Selection Strategies for Ambiguous Graph Matching by Evolutionary Optimisation

Richard Myers and Edwin R. Hancock

Department of Computer Science
University of York, York YO1 5DD.

Abstract. This paper considers how ambiguous graph matching can be realised using a hybrid genetic algorithm. The problem we address is how to maximise the solution yield of the genetic algorithm when the available attributes are ambiguous. We focus on the role of the selection operator. A multi-modal evolutionary optimisation framework is proposed, which is capable of simultaneously producing several good alternative solutions. Unlike other multi-modal genetic algorithms, the one reported here requires no extra parameters: solution yields are maximised by removing bias in the selection step, while optimisation performance is maintained by a local search step.

1 Introduction

In realistic settings graph-matching is invariably frustrated by structural error, and as a result it is not possible to locate an exact isomorphism. Early attempts at inexact matching used heuristics to reconcile dissimilar structures [1]. These heuristics have been augmented by information theoretic and probabilistic criteria [2]. Over the last ten years, there has been more interest in using statistical methods for inexact attributed graph-matching, instead of adopting a purely structural approach [3]. However, these methods invariably assume that there is a single best match. This approach works best where each of the attributes is distinctly located in the feature space, when it is possible to make a single minimum risk assignment. However, when the attributes are poor in the sense that there is considerable overlap in the feature space, restricting attention to the most probable assignments incurs a substantial risk of ignoring alternatives which are only slightly less good. For example, in a situation where there are two possible assignments with probabilities close to 0.5, it would be unwise to ignore the less likely one. This paper will demonstrate how to overcome this difficulty for graph matching using a multi-modal hybrid genetic algorithm.

The idea that genetic algorithms can be used to simultaneously find more than one solution to a problem was first mooted by Goldberg and Richardson in [4]. They attempted to prevent the formation of large clusters of identical individuals in the population by de-rating the fitness function. Other techniques include crowding [5], sequential niching [6], and distributed genetic algorithms [7]. A common feature of these approaches has been the necessity for extra parameters. Niching and crowding strategies typically require two or three extra

F.J. Ferri et al. (Eds.): SSPR&SPR 2000, LNCS 1876, pp. 397–406, 2000.

parameters to be controlled. These parameters are needed, for example, to determine when to de-rate the fitness of an individual, by how much, and the distance scale of the de-rating function. In distributed algorithms, it is necessary to decide how to arrange the sub-populations, their sizes, and under what conditions migration between them may occur. In [8], Smith and co-workers demonstrated a situation in which niching could occur in a standard genetic algorithm, without the need for any extra parameters.

The main conclusion that can be drawn from this literature is that the choice of selection operator plays a key role in determining the solution yield in multimodal or ambiguous optimisation problems. However, most of the reported work is focussed on toy or contrived problems. Our sim here is to undertake a systematic study of selection mechanisms for the problem of inexact graph matching. In particular, we will show how suitable algorithm modifications can improve solution yield without introducing any new parameters. We work with a hybrid genetic algorithm which incorporates a hill-climbing step, since Cross, Wilson and Hancock [9] have found that graph matching was only feasible with such an algorithm.

2 Bayesian Matching Criterion

The problems considered in this paper involve matching attributed relational graphs. An attributed relational graph is a triple, $G = (\mathbf{V}, \mathbf{E}, \mathbf{A})$, where $\mathbf{V}$ is the set of vertices or nodes, $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$ is the set of edges, and $\mathbf{A} \subset \mathbf{V} \times \Re^k$ is the set of measurement $k-$vectors relating to the original scene. Graph matching is the problem of establishing a correspondence between a data graph, $G_D = (\mathbf{V}_D, \mathbf{E}_D, \mathbf{A}_D)$, and a model graph, $G_M = (\mathbf{V}_M, \mathbf{E}_M, \mathbf{A}_M)$. This correspondence, $f : V_D \mapsto V_M \cup \{\phi\}$, is a labelling of the nodes in $\mathbf{V}_D$ with nodes from $\mathbf{V}_M$ or a special null label, ϕ, for unmatchable nodes.

