Multiparent Recombination in Evolutionary Computing

A.E. Eiben

Free University Amsterdam, The Netherlands
gusz@cs.vu.nl

Abstract. This chapter considers multiparent reproduction, where more than two parents are involved in creating offspring. First we give a survey of multiparent operators that have been introduced over the years in evolutionary computing and we reformulate the traditional mutation-or-crossover debate in the light of such operators. Second, we present some existing results on the usefulness of multiparent operators. We conclude the chapter with a look at future developments and some suggestions for further research.

1 Introduction

Despite the great diversity of life on Earth natural reproduction mechanisms work exclusively with one (asexual reproduction) or two parents (sexual reproduction). The majority of the species reproduce in an asexual manner, showing the viability of asexual reproduction. However, species that are higher in the evolutionary hierarchy use sexual reproduction, suggesting that sexual reproduction is more advanced. In simulated evolution, that is in evolutionary computation, many technical features are inspired by natural mechanisms. In particular, abstract variants of sexual and asexual reproduction are implemented as search operators. Some evolutionary techniques, e.g. evolutionary programming, have worked almost exclusively with mutation, i.e. they implement a simplification of asexual reproduction. Others, e.g. genetic algorithms, evolution strategies, and genetic programming, use recombination, i.e. they implement a simplification of sexual reproduction and mutation. There are several papers investigating the advantages and disadvantages of mutation with respect to crossover [20–22, 25, 37, 42]. At the moment the question of which of mutation or crossover is preferable under certain circumstances is still an open research issue.

The number of parents involved in reproduction can be technically expressed as the arity of the reproduction operators. Mutation and crossover have arity 1 and 2, respectively, and the question is whether unary or binary operators are preferable for typical optimization problems. From a purely technical point of view there is no need to restrict the arity of reproduction operators to one or two. In general, a reproduction operator can have an arity from one up to the population size. Thus the analogy with natural evolution breaks down; to our knowledge there are no species on Earth that would apply multiparent reproduction mechanisms where genetic material of more than two parents is mixed in *one* reproductive action. Simulating such reproduction operators, however, is no problem.

The objectives of this chapter are twofold.

1. To give an overview of multiparent operators that have been introduced over the years in evolutionary computing.
2. To present existing results on the performance of multiparent operators based on the author's own (co-authored) research papers.

The structure of the chapter is as follows. In Section 2, we discuss the vocabulary. A survey of multiparent operators from the literature is given in Section 3. Here we also briefly summarize the experimental or theoretical results concerning these operators. Section 4 contains the experimental results on the performance of multiparent reproduction mechanisms from the author's own work. The chapter is concluded by Section 5.

2 Terminology

Let us start by setting some conventions on terminology. The term *population* is used for a multiset of individuals that undergoes selection and reproduction. This terminology is maintained in genetic algorithms (GAs), evolutionary programming (EP), and genetic programming (GP), but in evolution strategies (ES) all μ individuals in a (μ, λ) or $(\mu + \lambda)$ strategy are called parents. In this chapter, however, the term *parents* is only used for those individuals that are selected to undergo recombination. That is, parents are those individuals that are actually used as inputs for a recombination operator; the *arity of a recombination operator* is the number of parents it uses. An individual is called a *donor* if it is a parent that actually contributes to (at least one of) the alleles of (at least one of) the child(ren) created by the recombination operator. This contribution can be, for instance, the delivery of an allele, as in uniform crossover in bitstring GAs, of participating in an averaging operation, as in intermediate recombination in ES. As an illustration, consider a steady-state GA where 100 individuals form the population and two of them are chosen to undergo uniform crossover to create one single child. These two individuals are then called parents. If, furthermore, by pure chance, the child only inherits alleles from parent 1, then parent 1 is a donor, and parent 2 is not.

3 Multiparent reproduction operators

Let us start this overview with an observation that could be the conclusion: there are more multiparent recombination mechanisms mentioned in the literature than one[1] would expect. In this section, these operators are surveyed more or less chronologically. After going through the survey some features emerge

[1] Definitely more than I expected before I started to dig in the literature, and I dare to bet that most readers will be also surprised in seeing so many multiparent operators.

that allow a systematic categorization of these operators. For didactical reasons, these features and the resulting categorization scheme will be discussed in the concluding section.

An early paper from 1966 mentioning multiparent recombination is [8] on solving systems of linear equations. It presents the definition of three different multiparent recombination mechanisms, called *majority mating*, *mating by crossing over*, and *mating by averaging*. All of these oparators are defined in a general way; that is, they can be applied to any number of m parents. Unfortunately, only very little was reported on the performance of these operators. Almost as early is the recombination mechanism of [29] for evolving models for a given process utilizing four models to create a new one.

Global recombination in evolution strategies has been known since the 1970s. It allows the use of more than two recombinants, [2, 39], because the two donors are drawn randomly for each position (gene) of the offspring anew. These drawings take the whole population of μ individuals into consideration. The multiparent character of global recombination is thus the consequence of redrawing the donors; therefore, possibly more than two individuals contribute to the offspring, but their number is not defined. It is clear that investigations on the effects of different numbers of parents on algorithm performance could not be performed in the traditional ES framework. The option of using multiple parents can be turned on or off, i.e. global recombination can be used or not, but the arity of the recombination operator is not tunable. Experimental studies on global versus two-parent recombination are possible, but so far there are almost no experimental results available on this subject. In [39] it is noted that "appreciable acceleration" is obtained by changing to a bisexual from an asexual scheme (i.e. adding recombination using two parents to the mutation-only algorithm), but only a "slight further increase" is obtained when changing from bisexual to multisexual recombination (i.e. using global recombination instead of the two-parent variant). Let us note that the terms bisexual and multisexual are not appropriate: individuals have no gender or sex, and recombination can be applied to any combination of individuals.

