EVOLUTION WITH A TELEOLOGY: THE GENETIC PROGRAMMING HEURISTIC APPROACH TO MODELING

Dominique Fischer

Universiti Malaya, Kuala Lumpur

Corresponding author: Prof. Dominique Fischer domfischer@gmail.com

Abstract

This paper illustrates the power of Genetic programming (GP) with a variety of simple examples. The general approach is described and the results are compared to regressions and Artificial Neural Network results. The superiority of GP results appears to be quite convincing. Less convincing could be the nature of the Darwinian metaphor that underpin the whole concept.

Keywords: Genetic programming, heuristic models, metaphors.

This paper is meant to introduce Genetic programming in simple terms, to illustrate the process with simple (and less simple) examples and then to warn against the temptation to take the Darwinian Evolutionary metaphor beyond what Genetic Programming actually does.

Since most of us don't really speak Greek it could be safe to define our terms.

- Teleology (Greek telos = end; logos = discourse) is the research of finality. In philosophy, it is based on the Aristotelian idea that the universe has a design and purpose. It can be opposed by Darwin's 'telosless' random-evolution of the natural world..
- Heuritic (from the Greek `heuriskein': to discover) is the research of results by trial and error.

1. Genetic programming

Genetic programming (GP) introduced by John Holland (1975)[1], is now commonly used in design problems where no 'optimal' and unique solution can be found by deterministic modelling. Thus, GP is commonly is used in electronic design, in engineering, biomedical sciences and applied mathematics.

More recently, as expected, it has also been discovered and applied to solve analytical and decision making issues in areas such as finance, marketing, behavioural economics or operation research. Predictably, a recent research (from the US, where else?) even applied GP to deal with cyberterrorism intrusion (Hansen, Lowry et al. 2007). Some of the relevant references limited to the areas of economics, finance and operation research are listed in references.

Genetic programming works particularly well with financial types of problems and decision driven issues because:

- They are payoff driven. The targets are measurable (in dollars, time, customer base, degree of satisfaction, etc.)..
- They are quantitative, and well-suited to parameter optimisation;

[1] For a review of J. Holland role on the field of economics, see Chen, S (2001)

- They are robust, allowing a large margin of freedom that is not acceptable for econometric methods. In particular GP calculations do not have to be constrained by any of the traditional Gaussian Markov Ten Commandments in econometrics.

So far, the GP approach has received little attention in the various property fields. The only references traced so far are a test of efficient markets based on long term series of price data for a quoted property investment company (Fyfe, Marney J. et al. 1999) and a study of residential submarkets (Lewis, Ware et al. 2001). It may be worth noting that these papers were not published in `property journals'.

Genetic progamming relies entirely - in general and in some of it's operational details – on the metaphor of Darwinian evolution. From extremely simple kernels of calculations, GP produces increasingly complex functions (made of small bits of code) that will eventually reach a pre-determined target. In a way, this approach was already germane in the innovative Adaptative Estimation Procedure that was applied to property valuation (Carbone and Longini 1977) but here the nature of 'adaptative procedure' is radically different.

In the words of the best know GP evangelist (Koza, 1992):

> *We breed the population of computer programs using the Darwinian principal of survival and reproduction of the fittest and the genetic operation of recombination (cross-over). Both reproduction and recombination are applied to computer programs selected from the population in proportion to their observed fitness in solving the problem. Over a period of many generations, we breed computer programs that are ever more fit to solve the problem at hand (Koza, 1992 p.4)*

2. The genetic analogy: from genes to strings of bits.

In nature, the mixing of genetic material proceeds through an equal exchange (half from her and half from him) of genes through the twisting around of the chains of DNA and associated proteins (chromosomes). This process of recombination is – usually – flawless except of course in the case of rare mutations that can lead to a new phenotype and thus to some evolutionary branching.

In genetic programming the 'twisting and fusing of chromosomes' is the metaphor for the recombination of strings (binary string bits) of assembler code such as.

1001010111010100101001110110111011111101

A large number of such 'models' (binary strings) are used to calculate some outcome (the target). As you may guess, it is very unlikely that any of them will lead to the right answer, but some of them may fit better than others. Each model receives a 'fitness score' and the best scorers are randomly selected, intermixed (0 and 1 from each.. not necessarily in equal numbers) and rescored again. Like in real sexual reproduction, each 'descendant; thus is endowed with chromosomes (data string) of each 'parent'.

The best scorers will have a higher probability of being combined in the following cross over (selection of random 'genes' ie: 0 and 1 from the model.

For example, from the two strings below:

10001001110010010

01010001001000011

The computer will, at random, choose a bit the length, say at position 9, and swap all the bits after that point. Now the descendant will be look like:

10001001101000011

01010001010010010

In a first term of a 'model' that comes out as: (v[0] - 0.5) (see the full formula later on in 4) and using four-bit code to represent the variable and operators[2] characters these first two terms would be coded coded as:

0001101101011101001O

Most readers (and certainly not this author) would not have the patience to push the example much further, but the general idea should make more sense now.

3. How to make it run in practice.

To run a GP model, you need to load a 'training' matrix and a 'validation matrix'. Both sets of data should come from the same population and have the same variables. Typically, you could divide your population in two subsets (chosen randomly) and use the two sets for training and validation purpose. The targets (the model results) are known and you could select your input variables on the basis of prior theoretical or empirical knowledge of the relationships. You could even cheat (as I did here) by running some regressions or Artificial Neural Network test to make sure that the chosen variables are relevant.

