Complex Systems **5** (1991) 327–333

A Two-Dimensional Genetic Algorithm for the Ising Problem

Charles A. Anderson
Kathryn F. Jones
Jennifer Ryan
University of Colorado at Denver, Denver, CO 80217-3364, USA

Abstract. The genetic algorithm is a powerful heuristic for the solution of hard combinatorial problems and has been investigated by numerous authors. Many problems, arising for example in communication networks, possess strong two-dimensional characteristics. We describe a genetic algorithm with a new crossover operator called *block-uniform crossover*, which exploits the two-dimensional character of a problem. The concept was tested on a version of the Ising model, which is important in physics. This new algorithm outperforms genetic algorithms with traditional crossover operators in all trials.

1. Introduction

Genetic algorithms (GAs) are designed to mimic biological evolution. As part of the class of heuristic methods, they do not always produce optimal solutions; rather, they are intended to provide reasonable answers to problems where the known methods of obtaining optimal answers are unreasonably time-consuming or otherwise unsuitable. They have been successfully applied to various problems that could not have been readily solved by more conventional techniques. Genetic algorithms were invented by John H. Holland and first widely disseminated in reference [5]. Holland's formulation was motivated by the observation that sexual reproduction in conjunction with the pressure of natural selection has resulted in the development of highly adapted species in the process of natural evolution. The principle impact of Holland's work is the central role played by the crossover operator as the underlying discovery mechanism.

In many problems of interest—for example, in communication networks—there is a strong two-dimensional nature to problem solutions. In the standard genetic algorithm, a chromosome-like bit vector is usually taken to represent a problem solution. Thus two-dimensional information is generally lost or not taken into account in designing a crossover operator. In this study, a number of crossover operators that attempt to exploit the two-dimensional nature of a problem space are considered. The Ising problem was chosen as a test bed for the algorithm because of its strong two-dimensional structure, the ease of manipulating the data structures, and known optimal solutions to certain test problems. The most effective of these new operators is a generalization of the standard uniform crossover, which has been observed by many authors to be an extremely reliable operator [1, 8]. Results indicate that this new crossover operator is quite effective in exploiting the two-dimensional structure of the Ising model. The purpose of this research was solely to study the effect of the two-dimensional crossover. We have not concerned ourselves with comparisons to other algorithms, but merely the comparison of the various crossovers within the GA.

In section 2 we discuss the Ising model. In section 3 we discuss the genetic algorithm and our implementation with a population of two-dimensional chromosomes. Finally, in section 4 we present comparison data on several problems.

2. The Ising spin problem

The Ising model was first studied by Ernst Ising [6] in the early twenties. It has remained an important model in physics in the study of thermodynamic properties, magnetic spin correlations, phase transitions, and other applications. For a more detailed treatment of the two-dimensional model that we have used to study our algorithms, see [7]. The basic model is a square lattice or grid of elements ("spin" variables) σ_α, which have two states, $+1$ and -1 (up, down or positive, negative). There is a mutual interaction energy between nearest neighbors α and α' in the grid defined by

$$-E(\alpha,\alpha')\sigma_\alpha\sigma_{\alpha'},$$

which is negative if the spins are parallel and positive if not. (Note: $E(\alpha,\alpha') = 0$ unless α and α' are nearest lattice neighbors. Typically $E(\alpha,\alpha')$ is $+1$ or -1 when α and α' are neighbors.) Additionally a spin may interact with an external magnetic field H with energy

$$-H(\alpha)\sigma_\alpha.$$

The general problem is to find an assignment of the spin variables that minimizes the total energy function for a given matrix E of interaction energies and external field H. Frequently one assumes that the energies affecting rows are the same as those affecting the columns of the grid, and that the external input is uniform or has a small percentage of -1s. In our testing, we have assumed that $H \equiv 0$.

3. The genetic algorithm

The genetic algorithm introduced by John Holland in 1975 is a heuristic that solves hard combinatorial problems by simulating biological evolution. The algorithm (GENSPIN) that we have developed is based on the standard

Holland-style genetic algorithm. A thorough overview of genetic algorithms can be found in [4]. The standard GA proceeds in three phases:

1. An initial population is generated, usually randomly or greedily. In the standard GA, the individuals of the population are represented by chromosome-like bit vectors or genotypes. The population size remains fixed from generation to generation and is typically between 50 and 200 individuals.

2. The population is reproduced according to the relative fitness (determined by an evaluation function that is application dependent) of an individual in the population. Reproductive trials are allocated according to the fitness of the genotype relative to the fitness of the remaining population.

3. Offspring are generated by the exchange of information from the parent genotypes. This is typically accomplished by the use of crossover operators. The simplest variant selects two parents at random from the gene pool as well as a crossover position within the binary encoding. The parents exchange information in the portion of the string to the right (say) of the crossover point. This is called 1-point crossover.

