A Hybrid Genetic Algorithm Based on Complete Graph Representation for the Sequential Ordering Problem

Dong-Il Seo and Byung-Ro Moon

School of Computer Science & Engineering, Seoul National University
Sillim-dong, Kwanak-gu, Seoul, 151-742 Korea
{diseo, moon}@soar.snu.ac.kr
http://soar.snu.ac.kr/~{diseo, moon}/

Abstract. A hybrid genetic algorithm is proposed for the sequential ordering problem. It is known that the performance of a genetic algorithm depends on the survival environment and the reproducibility of building blocks. For decades, various chromosomal structures and crossover operators were proposed for the purpose. In this paper, we use Voronoi quantized crossover that adopts complete graph representation. It showed remarkable improvement in comparison with state-of-the-art genetic algorithms.

1 Introduction

Given n nodes, *sequential ordering problem* (SOP) is the problem of finding a Hamiltonian path of minimum cost satisfying given *precedence constraints*. Formally, given a set of nodes $V = \{1, 2, \ldots, n\}$ and cost matrix $C = (c_{ij})$, $c_{ij} \in \mathbb{N} \cup \{\infty\}$, $i, j \in V$, it is the problem of finding a Hamiltonian path π that satisfies precedence constraints and minimizes the following:

$$\text{Cost}(\pi) = \sum_{i=1}^{n-1} c_{\pi(i)\pi(i+1)}.$$

Here, the precedence constraints are marked by infinity (∞) in the cost matrix, i.e., if $c_{ji} = \infty$, node j cannot precede node i in the path. The relationship is denoted by $i \prec j$; node i is called a *predecessor* of node j and node j is called a *successor* of node i. It is assumed that the path starts at node 1 and ends at node n, i.e., $1 \prec i$ and $i \prec n$ for all $i \in V \setminus \{1, n\}$. Generally, the cost matrix C is asymmetric and the precedence constraints are transitive and acyclic. The problem is also called 'asymmetric Hamiltonian path problem with precedence constraints'. The special case of SOP with empty precedence constraints is reduced to asymmetric traveling salesman problem (ATSP). As ATSP is an *NP*-hard problem, so is SOP.

The problem arises in various practical fields such as manufacturing, routing, and scheduling. However, not very much attention has been paid to the

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2723, pp. 669–680, 2003.
© Springer-Verlag Berlin Heidelberg 2003

problem, while TSP, which is a reduction of SOP, has been one of the most popular problems in the combinatorial optimization area. Cutting-plane approach [1], Lagrangian relax-and-cut method [2], and branch-and-cut algorithm [3] are mathematical model-based approaches. The genetic algorithm using a crossover called maximum partial order/arbitrary insertion (MPO/AI) [4] and the hybrid ant colony system called HAS-SOP [5] are state-of-the-art metaheuristics for SOP. Path preserving 3-Opt (pp-3-Opt) algorithm and its variants such as SOP-3-exchange [5] are the most popular local improvement heuristics for hybrid metaheuristics.

In this paper, we propose a new genetic algorithm for SOP. We adopt Voronoi quantized crossover to exploit the topological linkages of genes in the genetic search. The crossover is based on complete graph representation.

The rest of this paper is organized as follows. We mention the background in Section 2 and describe the proposed genetic operators in Section 3. The experimental results are provided in Section 4. Finally, the conclusions are given in Section 5.

2 Background

The building block hypothesis implies that the power of a genetic algorithm lies in its ability to create and grow the building blocks efficiently. Building blocks appear in interactive gene groups. The interaction between genes means the dependence of a gene's contribution to the fitness upon the values of other genes. The interaction is also called *epistasis* in GA, although it is wider than the biological definition of epistasis [6,7,8].

