Journal of Scientific & Industrial Research
Vol. 73, December 2014, pp. 763-766

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-hard[1] combinatorial optimization problem that has many real-life applications. It was first introduced by Koopmans and Beckmann[2] 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, $D=[d_{kl}]$, is specified. Also, let $a = \{a(1), a(2), \ldots\ldots, a(n)\}$ 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)

$$Z_a = \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij} d_{a(i)a(j)} \qquad \ldots(1)$$

Several exact and heuristic algorithms have been proposed for solving the QAP. Exact algorithms for the QAP include branch-and-bound[3], branch-and-cut[4], and lexisearch[5], etc. Heuristic methods for the QAP are constructive methods[6], enumerative methods[7], improvement methods[8], greedy algorithms[9], tabu search[10], genetic algorithms[11], simulated annealing[12], and ant colony optimization[13], 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 evolution[14]. 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 Lingle[15] proposed partially mapped crossover (PMX) for the TSP that has been modified by Migkikh *et al*[16] for the QAP. The ordered crossover (OX) operator is developed by Davis[17] for the TSP that has been applied to the QAP by Vázquez and Whitley[18]. One point crossover[14] (OPX) operators are classical methods which were widely used in early versions of GAs. Lim *et al*[19] proposed a variation of the OPX for the QAP. The cycle crossover[20] (CX) operator was proposed for the TSP, which was then applied for the QAP[21]. The uniform like crossover[22] (ULX) was slightly improved by Misevicius[23] which was then renamed as optimized crossover (OX) and used in the improved hybrid GA for the QAP. Misevičius and Kilda[24] 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 al*[11] developed a swap path crossover (SPX) for the QAP. Drezner[25] proposed another crossover operator, named cohesive crossover (COHX), for the QAP. Misevičius and Rubliauskas[26] 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[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[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[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 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, ...,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[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[27] (SCX) operator is successfully applied to the TSP as well as the TSP with some variations[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 α' and 'location β' 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 α' as the next location (suppose, c_α). Similarly, compute the cost of other incomplete offspring by incorporating 'location β' as the next location (suppose, c_β). Then go to Step 5. Suppose $(\alpha_1,\alpha_2,\alpha_3,\ldots,\alpha_{k-1})$ be a partially constructed offspring and 'location δ' is selected for concatenation, then the cost (value) of assigning this location for the facility k is calculated as follows:

$$C_\delta = \sum_{i=1}^{k-1}(f_{ik}d_{\alpha i\delta} + f_{ki}d_{\delta\alpha i}) \quad \ldots(2)$$

Step 5: If $c_\alpha \leq c_\beta$, then select 'location α', otherwise, 'location β' 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 modifies 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

$$Excess(\%) = 100 \times (ASV - BSV)/BSV \quad \ldots(3)$$

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 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

Table 1—Comparison of the crossover operators for some benchmark QAPLIB instances

Instance	BSV	Excess (%)			Avg. CPU Time
		OPX	SPX	SCX	
tai20a	703482	6.57	5.59	4.50	6
tai20b	122455319	8.73	9.12	6.56	6
tai25a	1167256	5.01	5.61	4.58	9
tai25b	344355646	12.92	15.42	5.24	9
tai30a	1818146	5.16	5.49	4.29	13
tai30b	637117113	13.4	10.55	8.78	13
tai35a	2422002	4.92	5.35	4.95	18
tai35b	283315445	7.49	6.76	5.00	18
tai40a	3139370	4.97	5.76	4.53	24
tai40b	637250948	9.44	11.57	7.30	24
tai50a	4938796	5.00	4.91	4.51	37
tai50b	458821517	7.46	6.68	5.60	37
tai60a	7205962	4.89	4.60	4.54	54
tai60b	608215054	6.95	8.14	5.12	54
tai80a	13499184	4.32	3.83	4.35	95
tai80b	818415043	6.15	6.07	6.63	95
tai100a	21052466	4.04	3.22	4.02	144
tai100b	1185996137	9.33	5.39	5.08	144
Average		7.04	6.89	5.31	

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

1 Sahni S & Gonzales T, P-complete approximation problems, *J Assoc Comput Mach*, **23** (1976) 555–565.

2 Koopmans TC & Beckmann M J, Assignment problems and the location of economic activities, *Econometrica,* **25** (1957) 53–76.

3 Hahn PM, Hightower W L, Johnson TA, Guignard-Spielberg M & Roucairol C, Tree elaboration strategies in branch and bound algorithms for solving the quadratic assignment problem, *Yugoslavian J Oper Res,* **11** (2001) 41–60.

4 Erdoğan G & Tansel B, A branch-and-cut algorithm for the quadratic assignment problems based on linearizations, *Comput Oper Res*, **34** (2007)1085-1106.

5 Ahmed ZH, A new reformulation and an exact algorithm for the quadratic assignment problem, *Indian J Sci Technol*, **6** (2013) 4368-4377.

6 Gilmore PC, Optimal and suboptimal algorithms for the quadratic assignment problem, *SIAM J Appl Math*, **10** (1962) 305–313.

