Putting the "Genetics" back into Genetic Algorithms (Reconsidering the Role of Crossover in Hybrid Operators)

Stephen Chen
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
chens@ri.cmu.edu

Stephen F. Smith
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sfs@ri.cmu.edu

Abstract

The original analysis of genetic algorithms presents combination to be the primary mechanism of crossover. Although good solutions can be found by combination, they are often not locally optimal. Thus, a popular technique is to locally optimize each crossover solution before adding it to the population. In these "hybrid" operators, crossover can be viewed as a means of restarting the local optimizer. Unfortunately, if crossover does little more than combine random parts of two parent solutions, the performance of the resulting hybrid operator may not be significantly different from random restart of the local optimizer. The design of the crossover operator affects the efficiency and effectiveness of hybrid operators. A new analysis presents preserving common schemata as an important design consideration for crossover.

Keywords: Schema Theorem, combination, implicit parallelism, commonality hypothesis, Commonality-Based Crossover Framework, heuristic operator, hybrid operator

1 Introduction

Traditionally, the design focus for crossover operators has been combination – random parts (schemata) from two parents are combined into an offspring solution. This design basis is a legacy of the original Schema Theorem. In it's development, the focus is on implicit parallelism and the ideal sampling of schemata. In particular, low-order schemata (building blocks) are important. However, the original analysis of crossover is based on a standard

representation, uses the viewpoint of a single parent, and is pre-occupied with disruption. Non-standard representations and hybrid operators were not intended extensions of the original design focus of combination.

In this paper, a new Schema Theorem is developed. The analysis of crossover is representation independent, uses the viewpoint of both parents, and ignores disruption. The new Schema Theorem suggests that common schemata are important. Instead of combining random schemata, the common schemata of two parents are used as a foundation on which a new offspring solution is built. Specifically, a new two step process is defined for crossover: 1) preserve the maximal common schema of two parents, and 2) complete the solution with a construction heuristic. The scope of this design framework extends to non-standard representations and hybrid operators.

Non-standard (sequence) representations have been previously studied [CS98], so this paper concentrates on hybrid operators. In hybrid operators, crossover solutions are locally optimized before they are added to the population. Compared with randomly restarting the local optimizer, crossover accelerates the search because it provides intermediate solutions of higher quality which can be processed in less time [UPvL+91]. However, this statement only addresses how efficiently local optimizers are used. It has also been shown that locally-optimal solutions can be more similar to each other than are random solutions, and that the better the solutions are, the more similar they can be [Müh91]. Thus, exploiting common schemata in hybrid operators may lead to the more effective use of local optimizers.

The remainder of this paper pursues this hypothesis. First, the original Schema Theorem and some of its implications are reviewed in section 2. In section 3, a new commonality-based Schema Theorem is developed. In section 4, the role of crossover in hybrid operators is examined. In section 5, the implications of the new Schema Theorem and the new experimental results are discussed. Finally, a brief summary of conclusions is given in section 6.

2 Background

Empirical analysis often provides more accurate models of a genetic algorithm's behavior than the Schema Theorem [Müh97]. However, these models have not meaningfully addressed the design influences descendant from the original Schema Theorem. Since these design influences are the focus of this paper, the theoretical review of GAs is restricted to the Schema Theorem. Further, only forms critical to our study of design influences are presented in detail.

2.1 The Schema Theorem

In his seminal work, Holland [Hol75] first studied the effects on schemata for a genetic algorithm with binary string solutions, a generational replacement scheme, roulette wheel parent selection, and one-point crossover. If at a time t there are m instances of a particular schema H in the population $\mathcal{B}(t)$, roulette wheel selection will result in an expected

$$m(H, t+1) = m(H, t) \cdot \frac{f(H)}{\overline{f}} \tag{1}$$

instances to be present in the next generation (at time $t+1$). No search is performed by

fitness-based selection – new trials are allocated by merely copying the solutions. Search occurs when schemata "are combined and tested in new contexts by crossing-over" [Hol75]. This process creates new solutions with a minimal amount of disruption to the allocation strategy represented in equation (1).

