Learning Individual Mating Preferences

Lisa M. Guntly
Natural Computation Laboratory
Department of Computer Science
Missouri University of Science and Technology
Rolla, Missouri, U.S.A.
lguntly@acm.org

Daniel R. Tauritz
Natural Computation Laboratory
Department of Computer Science
Missouri University of Science and Technology
Rolla, Missouri, U.S.A.
dtauritz@acm.org

ABSTRACT

Mate selection is a key step in evolutionary algorithms which traditionally has been panmictic and based solely on fitness. Various mate selection techniques have been published which show improved performance due to the introduction of mate restrictions or the use of genotypic/phenotypic features. Those techniques typically suffer from two major shortcomings: (1) they are fixed for the entire evolutionary run, which is suboptimal because problem specific mate selection may be expected to outperform general purpose mate selection and because the best mate selection configuration may be dependent on the state of the evolutionary run, and (2) they require problem specific tuning in order to obtain good performance, which often is a time consuming manual process. This paper introduces two versions of Learning Individual Mating Preferences (LIMP), a novel mate selection technique in which characteristics of good mates are learned during the evolutionary process. Centralized LIMP (C-LIMP) learns at the population level, while Decentralized LIMP (D-LIMP) learns at the individual level. Results are presented showing D-LIMP to outperform a traditional genetic algorithm (TGA), the Variable Dissortative Mating Genetic Algorithm (VDMGA), and C-LIMP on the DTRAP and MAXSAT benchmark problems, while both LIMP techniques perform comparably to VDMGA on NK Landscapes.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; I.2.6 [**Artificial Intelligence**]: Learning; G.1.6 [**Numerical Analysis**]: Optimization

General Terms

Algorithms, Performance

Keywords

Reinforcement Learning, Mate Selection, Evolutionary Algorithm

GECCO'11, July 12–16, 2011, Dublin, Ireland.

1. INTRODUCTION

Mate selection strategies in evolutionary algorithms (EAs) are traditionally panmictic. These strategies allow recombination between any individuals regardless of their genotypic or phenotypic differences. As a result, individuals tend to become increasingly similar over time, which can cause the population to converge to local optima [15]. Additionally, EAs have typically required that the individuals encoding better solutions be used more often in the mate selection process [15]. This is based on the understanding that combining higher quality solutions is more likely to result in better offspring. However, using some individuals to create offspring more often than others can also reduce the diversity of the population and lead the EA to prematurely converge on a suboptimal solution.

In nature, another mechanism for mate selection based on the genotypic or phenotypic traits of individuals is frequently observed. This mechanism encompasses various strategies based on the ancestry or likeness of individuals involved in reproduction. For example, humans are known to mate outside their family tree (in most cultures). In EAs, this type of mating would be based in some fashion on the ancestry or a distance measurement between individuals. Many of the published mate selection techniques based on pre-selected genotypic or phenotypic preferences use fixed preferences for the entire run of the EA. The inability to change mating preferences can result in suboptimal solutions due to problem specific details that a fixed method cannot adjust to handle. Additionally, the best configuration of genotypic or phenotypic preferences may depend on the current state of the population and the corresponding trade-off between exploration and exploitation. These mate selection techniques also need to be tuned correctly for problems, as mating restrictions or preferences are likely not universally helpful. Sexual selection can be considered an extension of genotypic or phenotypic trait-based selection, which appears in nature among animals. This type of selection involves developing more elaborate preferences for selecting a mate based on additional features of other individuals.

This paper introduces Learning Individual Mating Preferences (LIMP), which provides individuals a chance to develop an understanding of the best features to look for in a mate. This idea borrows from the concept of sexual selection in nature, where features of good mates are learned during the evolutionary process. Decentralized LIMP (D-LIMP) allows every individual to develop a genotypic preference for other individuals in the population by learning from the results of their prior reproductions. Centralized LIMP (C-

LIMP) develops a population wide genotypic preference for mates relative to the possible gene values. In D-LIMP, each individual examines the fitness of the offspring it produces and then modifies its mating preference based on the results. The individual likes the genes of the mate it selected if the offspring had a higher fitness than itself or dislikes the genes of the selected mate if the offspring was of lower quality than itself. Likewise, C-LIMP updates a population wide genotypic preference based on offspring fitness. The obtained knowledge in D-LIMP is passed on to the offspring generated, while in C-LIMP the knowledge is already known to all individuals. Both techniques allow the mate selection process to become more refined as evolution occurs.

2. RELATED WORK

Several nature-inspired alternatives to traditional panmictic mate selection schemes for EAs have been examined by the EA community [1, 2, 4, 5, 6, 8, 9, 12, 13]. The concept of niching in EAs derives from nature - where creatures evolve into distinct species in different areas of the environment. In EAs, niching is implemented via splitting populations into niches based on a predefined notion of distance. Individuals in these niches only mate with others in the same niche [1]. This reduces the loss of genetic diversity in the population due to genetic drift, but requires an a priori distance measurement. Unlike niching, LIMP uses a measurement of the similarity of an individual's learned preferred genes to potential mates' genes, and mating is not restricted, just less likely to occur between less incompatible individuals.

