

01 Mar 1993

## A Hybrid Genetic Algorithm for an NP-Complete Problem with an Expensive Evaluation Function

Richard Rankin

Ralph W. Wilkerson

Missouri University of Science and Technology, [ralphw@mst.edu](mailto:ralphw@mst.edu)

Geoff Harris

Jo Spring

Follow this and additional works at: [https://scholarsmine.mst.edu/comsci\\_facwork](https://scholarsmine.mst.edu/comsci_facwork)



Part of the [Computer Sciences Commons](#)

---

### Recommended Citation

R. Rankin et al., "A Hybrid Genetic Algorithm for an NP-Complete Problem with an Expensive Evaluation Function," *Proceedings of the ACM Symposium on Applied Computing*, pp. 251 - 256, Association for Computing Machinery, Mar 1993.

The definitive version is available at <https://doi.org/10.1145/162754.162889>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Computer Science Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact [scholarsmine@mst.edu](mailto:scholarsmine@mst.edu).

# A Hybrid Genetic Algorithm for an NP-Complete Problem With An Expensive Evaluation Function

Richard Rankin<sup>†</sup>, Ralph Wilkerson<sup>†</sup>, Geoff Harris\*, Jo Spring\*\*

<sup>†</sup>University of Missouri - Rolla

\*Griffith University

<sup>^</sup>University of Queensland

## Abstract

In this paper, a non-standard hybrid genetic algorithm is presented. The approach is non-standard in that it violates some of the common attributes associated with genetic algorithms in the literature. The algorithm presented uses local maxima to locate the global maximum value, uses haploid chromosomes with dominance mating instead of crossover, generates one offspring per set of parents, has no specific mutation operator, and is designed for rapid convergence. When applied to an NP-Complete problem, the results of this hybrid algorithm are shown to be very successful in reducing the complexity of the problem.

## Introduction

The Crozzle is a contest published monthly in the *Australian Womens Weekly* magazine. The contest is open to the residents of Australia and New Zealand. It is a crossword puzzle problem of enormous complexity [FO92]. Due to the challenges presented by this complexity several automated attempts at solving the puzzle have been made [FO92, HA292]. None of the automated attempts, however, have been able to consistently match the scores generated by the winning human contestants.

As discussed below, the Crozzle appears to be an NP-Complete problem, and as genetic algorithms have shown much promise in dealing with such problems, a genetic approach to the Crozzle is appropriate. This paper discusses the unusual problems presented by the Crozzle, and a hybrid genetic algorithm developed and implemented to decrease the size of the input lexicon, thus reducing the size of the search space to be traversed. (For an overview of genetic algorithms see, e.g., [GO89].)

## The Crozzle

The Crozzle is possibly the most complex of all the crossword puzzle variants. The game consists of generating the highest "scoring" theme crossword puzzle that will fit into a grid comprised of 15 columns and 10 rows of (initially) empty

squares. A lexicon (different each time the game is played) of approximately 120 words is used in the game, and all words used must be from the supplied word list. The object of the puzzle is to find the highest scoring puzzle that can be constructed according to the rules. Words from the word list are to be placed into the grid, resulting in a series of interlocked words which have the appearance of a standard crossword puzzle. Scoring is based upon the number of words used and the letters which appear in interlocked positions in the grid: each letter has a given value which can be added to the score only when that letter appears in two words at their intersection point. Words are not allowed to appear more than once in a solution. In the actual contest, a cash prize is awarded for the highest scoring entry submitted. For full details, and sample puzzles, see e.g. [FO92]

## The Hybrid Crozzle Genetic Algorithm

Hybrid genetic algorithms are genetic algorithms which do not directly solve the problem under consideration. They can be viewed as two-stage systems. The first stage, which contains the genetic algorithm, pre-processes the data from the problem domain. The output of the first stage is then used as input to the second stage. The second stage takes the pre-processed data and directly attacks the problem.

The approach used for the Crozzle is the Hybrid Crozzle Genetic Algorithm (HCGA). The HCGA has two parts just as has any hybrid genetic algorithm [GO89]. The first stage is the Crozzle Genetic Algorithm (CGA). The second stage is the Crozzle Solver (CS) itself. The output of the CGA is the data set which will be used by the second stage. This data set is intended to provide a smaller search space for the second stage without losing any of the data required to achieve the global maximum score.