You have various options to control on the process. Interestingly these options are laid out in a very evocative 'natural selection language (choice of cross-over rates, mutation frequencies, number of demes[3], cross-over between demes, migration rates among demes).

Then, you let the model run and you monitor the progress by observing the graphical 'searching process' (as illustrated later on) and by following the spreadsheet presentation of the results.

The stopping rules can be determined in the program set up, but typically you would stop the run when you fitness levels (proximity between target, validation and calculated results) is satisfactory.

- In function fitting problems, the program calculates the square of the residuals between targets and results. The user may choose to stop when this measurement is not improving.
- For classification problems, it calculates a percentage of hit-and-miss outcomes. Here again, the user may stop when the rate is good enough. In the example presented below, the hit rate was close to 96% in a few seconds of running time.

Finally, when finished, you can visualise the resulting graph and spreadsheets, you can compare the different results for the training and the validation sets and you may want to keep your best performer.

The other important output is a full sub-program written in Assembler or in C++ (the 'professional' version of this package also offers a Java option). This sub-program can then be integrated to a complete program that could manage the treatment of input and presentation of outputs. It can also be linked to other programs to contribute to the solution of more complex procedures. Unfortunately, in this version of the package, the results are not turned into an Excel or SPSS equivalent type of interface. Thus, it does require a sufficient knowledge of C++ to exploit the output to its full extent.

---

[2] 0: 0000, 1: 0001; 2: 0010; 3: 0011; 4: 0100; 5: 0101; 6: 0110; 7: 0111; 8: 1000; 9: 1001; +: 1010; -: 1011; *: 1100; /: 1101

[3] Demes are geographically separated populations. In nature, the separation of the species contributes to more genetic diversity. The migration rates between demes determines the amount of blending and crossovers. The program offers the options of choosing the number of demes and the rates of migration betweem demes. This feature seems to improve the production of a larger variety of models (strings of bits).

4. A simplistic example: land price and lot size

The trivial – and typical first example in any basic regression course - is used here to determine the influence of lot size and distance from CBD on lot prices.

Price = f (lot size).

The model generates a 'program' in C++ that can be translated in 'almost' English as:

```
((v[0] - 0.5) + v[0]) + ((v[0] - 0.5) + v[0])) + v[0]) / 0.5) + (fabs(((0 * v[0]) + 0.5)) / (((v[0] - 0.5) + v[0]) + ((v[0] - 0.5) + v[0])))) - 0.5);
```

Thus, we can see that the 'genetic' transformations are here limited to subtraction and division by a constant (0.5), and nothing else. Predictably, the results are right on the spot (it is a straight line), however why bother? We did not need such a heavy machinery to reach this result: a pencil and cheap plastic ruler would have done the job quite nicely.

Even more complex land pricing models do no really require such a fancy GP treatment (Fischer and Lai Pi-Ying 2007). As shown in the quoted paper, a multiple regression treatment is almost good enough. The results obtained from Artificial Neural Network treatments are indeed better than regression procedures, and very close to those obtained from GP. However, ANN treatments are more explicit and easier to apply to predictive models. So, once more, why bother?

5. Mimicking a hedonic model… with one variable only.

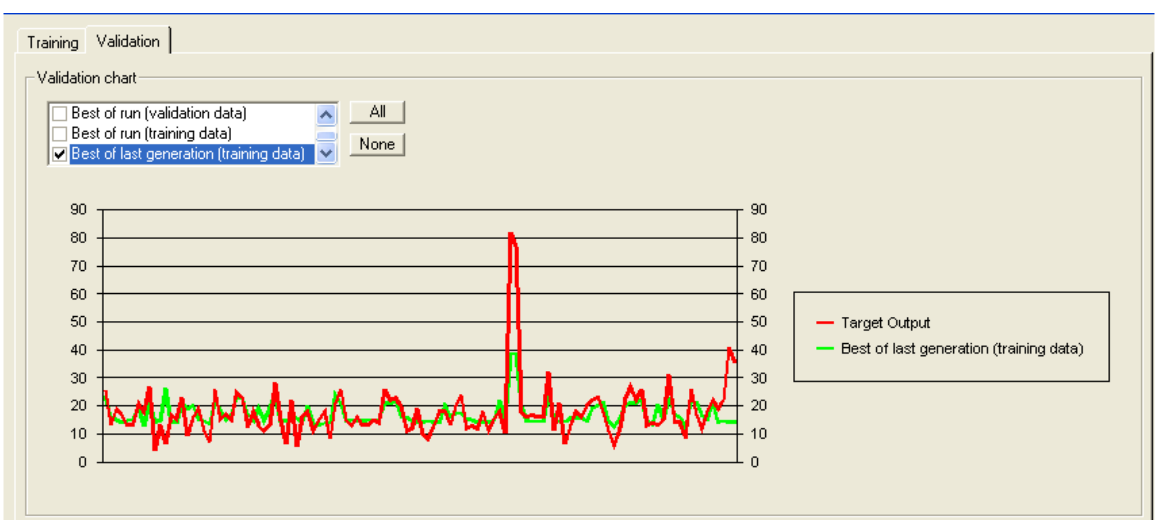
Why bother?... because GP is not meant to be used to find easy deterministic solutions. GP is mostly useful to deal with problems that do not have a 'calculable' outcome, or to problems that do not rely on a clear explanatory model, or on treatments that cannot rely on a sufficient number of variables to explain the outcome.