LIMP is similar to assortative mating techniques, which have individuals select a mate based on similarity (positive assortative mating) or dissimilarity (negative assortative mating) of their genotypes or phenotypes. Negative assortative mating is known to increase the genetic diversity of the population by increasing the frequency of heterozygous genotypes. The Variable Dissortative Mating Genetic Algorithm (VDMGA) was introduced as an adaptable method of negative assortative mating that implements a hamming distance threshold restriction on mating. The restriction tends to loosen over time but can become stricter as well [4]. LIMP is similarly a non-random mating technique for selecting mating pairs. However, rather than strictly comparing genotypes with a threshold value, LIMP compares a learned vector of desirable features developed from past experience to the genotype of potential mates. Also, LIMP learns about what genes individuals want to see in a mate, rather than assuming more dissimilarity between their genes is better.

Estimation of Distribution Algorithms (EDAs), use an estimation of the distribution of good solutions based on known solutions to probabilistically generate new solutions that are likely to be high quality [14]. Comparably, LIMP estimates the distribution of potential mates in the population that are likely to result in improved offspring quality. The results of previous matings are used to determine superior mating partners for the current generation. C-LIMP has more in common with EDAs than D-LIMP because C-LIMP uses population wide learning. EDAs and LIMP use a distribution of known features to select new features. However, EDAs use the distribution to generate new solutions, while LIMP uses the distribution to rate potential mates. In addition, EDAs use a distribution built from the entire population and lack some of the characteristic EA concepts of mate selection and recombination, while individuals in D-LIMP each have a separate distribution of good mating characteristics and traditional EA recombination methods are used to generate new solutions for both LIMP algorithms.

The use of offspring quality as a means to learn which individuals produce good offspring together has been examined in Reinforcement Learning for mate selection in Cellular Genetic Algorithms [12]. In this type of cellular genetic algorithm, individuals are more likely to mate with neighbors located near them on the topological grid of a cellular genetic algorithm. Reinforcement learning is used to reposition individuals on the grid so that individuals that produce good offspring are moved closer together and individuals that produce bad offspring are moved further apart. LIMP also looks at offspring fitness to update the pairing preferences for future mate selection. However, in LIMP each gene of an individual is considered in the search for a mate and the desirability of each gene is updated, while in Reinforcement Learning for mate selection in Cellular Genetic Algorithms the entire genotype is considered as a single unit.

The concept of LIMP is based on two prior mate selection algorithms [8, 9]. The original concept of Learning Offspring Optimizing Mate Selection [9] required every individual that is selected for mating to evaluate every other individual in the population to determine a compatible mate, resulting in significant overhead time for the algorithm. An alternate version of this algorithm that focused on reducing the overhead, Estimated Learning Offspring Optimizing Mate Selection probabilistically selected good enough mates for each individual in the population [8]. Instead of a probabilistic method, LIMP uses a tournament to select mates and does not require that mate to reciprocate an interest in mating. The tournament process reduces the overhead of LIMP compared to the original concept, as the probabilistic method does, but creates more compatible mating pairs as well. Additionally, the values representing mate quality and desirable features used in the original methods converge to intermediate values, reducing the information that can be discovered about potential mates. LIMP avoids this pitfall by updating the desirable features for potential mates in a way that allows possible convergence of mate quality and feature desirability values to one extreme or the other.

While many effective mate selection methods exist in literature, mate selection based on specific genotypic features learned during evolution has only recently been examined as an option [9, 8]. Other related algorithms base mate selection on genotypic features preselected by the algorithm designer. In this paper, we study the effects of D-LIMP and C-LIMP compared to a successful assortative mating algorithm, VDMGA, and a traditional genetic algorithm (TGA) using tournament mate selection methods on several problems with binary string-based representation.

3. LIMP METHODOLOGY

LIMP incorporates two additional components into a traditional EA. The first component, at the mate selection stage of an EA, involves examining a limited number of potential mates for an individual j by comparing the desired genes of j to other individuals' genotypes. The most suitable examined individual is selected to produce an offspring with j. The second component entails updating the desired genes of j and its mate, for D-LIMP, or the desired genes of the population, for C-LIMP, based on the fitness of the offspring relative to each parent.

Table 1: Example of individuals employing D-LIMP and their $MAC_{D-LIMP}(j,k)$ values

j	s_j	d_j				k=1	k=2	k=3	k=4
1	0101	.7	.6	.7	.2	.350	.650	.300	.450
2	1010	.3	.4	.8	.6	.475	.525	.525	.375
3	0001	.9	.5	.8	.4	.300	.700	.300	.500
4	1101	.2	.6	.6	.6	.600	.400	.550	.450

Table 2: Example of mate selection with D-LIMP using the MAC values of Table 1

Parent	Event	MAC	Result
1	examine 3	.300	
	examine 2	.650	mate pair (1,2)
4	examine 1	.600	
	examine 3	.550	mate pair (4,1)
2	examine 4	.375	
	examine 1	.475	mate pair (2,1)

3.1 Mate Selection

At the mate selection stage of LIMP, λ randomly selected individuals are given the opportunity to examine the suitability of potential mates, with each examining a limited number of potential mates and selecting the most suitable individual found to mate with. The order in which potential mates are examined is arbitrary.

3.1.1 D-LIMP

To estimate the suitability of a potential mate in D-LIMP, each individual j maintains a real-valued vector d_j, in addition to a bit string genotype encoding individual j's trial solution to a problem, s_j. Let d_j be referred to as the *desiredFeatures* vector of individual j. The length of d_j equals the length of s_j, and each element i of d_j, $d_j[i]$, represents how much individual j wants the i^{th} bit of its potential mate to be a one. All elements of d_j are initialized to $\neg s_j$.