The mechanism called stochastic iterated genetic hill-climbing (SIGH) applies a sophisticated probabilistic voting mechanism, where m "voters" (m being the size of the population) determine the values of a new bitstring, [1]. SIGH was shown to be better than a GA with one-point and uniform crossover on four out of six test functions and the overall conclusion was that it is "competitive in speed with a variety of existing algorithms".

The *p-sexual voting recombination* from [32] is applied for the quadratic assignment problem. The operator produces one child of p parents. Let us remark again that the name p-sexual is somewhat misleading, as there are no different genders and no restriction on having one representative of each gender for recombination. In the experiments it "worked surprisingly well", but comparison between this scheme and the usual two-parent recombination was not performed.

An interesting attempt to combine GAs with the simplex method in [5] resulted in the ternary *simplex crossover*. The simplex GA performed better than the standard GA on the De Jong functions. This idea has also been extended to a version with $n+1$ parents, where n is the dimensionality of the space, [6, 35].

Uniform crossover with two as well as with three parents in a GA using an integer representation is compared on the problem of placing actuators on space structures in [24]. Based on the experimental results the authors conclude that the use of three parents did not improve the performance.

Scanning crossover has been introduced in [14] as a generalization and extension of uniform crossover in GAs creating one child from r parents. The name is based on the following general procedure scanning parents and thus building the child from left to right. Let $x^1, \ldots, x^r$ be the selected parents of length L and let x denote the child.

```
Procedure scanning
 begin
  INITIALIZE (put markers at the 1st position in each parent)
  for i = 1 to i = L
    CHOOSE j from 1, ...,r
    let i-th allele of x be the i-th allele of parent j
    UPDATE position markers
  end
```

The above procedure provides a general framework for a certain style of multiparent recombination, where the precise execution, hence the exact definition of the operator, depends on the mechanisms CHOOSE and UPDATE. In the simplest case, UPDATE can shift the markers one position to the right (thus *scanning* the chromosomes from left to right). This is appropriate for bitstring, integer, and floating-point representations. Scanning can also be easily adapted to order-based representation, where each individual is a permutation, if UPDATE shifts to the first allele which is not yet in the child. This guarantees that each offspring will be a permutation, if the parents are permutations themselves. Depending on the mechanism to CHOOSE a parent (and thereby an allele) there are three different versions of scanning. The choice can be deterministic, choosing the allele with the highest number of occurrences, and breaking ties randomly (*occurrence-based scanning*). Alternatively it can be random, either unbiased, following a uniform distribution thus giving each parent an equal chance to deliver its allele (*uniform scanning*), or biased by the fitness of the parents, where the chance of being chosen is proportional to fitness (*fitness-based scanning*). Uniform scanning for $r = 2$ is the same as uniform crossover, although creating only one child, and the occurrence-based version is very much like the voting or majority mating mechanism discussed before. *Diagonal crossover* has been introduced in [14] as a generalization of one-point crossover in GAs. In its original form, diagonal crossover creates r children from r parents by selecting $(r-1)$ crossover points in the parents and composing the children by taking the resulting r chromosome segments from the parents "along the diagonals". Later on, a one-child version was introduced and used, [47]. Figure **??** illustrates both variants. It is easy to see that for $r = 2$ diagonal crossover coincides with one-point crossover, and in some sense it also generalizes traditional two-parent n-point crossover.

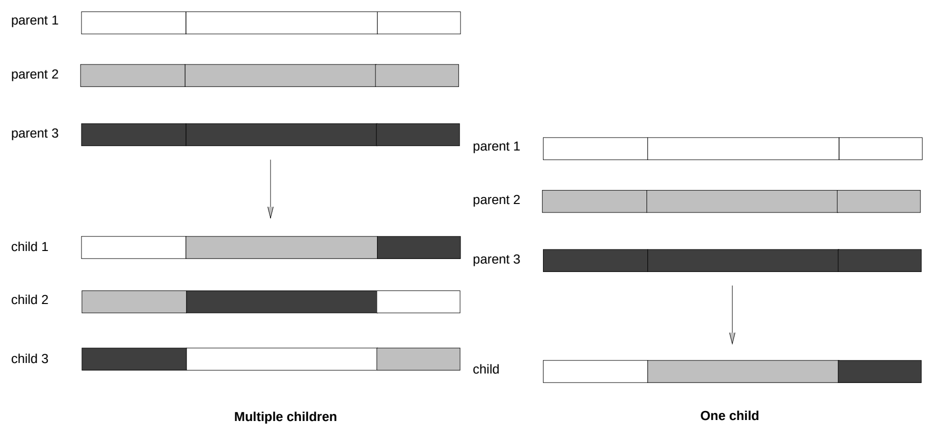

Fig. 1. Diagonal crossover (left) and its one-child version (right) for 3 parents

A so-called *triadic crossover* is introduced and tested in [34] for a multimodal spin-lattice problem. The triadic crossover is defined in terms of two parents and one extra individual for creating a child, but technically the result is identical to the outcome of a voting crossover on these three individuals as parents. A comparison between triadic, one-point, and uniform crossover is done, where triadic crossover turned out to deliver the best results.

