Quantum-inspired genetic algorithms

Ajit Narayanan and **Mark Moore**
Department of Computer Science
University of Exeter
Exeter, United Kingdom, EX4 4PT
ajit@dcs.exeter.ac.uk

Abstract— **A novel evolutionary computing method — quantum inspired genetic algorithms — is introduced, where concepts and principles of quantum mechanics are used to inform and inspire more efficient evolutionary computing methods. The basic terminology of quantum mechanics is introduced before a comparison is made between a classical genetic algorithm and a quantum inspired method for the travelling salesperson problem. It is informally shown that the quantum inspired genetic algorithm performs better than the classical counterpart for a small domain. The paper concludes with some speculative comments concerning the relationship between quantum inspired genetic algorithms and various complexity classes.**

I. Introduction

Quantum-inspired computing [7] is characterised by:

1. the use of a 'quantum-inspired' computational method which is inspired by certain principles of quantum mechanics, such as standing waves, interference and coherence; and
2. the use of a classical algorithm for checking that the 'candidate' solutions generated by the quantum-inspired computational method are in fact correct.

The nucleus of an atom contains particles called protons (positively charged) and neutrons (electrically neutral). The nucleus is surrounded by electrons (negatively charged). The orbits of electrons, according to quantum mechanics, is not planar, in the way that the Earth's orbit around the sun is planar. Instead, the orbit of electrons is best described as a wave (Figure 1). There are several different types of orbit, depending on the angular momentum and energy level. An electron 'jumps' orbit from an orbit with low energy level to an orbit with high energy level by absorbing energy (e.g. a photon). The term 'quantum' signifies that there can be no in-between states or orbits. The jumps are discrete, and there are 6 energy levels (for the hydrogen atom) which can be roughly characterised as the number of cycles an electron has to go through while orbiting the nucleus at that energy level (i.e. an electron with energy level 1 orbits the nucleus with one cycle, energy level 2 with two cycles, and so on). An electron can jump, again discretely, back to a lower energy level by releasing energy (a photon). Hence, an electron around a nucleus jumps states in discrete quanta by absorbing energy or releasing it.

Generally speaking, a quantum particle's *location* can best be described by a *quantum* state vector $|\Psi>$, a weighted sum which in the case of two possible locations equals w $|A> + x$ $|B>$, where w and x are complex number

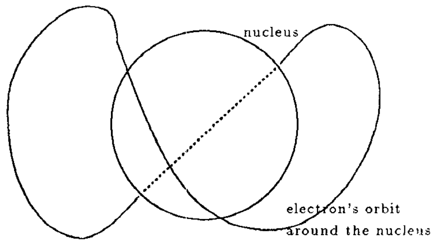

Fig. 1. An electron's orbit is not planar but is best described as a wave. In the figure, the electron's orbit is such that it goes through one complete 'cycle' around the nucleus, starting at some point in front of the nucleus, going around the back (dotted line) before completing its cycle.

weighting factors of the particle being in locations A and B, respectively, and where w $|A>$ and x $|B>$ are themselves state vectors [8]. For n possible states, there will be n different complex number weighting factors, each describing the weighted probability of the particle being at that location.[1] $|\Psi>$ represents a *linear superposition* of the particle given individual quantum state vectors: If there are n locations as given by n state vectors, the particle is said to be at all n locations at the same time ('many universes'). However, in the act of observing a quantum state (or wave function), it collapses to a single state.[2] Collapse doesn't depend on observation, however. Collapse can also occur through 'interactive decoherence', where nonseparable correlations between state vector locations of an object and its environment partition possible states into preferred states, thus effectively reducing state vectors (collapsing wave functions). Richard Feynman [2; 3] showed how a quantum system could be used to perform computations and could act as a simulator for quantum processes, where such simulations are impossible to achieve efficiently on a conventional automaton.

II. Quantum factoring

The 'many universes' interpretation is used by Shor [10] in his quantum computing method for extracting prime factors of very large integers, where a memory register is placed in a superposition of all possible integers it can con-

[1] The weighting factors are used in ratios to calculate the probability of the particle being at A compared with that of it being at B. They are not the probabilities themselves.