The *Mate Acceptance Chance* (MAC), first defined in [9], for an individual j examining an individual k in D-LIMP is defined as:

$$MAC_{D-LIMP}(j,k) = \frac{\sum_{i=1}^{L} 1 - (|s_k[i] - d_j[i]|)}{L} \quad (1)$$

where L is the length of the bit string solution and $s_k[i]$ is the i^{th} bit of k's trial solution. The value of each element of d_j is in the range [0,1], so the value of $MAC_{D-LIMP}(j,k)$ is also in the range [0,1]. This way $MAC_{D-LIMP}(j,k)$ estimates the suitability of k as a potential mate for j.

The following example demonstrates how D-LIMP works. Suppose the population consists of the 4 individuals shown in Table 1. The values of $MAC_{D-LIMP}(j,k)$ for all $j,k = 1,\ldots,4$ are also shown in Table 1. At the parent selection stage, λ parents are randomly sampled with replacement from the population and each looks for a mate. For each parent looking for a mate, t_{MAC}, the tournament size for the MAC value, individuals are randomly selected from the population, and the one with the highest MAC value is chosen to be the mating partner for that individual. An example of the mate selection process with $t_{MAC} = 2$ and $\lambda = 3$ is shown in Table 2.

Table 3: Example of individuals employing C-LIMP and their $MAC_{C-LIMP}(j,k)$ values

d_{P0} : .3 | .1 | .8

d_{P1} : .2 | .9 | .3

j	s_j	k=1	k=2	k=3
1	010	.60	.40	.33
2	101	.53	.47	.80
3	000	.47	.67	.60

Table 4: Example of mate selection with C-LIMP using the MAC values of Table 3

Parent	Event	MAC	Result
3	examine 3	.60	
	examine 2	.67	mate pair (3,2)
2	examine 1	.53	
	examine 2	.47	mate pair (2,1)

3.1.2 C-LIMP

To estimate the suitability of a potential mate in C-LIMP, the population maintains real-valued vectors d_{P0} and d_{P1} indicating how much any individual with a gene value of 0 at i or 1 at i respectively wants the i^{th} bit of its potential mate to be one. All elements of d_{P1} are initialized to 0 and all elements of d_{P0} to 1.

The MAC for individual j examining individual k in C-LIMP is defined as:

$$MAC_{C-LIMP}(j,k) = \frac{\sum_{i=1}^{L} adj[i]}{L} \quad (2)$$

where

$$adj[i] = \begin{cases} 1 - (|s_k[i] - d_{P1}[i]|) & \text{if } s_j[i] = 1 \\ 1 - (|s_k[i] - d_{P0}[i]|) & \text{if } s_j[i] = 0 \end{cases} \quad (3)$$

Table 3 provides an example similar to the D-LIMP example demonstrating how C-LIMP works with a population of three. An example of the mate selection process with $t_{MAC} = 2$ and $\lambda = 2$ is shown in Table 4.

3.2 Learning desirable mate qualities

When offspring are created via recombination in D-LIMP, the corresponding *desiredFeatures* vectors of the parents are recombined in the same fashion. Specifically, if an offspring obtained the i^{th} bit of its solution from parent A, then the i^{th} element of the offspring's *desiredFeatures* vector is copied from the corresponding data of parent A. This process ensures that the parents' knowledge about desirable mate qualities relevant to their offspring's genotype is passed on to their offspring.

Knowledge about desirable mate qualities is obtained by observing the outcome of reproduction. Each time an individual participates in reproduction, it updates its *desiredFeatures* vector in D-LIMP, or the population *desiredFeatures* vector in C-LIMP, depending on how the offspring's fitness compares to his own.

Let j and k be the sets of two parents described in Table 1 and identified as mating pairs in Table 2. Possible offspring, m, from each mating pair's recombination are described in Table 5.

The update of the *desiredFeatures* vectors depends on tracking which parent contributed each gene that child received in recombination. The following method describes

Table 5: Examples of offspring m created by the mating pairs from Table 2

m	inherited genes	s_m	d_m			
1	$s_1 s_1 s_2 s_2$	0110	.7	.6	.8	.6
2	$s_4 s_4 s_4 s_1$	1100	.3	.6	.7	.6
3	$s_2 s_1 s_1 s_2$	1101	.2	.6	.6	.2

how to update the *desiredFeatures* vectors for possible uniform and n-point crossover, where the origin of the child's genes can be determined.

Let $F(j)$, $F(k)$, and $F(m)$ be the fitness values of individuals j, k, and m respectively where m is the offspring of j and k. There are two general cases for updating the $desiredFeatures$ in LIMP:

Case 1: The child has an equal or lower fitness than a given parent. If $F(m) \leq F(j)$ then k was not a suitable mate for j. Specifically, j should avoid mating with individuals having the same genes that k gave to m. Correspondingly, if $F(m) \leq F(k)$ then j was not a suitable mate for k and k should avoid mating with individuals having the same genes that j gave to m.
Case 2: The child has a higher fitness than a given parent. If $F(m) > F(j)$ then individual j selected a suitable mate. Specifically, j should consider the genes k gave to m more suitable in future mates. Similarly, if $F(m) > F(k)$ then individual k found j to be a suitable mate and k should consider the genes j gave to m more suitable in future mates.