In [10], Wilson and Hancock described a framework in which both neighbourhood structure and node attributes were combined in a single measure of matching consistency. The goal is to optimise the following *a posteriori* probability criterion

$$P(f|\mathbf{A}_D, \mathbf{A}_M) = \left\{ \prod_{(u,v)\in f} P(u, v|x_u, x_v) \right\} \left[\frac{1}{|\mathbf{V}_D|} \sum_{u\in\mathbf{V}_D} \sum_{S_v\in\Theta_v} \exp\left[-k_e D(\Gamma_u, S_v)\right] \right] \tag{1}$$

where the posterior matching probability, $P(u, v|x_u, x_v)$, is the probability of node u from the data graph matching node v in the model graph given their measurements, x_u and x_v. Structural constraints are captured by a dictionary of legal assignments, Θ_v, over the neighbourhood of each node, v, in the data graph. The constant $k_e = \ln\left(\frac{1-P_e}{P_e}\right)$ is defined in terms of the probability of matching error P_e. The distance function $D(\Gamma_u, S_v)$ measures the similarity of the current matching assignment to the data-graph neighbourhood Γ_u and the consistent matching configuration S_v drawn from the dictionary. In [11], Myers, Wilson and Hancock have shown that the Levenshtein distance is a good choice for the distance function D.

2.1 Measurement Ambiguity

The measurement information contributes to the matching criterion via the posterior matching probability, $P(u, v|x_u, x_v)$, which has yet to be defined. In [10], Wilson and Hancock defined it in terms of the Euclidean distance between attribute pairs for non-null mappings:

$$P(u, v|x_u, x_v) = \begin{cases} P_\phi & \text{if } v = \phi \\ (1 - P_\phi) \frac{\exp\left[\frac{-(x_u - x_v)^2}{2\hat{\sigma}_v^2}\right]}{\sum_{w \in V_M} \exp\left[\frac{-(x_u - x_w)^2}{2\hat{\sigma}_w^2}\right]} & \text{otherwise} \end{cases} \tag{2}$$

where P_ϕ is the prior probability of a null match, $f(u) = \phi$, which may be computed using the size difference of the graph to be $2\left\|\frac{|\mathbf{V}_D| - |\mathbf{V}_M|}{|\mathbf{V}_D| + |\mathbf{V}_M|}\right\|$, and $\hat{\sigma}_v^2$ is the estimated variance of x_v. This effectively regards the model graph node measurement, x_v, as a mean about which the data graph node measurement, x_u, varies with estimated variance $\hat{\sigma}_v^2$, under the null hypothesis that the two measurements are the same (because the nodes match). This approach requires the assumption that a data measurement is only likely to be statistically close to **one** of the model measurements. This is ideal when there is little overlap between classes, e.g. for possible angles of line-fragments segmented from a radar image. However, if there is significant overlap, e.g. in the average intensities of regions, such a scheme will not reflect these ambiguities in its classification of features.

The alternative is to compare the data measurements to the model measurements using an artificial scale. This can be done by considering the number of standard deviations separating the data measurement from its class mean under the null hypothesis that the nodes match. Table 1 gives an example of such a scale for the arbitrary classes “similar”, “comparable”, and “different”.

Table 1. Example Scale for Measurement Comparisons.

