ISSN 1392-8619 print/ISSN 1822-3613 online
ŪKIO TECHNOLOGINIS IR EKONOMINIS VYSTYMAS
TECHNOLOGICAL AND ECONOMIC DEVELOPMENT OF ECONOMY
http://www.tede.vgtu.lt
2006, Vol XII, No 1, 37–43

TESTING OF CROSSOVER OPERATORS FOR THE GREY PATTERN PROBLEM

Alfonsas Misevičius

Kaunas University of Technology, Department of Practical Informatics,
Studentų g. 50-400a/416a, LT-51368 Kaunas, Lithuania
E-mail: alfonsas.misevicius@ktu.lt

Received 3 October 2005; accepted 4 January 2006

Abstract. Recently genetic algorithms (GAs) are a great success in solving combinatorial optimization problems. In this paper the performance issues related to the genetic search in the context of the grey pattern problem (GPP) are discussed. The main attention is paid to the investigation of the solution recombination, i.e. crossover operators, which play an important role developing robust genetic algorithms. We implemented seven crossover operators within the hybrid genetic algorithm (HGA) framework, and carried out the extensive experiments in order to test the influence of the recombination operators on the genetic search process. The results obtained from the experimentation with GPP test instances (benchmarks) demonstrate promising efficiency of so-called multiple parent crossover which is based on a special type of recombination of several solutions-parents.

Keywords: combinatorial optimization, heuristic algorithms, genetic algorithms, crossover operators, grey pattern problem.

1. Introduction

The grey pattern problem (GPP) [1] is based on a rectangle (grid) of dimensions $n_1 \times n_2$ containing $n = n_1 \times n_2$ points (square cases) with m black points and $n - m$ white points. By juxtaposing many of these rectangles, one gets a grey pattern (frame) of density m/n. The objective is to get the finest grey pattern, that is, the black points have to be spread on the rectangle as regularly as possible. The larger n, the more refined pattern is possible. The grey pattern problem is a special case of a more general problem, the quadratic assignment problem (QAP) [2]. QAP is formulated in the following way. Let two matrices $\boldsymbol{A} = (a_{ij})_{n\times n}$ and $\boldsymbol{B} = (b_{kl})_{n\times n}$ and set Π of all possible permutations of the integers from 1 to n be given. The goal is to find permutation $\pi = (\pi(1), \pi(2), ..., \pi(n)) \in \Pi$ that minimizes

$$z(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi(i)\pi(j)}. \qquad (1)$$

In the grey pattern problem matrix $(a_{ij})_{n\times n}$ is defined as $a_{ij} = 1$ for $i, j = 1, 2, ..., m$ and $a_{ij} = 0$ otherwise. Matrix $(b_{kl})_{n\times n}$ is defined by the given values – the distances between every two of n points. More precisely, $b_{kl} = b_{n_2(r-1)+s\ n_2(t-1)+u} = f_{rstu}$, where

$$f_{rstu} = \max_{v,w\in\{-1,0,1\}} \frac{1}{(r-t+n_1 v)^2 + (s-u+n_2 w)^2},$$

$r, t = 1, ..., n_1$, $s, u = 1, ..., n_2$. f_{rstu} may be thought of as an electrical repulsion force between two electrons (to be put on the grid points) i and j ($i, j = 1, ..., n$) located in positions $k = \pi(i)$ and $l = \pi(j)$ with coordinates (r, s) and (t, u). The ith ($i \le m$) element of permutation π, $\pi(i) = n_2(r-1) + s$, gives the location in the rectangle where a black point has to be placed. The coordinates of the location $\pi(i)$ of the black point are derived according to the formulas: $r = ((\pi(i) - 1) \operatorname{div} n_2) + 1$, $s = ((\pi(i) - 1) \bmod n_2) + 1$, $i = 1, 2, ..., m$. ($x \operatorname{div} y = \lfloor x/y \rfloor$; $x \bmod y = x - \lfloor x/y \rfloor \times y$, where $\lfloor x/y \rfloor$ denotes the integer part of x/y which is always smaller than (or equal to) x/y.)

Many heuristic approaches can be applied for solving both QAP and, at that time, its particular case - the grey pattern problem (see, for example, [1, 3]). Recently, genetic algorithms (GAs) are among the advanced heuristic techniques for the quadratic assignment like problems, among them, GPP [4–8].

