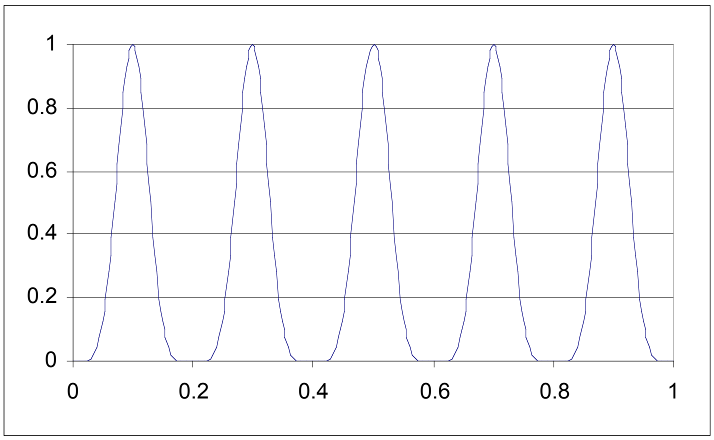
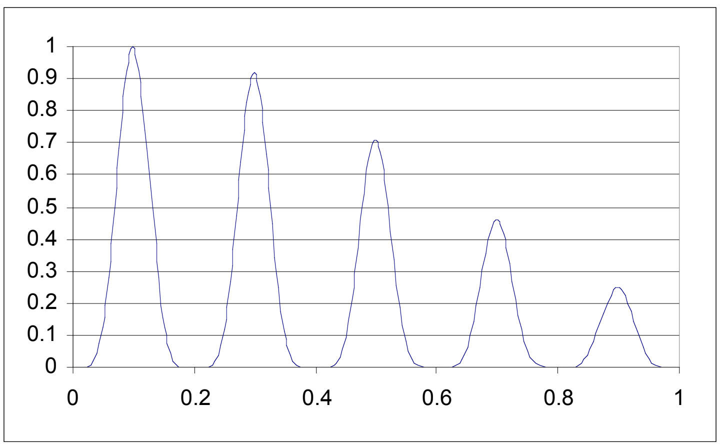
Niche and Speciation

- Environments often provide more than one solution of equal quality
    - From an optimization context these problems are denoted as Multimodal;
- So far, our basic GA does ***not*** show any evidence to support,
    - Stable subpopulations of strings (species)
        - Genotypic characteristic;
    - Stable identification of sub-domains associated with a function (niches)
        - Phenotypic characteristic;
- With respect to figure 1, our basic GA would result in identification of one of the peaks;



(a)



(b)

Fig 1. Simple Multimodal test functions.

- More generally (Rodgers and Prugel-Bennett 1999),
  - Change in mean fitness at generation '*t*' is proportional to fitness variance;
    - Variance reduces due to,
      - Selection pressure – rate of production of fitter members;
      - Genetic drift – stochastic changes in gene frequencies as opposed to a greedy selection of individuals;
        - Consider two 'niches' with same fitness and equal representation (50:50);
        - Stochastic nature of selection might result in sampling of these niches at the next generation such that one niche as slightly more representation (49:51);
        - Process now snowballs…
- Convergence to a single peak is particularly apparent in small populations.
- In multimodal problems, figure 1, interested in maintaining a distribution of solutions (niches) in proportion to quality of 'local' fitness.

Example – Simple 2-armed bandit problem

- Consider a 2-armed bandit,
  - Each arm has a different payoff.
    - $75 expected payoff on RH arm
    - $25 expected payoff on LH arm
  - Player is not aware of these payoffs
- Let there be 100 players, each receiving full payoff from the chosen arm at each play;
  - Rapidly expect that distribution of players will move to the RH arm.
- How might we remodel the game to encourage a proportional allocation of individuals to each arm?