The expected amount of disruption caused by crossover to a schema H is influenced by three factors: the length l of the solution string, the defining length $\delta(H)$ of the schema, and the probability p_c of crossover. From the viewpoint of a single parent, the probability of survival of schema H is

$$p_s \geq 1 - p_c \cdot \frac{\delta(H)}{l-1}. \tag{2}$$

The inequality arises because even if crossover occurs at a cut-point that affects schema H, the schema is not disrupted when the other parent (selected randomly) matches at the "disrupted" sites. These sites always match if the other parent is also an instance of schema H (i.e. schema H is common to both parents). Thus, the probability of survival is better estimated by

$$p_s \geq 1 - \left[p_c \cdot \frac{\delta(H)}{l-1}\right] [1 - m(H,t)]\,. \tag{3}$$

Combining equations (1) and (3), a schema H is expected to receive

$$m(H,t+1) \geq m(H,t) \cdot \frac{f(H)}{\overline{f}} \left\{1 - \left[p_c \cdot \frac{\delta(H)}{l-1}\right] [1 - m(H,t)]\right\} \tag{4}$$

trials in the next generation of a genetic algorithm with roulette-wheel parent selection and one-point crossover.

Equation (4) is Theorem 6.2.3 in Holland's book. It is the Schema Theorem, or the Fundamental Theorem of Genetic Algorithms. "[It] provides the first evidence of the [implicit] parallelism of genetic plans. *Each* schema represented in the population $\mathcal{B}(t)$ increases or decreases according to the above formulation *independently of what is happening to other schemata* in the population." [Hol75]

2.2 Modifications to the Schema Theorem

Many modifications to the original Schema Theorem have been made. For example, the advantages of selecting both parents by fitness has been demonstrated [Sch87], the effects of n-point crossover have been analyzed [SdJ91], and the Markov model of Nix and Vose may even be considered to be an exact form of the Schema Theorem [NV92]. However, with the notable exception of Radcliffe [Rad91], crossover design concepts have not been modified.

Radcliffe has proposed three design principles: respect, transmission, and assortment [Rad91]. Respect is an equivalent notion to preserving commonality – two parents that have a schema H in common should produce an offspring that also has schema H. Transmission is similar to "non-disruption" – schemata in the offspring should come from one of the parents. Assortment is combination – it should be possible for two compatible schemata, one from each parent, to both survive in the offspring. However, previous experiments have shown that transmission and assortment can handicap the performance of heuristic operators for non-standard representations [CS98].

Although respect hints at the importance of preserving common schemata, transmission still worries about disruption. The means by which disruption will eventually be removed

from the Schema Theorem are first presented for GENITOR – a specialized steady-state replacement scheme [WS90]. Here, offspring are added one-at-a-time, and each new offspring replaces the lowest-ranking member of the population. This process builds in elitism – the best solutions always survive. Over time, the instances of a schema H can be quantified by

$$m(H,t+1) = m(H,t) - losses + gains\ from\ schema\ H + other\ gains. \tag{5}$$

The *losses* occur if an instance of schema H is removed from the population. If the removed member is selected randomly, *losses* occur with probability $P(H,t) = m(H,t)/n$, the proportion of schema H in the population at time t. However, since the least fit member of the population is removed, the probability of a loss varies inversely with the fitness ratio, $FR = f(H)/\overline{f}$. The *losses* have been estimated by $P(H,t)/FR$.

The *gains from schema H* occur if the first parent has schema H, $FR \cdot P(H,t)$, and it is not disrupted. (Note: the fitness ratio is now for rank-based selection.) Disruption can occur if the second parent (also selected based on fitness) does not have schema H, $1 - FR \cdot P(H,t)$; and the schema is either disrupted by crossover or survives in the discarded offspring, $\delta(H)/(l-1)$ or $(1/2)[1-\{\delta(H)/(l-1)\}]$. Combining the above terms, the *gains from schema H* are conservatively approximated as:

$$FR \cdot P(H,t)\left[1 - \{1 - FR \cdot P(H,t)\}\left\{\frac{\delta(H)}{l-1} + \frac{1}{2}\left[1 - \frac{\delta(H)}{l-1}\right]\right\}\right]. \tag{6}$$

The *other gains* occur if two parents, each without schema H, fortuitously create schema H through crossover. This event has been dropped as part of the inequality. Overall, the resulting Schema Theorem for one-point crossover in GENITOR is

$$m(H,t+1) \geq m(H,t) - \frac{P(H,t)}{FR} + FR \cdot P(H,t)\left[1 - \{1 - FR \cdot P(H,t)\} \cdot \frac{1}{2}\left\{1 + \frac{\delta(H)}{l-1}\right\}\right]. \tag{7}$$

(Note: the time t represents an entire generation in equation (4), but it represents a single new solution in equation (7).)