To keep our comparisons 'comparable' we will now use a house pricing example based on observed prices (1999 – Perth, Western Australia). The sample is limited to duplex types of housing and we had to scale the prices by a factor of 10 000 to make the program work.

Here we try to predict the price of Duplex units on the basis of only one variable (Duplex surface). Thus, we drop all the other available traditional 'hedonic' variables (distance from CBD, construction type, roof, number of rooms, number of bath, garage, etc).

After a mere 10 seconds running time the 'running graph' looks like the following illustration.

Figure 1: Duplex house price calculated from the house surface only: the output after 10 seconds



In this case, the run could have been stopped after 1 minute since the gain in precision was negligible. However, as usual when playing with a new toy, the temptation is to let it run as long as you feel like. Here, after 5 minutes the results come with a surprising accuracy (see the output in appendix 1).

The average difference between observed prices and model generated prices is -980 $ and the standard deviation of the 'residuals' is 8 905 $ (to compare with average prices of 183 372 $

Running a regression on the same information leads to a coefficient of determination of only 15% and a standard error on residuals of 94 835 $. No contest indeed!

Further – it may worth repeating again – the Genetic Programming model requires absolutely no hypothesis on the shape of the model or the statistical nature of the variables.

Could we obtain better results with a run of Artificial Neural Network? The answer must be negative. The ANN treatment is far from producing results of the same accuracy.

Figure 2: ANN result on the one-variable 'hedonic model'

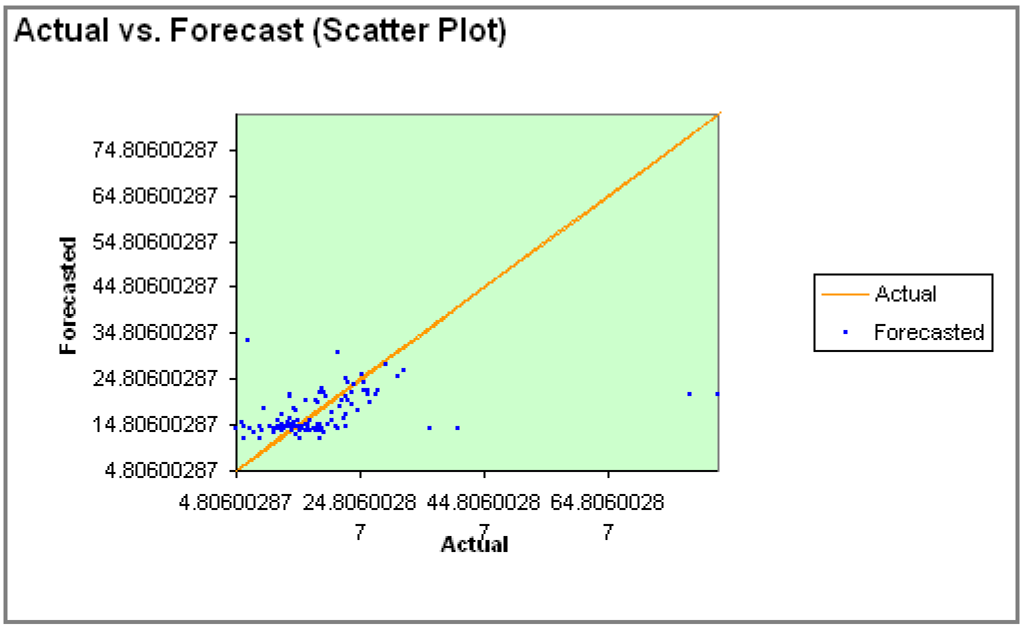


Table 1: ANN performance on the Duplex price-size calculation. Not great!

	Training set	Test set
# of rows:	96	20
Average AE:	4.2461876	8.451561
Average MSE:	66.270296	220.60129
Tolerance:	10%	30%
# of Good forecasts:	26 (27%)	14 (70%)
# of Bad forecasts:	70 (73%)	6 (30%)