A gene group is said to have *strong linkage* if the survival probability of the corresponding schema is higher than normal, and it is said to have *weak linkage* otherwise [6]. To make building blocks survive through recombinations, we must let the strongly epistatic gene groups have stronger linkage than ordinary gene groups [6,9]. The linkage of a gene group is affected by various factors. Particularly, the linkage determined by the relative positions of genes in the chromosome is called *topological linkage* [10]. In the case, each gene is placed in an Euclidean or non-Euclidean space, called *chromosomal space*, to represent the linkages between genes. In order to make the topological linkages reflect well the epistatic structure of a given problem, we need to choose an appropriate *chromosomal structure*. The chromosomal structure here means the conceptual structure of genes used for the crossover operator. A typical chromosomal structure is one-dimensional array. In general, multi-dimensional representations are more advantageous than simple one-dimensional representations for highly epistatic problems [10]. For example, two-dimensional array, two-dimensional real space (plane), and complete graph are available.

Recently, a large number of genetic algorithms that exploit the topological linkages of genes have been proposed. They are classified into three models: *static linkage model*, *adaptive linkage model*, and *evolvable linkage model* [10]. The linkages are fixed during the genetic process in the static linkage model.

```
1.  VQX(n, k, d_g, p_1, p_2)
2.  {
3.      I ← {1, 2, ..., n}; K ← {1, 2, ..., k};
4.      Select a subset R = {s_1, s_2, ..., s_k} ⊂ I at random;
5.      for each i ∈ I {
6.          r[i] ← arg min_{j∈K} {d_g(s_j, i)}, s_j ∈ R;
7.      }
8.      for each j ∈ K { u[j] ← 0 or 1 at random; }
9.      for each i ∈ I {
10.         if (u[r[i]] = 0 and u[r[p_1[i]]] = 0) then o[i] ← p_1[i];
11.         else if (u[r[i]] = 1 and u[r[p_2[i]]] = 1) then o[i] ← p_2[i];
12.         else o[i] ← nil;
13.     }
14.     o ← GreedyRepair(o);
15.     return o;
16. }
```

Fig. 1. Voronoi quantized crossover for SOP.

They adaptively changes in the adaptive linkage model, and evolve in parallel with the allele values in the evolvable linkage model. We adopt the Voronoi quantized crossover [11] and apply the static linkage model in this paper.

3 Genetic Operators

3.1 Voronoi Quantized Crossover

In Voronoi quantized crossover (VQX), a chromosome is a complete graph of genes where each edge weight, called *genic distance*, reflects the epistatic strength between the two corresponding genes. The graph is directed if the genic distance is asymmetric. In fact, the genes are assigned a position in a non-Euclidean space defined by the genic distances. By adopting such a non-Euclidean chromosomal space, we aim to reflect the epistases with minimal distortion in the crossover. The proposed heuristic for the genic distance assignment is described in Section 3.2. VQX was applied to the traveling salesman problem for the first time [11]. Applying VQX to SOP needs considerable modification. We describe the VQX for SOP in the following.

For the problem, we use the locus-based encoding[1] as in [12]; one gene is allocated for every node and the gene value represents the index of its next node in the path. VQX has a simple structure. Figure 1 shows the pseudo code

[1] The term *encoding* here must be distinguished from the term *representation* because we mean by encoding the actual scheme to store solutions not for crossover in this paper.

```
1.  GreedyRepair(o)
2.  {
3.      S ← Extract path segments from o;
4.      S ← PrecCycleDecomposition(S);
5.      s_0 ← the segment that contains node 1 in S;
6.      S ← S \ {s_0};
7.      do {
8.          s ← the nearest segment from s_0 among the segments, in S,
9.              all whose predecessors are already contained in the segment
10.             itself or in s_0;
11.         Attach s to s_0;  S ← S \ {s};
12.     } while (|S| > 0);
13.     o' ← the solution of the segment s_0;
14.     return o';
15. }
```

Fig. 2. Greedy repair.

of VQX where n is the number of genes and k is the crossover degree ranged from 2 to n. The function $d_g : I^2 \to R$ represents the genic distance. The two parents and the offspring are denoted by p_1, p_2, and o, respectively. Following the convention, the notation "arg min" takes the argument that minimizes the value. Given a number of vectors, the Voronoi region of a vector is defined to be the nearest neighborhood of the vector [13]. In VQX, the chromosomal space defined by d_g is quantized into k Voronoi regions determined by the k randomly selected genes (lines 4–7), then a sort of block-uniform crossover [14] is performed on the regions (lines 8–13). We use a random tie-breaking in the calculation of "arg min" in the crossover (line 6).