Gene-pool recombination (GPR) and its variants were introduced in [48, 33] as a multiparent recombination mechanism for discrete domains. It is defined as a generalization of two-parent recombination (TPR). Applying GPR is preceded by selecting a genepool consisting of would-be parents. Applying GPR two parent alleles are recombined to form the allele of an offspring, and the two parents are drawn from the gene pool. Similar to global recombination in ES, the arity of the operator is not defined in advance. GPR is shown to converge about 25% faster than TPR for ONEMAX, and its extension to continuous domains outperforms the corresponding two-parent operator on the spherical function.

A recombination mechanism with tunable arity in ES is proposed in [40]. The $(\mu, \kappa, \lambda, \rho)$-ES provides the possibility of freely adjusting the number of parents (called ancestors). The parameter ρ stands for the number of parents and global recombination is redefined for any given set of ρ parents. The discrete version chooses one of the parent alleles randomly, while the intermediate version takes the average of all parent alleles as the allele of the child. Observe that ρ-ary discrete recombination coincides with uniform scanning crossover, while ρ-ary intermediate recombination is a special case of mating by averaging. At this time there are no experimental results available on the effect of ρ within this framework. Related work in ES also uses ρ as the number of parents as an independent parameter for recombination, [7]. For the purposes of a theoretical analysis it is assumed that all parents are different, uniform randomly chosen from the population of μ individuals, and ρ-ary intermediate and discrete recombinations are defined similarly to [40]. Investigations are limited to the special case of $\rho = \mu$ on the spherical function. By this assumption it is not possible to draw conclusions on the effect of ρ, but the analysis shows that the optimal progress rate is a factor μ higher than that of the traditional (μ, λ)-ES, for both recombination mechanisms.

A very particular mechanism is the *linkage evolving genetic operator* (LEGO) as defined in [41]. The mechanism is designed to detect and propagate blocks of corresponding genes of potentially varying length during the evolution. Although the multiparent feature is only a side effect, LEGO is a mechanism where more than two parents can contribute to an offspring.

A recent generalization of global intermediate recombination in ES can be found in [11]. The new operator is applied after selecting ρ parent individuals from the population of μ, and the resampling of the two donors for each i takes only these ρ individuals into consideration. In this way, the original mechanism is kept as intact as possible, while a gradual variation between the two extremes $\rho = 2$ and $\rho = \mu$ is facilitated. Note that this operator differs from the ones defined in [7, 40].

The paper [45] presents no less than three multiparent operators for real-coded GAs: the *center of mass crossover* (CMX), which was introduced originally in [44], *multiparent feature-wise crossover* (MFX), and the *seed crossover* (SX). All three operators are obtained by using a so-called base operator with arity 2 (the authors use BLX-α from [19]) and a generalization template that makes an m-ary operator from a binary one. In this respect, the paper goes much further than others, because the templates to create a multiparent operator from a BLX-α operator are general. That is, the CMX template can be applied to any operator for real-coded chromosomes, and the other two can even be applied to operators for other representations. The experimental results with these operators indicate that "more than two parents lead to better performance", although there are differences between the operators and the results also depend on the test function in question.

The so-called simplex crossover (SXP) in [46] is also designed for real-coded GAs and is shown to deliver its best performance for three or four parents on the investigated set of test functions.

4 Some experimental results on higher operator arities

As the previous section clearly shows, quite a few papers have studied the effect of operator arity on EA performance. Here we give a more detailed treatment of those two operators that generalize the most commonly applied crossovers in GAs: that is, we will focus on diagonal crossover (generalizing one-point crossover) and scanning crossover (that generalizes uniform crossover).

Most experimental research papers consider numerical optimization problmes as test functions. In the evolutionary computation literature there are a number of such functions that are used for performance assesment of novel agorithm variants. We refer to these functions by their common names and omit the exact formulas that can be retrieved from the referred articles. Extensive treatment of such test functions can be found in [3, 39].

The performance of scanning crossover for different numbers of parents is studied in [14] in a generational GA with proportional selection. A canonical GA with bit-representation is applied for function optimization (De Jong functions F1–F4 and a function from Michalewicz) and an order-based GA for graph coloring and the traveling salesman problem (TSP). Different mechanisms to CHOOSE in the general scanning procedure are tested and compared. For the function optimization problems the number of generations needed to reach a solution is used as a performance measure. For the graph coloring problem, the percentage of runs that found a solution forms the basis of comparison, while for the TSP the length of the best tour found is used. On the numerical optimization test suite, more parents perform better than two; for the TSP and graph coloring two parents turn out to be advisable. Comparing different biases in choosing the child allele, on four out of the five numerical problems fitness-based scanning outperforms the other two and occurrence-based scanning is the worst operator. In this paper only the definition of diagonal crossover is given, there are no experiments with this operator.