Very roughly, genetic algorithms may be characterized as follows [9]. Let P be a subset of Π; it is referred to as population, and it is composed of individuals, i.e. solutions (permutations), π_1, π_2, ... $\pi_{PS = |P|}$. (Further, we also shall call the solution (permutation), π, as a chromosome, the single position, i, of the solution – as a gene, and the value at the given position (gene), $\pi(i)$ – as an allele.) Each individual (π_i) is associated with fitness, i.e. the corresponding objective function value ($z(\pi_i)$). In this case individual π_i is preferred to individual π_j if $z(\pi_i) < z(\pi_j)$. The following are the main steps of the genetic search. A pair of members of P is selected to be parents. New solutions (i.e. offspring) are created by combining the parents; this recombination operator is known as a crossover. Afterwards, a replacement scheme is applied to determine which individuals survive to form the next generation. In addition, some individuals may undergo mutations. Over many generations, less fit individuals (worse solutions) tend to die-off, while better individuals (solutions) tend to predominate. The process is continued until a certain termination criterion is met.

In this paper the issues related namely to the genetic search for the grey pattern problem are concerned. The main attention is paid to the investigation of the recombination operators which play an important role constructing efficient GAs. The paper is organized as follows. A hybrid genetic algorithm (HGA) framework and recombination operators are discussed in Section 2. In Section 3 we present the results of testing several crossover operators within the improved HGA. Section 4 completes the paper with conclusions.

2. A hybrid genetic algorithm framework and recombination operators

2.1. A hybrid genetic algorithm framework: the state-of-the-art and extensions

The state-of-the-art genetic algorithms are rather hybrid, i.e. combined genetic local search algorithms which incorporate additional heuristic components [5, 6]. The example of such component is a post-crossover pro-cedure which is used as a local improvement algorithm applied to the solution previously produced by the crossover. Heuristic algorithms can also be applied for the construction of high quality initial populations. As a result, the hybrid genetic search is done in an optimized search space where the populations consist solely of local optima – this appears to be a more effective process than searching in a random solution space.

Applying HGAs it does not necessary mean that near-optimal solutions are reached in reasonable time. Indeed, HGAs often use the elaborated improvement heuristics (like simulated annealing, tabu search) that, in general, are quite time-consuming. This could be thought of as a serious shortcoming, especially if we wish to create HGAs that are competitive with other optimization techniques. In this situation it is important to make some additional extensions of HGAs. The following are the basic principles of designing the extended hybrid genetic algorithms (EHGAs): 1. EHGAs should incorporate as robust local improvement algorithms as possible. Here, we assume that algorithm A_1 is more efficient than algorithm A_2, if A_1 finds (in average) the solution(s) with the average objective function value (quality) $f^\lozenge$ in less time than A_2. Naturally, the long time behaviour does not matter as long as we are speaking about the fast algorithms within EHGAs. 2. In EHGAs the compactness of the population is highly desirable. As long as the efficient improvement procedures are used, the large populations are not necessary: the small size of the population is compensated by the robustness of the improvement algorithm. Obviously, the compact populations allow to save the computation time when comparing to HGAs which deal with larger populations. 3. EHGAs must maintain a sufficient degree of diversity within the population. This is especially true for the small populations. Indeed, the smaller the size of the population, the larger the probability that diversity will be lost quickly. To overcome this difficulty, so-called "cold restarts" may be proposed; here, as a "cold restart" we call deep reconstruction of the population, for example, the mutations applied to the members of population with the subsequent local improvement. "Cold restart" takes place each time the fact of premature convergence of the algorithm is determined, i.e. the level of the diversity within the current population is below a certain threshold.

The template of the extended hybrid genetic algorithm is presented in Fig 1 (see also [6, 10]). Note that in our

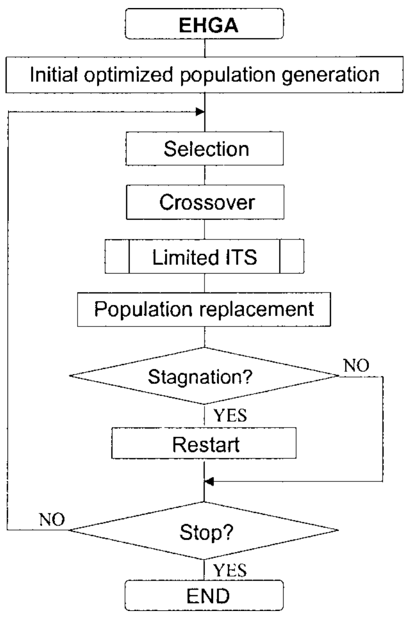


Fig 1. Basic flowchart of the extended hybrid genetic algorithm

experiments we applied a limited iterated tabu search (ITS) procedure in the role of a local improvement algorithm. ITS procedure (but with the increased number of iterations) is also used in both the initial population construction and the restart process. The details of ITS algorithm are omitted for the sake of brevity. Those interested in ITS approach are addressed to [11].