The part of gene inheritance (lines 8–13) goes as follows. At first, each region is masked white or gray at random. The white and gray correspond to 0 and 1, respectively, in line 8. Then the genes in the white regions are inherited from parent 1 and the others are inherited from parent 2 (lines 9–13). At this time, the gene values are not always copied but only when a gene (gene i) and the gene pointed by it (gene $p_1[i]$ or gene $p_2[i]$) belong to the same-colored region. That is, an arc in a parent has a chance to survive in the offspring when both end points belong to the same-colored region(s). The word *nil* is used for the genes whose values are not determined. As a result, a partial solution consisting of path segments is generated. We use a greedy approach to repair it. Figure 2 shows the pseudo code of the greedy repair. Beginning with the segment containing node 1 (lines 5–6), it repeatedly merge segments available (lines 7–12). An available segment is a segment all whose predecessors are contained in the segment itself or in the segments already merged.

Because the segments are inherited from the two parents, it may include precedence cycles. Therefore, a precedence cycle decomposition algorithm is re-

```
1.  PrecCycleDecomposition(S)
2.  {
3.      START:
4.      D ← ∅;  T ← ∅;
5.      do {
6.          Select a segment s from S \ D at random;
7.          D ← D ∪ {s}
8.          for each node i in s {
9.              for each predecessor i_p of i {
10.                 s_p ← the segment contains i_p in S;
11.                 if (s_p ≠ s and (s, s_p) ∉ T) {
12.                     if ((s_p, s) ∈ T) {
13.                         Split s into s' and s'';
14.                         S ← S \ {s} ∪ {s', s''};
15.                         goto START;
16.                     } else {
17.                         T ← T ∪ {(s, s_p)};
18.                         T ← TransitiveClosure(T);
19.                     }
20.                 }
21.             }
22.         }
23.     } while (|D| < |S|);
24.     return S;
25. }
```

Fig. 3. Precedence cycle decomposition algorithm.

quired before merging the segments (line 4 in Figure 2). Figure 3 shows the pseudo code of the algorithm. The algorithm inspects the precedence relationships between the segments and if it finds a precedence cycle, it decomposes the cycle by splitting a segment involved in the cycle into two sub-segments (lines 13–14). The splitting point is determined to be the position before the node i or the position after the node i in the figure. The position with more balanced sizes of the resulting segments is preferred. The splitting is repeated until no cycle is found (lines 3–23). TransitiveClosure() returns the transitive closure of a precedence relation T (line 18).

Figure 4 shows an example of VQX for SOP. In the figure, the nodes (genes) and the non-trivial precedence constraints are drawn by small circles and dashed arrows, respectively. For the convenience of illustration, we assumed the chromosomal space to be a two-dimensional Euclidean space. The assumption is merely for the visualization. At first, the chromosomal space is quantized into nine Voronoi regions as in (a). Then, the offspring inherits path segments from the parents. Figures 4(b)–(c) shows the two parents and Figure 4(d) shows the