2.3 Interpretations of the Schema Theorem

The competition among schemata has been compared to a k-armed bandit problem. In the two-armed bandit problem, it is desirable to both exploit the best-observed arm and continue exploration on the other arm. To minimize the expected loss, the trials allocated to the best-observed arm should increase at slightly faster than an exponential function relative to the trials allocated to the other arm [Hol75]. Consequently, the allocation of trials to schemata should reward the best-observed schemata with a similar increase relative to their competing alternatives. The Schema Theorem has been interpreted as guaranteeing this "... exponentially increasing number of trails to the best observed building blocks." [Gol89]

Further, the Schema Theorem applies equally to all schemata. Since each solution represents 2^l schemata, information is gained on all 2^l schemata with each evaluation. Overall, it has been estimated that a population of n solutions usefully processes $O(n^3)$ schemata

each generation [Gol89]. This computational leverage (believed to be unique to genetic algorithms) has been called *implicit parallelism* [Hol75].

Implicit parallelism heavily influences the design of crossover operators. Since it is believed that the source of power in genetic algorithms comes from this parallel search for good schemata, a necessary function of crossover is combination. Combination allows offspring to be created from the components found by the (independent) parallel searches. For example, if the first parent has a desirable schema H_1, and the second parent has a desirable schema H_2, then crossover should be able to produce an offspring with both schema H_1 and H_2 (provided that schema H_1 and H_2 are compatible).

The concept of combination has received heavy re-emphasis in the literature. Reinforcing views include Syswerda, "... the next step is to combine [good schemata] together in one genome. This is, after all, the overt purpose of crossover." [Sys89], Radcliffe, "Given instances of two compatible [schemata], it should be possible to cross them to produce a child which is an instance of both [schemata]." [Rad91], and Davis, "... the benefits of crossover ... is that crossover acts to combine building blocks of good solutions from diverse chromosomes." [Dav91].

3 A Commonality-Based Schema Theorem for Genetic Algorithms

In the original development of the Schema Theorem, the analysis of crossover is based on a standard representation, uses the viewpoint of a single parent, and is pre-occupied with disruption. However, with elitist replacement schemes, disruption is superseded by exploratory power. Eliminating disruption from the Schema Theorem conveniently results in a representation-independent Schema Theorem that uses the viewpoint of both parents. This new Schema Theorem supports the Commonality-Based Crossover Framework (a new design model for crossover).

3.1 Disruption vs. Search

In the original (generational) replacement scheme, it is possible for crossover to disrupt all the good schemata of the parents and produce offspring that are less fit than their parents. However, disruption is related to exploratory power, and it has been shown that crossover operators with greater exploratory power (and higher disruption) can perform better (e.g. [Sys89][Esh91][GS94]). The negative effects of disruption are eliminated by elitism. This hypothesis was made (but not confirmed) by Syswerda who thought that his experimental results which showed uniform crossover to perform better than both one-point and two-point crossover may have been partially caused by the use of a strongly elitist steady-state replacement scheme (vs. a traditional generational replacement scheme) [Sys89].

To isolate this hypothesis, an objective function highly susceptible to disruption was designed. Assume that a solution string of 100 bits is decomposed into 25 blocks of 4 bits each, and that the fitness increases by 1 if a block is "complete" – all `1`'s or `0`'s. Further, if there are n adjacent complete blocks of `1`'s (or `0`'s), the fitness increases by n^2. There are two optimal solutions to this problem – all `1`'s or all `0`'s (fitness = 625). Crossover can cause disruption by undoing a complete block or a set of adjacent complete blocks.

In an experiment using this objective function, the effects of disruption are found to be

significantly higher for uniform crossover than for one-point or two-point crossover. The offspring of one-point and two-point crossover have the same average fitness as their parents. However, the offspring of uniform crossover have an average fitness that is only 70% of their parents[1].

When using a generational replacement scheme, one-point and two-point crossover have similar performance, but uniform crossover performs worse. Without elitism, disruption can be dangerous. However, when using a strongly elitist replacement scheme like Syswerda, uniform crossover performs better than two-point crossover which itself performs better than one-point crossover. The performance of uniform crossover is clearly enhanced by elitism. (See Table 1.)

	Standard GA	GENITOR
One-point	34.89	54.99
Two-point	37.45	81.34
Uniform	29.06	98.22

Table 1: Performance of One-point, Two-Point, and Uniform crossover on a problem with an objective highly susceptible to disruption. Results are the average of 100 runs of 100 generations each using a population size of 100 solutions.

