Recessive Trait Cross Over Approach of GAs Population Inheritance for Evolutionary Optimisation

Amr Madkour, Alamgir Hossain, and Keshav Dahal

Modeling Optimization Scheduling And Intelligent Control (MOSAIC) Research Centre
Department of Computing, University of Bradford, Bradford, BD7 1DP, UK
{A.A.M.Madkour, m.a.hossain1, k.p.dahal}@bradford.ac.uk

***Abstract* — This research presents an investigation into a new population inheritance approach using a concept taken from the recessive trait idea for evolutionary optimization. Evolutionary human inheritance recessive trait idea is used to enhance the effectiveness of the traditional genetic algorithms. The capability of the modified approach is explored by two examples *(i)* a mathematical function of two variables, and *(ii)* an active vibration control of a flexible beam system. Finally, a comparative performance for convergence is presented and discussed to demonstrate the merits of the modified genetic algorithms approach over the traditional ones.**

Index Terms— Genetic algorithms, PEAKS function, System identification, Flexible beam, Active vibration control.

1 Introduction

Over the last decade, genetic algorithms (GAs) have been extensively used as search and optimization tools in various problem domains, including the science, commerce, and engineering. The primary reasons for their success are their broad applicability, ease of use, and global perspective.

The concept of the GAs was first conceived by John Holland of the University of Michigan [1]. Thereafter he and his student have contributed much to develop this field. Most of the initial research work has been reported in various literatures [1-8]. However, there now exist several textbooks on GAs; many journals are now dedicated to promote research in this field. In addition, most GAs applications can also be found in various domain-specific conferences and journals [2].

This investigation demonstrates a modified approach for population inheritance to reduce the randomization "lucky" of the traditional GA crossover operator (TCGA). The GA with this Recessive trait Crossover is referred to here as (RCGA). The merits and capability of the modified approach is investigated by two examples *(i)* a mathematical function of two variables, and *(ii)* control parameter optimization of an active vibration control (AVC) system for a flexible beam.

The mathematical function 'PEAKS' in Matlab is chosen because it is a simple function with only two variables and contains a good number of local minimum that can trap the minimal seeker. On the other hand, an active vibration control algorithm for a flexible beam system is considered for control parameter optimization. It is worth noting that many researchers have already been used similar platform to investigate system identification and active vibration control system using TCGA [9-15]. For both examples, comparative performances of the TCGA and RCGA for convergence are measured and verified. Finally, these performances are presented and discussed to demonstrate the merits and capabilities of the RCGA over the Traditional Uniform Crossover Genetic Algorithm TUCGA.

2 Genetic Algorithms

GAs simultaneously evaluates many points in the parameter space and converges towards the global solution. GAs differ from other search techniques by the use of concepts taken from natural genetics and evolution theory.

The theory of evolution originated with Darwin in the nineteenth century, however, the idea that species mutate over time has been around for a long time in one form or another [16, 17]. Darwin suggested that in the universal struggle for life, nature "selects" those individuals who are best suited (fittest) for the struggle, and these individuals in turn reproduce more than those who are less fit, thus changing the composition of the population.

In fact, when a baby is conceived it is supplied with two copies of every chromosome: one copy from the mother and the other one from the father. Each parent donates of his/ her own chromosomes. The information from all of those genes takes together to makes up the blueprint or plan for the human body, its functions and its properties [18].

There are three methods of human inheritance, dominant, recessive and sex linked [19]. The sex-linked properties expressing depend on the person’s sex. For dominant properties, only one genetic trait is needed for this property to be expressed. However, if a genetic trait is recessive, a person needs to inherit two copies of the gene for the trait to be expressed. Thus, both parents have to be carriers of a recessive trait in order to express a child for that trait. If both parents are carriers, there is a 25% chance with each child to show the recessive trait and it is becomes 100% if the both have that recessive trait.

Fig. 1 illustrates the inheritance of a recessive property. Let us assume that two brown eyes parents careering of a recessive trait of blue eyes -blue eyes carrier- are married, there is a 25% chance with each child to be blue eyes, a 25% chance with each child to be pure brown eyes, and a 50% chance with each child to be blue eyes carrier like there parents. From this second generation; if one of the carrier children married the blue eyes one there is a 50% chance with each child to be blue eyes, a 0% chance with each child to be pure brown eyes, and a 50% chance with each child to be blue eyes carrier. From the third generation of this relationship, if the two blue eyes children married, 100% of there children will be blue eyes.