Role of Selection

- Mating in our simple GA is effectively random, biased in favor of fitter individuals;
- (biological) speciation requires that individuals mate with individuals from the same species;
- Consider case of Figure 2.
- Normal binary encoding results in,

  - LH dominated by individuals predominantly composed from 0's;
  - RH dominated by individuals predominantly composed from 1's;
- Proportional selection will result in the most fit individuals being chosen for the parents.
  - What will be the most typical child?
    - Result are 'lethals'.
- Need some method to encourage species based selection;

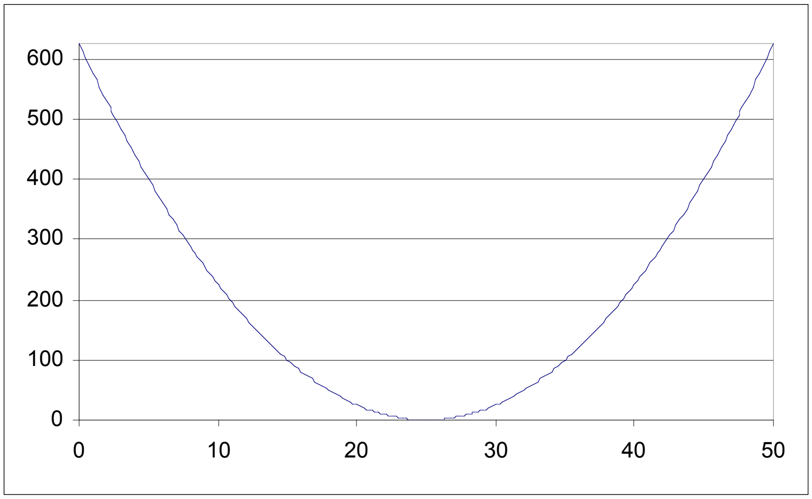


Fig 2. Case for mating restrictions

Niche Selection Methods in Binary Genetic Algorithms

1. Preselection (Cavicchio, 1970), (Mahfound, 1992)

   IF f (child) > f (parent)

   THEN parent ← child

2. Crowding (De Jong, 1975), (Mahfound, 1992)

   Replacement based on string similarity

   (i) Individual compared to random sub-population of *cf* members (crowding factor);

   (ii) New string replaces the sub-population individual with closest similarity (e.g. Hamming distance (encoded genotype) or Euclidean distance (decoded genotype)).

   De Jong demonstrated support of multiple populations when using population diversities defined by *cf* = 2 or 3.

Later work by Mahfound indicated that the algorithm could be significantly improved (Mahfound, 1992).

3. Sharing function (Miller and Shaw, 1995), (Deb and Goldberg, 1989)

- a function is used to explicitly define the degree of sharing
  - maps genotypic similarity (similarity) into the degree of sharing;
  - Shared fitness =

$$\frac{\text{fitness of individual}}{\text{Total degree of sharing}}$$

  - $f_s(x_i) = \frac{f(x_i)}{\sum_{j=1}^{n} s(d(x_i, x_j))}$
  - where, $s(\cdot)$ is the predefined sharing function; and $d(\cdot)$ is the phenotypic or genotypic similarity (distance) function.
  - Basic properties for such a sharing function might include,
    - As $d(x_i) \rightarrow d(x_j)$ then, $s(\ d(x_i, x_j)\ ) \rightarrow 1$
    - Likewise, as dissimilarity increases then $s(\ d(x_i, x_j)\ ) \rightarrow 0$
  - Implies that we have some concept of what constitutes 'similar' and what constitutes 'dissimilar'.
    - Results in the introduction of an *a priori* scaling factor, σ, or
    - $sh(d(\cdot)) = \begin{cases} 1 - (d(\cdot)/\sigma)^{\alpha} & if\ d(\cdot) \le \sigma \\ 0 & otherwise \end{cases}$
    - where α controls the slope of the sharing function, Figure 3.
- Notes
  - Selection of appropriate scaling factors, σ, is problem dependent;
    - Implies that the number of peaks must be known *a priori* !
  - Computational overhead in estimating genotypic distances over all individuals;