This paper discusses only the CGA portion of the HCGA. The CS portion, its efficiency and success rates have been presented in detail previously [FO92, etc.].

## The Global Maximum Score

The global maximum score for any one particular Crozzle is currently an unknown quantity. To date, no report has ever been made of a score higher than that published as a contest winner by the *Australian Womens Weekly*. These published scores will be referred to as the 'human winning solution' (HWS) for any

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

ACM-SAC '93/2/93/IN, USA  
© 1993 ACM 0-89791-368-2/93/0002/0251...\$1.50

Crozzle. No automated attempt has ever exceeded this score. Therefore, although still the subject of ongoing research, for the present purposes, the global maximum score is considered to be the score published as the winning solution for a Crozzle. We use the term HWS broadly to represent the score achieved by the winning contestant as well as the word list used in that solution and the geometry constructed with that word list.

If the HWS is considered the global maximum solution, then, for every Crozzle puzzle, it is known what the highest score is, which subset of the word list was used, and what the geometry for the global maximum is. This observation is important as it allows the CGA stage of the HCGA to be evaluated independently of the second stage CS. If the CGA is able to output a word list smaller than the original 'thematic' list, without losing any of the words used in the HWS, then because of its properties it can be assumed that the CS could locate the correct highest scoring Crozzle solution.

### Difficulties with the Crozzle

The Crozzle is an NP-Complete problem at two levels. Given the grid is originally empty, it is an NP-Complete problem to generate all the possible geometries (word slots) for the grid [GA79,p. 258]. Once any given geometry is determined, the second NP-Complete problem is to fill those slots with words [GA79,p. 258]. That is, for each solution within the first problem (locating the positions of the black squares) a separate NP-Complete problem (determining solutions within a specific geometry of black and white squares) must also be solved.

The size of the search space for a typical Crozzle has been estimated to be approximately  $10^{24}$  nodes for a typical input lexicon [FO92]. This size, coupled with the fact that the fastest known implementation can process 2000 nodes per second, makes the Crozzle intractable for a complete traversal of the search space. This requires the use of heuristics which trim large portions of the search space [see, e.g., SM89].

Although the geometries and word lists of each Crozzle were analyzed as a precursor to this project no consistent features were detected which could be universally applied to the reduction of the search space. For example, since the letter 'Z' results in the highest interlocking letter score, it would seem the winning solution should have interlocked 'Z's whenever words containing 'Z' are present. This is not, however, always the case, at least as far as the HWS' are concerned.

Another observation that can be made is that the winning solutions frequently have highly interlocked blocks in them, resulting in high interlocking letter scores in a relatively small portion of the grid. (Here, we introduce the terminology "basic block" to indicate a set of words interlocked such that the removal of any one of the words would result in an incomplete (and thus not legitimate) block.) This is one portion of the problem that has indeed yielded successfully to automation: the generation of basic blocks has not only been implemented

[HA93] but is significantly more efficient and effective than humans competitors. However, not all human winning solutions contain basic blocks and, indeed, of those that do, very few contain the "highest scoring" basic block that exists within the lexicon.

### Genetic Algorithms and the Crozzle

When developing a genetic algorithm (GA) for the Crozzle, at some point, an objective evaluation function is required which rates the fitness of the chromosomes in the population. This evaluation function is generally required to be computationally inexpensive because it is used on each individual in the population during each generation of processing. There were no apparent 'universal truths' detectable from past geometries and word lists which could be used for pre-processing the word lists. The only known methods of evaluating a chromosome representing a series of words which are candidate solutions for the Crozzle are those mentioned in [FO92], and referred to previously as CS (Crozzle Solvers).

The CS programs are not computationally inexpensive functions, but there are no known alternatives. Using them as the evaluation functions for the genetic algorithm necessitated extensive changes in the conceptual approach to the Crozzle Genetic Algorithm (CGA).

