

MACHINE DESIGNED TRADING STRATEGIES

TSL Version 1.3 Review

Michael L. Barna
President
Trading System Lab

Please place phone on Mute if not speaking

3/26/2012



Mike Barna, CTA

- Founder and President, Trading System Lab
- BS Mathematics, Arizona State University
- MS Astronautical and Aeronautical Engineering, Stanford University
- SVP Regency Stocks and Commodities Fund, LP, LLC (QEP Hedge Fund)
- R-MESA*, BIGBLUE*, MESA BONDS, MESA NOTES
- Open Platform Developer: TradeStation and others
- 30+ years trading experience, 30+ years programming, 20+ years trading system design, 7 years money management experience
- Former Defense Industry Rocket-Ramjet and Guidance Systems Engineer
- Star Wars Research and Development Management Engineer
- 2 CFTC Commodity Licenses, 10 FAA pilot certificates, Calif. Real Estate License

*Multiple awards received

WHO IS TSL?

- Trading System Lab-US Based trading research and development company with a small team of international and domestic programmers, third party developers and testers. Founded by former Hedge Fund Manager.
- Register Machine Learning, Inc.-US Based company with a team of international and domestic machine learning scientists, IP attorneys, statisticians and programmers. Involved in government contracts.

NEW WORLD = NEW PROFITS*

- Machines are primary on Wall Street
- Machine designed strategies are outperforming human designs*
- The Public realizes trading is critical
- Public Strategy trading at Low Frequencies began 10 years ago
- Public High Frequency Trading will gain popularity
- HF Machine Trading is the most profitable category on Wall Street but done by only a few firms
- Emerging countries are an untapped resource
- Customers demand more analysis capabilities in finance and trading

*Now, average investors control powerful machine designed trading algorithms

SIZE OF THE WORLDS MARKETS

- Equities: \$36.6T as of October 2008
- Derivatives: \$791T (11x the entire world economy)
- In 2009, High Frequency Algorithmic trading accounts for 73% of all US equity trading volume, but is conducted by only 2% of the firms
- The High Frequency Hedge Fund category is now the **most profitable** on Wall Street
- Emerging Markets are growing

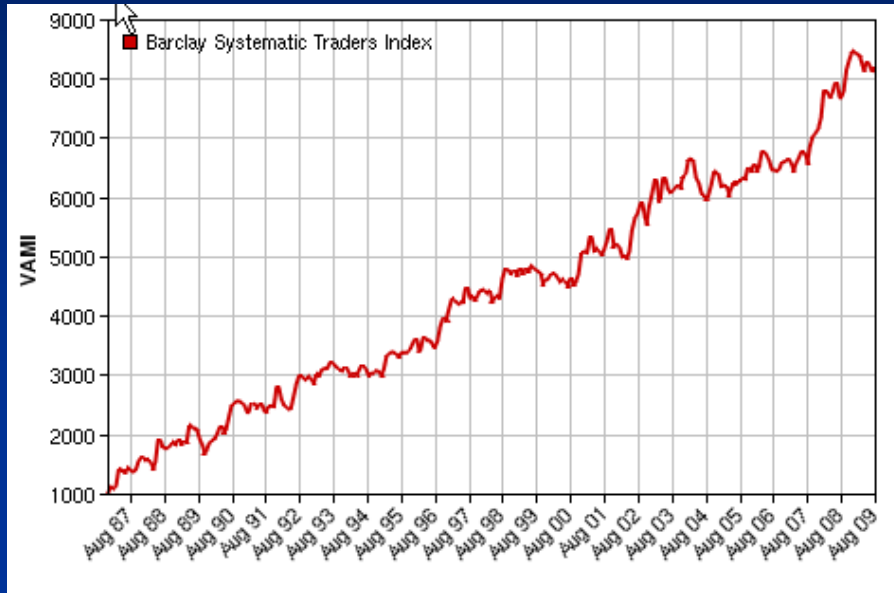
WORLD STOCK MARKETS

Rank 	Economy 	Stock Exchange 	Market Capitalization (USD Billions) 	Trade Value (USD Billions) 
1	 United States  Europe	NYSE Euronext	15,970	19,813
2	 United States  Europe	NASDAQ OMX	4,931	13,439
3	 Japan	Tokyo Stock Exchange	3,827	3,787
4	 United Kingdom	London Stock Exchange	3,613	2,741
5	 China	Shanghai Stock Exchange	2,717	4,496
6	 Hong Kong	Hong Kong Stock Exchange	2,711	1,496
7	 Canada	Toronto Stock Exchange	2,170	1,368
8	 India	Bombay Stock Exchange	1,631	258
9	 India	National Stock Exchange of India	1,596	801
10	 Brazil	BM&F Bovespa	1,545	868
11	 Australia	Australian Securities Exchange	1,454	1,062
12	 Germany	Deutsche Börse	1,429	1,628
13	 China	Shenzhen Stock Exchange	1,311	3,572
14	 Switzerland	SIX Swiss Exchange	1,229	788
15	 Spain	BME Spanish Exchanges	1,171	1,360
16	 South Korea	Korea Exchange	1,091	1,607
17	 Russia	MICEX	949	408
18	 South Africa	JSE Limited	925	340

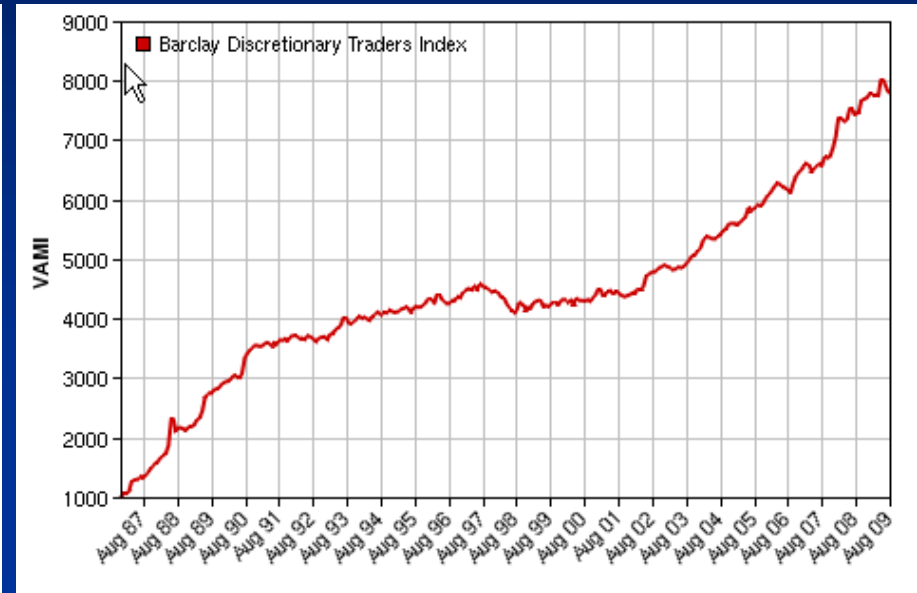
SYSTEMATIC VERSES DISCRETIONARY

VAMI

1987-2009



448 programs, \$149B



155 programs, \$15B

Source: BarclayHedge

ARE WE COMFORTABLE WITH MACHINES YET?

- Computers land airliners when pilots are not allowed to land them (Cat III)
- Control/Monitor Life Support in critical and non critical situations
- Engineering, Pharmacology, Research
- National Defense



SO WHY ARE WE STILL MANUALLY DESIGNING TRADING SYSTEMS?