Elitism can turn a highly disruptive crossover operator into a minimally restrictive search operator. Specifically, if the maximal common schema of two parents has n wild-card slots, the exploratory power [ECS89] of one-point crossover covers a set with $2 \cdot n$ possible offspring, two-point crossover can explore $n^2 - n$ possible offspring, and uniform crossover can explore 2^n possible offspring. Since these are supersets, the best possible offspring for two parents under uniform crossover is at least as good as that under two-point crossover which is at least as good as that under one-point crossover.

3.2 Redeveloping the Schema Theorem

If the negative effects of disruption can be eliminated by elitism, then disruption should also be eliminated from the Schema Theorem. Using equation (5) for a steady-state replacement scheme, the number of instances of schema H varies over time by the following:

$$m(H, t+1) = m(H, t) - losses + gains\ from\ schema\ H + other\ gains. \tag{8}$$

Eliminating the (crossover dependent) disruption terms from equation (6), the *gains from schema H* are

$$FR \cdot P(H,t) \left[1 - \{1 - FR \cdot P(H,t)\}\right]. \tag{9}$$

After simplifying, it can be seen that the *gains from schema H* occur if and only if both parents have schema H:

[1] In the steady-sate GA, this value is accurate for only the first few generations. After convergence, the offspring of uniform crossover also average 100% of the fitness of their parents

$$[FR \cdot P(H,t)]^2 \,. \tag{10}$$

When common schemata are preserved, the gains represented in equation (10) are completely independent of the solution representation and the crossover operator. Further, since the viewpoint of a single parent was only used to develop the (crossover dependent) disruption terms, the above analysis of crossover conveniently switches to the viewpoint of both parents.

The *losses* occur if one of the solutions with schema H is removed from the population $\mathcal{B}(t)$. The exact value of the *losses* is less important than that it varies with $P(H,t)$ and that it varies inversely with FR. The *other gains* are crossover dependent (i.e. they depend on the selected construction heuristic), so they are dropped to form an inequality. The resulting Schema Theorem for genetic algorithms is

$$m(H,t+1) \geq m(H,t) - \mathcal{F}(P(H,t), FR^{-1}) + [FR \cdot P(H,t)]^2 \,. \tag{11}$$

For genetic algorithms with an elitist steady-state replacement scheme, fitness-based selection of both parents, and crossover operators that preserve common schemata; schemata that are common to a large number of above-average solutions receive an increasing number of trials.

3.3 Implications of the New Schema Theorem

In the original development of the Schema Theorem, the effect of one-point crossover on a single parent is examined. The analysis measures the disruption and survival probabilities of various schemata. Since low-order schemata are less likely to be disrupted, it is suggested that they have an important role – they are combined. However, the negative effects of disruption can be eliminated by elitism. Further, disruptive effects are crossover dependent. The new Schema Theorem is crossover independent and representation independent. It suggests that preserving common schemata is the key attribute of crossover.

Crossover exploits the common schemata of two parents as a base to explore for better solutions. Exploration can be done randomly, with the left-over parts of the parents (i.e. combination), or with a problem-specific construction heuristic. To generalize, a commonality-based crossover operator is designed by following this general framework: 1) preserve the maximal common schema of two parents, and 2) complete the solution with a construction heuristic.

The two-parent perspective of the new crossover framework affects the concept of implicit parallelism. The allocation of trials to schemata now depends on which pairs of solutions are mated. Further, selection is not about allocating trials to competing schemata, but about identifying the (fit) common schemata from the solutions that survive the fitness-based competition. Preserving only common schemata adds a second form of commonality-based selection on top of fitness-based selection. This two-phase selection process makes the application of the k-armed bandit analogy more difficult.

4 Hybrid Genetic Algorithms

The primary benefit of the new Schema Theorem is the new design framework for crossover. This design framework appears to extend more naturally to hybrid operators than the

design focus of combination. Combination suggests that parts should be taken from existing solutions, but local optimizers may select different parts instead. Commonality suggests that search should be conducted in a region near the parents, and local optimizers can conduct this search.

4.1 Review

The design focus of combination results from the original development and application of genetic algorithms for real-valued functions defined in the $\mathcal{R}^n$ domain. In these domains, GAs have been shown to be robust optimization algorithms. However, when genetic algorithms were first applied to combinatorial optimization problems, the results were quite disappointing. To improve performance, heuristics have been embedded into crossover to select schemata (from the parents) [GGRG85], and local search methods have been appended to crossover to improve the solution before adding it to the population [SG87]. A crossover-local optimization pair can be viewed as a hybrid operator.