[2] Although the introduction here has concentrated on electrons, the principles of quantum mechanics apply to molecules, atoms, electrons, protons, neutrons, photons, quarks and all other subatomic particles.

u_0:	1	7	28	31	7	28	31	7
u_1:	7	28	31	7	28	31	7	28
u_2:	16	4	25	10	1	7	28	31
u_3:	13	13	13	13	13	13	13	13
u_4:	25	10	1	7	28	31	7	28
u_5:	10	1	7	28	31	7	28	31
u_6:	4	25	10	1	7	28	31	7
u_7:	28	31	7	28	31	7	28	31
u_8:	31	7	28	31	7	28	31	7
u_9:	19	19	19	19	19	19	19	19
u_{10}:	1	7	28	31	7	28	31	7
u_{11}:	7	28	31	7	28	31	7	28
u_{12}:	16	4	25	10	1	7	28	31
u_{13}:	13	13	13	13	13	13	13	13
u_{14}:	25	10	1	7	28	31	7	28
u_{15}:	10	1	7	28	31	7	28	31
u_{16}:	4	25	10	1	7	28	31	7

Table 1. 17 universes for identifying the prime factors of 33, given $x = 7$.

tain, followed by a different calculation being performed in each 'universe'. The computation halts when the different universes 'interfere' with each other because repeating sequences of integers (standing waves) are found in each universe and across universes (a form of interactive decoherence which does not rely on observation). Although there is no guarantee that the results are correct, a subsequent check can be made at this point to identify whether the numbers returned are indeed prime factors of the given large integer.

Cryptography is used to encode the secret transactions of banks, governments, and the military. Public key systems widely employ key production methods based on RSA. RSA is named after its inventors Ronald Rivest, Adi Shamir, and Leonard Adleman [9]. Briefly, two distinct large prime numbers are chosen randomly, p and q. Their product $r = p \times q$ is calculated. A large integer e is chosen that is relatively prime to $(p - 1) \times (q - 1)$; e is the encryption key. The decryption key d is the unique multiplicative inverse of e mod $(p - 1) \times (q - 1)$, i.e. $d \times e = 1$, mod $(p - 1) \times (q - 1)$. r (the original product) and e are published in the public domain, but d is kept private. For a secret communication from X to Y: X encrypts the message using Y's public key; only Y's private key is able to decrypt the message. For a signed communication from X to Y: X encrypts the message using X's private key; Y will use X's public key to decrypt the message. For a secret and signed message from X to Y, X uses X's private key then Y's public key, and Y must use Y's private key then X's public key to reconstruct the message.

The success of RSA depends on the seeming intractability of the problem of finding the prime factors of very large integers. When the system was invented back in 1977 its inventors challenged anyone to factorise a particular 129-digit number, now known as RSA-129. This challenge stood until 1994, when 1600 computers linked via the Internet were able to determine the factors in just over eight months. Despite this breakthrough, cryptographers reassured themselves with the thought of adding more digits to their codes. The problem of factoring numbers becomes exponentially harder the larger they are. However, Shor [10] outlined a quantum algorithm that would factorise RSA-129 in a few seconds, if a quantum computer could be built that ran as fast as a modern PC. To factor a number n (the product of two primes), specify an arbitrary number of parallel universes p (0, 1, 2, ...), and randomly select an integer x between 0 and n. Then, in every universe, raise x to the power of the number of the universe. Divide by n and store the remainder in the universe. For the next number in the sequence for a universe, x is raised to the power of the number last stored, divided by n, and the remainder stored. This continues in each universe to form a repeating sequence. For the sake of exposition, let n (the number to be factored into prime factors) be 33. Let $x = 7$ $(0 < x < n)$ and $p = 17$. Table 1 provides a detailed description of what happens in each universe u_0 to u_{16}.

Consider u_2 of Table 1: 7^2 mod $33 = 16$, 7^{16} mod $33 = 4$, 7^4 mod $33 = 25$, 7^{25} mod $33 = 10$, 7^{10} mod $33 = 1$, etc. Now consider u_5: 7^5 mod $33 = 10$, 7^{10} mod $33 = 1$, 7^1 mod $33 = 7$, 7^7 mod $33 = 28$, 7^{28} mod $33 = 31$, etc. The frequency of repeat across universes — *vertical frequency*, vf — is 10, and this can be seen by examining u_0 and u_{10}, u_1 and u_{11}, u_2 and u_{12}, and so on. That is, in each of these universes the repeating frequency starts at the same point and repeats the same numbers. While other universes share the repeating pattern, they do not share the common starting point. This is the quantum computation equivalent of standing waves in each universe: a repetition of numerical values. The calculation $x^{vf/2} - 1$ is now made, where x is the arbitrarily chosen number between 0 and n and vf is the frequency of common repeating patterns across universes. This gives $7^{10/2} - 1 (\text{mod } 33)$, which is 9. ($7^5 - 1 = 16806$; 16806 mod 33 = 9.) Now, finding the greatest common divisor of 9 and 33 gives one of the factors, i.e. 3.