As the search techniques utilized by the CS result in effectively infinite runtimes without heuristics, and are not guaranteed of success with the heuristics (various branch and bound methods), the goal of search space reduction is paramount. The genetic algorithm presented here addresses the problem of search space reduction, and not the problem of directly generating the global maximum solution to a particular Crozzle.

Most genetic algorithms require large populations and a large number of generations. With an expensive evaluation function, however, an indirect result is that it is not possible to have either large populations or many generations. For example, assume that a chromosome in the CGA represented only a 10 word list. If the population consisted of 5000 individuals, then a single generation, based on sample CS timings, would require approximately 80 hours of processing. Consequently, a single run of 500 generations would require over 4 years to complete.

The necessity of using an expensive evaluation function precludes some of the attributes of a 'normal' genetic algorithm. To obtain a result with the Crozzle problem, the GA must converge with few generations and with a small population. The primary effect to the CGA is that rapid convergence must be encouraged instead of discouraged, as is common in many genetic algorithms applications [BA85, MA84].

It should be specifically noted that the CS is used both as the evaluation function for the genetic algorithm of stage 1, and, by itself, as stage 2. It appears from a search of the literature that this arrangement is unique.

### The Goal of the Crozzle Genetic Algorithm (CGA)

The goal of the CGA is to directly reduce the search space of the Crozzle problem without losing the global maximum score. The technique used is that of reducing the size of the original thematic word list without trimming any of the words required in the HSW.

Specifically, the CGA attempts to eliminate words with lengths greater than 5 characters from the original word list. From empirical tests on the CS, it has been determined that, for every 5 words of length greater than 5 characters eliminated from the input word list, one order of magnitude has been eliminated from the search space. For example, eliminating 20 words of this length, would reduce the overall search space by 4 orders of magnitude.

Words with lengths shorter than 6 characters are used in the processing of the CGA. They are not generated as output, however.

From interviews with some Crozzle contestants, it appears that short words (lengths of 3,4 and 5) are used as links to piece together disjoint word groups while working towards a solution. This, plus the fact that 3 letter words can fit into any convenient space at any point in the process to increase the total score, suggested that all the short words should be available to the stage 2 (CS) process.

Longer words are more troublesome to place, but are also more strategic, in the sense that their placement in the grid can be crucial in fitting an acceptable solution into the 15 by 10 square grid. The longer words obviously provide more possible interlock locations and, thus, require more processing per word than a shorter word. Therefore, eliminating longer words also reduces more of the search space than does the elimination of a shorter word.

As a result of these observations and interviews, it was decided to ignore any attempts by the CGA to trim 3,4, and 5 letter words from the word list. Although these shorter words are available to the CGA while evaluating the chromosomes, they are not subject to elimination as are the longer words in the list, and are always available to stage 2 of the HCGA.

### Nonstandard Operators in the CGA

Since the CGA is geared towards rapid convergence, the operators involved were necessarily altered accordingly. The two operators most affected are those of reproduction and crossover. Mutation, a common operator in most genetic algorithms, was not used for the tests reported here.

The normal behavior of reproduction is to select those chromosomes from the population which will survive to the next generation for mating. The common methods used for this

process are stochastic sampling with replacement, and stochastic sampling with partial replacement, although other methods exist [BA87]. These methods generally try to assure that each chromosome gets its 'fair representation' in the next generation. In the CGA, however, it is imperative, due to the small number of generations which can be processed, that convergence to a maximum be rapid.

To assist rapid convergence, the CGA reproduction operator assures that every high quality individual survives into the next generation. The CGA determines the highest fitness score in the entire population. All chromosomes with that score are automatically carried over to the next generation. The balance of the population for the succeeding generation is then selected using stochastic sampling with replacement. This assures that the best chromosomes are at least represented, and probably over-represented in the surviving population, assisting in rapid convergence.