For D-LIMP, in all cases, each individual updates its *desiredFeatures* vector to consider individuals with more suitable genotypic qualities. For individual j,d_j is updated only for the gene locations that k gave to the child and the rest of d_j remains unchanged. The d_j values are updated such that in the next generation, the value of $MAC(j,k)$ is higher if j found k to be a suitable mate and lower if j found k to be a bad mate.

Updates in C-LIMP are performed in the same fashion, except that all updates that would happen to location i in d_j instead happen to $d_{P1}[i]$ if $s_j[i] = 1$ or happen to $d_{P0}[i]$ if $s_j[i] = 0$ and all updates that would happen to a location in d_k instead happen to d_{P1} if $s_k[i] = 1$ or d_{P0} if $s_k[i] = 0$.

The size of the updates, u_j and u_k for each location receiving an update in d_j and d_k are defined as:

$$u_j = \left| 1 - \frac{F(m)}{F(k)} \right| \tag{4}$$

and

$$u_k = \left| 1 - \frac{F(m)}{F(j)} \right| \tag{5}$$

where $F(j)$ and $F(k)$ are not allowed to be zero, and the fitness range must be positive values. The equations indicate the degree to which j found k to be a suitable mate or a bad mate and the reverse for k considering j by calculating the percentage change of the child's fitness relative to the parent's fitness. Using these calculations for updating the *desiredFeatures* allows offspring that are much better or worse than the parent to more drastically change the *desiredFeatures* of the parent or population. Consequently, offspring that are only slightly better or worse than the parent will only be allowed to slightly change the *desiredFeatures* of the parent or population, as they provide less significant information about what the parent or population actually prefers in a mate. The range of all values in $desiredFeatures$ is $[0, 1]$, so any update that causes a value to fall outside that range is limited to setting the value to 0 or 1.

Table 6: Example of updating d vectors in D-LIMP with m receiving $s[1], s[2]$ from j and $s[3]$ from k

	j			k			m		
s	101			110			100		
$d(old)$	.7	.2	.7	.4	.3	.8	.7	.2	.8
fit	20			15			18		
u	0.1			0.2			–		
$d(new)$	.7	.2	.8	.6	.1	.8	.7	.2	.8

For D-LIMP, if j found k to be a suitable mate, for all gene locations i that k gave to m, if $s_k[i] = 1$, the value of $d_j[i]$ is increased by u_j; otherwise $s_k[i] = 0$ and the value of $d_j[i]$ is decreased by u_j. In C-LIMP, if $s_k[i] = 1$ then either $d_{P0}[i]$ or $d_{P1}[i]$ (depending on $s_j[i]$) is increased by u_j; otherwise $s_k[i] = 0$ and the value of $d_{P0}[i]$ or $d_{P1}[i]$ is decreased by u_j. This ensures that in the next generation the value of $MAC(j,k)$ is higher than it was in the current generation.

For D-LIMP, if j found k to be a bad mate, for all gene locations i that k gave to m, if $s_k[i] = 0$, the value of $d_j[i]$ is increased by u_j; otherwise $s_k[i] = 1$ and the value of $d_j[i]$ is decreased by u_j. In C-LIMP, if $s_k[i]$ is 0 then either $d_{P0}[i]$ or $d_{P1}[i]$ (depending on $s_j[i]$) is increased by u_j; otherwise $s_k[i]$ is 1 and the value of $d_{P0}[i]$ or $d_{P1}[i]$ is decreased by u_j. This ensures that in the next generation the value of $MAC(j,k)$ is lower than it was in the current generation.

Table 6 provides an example of the $desiredFeatures$ updating process for D-LIMP, while C-LIMP updates different $desiredFeatures$ vectors likewise. If mutation is applied to an offspring's candidate solution after crossover, the updates to the *desiredFeatures* vectors are performed *after* mutation took place to avoid evaluating offspring fitness twice. Note that *desiredFeatures* vectors are always updated deterministically and are never mutated. Thus mutation of candidate solutions may occasionally introduce inaccuracies in *desiredFeatures* vectors, as the updates are based on the crossover operator only, while the offspring fitness value depends on both crossover and mutation.

4. EXPERIMENTAL SETUP

D-LIMP and C-LIMP were compared to a traditional genetic algorithm (TGA) and a recent adaptive assortative mating algorithm VDMGA [4] on a 4-bit fully deceptive trap problem, unrestricted NK Landscapes, and MAXSAT.

4.1 Test Problems

This section describes the test problems use for experimental comparison in this paper.

4.1.1 Deceptive Trap (DTRAP)

In a DTRAP with a trap size of four, a binary string is separated into groups of 4 bits each that are considered independently for calculating fitness [3]. The fitness of a candidate solution is the sum of the fitnesses of each of the traps.

Each trap has a fitness that is a function of the number of ones in the trap and is computed as:

$$f(n) = \begin{cases} 3-n & \text{if } n < 4 \\ 4 & \text{if } n = 4 \end{cases} \quad (6)$$

where n is the number of 1s in the trap string of 4 bits. The objective is to maximize the sum of these trap functions. These tests considered concatenated DTRAP problems with overall lengths $L = 100$, $L = 300$, and $L = 500$. The best performing n in n-point crossover for each DTRAP problem length was used for all algorithms. The values for n corresponding to increasing L are noted in Table 7.

In addition to concatenated DTRAP problems, a 4-bit trap ($L = 100$) where the bits of each trap are maximally distanced throughout the genotype is examined. By splitting the bits of the trap apart, the usefulness of positional bias from n-point crossover is eliminated, allowing the examination of the usefulness of a properly biased crossover operator for each algorithm. Concatenated DTRAP instances are referred to as DTRAP1 (DT1) problems and the maximally split DTRAP type is referred to as DTRAP2 (DT2).