From this example, we can understand why the blue eyes, yellowy heir and white skin are very common at the western societies, in contrast, black hair and eyes and dark skin are common at the African societies.

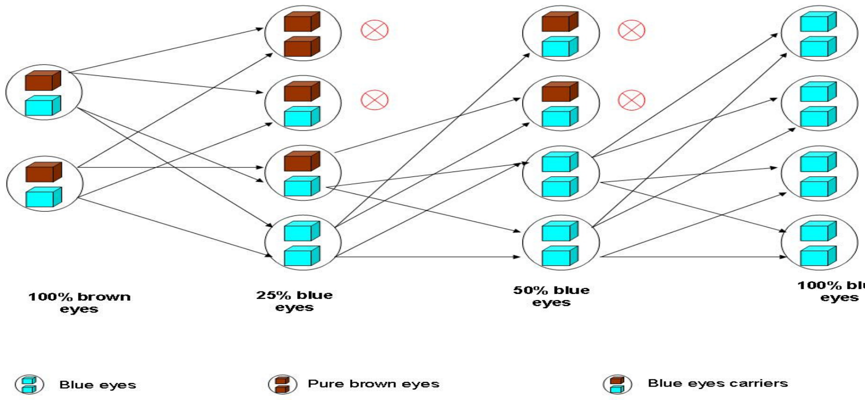

Fig. 1 : Inheritance of blue eyes recessive property

3 Genetic Algorithms as a computation work

According to Goldberg [3], GAs are different from more normal optimization and search procedures in four ways:

- GAs work with a coding of the parameter set, not the parameters themselves.
- GAs search from a population of points, not a single point.
- GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge.
- GAs use probabilistic transition rules, not deterministic rules.

3.1 Traditional Crossovers Genetic Algorithms (TCGA)

The TCGA evolution can be summarized as follows: create a population of individuals, evaluate their fitness, generate a new population by applying genetic operators, and repeat this process for a number of times.

To start the optimization process, TCGA generates randomly or with other means a population of individuals, say N individuals. Generally, each individual in the population consists of encoded strings representing a solution. Each solution has a fitness value evaluated by the some objective function and constraint satisfaction. The individuals with higher fitness value are usually selected and sent to the mating pool. Different selection methods such as roulette wheel selection and stochastic universal sampling can be used for this operation. Solutions having higher fitness values are most likely to survive for the next generation. Crossover operator is used on these strings to obtain the new solutions that inherit the good and bad properties of their parent solutions. The crossover operator works on randomly selected pairs of selected solutions from mating pool with certain crossover rate. The crossover rate is defined as the probability of applying crossover to a pair of selected solutions. There are many ways of defining this operator such as single point, double point, multipoint and uniform crossover. These traditional crossover operators are discussed in [6].

3.2 Recessive trait Crossover Genetic Algorithms (RCGA)

The TCGA works on randomly pairs of selected solutions from mating pool with certain crossover rate. This operation exchanges the genes between the two random selected solutions. Using the concepts taken from the recessive property inheritance and Darwin theory of evolution the RCGA produces children by selecting the common genes between both parents, and choosing the remaining genes randomly according to the fact of the complementary of all of the chromosome parts makes its survival fitness. The main difference between the two algorithms is the way of how the new population is inherited from the previous generations.

Now let us assume that two parents have the eight genes chromosome as shown in Table 1. It is worth noting that those parents have common genes at (1, 3, 5, and 6). Over the evolution process the survival fitness of two parents depend on those common genes. So we will keep these common genes without any change when children are reproduced and try to make better children by introducing the different genes using the four possible binary combinations randomly. This is the only random operation in this crossover approach. The new solutions are as shown in Table 2.

Table 1: An example for two parents have 8 genes chromosome

Gene NO.	1	2	3	4	5	6	7	8
Parent 1	0	1	1	1	0	1	0	1
Parent 2	0	0	1	0	0	1	1	0

Table 2: The reproducing in table 1 using RCGA

Gene NO.	1	2	3	4	5	6	7	8
Child 1	0	1	1	1	0	1	0	0
Child 2	0	1	1	0	0	1	1	0
Child 3	0	0	1	0	0	1	0	1
Child 4	0	0	1	1	0	1	1	1

Referring to recessive trail behavior, the selection of the married parents is very important as both of them should be at lest fitness trait carrier. This selection operation will be done by sorting the old populations according to their fitness and then reproducing the first parent with the second to generate four new populations, as shown in the above example, and so on.