- Software crisis-s/w costs are >10 times hardware costs
- 99% of CPU cycles are not used
- Lack of Industry Focus
- Past AI in Finance Failures
- Lack of confidence or understanding: fear?
- Note: Very few Futures Truth Trading Systems using AI

<http://www.tradingsystemlab.com/files/CISC%20Architectures.pdf>

TSL Clients and Markets

- Major Wall Street Investment bank >\$100M
- Small and mid size funds \$10-\$100M
- Individual traders
- Numerous International Traders

WHAT DID IT TAKE TO CREATE TSL?

- The work by two companies over many years
- Multiple U.S. patents
- Approx. One million lines of code
- World class scientists and programmers
- A lot of \$
- Many obstacles along the way

WHAT IS TSL?

TSL is a stochastic, evolutionary, multi-run, Trading Strategy Automatic Design engine employing a Register Based Automatic Induction of Machine Code with Genetic Programming algorithm.

An “Explosion of Ideas and Mathematics”



WHAT IS TSL?

- TSL is a platform for the automatic design of Trading Strategies
- Linear Automatic Induction of Machine Code with Genetic Programming (LAIMGp) (algorithms are nonlinear)
- Trading Strategies are designed for you-NO PROGRAMMING!
- Code is written for you in different languages
- Strategies are tested OOS during design
- TSL is fast, very fast-can't do this work with normal software
- Patented and Trademarked
- Single Market Systems and Daytrading
- Pairs
- Portfolios
- Options



WHY DO FIRMS OR TRADERS BUY TSL?

- They cannot build this in-house
- Offers immediate alpha generating potential
- Accelerates implementation time
- Enhances research capabilities
- Improves their marketability

HOW GOOD ARE THE MACHINE DESIGNED STRATEGIES?

In 2007, TSL submitted several early “Machine Designed” Strategies to Futures Truth after running TSL for about 10 minutes per system. These strategies were held for over 18 months, then tested, compared and ranked against hundreds of submissions from worldwide “human” strategy designers. These systems have not been touched since and reporting on these “Machine Designed” Strategies continues.

The results?



HOW GOOD ARE TSL MACHINE DESIGNED STRATEGIES?”

This is the first time that a “Machine” out-designed a human
Is there any other algorithm that can make this claim?

Top 10 Systems Since Their Release Date

Issue #2 2011 - published in July 2011

Systems included in this table must have been released for at least 18 months. Results based on performance through April 30, 2011.

Return is based on three times the required margin.

Rank	System Name	Annual % Return
1.	TSL_CEL_NG_1.1	183.9%
2.	Natural Gas Offense	164.3%
3.	Trend Finder - Tiger	162.9%
4.	Trend Finder - Lion 2	120.2%
5.	Natural Gas Trader - GA	115.5%
6.	Auto Core Duo	96.8%
7.	CTB3 SW	90.3%
8.	Dual Thrust	87.7%
9.	Propero ES	87.4%
10.	Catscan IV	86.1%



TSL TOP S&P SYSTEMS DESIGNED AND FROZEN IN 2007

During the worst stock market environment of our lifetimes

Top 10 S&P Systems

Issue #2 2011 - published in July 2011

Ranking based on performance since their release to us. Some systems have been out for a short period of time. Results based on performance through April 30, 2011. Return is based on three times the required margin.

Rank	System Name	Annual % Return
1.	TSL_SP_1.0Z	79.5%
2.	TSL_CEL_SP1	75.5%
3.	Big Blue 2	67.6%
4.	Impetus SP	54.7%
5.	Keystone	50.7%
6.	STC S&P DayTrade	41.0%
7.	%C DayBreaker	39.1%
8.	Cyclone	38.0%
9.	R-Breaker	37.2%
10.	Strategic 500	36.0%

HOW GOOD ARE TSL MACHINE DESIGNED STRATEGIES?”

This is the first time that a “Machine” out-designed a human
Is there any other algorithm that can make this claim?

Top 10 Systems Since Their Release Date

Issue #4 2011 - published in January 2012

Systems included in this table must have been released for at least 18 months. Results based on performance through October 31, 2011.
Return is based on three times the required margin.

Rank	System Name	Annual % Return
1.	TSL_CEL_NG_1.1	276.55%
2.	Natural Gas Offense	268.9%
3.	Delphi II EMD	166.5%
4.	TrendFinder - Tiger	145.5%
5.	Trend Finder - Lion 2	131.7%
6.	TSL_SP_1.0Z	118.3%
7.	Jen Jamer	100.1%
8.	Propero ES	98.1%
9.	TSL_CEL_SP1	94.6%
10.	Delphi II ER Aggressive	88.8%

TSL TOP S&P SYSTEMS DESIGNED AND FROZEN IN 2007

During the worst stock market environment of our lifetimes
January 2012 FT Report

Top 10 S&P Systems

Issue #4 2011 - published in January 2012

Rank	System Name	Annual % Return
1.	TSL-SP_1.0Z	118.3%
2.	TSL-CEL_SP1	94.6%
3.	Impetus	88.4%
4.	Keystone	81.3%
5.	FT Classic	79.4%
6.	STC S&P DayTrade	60.4%
7.	R-Breaker	55.4%
8.	Strategic 500	54.6%
9.	%C DayBreaker	53.8%
10.	Cyclone	52.0%

DISTRIBUTION ANALYSIS

WILCOXON TEST

Mann-Whitney

t test		
1	Table Analyzed	Paired t test data
2	Column A	AVGRET/SDRET
3	vs	vs
4	Column B	AVGRET/SDRET
5		
6	Wilcoxon matched-pairs signed rank test	
7	P value	< 0.0001
8	Exact or approximate P value?	Gaussian Approximation
9	P value summary	****
10	Are medians signif. different? (P < 0.05)	Yes
11	One- or two-tailed P value?	Two-tailed
12	Sum of positive, negative ranks	9.596e+007 , -1.564e+008
13	Sum of signed ranks (W)	-6.043e+007
14		
15	How effective was the pairing?	
16	rs (Spearman, Approximation)	0.0370
17	P Value (one tailed)	< 0.0001
18	P value summary	****
19	Was the pairing significantly effective?	Yes

59,055 systems. Serial duplicates eliminated:

22,465 systems

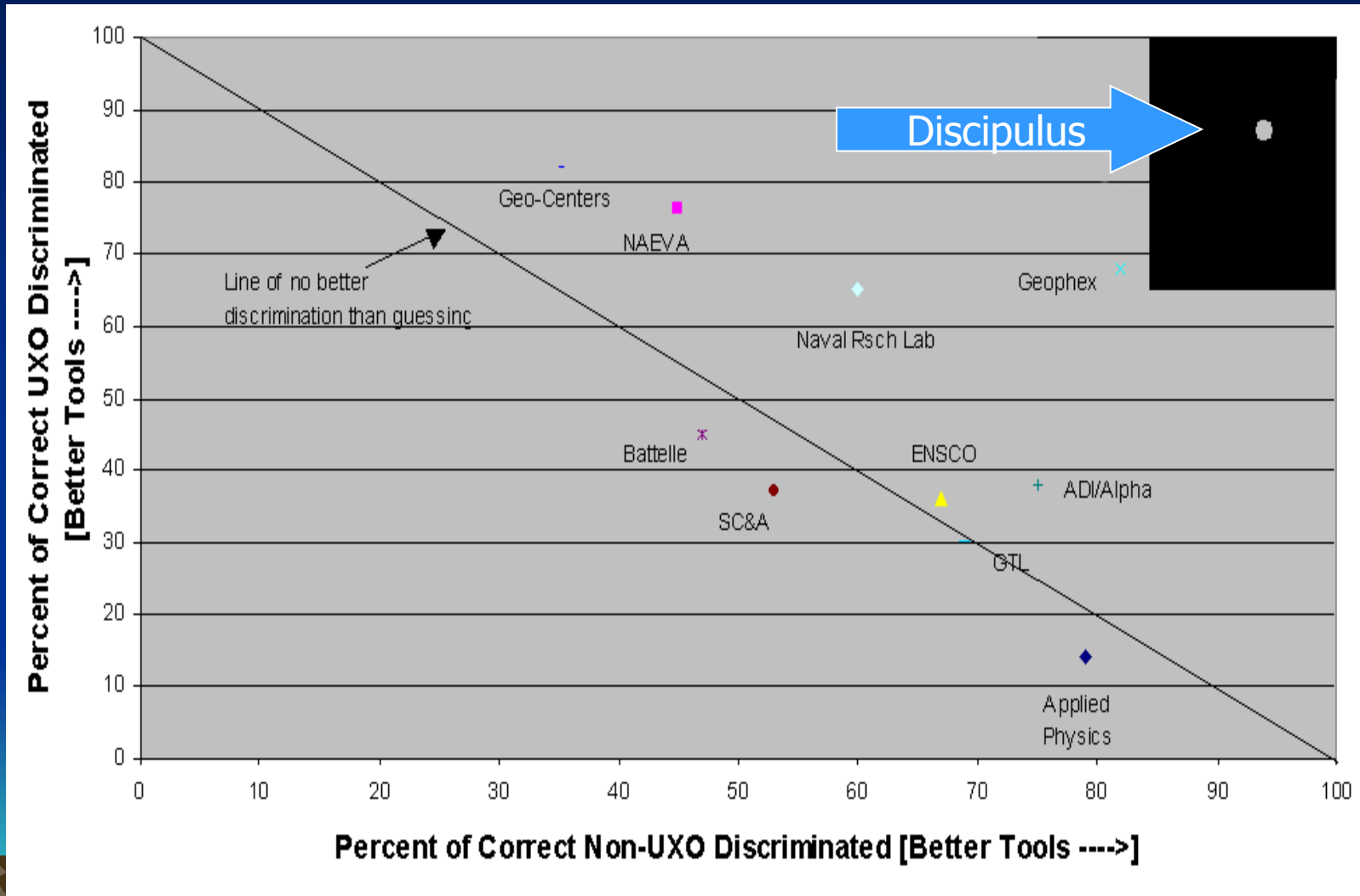
SPY Long Only.

Distribution assumed Non-Gaussian.

WW Daniel, [Applied Nonparametric Statistics](#)

RML'S DISCIPULUS OUTPERFORMED EXISTING PUBLISHED RESULTS

UXO Discrimination Tests



TOP COMPLAINTS OF TRADING STRATEGY USERS

- “I want systems but I can’t write code”
- “Easy Language is not easy”
- “I need someone to write a strategy”
- “I wish strategies were more available”
- “I can’t HFT with these high commissions”

HOW DO WE REDUCE THESE COMPLAINTS?

- Provide “PROVEN” Strategy auto-design tools
- Provide for HFT
- Provide lower commissions
- Provide for International trading
- Don't use 30 year old technology that is not third party ranked



HOW DO WE REDUCE THESE COMPLAINTS?

- Work hard on robustness
- Focus on customer support and training
- Appeal to the non programmer
- Appeal to the advanced technician

TSL GP MARKET COMPARISON

TSL has no serious competition

	TSL-LAIMGP	OTHER (GA/GP)
SPEED	900 MISBS**	Way Slower
# Inputs*	56	5-?
# Functions	34	5-10
# Entries	18	1-5
# Exits	4	1-3
# Stops	3	1-3
Automatic OOS?	YES	MAYBE
Daytrade?	YES	MAYBE
Pairs?	YES	NO
Portfolio/MM?	YES	MAYBE
Options?	YES	NO
# Preprocessors	5 (custom)	1
H, Non H Cross?	YES	NO
Rated top performing?	YES-#1 by FT	NO
Patented Algorithm	YES	NO
Price	More Expensive	Less Expensive

*Speed may be proportional to inputs

**Million Input-System-Bars per Second, HT TSL

3/26/2012

HOW DO WE REDUCE THESE COMPLAINTS?

1. Robustness
2. Uniqueness
3. Transparency
4. Speed
5. Portability
6. Diversity

ROBUSTNESS

(Over Fit Avoidance)

- Blind OOS Testing (walk forward)
- Run Path Logs (path intelligence)
- Unbiased Terminal Set (directionless inputs)
- Multi-Run, Randomized Criteria (global optimum)
- Zero Point Origin (no predefined initial point)
- Parsimony Pressure (Occam's razor)



WHAT IS OUT OF SAMPLE (OOS) TESTING?

- Blind testing on data not used in design
- Hidden data
- Walk forward data

This data is untouched by the design engine. The only use that is made of it is that you may look at it as it is generated. Good OOS may be indicative that the design is proceeding down a robust path. Additional second blind testing may be accomplished as well.

Do not confuse Validation Data with OOS. Validation is used to populate the best programs report, and is considered “In Sample”.



SAY GOODBY TO MANUALLY DESIGNED SYSTEMS

- Develop or use existing theory
- Use TA books, indicators, patterns, etc.
- Hand code the system
- Test, Optimize, Test, Optimize (Curve Fit? Use a GA?)
- Try to select best parameters
- Forward walk (maybe?)
- Implement and hope for the best
- Time of work flow: days to months

SAY GOODBYE TO MANUALLY DESIGNED SYSTEMS

- Develop or use existing theory
- Use TA books, indicators, patterns, etc.
- Hand code the system
- Test, Optimize, Test, Optimize (Curve Fit? Use a GA?)
- Try to select best parameters
- Forward walk (maybe?)
- Implement and hope for the best
- Time of work flow: days to months



SAY HELLO TO MACHINE DESIGNED SYSTEMS

- Select Market
- Run Strategy generation algorithm
- Observe Out of Sample performance
- Decide if adequate
- Implement via auto-created Easy Language code
- Adjust, research, study, tweak, learn, etc.
- Time of work flow: minutes to hours (longer for daytrading systems)

THIS IS OUR TRADING SYSTEM WHEN WE BEGIN A RUN IN TSL

(null starting point)



THIS IS OUR TRADING SYSTEM CORE LOGIC CODE AFTER A RUN HAS COMPLETED MACHINE CODE TRANSLATED TO C, EL, JAVA, ETC.

```
long double f[8];  
long double tmp = 0;  
int cflag = 0;  
  
f[0]=f[1]=f[2]=f[3]=f[4]=f[5]=f[6]=f[7]=0;  
  
L0:      f[0]=v[25];  
L1:      f[0]+=v[43];  
L2:      f[0]=fabs(f[0]);  
L3:      f[0]=v[13];  
L4:      f[0]=v[49];  
L5:      f[0]=v[41];  
L6:      f[0]*=f[0];  
L7:      f[1]=f[0];  
L8:      f[0]+=v[22];  
L9:      tmp=f[1]; f[1]=f[0]; f[0]=tmp;  
L10:     cflag=(f[0] < f[2]);  
L11:     f[0]=v[39];  
  
if (!_finite(f[0])) f[0]=0;  
  
return f[0];
```

 Easy Language

What do I get with TSL?