2.2. Recombination operators within hybrid genetic algorithms

As mentioned above, HGAs operate with high quality optimized populations. Despite this fact the recombination of solutions still remains one of the critical things constructing competitive genetic algorithms. Very likely, the role of recombination operators within HGAs is more significant than in ordinary GAs. In fact, we can think of HGA as a process that combines intensification and diversification (I&D) of the search [11, 12].

The intensification (local improvement) concentrates the search in limited portions of the solution space, while the diversification is responsible for escaping from the current local optimum and moving towards unvisited so far solutions. From this point of view, the crossover is a special sort diversification mechanism which guides the global search, i.e. exploration of new regions of the solution space. Thus, the proper exploration strategy is, in some sense, even more severe than the intensification process. In this situation we naturally make additional demands of crossover operators. The crossover is highly desirable to be "strong" enough to minimize the possibility of possible falling back into the previous local optima. On the other hand, if the crossover is too "disruptive", the resulting algorithm may be similar to a "blind" random multistart which is known to be not a very efficient method.

We start our discussion of the recombination operators with the crossover by Tate and Smith, 1995 [13]. It is called a uniform like crossover (ULX). ULX works as follows. First, all items assigned to the same position in both parents are copied to this position in the child. Second, the unassigned positions are scanned from left to right: for the unassigned position, an item is chosen randomly, uniformly from those in the parents, if they are not yet included in the child. Third, remaining items are assigned at random.

One of the modifications of ULX operator is a so-called block uniform like crossover (or simply block crossover (BX)). BX is distinguished for the fact that some blocks (segments) of elements are considered instead of the single elements. The block size is in the range $[1, \lfloor n/2 \rfloor]$. Copying blocks the feasibility of permutation must be kept.

The other recombination operator is a cycle crossover (CX) [14]. The key idea of this operator is that CX preserves the information contained in both parents, that is, all the alleles of the offspring are taken either from the first or the second parent. The main steps of CX are as follows. 1. All the alleles found at the same locations in both parents are assigned to the corresponding locations in the child. 2. Starting from the first (or randomly chosen) location, provided that the corresponding element has not been included in the offspring, an element is chosen in a random way from the parents. After this, one performs additional assignments to ensure that no random assignment occurs. Then, the next unassigned location is processed in the same manner until all the locations have been considered.

Ahuja et al., 2000, proposed a swap path crossover (SPX) [15]. Let π', π'' be a pair of parents. In SPX one starts at some random gene and the parents are examined from left to right until all the genes have been considered. If the alleles at the position being looked at are the same, one moves to the next position; otherwise, one performs a swap (interchange) of two alleles in π' or in π'' so that the alleles at the current position become alike. (For example, if the current gene is i, and $a = \pi'(i)$, $b = \pi''(i)$, then, after a swap, either $\pi'(i)$ becomes b, or $\pi''(i)$ becomes a.) Ahuja et al. suggests to perform the swap for which the corresponding solution has a lower objective function value. The elements in the two resulting solutions are then considered, starting at the next position, and so on. The best solution obtained (the fittest child) serves as an off-spring. The swap path crossover is illustrated in Fig 2.

One point crossover (OPX) operators are classical solution recombination procedures widely used in early versions of genetic algorithms [9]. One of the variants of OPX for QAP is due to Lim et al. [16]. The idea of OPX is quite simple. A crossing point (site) is chosen randomly between 1 and $n - 1$ in one of the parents. As a result, a child chro-