The RCGA can be written as:

1. Create a random population of *N* individuals

2. Evaluate their fitness.
3. Sort the individuals in the population according to their fitness.
4. Choose the best *N/2* individuals to generate the new *N* population.
5. Generate the new *N* individuals by reproducing the nearest two sorted parents keeping the common genes and change the different genes using the four possible binary combinations randomly (every two parents generate four children).
6. Apply mutation operation with a probability.
7. Repeat steps from 2 to 6 for the best fitness value.

4 Numerical examples

To demonstrate the effectiveness of the RCGA as compared to the TUCGA, two numerical examples are presented below. The parameter sensitivity analysis is done for the first example using a set of mutation rate and set of population size for a certain number of experiments. Each experiment started with the same random initial population set for both algorithms.

4.1 PEAKS function

PEAKS is a function of two variables, obtained by translating and scaling Gaussian distributions evaluated as [13].

$$f(x, y) = 3(1 - x)^2 e^{-x^2 - (y+1)^2} - 10(\frac{x}{5} - x^3 - y^5) e^{-x^2 - y^2} - \frac{1}{3} e^{-(x+1)^2 - y^2} \quad (1)$$

The surface plot of this function is shown in Fig 2, it can be obtained directly by using MATLAB peaks function [21]. This function was chosen because it contains a good number of local minimum that can trap the minimal seeker.

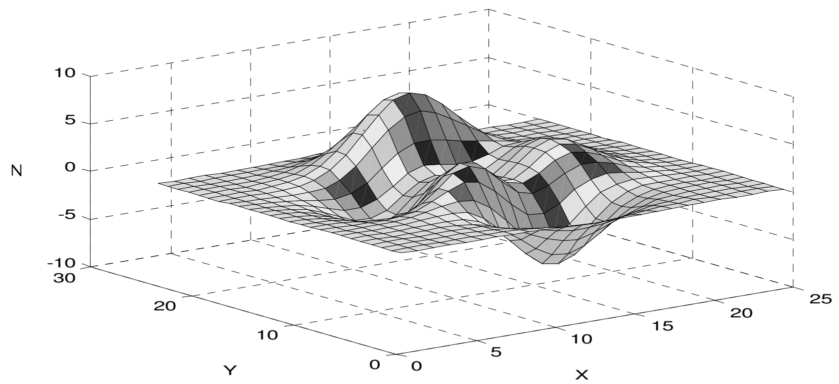

Fig. 2: Matlab peaks function

Using the minimum value of the peaks as a fitness function, one hundred experiments for every item of both, mutation rate and population size were done to compare the two algorithms. Those experiments were done by using mutation rate set defined as [0% %5 10% 15% 20% 50%] and population size set defined as [20 60 100 200 300 500].

The average results obtained after running the two algorithms for 100 generations using the same initial populations of 16-bit representation for each variable of the one hundred experiments is shown in table 3, where 'T' represents TUCGA and 'R' represents RCG.

Table 3: The average of the peaks function values obtained after 100 generations

		Population size					
Mutation		20	60	100	200	300	500
0%	R	-5.35204	-6.31998	-6.48472	-6.53505	-6.54830	-6.55063
	T	-5.03608	-6.38537	-6.49890	-6.54637	-6.55017	-6.55107
5%	R	-6.42770	-6.52496	-6.54628	-6.55113	-6.55113	-6.55113
	T	-6.37187	-6.49793	-6.54395	-6.54953	-6.55113	-6.55113
10%	R	-6.41143	-6.52071	-6.55107	-6.55113	-6.55113	-6.55113
	T	-6.36684	-6.53314	-6.54641	-6.55113	-6.55113	-6.55113
15%	R	-6.48776	-6.54639	-6.55113	-6.55113	-6.55113	-6.55113
	T	-6.45994	-6.54775	-6.55111	-6.55113	-6.55113	-6.55113
20%	R	-6.52503	-6.55110	-6.55111	-6.55113	-6.55113	-6.55113
	T	-6.52573	-6.54930	-6.55105	-6.55110	-6.55111	-6.55112
50%	R	-6.48626	-6.52943	-6.53825	-6.54337	-6.54678	-654856
	T	-6.47995	-6.52497	-6.53935	-6.54352	-6.54628	-6.54848

Table 4 shows the generations number average of the one hundred experiments to obtained the peaks function minimum value of (-6.551) using the same initial populations of 16-bit representation for each variable for the two algorithms.