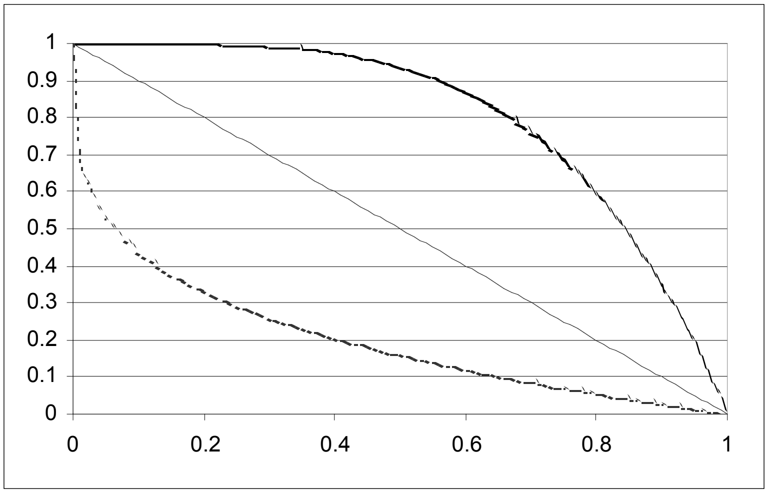


Fig 3. Power law sharing functions – *x*-axis denotes $d(\cdot)/\sigma$; *y*-axis is the degree of sharing. $\alpha = 1$ corresponds to a linear relation; $\alpha > 1$ implies magnification.

Pragmatics

- Establishing Distance metric on binary spaces is relatively straightforward (Hamming distance normalized by the number of bits).
- What might you use for expressing the (genotypic) distance on individuals expressing a discrete permutation of indexes (e.g. as in the indirect encoding for the scheduling problem)?

  E.g. what is the distance between two individuals for a TSP problem of the form,

  [1 2 3 4 5] and [3 4 5 1 2]

  i.e. alignment now becomes an issue where this is problem dependent.

Competitive Fitness sharing in Genetic Programming (Rosin and Belew, 1997)

- No reason why an individual should 'solve' all instances of a task equally
- No reason why the path to solving exemplars within training partition should reward a single monolithic path from current 'best' individual to ultimate solution
- Maintaining multiple candidate solutions implies that the eventual 'best' individual might evolve through constructing a composite of multiple candidate solutions.
- A (competitive) fitness sharing function discounts cherry picking to also support individuals that ***label less in total***, but ***more of the outstanding instances***.
- $$s_i = \sum_{k=0}^{P-1} \left(\frac{G(y_i, p_k)}{\sum_{j=0}^{N-1} G(y_j, p_k)} \right)^{\alpha}$$
- where:
  - P is the set of training instances;
  - $G(y_j, p_k)$ is the domain specific function returning the ***outcome*** from GP individual y_i on training instance p_k. Hence under a classification domain, the outcome might be either 1 for a correctly classified instance or 0 on an incorrectly classified instance;[1]
  - $\sum_{j=0}^{N-1} G(y_j, p_k)$ provides the renormalization factor i.e., for training instances that most (few) individuals in the population get correct, the sum will be high (low). Some individuals might adopt a policy in which few correct outcomes result, but they may still be fit overall if the majority of the population get these instances wrong.
  - α is a positive integer. Values typically employed include 2 and 3 (see (Rosin and Belew, 1997) and (Lichodzijewski and Heywood, 2008) respectively).
- Naturally, such a function says nothing about how to combine independent individuals into a cohesive solution. For one example of this see the approach of context learning though bidding (Lichodzijewski and Heywood, 2008) or in the case of genotypic 'tags' (Stanley and Miikkulainen 2002).

[1] Note that the outcomes of 0 and 1 are independent from the number of classes in the data set.

Multi-objective methods based on a Pareto Dominance Relation (Eiben and Smith, 2003), (Deb, 2001)