7 Burkard RE & Bonniger T, A heuristic for quadratic Boolean programs with applications to quadratic assignment problems, *Eur J Oper Res*, **13** (1983) 374–386.

8 Mills P, Tsang E & Ford J, Applying an extended guided local search to the quadratic assignment problem, *Ann Oper Res*, **118** (2003) 121–135.

9 Oliveira CAS, Pardalos MP & Resende MGG, GRASP with path relinking for the quadratic assignment problem, *Lect Note Comp Sci,* **3059** (2004) 356–368.

10 Paul G, An efficient implementation of the robust Tabu search heuristic for sparse quadratic assignment problems, *Eur J Oper Res,* **209** (2011) 215–218.

11 Ahuja R, Orlin JB & Tiwari A, A greedy genetic algorithm for the quadratic assignment problem, *Comput Oper Res*, **27** (2000) 917–934.

12 Wilhelm MR & Ward TL, Solving quadratic assignment problems by simulated annealing, *IEEE Trans*, **19** (1987) 107–119.

13 Acan A, An external partial permutations memory for ant colony optimization, *Lect Notes Comput Sci,* **3448** (2005) 1–11.

14 Goldberg DE, *Genetic Algorithms in Search, Optimization, and Machine Learning,* (Addison-Wesley, New York) 1989.

15 Goldberg DE & Lingle R, Alleles, Loci and the Travelling Salesman Problem, *Proc 1st IntConf Genet Algor Appl* (Lawrence Erlbaum Associates, Hilladale, NJ) 1985, 154-159.

16 Migkikh VV, Topchy AA, Kureichik VM & Tetelbaum AY, Combined genetic and local search algorithm for the quadratic assignment problem, *Proc 1st Int Conf Evolut Comput Appl* (Presidium of the Russian Academy of Sciences, Moscow) 1996, 335−341.

17 Davis L, Job-shop Scheduling with Genetic Algorithms, *Proc Int Conf Genet Algor Appl*, 1985, 136-140.

18 Vázquez M & Whitley LD, A hybrid genetic algorithm for the quadratic assignment problem, *Proc Genet Evolut Comput Conf* (Morgan Kaufmann, San Francisco) 2000, 135–142.

19 Lim MH, Yuan Y & Omatu S, Efficient genetic algorithms using simple genes exchange local search policy for the quadratic assignment problem, *Comput Optim Appl*, **15** (2000) 249−268.

20 Oliver IM, Smith DJ & Holland JRC, A Study of Permutation Crossover Operators on the Travelling Salesman Problem, *Proc 2nd Int Conf Genet Algor,* (Lawrence Erlbaum Associates, Hilladale, NJ) 1987, 224-230.

21 Merz P & Freisleben B, Fitness landscape analysis and memetic algorithms for the quadratic assignment problem, *IEEE T Evolut Comput*, **4** (2000) 337−352.

22 Tate DE & Smith AE, A genetic approach to the quadratic assignment problem, *Comput Oper Res,* **22** (1995) 73–83.

23 Misevicius A, An improved hybrid optimization algorithm for the quadratic assignment problem, *Math Model Anal,* **9** (2004) 149–168.

24 Misevicius A & Kilda B, Comparison of crossover operators for the quadratic assignment problem, *Inf Technol Control*, **34** (2005) 109–119.

25 Drezner Z, A new genetic algorithm for the quadratic assignment problem, *Informs J Comput*, **15** (2003) 320–330.

26 Misevicius A & Rubliauskas D, Performance of hybrid genetic algorithm for the grey pattern problem, *Inf Technol Control*, **34** (2005) 15–24.

27 Ahmed ZH, Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator, *Int J Biomet Bioinf*, **3** (2010) 96-105.

28 Burkard RE, Cela E, Karisch SE & Rendl F, QAPLIB - a quadratic assignment problem library, *J Global Optim*, **10** (1997) 391-403. See also at http://www.seas.upenn.edu/qaplib/.

29 Deb K, *Optimization for Engineering Design: Algorithms and Examples*, (Prentice Hall of India Pvt Ltd, New Delhi, India) 1995.

30 Ahmed ZH, A hybrid genetic algorithm for the bottleneck traveling salesman problem, *ACM T Embed Comput S*, **12** (2013) Art. No. 9.

31 Ahmed Z H, An experimental study of a hybrid genetic algorithm for the maximum travelling salesman problem, *Math Sci,* **7** (2013) 1-7.

32 Ahmed ZH, The ordered clustered travelling salesman problem: A hybrid genetic algorithm, *Sci World J,* Art ID **258207** (2014) 13 pages. doi:10.1155/2014/258207.

33 Ahmed, ZH, Improved genetic algorithms for the traveling salesman problem, *Int J of Process Manage Bench*, **4** (2014) 109-124.

34 Ahmed ZH, Multi-parent extension of sequential constructive crossover for the traveling salesman problem, *Int J Oper Res,* **11** (2011) 331-342.