It has been shown that local optimizers can improve the performance of crossover [SG87]. Local optimization increases the fitness of schemata in a population by improving each solution. Conversely, it has also been shown that crossover can improve the performance of local optimizers [UPvL+91]. Crossover solutions consist of schemata from previous locally optimal solutions. The better start solutions allow the local optimizer to avoid the ineffective process of bringing each (random) start solution into the near-optimal region. However, the synergistic opportunities between crossover and local optimizers is largely unstudied.

4.2 Hybrid Operators for the Traveling Salesman Problem

In this section, the role of crossover is examined in three 2-opt-based hybrid operators for the Traveling Salesman Problem (TSP). Since 2-opt is held constant as the local optimizer, the experiments isolate the contribution provided by the crossover operator. It will be shown that the Commonality-Based Crossover Framework provides useful guidance to design more synergistic operators.

4.2.1 The Crossover Operators

Common Sub-Tours/Nearest Neighbors (CST/NN) is a commonality-based heuristic operator [CS98]. The operator preserves the Common Sub-Tours (CST), and thus all the common edges, from two parents. Starting with a random common sub-tour, it connects the remaining elements and sub-tours by using the Nearest Neighbor (NN) construction heuristic. (See Figure 1.)

Greedy Crossover (GX) is a heuristic operator that focuses on combination [GGRG85]. The version used in this paper starts at a random point and selects the shortest (undirected) edge from the parents that does not introduce a cycle. Undirected edges are used instead of directed edges as an influence from Edge Recombination [SMM+91]. If all parent edges introduce a cycle, then NN is used instead.

Random, Respectful Recombination (RRR) is a domain-independent operator [Rad91]. The operator preserves all common edges and then selects the remaining edges randomly. However, without additional heuristics, RRR fits the comment that it is "essential ... to incorporate ... local improvement operators" [SG87] to make a competitive GA.

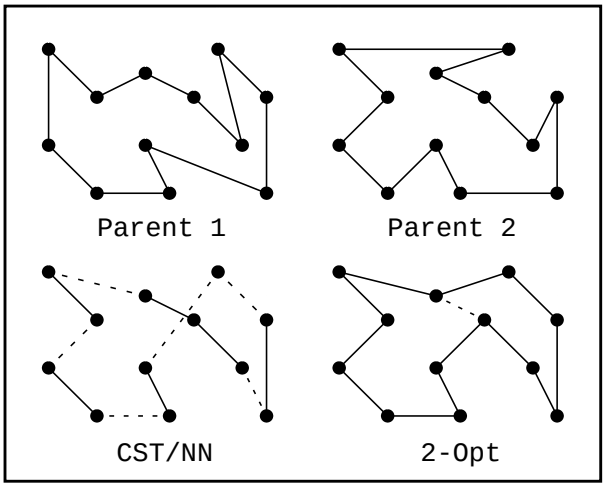

Figure 1: Example of CST/NN and CST/NN-2-Opt. One common edge is lost during 2-opt.

In previous work, it has been shown that CST/NN performs better than GX. Although both use NN, GX selects (combines) many uncommon edges which appear to handicap the performance of NN [CS98]. Without NN, RRR is uncompetitive with CST/NN and GX.

4.2.2 The Hybrid Operators

Recognizing that GX solutions often have crossing edges that 2-opt can correct, the GX-2-Opt hybrid operator was developed [SG87]. Similarly, 2-opt can be added to CST/NN to make CST/NN-2-Opt. For RRR, a 2-opt swap is the Binomial Minimal Mutation (BMM) [RS94], so RRR-BMM is also a 2-opt-based hybrid operator. (note: RRR-BMM is equivalent to the operator used in CHC [Esh91].) From previous work [CS98], the solutions found by CST/NN-2-Opt are better than those found by GX-2-Opt and RRR-BMM. (See Table 2.)

