# Using Dynastic Exploring Recombination to Promote Diversity in Genetic Search

Carlos Cotta, José M. Troya

Dpto. de Lenguajes y Ciencias de la Computación, Univ. de Málaga
Campus de Teatinos (3.2.49), 29071 - Málaga - Spain
{ccottap, troya}@lcc.uma.es

**Abstract.** A family of recombination operators is studied in this work. These operators are based on keeping and using certain information about the past evolution of the algorithm to guide the recombination process. Within this framework, several recombination operators are specifically designed to preserve diversity within the population, while avoiding implicit mutations. The empirical evaluation of these operators on instances of two test problems ($k-$EMP and permutation flowshop) shows an improvement of the results with respect to other classical operators. This improvement seems to related to the increasing degree of epistasis of the problem.

## 1 Introduction

One of the most distinctive features of genetic algorithms with respect to other related techniques such as evolution strategies or evolutionary programming is the emphasis put on the use of recombination operators. In the most classical scenario, recombination is given a exploitative rôle: its purpose is to combine valuable portions of solutions independently discovered, while the more exploratory task of introducing new genetic material corresponds to the mutation operator [11]. In this context, the loss of diversity is one of the main problems that can take place, resulting in the stagnation of the algorithm (which turns to be incapable of producing new promising solutions) and the increasing resampling of solutions [4] (with the subsequent waste of computational resources).

These problems are usually tackled by means of rising the mutation rate [10] or by using non-panmictic populations[1] [17, 19]. Despite both options can be effective, they require determining and adjusting several parameters (rate of change of the mutation rate, interconnection topology, migration frequency, etc.). This is generally a difficult step, and in spite of active research being conducted to assist setting these parameters (e.g., see [2, 3]), trial-and-error is still very often used by many researchers. In this work, an alternative (and complementary) approach is presented. This approach is termed *Dynastic Exploring Recombination* and comprises a family of recombination operators that consider the past

[1] Populations with spatial structure: islands, grids, etc.

evolution of the algorithm, trying to build promising unexplored macroformae [14]. This family of operators has been successfully evaluated on two problems: the $k-$EMP problem (a tunable-epistasis theoretical problem) and makespan minimisation in permutation flowshop problems [12].

The remainder of the paper is organised as follows. First, the basis of Dynastic Exploring Recombination is shown (Sect. 2). Subsequently, an empirical evaluation of these recombination operators is presented, focusing on aspects related to diversity and epistasis (Sect. 3). Finally, some conclusions are extracted and future work is outlined (Sect. 4).

## 2 Dynastic Exploring Recombination

Before defining Dynastic Exploring Recombination (DER) some previous concepts must be stated. Such background is presented is Subsect. 2.1. Next, some insight on the internal mechanics of recombination is provided in Subsect. 2.2. Finally, random transmission and dynastic exploring operators are introduced in Subsect. 2.3.

### 2.1 Background

Let $\Xi = \{\psi_1, \cdots, \psi_n\}$ be a set of $n$ independent equivalence relations defined over a discrete search space $\mathcal{S}$ verifying that for every possible pair of solutions $x$ and $y$, there exists at least an equivalence relation in $\Xi$ such that $x$ and $y$ belong to different equivalence classes. In this case, $\Xi$ *covers* the search space $\mathcal{S}$, and every $x \in \mathcal{S}$ can be univocally represented as $x = \{\eta_1, \cdots, \eta_n\}$, where $\eta_i$ is the equivalence class (for simplicity, the same symbol is used both to denote a equivalence class and for labelling it) to which $x$ belong under $\psi_i$. Thus, $x = \{\eta_1, \cdots, \eta_n\} \Longleftrightarrow x \in \bigcap_{i=1}^{n} \eta_i$. Each of these equivalence classes $\eta_i$ is termed a *basic forma* [14].