4.1.2 NK Landscapes

An NK Landscape [11] is a tunable fitness landscape based on the length of the problem, n, and the number of interactions between genes, k. The fitness of a genotype is the sum of the contributions from each gene. The contribution of each gene to the overall fitness depends on its own value and the value of k other genes without restrictions. Because finding a global optimum for unrestricted NK Landscapes is NP-complete for $k > 1$ and the optimum is not predetermined, the best fitness found through any of the experiments on a given NK Landscape is used as the optimum fitness for scaling mean best fitness (MBF) values in graphs. This method is more likely to be accurate with smaller k values, but is useful to approximate the global optimum for NK Landscapes that are not small enough to have the global optimum calculated exactly. These experiments evaluated the algorithms on unrestricted NK Landscapes with $n = 50$ and $k = 2$, $k = 3$, $k = 4$, and $k = 5$.

4.1.3 MAXSAT

The maximum satisfiability problem (MAXSAT) entails determining the maximum number of clauses in a conjunctive normal form (CNF) formula that can be satisfied by an assignment of the variables. Our experiments were performed on randomly generated 3-CNF formulas with 200 variables and clause-to-variable ratios of $r = 2$, $r = 3$, $r = 4$, and $r = 5$. MAXSAT for 3-CNF formulas is an NP-complete problem [10] and has qualities that are typically difficult for EAs, such as deceptiveness and high multimodality.

4.2 Tested Algorithms

All three of the algorithms in these experiments use a traditional EA structure where a population of candidate solutions is generated uniform randomly. In these experiments we terminate each run when the global optimum has been found or a maximum number of evaluations has been reached. Convergence of the algorithms was measured by tracking the number of evaluations used at point when the entire population converged on a single genotype. For parent selection, D-LIMP and C-LIMP use a tournament on MAC values for comparing mates with size t_{MAC}. TGA and VDMGA both employ tournament parent selection with size t_p. For survival selection, two types of selection are considered. A tournament selection keeping the best of t_c individuals in each tournament is considered for all problems. The other method examined is restricted tournament replacement (RTR) [7] where offspring replace the most similar of t_{RTR} individuals in the population, measured in hamming distance between genotypes, only if the offspring is better than the most similar individual. RTR typically reduces the population size required for an EA by preserving diversity, so smaller population sizes are used for RTR. Overall parameters used are given in Table 7. All results are based on 60 runs of the specified algorithm and statistical comparisons are done using a one-way ANOVA at a .05 level of significance.

Table 7: Parameters used in the experiments

Parameter	DT1	DT2	NK	MAXSAT
μ_t	1000	500		
μ_{RTR}	500		100	
λ	50			
$t_{p(TGA,VDMGA)}$	2			
t_c	5			
t_{RTR}	20		10	
t_{MAC}	20			
recombination	n-pt (3,7,13)	uniform		
mutation	bit flip			
mutation prob.	$1/L$			
max evals	200000			

5. EXPERIMENTAL RESULTS

5.1 DTRAP

The mean best fitness (MBF), standard deviation, and success rate (SR) for the selected algorithms on DTRAP1 are given in Table 8. Comparisons of the algorithms on MBF and number of evaluations needed to converge on a solution are shown in Figure 1. D-LIMP exhibited the best performance across all problem sizes, and notably scaled very well compared to the other algorithms on larger problem sizes. D-LIMP was the only algorithm to successfully find the optimum with $L = 500$. C-LIMP and VDMGA performed and scaled similarly and required comparable numbers of evaluations to converge. D-LIMP produced the best overall results with tournament selection, but also required the most evaluations to get those results. TGA performed and scaled poorly compared to the other algorithms. For a problem with little disruption due to crossover, D-LIMP clearly outperforms C-LIMP and the other algorithms. Using RTR selection proved to only be beneficial to TGA, with tournament selection producing statistically better results for all other algorithms on the larger problem sizes. The smaller population size used with RTR – a typical benefit of RTR over tournament selection – likely contributed to this result.

The results on DTRAP2 for examining the importance of an appropriate crossover operator are shown in Table 9. All algorithms performed significantly worse than in DTRAP1, where positionally biased n-point crossover aided the algorithms' performances. While D-LIMP and C-LIMP provided the best results of the algorithms on DTRAP2, overall

Table 8: Results on DTRAP1

	stat	TGA	VDMGA	D-LIMP	C-LIMP
$L100$	SR	81.8	100	100	100
Tour.	MBF	99.85	100	**100**	100
	st. dev	0.376	0	0	0
$L100$	SR	93.46	100	100	100
RTR	MBF	99.94	100	100	100
	st. dev	0.242	0	0	0
$L300$	SR	0	28.9	91.3	26.2
Tour.	MBF	97.51	99.73	**99.98**	99.41
	st. dev	2.199	0.859	0.236	1.261
$L300$	SR	1.3	19.5	91.7	19.3
RTR	MBF	98.25	99.49	99.97	99.33
	st. dev	1.834	1.357	0.212	1.661
$L500$	SR	0	0	79.4	0
Tour.	MBF	95.89	98.73	**99.95**	98.55
	st. dev	2.822	2.211	0.478	2.645
$L500$	SR	0	0	26.2	0
RTR	MBF	96.03	98.94	99.78	97.36
	st. dev	3.845	2.130	1.528	4.964