4. Steps 2 and 3 are repeated for a fixed number of generations or until the process converges.

The GENSPIN algorithm proceeds as follows:

1. An initial population is generated randomly. The individuals of the population are $n \times n$ grids of spins that are $+1$ or -1.

2. The total energy of each individual (grid) is calculated for a fixed matrix E of interaction energies between nearest neighbors, and then a fitness function is defined that assigns a probability of reproduction to the most fit individual that is proportionately larger than the probability of reproduction of the least fit individual. GENSPIN contains a modification here that exploits some a priori knowledge of the problem space. Since the fitness of each individual depends on the parity of nearest neighbors, individuals with equal fitness occur in pairs. For each individual with a given fitness, another individual with the same fitness can be obtained by exchanging $+1$s and -1s. Thus we introduced a "switching" operator that tends to drive the algorithm toward solutions with a majority of $+1$s.

3. GENSPIN uses a variety of crossover operators: standard ones such as two-point and uniform, and others that attempt to exploit the two-dimensionality of the problem space. These are vertical and horizontal band operators, block operators, and (the most successful) a generalization of uniform crossover called block-uniform crossover.

4. GENSPIN has been tested on generation sizes of 150, 200, and 300.

Initial trials were run with a version of GENSPIN that used none of the standard techniques to produce better solutions in a GA, such as using mutation as an infrequent operator to preserve population diversity. A later version incorporated mutation operators as well as a test for convergence. In this latter version when the population was approaching convergence, an "inversion" operator was applied to introduce more diversity into the population. This inversion operator, which we call "complementation," is described in section 4 below.

4. Comparison of crossover operators

The primary emphasis of this research is to study the behavior of various crossover operators for chromosomes that are represented by an $n \times n$ grid. Thus, we deliberately designed a "plain vanilla" Holland-style genetic algorithm incorporating no special techniques to improve problem solutions.

In initial trials, we tested six different crossover operations:

1. Two-point: Here the grid is treated as a vector of length n^2, two random points are generated, and information is exchanged between the points according to fixed probabilities.
2. Uniform crossover: Information is exchanged between individual grid points based on a fixed probability.
3. Vertical band crossover: Two random numbers are generated, and information inside the vertical region of the grid determined by the numbers is exchanged based on a fixed probability.
4. Horizontal band crossover: This is the same as vertical, except exchange occurs within horizontal bands.
5. Block crossover: Exchange occurs inside a rectangular block whose size and location are determined randomly.
6. Mixed crossover: The operator, taken from the above five types, is determined randomly.

We tested these operators on a 10×10 grid with $E(\sigma, \sigma') \equiv 1$ for all nearest neighbors σ and σ'. For this problem, the optimum solution can be determined exactly. We learned that these two-dimensional operators did not perform as well as uniform crossover or even as well as two-point crossover for various choices of probabilities. Thus, as noted in various places in the GA literature [2, 3], two-point and uniform are very robust crossover operators even though they do not exploit the structure of the problem space. However, with the introduction of the block-uniform crossover operator, we obtained significantly improved results.

The block-uniform crossover divides the grid into $i \times j$ blocks where i and j are chosen randomly. Then each block of one parent is interchanged randomly with the corresponding block of the second parent based on a preassigned percentage. The best results were obtained with 50% and 60% block exchange. GENSPIN seems to be relatively insensitive to the percentage of total crossover. We ran the remaining trials with a 75% chance that two individuals would exchange information.

In generating an original population of size 100, the best solution obtained is on average 60% of the optimal solution. Using uniform crossover, we obtain on average a best solution that is 79% optimal and is found in generation 111. With block-uniform crossover, a best solution that is 80.5% optimal on average is found in generation 110. The best solution found in all trials was 90% optimal and was found by using block-uniform crossover.

In an attempt to deter premature convergence we introduced a "complementation" operator, which exchanged +1s and −1s in a small random proportion of the population. This process mimics standard "inversion" processes that are often introduced in GAs to create diversity in a stagnating population. This had a devastating effect on GENSPIN for both uniform and block-uniform crossover, and reduced the best solutions to 66.5% and 69% optimality, respectively.

Finally, we tested the initial version of GENSPIN with the "switching" operator (described in section 3 above). This has the effect of cutting the search space in half. (Note: The search space is of size 2^{n^2}.) The results were dramatic. Both uniform and block-uniform crossover produced much better solutions. On average, uniform gave solutions 86% optimal and block-uniform gave solutions 89% optimal. The best solution was 97% optimal and was found by GENSPIN with block-uniform crossover. The best solutions were still found on average in generation 110.