Crossover is normally done by selecting two chromosomes and a crossover site. Two new offspring are created, each having the genes of one chromosome from one parent up to the crossover site combined with the genes of the second parent after the crossover site. For example, given the two chromosomes X and Y, below, represented in binary form, and the crossover site 4, the resulting offspring X' and Y' are shown.

```
X = 111111
Y = 000000
X' = 111100
Y' = 000011
```

Crossover in the CGA uses haploid (single strand) chromosomes, but borrows from the diploid (dual strand) chromosome concept of dominance between two genes. In the CGA, two haploid parents are selected. However, when they are actually mated, only a single offspring is produced, having the most dominant gene in every position from the two parents. For example, if two chromosomes X'' and Y'' existed with 5 genes, and the dominance values for each gene were as shown below, the single offspring Z would be produced with the dominance values shown.

```
X'' = 10 40 45 76 21
Y'' = 32 20 44 80 20
Z = 32 40 45 80 21
```

Since in position 1, the Y'' gene was dominant (with the higher value of 32), it would be carried on to Z in position 1. In position 2, the X'' gene is dominant (with the higher value of 40), so it would be carried on to Z, and so on.

The dominance factor used is called Average Potential Word Score (APWS) which is the sum of the letter values in the word represented, divided by the word's length. This function was selected as a large portion of the words in the Crozzle word lists with high APWS are generally used in the winning solution.

The changes in the crossover and reproduction operators are the primary distinguishing features of the CGA, and proved to be crucial to the success of the project. When the 'normal' implementations of these operators were used, results, although better than a purely random approach, were not sufficiently refined to be considered successful. The possibility, if not the probability, exists that this lack of success could be directly tied to the small population size and the small number of generations. These parameters, however, as discussed above, were relatively fixed. Therefore, the operators themselves needed to be changed to work within the fixed parameters of population size and number of generations.

### Implementation

The implementation issues involved in the CGA revolve strongly around the necessity of converging rapidly to solutions. As discussed, the operators were altered from their normal behavior. A severe limit was required on both population size and the number of generations which could be processed. Externally imposed constraints were that the CGA run on machines readily available (10 networked NeXT computers), and finish in less than 24 hours.

Due to the processing time required for word lists of any substantial length, the CGA attempts to find a series of local maxima involving short word lists. When a local maximum is found, the word list producing that local maximum is recorded. At the end of the entire process, all the word lists produced by local maxima are combined. This combined list is the output of the CGA and would be the input to stage 2 (CS).

The operation is as follows:

- 1) Ten word lists are randomly generated from the thematic word list. Each of the ten word lists contains approximately 25 words (duplication is allowed among lists).
- 2) The ten word lists are processed by ten instances of the CGA, each using 8 genes (words) per chromosome, a population of 150 chromosomes, and 15 generations. Each of the ten instances (called GA8) produces a word list which represents the highest scoring chromosome in the population over the 15 generations. These word lists are combined without duplication into the GA8 master list.
- 3) Ten new word lists are generated as in step 1.
- 4) Ten new instances of the CGA process the ten new word lists. Each of these instances (called GA10) uses chromosomes with 10 genes, 100 chromosomes, and 15 generations. The highest scoring word lists from each instance are combined, as in step 2, to create the GA10 master list.
- 5) Stage 2 (CS) can process one of three alternate lists: the GA8 master list, the GA10 master list, or the list derived from

combining both master lists, called the Union List. Any of these is amended through the addition of the shorter words of length 3,4 and 5.

As noted previously, step 5 can be effectively ignored when processing Crozzles with a published HWS. If the word subset used in the HWS is contained in any of the three alternate lists, it is assumed that the CS would find the correct solution.

The number of instances (10) of each version of the CGA was based on the number of machines generally available. The number of chromosomes selected was based upon empirical testing and timed runs. As noted, a time limit of 24 hours was arbitrarily imposed, and, from the timing runs done, these combinations of chromosome lengths seemed to meet the proper overall time constraints.

The original plan was to test the results only of the Union List.

In practice, however, the GA8 master lists and the GA10 master lists produced very satisfactory results. The union list generated results better than anticipated.

### Results

Ten Crozzles were selected for testing. The results of these tests appears in Table 1. The results of GA8 and GA10 are compared to random selection. The 'expected' column shows how many of the long words in the HWS would be expected to appear in a randomly selected word list of the size generated by the GA from the initial theme word list. The 'Actual' columns show both the number of long words in the HWS which actually appeared in the GA list, and how many long words were in the HWS. When the two numbers are not the same, that master list was considered to have 'failed'.