Diagonal crossover is investigated in [13]. It is compared to the classical two-parent n-point crossover and uniform scanning in a steady-state GA with linear ranked-biased selection ($b = 1.2$) and worst-fitness deletion. The test suite consists of two two-dimensional problems (De Jong's F2 and a function from Michalewicz) and four scalable functions (after Ackley, Griewangk, Rastrigin, and Schwefel). When monitoring the performance two different measures were used, namely efficiency (speed) and success rate (percentage of cases when an optimum was found). Speed was measured by the total number of function evaluations (averaged over all runs). The performance of diagonal crossover using r parents and n-point crossover (for two parents) showed a significant correspondence with r, respectively n. The best performance was always obtained with high values, between 10 and 15, where 15 was the maximum tested. Besides, on all problems diagonal crossover is better than n-point crossover using the same number of crossover points ($r = n - 1$), thus representing the same level of disruptiveness. For illustration we present the optimal number of parents and the corresponding success rates in Table 1. An interesting observation in [13] is that for scanning the relation between r and performance is less clear than for diagonal crossover, although the best performance is achieved for more than two parents on five out of the six test functions. A concise overview of all experiments in this study can be found in [17].

The interaction between selection pressure and the parameters r for diagonal crossover, respectively n for n-point crossover, is investigated in [47]. A steady-state GA with tournament selection (tournament size between 1 and 6) combined

Test function	Scanning Xover #par	Scanning Xover succ.	Diagonal Xover #par	Diagonal Xover succ.	N-point Xover #Xover points	N-point Xover succ.
F2	7	.91 (.73)	11	.88 (.38)	11	.84
Ackl	8	.90 (.84)	15	.89 (.00)	10	.24
Grie	10	.48 (.22)	14	.32 (.04)	10	.15
Mic	10	.72 (.57)	15	.76 (.34)	15	.6
Ras	5	.10 (.00)	13	.28 (.00)	15	.06
Schw	2	.02 (.02)	15	.24 (.00)	10	.1

Table 1. Optimal number of parents and corresponding success rates. The brackets contain the results for two parents.

with random deletion and worst-fitness deletion is applied to the Griewangk and the Schwefel functions. The disruptiveness of both operators increases in parallel as the values for r and n are raised, but the experiments show that diagonal crossover consistently outperforms n-point crossover. The best option proves to be low selection pressure and high r in diagonal crossover combined with worst-fitness deletion.

The aforementioned studies have given sufficient indication that using multiparent operators with higher arities can increase GA performance. Such an increase, however, does not always occur, and the most recent studies shifted the focus of attention from showing that it occurs to investigating when it occurs. Such an analysis implicitly assumes a relationship between the characteristics of the test functions and the observed GA behavior. Unfortunately, it is very difficult to characterize the commonly used test functions.

Motivated by the difficulties of characterizing the shapes of numerical objective functions, the effects of operator arity are studied on fitness landscapes with controllable ruggedness in [15]. The NK-landscapes of Kauffman [28], where the level of epistasis, hence the ruggedness of the landscape, can be tuned by the parameter K, are used for this purpose. The multiple-children and the one-child versions of diagonal crossover and uniform scanning are tested within a steady-state GA with linear ranked-biased selection ($b = 1.2$) and worst-fitness deletion for $N = 100$ and different values of K. Two kinds of epistatic interactions, nearest neighbor interaction (NNI) and random neighbor interactions (RNI), are considered. As the NK-landscapes are generated randomly, the exact optimum is not known for any combination of N and K. This study uses the "practical global optimum" (being the highest value ever found during the tests) as the basis of performance comparison. The quality of a particular algorithm variant is evaluated by the distance to this practical global maximum which is computed by

$$\Delta = \frac{f_{maximal} - f_{obtained}}{f_{maximal}}$$

where $f_{obtained}$ is the best value found by the given variant. Similar to earlier findings, [13], the tests show that the performance of uniform scanning cannot be related to the number of parents. The two versions of diagonal crossover behave identically, and for both operators there is a consequent improvement with increasing r. However, as the epistasis (ruggedness of the landscape) grows from $K = 1$ to $K = 5$ the advantage of more parents becomes smaller. On landscapes with significantly high epistasis ($K = 25$) the relationship between operator arity and algorithm performance seems to diminish; see Figure 2 for example, where the error at termination (Δ) is shown for nearest neighbor interaction. The final conclusions of this investigation can be very well related to works of [37, 20] and [25] on the usefulness of (two-parent) recombination. It seems that if and when crossover is useful, i.e. on mildly epistatic problems, then multiparent crossover is more useful than the two-parent variants.