To explore further, we executed both algorithms for 100 generations starting from the same initial population set of 60 members with 16-bit representation and 10% mutation rate. The performance for both the algorithms is shown in Fig 3. The minimum value of the Matlab PEAKS function (equation 1) as a fitness function and its corresponding two variables (x, y) is show in Table 5. It is noted that the RCGA successfully obtained the global minimum of the function after 10 generations. In contrast, the TUCGA achieved very close value of the global minimum after 15 generations using the same initial populations

Table 4:		Execution time (sec) average taken for 100 generations					
		Population size					
Mutation		20	60	100	200	300	500
0%	R	82.12	52.41	44.51	20.42	19.78	16.74
	T	Fail	55.65	47.59	25.65	23.48	20.25
5%	R	55.69	45.18	25.74	17.3	12.42	10.76
	T	60.50	57.42	47.90	22.62	21.62	17.56
10%	R	49.20	36.34	21.9	17.8	13.66	12.32
	T	53.58	49.44	37.30	28.96	24.90	23.79
15%	R	47.36	33.46	29.38	22.56	18.76	16.14
	T	79.56	63.76	50.40	35.88	32.96	29.62
20%	R	72.37	58.12	42.12	35.04	30.22	23.12
	T	98.23	87.60	76.40	51.68	50.96	39.64
50%	R	Fail	Fail	100	99.52	99.02	98.08
	T	Fail	Fail	Fail	Fail	Fail	Fail

Table 5:	The result after 100 iterations with 16- bit representation, 10% mutation rate, and population size of 60.		
Algorithm	F(x, y)	x	y
TUCGA	-6.3259	0.2579	-1.5000
RCGA	-6.5511	0.2283	-1.6255

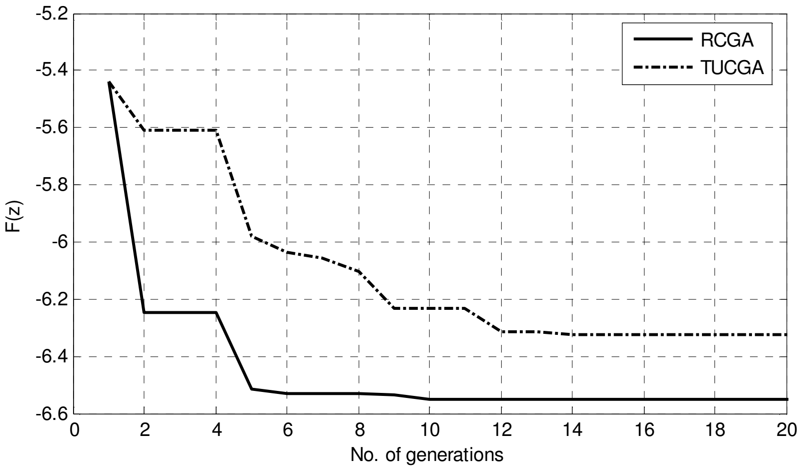

Fig. 3: Minimum value evaluation of the peaks function got by the two algorithms

4.2 The flexible beam system

To demonstrate the effectiveness further in real application, we consider a cantilever beam system with a force $F(x,t)$ applied at a distance x from its fixed (clamped) end at time t. This results a deflection $y(x,t)$ of the beam from its stationary position at the point where the force has been applied as shown in Fig. 4. This flexible beam was used by many researchers as a platform to investigates there algorithms for system identification and active vibration control [9-15].

TUCGA and RCGA are used to estimate the parameters of the AVC system. The algorithms for control parameters estimation and similar work using TCGA is reported earlier [11].

Fig. 5 depicts the auto-power spectral density before cancellation and the auto-power spectral density after cancellation in implementing the AVC system using TUCGA, and RCGA. Fig. 6 shows the time-domain performance in implementing the AVC system using TUCGA and RCGA, where the solid lines represent fluctuation of the beam at the end point before cancellation.