Class	Range of standard deviations from x_v
Similar	[0,1.0]
Comparable	(1.0,2.0]
Different	(2.0,∞]

Consider the standardised distance, $\Delta_{uv} = \|x_u - x_v\|/\hat{\sigma}_v$. The probability that Δ_{uv} lies within the interval $[a, b]$ is twice the standard Normal integral from a to b:

$$P(a \leq \Delta_{uv} \leq b) = \sqrt{\frac{2}{\pi}} \int_a^b e^{-\frac{1}{2}z^2} dz \text{erf}\left(\frac{b}{\sqrt{2}}\right) - \text{erf}\left(\frac{a}{\sqrt{2}}\right) \tag{3}$$

Each of the classes in table 1 corresponds to a separate interval which must be considered. Rather than introduce so many extra parameters, it is better to simplify the classification to "similar" if $\Delta_{uv} \in [0, a]$ and "dissimilar" otherwise. Thus, $P(u, v|x_u, x_v)$ can be defined as follows

$$P(u, v|x_u, x_v) = \begin{cases} P_\phi & \text{if } v = \phi \\ (1 - P_\phi)\dfrac{\exp\left[\frac{-(x_u - x_v)^2}{2\hat{\sigma}_v^2}\right]}{\sum_{w \in V_M} \exp\left[\frac{-(x_u - x_w)^2}{2\hat{\sigma}_w^2}\right]} & \text{if } a = 0 \\ (1 - P_\phi)P[\Delta_{uv} \leq a] & \text{if } \Delta_{uv} \leq a \\ (1 - P_\phi)(1 - P[\Delta_{uv} \leq a]) & \text{otherwise} \end{cases} \tag{4}$$

For convenience, the original unambiguous definition is used when $a = 0$. At the cost of an extra parameter, a, ambiguous measurements can now be handled. The important property of equation 4 is that when $a > 0$, it assigns the exact same probability to sets of mappings, thus enabling different alternatives to be considered.

3 Genetic Algorithms

Having defined an attribute model which captures the ambiguous nature of the raw image attributes, in this section we consider how to use genetic algorithms to recover multiple solutions to the graph-matching problem. In a standard genetic algorithm, selection is crucial to the algorithm's search performance. Whereas mutation, crossover and local search are all "next-state" operators, selection imposes a stochastic acceptance criterion. The standard "roulette" selection algorithm, described by Goldberg in [12], assigns each individual a probability of selection, p_i, proportional to its fitness, $\mathcal{F}_i$. The genetic algorithm used here allows the population, $\boldsymbol{\Psi}$, to grow transiently and then selects the next generation from this expanded population. Denoting the expanded population by $\boldsymbol{\Psi}_e$, the selection probability of the i^{th} individual, p_i, is given by

$$p_i = \frac{\mathcal{F}_i}{\sum_{j \in \boldsymbol{\Psi}_e} \mathcal{F}_j} \tag{5}$$

The algorithm then holds selection trials for each "slot" in the new population, for a total of $|\boldsymbol{\Psi}|$ trials. Since selection is with replacement, the constitution of the new population is governed by the multinomial distribution, and the copy number of a particular individual, $N(i)$, is distributed binomially:

$$P(N(i) = r) = \binom{|\boldsymbol{\Psi}|}{r} p_i^r (1 - p_i)^{|\boldsymbol{\Psi}| - r} \tag{6}$$

and so the expectation of $N(i)$, is $\mathrm{E}[N(i)] = |\boldsymbol{\Psi}| p_i$, and its variance is $\mathrm{Var}[N(i)] = |\boldsymbol{\Psi}| p_i (1 - p_i)$.

The search power of the standard genetic algorithm arises from the fact that if the individual in question is highly fit, p_i will be much larger than the average, and hence the expectation will be that the copy number will increase. This approach has two disadvantages. The first is that for small populations, sampling errors may lead to copy numbers very much higher or lower than the expected values. This can lead to premature convergence of the algorithm to a local optimum. In [13], Baker proposed "stochastic remainder sampling", which guarantees that the copy number will not be much different from the expectation by stipulating that $\lfloor \mathrm{E}[N(i)] \rfloor \leq N(i) \leq \lceil \mathrm{E}[N(i)] \rceil$. However, the larger the population, the less need there is for Baker's algorithm. The second disadvantage is that less fit individuals have lower expectations, and that the lower the fitness, the lower the variance of the copy number. In other words, less fit individuals are increasingly likely to have lower copy numbers. When $E[N(i)]$ falls below 1, the individual will probably disappear from the population. In general, the copy number variance decreases with decreasing fitness. Only when $p_i > 0.5$ does the variance decrease with increasing fitness. This occurs when the fitness of one individual accounts for at least half the total fitness of the population, i.e. when it is at least $|\boldsymbol{\Psi}_e| - 1$ times as fit as any other individual.