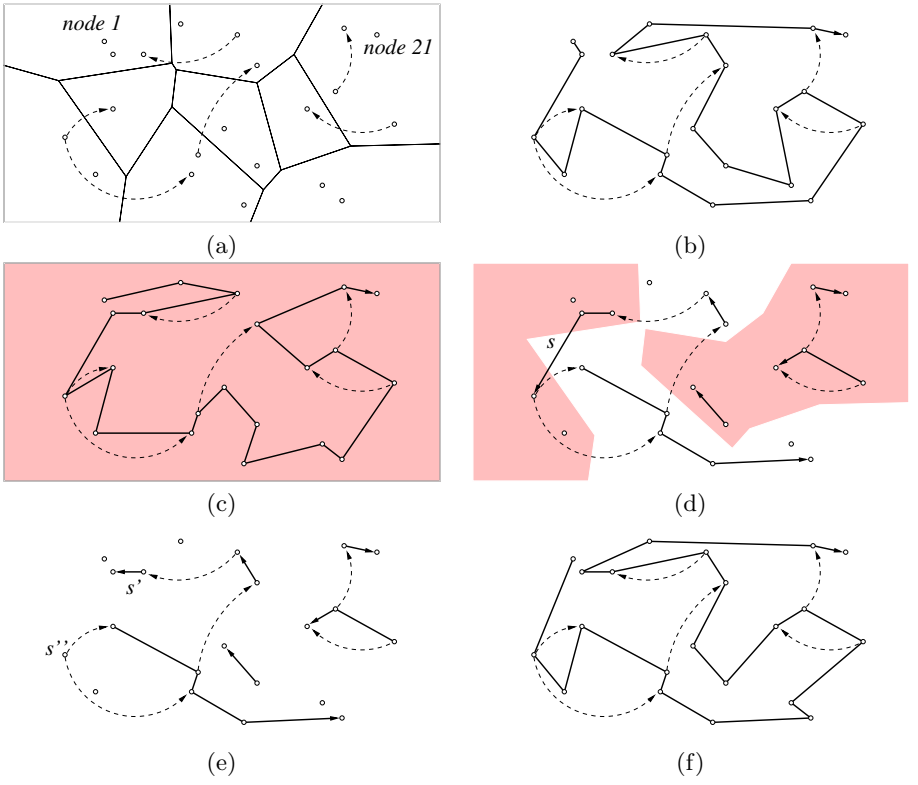

Fig. 4. An illustration of VQX for SOP. (a) A chromosomal space quantized into nine Voronoi regions. (b) Parent 1. (c) Parent 2. (d) Inherited path segments. (e) After precedence cycle decomposition. (f) Repaired path segments.

inherited path segments. By the precedence cycle decomposition, the segment s in (d) is split into segments s' and s'' in (e). Finally, an offspring is generated by the greedy repair as in (f).

3.2 Genic Distance Assignment

We apply the static linkage model to the genetic algorithm, i.e., the genic distances are assigned statically before running the genetic algorithm. Intuitively, an ideal value of a genic distance is a value inversely proportional to the strength of the epistasis. However, no practical method to get the exact values of the epistases is known yet. Therefore, we rely on heuristics. The genic distance from gene i to gene j is defined as

$$d_g(i,j) = |\{l \in V : c_{il} < c_{ij}\}| \tag{1}$$

where V is the set of nodes and c_{pq} is the (p, q) element of the cost matrix. It is based on the fact that the epistasis reflects the topological locality of the nodes. The genic distance is asymmetric as the cost matrix C is asymmetric.

3.3 Heterogeneous Mating

It is known that VQX shows faster convergence than other crossovers; this may cause the premature convergence of genetic algorithms. To avoid it, we use a special type of mating used in [11]. In the mating, each individual is mated with one of its dissimilar individuals. Hollstien called this type of breeding a *negative assortive mating* [15]. The *heterogeneous mating* is done similarly to a selection method called crowding [16]. First, given an individual p_1, m candidate individuals are selected from the population P by roulette-wheel selection. Among them, the most different one from p_1 is selected as p_2. Hamming distance[2] is used for the distance measure. The heterogeneous mating improved the performance of VQX by slowing down the convergence of the genetic algorithm. It is notable that we could not found any synergy effect between the mating and other crossovers such as k-point crossover and uniform crossover in our experiments.

3.4 Properties of VQX

VQX has two notable properties:

- *Convexity* — Voronoi regions are convex[3] (see [13] p. 330).
- *Diversity* — It has $\binom{n}{k}2^k$ crossover operators.

In VQX, genes in the chromosome are quantized into several groups by randomly selected Voronoi regions, and the gene values in the same group are inherited from the same parent. Therefore, the first property that Voronoi regions are convex implies that the gene groups of relatively short genic distance have high survival probabilities, i.e., strong linkages.