Shor's method is not guaranteed to always work, and if the derived number turns out not to be a prime factor of n, the procedure is repeated using a different x. It is claimed that on average only a few trials will be required to factorise n, even when n is very large. So although there are no known fast classical algorithms for factorising large numbers into primes, Shor's method uses known fast algorithms for taking a candidate prime factor of n and determining whether it is in fact a prime factor. Shor's method has been extended to deal with factoring large even numbers and squares [6], sorting [7] and neural networks [5].

III. The travelling salesperson problem

Given n points and a distance D, is there a tour that visits each of the n cities exactly once, returning to its starting point, and has total length $\leq D$? A description of a network D is provided in Table 2. The i and j axes correspond to the nodes in the network. The numbers within the table refer to the distances between pairs of nodes. For

I	4	27	16	12	18	29	22	12	∞
H	16	21	29	10	9	23	13	∞	12
G	8	19	24	15	5	16	∞	13	22
F	12	13	5	20	32	∞	16	23	29
E	13	6	10	30	∞	32	5	9	18
D	19	2	25	∞	30	20	15	10	12
C	7	11	∞	25	10	5	24	29	16
B	13	∞	11	2	6	13	19	21	27
A	∞	13	7	19	13	12	8	16	4
	A	B	C	D	E	F	G	H	I

Table 2. Table of distances between cities. The table is symmetric, but need not necessarily be so (e.g. one-way routes which involve greater or less distance than their return).

example, the distance between nodes A and G is 8. Every combination could be tried in this examples, since there are only 9! = 362880, but as the number of nodes is increased, brute-force testing eventually becomes impractical. Classically, a route through the network can be represented as a chromosome consisting of ten genes, each of which is a letter between A and I. The first gene in the chromosome is identical to the tenth gene in every case, as each route through the network is a tour and must return to its starting node. For the sake of exposition, let us consider the following chromosome:

B	C	E	I	G	A	H	D	F	B

This represents the following tour: $B \to C \to E \to I \to G \to A \to H \to D \to F \to B$. This route through the network has a total distance of: 11 + 10 + 18 + 22 + 8 + 16 + 10 + 20 + 13 = 128 (Table 2). For mutation, two of the chromosome's first nine genetic values (alleles) are chosen randomly and exchange positions, ensuring that the first and tenth alleles remain identical in the result (to preserve looping back to the starting point). For example, the above chromosome can mutate into:

G	C	E	I	B	A	H	D	F	G

where the B in first position is exchanged with the G in fifth position and the final B is changed to a G to preserve looping. The probability of any chromosome within the population mutating is chosen arbitrarily, and is 0.667.

Two chromosomes can undergo crossover. This crossover is a quantum inspired form of fixed point: take the 1st element of chromosome one, 2nd element of chromosome two, 3rd element of chromosome one, 4th element of chromosome two, etc. If an element is already present in the chromosome, choose the next alphabetical allele not already contained in the chromosome. For example, crossing

A	C	D	I	B	G	H	F	E	A

with

G	C	E	H	B	A	I	D	F	G

produces the following two chromosomes:

A	C	D	H	B	E	I	F	G	A

and

G	C	E	I	B	H	A	F	D	G

where the first chromosome consists of the A in first position in the upper chromosome (uc), the C in second position of the lower chromosome (lc), the D in third position of uc, the H in the fourth position of lc and the B in fifth position of uc. At this point, the A in the sixth position of lc cannot be included since A already occurs in the first resulting chromosome, so the next alphabetical allele given the alleles already in the resulting chromosome is chosen. Since the first resulting chromosome already contains A, B, C and D, E is chosen instead of A. The probability of a chromosome being selected for crossover is chosen arbitrarily, and is 0.667. No duplicate chromosomes are allowed.