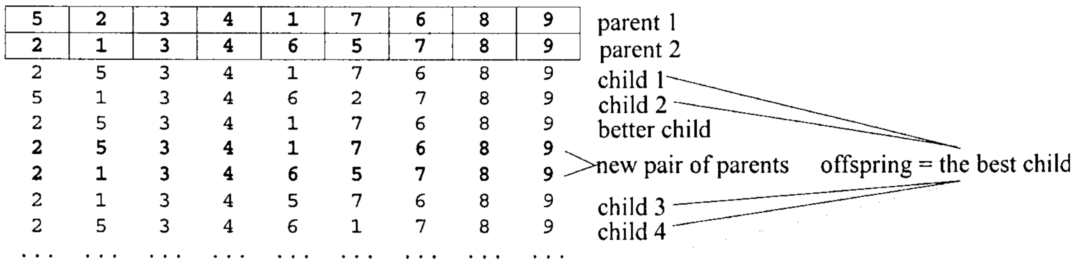

5 2 3 4 1 7 6 8 9 parent 1
2 1 3 4 6 5 7 8 9 parent 2
2 5 3 4 1 7 6 8 9 child 1
5 1 3 4 6 2 7 8 9 child 2
2 5 3 4 1 7 6 8 9 better child
2 5 3 4 1 7 6 8 9
2 1 3 4 6 5 7 8 9 new pair of parents
offspring = the best child
2 1 3 4 5 7 6 8 9 child 3
2 5 3 4 6 1 7 8 9 child 4
... ...

Fig 2. Example of a swap path crossover

mosome is obtained containing information partially determined by each of parent chromosomes.

Recently, Drezner introduced an original recombination operator – a cohesive crossover (COHX) [4]. COHX produces the offspring in several steps. At the beginning, some mask – $n_1 \times n_2$ matrix $\boldsymbol{M}$ – is created (n_1, n_2 are GPP dimensions). The initial mask position is fixed at (i_0, j_0), where $i_0 \in \{1, 2, ..., n_1\}$, $j_0 \in \{1, 2, ..., n_2\}$. Matrix $\boldsymbol{M}$ is then filled in according to a wave propagation fashion (see Fig 3).

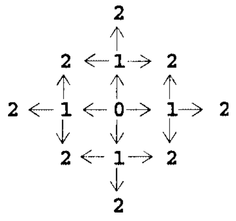


Fig 3. Filling in a mask

There exist n different masks $\boldsymbol{M}^{(1)}, \boldsymbol{M}^{(2)}, ..., \boldsymbol{M}^{(k)}, ..., \boldsymbol{M}^{(n)}$. k, i_0, and j_0 are in the following relation: $k = n_2(i_0 - 1) + j_0$, $i_0 = 1, 2, ..., n_1$, $j_0 = 1, 2, ..., n_2$. kth recombined solution $\pi^{(k)}$ ($k \in \{1, 2, ..., n\}$) is generated in three steps:

$$1)\ \pi^{(k)}(i) = \begin{cases} \pi_b(i) & \text{if } \boldsymbol{M}^{(k)}(((k-1) \operatorname{div} n_2)+1, \\ & \qquad ((k-1) \bmod n_2)+1) \le \eta, \\ 0 & \text{otherwise} \end{cases}$$

where $i = 1, 2, ..., n$, $\pi_b = \operatorname{argmin} \{z(\pi'), z(\pi'')\}$, π', π'' are the solutions-parents, and η is the median of $\boldsymbol{M}^{(k)}$, i.e.

$$\eta = \frac{\sum_i^{n_1} \sum_j^{n_2} \boldsymbol{M}^{(k)}(i,j)}{n_1 n_2};$$

$$2)\ \pi^{(k)}(i) = \begin{cases} \pi^{(k)}(i) & \text{if } \pi^{(k)}(i) > 0 \\ \pi_w(i) & \text{if } \pi^{k}(i) = 0 \wedge \pi_w(i) \text{ not in } \pi^{(k)}; \\ 0 & \text{otherwise} \end{cases}$$

where $i = 1, 2, ..., n$, $\pi_w = \operatorname{argmax} \{z(\pi'), z(\pi'')\}$;
3) for every unassigned position i ($\pi^{(k)}(i) = 0$), an item is chosen randomly from those not yet included in the offspring.

A visual example of generation of a solution is given in Fig 4. As a result, n solutions are produced, but only the best of them is regarded as an offspring, i.e. $\pi^\circ = \arg\min_{k=1,2,...n} z(\pi^{(k)})$.

Multiple parent crossover (MPX) was described by Misevičius in [7], although the idea of using combinations of several solutions goes back to Boese et al al. [17]. MPX is distinguished for the fact that the offspring derives the information from many parents – this is the contrast and, at that time, the advantage to the traditional operators where two parents are used only. In MPX, the ith element $\pi^\circ(i)$ is created by choosing a not yet chosen number j in such a way that probability $\Pr(\pi^\circ(i) = j)$ is maximized. Here, probability $\Pr(\pi^\circ(i) = j)$ is equal to $d_{ij} \big/ \sum_j^n d_{ij}$, where d_{ij} is the entry of desirability matrix $\boldsymbol{D} = (d_{ij})_{n \times n}$. The value of d_{ij} is determined by sum $q_{ij} + \varepsilon$, where q_{ij} is the number of times that element i is assigned to position $j = \pi(i)$ in μ parents (which participate in the creation of the child), and ε is a correction (noise). The process is to be continued until all the genes of the offspring take on their values. An example of producing the offspring in multiple parent crossover ($\mu = 5$) is given in Fig 5.