In short, the problem with roulette selection is that it imposes too strict an acceptance criterion on individuals with below average fitness. Several alternative strategies have been proposed to avoid this problem. "Sigma truncation", rank selection and tournament selection [12] all seek to maintain constant selection pressure by requiring individuals not to compete on the basis of their fitness, but on some indirect figure of merit such as the rank of their fitness, or the distance between their fitness and the average in standard units. Taking rank selection as a typical example of these strategies, the selection probabilities are assigned by substituting the rank of the individual for its fitness in equation 5, with the best individual having the highest rank. The implication of this is that the expected copy numbers of the best and worst individuals are given by:

$$\left.\begin{array}{r} \mathrm{E}[N(\mathrm{best})] = \frac{2|\boldsymbol{\Psi}|}{(|\boldsymbol{\Psi}_e|-1)} \\ \mathrm{E}[N(\mathrm{worst})] = \frac{2|\boldsymbol{\Psi}|}{|\boldsymbol{\Psi}_e|(|\boldsymbol{\Psi}_e|-1)} \end{array}\right\} \tag{7}$$

So, the expected copy number of the fittest individual differs from that of the least fit by a factor of $|\boldsymbol{\Psi}_e|$. Moreover, if $|\boldsymbol{\Psi}_e|$ is even moderately large, $\mathrm{E}[N(\mathrm{worst})]$ will be much less than 1. Indeed, $\mathrm{E}[N(i)]$ will be less than 1 for about half the population. Thus, under rank selection, less fit individuals are highly likely to disappear, even if they are quite good.

A second alternative to roulette selection is Boltzmann selection [14,15]. This strategy borrows the idea from simulated annealing, that at thermal equilibrium the probability of a system being in a particular state depends on the temperature and the system's energy. The idea is that as the temperature is lowered, high energy (low fitness) states are less likely. The difficulty with this analogy is that it requires the system to have reached thermal equilibrium. In simulated annealing, this is achieved after very many updates at a particular temperature. However, in a genetic algorithm this would require many iterations at each temperature level to achieve equilibrium, coupled with a slow "cooling". Within the 10 or so iterations allowed for hybrid genetic algorithms, equilibrium cannot even be attained, let alone annealing occur.

It would appear, then, that there is a tradeoff between premature convergence and the strength of the selection operator. The problem arises from the fact that expected copy numbers of fit individuals may be greater than one, while those of unfit individuals may be less than one. One way of preventing the increase in copy number of highly fit individuals is to use "truncation selection", as used in Rechenberg and Schwefel's evolution strategies [16,17]. Truncation selection would simply take the best $|\boldsymbol{\Psi}|$ individuals from the expanded population, $\boldsymbol{\Psi}_e$, to form the new population. The copy number of each individual is simply 1 or 0, depending on its rank. Although no individual may increase its copy number, the selection pressure might still be quite severe, since for the algorithm used in this paper, $|\boldsymbol{\Psi}_e|$ can be as large as $3|\boldsymbol{\Psi}|$. In other words, less fit individuals still disappear at an early stage. The fact that individuals never increase their copy number makes this a relatively weak search operator, and probably unsuitable for a standard genetic algorithm. However, the gradient ascent step is itself a powerful optimiser [10], and may be mostly responsible for the optimisation performance of the algorithm. If this is so, selection would be a much less important search operator for this hybrid algorithm than it is for standard genetic algorithms. It may therefore be beneficial to trade search performance for greater diversity.