The population size has been chosen to equal the number of nodes in the network, i.e. 9. If after mutation and crossover there are more than 9 chromosomes, only 9 may continue into the next generation. A new generation is created by keeping the chromosome with the shortest network tour, and the remaining 8 survivors are selected at random from the remaining candidates according to the rank-space method, a fitness method considering the combined ranking of distance rank and diversity rank [11]. To obtain a distance ranking, the remaining candidates are sorted into tour distance order, then the candidate with the shortest distance is given distance ranking of 1, the next best candidate is given a distance ranking of 2, etc. The diversity rank of a chromosome is determined by calculating the sum of the inverse squared differences between that chromosome and the other, already selected, chromosomes. This is done by assigning each allele an integer, e.g. A =1, B = 2, C = 3, D = 4, etc. The diversity rank is calculated as follows: $\sum_i \frac{1}{d_i^2}$. The candidate with the smallest diversity value is given a diversity ranking of 1, the next best candidate is given a diversity ranking of 2, etc. Next the rank scores for distance and diversity are added together to give a combined score. The candidate with the lowest combined score is given a combined rank of 1, the next best candidate is given a combined rank of 2, etc.

Now the rank fitness method is applied. The fitness of the best combined rank candidate is some fixed constant, in this case $p = 0.667$. This p value is a probability. If the best candidate, the one ranked number 1, is not selected, then the next best candidate, the one ranked number 2, is selected with fitness p. This process of selection proceeds until a candidate is selected for the next generation or there is only one remaining, in which case that last-ranked candidate is selected.

For example, given the network in Table 2 and supposing that the chromosome

E	H	G	I	A	F	C	B	D	E

has been selected, this chromosome is represented as the integer list 5—8—7—9—1—6—3—2—4—5 and eight more remain to be chosen.

Table 4 describes the selection procedure in more detail. The diversity rank describes how different a chromosome is from the winning chromosome, and chromosomes with high diversity are given a high ranking (to signify the principle that it is as good to be different as it is to be fit — a useful way of keeping local maxima in the search space for future use in identifying potential global maxima). The distance rank is based on the actual distance of the route represented by the chromosome, with shorter routes having higher rankings. The rank sum of these two ranks leads to an overall combined rank, with a probability that the first will chromosome will have 0.667 chance of surviving, the second 0.222, etc. If 8 chromosomes do not make it through this selection process to join the winning chromosome, a random selection is made to make up the numbers. It is likely that 1—3—2—5—6—4—7—8—9—1 is selected for the next generation with a probability of 0.667. Summarising this 'classical genetic algorithm': (a) The search starts with nine chromosomes that are randomly chosen; (b) No chromosomes are allowed to appear more than once in each generation; (c) A maximum of nine chromosomes are allowed to proceed into the next generation; (d) a survivor chromosome mutates with the user defined probability of 0.667; (e) two genes are selected at random in each of these survivors, and exchange positions within the chromosome; (f) Chromosomes are selected for crossover with the user defined probability 0.667; (g) The chromosome with the smallest tour distance survives to the next generation; (h) The remaining survivors are selected from the remaining candidates, according to the rank-space method and further random selection if required; (i) Evolution ceases once an 'optimum' distance is not bettered for a number of generations equal to half of the number generations it took to find the 'optimal' chromosome (Table 4).

IV. A quantum-inspired genetic algorithm

The number of universes required for the algorithm, u, is equal to the number of nodes in the network, in this case 9. Each universe contains its own population of chromosomes. The populations in each universe obey identical rules to the classical case, and evolve in parallel (i.e. the generation numbers are always equal). However, there is one difference — the universes can interfere with one another in each generation. This interference occurs as a type of crossover involving 9 chromosomes, and takes place just after classical crossover within each universe. In each universe there is a list of chromosomes constituting the population, and every chromosome in each universe has a list index position i in its universe. The probability of chromosomes in positions i in all universes mating is chosen arbitrarily to be 0.667. Only genes in the separate universes with the same list index position are given the possibility of crossover. The resulting nine chromosomes are placed so that one is assigned to each universe. As with the classical algorithm, no duplicates are permitted within the same universe.