3. Testing of the extended hybrid genetic algorithm for the grey pattern problem

In this section we present the results of experimental comparison of the crossovers outlined above. In the experiments we used the instances of GPP generated according to the method described in [1]. For the set of problems tested the size of instances, n, equal to 256, and the frames (rectangles) are of dimensions 16 × 16, i.e. $n_1 = n_2 = 16$. The instances are denoted by the name grey_16_16_m, where m is the number of black points. Remind that for these instances data matrix $\boldsymbol{B}$ remains unchanged, while the data matrix $\boldsymbol{A}$ is of the form $\begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$, where $\mathbf{1}$ is a sub-matrix of size $m \times m$ composed of 1s only.

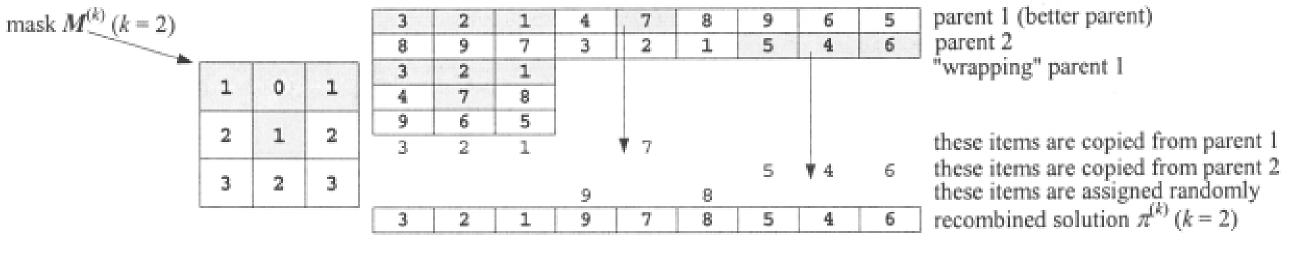
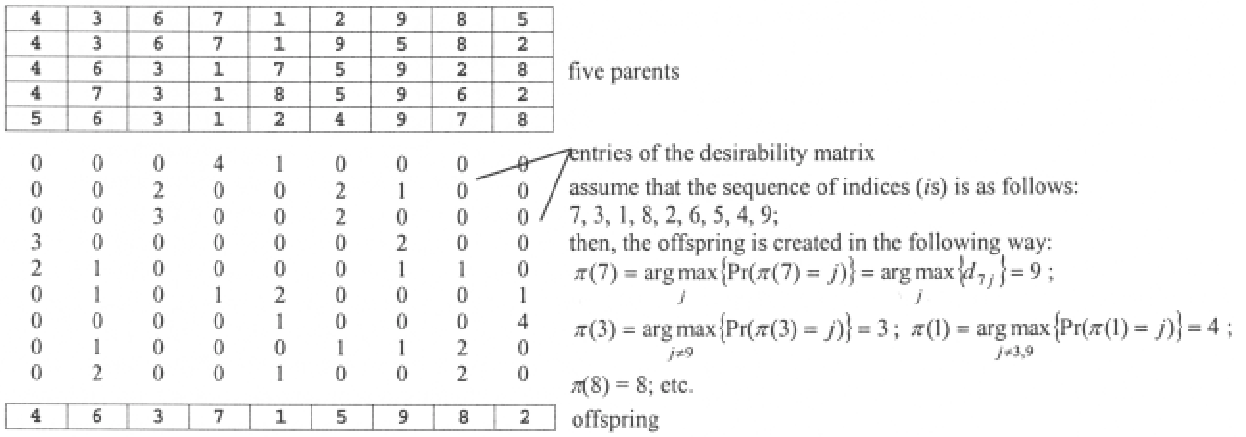


Fig 4. Example of a cohesive crossover



4	3	6	7	1	2	9	8	5
4	3	6	7	1	9	5	8	2
4	6	3	1	7	5	9	2	8
4	7	3	1	8	5	9	6	2
5	6	3	1	2	4	9	7	8

five parents

$$\begin{matrix} 0 & 0 & 0 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 2 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 2 & 0 \\ 0 & 2 & 0 & 0 & 1 & 0 & 0 & 2 & 0 \end{matrix}$$

entries of the desirability matrix

assume that the sequence of indices (is) is as follows:
7, 3, 1, 8, 2, 6, 5, 4, 9;
then, the offspring is created in the following way:

$\pi(7) = \arg\max_{j} \{\Pr(\pi(7) = j)\} = \arg\max_{j} \{d_{7j}\} = 9$;

$\pi(3) = \arg\max_{j \neq 9} \{\Pr(\pi(3) = j)\} = 3$; $\pi(1) = \arg\max_{j \neq 3,9} \{\Pr(\pi(1) = j)\} = 4$;

$\pi(8) = 8$; etc.