- Unlimited number of systems*
- Unlimited number of time frames
- Unlimited number of markets
- Large selection of languages
- Fastest available speed
- Huge research capability

*The user builds the systems

BROKERAGE FIRMS WILL ATTRACT MORE TRADERS

- Perfect for beginners who want systems
- Makes Easy Language more available
- Intermediate level users gain advantage
- Advanced Users can do advanced work

TSL CRITICAL FEATURES

- >16 million sys-bars/sec (core i7 990x or better)
- Simultaneous Design/ Walk Forward during design time
- 56 inputs, full custom inputs possible
- 200,000 bar limit
- 5 preprocessed fact sets for each of 2 main setup options
- 34 instruction sets, fully configurable
- 40 Fitness Functions, includes multi fitness
- 25 Entry tactics, including multi entry tactics
- 7 exit tactics, including GP adaptive stops and exits
- Risk/Size, Constant dollars embedded and evolved
- Money Management
- Pairs, full hedge, partial hedge, Portfolios
- 17 Option Tactics and combinations (CRR-BTREE, Bjerk-Stens)
- Daytrading-custom ID preprocessing, DT entry/exit types

TSL EVOLVED ENTRY AND EXIT TACTICS

Single Market Directional Strategies (partial list)

1. This Bar's Close
2. This Bar's Close +/- Stop
3. Next Bar's Open
4. Next Bar's Open +/- Stop
5. Enter on Limit
6. Next Bar on Stop
7. Ent/Ex Next Bar on Limit-Non SAR
8. Ent on Close or Next Open +/- Stop
9. Ent on Stop or Close
10. Ent on Close or Next Open
11. Ent on Close, Next Open or Stop
12. Ent on a GPStop: use 62n data
13. Ent on Close +/- Limit
14. Next Bar's Open +/- Limit
15. Next Bar's Close (Mutual Funds)
16. Ent/Ex Next Bar on GP Stop non SAR
17. NxtOpn +/- Limit or ThisClose +/- Limit
18. Next Bar's Open +/- Stop Mod1
19. Enter on Stop or Limit
- 20-25 Not Shown

- Exit on Fixed \$ Stop Loss
- Exit on Fixed % Stop Loss
- Exit on GP Evolved Stop Loss
- Exit on Fixed \$ Limit Profit
- Exit on Fixed % Limit Profit
- Exit on Evolved Limit Profit
- Exit on N Bar

BIG PICTURE QUESTION:

How many CPU cycles does it take to compute the following?

$$X = Y + Z$$

Reference: Efficient Evolution of Machine Code for CISC Architectures using blocks and Homologous Crossover,
Peter Nordin, Wolfgang Banzhaf, Frank Francone

BIG PICTURE QUESTION:

How many CPU cycles does it take to compute the following?

$$X = Y + Z$$

High level languages: 20 Clock cycles

CGPS(LAIMGP): 1 Clock cycle

So, TSL should be at least 20 times
faster than higher level languages
“on a single thread”

Reference: Efficient Evolution of Machine Code for CISC Architectures using blocks and Homologous Crossover,
Peter Nordin, Wolfgang Banzhaf, Frank Francone

WHICH SYSTEM SHOULD YOU TRADE?

- Trader 1:
35% accurate
Average Win is 180% of Average Loss
- Trader 2:
90% accurate
Average Win is 10% of Average Loss



WHAT EQUATION PRODUCES ANY TRADING SYSTEM'S EXPECTATION?

$$EV = PW * AW - PL * AL$$

EV = expected value or average trade

PW = probability of a win

PL = probability of a loss

AW = amount won in winning trades

AL = amount lost in losing trades



Trading System Lab[®]

FOR OUR 2 TRADERS:

Trader 1: EV = -.02

Trader 2: EV = -.01

So, neither Trader has a positive expectation

Note:

A system that is 90% accurate can have a negative EV

A system that is 30% accurate can have a positive EV



Trading System Lab[®]

TRADING SYSTEMS DESIGNED IN 3 EASY STEPS

1. Preprocess (TradeStation EL)
2. Evolve (TSL)
3. Translate (TradeStation OMS/EMS)

TSL ACCEPTS 56 INPUTS

THEN REDUCES DOWN TO JUST A FEW

Example Boolean Indicators:

```
if h>=highest(h, dc)[1] and hhit=0 then begin
    hhit=1;
    lhit=0;
end;
if l<=lowest(l, dc)[1] and lhit=0 then begin
    hhit=0;
    lhit=1;
end;
```

Example Boolean Patterns:

```
cond[45] = c>=c[1];
cond[46] = c>=c[2];
cond[47] = o>=c[1];
cond[48] = o>=c[2];
```

Consider: Intermarket data, Fundamental data, etc.



Trading System Lab®

PREPROCESSOR (DNA)

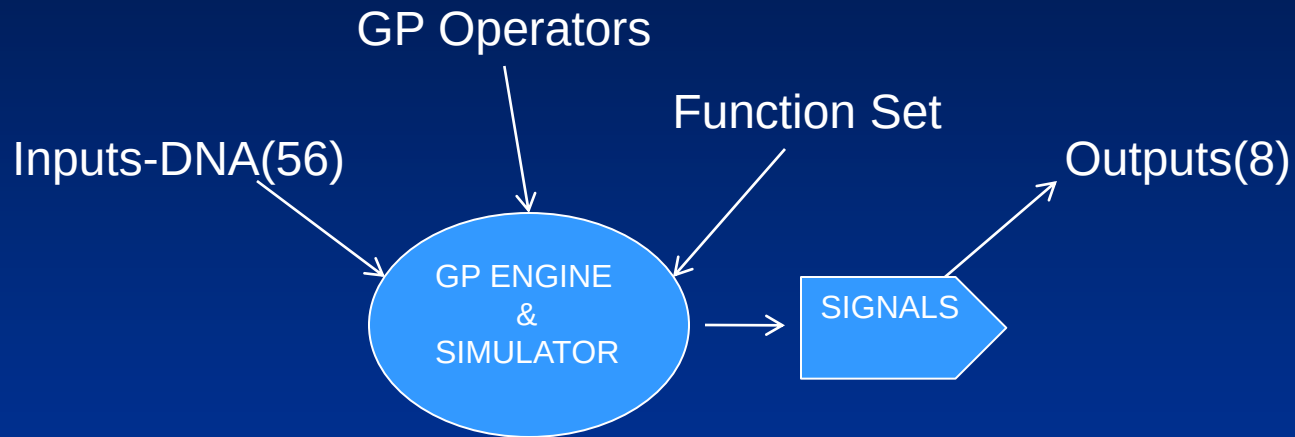
Numeric or Boolean

56 inputs + Date, O, H, L, C, Time

(Boolean data shown below)

[illegible]

TSL EVOLUTION



Outputs are used for basic and higher level learning:

1. Basic trading signals
2. Complex trading signals
3. Money management
4. Adaptive risk
5. Targets and stops
6. Optimal GP

GENETIC PROGRAMMING

- Based loosely on biological models of evolution and eucaryotic* sexual reproduction
- Simulates the path a biological species goes through as it evolves:
 - Starts off simple
 - Adapts to hostile environment
 - Strong Parents give birth to strong children
 - Random mutations may help

*Based on complex cells with membranes



GA TO GP CONTINUUM

GA
ADJUSTING
PRE DEFINED
STRATEGY
(SLOW)

GA
SELECTING STRATEGY
RULES
(SLOW)

GP-TREE
EVOLVES NEW
STRATEGY RULES
(SLOW)

Old Technology

New Technology

LAIMGP
EVOLVES NEW
RULES IN
MACHINE CODE
AND/OR
SELECTS
PARAMETERS
(FAST)

Genetic Programs verses Genetic Algorithms

- TSL's GP creates actual computer programs (Trading Systems). GA's simply optimize the existing parameter set which the user must supply
- GP's use variable length genomes and vary their size through non-homologous crossover. GA's use fixed length genomes.
Thus the Trading Systems grow and shrink in size during evolution
- GP's discover new mathematical relationships. GA's just optimize what is already present and discover nothing new
- The binary AIMGP is the fastest applicable algorithm