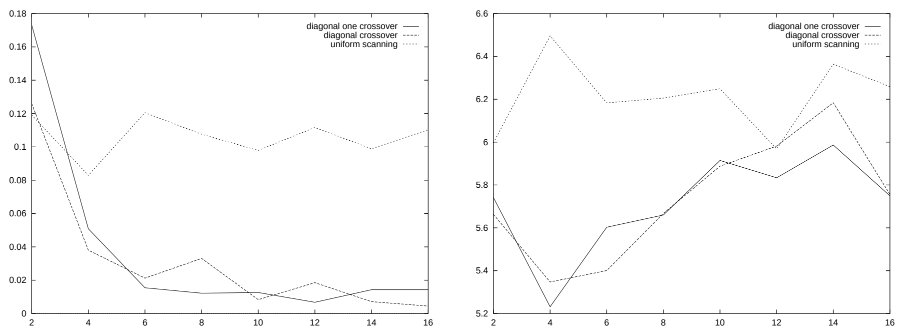

Fig. 2. Effect of the number of parents (horizontal axis) on Δ (vertical axis) on NK-landscapes with nearest neighbor interaction, $N = 100$, $K = 1$ (left), $K = 25$ (right)

A recent investigation analyzes diagonal crossover in detail on numerical optimization problems pursuing two research objectives, [18]. First, trying to find connections between the structure of the fitness landscape and the performance of diagonal crossover, i.e. establishing on what kinds of landscapes it is advantageous to increase the number of parents. Second, trying to disclose the source of increased performance of the diagonal crossover with more parents if and when it is superior to two-parent recombination. As for the first goal, the functions in the applied test suite are characterized by their modality, separability, and the arrangement of local otpima. Regarding the second goal, a number of working hypotheses are formed based on two observations. It is observed that the increase in the number of parents in diagonal crossover automatically leads to an increased number of crossover points. It can be the case that higher performance is not the result of using more parents, but simply comes from being more disruptive by using more crossover points. Furthermore, it is noticed that when applying diagonal crossover, r parents create r children in one go. In a steady-state GA the population is updated, i.e. offspring are inserted, after each application of crossover (followed by mutation), which means that a GA using 10-parent diagonal crossover has more information before performing the selection step than a GA using the two-parent version. In other words, GAs with higher operator arity have a bigger generational gap which might cause a bias in their favor. Based on these observations the following working hypotheses are made.

Hypothesis 1: using more crossover points leads to better performance.
Hypothesis 2: bigger generational gap leads to better performance.
Hypothesis 3: using more parents leads to better performance.

These hypotheses are tested by using a steady-state GA with uniform random parent selection (!) and worst-fitness deletion on eight different numerical optimization test functions. Performance is measured by accuracy (error at termination), speed (median number of fitness evaluations before termination), and success rate, if the first two measures are inconlusive. Unfortunately, the outcomes do not provide a sufficient basis for well-grounded conclusions on the relationship between the structure of the fitness landscape and the performance of diagonal crossover. A surprising result is that on Rosenbrock's saddle (De Jong's F2) increasing r decreases the performance. This function is low-dimensional ($n = 2$), unimodal, and non-separable, but none of the other functions with these features has led to such behavior. As for the working hypotheses, Hypothesis 2 is clearly rejected by the similar behavior of the one-child, respectively original, variant of diagonal crossover. Hypothesis 1 is supported by the increasing performance of N-point crossover if N is increased. This, however, does not imply rejection of Hypothesis 3, i.e. that better performance for higher N's would *only* come from having more crossover points. In fact diagonal crossover was better than N-point crossover on all but two functions: on Rosenbrock's saddle (F2) and on the Fletcher–Powell function (multimodal, non-separable, with a random arrangement of local optima). On the Fletcher–Powell function, diagonal crossover was very similar to N-point crossover, indicating that increased performance for higher r's and N's seems to be the result of crossovers' effect as macro mutation, which effect is intensified by more crossover points. An illustration is given in Figure 3.

The working of multiparent recombination operators in continuous search spaces, in particular within ES is investigated in [11]. This study compares ρ-ary global intermediate recombination as defined at the end of Section 2, ρ-ary discrete recombination, which is identical to uniform scanning crossover, and diagonal crossover with one child. The main working hypothesis is that increasing the number of parents leads to increased EA performance in terms of achieved accuracy, i.e. distance from the global optimum at termination. This is divided into two subhypotheses: A) increasing the number of parents from one to two leads to increased EA performance; B) increasing the number of parents from two to larger numbers leads to increased EA performance. Note that subhypothesis A amounts to hypothesizing that recombination and mutation work better than mutation alone. Experiments are performed on unimodal landscapes (sphere model and Schwefel's double sum), multimodal functions with regularly arranged optima, and a superimposed unimodal topology (Ackley, Griewangk,

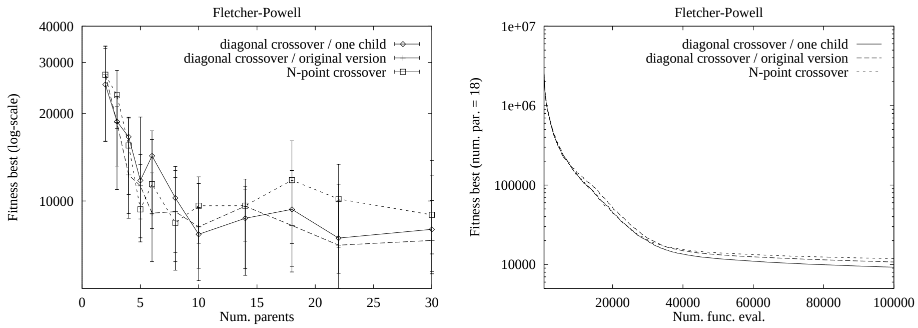

Fig. 3. Fletcher–Powell function. Left: effect of the number of parents, resp. crossover points, on accuracy. Right: population's best fitness during a run as a function of time (number of fitness evaluations), for $r = 18$, $N = 17$.

and Rastrigin functions). Furthermore, two functions with an irregular, random arrangement of local optima are studied, the Fletcher–Powell function and the Langerman function. A classification of the investigated fitness landscapes is presented in order to find the relationships between fitness landscape characteristics and performance of operators, see Table 2.