The other property means that VQX has a lot of crossover operators. The number of crossover operators affects the creativity of new schemata. The number of crossover operators of k-point crossover is $\binom{n-1}{k}$. For $n = 10000$ and $k = 12$, for example, VQX has about 10^{43} crossover operators, while k-point crossover has about 10^{39}. However, we should mention that we do not pursue the maximal number of crossover operators.

4 Experimental Results

The genetic algorithms used in this paper are steady-state hybrid genetic algorithms. Figure 5 shows the template. In the template, n is the problem size, m is the group size of heterogeneous mating, k is the crossover degree, and d_g is

[2] the number of different edges between two paths.

[3] A set $S \in R^k$ is *convex* if $a, b \in S$ implies that $\alpha a + (1 - \alpha)b \in S$ for all $0 < \alpha < 1$.

```
1.  VGA(n, m, k, d_g)
2.  {
3.      Initialize population P;
4.      repeat {
5.          p_1 ← Selection(P);
6.          p_2 ← MateSelection(P, m, p_1);
7.          o ← VQX(n, k, d_g, p_1, p_2);
8.          o ← Mutation(o);
9.          o ← LocalImprovement(o);
10.         P ← Replacement(P, p_1, p_2, o);
11.     } until (stopping condition);
12.     return the best of P;
13. }
```

Fig. 5. The steady-state hybrid genetic algorithm for SOP.

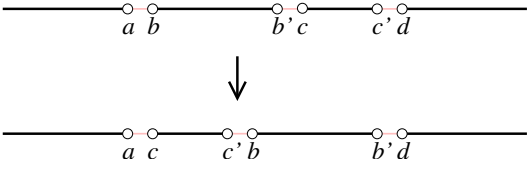

Fig. 6. An illustration of the path-preserving 3-exchange.

the genic distance. The two selected parents and the offspring are denoted by p_1, p_2, and o, respectively. The genetic operators and their parameters used in this paper are summarized in the following.

- Population Initialization — Initial solutions are generated at random, then the local improvement algorithm is applied to each of them. All the solutions in the population are feasible.
- Population Size — $|P| = 50$.
- Selection — Roulette-wheel selection, i.e., the fitness value f_i of the solution i is calculated as

$$f_i = (C_w - C_i) + (C_w - C_b)/4 \tag{2}$$

  where C_i, C_w, and C_b are the costs of the solution i, the worst solution, and the best solution in the population, respectively. The fitness value of the best solution is five times as great as that of the worst solution in the population.
- Group Size of Heterogeneous Mating — $m = 3$.
- Crossover Degree — $k = 6$.
- Mutation — Five random feasible-path-preserving 3-exchanges are applied to each offspring with probability 0.1. Figure 6 shows a symbolic drawing of the exchange.

Table 1. The experimental results for ESC78 and ft70.∗.

Graph (Bst-Kn)	GA	BK#/t	Best (%)	Avg (%)	$\sigma/\sqrt{t}$	Gen	Time (s)
ESC78 (18230)	DGA	1000/1000	18230 (0.000)	**18230.00** (0.000)	0.00	223	2.68
	MGA	1000/1000	18230 (0.000)	**18230.00** (0.000)	0.00	335	1.83
	VGA	1000/1000	18230 (0.000)	**18230.00** (0.000)	0.00	115	0.91
ft70.1 (39313)	DGA	953/1000	39313 (0.000)	39315.75 (0.007)	0.39	268	4.40
	MGA	548/1000	39313 (0.000)	39351.03 (0.097)	1.35	2256	8.48
	VGA	1000/1000	39313 (0.000)	**39313.00** (0.000)	0.00	629	6.27
ft70.2 (40419)	DGA	718/1000	40419 (0.000)	40421.26 (0.006)	0.59	710	7.66
	MGA	117/1000	40419 (0.000)	40424.45 (0.013)	0.68	601	3.48
	VGA	930/1000	40419 (0.000)	**40419.18** (0.000)	0.02	1190	7.66
ft70.3 (42535)	DGA	526/1000	42535 (0.000)	42549.87 (0.035)	0.50	205	2.41
	MGA	619/1000	42535 (0.000)	42546.86 (0.028)	0.48	177	1.45
	VGA	909/1000	42535 (0.000)	**42537.82** (0.007)	0.28	319	2.38
ft70.4 (53530)	DGA	405/1000	53530 (0.000)	53560.35 (0.057)	0.88	594	4.59
	MGA	12/1000	53530 (0.000)	53571.90 (0.078)	0.29	666	2.54
	VGA	618/1000	53530 (0.000)	**53543.97** (0.026)	0.58	559	3.83