4	6	3	7	1	5	9	8	2

offspring

Fig 5. Example of a multiple parent crossover

We used the extended hybrid genetic algorithm discussed in Section 2.1 as an experimental basis for the crossover operators. The efficiency measure for the crossover operators is the average deviation of solutions obtained from the best known solution – $\bar{\delta}$ ($\bar{\delta} = 100(\bar{z} - \breve{z})/\breve{z}$ [%], where $\bar{z}$ is the average objective function value over 10 restarts (single applications of EHGA to a given instance), and $\breve{z}$ is the best known value (BKV) of the objective function). In the experimental comparison equal conditions are created: all the crossover variants use the identical initial solutions and require approximately the same CPU time. The following are the values of the control parameters of EHGA: population size – 8; number of generations – 25; number of offsprings per generation – 1; number of iterations of the post-crossover (ITS procedure) – $5n$. The number of parents in MPX crossover is equal to the population size.

The results of the comparison of the crossover operators are presented in Table 1. The results from Table 1 demonstrate that crossovers have considerable influence on the final solutions produced by the genetic algorithm. This is true despite of the fact that the powerful post-crossover procedure is used. This indicates that the recombination operators, which are responsible for the exploration of new regions in the solution space hide high potential. The performance of different crossovers varies in quite large ranges; nevertheless, some regularities can be discovered. For example, less disruptive crossovers (OPX, COHX) appear to be more efficient than highly disruptive crossovers (ULX); surprisingly, the cycle crossover (the minimally available disruptive crossover) produces only medium-quality results. So, it could be concluded that a good crossover should bring some randomness to the offspring, however this must be done in a subtle way. It can also be seen that the crossovers that incorporate some a priori knowledge about the problem being solved (for example, SPX) seem to be better than the "pure" operators (for example, BX). The preliminary

Table 1. Comparison of the crossover operators for GPP. The best results obtained are printed in bold face. CPU time per restart is given in seconds. 3 GHz PENTIUM computer was used in the experiments

Instance	BKV	$\bar{\delta}$							CPU time
		ULX	BX	CX	SPX	OPX	COHX	MPX	
grey_16_16_50	11017342[a]	0.032	0.023	0.032	**0.019**	0.026	0.032	0.023	*3.5*
grey_16_16_55	13661614[b]	0.071	0.058	0.055	0.050	0.034	0.055	**0.031**	*5.0*
grey_16_16_60	16575644[a]	0.017	0.012	0.025	0.009	**0.004**	0.012	0.015	*5.9*
grey_16_16_65	19848790[b]	0.024	0.019	0.035	0.025	**0.018**	0.025	0.031	*6.5*
grey_16_16_70	23852796[b]	0.229	0.208	0.222	**0.185**	0.203	0.213	0.224	*6.8*
grey_16_16_75	28114952[b]	0.127	0.113	0.115	0.117	0.120	0.116	**0.106**	*7.0*
grey_16_16_80	32593088[b]	0.180	0.185	0.177	0.179	0.162	0.179	**0.147**	*7.2*
grey_16_16_85	37379304[b]	0.154	0.135	0.135	0.112	0.132	0.114	**0.107**	*7.7*
grey_16_16_90	42608826[c]	0.138	0.104	0.121	0.103	0.116	**0.090**	0.099	*8.3*
grey_16_16_95	48081112[d]	0.176	0.161	0.169	0.153	**0.141**	0.160	0.152	*8.5*
grey_16_16_100	53838088[a]	0.129	0.110	0.113	0.108	0.108	**0.105**	0.111	*9.0*
Average:		0.116	0.103	0.109	0.096	0.097	0.100	**0.095**	

[a] comes from [8]; [b] comes from [12]; [c] comes from [5]; [d] comes from [11].