SEPARABILITY	MODALITY	
	UNIMODAL	MULTIMODAL
YES	Sphere	Rastrigin (regular)
NO	Schwefel	Ackley (regular) Griewangk (regular) Fletcher–Powell (irregular) Langerman (irregular)

Table 2. Characterization of the test functions

The results indicate that a diversity of possible outcomes can occur, and whether the working hypotheses hold or not depends on the particular combination of objective function and recombination operator. Subhypothesis A holds in more than 80% of the cases studied, but a further increase of the number of parents beyond two (subhypothesis B) does not necessarily have an advantageous effect on the accuracy achieved. In fact, there might be no significant impact, or even a negative impact; it can happen that A holds, but increasing the number of parents above two leads to worse results. Out of the 21 cases (3 operators, 7 test functions), multiparent recombination leads to a deterioration of solution quality only in 2 cases, it has no significant effect in 7 cases, and in the majority of the cases (12 out of 21 in total) it has a positive effect. These outcomes imply that, although there is no guarantee of success, it is reasonable to try multiparent recombination in an EA.

It is very interesting to consider the results on the randomly arranged fitness landscapes. With the Fletcher–Powell function there was no correlation between the number of parents and the ES performance [11], while in a genetic algorithm higher arities of diagonal crossover do lead to better performance [18]. This observation discloses that the same operator can behave differently on the same function, depending on the EA it is used in. Furthermore, the Fletcher–Powell function and the Langerman function are of the same type, non-separable and multimodal with an irregular, random arrangement of local optima. Still, the behavior of the ES is different: there is no clear effect of using more parents and the Fletcher–Powell function, while an advantage of higher arities can be observed with the Langerman function. This prevents general conclusions on ES behavior on quasirandom landscapes.

Let us note that introducing operator arity as a new parameter implies an obligation of setting its value. Since so far there are no reliable heuristics for setting this parameter, finding good values may require numerous tests. A way to circumvent this problem can be based on previous work on adapting or self-adapting the frequency of applying different operators [10, 43], or using a number of competing subpopulations [38], each applying an operator with a different arity. A first assessment of this technique can be found in [16], where subpopulations with greater progress, i.e. with more powerful operators, become larger. As a "sideeffect", this study also compares six-parent diagonal crossover and two-parent one-point crossover within the traditional one population scheme on seven different fitness landscapes that have been specifically designed for studying the effect of crossover. On all of these landscapes (Onemax, Plateau, Plateau-d, Trap, Trap-d from [37], Twin Peaks from [20], and Royal Road from [30]) the multiparent operator is superior. As an illustration we present in Table 3 the mean best fitness values found and the number of fitness evaluations needed to find an optimum for these two operators.

	six-parent diagonal crossover		one-point crossover	
Problem	mean best	no. of evals	mean best	no. of evals
Onemax	100.0	3095	100.0	5691
Twin Peaks	100.0	3839	100.0	6021
Plateau	100.0	8060	97.6	25021
Plateau-d	99.2	23479	93.9	35063
Trap	189.8	3701	168.3	-
Trap-d	138.8	-	136.7	-
Royal Road	595.8	10228	480.2	30050

Table 3. Diagonal crossover with six parents vs. one-point crossover. Note: all problems are to be maximized; the number of fitness evaluations is undefined (-) if no runs found the optimum.

5 Conclusions

Recombination operators using more than two parents in an evolutionary problem solver have been repeatedly (re)introduced in evolutionary computation. However, there is still much systematic investigation needed to establish the effects of operator arity on EA performance. Of course, it can be questioned whether multiparent recombination is a single phenomenon showing one behavioral pattern. The present overview shows that there are (at least) two features for grouping multiparent operators. That is, multiparent recombination operators can be distinguished by their:

- Arity, for instance
  - fixed arity, e.g. the triadic crossover in [34],
  - variable arity that is tunable (can be set between 2 and X), e.g. the scanning crossover in [14],
  - variable arity that is undefined (a random number), e.g. global recombination in evolution strategies [2].
- Type, for instance
  - based on allele frequencies, e.g. the p-sexual voting from [32],
  - based on segmentation and recombination of the parents, e.g. the diagonal crossover in [14],
  - based on numerical operations on (real-valued) alleles like averaging, e.g. the center of mass crossover, in [44].

In general, it cannot be expected that operators in different classes show the same behavior. There are also experimental results supporting differentiation among various multiparent mechanisms. For instance, there seems to be no clear relationship between the number of parents and the performance of uniform scanning crossover, while the opposite is true for diagonal crossover [15].