Let $x = \{\eta_1, \cdots, \eta_n\}$ and $y = \{\zeta_1, \cdots, \zeta_n\}$ be two feasible solutions. A recombination operator $X$ can be defined as a function $X : \mathcal{S} \times \mathcal{S} \times \mathcal{S} \rightarrow [0, 1]$, where $X(x, y, z)$ is the probability of generating $z$ when $x$ and $y$ are recombined using $X$. Now, the *Immediate Dynastic Span* [15] of $x$ and $y$ with respect to a recombination operator $X$ is $\Gamma^1_X(\{x, y\}) = \{z \mid X(x, y, z) > 0\}$, i.e., the set of solutions that can be obtained when $X$ is applied on $x$ and $y$. On the other hand, the *Dynastic Potential* $\Gamma(\{x, y\})$ of $x$ and $y$ is defined as $\Gamma(\{x, y\}) = \{z \mid \forall \xi : z \in \xi \Rightarrow (x \in \xi) \vee (y \in \xi)\}$, where $\xi$ is any basic forma. Hence, the dynastic potential of two individuals is the set of offspring that can be constructed using nothing but the information contained in the parents.

If $\Gamma^1_X(\{x, y\}) \subseteq \Gamma(\{x, y\})$, $X$ is termed a *transmitting operator* [15]. As already mentioned in Sect. 1, this property of transmission captures the most classical rôle of the recombination operator. Next subsection will study the functioning mechanics of these operators.

### 2.2 The Mechanics of Recombination

A transmitting recombination can be generally considered as a process in which information is incrementally taken from the parents to construct the offspring: starting from a totally unspecified solution, properties of the any of the parents are selected and assigned to the child until a fully-specified solution is obtained. This incremental process is not necessarily linear but may exhibit a more complex behaviour, including some kind of backtracking in the assignment of properties (e.g., Dynastically Optimal Recombination [5]). With this consideration in mind, any transmitting recombination operator $X_t$ can be defined as

$$X_t(x, y, z) = \prod_{i=1}^{n} \delta_{X_t}(\Psi_{i-1}, \xi_i, x, y), \tag{1}$$

where $z = \{\xi_1, \cdots, \xi_n\}$ and $\delta_{X_t}$ is a function $\delta_{X_t} : 2^{\mathcal{S}} \times 2^{\mathcal{S}} \times \mathcal{S} \times \mathcal{S} \to [0, 1]$. In this function, the first parameter represents the partially-specified solution at a certain step, i.e., $\Psi_0 = \mathcal{S}$, $\Psi_i = \Psi_{i-1} \cap \xi_i$. Since $\Xi$ covers $\mathcal{S}$, it is easy to see that $\Psi_n = \{z\}$. Now, $\delta_{X_t}(\Psi, \xi, x, y)$ is the probability that $X_t$ selects forma $\xi$, given that the partially-specified solution so far is $\Psi \ni z$. Clearly, a transmitting operator must verify that

$$[(x \notin \xi) \wedge (y \notin \xi)] \Rightarrow \delta_{X_t}(\Psi, \xi, x, y) = 0. \tag{2}$$

The nature of $\delta_{X_t}$ allows classifying transmitting operators into different categories. In particular, this work will focus on the case in which no backtracking in the assignment of properties is performed. The resulting family of operators is hence characterised for having a linear functioning mechanics, thus introducing a lower computational overhead in the algorithm. Some elements of this family are studied in the next subsection.

### 2.3 Random Transmission vs. Exploring Transmission

The simplest way of performing transmission is at random. Operators working in this way are comprised within the so-called *Random Transmitting Recombination* (RTR) [15], which is defined as follows:

$$\mathrm{RTR}(x, y, z) = \begin{cases} \frac{1}{|\Gamma(\{x,y\})|} & z \in \Gamma(\{x, y\}) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Thus, RTR returns a random individual in the dynastic potential of the recombined solutions. If $\Xi$ is orthogonal, i.e., if all combinations of formae induced by different equivalence relations are feasible, it is easy to see that $\Gamma(\{x, y\}) \equiv \prod_{i=1}^{n} \{\eta_i, \zeta_i\}$, that is, the $n$-dimensional Cartesian product of all basic formae to which $x$ or $y$ belong. In this situation, RTR is defined by the following $\delta_{\mathrm{RTR}}$:

$$\delta_{\mathrm{RTR}}(\Psi, \xi, x, y) = \begin{cases} 0 & (x \notin \xi) \wedge (y \notin \xi) \\ 1 & (x \in \xi) \wedge (y \in \xi) \\ 1/2 & \text{otherwise} \end{cases} \tag{4}$$