Summary of Evolutionary Algorithms

YEAR	INVENTOR	TECHNIQUE	INDIVIDUAL
1958	Freidberg	Learning Machine	Virtual Assembler
1959	Samuel	Mathematics	Polynomial
1965	Fogel, Owens and Walsh	Evolutionary Programming	Automation
1965	Rechenberg, Schwefel	Evolutionary Strategies	Real-Numbered Vector
1975	Holland	Genetic Algorithms	Fixed Size Bit String
1978	Holland and Reitmann	Genetic Classifier Systems	Rules
1980	Smith	Early Genetic Programming	Var-Size Bit String
1985	Cramer	Early Genetic Programming	Tree
1986	Hicklin	Early Genetic Programming	LISP
1987	Fujiki and Dickinson	Early Genetic Programming	LISP
1987	Dickmanns, Schmidhuber and Winklhofer	Early Genetic Programming	Assembler
1992	Koza	Genetic Programming	Tree

ALGORITHM COMPARISON

	FAST?	ACCURATE?	WRITES CODE?	CREATE NEW MODELS?
Neural Networks	NO	YES	NO	NO
Case Based Reasoning	YES	YES	NO	NO
Classification	YES	YES	NO	NO
Regression	MAYBE	MAYBE	NO	NO
Expert Systems	MAYBE	MAYBE	NO	NO
Intelligent Agents	NO	YES	NO	NO
Genetic Algorithms	YES	YES	NO	NO
Genetic Programming	YES	YES	YES	YES

Genetic Programs are fast, accurate, produce something new and write code

THE TROUBLE WITH TREE BASED GP

- Tendency to over-fit data
- Slow-implemented as high level code
- Memory is not usually held Globally
- Are there any Tree GP top systems?



TSL GP LEARNING

- Supervised Learning
- Population is initialized
- Tournament is run within population
- Mutation causes random changes in winners
- Crossover exchanges DNA between winners
- Reproduction is applied on remainder
- Demes enhance genetic diversity
- Parsimony Pressure favor simpler solutions
- If n GWI occur then run restarts
- After x runs or user termination, all runs stop

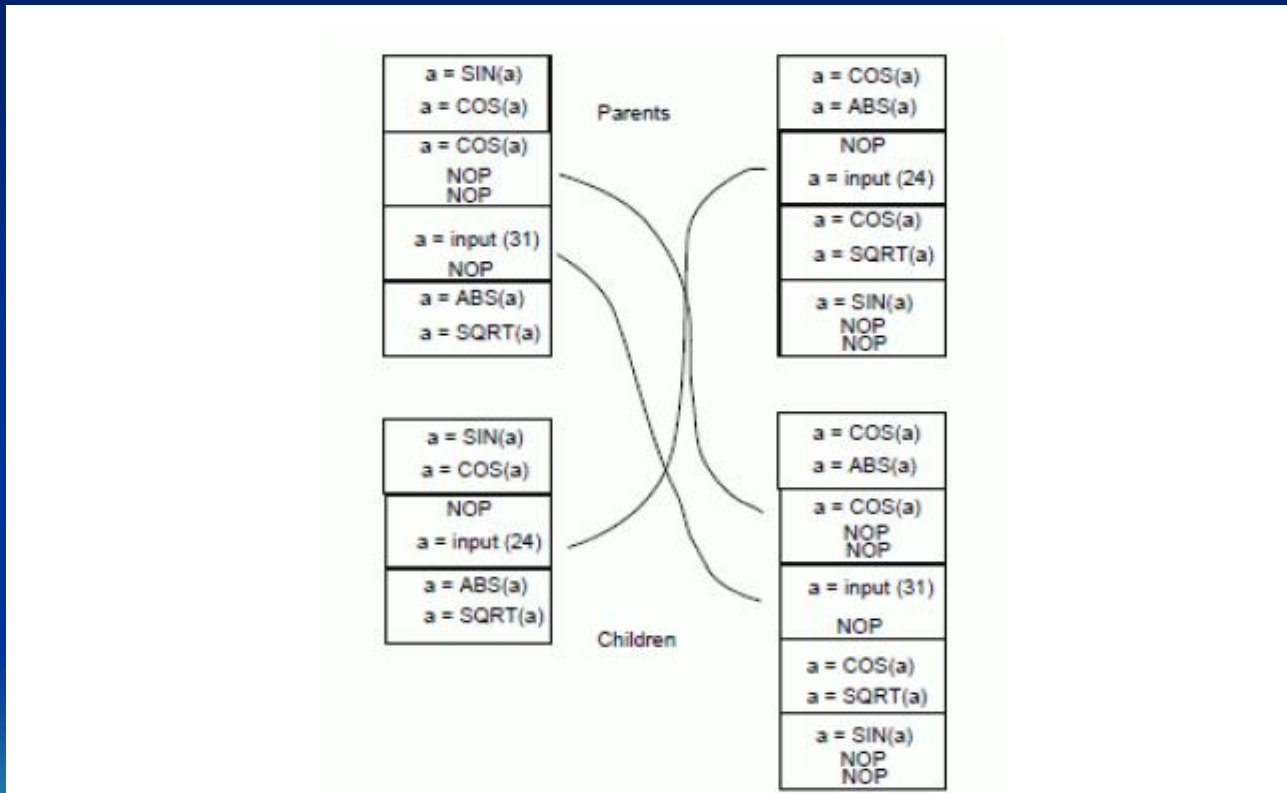


THE OPERATORS

- Crossover: Child shares parents genes
- Reproduction: Parent allowed to birth
- Mutation: Child genes altered
- Demes: Species Interbreed

LAIMGP REPRODUCTIVE CROSSOVER

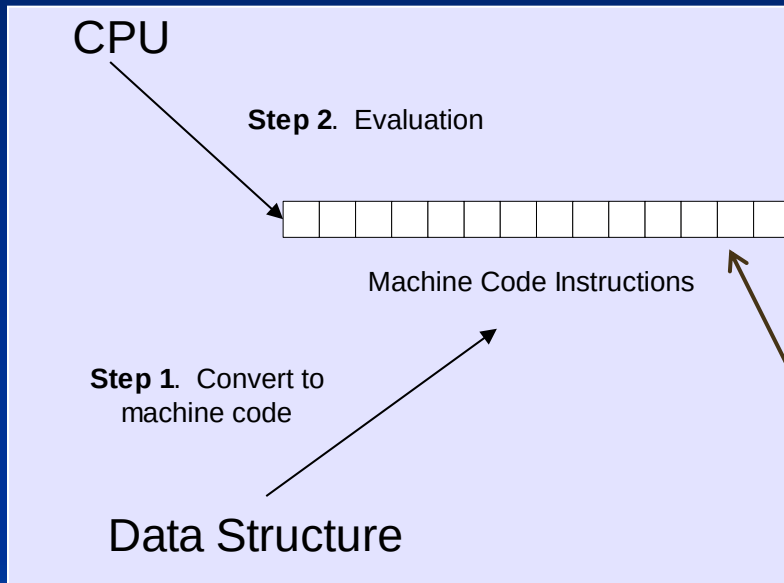
Homologous and Non-Homologous



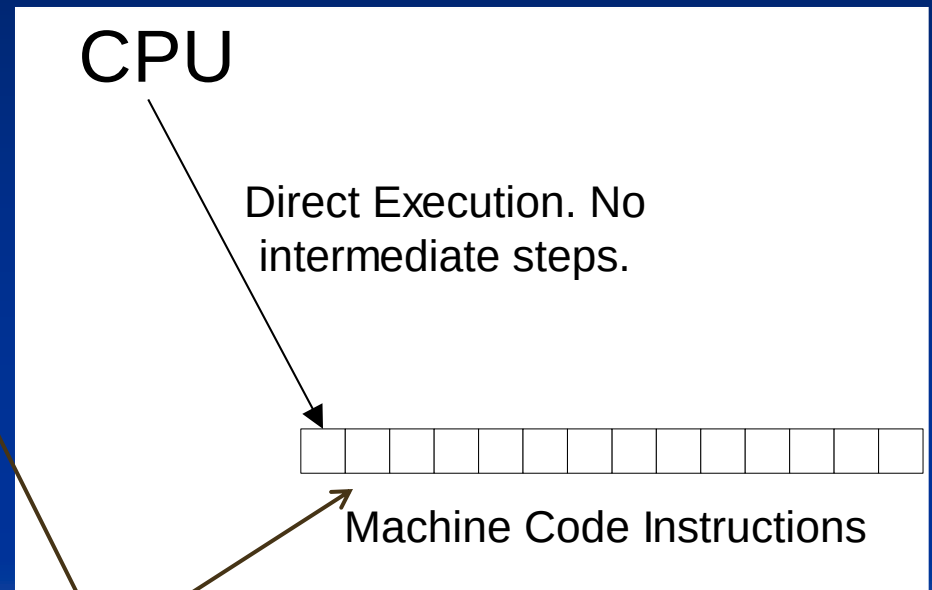
Reference: Frank D. Francone Licensiate Thesis (2009)