Studies on multiparent operators also have to consider possibly different behavior on different types of fitness landscapes. So far, there are insufficient experimental data to support general conclusions. Some studies indicate that on irregular landscapes, such as NK-landscapes with relatively high K values [15] or the Fletcher–Powell function [11], they do not work. There are also results indicating the opposite on the Fletcher–Powell function [18], and on the Langerman function [11]. This stresses the importance of more experimental and theoretical research, following the tradition of studying the (dis)advantages of two-parent crossovers under different circumstances, [20, 25, 37, 42]. In particular, there is a need for a better vocabulary to characterize objetcive functions or fitness landscapes. Such a vocabulary need not necessarily be universal, but applicable in EC research. That is, the function/landscape categories distinguishable by this vocabulary should coincide with the behavioral patterns of EAs. In fact, this is a methodological issue for experimental EC research concerning (the lack of) meaningful problem classes that can support generalizable experimental results [12]. Strongly related to this issue is the lack of a (general) theory that would illuminate why and when multiparent recombination works better than two-parent recombination.

One of the most inspiring approaches to multiparent reproduction is represented by the generalization templates in [45]. Although the authors do not elaborate on this aspect, their definitions can be applied to any two-parent operator matching the representation. On the one hand, this provides a technique to define new multiparent operators. On the other hand, it opens up a new trajectory for research, that of studying generalization templates as meta-operators acting on operators.

Summarizing this overview, let us note the following. While there are no biological analogies of multiparent recombination, computer simulations have already provided substantial evidence that applying more than two parents can "boost" evolution. Thus, although there is still much to be investigated, multiparent recombination mechanisms form a promising design heuristics for practitioners and a challenge for theoretical analysis.

Acknowledgments

I am grateful for the help of all those colleagues who contributed to this chapter, as co-authors of experimental research papers, or by exchanging ideas. Special thanks go to Thomas Bäck and Cees van Kemenade.

References

1. D.H. Ackley. An empirical study of bit vector function optimization. In L. Davis, editor, *Genetic Algorithms and Simulated Annealing*, pages 170–215. Morgan Kaufmann, 1987.
2. T. Bäck. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, New York, 1996.
3. T. Bäck and Z. Michalewicz. Test landscapes. In T. Bäck, D. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, pages B2.7:14–B2.7:20. Institute of Physics Publishing, Bristol, and Oxford University Press, New York, 1997.
4. R.K. Belew and L.B. Booker, editors. *Proceedings of the 4th International Conference on Genetic Algorithms.* Morgan Kaufmann, San Francisco, 1991.
5. H. Bersini and G. Seront. In search of a good evolution-optimization crossover. In R. Männer and B. Manderick, editors, *Proceedings of the 2nd Conference on Parallel Problem Solving from Nature*, pages 479–488. North-Holland, Amsterdam, 1992.
6. H. Bersini and F.J. Varela. The immune recruitment mechanism: A selective evolutionary strategy. In Belew and Booker [4], pages 520–526.
7. H.-G. Beyer. Toward a theory of evolution strategies: On the benefits of sex- the $(\mu/\mu, \lambda)$ theory. *Evolutionary Computation*, 3(1):81–111, 1995.
8. H.J. Bremermann, M. Rogson, and S. Salaff. Global properties of evolution processes. In H.H. Pattee, E.A. Edlsack, L. Fein, and A.B. Callahan, editors, *Natural Automata and Useful Simulations*, pages 3–41. Spartan Books, Washington DC, 1966.
9. Y. Davidor, H.-P. Schwefel, and R. Männer, editors. *Proceedings of the 3rd Conference on Parallel Problem Solving from Nature*, number 866 in Lecture Notes in Computer Science. Springer, Berlin, 1994.

10. L. Davis. Adapting operator probabilities in genetic algorithms. In Schaffer [36], pages 61–69.
11. A.E. Eiben and Th. Bäck. An empirical investigation of multi-parent recombination operators in evolution strategies. *Evolutionary Computation*, 5(3):347–365, 1997.
12. A.E. Eiben and M. Jelasity. A critical note on experimental research methodology in EC. In *2002 Congress on Evolutionary Computation (CEC'2002)*, pages 582–587. IEEE Press, Piscataway, NJ, 2002.
13. A.E. Eiben, C.H.M. van Kemenade, and J.N. Kok. Orgy in the computer: Multi-parent reproduction in genetic algorithms. In Morán et al. [31], pages 934–945.
14. A.E. Eiben, P-E. Raué, and Zs. Ruttkay. Genetic algorithms with multi-parent recombination. In Davidor et al. [9], pages 78–87.
15. A.E. Eiben and C.A. Schippers. Multi-parent's niche: n-ary crossovers on NK-landscapes. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Proceedings of the 4th Conference on Parallel Problem Solving from Nature*, number 1141 in Lecture Notes in Computer Science, pages 319–328. Springer, Berlin, 1996.
16. A.E. Eiben, I.G. Sprinkhuizen-Kuyper, and B.A. Thijssen. Competing crossovers in an adaptive GA framework. In *Proceedings of the 5th IEEE Conference on Evolutionary Computation*, pages 787–792. IEEE Press, 1998.
17. A.E. Eiben and C.H.M. van Kemenade. Performance of multi-parent crossover operators on numerical function optimization problems. Technical Report TR-95-33, Leiden University, 1995. available from http://www.liacs.nl/TechRep/1995/.
18. A.E. Eiben and C.H.M. van Kemenade. Diagonal crossover in genetic algorithms for numerical optimization. *Journal of Control and Cybernetics*, 26(3):447–465, 1997.
19. L.J. Eshelman, K.E. Mathias, and J.D. Schaffer. Crossover operator biases: Exploiting the population distribution. In Th. Bäck, editor, *Proceedings of the 7th International Conference on Genetic Algorithms*, pages 354–361. Morgan Kaufmann, San Francisco, 1997.
20. L.J. Eshelman and J.D. Schaffer. Crossover's niche. In Forrest [23], pages 9–14.
21. D.B. Fogel and J.W. Atmar. Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear systems. *Biological Cybernetics*, 63:111–114, 1990.
22. D.B. Fogel and L.C. Stayton. On the effectiveness of crossover in simulated evolutionary optimization. *Biosystems*, 32:171–182, 1994.
23. S. Forrest, editor. *Proceedings of the 5th International Conference on Genetic Algorithms*. Morgan Kaufmann, San Francisco, 1993.
24. H. Furuya and R.T. Haftka. Genetic algorithms for placing actuators on space structures. In Forrest [23], pages 536–542.
25. W. Hordijk and B. Manderick. The usefulness of recombination. In Morán et al. [31], pages 908–919.
26. *Proceedings of the 1995 IEEE Conference on Evolutionary Computation*. IEEE Press, Piscataway, NJ, 1995.
27. *Proceedings of the 1996 IEEE Conference on Evolutionary Computation*. IEEE Press, Piscataway, NJ, 1996.
28. S.A. Kauffman. *Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, New York, NY, 1993.
29. H. Kaufman. An experimental investigation of process identification by competitive evolution. *IEEE Transactions on Systems Science and Cybernetics*, SSC-3(1):11–16, 1967.