It is noted that the convergence achieved using the RCGA is consistently better than the TUCGA. The proposed RCGA using a new population inheritance approach through the recessive trait crossover offers for propagating good building blocks of genes to subsequent generations. These features RCGA, we believe, offered a significant impact in providing better results and convergence.

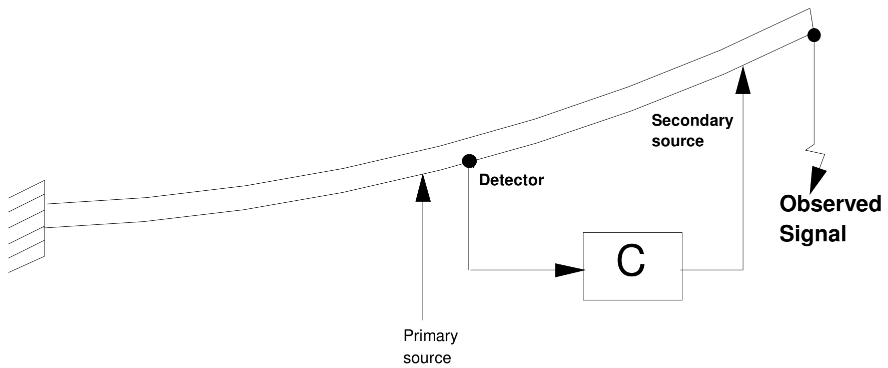

Fig. 4: Active vibration control structure

5 Conclusion

This paper has presented the investigation into a modified GAs population inheritance using a concept taken from the recessive trait idea. A comparative performance of the modified and traditional approaches has been presented and discussed through a set of experiments.

Two numerical examples have been provided to demonstrate the merits and capabilities of the modified cross over approach. In both cases, it is noted that using same initial populations, bit representation, and mutation rate, the RCGA offered better convergence, higher accuracy and faster solution for each problem. Farther more, the RCGA is very sample and easy to use for any numerical optimization problem not only as a minimum or maximum seeker but also for any fitness function.