INDUCTION OF MACHINE CODE

Past: Slow



Present: Fast



TRADING SYSTEM CODE

GP's FIND CLOSED FORM SOLUTIONS, FIT FUNCTIONS AND PRODUCE CLASSIFICATIONS

Inputs

2 4

3 9

4 16

5 25

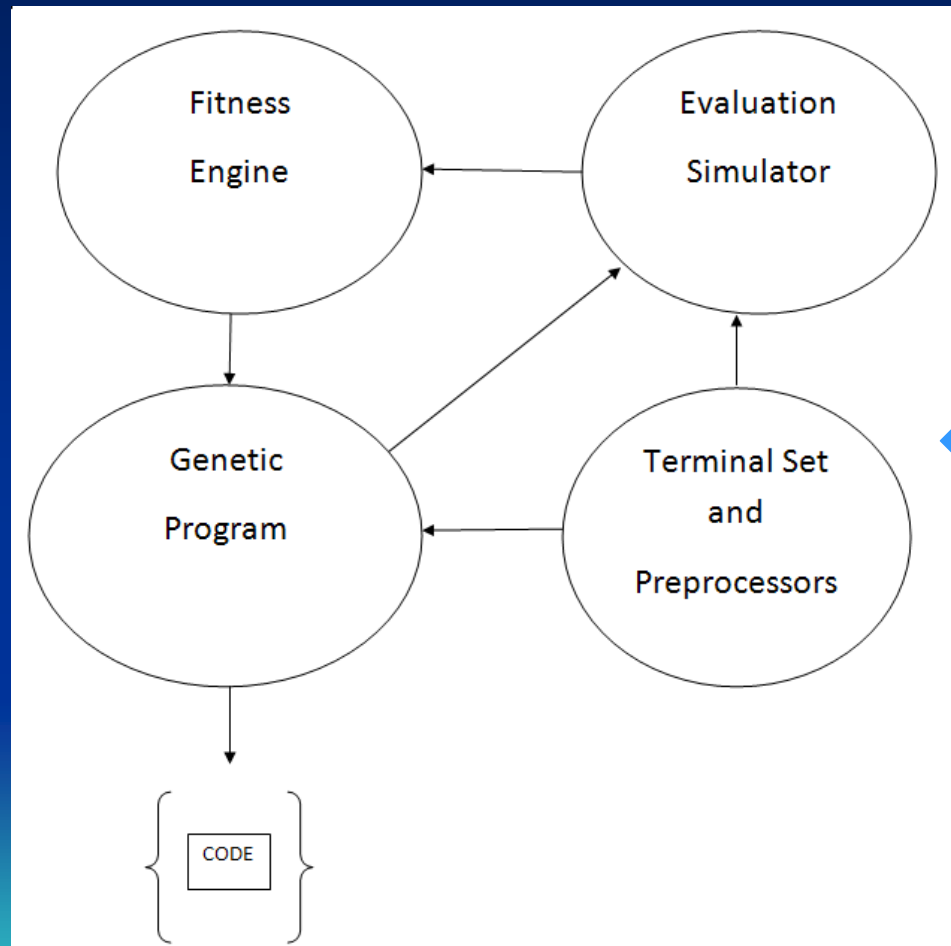
Output

$$y = x^2$$

← Closed Form

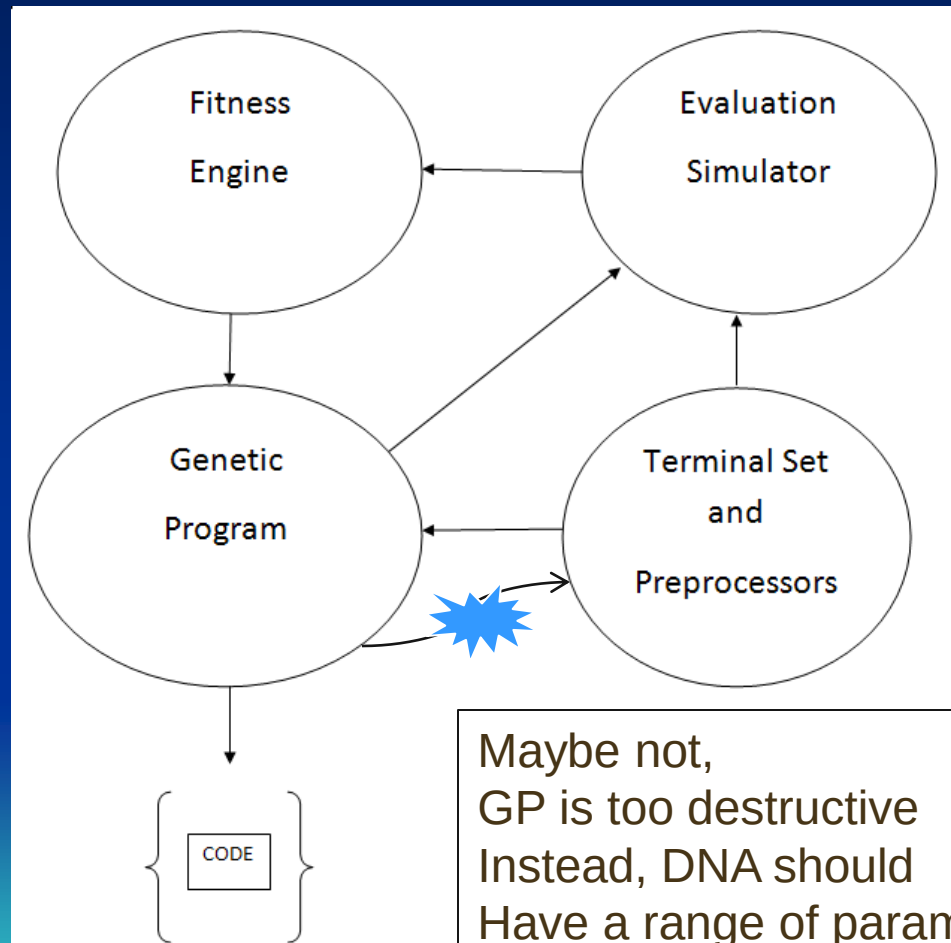
We target a Fitness Function like SR

TSL DATA FLOW



Should this Allow
DNA parameters to
Adapt??

DO WE WANT THE GP TO DETERMINE DNA PARAMETERS?