3.1 Neutral Selection

The benefits of stochastic selection can be combined with the evenness of truncation selection by selecting without replacement. This strategy can be called "biased selection without replacement", since it is biased first in favour of fitter individuals, although it may also favour less fit ones.

The alternative is to abandon fitness based selection altogether, and rely on the local search step to do all the optimisation. If the genetic algorithm's rôle is explicitly limited to assembling a good initial guess for the local search operator, the selection probabilities can be assigned uniformly, i.e. $\forall_{i \in \boldsymbol{\Psi}_e} p_i = \frac{1}{|\boldsymbol{\Psi}|}$. This operator is called "neutral selection". Neutral selection without replacement can be implemented very efficiently by shuffling $\boldsymbol{\Psi}_e$ and choosing the "top" $|\boldsymbol{\Psi}|$ individuals. This strategy shares the advantage with truncation selection, that the minimum number of individuals are excluded from the new population, but also maintains the global stochastic acceptance properties of standard selection operators.

3.2 Elitism

Elitist selection guarantees that at least one copy of the best individual so far found is selected for the new population. This heuristic is very widely used in genetic algorithms. In [18], Rudolph showed that the algorithm's eventual convergence cannot be guaranteed without it. The elitist heuristic can be modified in two ways to help maintain diversity. First, it seems natural that if the goal is to simultaneously obtain several solutions to the problem in hand, several of the fittest individuals should be guaranteed in this way. This is called "multiple elitism". Second, if one wishes to avoid losing too many unfit individuals, the *worst* individual can also be granted free passage to the new population. This is called “anti-elitism”. These heuristics, together with the selection strategies discussed earlier, are evaluated at the end of section 4.

4 Experiments

This experimental study establishes the suitability of the hybrid genetic algorithm for ambiguous graph matching, and compares the selection strategies discussed in the previous section. The algorithm was tested on 30-node synthetic graphs Data graphs were generated by randomly perturbing the node attributes, and then duplicating 10% of the nodes and perturbing their attributes. The intention was to simulate segmentation errors expected of region extraction, such as the splitting of one region into two similar ones.

4.1 Comparative Study

A comparative study was performed to determine the best algorithm for ambiguous matching. The algorithms used were the hybrid genetic algorithm with and without mutation, crossover or both (hGA, hGA-m, hGA-x and hGA-xm),[1] a hybrid version of Eshelman’s CHC algorithm [19] (hCHC), and plain gradient ascent (HC). The experimental conditions are summarised in table 2.

Table 2. Algorithms for Graph Matching. Each algorithm, apart from HC, made approximately 700,000 fitness evaluations. Abbreviations: hGA = hybrid genetic algorithm, hGA-m = hGA without mutation, hGA-x = hGA without crossover, hGA-xm = hGA with neither mutation nor crossover, hCHC = hybrid CHC, and HC = gradient ascent (hillclimbing).

	hGA	hGA-m	hGA-x	hGA-xm	hCHC	HC
Population	50	50	120	120	100	1
Iterations	5	5	5	5	5	10
Crossover	Uniform	Uniform	Uniform	Uniform	HUX	n/a
Cross rate	0.9	0.9	0.0	0.0	1.0	n/a
Mutate rate	0.3	0.0	0.3	0.0	0.35	n/a

Each of the algorithms listed in table 2, except HC, was run 100 times. Since HC is deterministic, it was only run once per graph. The results for the different graphs were pooled to give 400 observations per algorithm. Algorithm performance was assessed according to two criteria. The first was the average fraction of correct mappings in the final population. The second was the proportion of distinct individuals in the final population with more than 95% correct mappings. The results are reported in table 3.

[1] These should be regarded as different algorithms, not merely different parameter sets for a genetic algorithm, because a genetic algorithm with no crossover or mutation is fundamentally different from one which has these operators. For example, the hGA-xm algorithm is really just multiple restarts of gradient ascent with a selection step.