Table 1: Performance of GA's compared to random selection of long words.

	Randomly Expected, GA8	Actual GA8 /HWS	Randomly Expected, GA10	Actual GA10 /HWS
Jan92	5.77	10/11	6.11	10/11
Feb92	4.81	9/10	5.8	10/10
Jul91	1.52	3/3	1.72	3/3
Apr91	2.36	4/5	2.55	5/5
Dec90	3.60	6/6	3.16	4/6
Feb90	3.80	8/8	3.60	5/8
Aug89	5.00	8/8	4.38	8/8
Oct89	4.26	8/8	3.91	8/8
Feb88	3.11	6/6	3.64	6/6
Oct90	1.75	4/4	2.46	4/4
AVG.	3.60	6.70/6.90	3.73	6.30/6.90

With the Union List, the global maximum was retained in 9 of

the 10 Crozzles. Only January 1992 did not succeed according to this criteria, missing a single word. This means that the original idea of combining the two master lists into a Union List resulted in a 90% success rate. The GA8 master lists, by themselves, retained the global maximum in 7 of 10 Crozzles. The GA10 master lists did the same. Therefore, even separately, the master lists each retained the global maximum 70% of the time.

The importance of these results, however, lies in the number of words eliminated from the word list while retaining those words in the global maximum. From empirical tests [RA92], it has been determined that, for each 5 words eliminated, one order of magnitude has been eliminated from the search space. By looking at Table 2, the number of words eliminated can be seen. On the July 1991 Crozzle, for example, GA8 retained the global maximum while trimming approximately 30 words from the word list. This results in 6 orders of magnitude smaller search space. GA10, for October 1989, retained the global maximum while trimming nearly 50 words, or 10 orders of magnitude, from the search space.

Table 2: Trimming results on words with lengths greater than 5.

	Original Long Words	GA8 Total	GA8 Words Trimmed	GA10 Total	GA10 Words Trimmed
Jan92	99	52	47	55	44
Feb92	81	39	42	47	34
Jul91	61	31	30	35	26
Apr91	55	26	29	28	27
Dec90	55	33	22	29	26
Feb90	80	38	42	36	44
Aug89	62	39	23	34	28
Oct89	94	50	44	46	48
Feb88	56	29	27	34	22
Oct90	57	25	32	35	22
AVG.	70.43	36.86	33.8	37.71	32.1

The 70% success rates of the GA8 and GA10 master lists may prove to be more valuable than the 90% success rate of the Union List in practice. Each of these lists is obviously shorter or equal to the Union List in length, and therefore, results in a smaller search space. If the 70% retention rate of the global maximum is acceptable to the user, then a more vigorous search may be done over the smaller search spaces, providing a greater chance of locating the maximum within that space. If the maximum is located, the test results indicate a 70% chance that the maximum of that space is, in fact, the global maximum for the entire word list.

As shown, the average was approximately 35 words trimmed from the word list, resulting in an estimated 7 orders of magnitude smaller search space for stage 2. These huge gains in efficiency allow the stage 2 portion to do tighter traversals of the

search space than before without losing the global maximum score in the process.

### Conclusion and Future Research

A genetic algorithm for an NP-Complete problem known as the Crozzle has been presented which makes use of rapid convergence along with unusual variants of the reproduction and crossover operators. The lack of an inexpensive evaluation function for the genetic algorithm required that small populations and few generations be processed. For the same reason, a series of local maxima were collected and combined to attempt the generation of the global maximum score from the search space.

The hybrid CGA retained the global maximum score 90% of the time. More importantly, the smaller GA8 and GA10 master lists retained the global maximum score 70% of the time while eliminating approximately 7 orders of magnitude from the overall search space. This reduction, combined with other heuristics in the stage 2 portion of the hybrid CGA allow for much fuller traversals of the enormous search space.