Maybe not,
GP is too destructive
Instead, DNA should
Have a range of parameters

INSTRUCTION FUNCTION SETS

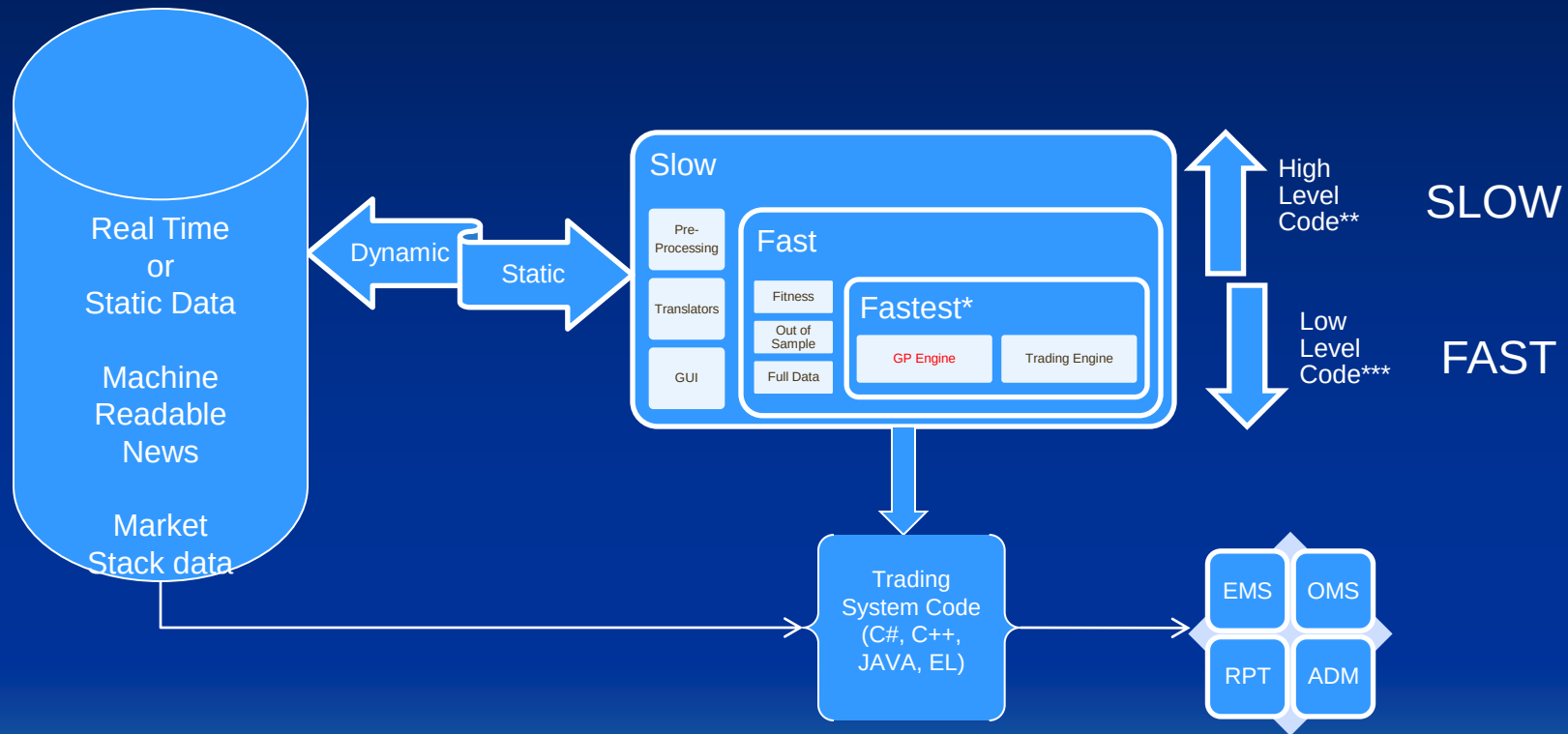
We need to tell the machine it can add

- Arithmetic Functions: +, -, *, ÷, ABS, SQRT, CHS, SCALE
- Transcendental Functions: Trig, Log
- Exponential
- Stack Rotation
- Comparison
- Conditional Statements
- Jumps
- Subroutines-Headers

We need to tell the machine what functions it can use to learn

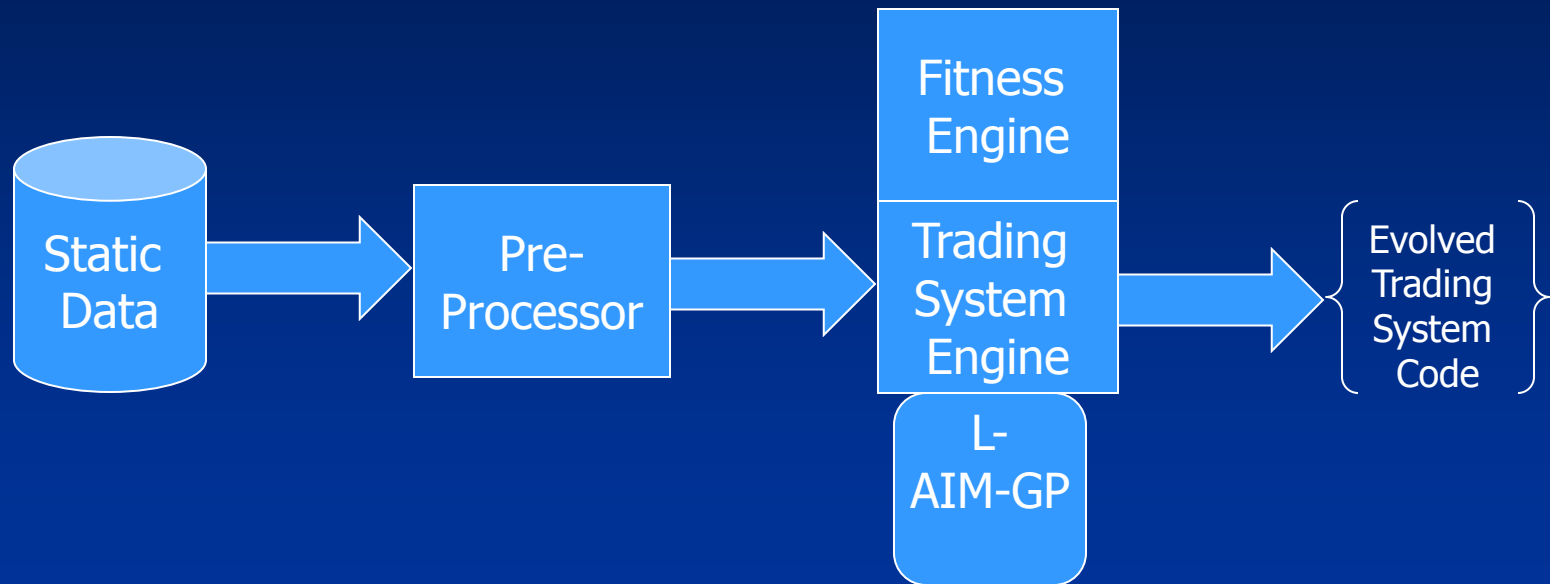
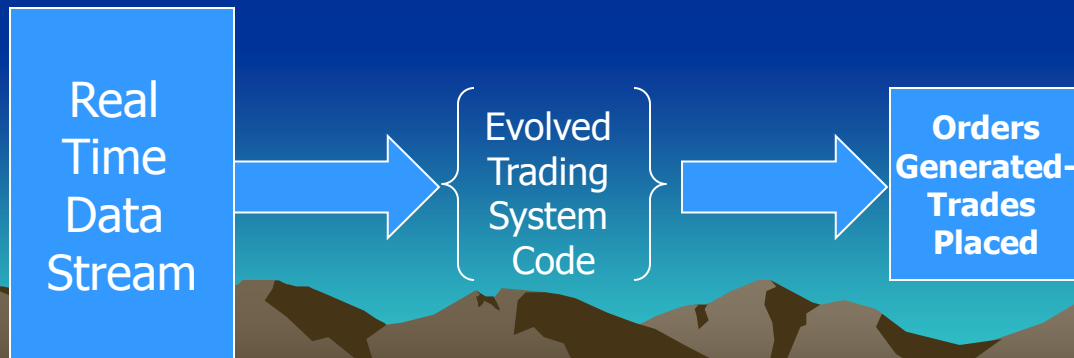
THE ARCHITECTURE