Notice that RTR $\equiv$ UX if each $\psi_i$ induce two equivalence classes. In a more general situation (non-orthogonal separability [14]), it is necessary to consider higher-level units termed *compatibility sets* [5] into which solutions are structured. Using the notation $\xi \triangleright \Psi$ to denote that, given $\Psi = \cap_{j=1}^{s} \theta_j$, $\exists j : \xi \equiv \theta_j$, where $\xi$ and $\theta_j$ are formae induced by the same equivalence relation $\psi_i \in \Xi$, each of these compatibility sets is inductively defined as[2]

$$\eta_{j_1} \triangleright K(\eta_{j_1}, x, y) \tag{5}$$

$$[\Gamma(\{x, y\}) \cap K(\eta_{j_1}, x, y) \cap \varpi(\eta_{j_k}, x, y) = \emptyset] \Rightarrow \eta_{j_k} \triangleright K(\eta_{j_1}, x, y), \tag{6}$$

where $x \in \eta_{j_1}$ and $\varpi(\eta_{j_k}, x, y)$ is the *dual forma* of $\eta_{j_k}$, that is, $\varpi(\eta_{j_k}, x, y) = \zeta_{j_k}$ if $x \in \eta_{j_k}$, $y \in \zeta_{j_k}$ and both $\eta_{j_k}$ and $\zeta_{j_k}$ are formae induced by the same $\psi_r$ (idem changing $x$ by $y$). For the sake of simplicity, it is assumed that $K(\eta, x, y) = K(\eta, y, x)$ whenever $y \in \eta, x \notin \eta$. Now, RTR is determined by the following function $\delta^K_{\mathrm{RTR}}$:

$$\delta^K_{\mathrm{RTR}}(\Psi, \xi, x, y) = \begin{cases} 0 & [(x \notin \xi) \wedge (y \notin \xi)] \vee [\exists \zeta \triangleright \Psi : \varpi(\xi, x, y) \triangleright K(\zeta, x, y)] \\ 1 & [(x \in \xi) \wedge (y \in \xi)] \vee [\exists \zeta \triangleright \Psi : \xi \triangleright K(\zeta, x, y)] \\ 1/2 & \text{otherwise} \end{cases} \tag{7}$$

Thus, a random selection is done over compatibility sets rather than over isolated formae. As mentioned in Sect. 1, defining a uniform probability distribution over these compatibility sets may be inappropriate for several reasons. First, resampling becomes a problem as the algorithm converges [4]. Second, it is sensitive to stochastic errors due the finite size of the population (e.g., valuable genetic material may be lost before it is exploited).

In order to avoid these undesirable effects, recombination operators that consider the past evolution of the algorithm can be used. Such past evolution is comprised in the histogram function $h_{\mathcal{A}}$ of algorithm $\mathcal{A}$ under consideration. This function is defined as $h_{\mathcal{A}} : \mathcal{S} \to 2^{\mathcal{N} \times \mathbb{R}}$, such that $h_{\mathcal{A}}(x) = \{(e_1, f_1), \cdots, (e_k, f_k)\}$ implies that element $x$ has been evaluated in iterations $e_1, \cdots, e_k$, and has respectively obtained the fitness values $f_1, \cdots, f_k$ (typically $\forall i[f_i = f]$). Several ways exist for using the information provided by the histogram function. For example, it might be useful to avoid generating solutions that were already considered in the past. However, such an operator would be computationally expensive and would in general exhibit a non-linear behaviour. A much more efficient template is shown below (orthogonality is assumed for simplicity):

$$\delta_{\mathrm{DER}}(\Psi, \xi, x, y) = \begin{cases} 0 & [(x \notin \xi) \wedge (y \notin \xi)] \vee [\phi(\varpi(\xi, x, y), h_{\mathcal{A}}) < \phi(\xi, h_{\mathcal{A}})] \\ 1 & [(x \in \xi) \wedge (y \in \xi)] \vee [\phi(\xi, h_{\mathcal{A}}) < \phi(\varpi(\xi, x, y), h_{\mathcal{A}})] \\ 1/2 & \text{otherwise} \end{cases} \tag{8}$$