6. A classification problem

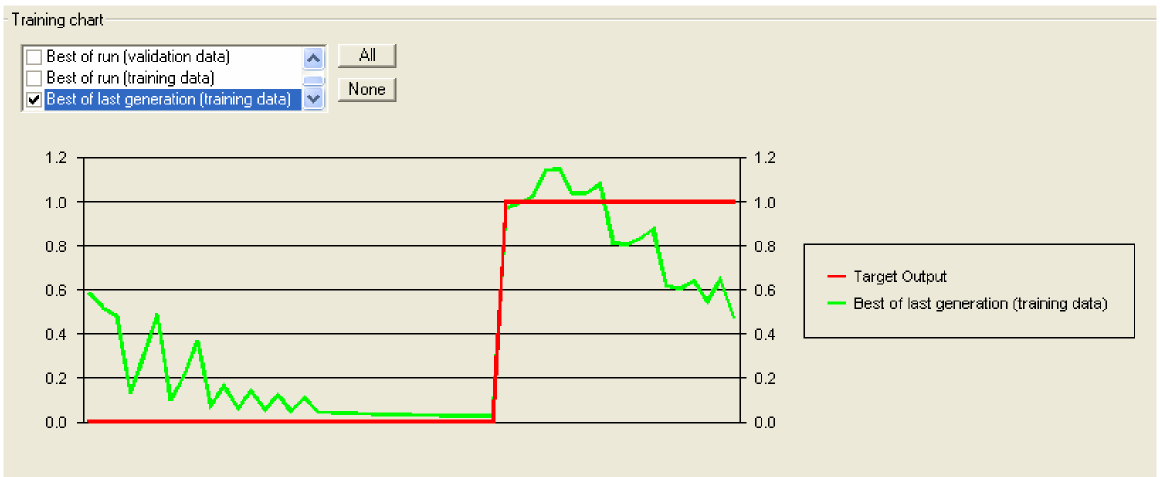
The previous examples (land price predictions) are problems similar - in their structure and objectives - to multiple regression analysis: a set of input variables are used to determine the values of a numerical dependent variable (target variable).

GP can also be applied to problems, where the outcome is a hit or miss result. Such problems could also be treated with regression analysis (with a dummy dependent variable), or better with a logit model. The treatment of mortgage default is a good example of such problem, where underwriting criteria are used as input variables to predict a default outcome coded as 0 (no default) or 1 (default).

The procedure is now briefly illustrated in the case of a – very clearly – contrived example (see the data set in appendix 2). The underwriting criteria chosen here are the usual suspects: household income, length of residence, ‘sin level’ (credit rating impediments), % of equity and house value.

The Genetic programming package used here has different stopping rules for classification problems and essentially – you can manually stop it whenever the hit rates are satisfactory. In our case, the hit rate was up to 95.9% on both the training and validation test within less than a minute and the intermediate output graph (after 12 seconds) is presented below in Figure 3

Figure 3: The Mortgage default model after 12 seconds.



In contrast, the Artificial Neural Network model did not perform well at all with this type of classification problem.The summary of the ANN output is presented in Table 2

Table 2: The mortgage default ANN output.

	Training set	Test set
# of rows:	41	8
Average AE:	0.0809943	0.28308077
Average MSE:	0.0344385	0.19441146
Tolerance:	10%	30%
# of Good forecasts:	14 (34%)	0 (0%)
# of Bad forecasts:	27 (66%)	8 (100%)

From these two simple illustrations we can conclude that Genetic programming works. Even in the very naïve hands of a first time experimenter using the cheap version of a commercial package.

Of course, as with Artificial Neural Network, the model is a very much a black box, but – in view of the power of the tool – this `black boxness’ can easily be tolerated. It should also be said that – for many ‘down town’ users, multiple regression software packages are – at least – as obscure and impenetrable: this does not seem to prevent the widespread usage of regression results.

Having established the usefulness of the instrument and – after this maiden flight – hoping to use it more in the future, I would like to briefly discuss the nature and limitation of the Darwinian metaphor and thus – at last - clarify the meaning of this paper's title.

7. Let's beware of metaphors

Genetic programming borrows its name and metaphor from the domain of biology in the same way as, previously, artificial neural network borrowed its own metaphor from neurology and medical sciences.

This reverential support from the 'real sciences' is quite typical of the epistemological bias that has burdened the development of economics and other social sciences. However in GP, the borrowing was not initiated by economics or social sciences but by other 'real sciences' such as computer sciences and operation research. Still, we suggest that the analogies to the natural world are used – to a certain extent – as a 'blinding by science'[4] argument: this reverential reference to biology has the effect of making the argument more authoritative. In other words the reference to fundamental natural and biological processes confers nobility and credibility to the esoteric and non-intuitive machinery behind the algorithms.

This recourse to the biological metaphor is particularly interesting in the case of artificial neural network. Nowadays, most introductory presentations of artificial Neural Network applications rely on nifty Power Point pictures of the brain with neurons and synapses (preferably in colour) actively engaged in smart connectivity. Unfortunately, this picturesque description of artificial neural network is far too reductionist. ANN algorithms are much (much, much) simpler than real biological brains neural functioning.

ANN simply proceeds through searching algorithms that filter out the non-performing branching of quasi-random calculations. The fact that the screening may go through many levels does not change the nature of the process and certainly does not make more 'like a real brain'. ANN is nothing more that a streamlined heuristic procedure. The efficiency and performance of ANN is indeed quite impressive, but the over-analogizing it to biological brain chemistry by many of its proponents is borderline false representation.

In a sense, Genetic programming suffers from the same 'over-analogising'. However, at least to a certain point, the analogy has more pedagogical power than the analogy used in ANN. As we have seen, the borrowed language and concepts are quite useful similes that do facilitate the understanding and probably the development of the GP instruments. Still, the metaphor is only a metaphor and it should not be pushed beyond its pedagogical function. GP algorithms and real natural selection are quite different: some of the differences are obvious, some of them are more subtle.