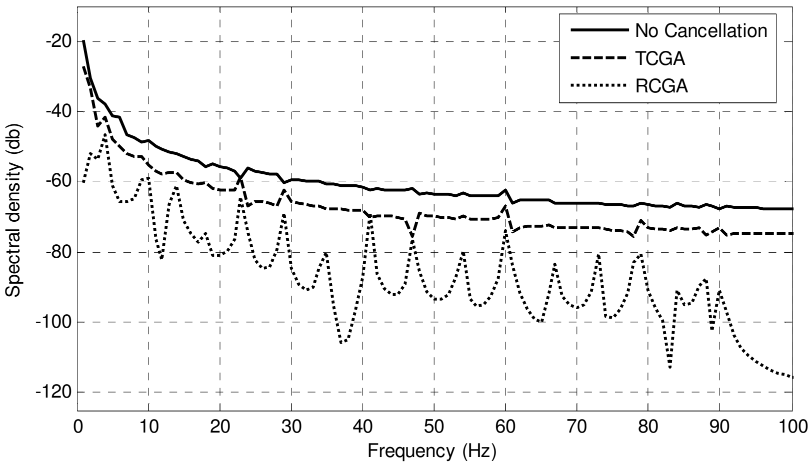

Fig. 5: Performance of the TUGA and RCGA in auto-power spectral density

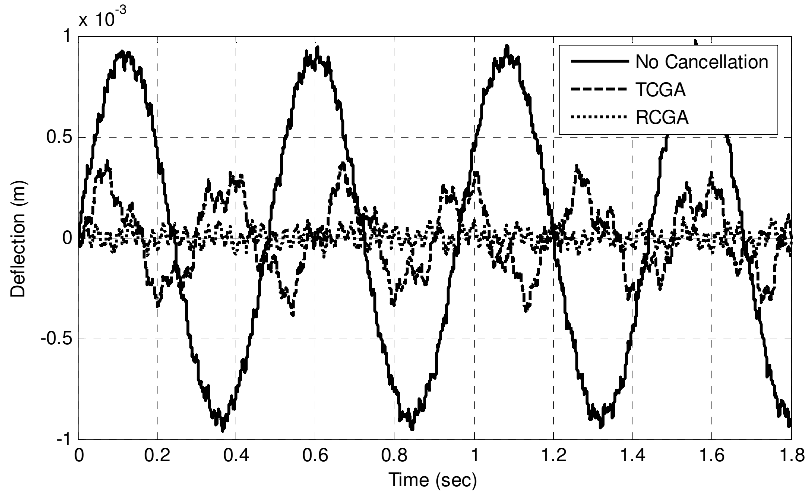

Fig. 6: Beam fluctuation at the end point

REFERENCES

[1] J. H. Holland, 'Adaptation in Natural and Artificial Systems' The University of Michigan Press, 1975.

[2] K. Deb, 'Multi-objective optimization using evolutionary algorithms', John wiley & sons LTD, 81-169,2001.

[3] D. E. Goldberg, 'Genetic Algorithms for Search, Optimization, and Machine Learning', Reading, MA: Addison-Wesley, 1989.

[4] M. Gen, R. Cheng, 'Genetic Algorithms and Engineering Design', New York: Wiley, 1997.

[5] Z. Michalewicz, 'Genetic Algorithms + Data Structures = Evolution Programs' , Berlin: Springer-Verlag, 1992.

[6] M., Mitchell, 'Introduction to Genetic Algorithms' Ann Arbor, MI: MIT Press 1996

[7] M. D. Vose, ' Simple Genetic Algorithm: Foundation and Theory' , Ann Arbor, MI: MIT Press 1999

[8] K.F.Man, K.S.Tang and S.Kwong, 'Genetic Algorithms: Concept and Designs', Springer, London.,1999

[9] P. K. Kourmoulis, 'Parallel processing in the simulation and control of flexible beam structure systems', PhD thesis, Dept. of Automatic Control & Systems Engineering, The University of Sheffield, 1990.

[10] M. A Hossain. 'Digital signal processing and parallel processing for real-time adaptive noise and vibration control', Ph.D. thesis, Department of Automatic Control and System Engineering, The University of Sheffield, UK.,1995.

[11] M. A. Hossain,and M. O. Tokhi 'Evolutionary adaptive active vibration control, Proc Inst. Mechanical Eng'., 211(1), pp. 183-193, 1997.

[12] M. O Tokhi,. M. A. Hossain, and M. H. Shaheed, 'Parallel Computing for Real-time Signal Processing and Control,' Springer, London,2002.

[13] A. A. M. Madkour, M. A, Hossain, K. P. Dahal, and H. Yu, 'Real-time System Identification using Intelligent Algorithms', Proceedings of IEEE SMC UK-RI Chapter Conference 2004 on Intelligent Cybernetic Systems, pp. 236-241,2004.

[14] M. A, Hossain, A. A. M. Madkour, K. P. Dahal, and H. Yu,. Intelligent Active Vibration Control for a Flexible Beam System', Proceedings of IEEE SMC UK-RI Chapter Conference 2004 on2004 on Intelligent Cybernetic Systems, pp. 236-241,2004.

[15] S. Z. Mohd Hashim, M. O. Tokhi, and I. Z. Mat Darus, 'Genetic Adaptive Active Vibration Control of Flexible Structures', Proceedings of IEEE SMC UK-RI Chapter Conference 2004 on2004 on Intelligent Cybernetic Systems, pp. 166-171,2004.

[16] G. Himmelfarb, 'Darwin and the Darwinian Revolution', Doubleaday & Company Inc. New York, 1959

[17] Fogel, D. B., Evolutionary Computation, Toward a New Philosophy of Machine Intelligence, IEEE Press, Piscataway, New Jersey, 1995.

[18] http://www.muscle.ca/content/index.php?id=315, Produced by Muscular Dystrophy Canada,Updated: December 2002

[19] E. Richard and M. D Robert.,"Nelson Essentials of Paediatrics", 3ed edition,W.B. Saunders Company, ISBN 0-7216-7229-9 , 1998.

[20] J. S. R. Jang, C. T. Sun, and E. Mizutani,'Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence'Prentice-Hall Inc.,USA,ISBN 0-13-261066-3,1997

[21] http://www.mathworks.com/access/helpdesk/help/techdoc/ref/peaks.html, Produced by the Mathworks Inc., December 2005.