[2] This definition of compatibility sets slightly differs from [5] since separability implies independence with respect to the partially specified solution $\Psi$.

where $\phi$ is a function that returns a metric of forma $\xi$ with respect to the past history of the algorithm. According to this definition, DER (*Dynastic Exploring Recombination*) would try to combine formae that locally minimise such metric. We are especially concerned about promoting diversity in the population and hence we have considered the following possibilities:

- $\phi_1(\xi, h_{\mathcal{A}}) = \sum_{x \in \xi} |h_{\mathcal{A}}(x)|$, that is, the number of times a solution $x \in \xi$ has been evaluated.

- $\phi_2(\xi, h_{\mathcal{A}}) = \sum_{x \in \xi} \sum_{(e,f) \in h_{\mathcal{A}}(x)} f$, that is, the accumulated fitness of all solutions in $\xi$ evaluated so far.

These two metrics are intended to direct the search towards regions of the search space not yet explored or at least not so explored as other regions. Thus, both $\text{DER}_{\phi_1}$ and $\text{DER}_{\phi_2}$ try to mitigate the exploitative side of recombination, boosting exploration.

In addition to these two metrics, a third metric $\phi_3$ has been considered. This metric is defined as $\phi_3(\xi, h_{\mathcal{A}}) = \phi_2(\xi, h_{\mathcal{A}}) \,/\, \phi_1(\xi, h_{\mathcal{A}})$, that is, the average fitness of solutions in $\xi$ evaluated so far. This is an interesting measure since it fits the traditional vision of genetic algorithms as (above-average) schema processors, sharing some similitude with Rosete *et al.*'s explicit schema processing [16].

Notice that the bookkeeping involved in calculating any of these metrics is very simple and does not require storing every visited solution. In fact, it suffices to keep a table with the accumulated values of these metrics for each basic forma (the size of this table would be $n \times m$, where $n$ is the number of genes –equivalence relations in $\Xi$– and $m$ is the number of alleles per gene –basic formae per equivalence relation–). Subsequently, gene values are picked according to the distribution probability shown in Eq.(8).

## 3 Experimental Results

This section describes the experimental evaluation of the previously described operator. First, the experimental setup used is described. The empirical results obtained are subsequently reported.

### 3.1 Experimental Setup

The test suite used for testing the operators described comprises instances of two families of problems: $k$-epistatic minimal permutation problem and makespan minimisation in permutation flowshop problems.

The $k$-Epistatic Minimal Permutation ($k-$EMP) problem is a generalisation of the Minimal Permutation (MP) problem [5]. The latter is a minimisation problem defined by a $n \times n$ matrix $M = \{m_{ij} \mid 1 \leq i, j \leq n\}$ such that each row of $M$ is a permutation of the elements $\{0, \cdots, n-1\}$ and no column has more than one zero. Subsequently, a permutation $p = p_1 p_2 \cdots p_n$ is evaluated as $\text{MP}(p) = \sum_{1 \leq i \leq n} m_{i,p_i}$.

The constraints posed on $M$ ensure that there is a unique permutation (the minimal permutation) whose fitness value is 0. The $k-$EMP problem adds epistatic relations to the above expression. To be precise, it is defined as

$$k - \mathrm{EMP}(p) = \sum_{1 \leq i \leq n} \left[ m_{i,p_i} \cdot \prod_{j=\min(1,i-k)}^{i-1} \alpha(p_i, p_j) \right]. \tag{9}$$

In the instances considered in this work, the coefficients $\alpha(p_i, p_j)$ are drawn from a uniform distribution in $[1, 2]$.

As to the second problem, it is a well-known member of the $\mathcal{NP}$-hard class. It involves determining the order in which a set of jobs must be fed into a production chain composed of a number of machines. Provided that each job requires exclusive use of each machine for a certain time, the goal is to minimise the total completion time of the jobs [6, 12].