TSP	Avg. 2-opt Start	Avg. Best CST/NN-2-Opt	Kept by 2-opt	Avg. Best GX-2-Opt	Kept by 2-opt	Avg. Best RRR-BMM	Kept by 2-opt
d198	+ 3.06 %	+ 0.87 %	89.9 %	+ 1.12 %	87.1 %	+ 1.99 %	80.7 %
lin318	+ 5.16 %	+ 0.31 %	93.5 %	+ 0.77 %	92.1 %	+ 3.04 %	86.8 %
fl417	+ 2.54 %	+ 1.16 %	86.6 %	+ 1.19 %	83.2 %	+ 1.66 %	76.2 %
pcb442	+ 7.29 %	+ 1.22 %	91.1 %	+ 2.28 %	87.9 %	+ 5.61 %	77.8 %
u574	+ 8.26 %	+ 2.68 %	89.7 %	+ 3.66 %	87.5 %	+ 6.75 %	80.0 %
Avg.	+ 5.26 %	+ 1.25 %	90.2 %	+ 1.80 %	87.6 %	+ 3.81 %	80.3 %

Table 2: Comparison of CST/NN-2-Opt, GX-2-Opt, and RRR-BMM on 5 TSP instances from TSPLIB. Results are with GENITOR using a population size of 400 solutions, run until 10 generations pass without an improvement. Values are percent distance from known optimum for the average of 5 runs.

Focusing on how the initial crossover solution is used, some of the initially preserved common

edges can be disrupted by 2-opt. (See Figure 1.) However, over 90% of the common (CST) edges from the parents are kept after 2-opt during the first generation of new solutions generated by CST/NN-2-Opt. In GX-2-Opt, 2.6% more common edges are disrupted, and 9.9% more are disrupted in RRR-BMM.

The number of disrupted common edges is a measure of how far the final offspring is from its parents. A "GA should explore ... new regions [that] are not too far ... from the currently exploited regions" [MdWS91] because fitness information of the current parents does not necessarily apply to distant neighbors. CST/NN is more effective than GX and RRR at guiding 2-opt to stay near the neighborhood of good solutions as identified by the parent solutions.

The number of disrupted common edges is also a measure of how far the initial crossover solution is from its 2-opt minimum. Using this measure, the CST/NN solution is closer to its 2-opt minimum than the GX and RRR solutions. Thus, in addition to finding better solutions, the 2-opt local optimizer is also more (time) efficient when started from a CST/NN solution.

Independently, CST/NN solutions are better than GX and RRR solutions. Further, the amount of work performed by 2-opt is least when it is restarted from CST/NN solutions. With this more balanced work load between the crossover operator and the local optimizer, a better overall performance is also observed in CST/NN-2-Opt.

4.2.3 The Role of Crossover in a Hybrid Operator

In a hybrid operator, crossover can be viewed as a means of restarting the local optimizer. However, using crossover may not be significantly different that using random restart. In random search, each solution examined has an equal probability of being the best. For a series of (random) independent trials, a graph of when the best solution is found should initially approximate a uniform distribution, and then tail to zero. However, if the search process generates new solutions by improving existing solutions, the graph should be zero for an initial period. After this period, the graph could have a large peak – representing strong convergence during the search process; or the graph could be flat – representing a largely random search after some initial improvements.

Using the lin318 TSP instance, GENITOR was run with CST/NN-2-Opt, GX-2-Opt, and RRR-BMM for 250 generations. The initial population was 400 random 2-opt solutions. For 100 independent runs, the generation in which the best solution is found was recorded. The results are compiled in bar graphs. (See Figure 2.) The graph for CST/NN-2-Opt shows a strong convergence around 80 generations. The graph for RRR-BMM is roughly flat after 100 generations.

5 Discussion

It is a popular technique to use a local optimizer to post-process crossover solutions. However, the effectiveness of the resulting hybrid operator may be due more to the local optimizer than crossover. In particular, the quality of the initial crossover solution (e.g. RRR) can be poor. For these hybrid operators, crossover has a secondary role as a restart method for the local optimizer. In fact, it has been argued that it is more productive to view a hybrid GA as a best-of-k-runs approach instead of as a neighborhood search [JM97]. This paper