– Local Improvement — A simple path-preserving 3-Opt (pp-3-Opt) algorithm is used. In the algorithm, a path-preserving 3-exchanges of maximum gain is selected and performed repeatedly. The gain of an exchange, with Figure 6 as an example, is computed by

$$gain = c_{ab} + c_{b'c} + c_{c'd} - c_{ac} - c_{c'b} - c_{b'd} \tag{3}$$

where c_{pq} is the (p, q) element of the cost matrix. For efficient feasibility checking, a marking technique is used as the SOP labeling procedure in [5].

– Replacement — A variant of preselection [17] is used as in [12]. Each offspring is replaced with (i) its more similar parent if the offspring is better, (ii) the other parent if the offspring is better, (iii) the worst solution in the population, otherwise.

– Stopping Condition — Until 70 percent of the population converges with the same cost as the best solution. This takes account of the cases that more than one best solution of the same quality competes with each other.

The algorithms were implemented in C on Pentium III 1132 MHz running Linux 2.2.14. We tested on eighteen SOP instances taken from [18]. They are all instances that have more than seventy nodes.

Tables 1–3 compare the performance of VGA with DGA and MGA. VGA represents the genetic algorithms using Voronoi quantized crossover (VQX) with the genic distance assignment heuristic described in Section 3.2. DGA and MGA represent the genetic algorithms using distance preserving crossover (DPX) and

Table 2. The experimental results for kro124p.∗ and prob.100.

Graph (Bst-Kn)	GA	BK#/t	Best (%)	Avg (%)	$\sigma/\sqrt{t}$	Gen	Time (s)
	DGA	357/1000	39420 (0.000)	39481.95 (0.157)	1.58	431	25.46
kro124p.1	MGA	565/1000	39420 (0.000)	39505.79 (0.218)	6.12	902	15.64
(39420)	VGA	930/1000	39420 (0.000)	**39426.45** (0.016)	0.95	518	12.92
	DGA	876/1000	41336 (0.000)	**41344.27** (0.020)	0.70	529	27.96
kro124p.2	MGA	543/1000	41336 (0.000)	41566.05 (0.557)	12.77	1079	14.91
(41336)	VGA	789/1000	41336 (0.000)	41353.22 (0.042)	1.76	688	12.49
	DGA	6/1000	49499 (0.000)	50035.24 (1.083)	9.16	3884	42.68
kro124p.3	MGA	78/1000	49499 (0.000)	50029.73 (1.072)	12.81	3051	17.05
(49449)	VGA	705/1000	49499 (0.000)	**49582.64** (0.169)	6.27	1146	12.71
	DGA	999/1000	76103 (0.000)	76103.27 (0.000)	0.27	227	11.75
kro124p.4	MGA	841/1000	76103 (0.000)	76138.68 (0.047)	2.61	298	7.00
(76103)	VGA	1000/1000	76103 (0.000)	**76103.00** (0.000)	0.00	249	8.36
	DGA	0/50	1197 (0.588)	1260.72 (5.943)	5.62	112869	5108
prob.100	MGA	1/50	1175 (−1.261)	**1244.36** (4.568)	4.28	2165330	54166
(1190)	VGA	2/50	1163 (−2.269)	1255.86 (5.534)	5.85	122586	1767