The solution space for both problems can be adequately represented in terms of non-orthogonal separable formae. More precisely, solutions for both problems are permutations of a set of elements, being the properties of these permutations appropriately grasped by means of both position and block formae. The former are defined as assignments of elements to individual positions, while the latter are defined as the intersection of a compact set of adjacent positions. Because of space limitations, we refer to [6, 7] for a formal description of these formae.

All experiments have been conducted with a steady-state [18] genetic algorithm (popsize = 100, $p_c = .9$, $p_m = 1/n$, where $n$ is the dimensionality of the problem) using ranking selection ($\eta^+ = 2.0$) and the swap mutation operator [13]. Each run of the algorithm comprises $10^5$ evaluations of the target function.

### 3.2 Empirical Evaluation

Table 1 shows results for different recombination operators on $k-$EMP instances of different dimensionalities and degrees of epistasis. As it can be seen, DER operators (with the exception of $\mathrm{DER}^*_{\phi_3}$) achieve a very good performance with respect to transmitting operators such as UCX ($\mathrm{RTR}^{\mathrm{pos}}$) and UBX ($\mathrm{RTR}^{\mathrm{block}}$), as well as with respect to classical operators such as PMX and OX. In these experiments, the improvement is clearer when epistasis is increased. This seems reasonable, since exploitative behaviours are more appropriate in the case of low (or null) epistasis in order to proceed towards near-optimal solutions. In this context, any mechanism introduced for promoting exploration simply slows down convergence. The scenario is different in the case of higher epistasis: keeping diversity becomes an important issue, and DER is better than RTR at this. Fig. 1 illustrates this fact, using population-entropy as a measure of diversity [8].

An important fact that must be noted is that DER keeps a higher diversity than UCX while manipulating the same basic units, and without introducing implicit mutation. An operator such as OX may provide a diversity measure similar or higher than DER, but this is done at the expense of introducing a considerable amount of implicit mutation. In this case, such implicit mutation is clearly detrimental as shown in Table 1.

**Table 1.** Results averaged for ten runs of different recombination operators on $k-$EMP instances of 50 and 75 elements.

| | 50 elements | | | | | | |
|---|---|---|---|---|---|---|---|
| Operator | k = 0 | k = 1 | k = 2 | k = 5 | k = 10 | k = 15 | k = 20 |
| OX | 134.4 | 184.4 | 271.6 | 614.0 | 2465.7 | 8582.5 | 25331.8 |
| UCX ($\mathrm{RTR}^{\mathrm{pos}}$) | 50.5 | 72.5 | 99.9 | 282.3 | 1060.5 | 4880.9 | 16303.6 |
| UBX ($\mathrm{RTR}^{\mathrm{block}}$) | 60.0 | 78.5 | 118.8 | 355.4 | 1256.1 | 4709.1 | 14877.1 |
| PMX | 61.3 | 91.2 | 104.9 | 312.6 | 1274.9 | 5103.0 | 17899.8 |
| $\mathrm{DER}_{\phi_1}^{\mathrm{pos}}$ | 88.6 | 75.0 | 102.8 | 276.2 | 966.8 | 3657.5 | 10483.2 |
| $\mathrm{DER}_{\phi_1}^{\mathrm{block}}$ | 61.0 | 83.4 | 109.9 | 269.1 | 1015.8 | 3728.4 | 8154.3 |
| $\mathrm{DER}_{\phi_2}^{\mathrm{pos}}$ | 53.0 | 80.6 | 109.5 | 272.3 | 930.2 | 2820.6 | 7960.8 |
| $\mathrm{DER}_{\phi_2}^{\mathrm{block}}$ | 53.6 | 71.9 | 102.7 | 267.3 | 943.4 | 3856.8 | 10874.4 |
| $\mathrm{DER}_{\phi_3}^{\mathrm{pos}}$ | 88.6 | 114.0 | 162.0 | 413.7 | 1810.3 | 8431.0 | 31418.4 |
| $\mathrm{DER}_{\phi_3}^{\mathrm{block}}$ | 61.0 | 97.5 | 143.0 | 377.2 | 1612.3 | 6279.6 | 28170.5 |
| | 75 elements | | | | | | |
| Operator | k = 0 | k = 1 | k = 2 | k = 5 | k = 10 | k = 15 | k = 20 |
| OX | 358.0 | 529.9 | 787.0 | 1927.1 | 8521.9 | 31552.1 | 111858.8 |
| UCX ($\mathrm{RTR}^{\mathrm{pos}}$) | 144.6 | 227.1 | 325.2 | 858.7 | 4309.8 | 18243.2 | 74282.2 |
| UBX ($\mathrm{RTR}^{\mathrm{block}}$) | 193.0 | 263.3 | 369.8 | 1059.6 | 4916.9 | 18388.9 | 75397.1 |
| PMX | 193.3 | 259.8 | 377.1 | 1009.2 | 4460.0 | 17644.6 | 76752.7 |
| $\mathrm{DER}_{\phi_1}^{\mathrm{pos}}$ | 174.0 | 261.3 | 396.5 | 1023.5 | 4193.0 | 15629.9 | 56070.4 |
| $\mathrm{DER}_{\phi_1}^{\mathrm{block}}$ | 180.7 | 246.3 | 338.5 | 950.8 | 3988.1 | 16997.7 | 54932.3 |
| $\mathrm{DER}_{\phi_2}^{\mathrm{pos}}$ | 192.0 | 256.2 | 375.2 | 1028.7 | 4189.1 | 17044.3 | 63446.8 |
| $\mathrm{DER}_{\phi_2}^{\mathrm{block}}$ | 180.7 | 245.4 | 354.3 | 904.8 | 3950.5 | 17486.6 | 56186.0 |
| $\mathrm{DER}_{\phi_3}^{\mathrm{pos}}$ | 213.3 | 300.4 | 449.7 | 1148.3 | 6320.1 | 24476.2 | 122873.6 |
| $\mathrm{DER}_{\phi_3}^{\mathrm{block}}$ | 202.4 | 294.2 | 394.8 | 1149.6 | 5420.5 | 21559.1 | 91910.5 |