Observation of our results at this point led us to conclude that block-uniform crossover was exploiting the two-dimensional structure of the Ising model but, though superior to uniform crossover, failing to produce optimal solutions. The algorithm was converging too fast before an optimal solution was obtained. In order to forcibly prevent premature convergence we introduced a measure of "variation" in the population. We fixed a factor p. If every member of the population had a value within p% of the "best-so-far," a concerted effort was made to introduce new genetic material into the population. This was accomplished using the "complementation" inversion operator described above on a randomly chosen (but large percentage) of the population. This procedure had the desired effect of forcing the GA to continue its exploration. In addition we introduced a straightforward hill-climbing procedure as a post-processor to improve the best solution found by the GA. These enhancements to the algorithm improved results significantly.

In particular, on the test problem described above, the results were 87.5% of optimal on average using uniform crossover. Using the block-uniform crossover, the results were 97.8% of optimal on average, and the optimal solution was found in 40% of the runs. We tested the final version of our algorithm on a variety of test problems to ensure that our earlier observations were not problem dependent.

In the table below, we summarize some of the test data from the comparison of GENSPIN using block-uniform crossover versus using uniform crossover. The parameter settings for all tests are the following: population size is 150 and the number of generations is 300. We always do crossover on the parent genotypes. The likelihood of crossover of a block in block-uniform crossover, or of a bit in uniform crossover, is 50%. In previous tests we determined that GENSPIN is relatively insensitive to the total amount of crossover, but gives slightly improved results when the percentage of block (or bit) crossover is between 50% and 60%. The test problems are 10×10 grids and the interaction energy matrices for test problems are randomly generated. The results reported are averages for five trials. The previous best reported is the result of applying a simple tabu search to the same problem. This comparison is given only to verify that the GA was finding near-optimal solutions. The best value found by randomly generating 1500 solutions is reported in the last column below.

Problem	Crossover	Average generation found	Average best found	Average best with hillclimb	Previous best	Random
10a	Block	264	−164	−172	−180*	
10a	Uniform	34	−127	−135		
10b	Block	70	−125	−126	−122	−46
10b	Uniform	25	−112	−115		
10c	Block	74	−122	−123	−130	−42
10c	Uniform	32	−98	−109		
10d	Block	56	−120	−123	−126	−48
10d	Uniform	31	−108	−111		

* known optimal solution

GENSPIN produces solutions that compare favorably with those found by the tabu search. The results of these tests are encouraging. Among various types of crossover operators designed to exploit the two-dimensional nature of a solution, we have identified an operator that is quite successful. Incorporating the block-uniform crossover operator has resulted in a GA that produces solutions superior to a GA that uses a traditional crossover operator.

5. Conclusion

In this study we investigated several crossover operators designed to incorporate two-dimensional information in a solution space, and we compared the performance of the genetic algorithm using these operators to traditional crossover operators. The block-uniform crossover operator has proved to be superior to all other operators tested in a genetic algorithm to solve the Ising problem. This suggests that, in problems with strong two-dimensional characteristics, performance in a genetic algorithm can be improved by exploitation of these characteristics using a variation of block-uniform crossover. In a problem without the strong nearest neighbor properties of the Ising model, partitioning of the domain may be the appropriate construct for using block-uniform crossover. Preliminary work has been done on designing a GA using this variation of block-uniform on a network topology problem. Initial tests indicate that, as in the Ising problem, simple crossover operators that use single blocks or bands do not give improved results. Additional research will establish whether a variation of block-uniform crossover is a more effective crossover operator than those commonly used in other problem domains.

Acknowledgments

This research was supported by a grant from the Colorado Institute of Artificial Intelligence. Also, the work of J. R. was partially supported by the Air Force Office of Scientific Research and the Office of Naval Research Contract #F49620-90-C-0033.

References

[1] C. A. Anderson, "Effects of Parameter Settings on the Functioning of a Genetic Algorithm," *CU Denver Mathematics Department Research Report*, MC89f002 (1989) 2.1–2.15.

[2] D. H. Ackley, "An Empirical Study of Bit Vector Function Optimization," in *Genetic Algorithms and Simulated Annealing*, edited by L. Davis (San Mateo, CA, Morgan Kaufman, 1987).

[3] K. DeJong, "Genetic Algorithms: A Ten Year Perspective," *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, edited by J. J. Grefenstetter (Hillsdale, NJ, Lawrence Erlbaum, 1988).

[4] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Reading, MA, Addison-Wesley, 1988).

[5] J. H. Holland, *Adaptation in Natural and Artificial Systems* (Ann Arbor, University of Michigan Press, 1975).

[6] E. Ising, "Beitrag zur Theorie des Ferromagnetismus," *Zeitschrift für Physik*, **31** (1925) 253–258.

[7] B. M. McCoy and T. T. Wu, *The Two-Dimensional Ising Model* (Cambridge, MA, Harvard University Press, 1973).

[8] G. Syswerda, "Uniform Crossover in Genetic Algorithms," *Proceedings of the Third International Conference on Genetic Algorithms*, edited by J. D. Schaffer (San Mateo, CA, Morgan Kaufman, 1989).