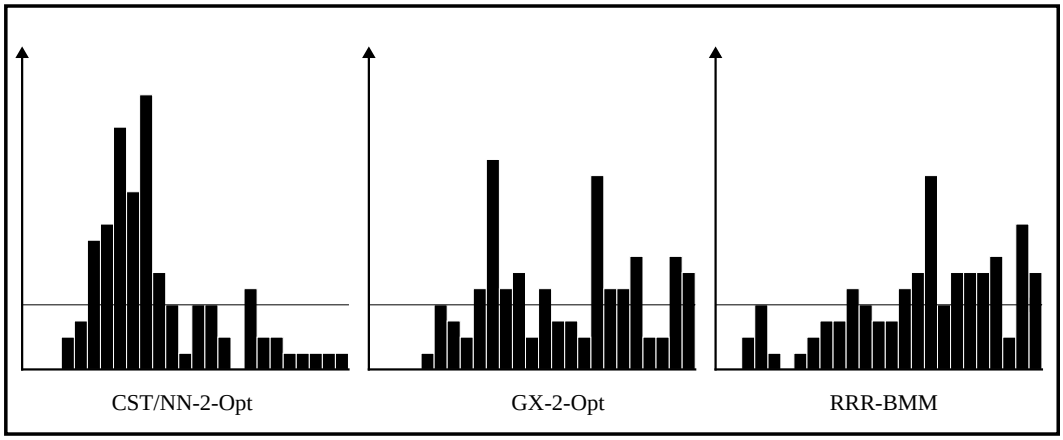

Figure 2: Generation in which best solution is found for 100 independent runs of CST/NN-2-Opt, GX-2-Opt, and RRR-BMM; each run lasts 250 generations. Bars represent 10 generations. Horizontal line represents approximate initial (uniform) distribution of random search.

attempts to put the "genetics" back into genetic algorithms by returning the emphasis to the role of crossover.

5.1 The Schema Theorem

The original Schema Theorem was developed for standard representations and one-point crossover – features absent from many current GAs. Further, the basis of implicit parallelism [GB89] and the value of the Schema Theorem altogether [Müh97] have been cast into doubt. Nonetheless, the dominating design focus for crossover is still combination.

The new Schema Theorem developed in this paper is representation independent and crossover independent. It presents common schemata to be important. The mechanism of crossover is to use the common schemata of two parents as a base on which to build a new offspring. Specifically, a new two step process is defined for crossover: 1) preserve the maximal common schema of two parents, and 2) complete the solution with a construction heuristic. It seems more appropriate to use this Commonality-Based Crossover Framework for designing non-standard crossover operators.

When following this design framework, it can be viewed that commonality-based operators actively identify the schema that is responsible for the high observed fitness of their parents. This process of active identification provides a means by which (construction) heuristics can be seamlessly embedded into crossover operators. However, the concept of implicit parallelism is affected by the use of construction heuristics and the use of two parents to select schemata. In particular, the trials allocated to each schema are no longer completely independent of all other schemata. The allocation of trials now depends on which *pairs* of solutions are selected for crossover, and which schemata are subsequently added by the construction heuristic.

5.2 Hybrid Operators

The Commonality-Based Crossover Framework extends to hybrid operators. Here, the role of crossover is not necessarily to find good solutions, but to find good start points. One characteristic of a good start point is that it is a solution of reasonable quality. This increases the efficiency of the local optimizer [UPvL+91]. A second characteristic of a good start point is that it leads to a good final solution. Since good solutions can be highly similar [Müh91], it is reasonable to search in the neighborhood of (good) parent solutions.

By preserving common schemata (and incorporating construction heuristics), an offspring that is similar to (in the same hyper-plane of) its parents can be created. A local optimizer that is restarted from this offspring is likely to produce a final solution that is in a neighborhood close to the parent solutions. Following the original intention of genetic algorithms, information from good parent solutions is exploited to help find better solutions. Conversely, restarting a local optimizer from an offspring built by combining random parts of two parents may not be significantly different than a random restart.

Experimentally, the final 2-opt solutions of start points provided by CST/NN are close to their parents. Effectively exploiting the information on good schemata (edges) provided by the parents, search converges (in a neighborhood near the parents) and finds good solutions. Conversely, GX (combination induced hitch-hiking) and RRR (weak construction heuristic) provide start points that are more different from their parents. Instead of being able to effectively exploit the schemata of good parent solutions, the "exploratory horizon beyond which genetic search degrades to random search" is exceeded [MdWS91].

6 Conclusions

The suggestion that combination should be the design focus of crossover was not intended for extension to hybrid operators. Restarting a local optimizer from the randomly combined parts of two parent solutions may not be significantly different than a random restart. Subsequently, the overall effectiveness of these hybrid operators is due mostly to the local optimizer. However, the defining attribute of a genetic algorithm should be crossover. A well designed (commonality-based) crossover operator can improve the performance of a local optimizer. With this reestablished role for crossover in hybrid operators, the overall search method can again be called a "genetic" algorithm.