Interference crossover occurs as follows: take the 1st element of chromosome one, 2nd element of chromosome two, 3rd element of chromosome three, 4th element of chromosome four, etc. If an element is already present in the chromosome, choose the next alphabetical element not already contained in the chromosome. Table 5 describes the universes prior to interference crossover and Table 3 the

Universe	Generated chromosomes									
u0:	G	C	H	I	A	B	D	E	F	G
u1:	E	A	C	H	D	I	F	G	H	E
u2:	I	D	A	H	E	B	G	C	F	I
u3:	A	F	G	H	I	B	E	C	D	A
u4:	D	A	I	E	B	F	H	C	G	D
u5:	C	F	G	H	I	A	E	B	D	C
u6:	B	H	D	E	F	C	G	I	A	B
u7:	I	C	B	D	E	F	G	H	A	I
u8:	A	B	C	G	E	D	F	H	I	A

Table 3. Table describing the results of interference crossover. The diagonal elements of Table 5 are put into the universes as given by the start node of each chromosome. For instance, the diagonal chromosome (E A C H D I F G H E) in u1 is put in the second universe since its starting node E is the first node of u1 in Table 5.

universes after interference crossover, for one chromosome from each universe. For Table 5, G (first position in first chromosome in u0) is combined with C (second position in second chromosomes in u1) and with H (third position in third chromosome in u3), etc. If a gene value already appears, then the next alphabetic allele is chosen, as before. Note the 'wrap-around': A (first position of first chromosome in u8) is followed by A (second position of second chromosome in u0) and is changed to B as an alphabetic substitution for duplication. The shadings indicate this wrap-around. Note also that the final allele (final column, not shaded) must be the same as the first to preserve returning to the start point.

V. Discussion

An experiment was performed to contrast the efficiency of both algorithms. The algorithms were tested on the network given Table 2. Computation ceased when the 'optimal' tour length of 67 was found. Both algorithms were executed 50 times and yielded the following results:

Algorithm type	Average number of generations to locate optimal route
Classical	24.16
Quantum-inspired	16.10

The quantum-inspired genetic algorithm in this case outperforms the classical version by a significant margin, despite the fact the network is relatively small. This increased performance may be attributed to interference crossover, which provides a larger number of chromosomes to choose from when selecting survivors for the next generation. Further simulations will be carried out with many different networks of a greater size. It will also be interesting to monitor the effects of changing the population size, mutation rate, crossover rate, and p (the rank fitness probability) for both algorithms. Further work is also required to classify interference crossover techniques and methods and relate them to existing, classical crossover techniques and methods.

It has been estimated that every two years for the past 50 years computers have become twice as fast while their components have become twice as small [4]. If progress continues at this rate future computer circuits will be based on nanotechnology[3] and the behaviour of such circuits will have to be given in quantum mechanical terms rather than in terms of classical physics, since on the atomic scale matter obeys the laws of quantum mechanics [1]. As technology heads towards the nano and quantum level, there will be an increasing need for computational paradigms which can benefit from their proximity to the quantum hardware. Such computational paradigms will require less translation to quantum 'machine language' than their classical counterparts and will therefore have efficiency benefits. Quantum-inspired genetic algorithms as described here are still one step removed from true quantum genetic algorithms (where such notions as chromosomes will be given a quantum or molecular rather than symbolic description) but perform a useful function in helping current researchers identify potential trends in evolutionary computing. It is currently not clear how true quantum computation algorithms will be related to quantum hardware (e.g. quantum logic gates [1]). It is possible that quantum-inspired computing will provide the benefits of portability across different quantum hardware platforms should 'quantum compilers' be feasible.

The quantum-inspired techniques described here enrich our knowledge of how parallel computation techniques can be used for implementing genetic algorithms. The 'parallel universes' point of view lends itself naturally to parallel computation, with the main difference being the use of interference across computations (worlds) and the possible use of interactive decoherence for allowing the computations in different worlds to stop by themselves when a potential acceptable solution is found. Further work is required to relate explicitly parallel computation with quantum-inspired computation.