— ***Incomparable time scales***

One of the obvious difference is the vast difference in time scale. Natural selection spreads out in the past and in the future over an unknown billions of years. The real biological computer runs for a very very long period and it runs very very slowly. The rate of mutation is slow (e.g. for Homo sapiens, significant mutations seem to occur only every 10 000 years) but still, over the eons the number of steps taken by the 'algorithm' is immensely larger than the one taken by the most powerful computer programs. In fact, commercial software providers sell their products on the basis of their extreme speed. The package used for the purpose of this paper runs 'only' a few millions 'tournaments' for the land pricing example in less than 30 minutes on a notoriously sluggish PC and Intel based system.

— ***Natural selection is extremely wasteful***

---

[4] Expression borrowed from R. Dawkins (2002).

Once again, the numbers are intelligible, but an unknown and prodigiously high number of evolutionary attempts are wasted in the natural selection process. Zillions of variants and species just do not make it.

In contrast, GP algorithms try their best to minimise the wastage by imposing elimination rules on the less performing models. Thus, a ruthless screening of the losers has the advantage of producing a smaller number of 'losing' descendants. Cutting the evolution branch as early as feasible has the beneficial effect of reducing the wastage and – more to the point – of reducing memory requirements and computer running time.

— ***Natural selection has no teleogy***

This point may be less obvious and certainly less palatable to Theists. Natural selection has no final overreaching objective. It certainly does not try to reach some form of ideal survivor. Natural selection occurs in perpetuity without any 'target'. The engine of the process is not its finality but only the fundamental genes reproductive necessity.

The observable present result of evolution (a few millions species and one specie that can even count the others) is extremely transitory and subject to constant transformations. No specie will survive for very long and certainly no specie can be considered as 'closer to the target'. Evolution churns along multitude of variants that adapt to the changing environments, the variants are short lived (relatively speaking) because the environments are changing fast (again, relatively speaking).

By contrast - and this is the point made in the title of this essay – GP algorithm have very specific targets: it has a *'teleos'*. Genetic algorithms have the declared objective to find the 'fittest' the model that will track the target as close as possible. The targets are defined narrowly (a vector of numbers) and the algorithm is 'trained' to get results that are the best approximation of the results observed in the validation matrix. The process is not normative: it does not try to find some 'optimal solution' (optimal?.. with respect to which criteria?). It is a pure heuristic: it tries, fails, tries again and eventually gets close enough to stop.

Once again, the GP Darwinian metaphor is useful but it is only a metaphor. Our ingrained scientific scepticism should keep us vigilant enough not to turn our biological metaphors into allegories. No one but Deidre Mc Closkey could put it better:

> *When the metaphors do battle with the story, the result is nonsense, nonsense that can hurt when people believe it. People do. People especially believe in allegories, such as the combined metaphors and stories of economics, because an allegory in its completeness protects the illusion of prediction and control. (McCloskey 1992) (p. 97)*

Appendix 1

Table 3 The 'one-variable' Duplex pricing GP best run.

	Duplex surface in m2	Observed prices in AUD (rounded)	Prices predicted by the best GP run (5 minutes running time)	Difference (in AUD)
	78	48,100	44,855	3,245
	91	56,300	62,820	-6,520
	56	61,700	67,297	-5,597
	82	62,000	62,777	-777
	221	67,000	78,425	-11,425
	78	70,100	84,855	-14,755
	70	76,800	83,464	-6,664
	55	87,100	82,949	4,151
	82	87,200	82,777	4,423
	73	89,200	86,055	3,145
	115	94,500	91,327	3,173
	82	104,300	122,777	-18,477
	76	108,600	107,154	1,446
	71	109,400	128,418	-19,018
	82	114,900	122,777	-7,877
	79	115,100	125,771	-10,671
	93	117,100	138,252	-21,152
	82	117,900	122,777	-4,877
	83	119,200	118,787	413
	105	122,000	122,125	-125
	73	122,200	116,055	6,145
	86	123,500	136,991	-13,491
	80	126,000	134,735	-8,735
	76	126,300	127,154	-854
	80	127,700	134,735	-7,035
	80	128,100	134,735	-6,635
	80	130,600	134,735	-4,135
	90	132,200	144,765	-12,565
	82	133,500	130,777	2,723
	85	133,600	141,056	-7,456
	99	134,500	136,419	-1,919
	98	134,600	154,812	-20,212
	136	134,800	135,758	-958
	140	135,600	139,267	-3,667
	84	135,700	139,919	-4,219
	75	136,900	139,120	-2,220
	87	137,100	135,731	1,369
	76	137,800	137,154	646
	84	137,800	139,919	-2,119
	91	140,200	142,820	-2,620
	118	140,800	143,783	-2,983
	88	142,600	145,824	-3,224
	83	143,300	148,787	-5,487
	113	145,500	155,524	-10,024
	63	146,300	140,076	6,224
	80	147,300	148,275	-975
	92	147,400	150,371	-2,971
	56	151,300	157,297	-5,997
	75	151,800	149,120	2,680
	80	153,800	134,735	19,065
	79	155,900	135,771	20,129

