHMED: A SIMPLE GENETIC ALGORITHM USING SEQUENTIAL CROSSOVER

765
multiple-parent crossover is better than the traditional two-parent crossover; hence we are also planning to develop a multiple-parent SCX for the QAP.

Acknowledgements

This research was supported by the NSTIP strategic technologies program number (10) in the Kingdom of Saudi Arabia vide Award No. 11-INF1788-08. The author is very much thankful to the NSTIP for its financial and technical supports.

Reference