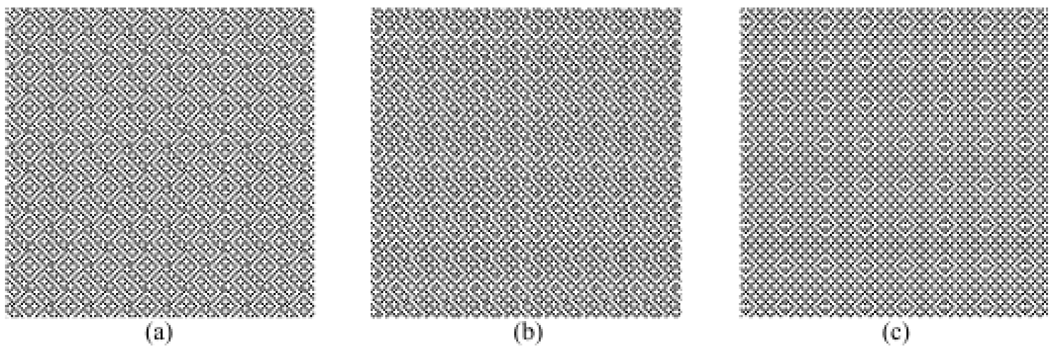


Fig 6. Examples of grey frames of densities 91/256 (a), 93/256 (b), 94/256 (c)

Table 2. New best known solutions of the grey pattern problem

Instance name	Previous best known value	New best known value
grey_16_16_73	26382310[a]	26375828
grey_16_16_83	35444806[b]	35443938
grey_16_16_84	36397376[b]	36395172
grey_16_16_85	37379304[a]	37378800
grey_16_16_90	42608826[b]	42597626
grey_16_16_91	43694968[c]	43676474
grey_16_16_93	45883642[c]	45870244
grey_16_16_94	46979436[c]	46975856

[a] comes from [12]; [b] comes from [5]; [c] comes from [11].

ranking of the crossover operators (sorted according to the decreasing quality of solutions) looks as follows: **MPX-SPX-OPX**-COHX-BX-CX-ULX. (It was somewhat unexpected that OPX produced slightly better results than COHX, which, in turn, was shown by Drezner [4] to be very effective for QAP. Thus, some more experiments would be useful in order to acknowledge the above ranking as really fair.) To summarize, SPX, OPX, and especially MPX appear to be superior to the remaining crossovers and could be recommended as proper recombination operators for the designers of new genetic algorithms for GPP, QAP and similar problems.

After the additional extensive experimentation we managed to solve many instances to pseudo-optimality. Moreover, we were successful in finding new record-breaking solutions for several instances. These solutions are presented in Table 2. We also give the graphical illustrations of three new solutions in Fig 6.

4. Conclusions

In this paper we present the results of testing of the hybrid genetic algorithm applied to the grey pattern problem, the special case of the quadratic assignment problem. The main attention was paid to the investigation of the recombination, i.e. crossover operators, which play one of the main roles in the efficient genetic algorithms.

We implemented seven crossover procedures and carried out several experiments in order to find out what is the difference of the solutions produced by these cross-overs. From the results obtained it can be seen that the crossover operators influence the final results of GA considerably, even in the cases when powerful post-crossover procedures are applied. The results of the experimental computations show relatively high performance of the crossovers with a lower degree of disruption as well as the crossovers that incorporate the problem-oriented knowledge (COHX, OPX, SPX). Another effective operator is the multiple parent crossover - MPX - which is based on a special type of recombination of several solutions. The results demonstrate that MPX enables to achieve better solutions than the standard two-parent operators. More precisely, the results of MPX are up to 22 % better than those of 2-parent crossovers. The power of MPX is also corroborated by the fact that new best known solutions for eight GPP instances have been discovered.

Further elaboration of the multiple parent crossover operator for GPP and similar combinatorial optimization problems, like QAP, TSP, could be one of the promising directions for future research.

References

1. Taillard, E. Comparison of iterative searches for the quadratic assignment problem. *Location Science*, Vol 3, 1995, p. 87–105.
2. Koopmans, T.; Beckmann, M. Assignment problems and the location of economic activities. *Econometrica*, Vol 25, 1957, p. 53–76.
3. Burkard, R. E.; Çela, E.; Pardalos, P. M.; Pitsoulis, L. The quadratic assignment problem. In: Handbook of Combinatorial Optimization, Du D.Z., Pardalos P. M., editors, Vol 3. Dordrecht: Kluwer, 1998, p. 241–337.
4. Drezner, Z. A new genetic algorithm for the quadratic assignment problem. *INFORMS Journal on Computing*, Vol 15, 2003, p. 320–330.
5. Misevicius, A. Genetic algorithm hybridized with ruin and recreate procedure: application to the quadratic assignment problem. *Knowledge-Based Systems*, Vol 16, 2003, p. 261–268.