The aim of this paper is not to present a formal exposition of quantum-inspired genetic algorithms but to demonstrate the feasibility of such methods and the novelty of the paradigm. As Shor points out [10], in addition to the complexity class NP (intuitively, the class of exponential search problems) there is the class PSPACE, which are those problems which can be solved with an amount of memory polynomial with respect to the input size, and the class BPP, which are problems which can be solved with high probability in polynomial time (given access to a random number generator). It is possible that quantum-inspired genetic algorithms fall somewhere between PSPACE, NP and BPP. The relationship between NP and BPP is not currently known, and quantum-inspired genetic algorithms may offer insight into the precise relationship between NP and BPP.

References

[1] D. Deutsch. Quantum computational networks. *Proceedings of the Royal Society of London*, A425:73–90, 1989.

[2] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7):467–488, 1982.

[3] R. Feynman. Quantum mechanical computers. *Foundations of Physics*, 16:507–531, 1986.

[4] S. Lloyd. Quantum-mechanical computers. Scientific American, October 1995.

[5] T. S. I. Menneer and A. Narayanan. Quantum-inspired neural networks (QUINNS). Technical report, Department of Computer Science, University of Exeter, Exeter EX4 4PT, UK., 1995. Research Report 329, available via http://www.dcs.exeter.ac.uk.

[6] M. P. Moore. Quantum-inspired algorithms and a method for their construction. Master's thesis, Department of Computer Science, University of Exeter, Exeter, UK, 1995.

[7] M. P. Moore and A. Narayanan. Quantum-inspired computing. Technical report, Department of Computer Science, University of Exeter, Exeter EX4 4PT, UK., 1995. Research Report 341, available via http://www.dcs.exeter.ac.uk.

[8] R. Penrose. *Shadows of the Mind.* Oxford University Press, 1994.

[9] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital and public-key cryptosystems. *Communications of the ACM*, 21(2), 1978.

[10] P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science*. IEEE Press, 1994.

[11] P. H. Winston. *Artificial Intelligence (3rd Edition)*. Addison Wesley, 1992.

[3]Nano = 10^{-9}, i.e. devices which are billionths of a metre large and work in billionths of a second. One nanometre is roughly the size of 10 atoms.

Chromosome	$\sum_i \frac{1}{d_i^2}$	Diversity Rank	Distance Rank	Rank Sum	Combined Rank	Fitness	Distance
3—7—2—9—4—5—6—8—1—3	0.011	11	15	26	14	0.000	190
1—2—5—4—6—3—7—9—8—1	0.005	01	10	11	04	0.025	148
1—3—8—5—9—2—4—6—7—1	0.006	05	07	12	07	0.001	136
1—9—2—8—6—7—3—4—5—1	0.014	13	14	27	15	0.000	183
5—6—7—8—4—9—1—2—3—5	0.036	15	04	19	10	0.000	121
7—8—9—3—1—2—6—4—5—7	0.014	14	05	19	09	0.000	129
9—4—8—5—7—3—1—6—2—9	0.008	08	01	09	03	0.074	119
2—8—1—7—4—5—6—9—3—2	0.008	07	13	20	11	0.000	178
2—6—3—8—4—5—7—9—1—2	0.009	10	06	16	08	0.000	131
5—2—7—4—1—3—6—9—8—5	0.007	06	03	09	02	0.222	121
1—6—5—8—7—2—9—3—4—1	0.009	09	12	21	12	0.000	172
1—3—5—6—7—2—8—9—4—1	0.006	03	09	12	06	0.003	148
1—2—8—4—9—3—5—6—7—1	0.006	04	08	12	05	0.008	138
1—9—8—2—3—7—4—5—6—1	0.012	12	11	23	13	0.000	161
1—3—2—5—6—4—7—8—9—1	0.005	02	02	04	01	0.667	120

Table 4. Table describing which of 15 other chromosomes will be selected for the next generation given the 'winning' chromosome 5—8—7—9—1—6—3—2—5.

U	Chromosomes									
u0	G	A	B	D	C	F	E	I	H	G
u1	E	C	A	B	D	G	H	F	I	E
u2	I	A	H	G	B	C	E	F	D	I
u3	A	D	C	I	E	B	G	H	F	A
u4	D	F	I	H	G	A	C	B	E	D
u5	C	A	G	H	D	B	F	I	E	C
u6	B	F	I	G	E	H	A	C	D	B
u7	I	H	F	E	G	B	C	D	A	I
u8	A	C	D	H	I	F	G	E	B	A

Table 5. A table describing interference crossover between 9 chromosomes in 9 different universes.