76	156,100	157,154	-1,054
75	159,500	159,120	380
132	160,200	171,359	-11,159
84	161,400	159,919	1,481
83	163,100	148,787	14,313
84	164,200	159,919	4,281
83	165,300	148,787	16,513
94	166,200	169,383	-3,183
72	167,300	167,121	179
79	172,500	175,771	-3,271
74	176,700	172,379	4,321
130	177,500	182,996	-5,496
74	177,900	167,356	10,544
77	178,100	189,331	-11,231
75	178,500	179,120	-620
75	178,700	179,120	-420
127	179,000	178,701	299
85	180,200	181,056	-856
76	181,200	187,154	-5,954
56	182,200	187,297	-5,097
144	182,800	196,557	-13,757
87	182,900	175,731	7,169
82	184,000	182,777	1,223
73	185,200	186,055	-855
149	187,700	186,105	1,595
76	188,200	197,154	-8,954
145	189,200	199,588	-10,388
70	189,600	183,464	6,136
137	191,800	192,728	-928
85	197,200	185,056	12,144
85	197,500	181,056	16,444
110	202,900	203,959	-1,059
94	203,900	209,383	-5,483
83	209,600	208,787	813
78	211,400	214,855	-3,455
205	213,200	227,646	-14,446
119	215,600	204,347	11,253
132	220,300	221,359	-1,059
98	222,700	214,812	7,888
138	225,000	223,590	1,410
167	226,000	237,484	-11,484
104	226,800	224,636	2,164
82	227,000	222,777	4,223
132	228,500	222,359	6,141
160	230,100	238,186	-8,086
123	234,600	232,540	2,060
144	236,300	226,557	9,743
158	237,600	209,039	28,561
114	244,900	240,442	4,458
172	252,700	265,746	-13,046
161	255,100	251,862	3,238
160	255,400	248,186	7,214
146	255,800	266,824	-11,024
140	261,400	279,267	-17,867
146	261,700	266,824	-5,124
128	263,100	268,973	-5,873
142	275,100	266,726	8,374
146	276,900	266,824	10,076
187	288,600	273,886	14,714
170	309,800	308,620	1,180
180	319,900	304,742	15,158

	76	362,700	367,154	-4,454
	76	406,300	407,154	-854
	140	779,300	769,267	10,033
	140	825,900	819,267	6,633
mean	102.2241379	183,372	184,352	-980
standard deviation				8,805

The one variable duplex pricing: Regression results

Regression Statistics	
Multiple R	0.397024802
R Square	0.157628693
Adjusted R Square	0.150239471
Standard Error	9.545342466
Observations	116

ANOVA

	df	*SS*	*MS*	*F*
Regression	1	1943.656834	1943.656834	21.33224489
Residual	114	10386.94616	91.11356279	
Total	115	12330.60299		

	Coefficients	*Standard Error*	*t Stat*	*P-value*
Intercept	5.718222622	2.763491593	2.069202105	0.040787685
AREA_HSE	0.118264731	0.025605718	4.618684324	1.02162E-05

Observation	*Predicted*	*Residuals*
1	24.75884425	0.491155746
2	17.42643095	-4.626430955
3	15.77072473	3.529275274
4	14.46981269	2.530187311
5	15.41593053	-2.165930534
6	16.00725419	-3.207254187
7	19.79172557	0.908274433
8	12.34104754	5.258952463
9	22.51181437	4.488185628
10	14.94287161	-10.74287161
11	15.77072473	-3.120724726
12	31.85472809	-26.05472809
13	14.46981269	2.330187311
14	14.70634215	0.61865785
15	22.74834383	0.451656167
16	19.31866664	-10.81866664
17	21.32916707	-5.529167066
18	15.77072473	3.329275274
19	15.41593053	-4.415930534
20	13.99675377	-7.296753766
21	22.98487329	2.515126705
22	19.08213718	-4.582137184
23	15.77072473	1.479275274
24	16.59857784	-2.09857784
25	26.05975629	-1.259756291
26	25.46843264	-2.968432638
27	15.17940107	-2.879401073
28	15.41593053	2.984069466
29	19.67346084	-6.473460837
30	15.06113634	-3.911136342
31	22.27528491	-8.825284911
32	27.83372725	0.166272749
33	15.65246	-0.252459995
34	15.41593053	-9.415930534
35	17.30816622	4.191833776
36	16.48031311	-11.28031311

37	14.23328323	1.866716773
38	22.74834383	-4.748343833
39	14.35154796	-2.951547958
40	12.34104754	2.158952463
41	14.35154796	3.848452042
42	14.35154796	-6.051547958
43	29.9624924	-8.962492402
44	22.27528491	3.724715089
45	15.06113634	-0.361136342
46	15.53419526	-2.134195264
47	15.53419526	0.465804736
48	15.17940107	-2.679401073
49	15.65246	-2.952459995
50	15.53419526	-0.134195264
51	16.12551892	-1.925518918
52	22.98487329	3.015126705
53	24.64057952	-2.440579524
54	24.40405006	-1.404050062
55	16.8351073	3.164892699
56	16.71684257	-5.916842571
57	15.17940107	-2.879401073
58	13.99675377	4.503246234
59	14.70634215	-4.22634215
60	15.41593053	-6.925930534
61	14.70634215	-1.22634215
62	14.70634215	2.79365785
63	21.92049072	-3.670490719
64	17.30816622	-4.508166224
65	18.72734299	1.272657008
66	19.20040191	4.799598086
67	15.88898946	-3.888989456
68	15.65246	-2.232459995
69	14.70634215	-2.30634215
70	14.82460688	2.675393119
71	14.1150185	-3.615018497
72	15.17940107	-0.679401073
73	22.86660856	-4.716608564
74	15.41593053	-5.115930534
75	22.27528491	59.72471509
76	22.27528491	54.72471509
77	20.73784341	-3.237843413
78	15.65246	-0.152459995
79	14.58807742	2.311922581
80	14.58807742	1.211922581
81	15.06113634	1.238863658
82	27.00587414	4.494125864
83	15.53419526	-4.534195264
84	14.94287161	5.557128389
85	14.94287161	-8.742871611
86	14.58807742	-1.388077419
87	14.58807742	2.911922581
88	16.8351073	-0.635107301
89	15.53419526	4.465804736
90	18.01775461	3.682245392
91	20.26478449	2.33521551
92	23.33966749	-5.339667487
93	15.41593053	-4.215930534
94	12.34104754	-6.341047537
95	16.36204838	-4.062048379
96	21.32916707	1.470832934
97	22.98487329	3.815126705