Table 3. Graph Matching Results. Standard errors are given in parentheses. Abbreviations: hGA = hybrid genetic algorithm, hGA-m = hGA without mutation, hGA-x = hGA without crossover, hGA-xm = hGA with neither mutation nor crossover, hCHC = hybrid CHC, and HC = gradient ascent (hillclimbing).

Algorithm	Average Fraction Correct	Average Fraction Distinct
hGA	0.90 (0.0044)	0.078 (0.0019)
hGA-m	0.88 (0.0051)	0.040 (0.0012)
hGA-x	0.84 (0.0052)	0.044 (0.00094)
hGA-xm	0.76 (0.0068)	0.013 (0.00036)
hCHC	0.92 (0.0042)	0.012 (0.00033)
HC	0.97 (n/a)	n/a

At first sight, pure gradient ascent appears to outperform all the other algorithms. The reason for this is partly that the gradient ascent algorithm starts from an initial guess in which about 50% of the mappings are correct, whereas the other algorithms start with random initial guesses. More importantly, the final population of a genetic algorithm typically contains solutions much better and worse than the average. Thus, this comparison is not really fair: a fairer comparison of optimisation performance comes from considering hGA-xm, which is multiple random restarts of gradient ascent. Furthermore, gradient ascent is deterministic, and therefore always gives the same result, but the genetic algorithm is stochastic and may do significantly better or worse than gradient ascent. Indeed, the genetic algorithm occasionally found matches with 100% correct mappings. However, the performance of gradient ascent alone suggests that for unambiguous problems, genetic algorithms may not necessarily be the method of choice. Apart from pure gradient ascent, the best optimiser was hCHC, which is only slightly better than hGA. The results for hGA-m and hGA-x indicate that crossover and mutation are playing an active part in the optimisation process. Turning to the fraction of distinct individuals with over 95% correct mappings, it is clear that pure gradient ascent is incapable of finding more than one solution. The hCHC algorithm appears to converge to fewer solutions than the hGA algorithm. In all, the hybrid genetic algorithm (hGA) combines strong optimisation performance with the highest solution yield, and it is this algorithm which will be the subject of the remainder of this study.

4.2 Selection

Two sets of experiments were conducted to evaluate different selection strategies with and without elitism. In each case, a hybrid genetic algorithm was used, with a population size of 20, and uniform crossover was used at a rate of 1.0. The mutation rate was fixed at 0.4. The first set of experiments used 20, 30, 40 and 50 node graphs, and for these the population size was set to 10, and the algorithm run for 5 iterations. The second set of experiments used four 30 node graphs, with a population size of 20 and 10 iterations. Five different selection strategies were compared: they were standard roulette, rank, and truncation selection, and neutral and biased selection without replacement. Five combinations of elitist heuristics were considered: they were no elitism, single elitism, multiple elitism, anti-elitism, and a combination of multiple and anti-elitism. The experimental design was therefore a 5x5x4 factorial with 100 cells. The first set of experiments had 40 replications for a total of 4000 observations; and the second set had 50 replications for 5000 observations. Figures 1b and 1a summarise the results.

Both plots show that neutral selection without replacement produced the best yields, and that truncation selection produced the worst. Biased and roulette selection strategies gave similar results, and were both outperformed by rank selection. Linear logistic regression analysis of both data sets confirmed this ranking of selection strategies. The results for elitism heuristic were not so convincing. It is questionable whether elitism has any overall effect: the regression analysis of the second data set found no significant effect of varying the elitism strategy. The analysis of the first data set did show that either standard (single) or multiple elitism gave significantly better yields, but that the effect was small.