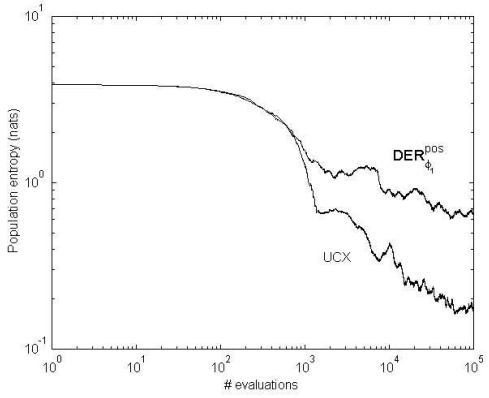


**Fig. 1.** Loss of diversity in a run of the GA for different recombination operators (UCX vs. $\mathrm{DER}_{\phi_1}^{\mathrm{pos}}$). The graph corresponds to a 20-EMP instance. Variants in $\mathrm{DER}_{\{\phi_1|\phi_2\}}^{*}$ behave very similar to $\mathrm{DER}_{\phi_1}^{\mathrm{pos}}$.

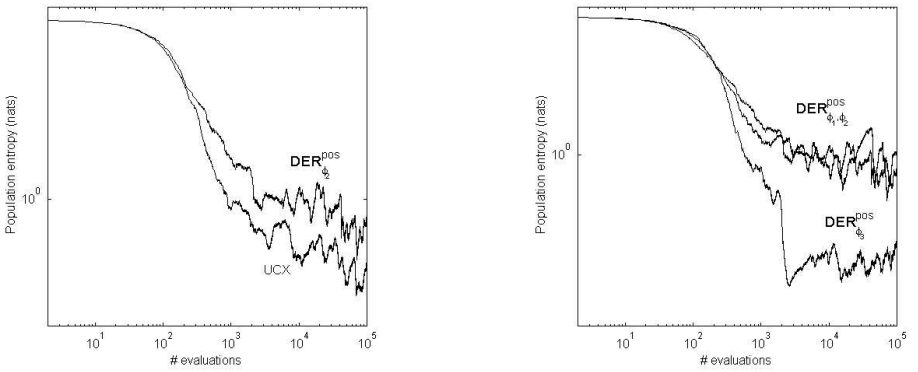


**Fig. 2.** Loss of diversity for different recombination operators in a run of the GA on the `rec31` instance of flowshop scheduling. The left graph corresponds to UCX vs. $\mathrm{DER}^{\mathrm{pos}}_{\phi_2}$, and the right one to the three variants of $\mathrm{DER}^{\mathrm{pos}}$.