- Performance within a multi-objective context implies that fitness is expressed in terms of,
  - More than one objective;
  - The objectives are not necessarily complementary.
- Consider a clustering application; Fitness might be expressed in terms of,
  - Minimizing the overlap in clusters;
  - Maximizing the density of points associated with a cluster;
  - Minimizing the number of features necessary to support a cluster.
- Bottom line is that at some point we need make a comparison between competing solutions a multi-dimensional error surface.
- Pairwise ***dominance*** relation provides the basic methodology for doing so.
  - Consider 2 candidate solutions, *A* and *B*.
  - Solution *A* is said to dominate *B* if for all *n* objectives the score of *A* is better than that for *B* in *at least one case*, and *equally good* in the other cases.
  - Formally, ***Pareto Dominance*** condition is expressed by the '$\succ$' operator as follows,
    - $A \succ B : \forall i \in \{1,...,n\}\, a_i \geq b_i, and \exists i \in \{1,...,n\}\, a_i > b_i$
  - ***nondominated*** (or *noninferior*) solutions are those that cannot be improved upon without negatively impacting the performance on one of the '*n*' objectives, resulting in a dominated solution.
  - The set of nondominated solutions constitutes the Pareto Set.
- The set of nondominated solutions identifies a 'front' in the multiobjective search space, Figure 4.

Example algorithm: MOGA or Multiobjective GA **(Deb, 2001)**

- One of the first established examples of a GA based on a Pareto dominance fitness function applied to multiobjective problems was the MOGA algorithm.
    - Evaluate score of each individual under all '*n*' objectives.
    - Raw fitness of individual *A*, *f*(*A*), is proportional to the number of members dominated by *A*.
    - Fitness sharing (between individuals of same rank) and fitness proportional selection provide diversity mechanism.
- Such an algorithm is nonelitist, resulting in the potential loss of good solutions.
- Relies on the correct selection of appropriate niching parameters.
- Recent interest in elitist strategies in which some form of archiving is used to maintain a set of nondominated solutions (see later group presentation).
    - Specific examples include NSGA-II (Deb *et al*., 2002), SPEA (Zitzler *et al*., 2001), PCGA (Kumar and Rockett, 2002), and ε-dominance (Deb *et al*., 2005).
- From the perspective of 'model based' methods – Genetic Programming, SVM, Decision Trees, Neural Networks – Pareto multi-objective methods are appearing with increasing frequency (Jin, 2006)