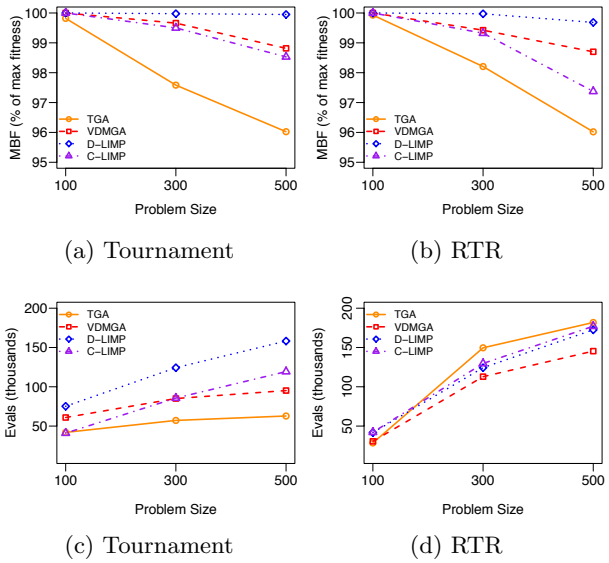
Table 9: Results on DTRAP2

	stat	TGA	VDMGA	D-LIMP	C-LIMP
$L100$	MBF	79.78	82.58	78.92	80.99
Tour.	st. dev	1.883	5.911	2.016	1.973
$L100$	MBF	83.64	70.21	**87.98**	86.65
RTR	st. dev	2.048	1.532	1.783	1.462

performance was poor and indicative of a need to examine more advanced crossover methods in conjunction with these algorithms. The benefit of RTR is more noticeable in these results since the same population size was used for both RTR and tournament selection. Except for the VDMGA, which did not interact well with RTR on this problem, the algorithms performed statistically better with RTR.

5.2 NK Landscapes

The algorithms' performances on NK Landscape problems are provided in Table 10, and comparisons of the algorithms are shown for MBF and evaluations until convergence in Figure 2. VDMGA was the best performing algorithm at larger values of k, but scaled poorly for the number of evaluations required to converge on NK Landscapes. D-LIMP and C-LIMP had similar results and scaled comparably in evaluations required for all values of k, but C-LIMP had significantly better performance than D-LIMP using RTR. This could indicate that RTR is of more use to a centralized learning mechanism than a decentralized one. VDMGA and C-LIMP performed similarly with both replacement methods, while D-LIMP and TGA scale worse with RTR compared to tournament selection. TGA was again the worst performing algorithm with all other algorithms statistically outperforming it for $k > 3$. The anomaly of increased performance for $k = 5$ for some algorithms is likely due to the global optimum going undiscovered. Consequently, the MBF over the best found fitness was higher than it would be if the best found fitness was the optimum.



(a) Tournament (b) RTR

(c) Tournament (d) RTR

Figure 1: DTRAP1 problem results comparing MBF (a,b) and convergence speed (c,d) of the tested algorithms with the indicated survival selection method

5.3 MAXSAT

Results of the algorithms on MAXSAT problems are listed in Table 11 and MBF and number of evaluations until convergence comparisons are plotted in Figure 3. These results show D-LIMP performing slightly, statistically better than VDMGA for the higher values of r, and both scale similarly. The superior performance of D-LIMP comes at the cost of requiring more evaluations to converge on a solution, but this appears to scale better than the other algorithms as r increases. C-LIMP performs the worst on MAXSAT for all values of r, and TGA also produces poor results. The inferior performance of C-LIMP can be explained by the method's lack of scaling with the number of bits in the problem, as the MAXSAT problems used the longest bitstrings of any of the test problems.

6. DISCUSSION

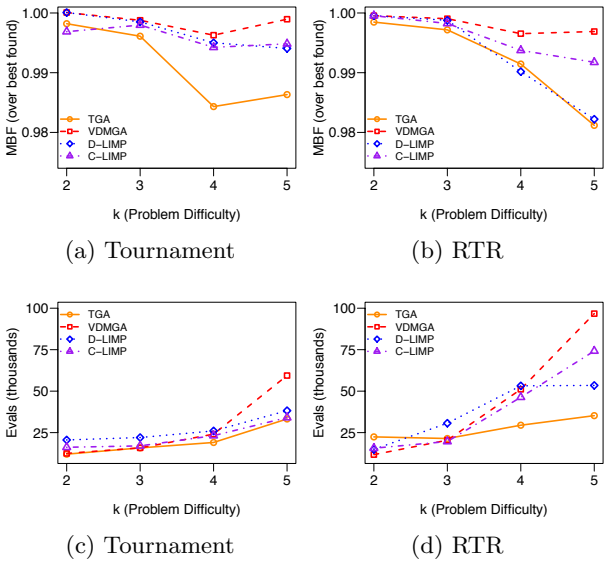
The results indicate that both D-LIMP and C-LIMP are promising mate selection techniques, with D-LIMP especially performing well on DTRAP and MAXSAT and C-LIMP performing well on NK Landscapes. D-LIMP on DTRAP and MAXSAT problems outperforms all other mate selection techniques examined. C-LIMP produced better results than TGA and D-LIMP on NK Landscapes, but VDMGA performed the best. Despite similarities in design, D-LIMP and C-LIMP appear to have different strengths and weaknesses. D-LIMP is able to scale well with larger problem sizes, but C-LIMP performs better on the smaller problem size of NK Landscapes with difficult linkages between the bits. The ability of D-LIMP to maintain more diversity in *desiredFeatures* and thus more diversity in the population compared to C-LIMP could explain this difference in scalability. The problem size of NK Landscapes was small enough that C-LIMP did not need the additional diversity preservation provided by D-LIMP.