The fact that $\mathrm{DER}^*_{\phi_3}$ provide comparatively worse results can be explained as follows: unlike $\mathrm{DER}^*_{\phi_1}$ and $\mathrm{DER}^*_{\phi_2}$, $\mathrm{DER}^*_{\phi_3}$ does not receive any negative feedback from the histogram function $h_{\mathcal{A}}$. Thus, if $\mathrm{DER}^*_{\phi_3}$ directs the search towards a certain region of $\mathcal{S}$ in which several formae have above-average fitness, it is simply boosting the theoretical behaviour of the algorithm as a forma/schema processor, increasing the rate of convergence (see Fig. 2 - right) and hence the possibility of falling into local optima. On the contrary, both $\mathrm{DER}^*_{\phi_1}$ and $\mathrm{DER}^*_{\phi_2}$ receive negative feedback, and try to direct the algorithm to regions of $\mathcal{S}$ not yet explored. Moreover, they try to avoid strongly-exploited regions. As mentioned above, this compensatory strategy can be more useful in the presence of epistasis.

These results are confirmed on flowshop scheduling instances taken from the OR-Library [1] (see Table 2). It must be noted that this problem has a high degree of epistasis, since scheduling a task at a given position affects all tasks subsequently scheduled. For this reason, both $\mathrm{DER}^*_{\phi_1}$ and $\mathrm{DER}^*_{\phi_2}$ yield the best results on these instances.

## 4 Conclusions

This work has studied the use of recombination operators specifically designed to promote exploration. This is done by keeping information about the past evolution of the algorithm in order to compensate the exploitative side of recombination. Unlike some related approaches (e.g., CHC [9]), rather than preventing the recombination of similar parents, the generation of different descendants is sought. It is important to consider that, according to the No Free Lunch Theorem [20], there exist problems in which the past information could mislead the algorithm. This seems to be the case of low-epistatic problems, in which exploitation is a better strategy and hence this compensation is detrimental.

**Table 2.** Results averaged for ten runs of different recombination operators on permutation flowshop instances.

| Operator | Problem instance | | | | | |
|---|---|---|---|---|---|---|
| | Rec19 | Rec25 | Rec31 | Rec37 | Rec39 | Rec41 |
| OX | 2121.8 | 2571.6 | 3144.1 | 5223.3 | 5296.9 | 5249.7 |
| UCX ($RTR^{pos}$) | 2126.6 | 2566.7 | 3127.0 | 5192.1 | 5270.6 | 5206.4 |
| UBX ($RTR^{block}$) | 2124.3 | 2575.2 | 3124.5 | 5197.2 | 5275.6 | 5205.8 |
| PMX | 2123.7 | 2577.2 | 3120.0 | 5208.0 | 5282.7 | 5207.2 |
| $DER^{pos}_{\phi_1}$ | 2119.4 | 2557.8 | 3122.3 | 5159.5 | 5268.9 | 5189.0 |
| $DER^{block}_{\phi_1}$ | 2119.5 | 2553.0 | 3122.2 | 5179.9 | 5337.2 | 5189.8 |
| $DER^{pos}_{\phi_2}$ | 2116.9 | 2551.8 | 3111.1 | 5172.1 | 5272.8 | 5188.6 |
| $DER^{block}_{\phi_2}$ | 2115.6 | 2563.0 | 3115.0 | 5174.3 | 5247.7 | 5194.2 |
| $DER^{pos}_{\phi_3}$ | 2134.9 | 2607.5 | 3147.0 | 5210.0 | 5307.3 | 5245.5 |
| $DER^{block}_{\phi_3}$ | 2125.6 | 2570.2 | 3137.0 | 5211.2 | 5279.5 | 5258.6 |

The experimental results are encouraging. It is empirically corroborated that some members of the DER family keep a higher level of diversity. Furthermore, performance improves in the presence of appreciable levels of epistasis, scenario in which convergence to local optima becomes more of an issue.