maximum partial order/arbitrary insertion (MPO/AI)[4] [4], respectively. DPX tries to generate an offspring that has equal Hamming distance to both of its parents, i.e., its aim is to achieve that the three Hamming distances between offspring and parent 1, offspring and parent 2, and parent 1 and parent 2 are identical. It was proposed originally for traveling salesman problem [19]. In MPO/AI, the longest common subsequence (maximum partial order) of the two parents is inherited to the offspring and the crossover is completed by repeatedly inserting arbitrary nodes (arbitrary insertion) not yet included into a feasible position of minimum cost. The same local improvement algorithm was used in all the genetic algorithms. In the tables, the frequency of finding solutions better than or equal to the best-known (BK#), the best cost (Best), average cost (Avg), group standard deviation ($\sigma/\sqrt{t}$), average generation (Gen), and average running time (Time) are presented. We got the results from 1000 ($= t$) runs on ESC78, ft70.∗, kro124p.∗, rbg1∗, and 50 runs on prob.100, rbg2∗, and rbg3∗. The values (%) after the best and average costs represent the percentages above the best-known[5]. VGA outperformed other genetic algorithms for twelve instances, while DGA and MGA outperformed the others for four instances and one instance, respectively. VGA broke the best-known for prob.100, rbg323a, and rbg341a. All three genetic algorithms consumed comparable running time for all instances except prob.100, rbg341a, rbg358a, and rbg378a. The overall results show that VGA is the most efficient and stable among them.

[4] Available at `http://www.cs.cmu.edu/afs/cs.cmu.edu/user/chens/WWW/MPOAI_SOP.tar.gz`.

[5] Available at `http://www.idsia.ch/~luca/has-sop.html`.

Table 3. The experimental results for rbg∗.

Graph (Bst-Kn)	GA	BK#/t	Best (%)	Avg (%)	$\sigma/\sqrt{t}$	Gen	Time (s)
	DGA	956/1000	1038 (0.000)	**1038.07** (0.007)	0.01	97	11.19
rbg109a	MGA	177/1000	1038 (0.000)	1039.88 (0.181)	0.04	772	16.51
(1038)	VGA	953/1000	1038 (0.000)	1038.12 (0.011)	0.02	209	11.88
	DGA	987/1000	1750 (0.000)	**1750.04** (0.002)	0.01	77	31.14
rbg150a	MGA	108/1000	1750 (0.000)	1752.63 (0.150)	0.03	331	33.12
(1750)	VGA	901/1000	1750 (0.000)	1750.30 (0.017)	0.03	216	34.20
	DGA	994/1000	2033 (0.000)	**2033.01** (0.001)	0.01	192	78.37
rbg174a	MGA	623/1000	2033 (0.000)	2033.71 (0.035)	0.04	381	67.14
(2033)	VGA	927/1000	2033 (0.000)	2033.15 (0.007)	0.02	433	85.85
	DGA	36/50	2950 (0.000)	2950.32 (0.011)	0.08	199	346
rbg253a	MGA	47/50	2950 (0.000)	2950.08 (0.003)	0.05	155	222
(2950)	VGA	50/50	2950 (0.000)	**2950.00** (0.000)	0.00	382	325
	DGA	1/50	3141 (0.000)	3144.20 (0.102)	0.28	866	2559
rbg323a	MGA	0/50	3142 (0.032)	3142.42 (0.045)	0.07	628	1281
(3141)	VGA	16/50	3140 (−0.032)	**3141.94** (0.030)	0.13	1358	2515
	DGA	0/50	2572 (0.078)	2575.30 (0.206)	0.33	1281	4262
rbg341a	MGA	0/50	2571 (0.039)	2578.32 (0.324)	0.55	1686	3174
(2570)	VGA	12/50	2568 (−0.078)	**2571.88** (0.073)	0.28	5620	10164
	DGA	3/50	2545 (0.000)	2553.98 (0.353)	0.76	1890	7345
rbg358a	MGA	0/50	2549 (0.157)	2555.24 (0.402)	0.54	17355	34675
(2545)	VGA	9/50	2545 (0.000)	**2548.56** (0.140)	0.41	8640	24340
	DGA	0/50	2819 (0.107)	2819.86 (0.137)	0.31	1065	7785
rbg378a	MGA	2/50	2816 (0.000)	2818.96 (0.105)	0.22	3873	11669
(2816)	VGA	22/50	2816 (0.000)	**2818.44** (0.087)	0.45	7814	33774