98	22.03875545	-0.23875545
99	24.64057952	0.859420476
100	15.17940107	-2.579401073
101	13.16890065	1.031099348
102	21.80222599	-9.250225988
103	14.58807742	0.561922581
104	25.82322683	4.92677317
105	16.48031311	-2.98031311
106	15.17940107	-0.979401073
107	12.22278281	-4.272782807
108	20.85610814	5.143891857
109	21.0926376	-4.092637604
110	18.13601934	-5.936019338
111	16.00725419	1.892745813
112	21.32916707	0.670832934
113	14.70634215	3.79365785
114	15.41593053	6.584069466
115	14.70634215	25.79365785
116	14.70634215	20.79365785
		9.503750362

Banshaf, W., P. Nordin, et al. (1998). Genetic Programming, an introduction,. San Francisco, CA, Morgan Kauffman Publishers inc.

Bauer, J., R.J and F. G. FitzGerald (2000). "Using genetic programming to design a generalized trading system." Managerial Finance **26**(6): 1.

Bhattacharyya, S. (2003). "Evolutionary computation for database marketing." Journal of Database Marketing **10**(4): 343.

Birchenhall, C. R. (1995). "Genetic algorithms, classifier systems and genetic programming and their use in models of adaptive behaviour and learning." The Economic Journal **105**(430): 788.

Bodnovich, T. (1995). Genetic Algorithms in Business and Their Supportive Role in Decision Making, College of Business Administration Kent State University.

Carbone, R. and R. L. Longini (1977). "A Feedback Model for Automated Real Estate Assessment." Management Science **24**(3): 241-248.

Chen, S. (2001). "John Holland's Legacy in Economics: Artificial Adaptive Economic Agents -- From 1986 to the Present in Retrospect." Computing in Economics and Finance (April).

Chen, S.-H., Ed. (1996). Genetic programming, predictability, and stock market efficiency. Oxford, Pergamon Press.

Chen, S.-H. (2002). Genetic Algorithms and genetic programming in computational finance. Dordrecht, Kluwer Academic Publishers.

Chen, S. H. and C. H. Yeh (2000). "Simulating economic transition processes by genetic programming." Annals of Operations Research **97**(1): 265.

Chen, S. H. and C. H. Yeh (2001). "Evolving traders and the business school with genetic programming: A new architecture of the agent-based artificial stock market." Journal of Economic Dynamics & Control **25**(3,4): 363.

Chen, S. H. and C. H. Yeh (2002). "On the emergent properties of artificial stock markets: The efficient market hypothesis and the rational expectations hypothesis." Journal of Economic Behavior & Organization **49**(2): 217.

Chen, S.-H. and C.-C. Ni, Eds. (1997). Evolutionary artificial neural networks and genetic programming: a comparative study based on financial data. Vienna, Springer Verlag.

Chen, S.-H. and C.-H. Yeh (1997). "Toward a computable approach to the efficient market hypothesis: an application of genetic programming." Journal of Economic Dynamics and Control **21**: 1043-1063.

Chughtai, M. (1995). Determining Economic Equilibria using Genetic Algorithms. London, Imperial College.

Colin, F., M. John Paul, et al. (2005). "Risk adjusted returns from technical trading: a genetic programming approach." Applied Financial Economics **15**(15): 1073.

Darrell, W. (2001). "An overview of evolutionary algorithms: Practical issues and common pitfalls." Information and Software Technology **43**(14): 817.

Dawkins, R. (2006). God Delusion.

Do, Q. and G. Grudnitski (1992). "A neural network approach to residential property appraisal." Real Estate Appraiser(December): 38-45.

Dworman, G., S. O. Kimbrough, et al. (1995). "On automated discovery of models using genetic programming: Bargaining in a three-agent coalitions game." Journal of Management

Information Systems **12**(3): 97.

Fischer, D. and P. Lai Pin-Ying (2006). ANN application to Mass Appraisal. Pacific Rim Real Estate Society, Auckland.

Fischer, D. and P. Lai Pi-Ying (2007). Land price modeling with genetic algorithms and artificial neural network procedures. PRRES. Kuala Lumpur.

Fyfe, C., P. Marney J., et al. (1999). "Technical analysis versus market efficiency - A genetic programming approach." Applied Financial Economics **9**(2): 183.