6. Misevičius, A. An extension of hybrid genetic algorithm for the quadratic assignment problem. *Information Technology and Control*, Vol 33, 2004, p. 53–60.
7. Misevičius, A.; Rubliauskas, D. Performance of hybrid genetic algorithm for the grey pattern problem. *Information Technology and Control*, Vol 34, No 1, 2005, p. 15–24.
8. Taillard, E.; Gambardella, L. M. Adaptive memories for the quadratic assignment problem. Tech. Report IDSIA-87-97, Lugano, Switzerland, 1997.
9. Goldberg, D. E. Genetic Algorithms in Search, Optimization and Machine Learning. Reading: Addison-Wesley, 1989.
10. Misevičius, A.; Kilda, B. Comparison of crossover operators for the quadratic assignment problem. *Information Technology and Control*, Vol 34, No 2, 2005, p. 109–119.
11. Misevicius, A. A tabu search algorithm for the quadratic assignment problem. *Computational Optimization and Applications*, Vol 30, 2005, p. 95–111.
12. Misevicius, A. Ruin and recreate principle based approach for the quadratic assignment problem. In: Lecture Notes in Computer Science, Cantú-Paz E., Foster J. A., Deb K. et al., editors, Vol 2723. Genetic and Evolutionary Computation - GECCO 2003, Proceedings, Part I. Berlin-Heidelberg: Springer, 2003, p. 598–609.
13. Tate, D.M.; Smith, A. E. A genetic approach to the quadratic assignment problem. *Computers & Operations Research*, Vol 1, 1995, p. 73–83.
14. Merz, P.; Freisleben, B. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Transactions on Evolutionary Computation*, Vol 4, 2000, p. 337–352.
15. Ahuja, R. K.; Orlin, J. B.; Tiwari, A. A greedy genetic algorithm for the quadratic assignment problem. *Computers & Operations Research*, Vol 27, 2000, p. 917–934.
16. Lim, M. H.; Yuan, Y.; Omatu, S. Efficient genetic algorithms using simple genes exchange local search policy for the quadratic assignment problem. *Computational Optimization and Applications*, Vol 15, 2000, p. 249–268.
17. Boese, K. D.; Kahng, A. B.; Muddu, S. A new adaptive multistart technique for combinatorial global optimizations. *Operations Research Letters*, Vol 16, 1994, p. 101–113.

KRYŽMINIMO (KROSOVERIO) OPERATORIŲ TYRIMAS SPRENDŽIANT „PILKŲ ŠABLONŲ“ SUDARYMO UŽDAVINĮ

A. Misevičius

S a n t r a u k a

Pastaraisiais metais pasiektas didelis progresas sprendžiant kombinatorinio optimizavimo uždavinius genetiniais algoritmais (GA). Šiame straipsnyje nagrinėjami GA efektyvumo klausimai „pilkų šablonų“ sudarymo (formavimo) uždavinio kontekste. Daugiausia dėmesio skiriama sprendinių kryžminimo (krosoverio) operatoriams, atliekantiems svarbų vaidmenį genetiniuose algoritmuose, tirti. Realizuoti septyni skirtingi kryžminimo algoritmai, kurie įtraukti į hibridinio genetinio algoritmo sudėtį, išbandyti sprendžiant minėtą uždavinį. Eksperimentinių tyrimų tikslas – nustatyti kryžminimo operatorių įtaką GA gaunamiems sprendiniams. Eksperimentų, atliktų su įvairiais „pilkų šablonų“ sudarymo uždavinio testiniais pavyzdžiais (duomenimis), rezultatai liudija labai aukštą kai kurių kryžminimo procedūrų efektyvumo laipsnį. Tai visų pirma pasakytina apie vadinamąjį „daugelio tėvų“ kryžminimą, kuris pagrįstas netrivialaus kelių sprendinių-tėvų požymių kombinavimu, naudojant algoritmą.

Pagrindiniai žodžiai: kombinatorinis optimizavimas, euristiniai algoritmai, genetiniai algoritmai, kryžminimo (krosoverio) operatoriai, „pilkų šablonų“ sudarymo uždavinys.

Alfonsas Misevičius. Doctor, Associate Professor, Department of Practical Informatics, Kaunas University of Technology. Author and co-author of over 60 articles on different topics of computer science, participant of 20 scientific conferences. Research interests: computer-aided design, design and applications of heuristics and meta- heuristics for combinatorial optimization problems.