At least two avenues of future research directly involving the Crozzle are suggested by the success of this project. First, the number of instances of both GA8 and GA10 were determined by the hardware resources available. It is unknown how changing the number of instances would affect the results and what the optimal number of such instances might be. Second, through empirical analysis, it can be determined, without even running the CGA, that certain Crozzle word lists are not likely to be successfully trimmed. These 'pre-destined failures' relate directly to the number of 3,4, and 5 letter words in the original input word list, even though such words are not subject to pruning by the CGA. Various attempts to solve this particular problem have failed to date. At the present time, the explanation for this phenomenon has not been determined.

### References and Bibliography

- [BA87] Baker, James Edward, "Reducing Bias and Inefficiency in the Selection Algorithm", Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms, John J. Greffentette, ed., 1987 Cambridge, MA, pp. 14 -21.
- [BA85] Baker, James Edward, "Adaptive Selection Methods for Genetic Algorithms", Proceedings of the First International Conference on Genetic Algorithms and Their Applications, John J. Greffentette, ed., Hillsdale, NJ, pp. 101-111.
- [BE87] Berghel, H., "Crossword Compilation with Horn Clauses", Computer Journal, Vol. 30, No. 2, pp. 183- 188.
- [BE90] Berghel, H and R. Rankin, "A Proposed Standard for Measuring Crossword Compilation Efficiency", Computer Journal, Vol. 32, No.3, pp.276-280.

- [BE89] Berghel, H. and C. Yi, "Crossword-Compiler Compilation", Computer Journal, Vol. 32, No. 3, pp. 276-280.
- [FO92] Forster, J.J.H., G.H. Harris, P.D. Smith, "The Crozzle - A Problem for Automation", Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing, ACM Press, NY, 1992.
- [GA79] Garey, Michael R., and David S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Co., NY 1979.
- [GO89] Goldberg, David E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1992.
- [HA192] Harris, G., L.J. Spring, J.J.H. Forster, "An Efficient Algorithm for Crossword Puzzle Solutions", Computer Journal, Vol. 35, pp. A181-A183.
- [HA292] Harris, G., and J.J.H. Forster, "The Crozzle: A New Lexicographical NP-Complete Problem", in preparation.
- [HA392] Harris, G. and J.J. H. Forster, "On the Number of Solutions,  $S(n,k)$ , to a Crossword Puzzle", Computer Journal, Vol. 35, pp. A177-A180.
- [HA190] Harris, Geoff, "Generation of Solution Sets for the Unconstrained Crossword Puzzle", 1990 Symposium on Applied Computing, IEEE Computer Society Press, Los Alamitos, 1990
- [HA290] Harris, G.H. and J.J.H. Forster, "On the Bayesian Estimation and Computation of the Number of Solutions to Crossword Puzzles", 1990 Symposium on Applied Computing, IEEE Computer Society Press, Los Alamitos, 1990
- [HA492] Harris, G., D. Roach, H. Berghel, P.D. Smith, "Dynamic Crossword Slot Table Implementation", Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing, ACM Press, NY, 1992.
- [HA93] Harris, G., and R. Rankin, "A Branch and Bound Approach to the Crozzle", forthcoming, 1993 ACM/SIGAPP Symposium on Applied Computing.
- [MA84] Mauldin, Michael, "Maintaining Diversity in Genetic Search", Proceedings of the National Conference on Artificial Intelligence, American Association of Artificial Intelligence, Austin, TX, 1984, pp. 247-250.
- [RA92] Rankin, Richard, G. Harris, L. J. Spring, "A Non-Standard Genetic Algorithm", in preparation for The Australian Computer Journal.
- [SM89] Smith, Jeffrey D., Design and Analysis of Algorithms, PWS-Kent Publishing Company, Boston, 1989
- [SP93] Spring, L. J., "Testing Automated Solutions to the Crozzle", forthcoming, 1993 ACM/SIGAPP Symposium on Applied Computing.
- [SP92] Spring, J., Berghel, H., Harris, G.H., and Forster, J.J.H., "A Proposed Benchmark for Testing Implementations of Crossword Puzzle Algorithms", Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing, ACM Press, NY, 1992.
- [WI89] Wilson, J.M., "Crossword Compilation Using Integer Programming", Computer Journal, Vol. 32, No. 3, pp. 273-275.