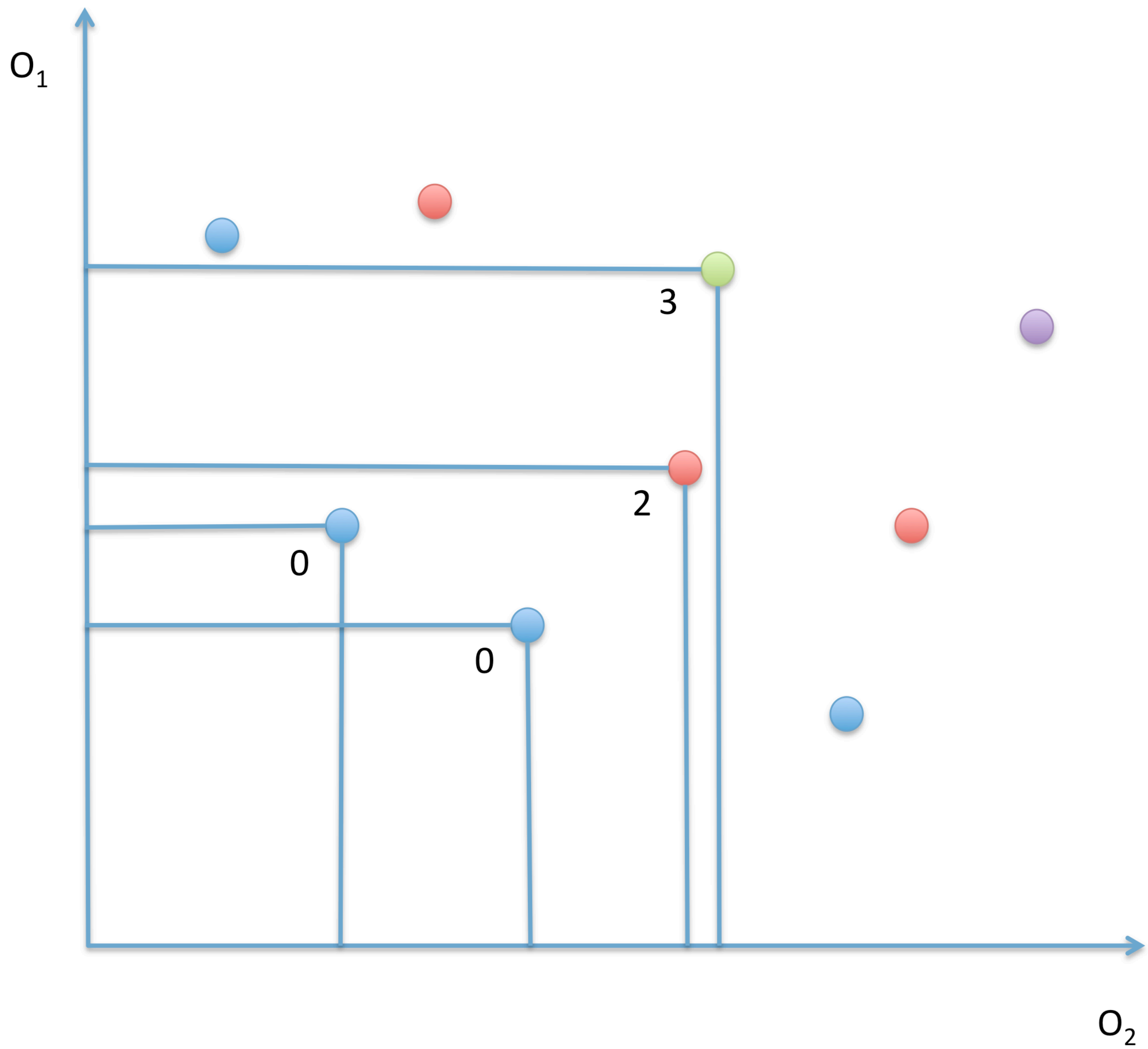


Figure 4. Dominated / Nondominated solutions to an example dual objective minimization problem. Minimization is better.

Further Reading

- Deb K., (2001) Multi-objective Optimization using Evolutionary Algorithms. John Wiley and Sons.
- Deb K., Goldberg D.E. (1989) An Investigation of Niche and Species Formation in Genetic Function Optimization. Proceedings of the 3rd International Conference on Genetic Algorithms, 1989, pp 42-50.

- Deb K., Mohan M., Mishra S. (2005), Evaluating the ε-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. Evolutionary Computation 13(4), pp 501-526.
- Deb K., Pratap A., Agarwal S., Meyarivan T. (2002) A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation. 6(2) pp 182-197.
- Eiben A.E., Smith J.E. (2003) Introduction to Evolutionary Computing, Natural Computing Series. Springer-Verlag. 2003, ISBN 3-540-40184-9.
- Jin Y. (ed) (2006) Multi-Objective Machine Learning. Studies in Computational Intelligence 16. Springer-verlag.
- Kumar R., Rockett P. (2002) Improved Sampling of the Pareto-Front in Multiobjective Genetic Optimization by Steady-State Evolution: A Pareto Converging Genetic Algorithm. Evolutionary Computation 10(3), pp 283-314.
- Lichodzijewski P., Heywood M.I. (2008) Managing Team-based Problem Solving with Symbiotic Bid-Based Genetic Programming. ACM GECCO, pp 363- 370.
- Mahfound S.W. (1992) Crowding and Preselection Revisited. Parallel Problem Solving from Nature – II, 1992. pp 27-36. Also available as IlliGAL Technical Report 92004; http://www-illigal.ge.uiuc.edu/techreps.php3
- Miller B.L., Shaw M.J. (1995) Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization. IlliGAL Technical Report 95010, December 1995; http://www-illigal.ge.uiuc.edu/techreps.php3
- Rodgers A., Prugel-Bennett A. (1999) Genetic Drift in GA Selection Schemes, IEEE Transactions on Evolutionary Computation. 3(4), pp 298-303.
- Rosin C.D., Below R.K. (1997) New methods for Competitive Coevolution. Evolutionary Computation. 5(1): 1-29.
- Stanley K.O. and Miikkulainen R. (2002) Evolving Neural Networks through augmenting topologies. Evolutionary Computation. 10(2):99-128
- Zitzler E., Laumanns M., Thiele L. (2001) SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK-Report 103, Swiss Federal Institute of Technology (ETH).