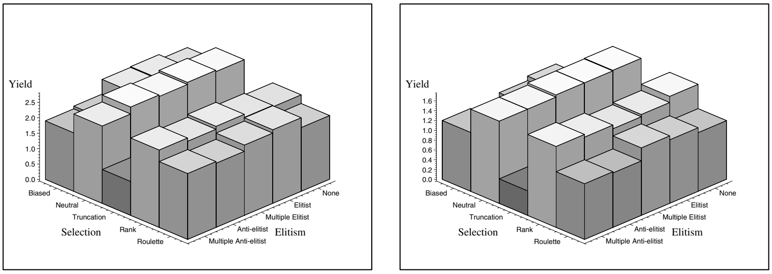

Fig. 1. Average Yields versus Selection and Elitism . Data from all four graphs has been pooled.

5 Conclusion

This paper has presented a method of matching ambiguous feature sets with a hybrid genetic algorithm, which does not require any additional parameters to achieve multimodal optimisation. The first contribution made was to develop an attribute process for ambiguous feature measurements. The second contribution has been to explore the hybrid genetic algorithm as a suitable optimisation framework for ambiguous graph matching. If most of the optimisation is undertaken in the gradient ascent step, the tradeoff between effective search and maintenance of diversity, which must be made in choosing a selection operator for standard genetic algorithms, can be abandoned. Neutral selection without replacement maximises the diversity in the next generation with no regard to individuals' fitnesses.

References

1. L. G. Shaprio and R. M. Haralick. A metric for comparing relational descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:90–94, 1985.
2. A. K. C. Wong and M. You. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:599–609, 1985.
3. J. Kittler, W. J. Christmas, and M. Petrou. Probabilistic relaxation for matching problems in computer vision. In *Proceedings of the* 4th *IEEE International Conference on Computer Vision*, pages 666–673, 1993.
4. D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the* 2nd *International Conference on Genetic Algorithms*, pages 41–49, 1987.
5. W. Cedeño, V. R. Vemuri, and T. Slezak. Multiniche crowding in genetic algorithms and its application to the assembly of DNA restriction-fragments. *Evolutionary Computation*, 2:321–345, 1995.
6. D. Beasley, D. R. Bull, and R. R. Martin. A sequential niche technique for multimodal function optimisation. *Evolutionary Computation*, 1:101–125, 1993.
7. M. Gorges-Schleuter. ASPARAGOS: A parallel genetic algorithm for population genetics. In *Parallelism, Learning, Evolution. Workshop on Evolutionary Models and Strategies*, pages 407–418, 1991.
8. R. E. Smith, S. Forrest, and A. S. Perelson. Searching for diverse, cooperative populations with genetic algorithms. *Evolutionary Computation*, 1:127–149, 1993.
9. A. D. J. Cross, R. C. Wilson, and E. R. Hancock. Inexact graph matching using genetic search. *Pattern Recognition*, 30:953–970, 1997.
10. R. C. Wilson and E. R. Hancock. Structural matching by discrete relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:634–648, 1997.
11. R. Myers, R. C. Wilson, and E. R. Hancock. Efficient relational matching with local edit distance. In *Proceedings of the* 14th *International Conference on Pattern Recognition*, pages 1711–1714, 1998.
12. D. Goldberg. *Genetic Algorithms in Search, Optimisation and Machine Learning.* Addison-Wesley, 1989.
13. J. E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the* 2nd *International Conference on Genetic Algorithms*, pages 14–21, 1987.
14. D. E. Goldberg. A note on Boltzmann tournament selection for genetic algorithms and population-based simulated annealing. *Complex Systems*, 4:445–460, 1990.
15. A. Prügel-Bennett and J. L. Shapiro. An analysis of genetic algorithms using statistical physics. *Physical Review Letters*, 72:1305–1309, 1994.
16. I. Rechenberg. *Evolutionsstrategie - Optimierung Technischer Systeme nach Prinzipien der biologischen Information.* Fromman Verlag, 1973.
17. H-P. Schwefel. *Numerical Optimization of Computer Models.* Wiley, 1981.
18. G. Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 5:96–101, 1994.
19. L. J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms*, volume 1, pages 265–283. Morgan Kaufmann, 1991.