30. M. Mitchell, S. Forrest, and J.H. Holland. The royal road for genetic algorithms: Fitness landscapes and GA performance. In F.J. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the 1st European Conference on Artificial Life*, pages 245–254. The MIT Press, Cambridge, MA, 1994.
31. F. Morán, A. Moreno, J. J. Merelo, and P. Chacón, editors. *Advances in Artificial Life. Third International Conference on Artificial Life*, volume 929 of *Lecture Notes in Artificial Intelligence*. Springer, Berlin, 1995.
32. H. Mühlenbein. Parallel genetic algorithms, population genetics and combinatorial optimization. In Schaffer [36], pages 416–421.
33. H. Mühlenbein and H.-M. Voigt. Gene pool recombination in genetic algorithms. In I.H. Osman and J.P. Kelly, editors, *Meta-Heuristics: Theory and Applications*, pages 53–62, Boston, London, Dordrecht, 1996. Kluwer Academic Publishers.
34. K.F. Pál. Selection schemes with spatial isolation for genetic optimization. In Davidor et al. [9], pages 170–179.
35. J.-M. Renders and H. Bersini. Hybridizing genetic algorithms with hill-climbing methods for global optimization: two possible ways. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 312–317. IEEE Press, Piscataway, NJ, 1994.
36. J.D. Schaffer, editor. *Proceedings of the 3rd International Conference on Genetic Algorithms*. Morgan Kaufmann, San Francisco, 1989.
37. J.D. Schaffer and L.J. Eshelman. On crossover as an evolutionary viable strategy. In Belew and Booker [4], pages 61–68.
38. D. Schlierkamp-Voosen and H. Mühlenbein. Adaptation of population sizes by competing subpopulations. In IEEE [27], pages 330–335.
39. H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, 1995.
40. H.-P. Schwefel and G. Rudolph. Contemporary evolution strategies. In Morán et al. [31], pages 893–907.
41. J. Smith and T.C. Fogarty. Recombination strategy adaptation via evolution of gene linkage. In IEEE [27], pages 826–831.
42. W.M. Spears. Crossover or mutation? In L.D. Whitley, editor, *Foundations of Genetic Algorithms - 2*, pages 221–238. Morgan Kaufmann, 1993.
43. W.M. Spears. Adapting crossover in evolutionary algorithms. In J.R. McDonnell, R.G. Reynolds, and D.B. Fogel, editors, *Proceedings of the 4th Annual Conference on Evolutionary Programming*, pages 367–384. MIT Press, 1995.
44. S. Tsutsui. Multi-parent recombination in genetic algorithms with search space boundary extension by mirroring. In A.E. Eiben, Th. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Proceedings of the 5th Conference on Parallel Problem Solving from Nature*, number 1498 in Lecture Notes in Computer Science, pages 428–437. Springer, Berlin, 1998.
45. S. Tsutsui and A. Ghosh. A study on the effect of multi-parent recombination in real coded genetic algorithms. In *Proceedings of the 1998 IEEE Conference on Evolutionary Computation*, pages 828–833. IEEE Press, Piscataway, NJ, 1998.
46. S. Tsutsui, M. Yamamura, and T. Higuchi. Multi-parent recombination with simplex crossover in real coded genetic algorithms. In W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, and R.E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-1999)*, pages 657–664. Morgan Kaufmann, 1999.
47. C.H.M. van Kemenade, J.N. Kok, and A.E. Eiben. Raising GA performance by simultaneous tuning of selective pressure and recombination disruptiveness. In IEEE [26], pages 346–351.

48. H.-M. Voigt and H. Mühlenbein. Gene pool recombination and utilization of covariances for the Breeder Genetic Algorithm. In IEEE [26], pages 172–177.