Table 10: Results on NK Landscapes

	stat	TGA	VDMGA	D-LIMP	C-LIMP
$k = 2$	MBF	37.54	37.38	**37.39**	37.26
Tour.	st. dev	0.241	0.003	0.002	0.318
$k = 2$	MBF	37.32	37.37	37.37	37.37
RTR	st. dev	0.152	0.089	0.090	0.092
$k = 3$	MBF	37.41	37.50	37.49	37.47
Tour.	st. dev	0.179	0.086	0.095	0.126
$k = 3$	MBF	37.44	**37.51**	37.50	37.48
RTR	st. dev	0.143	0.074	0.083	0.106
$k = 4$	MBF	38.68	39.15	39.11	39.07
Tour.	st. dev	0.602	0.081	0.205	0.216
$k = 4$	MBF	38.95	**39.14**	38.92	39.05
RTR	st. dev	0.336	0.197	0.420	0.292
$k = 5$	MBF	38.29	**38.79**	38.61	38.63
Tour.	st. dev	0.467	0.073	0.258	0.290
$k = 5$	MBF	38.12	38.67	38.16	38.51
RTR	st. dev	0.514	0.173	0.486	0.324

Table 11: Results on MAXSAT

	stat	TGA	VDMGA	D-LIMP	C-LIMP
$r = 2$	MBF	99.99	100	**100**	99.92
Tour.	st. dev	0.053	0	0	0.142
$r = 2$	MBF	99.96	99.99	100	99.98
RTR	st. dev	0.088	0.033	0	0.086
$r = 3$	MBF	99.63	**99.80**	99.79	99.49
Tour.	st. dev	0.137	0.145	0.122	0.259
$r = 3$	MBF	99.57	99.76	99.79	99.54
RTR	st. dev	0.198	0.132	0.133	0.222
$r = 4$	MBF	99.16	99.58	**99.62**	98.91
Tour.	st. dev	0.193	0.132	0.117	0.352
$r = 4$	MBF	98.94	99.42	99.47	99.01
RTR	st. dev	0.258	0.185	0.146	0.252
$r = 5$	MBF	98.41	98.79	**98.85**	98.13
Tour.	st. dev	0.198	0.135	0.119	0.283
$r = 5$	MBF	98.27	98.72	98.83	98.33
RTR	st. dev	0.251	0.203	0.142	0.276



(a) Tournament (b) RTR

(c) Tournament (d) RTR

Figure 2: NK Landscape problem results comparing MBF (a,b) and convergence speed (c,d) of the tested algorithms with the indicated survival selection method

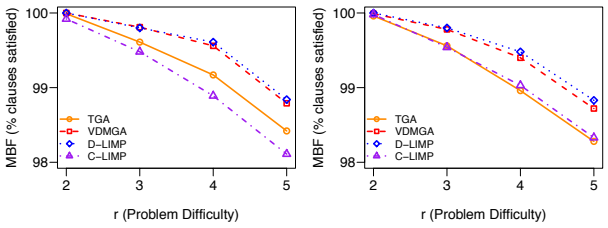
In general, D-LIMP produced better results with tournament selection, while C-LIMP performed either comparably or better using RTR selection. Because RTR helps preserve diversity, this adds further support to C-LIMP maintaining less diversity in the population. D-LIMP preserves diversity sufficiently on its own in many cases and thus does not benefit as much from RTR. An extensive examination of why this occurs would be beneficial in considering the usefulness of each algorithm and would provide insight into these methods' impact on the mate selection process. D-LIMP's overall performance was better than C-LIMP's overall performance, suggesting that decentralized learning for mate selection may outperform centralized learning. Further testing is necessary to confirm this hypothesis.

The number of evaluations needed to converge on a solution was typically less for VDMGA than for C-LIMP or D-LIMP, but VDMGA has additional scaling overhead not measured in the number of evaluations that can be significant if the time to evaluate a solution is relatively small. VDMGA searches through $\mu/2$ members of the population to look for an individual's mate, which means that μ affects the number of hamming distance calculations that must be performed. C-LIMP and D-LIMP examine fewer members of the population in searching for a mate, and do not scale the search on population size.

A possible explanation for VDMGA's better performance on NK Landscapes than either C-LIMP or D-LIMP is that VDMGA does not have to produce the full number of offspring each generation, which can allow VDMGA to perform more exploitation than exploration. Focusing more on exploitation can be beneficial, especially on smaller problem sizes; so examining methods that have a similar impact on C-LIMP and D-LIMP could be useful.

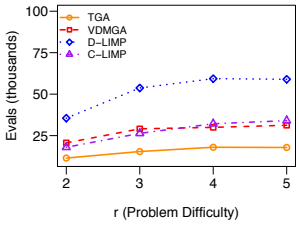
The importance of a proper recombination operator and an understanding of the linkages between variables can be seen from the results of DTRAP1 compared to DTRAP2. From those results, as well as the suboptimal results on NK Landscapes and MAXSAT, both D-LIMP and C-LIMP appear likely to benefit from learning the linkages between variables. Not only could learning variable linkages allow the algorithms to implement a crossover operator based on those linkages, but also the variable linkages could be used to improve the accuracy of the learning process for determining what types of mates are good for an individual.