7 Acknowledgements

The work described in this paper was sponsored in part by the Advanced Research Projects Agency and Rome Laboratory, Air Force Material Command, USAF, under grant numbers F30602-95-1-0018 and F30602-97-C-0227, and the CMU Robotics Institute. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Advanced Research Projects Agency and Rome Laboratory or the U.S. Government.

References

[CS98] S. Chen and S.F. Smith. Experiments on commonality in sequencing operators. In *Genetic Programming 1998: Proceedings of the Third Annual Conference*, 1998.

[Dav91] L. Davis. *Handbook of Genetic Algorithms.* Van Nostrande Reinhold, 1991.

[ECS89] L.J. Eshelman, R.A. Caruana, and J.D. Schaffer. Biases in the crossover landscape. In *Proc. Third International Conference on Genetic Algorithms*, 1989.

[Esh91] L.J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In G. Rawlins, editor, *Foundations of Genetic Algorithms.* Morgan Kaufmann, 1991.

[GB89] J. Grefenstette and J.E. Baker. How genetic algorithms work: A critical look at implicit parallelism. In *Proc. Third International Conference on Genetic Algorithms*, 1989.

[GGRG85] J. Grefenstette, R. Gopal, B. Rosmaita, and D. Van Gucht. Genetic algorithms for the traveling salesman problem. In *Proc. of an International Conference on Genetic Algorithms and their Applications*, 1985.

[Gol89] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

[GS94] D.P. Greene and S.F. Smith. Using coverage as a model building constraint in learning classifier systems. *Evolutionary Computation*, 2:67–91, 1994.

[Hol75] J. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, 1975.

[JM97] D.S. Johnson and L.A. McGeoch. The traveling salesman problem: A case study. In E.H.L. Aarts and J.K. Lenstra, editors, *Local Search in Combinatorial Optimization.* John Wiley & Sons, 1997.

[MdWS91] B. Manderick, M. de Weger, and P. Spiessens. The genetic algorithm and the structure of the fitness landscape. In *Proc. Fourth International Conference on Genetic Algorithms*, 1991.

[Müh91] H. Mühlenbein. Evolution in time and space–the parallel genetic algorithm. In G. Rawlins, editor, *Foundations of Genetic Algorithms.* Morgan Kaufmann, 1991.

[Müh97] H. Mühlenbein. Genetic algorithms. In E.H.L. Aarts and J.K. Lenstra, editors, *Local Search in Combinatorial Optimization.* John Wiley & Sons, 1997.

[NV92] A. Nix and M. Vose. Modeling genetic algorithms with Markov chains. *Annals of Mathematics and Artificial Intelligence*, 5:79–88, 1992.

[Rad91] N.J. Radcliffe. Forma analysis and random respectful recombination. In *Proc. Fourth International Conference on Genetic Algorithms*, 1991.

[RS94] N.J. Radcliffe and P.D. Surry. Formal memetic algorithms. In T.C. Fogarty, editor, *Evolutionary Computing. AISB Workshop.* Springer-Verlag, 1994.

[Sch87] J.D. Schaffer. Some effects of selection procedures on hyperplane sampling by genetic algorithms. In L. Davis, editor, *Genetic Algorithms and Simulated Annealing.* Morgan Kaufmann, 1987.

[SdJ91] W.M. Spears and K.A. de Jong. An analysis of multi-point crossover. In G. Rawlins, editor, *Foundations of Genetic Algorithms*. Morgan Kaufmann, 1991.

[SG87] J.Y. Suh and D. Van Gucht. Incorporating heuristic information into genetic search. In *Proc. Second International Conference on Genetic Algorithms and their Applications*, 1987.

[SMM+91] T. Starkweather, S. McDaniel, K. Mathias, C. Whitley, and L.D. Whitley. A comparison of genetic sequencing operators. In *Proc. Fourth International Conference on Genetic Algorithms*, 1991.

[Sys89] G. Syswerda. Uniform crossover in genetic algorithms. In *Proc. Third International Conference on Genetic Algorithms*, 1989.

[UPvL+91] N.L.J. Ulder, E. Pesch, P.J.M. van Laarhoven, H.-J. Bandelt, and E.H.L. Aarts. Improving tsp exchange heuristics by population genetics. In R. Männer and H.-P. Schwefel, editors, *Parallel Problem Solving In Nature*. Springer-Verlag, 1991.

[WS90] L.D. Whitley and T. Starkweather. Genitor II: A distributed genetic algorithm. *Journal of Experimental and Theoretical Artificial Intelligence*, 2:189–214, 1990.