5 Conclusions

In this paper, we proposed a new hybrid genetic algorithm for the sequential ordering problem (SOP). It adopts a crossover, called Voronoi quantized crossover (VQX), on a complete graph representation. The crossover was modified by employing several new features for SOP. In the experiments, the proposed genetic algorithm outperformed state-of-the-art genetic algorithms for SOP. We suspect that the power of VQX is based on two main properties, convexity and diversity. The properties are believed to improve the performance of genetic algorithms by encouraging the survival probability and reproducibility of high-quality building blocks in the genetic process.

Acknowledgments. This work was partly supported by Optus Inc. and Brain Korea 21 Project. The RIACT at Seoul National University provided research facilities for this study.

References

1. N. Ascheuer, L. F. Escudero, M. Grotschel, and M. Stoer. A cutting plane approach to the sequential ordering problem (with applications to job scheduling in manufacturing). *SIAM Journal on Optimization*, 3:25–42, 1993.
2. L. F. Escudero, M. Guignard, and K. Malik. A Lagrangian relax-and-cut approach for the sequential ordering problem with precedence relationships. *Annals of Operations Research*, 50:219–237, 1994.
3. N. Ascheuer, M. Jünger, and G. Reinelt. A branch & cut algorithm for the asymmetric traveling salesman problem with precedence constraints. *Computational Optimization and Applications*, 17(1):61–84, 2000.
4. S. Chen and S. Smith. Commonality and genetic algorithms. Technical Report CMU-RI-TR-96-27, The Robotic Institute, Carnegie Mellon University, 1996.
5. L. M. Gambardella and M. Dorigo. An ant colony system hybridized with a new local search for the sequential ordering problem. *INFORMS Journal on Computing*, 12(3):237–255, 2000.
6. J. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, 1975.
7. Y. Davidor. Epistasis variance: Suitability of a representation to genetic algorithms. *Complex Systems*, 4:369–383, 1990.
8. D. I. Seo, Y. H. Kim, and B. R. Moon. New entropy-based measures of gene significance and epistasis. In *Genetic and Evolutionary Computation Conference*, 2003.
9. D. E. Goldberg. *Genetic Algorithms in Search, Optimization, Machine Learning.* Addison-Wesley, 1989.
10. D. I. Seo and B. R. Moon. A survey on chromosomal structures and operators for exploiting topological linkages of genes. In *Genetic and Evolutionary Computation Conference*, 2003.
11. D. I. Seo and B. R. Moon. Voronoi quantized crossover for traveling salesman problem. In *Genetic and Evolutionary Computation Conference*, pages 544–552, 2002.
12. T. N. Bui and B. R. Moon. A new genetic approach for the traveling salesman problem. In *IEEE Conference on Evolutionary Computation*, pages 7–12, 1994.
13. A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, 1992.
14. C. Anderson, K. Jones, and J. Ryan. A two-dimensional genetic algorithm for the Ising problem. *Complex Systems*, 5:327–333, 1991.
15. R. B. Hollstien. *Artificial Genetic Adaptation in Computer Control Systems.* PhD thesis, University of Michigan, 1971.
16. K. De Jong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan, 1975.
17. D. Cavicchio. *Adaptive Search Using Simulated Evolution.* PhD thesis, University of Michigan, 1970.
18. TSPLIB. http://www.iwr.uni-heidelberg.de/ groups/comopt/software/TSPLIB95/.
19. B. Freisleben and P. Merz. New genetic local search operators for the traveling salesman problem. In *Parallel Problem Solving from Nature*, pages 890–900. 1996.