While the method of updating which features are beneficial in mates in C-LIMP and D-LIMP appeared to generally be successful in the experiments performed, other possibilities for updating these features should be explored. In particular, methods that allow adapting the size of the updates based on characteristics of the population indicating the usefulness of exploration versus exploitation could prove useful. D-LIMP could be modified to work with real-valued vectors assuming a measurement of the distance between a desired feature and the value of a gene and an appropriate calculation for the size of updates to desired features. How-

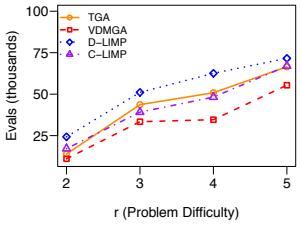


(a) Tournament

(b) RTR



(c) Tournament



(d) RTR

Figure 3: MAXSAT problem results comparing MBF (a,b) and convergence speed (c,d) of the tested algorithms with the indicated survival selection method

ever, C-LIMP is dependent on the possible values of genes, making a direct extension to real-valued vectors infeasible.

7. CONCLUSION

This paper introduces Learning Individual Mating Preferences (LIMP), which provides individuals a chance to develop an understanding of the best features to look for in a mate. This paper describes two versions of LIMP, namely C-LIMP and D-LIMP, for learning the mating preferences of individuals in an EA based on the quality of offspring produced to improve the mate selection process. These methods were compared to a traditional tournament selection for picking parents and VDMGA, which adaptively restricts mating to individuals as dissimilar as possible in the population. Two different replacement methods, tournament and RTR, were examined for each algorithm as well.

The results show D-LIMP with tournament selection outperforming the other algorithms on MAXSAT and DTRAP problem instances, although it tends to require more evaluations to converge. VDMGA produced the best results on NK Landscapes, while C-LIMP produced good results with RTR selection on NK Landscapes. D-LIMP and VDMGA were more successful using tournament selection replacement, while C-LIMP was more successful using RTR.

Results comparing a concatenated trap problem to one with the trap bits maximally split in the genotype showed the significance of using a crossover operator that is correctly biased for the problem structure, and the algorithms should be examined using methods of determining problem structure to aid in recombination and learning.

8. REFERENCES

[1] L. Booker. Improving the Performance of Genetic Algorithms in Classifier Systems. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 80–92, 1985.

[2] R. Craighurst and W. Martin. Enhancing GA Performance through Crossover Prohibitions Based on Ancestry. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 130–135, San Francisco, California, USA, 1995. Morgan Kaufmann Publishers Inc.

[3] K. Deb and D. Goldberg. Analyzing Deception in Trap Functions. In *Proceedings of FOGA II: the Second Workshop on Foundations of Genetic Algorithms*, pages 93–108, 1992.

[4] C. Fernandes and A. C. Rosa. Self-adjusting the Intensity of Assortative Mating in Genetic Algorithms. *Soft Comput.*, 12:955–979, May 2008.

[5] C. Fernandes, R. Tavares, C. Munteanu, and A. Rosa. Using Assortative Mating in Genetic Algorithms for Vector Quantization Problems. In *Proceedings of SAC 2001: the ACM Symposium on Applied Computing*, pages 361–365, New York, NY, USA, 2001. ACM.

[6] R. Fry, S. Smith, and A. Tyrrell. A Self-Adaptive Mate Selection Model for Genetic Programming. In *Proceedings of CEC 2005: the IEEE Congress on Evolutionary Computation*, volume 3, pages 2707 – 2714, München, Germany, September 2005.

[7] G. Harik. Finding Multimodal Solutions Using Restricted Tournament Selection. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 24–31. Morgan Kaufmann, 1995.

[8] E. A. Holdener. *The Art of Parameterless Evolutionary Algorithms*. PhD thesis, Missouri University of Science and Technology, 2008.

[9] E. A. Holdener and D. R. Tauritz. Learning Offspring Optimizing Mate Selection. In *Proceedings of GECCO 2008: Genetic and Evolutionary Computation Conference*, pages 1109–1110, 2008. ACM.

[10] R. Karp. Reducibility Among Combinatorial Problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[11] S. A. Kauffman. Adaptation on Rugged Fitness Landscapes. *Lectures in the Sciences of Complexity*, 1:527–618, 1989.

[12] C.-Y. Lee and E. Antonsson. Reinforcement Learning in Steady-state Cellular Genetic Algorithms. In *Proceedings of CEC 2002: the IEEE Congress on Evolutionary Computation*, volume 2, pages 1793 – 1797, May 2002.

[13] G. Miller. Exploiting Mate Choice in Evolutionary Computation: Sexual Selection as a Process of Search, Optimization, And Diversification. In *Selected Papers from AISB Workshop on Evolutionary Computing*, pages 65–79, London, UK, 1994. Springer-Verlag.

[14] M. Pelikan, D. Goldberg, and F. Lobo. A Survey of Optimization by Building and Using Probabilistic Models. *Computational Optimization and Applications*, 21(1):5–20, 2002.

[15] W. R. M. U. K. Wickramasinghe, M. van Steen, and A. Eiben. Peer-to-peer Evolutionary Algorithms with Adaptive Autonomous Selection. In *Proceedings of GECCO 2007: Genetic and Evolutionary Computation Conference*, pages 1460–1467, 2007. ACM.