Several lines of future work are open. Firstly, it is clearly necessary to conduct more experiments so as to verify these results on other test problems. Secondly, it is interesting to study the relationship and interplay between these operators and spatially-structured evolution models. These two strategies can complement each other as mentioned in Sect. 1. In this sense, factors such as the convenience of using decentralised histogram functions are worth studying. Work is already in progress in this area.

## Acknowledgments

The authors wish to thank the anonymous reviewers for their interesting comments. Limitation of space has prevented us for further elaborating on some of the points they judiciously comment.
This work is partially supported by the Spanish *Comisión Interministerial de Ciencia y Tecnología* (CICYT) under grant TIC99-0754-C03-03.

## References

1. J.E. Beasley. OR-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990.
2. E. Cantú-Paz. Designing efficient and accurate parallel genetic algorithms. Technical Report 99017, Illinois Genetic Algorithm Laboratory, 1999.
3. E. Cantú-Paz. Topologies, migration rates and multi-population parallel genetic algorithms. Technical Report 99007, Illinois Genetic Algorithm Laboratory, 1999.

4. C. Cotta. On resampling in nature-inspired heuristics. In V. Botti, editor, *Proceedings of the Seventh Conference of the Spanish Association for Artificial Intelligence*, pages 145–154, 1997. In Spanish.
5. C. Cotta, E. Alba, and J.M. Troya. Utilising dynastically optimal forma recombination in hybrid genetic algorithms. In A.E. Eiben, Th. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving From Nature V*, volume 1498 of *Lecture Notes in Computer Science*, pages 305–314. Springer-Verlag, Berlin, 1998.
6. C. Cotta and J.M. Troya. Genetic forma recombination in permutation flowshop problems. *Evolutionary Computation*, 6(1):25–44, 1998.
7. C. Cotta and J.M. Troya. On the influence of the representation granularity in heuristic forma recombination. In J Carroll, E. Damiani, H. Haddad, and D. Oppenheim, editors, *ACM Symposium on Applied Computing 2000*, pages 433–439. ACM Press, 2000.
8. Y. Davidor and O. Ben-Kiki. The interplay among the genetic algorithm operators: Information theory tools used in a holistic way. In R. Männer and B. Manderick, editors, *Parallel Problem Solving From Nature II*, pages 75–84, Amsterdam, 1992. Elsevier Science Publishers B.V.
9. L. Eshelman. The CHC adaptive search algorithm. In G.J.E. Rawlins, editor, *Foundations of Genetic Algorithms I*, pages 265–283, San Mateo CA, 1991. Morgan Kauffman.
10. T.C. Fogarty. Varying the probability of mutation in the genetic algoritm. In J.D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 104–109, San Mateo, CA, 1989. Morgan Kaufmann.
11. D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, 1989.
12. F.S. Hillier and G.J. Lieberman. *Introduction to Operations Research*. Holden-Day, San Francisco CA, 1967.
13. B. Manderick, M. de Weger, and P. Spiessens. The genetic algorithm and the structure of the fitness landscape. In R.K. Belew and L.B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 143–150, San Mateo CA, 1991. Morgan Kaufmann.
14. N.J. Radcliffe. Equivalence class analysis of genetic algorithms. *Complex Systems*, 5:183–205, 1991.
15. N.J. Radcliffe. The algebra of genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 10:339–384, 1994.
16. A. Rosete, A. Ochoa, and M. Sebag. A comparison of schema-processing algorithms. In *Proceedings of the Second International Symposium on Artificial Intelligence*, pages 22–26, La Habana, 1999.
17. P. Spiessens and B. Manderick. A massively parallel genetic algorithm. In R.K. Belew and L.B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 279–286, San Mateo CA, 1991. Morgan Kauffman.
18. G. Syswerda. A study of reproduction in generational and steady-state genetic algorithms. In G.J.E. Rawlins, editor, *Foundations of Genetic Algorithms I*, pages 94–101, San Mateo, CA, 1991. Morgan Kaufmann.
19. R. Tanese. Distributed genetic algorithms. In J.D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 434–439, San Mateo, CA, 1989. Morgan Kaufmann.
20. D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.