Hansen, J., P. B. Lowry, et al. (2007). "Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection." Decision Support Systems **43**(4): 1362.

Holland, J. (1960). Iterative circuit computers Western Join Computer Conference.

Holland, J. (1975). Adaptation in Natural and Artificial Systems. Ann Arbor, University of Michigan Press.

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

Huimin, Z. (2007). "A multi-objective genetic programming approach to developing Pareto optimal decision trees." Decision Support Systems **43**(3): 809.

Jain, A. K., J. Mao, et al. (1996). "Artificial neural networks: a tutorial." Computer Publication **29**(3).

Kaboudan, M. A. (1999). "A measure of time series' predictability using genetic programming applied to stock returns." Journal of Forecasting **18**(5): 345.

Kaboudan, M. A. (2000). "Genetic Programming Prediction of Stock Prices." Computational Economics **16**(3): 207.

Koza, J., F. H. Bennett III, et al. (1999). Genetic Programming III: Darwinian Invention and Problem Solving, Morgan Kaufmann.

Koza, J., M. Keane, et al. (2003). "Evolving inventions." Scientific American.

Koza, J. R. (1992). Genetic programming: On the Programming of Computers by Means of Natural Selection Cambridge, MIT Press.

Lensberg, T. (1999). "Investment behavior under Knightian uncertainty - An evolutionary approach." Journal of Economic Dynamics & Control **23**(9,10): 1587.

Lensberg, T., A. Eilifsen, et al. (2006). "Bankruptcy theory development and classification via genetic programming." European Journal of Operational Research **169**(2): 677.

Lewis, O. M., J. A. Ware, et al. (2001). "Identification of Residential Property Sub-Markets using Evolutionary and Neural Computing Techniques." Neural Computing & Applications **10**(2): 108-119.

Marcos, Á.-D. and Á. Alberto (2005). "Genetic multi-model composite forecast for non-linear prediction of exchange rates." Empirical Economics **30**(3): 643.

Matthew, C. R. (2005). "Technical analysis and genetic programming: Constructing and testing a commodity portfolio." The Journal of Futures Markets **25**(7): 643.

Mc Greal, S., A. Adair, et al. (1998). "Neural Networks and the prediction of residential values." Journal of Property Valuation and Investment.

McCloskey, D. N. (1992). If You're So Smart: The Narrative of Economic Expertise. Chicago, University of Chicago Press, 1992,.

McCluskey, W. and S. Anand (1999). "The application of intelligent hybrid techniques for the mass appraisal of residential properties." Journal of Property Investment & Finance **17**(3): 218-239.

McKee, T. E. and T. Lensberg (2002). "Genetic programming and rough sets: A hybrid approach to bankruptcy classification." European Journal of Operational Research **138**(2): 436.

Neely, C. (2003). "Risk-adjusted, ex ante, optimal technical trading rules in equity markets." International Review of Economics & Finance **12**(1): 69.

Neely, C., P. Weller, et al. (1996). "Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach." The Journal of Financial and Quantitative Analysis **32**(4): 405-426.

Neely, C. and P. A. Weller (2002). "Predicting exchange rate volatility: Genetic programming versus GARCH and RiskMetrics." Review - Federal Reserve Bank of St. Louis **84**(3): 43.

Nguyen, N. and A. Cripps (2001). "Predicting Housing Value: A Comparison of Multiple Regression Analysis and Artificial Neural Networks." Journal of Real Estate Research **22**(3): 313-336.

Oakley, H. (1996). Genetic programming, the reflection of chaos, and the bootstrap: toward a useful test for chaos., MIT Press.

Rossini, P. (1997). Application of Artificial Neural Network to the Valuation of Residential Properties. Pacific Rim Real Estate Society, Palmerston, New Zealand.

Salcedo-Sanz, S., F.-V. JL, et al. (2005). "Genetic programming for the prediction of insolvency in non-life insurance companies." Computers & Operations Research **32**(4): 749.

Schmerken, I. (2004). "Survival of the Fittest Trading Models." Wall Street & Technology: 32.

Trigueros, J. R. (2000). Essays on the application of evolutionary computing to accounting and finance. United States -- Louisiana, Tulane University.

Wagner, N. and J. Brauer (2007). "Using genetic dynamic programming to forecast the the United States Gross Domestic product with military expenditure as an explanatory variable.E." Defence and Peace Economics **18**(5): 451.

Wang, J. (2000). "Trading and hedging in S&P 500 spot and futures markets using genetic programming." The Journal of Futures Markets **20**(10): 911.

Whitley, L. D. and M. Vose (1995). Foundatiions of Genetic Algorithms, Morgan Kaufmann Publishers, Inc.

Wilson, I. D., S. D. Paris, et al. (2002). "Residential property price time series forecasting with neural networks." Knowledge-Based Systems **15**(5-6): 335-341.

Winter, G., J.Periaux & M.Galan, Ed. (1995). Genetic Algorithms in Engineering and Computer Science, John Wiley and sons.

Wonga, B. K. and Y. Selvib (1998). "Neural network applications in finance: A review and analysis of literature (1990–1996)." Information and Management **34**(3): 129-139.

Worzala, E., M. Lenk, et al. (1995). "An Exploration of Neural Networks and Its Application to Real Estate Valuation." Journal of Real Estate Research **10**(2): 185-202.