Development followed no specific model since no specific model (~extreme programming) was readily available
~1 million lines of code, 9 languages, 2 companies

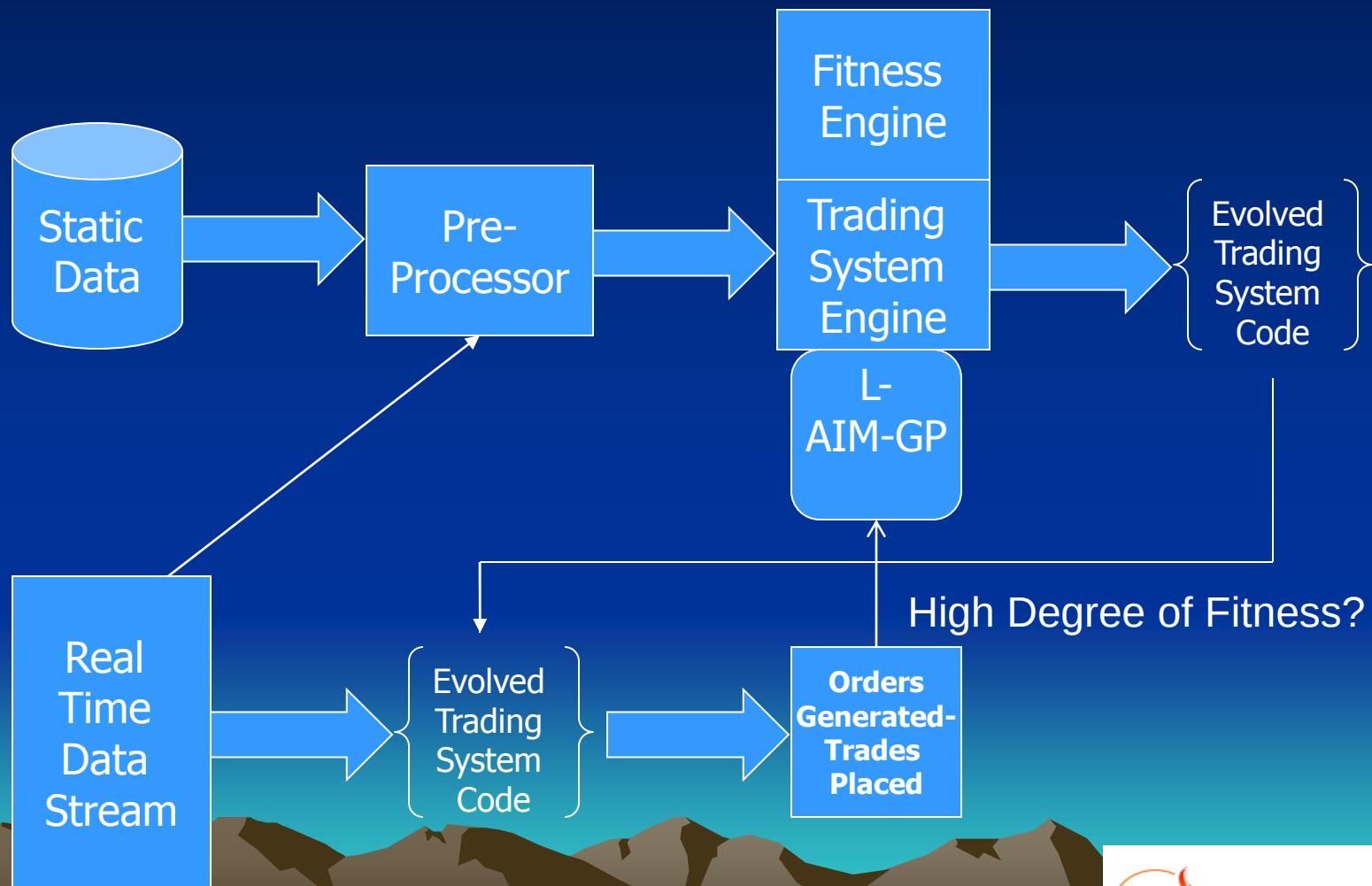


- *GPU/G80, EP or Many Core Implementations
- **VC++.NET, VB, C#, EL
- ***Assembler, C, C++

Algorithm Produces Algorithms



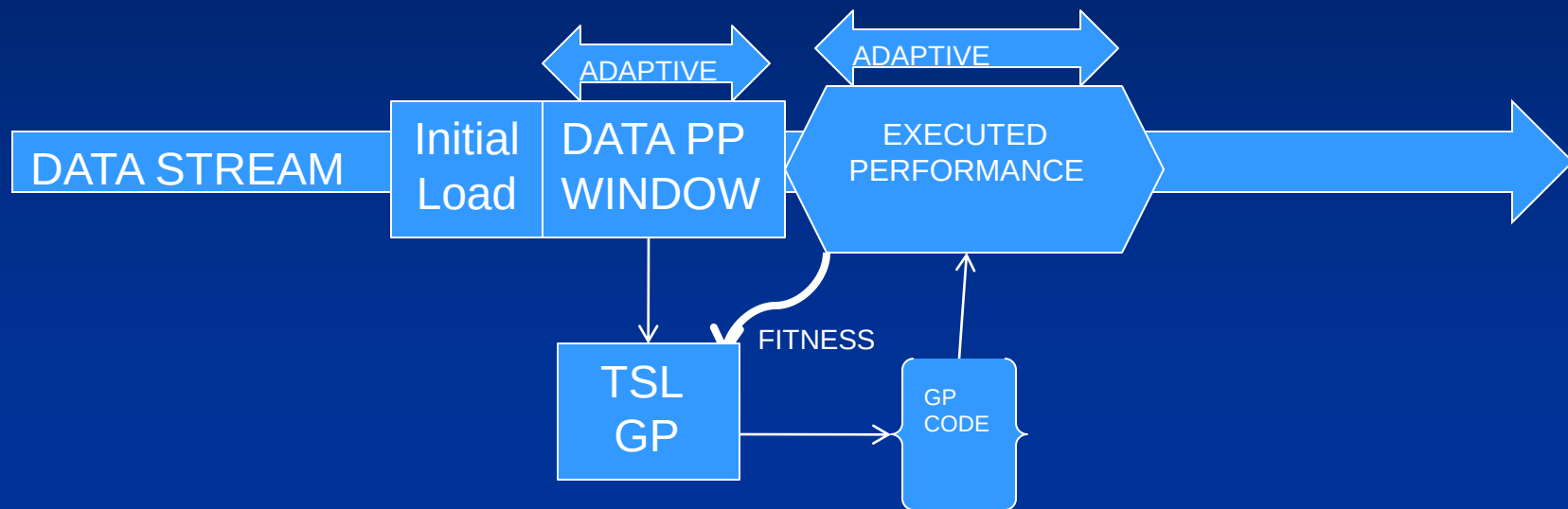
Algorithm on The Fly



POSSIBLE FUTURE ASYNCHRONOUS “WINDOW” MACHINE LEARNING

(Very High Freq Adaptive Trader)

EVOLVED CODE IS UPDATED BASED ON RT FITNESS



FITNESS CAN BE MULTI GOAL

Machine Design Allows Us to Adjust Critical System Metrics as Targeted Fitness Function



Net Profit



Drawdown



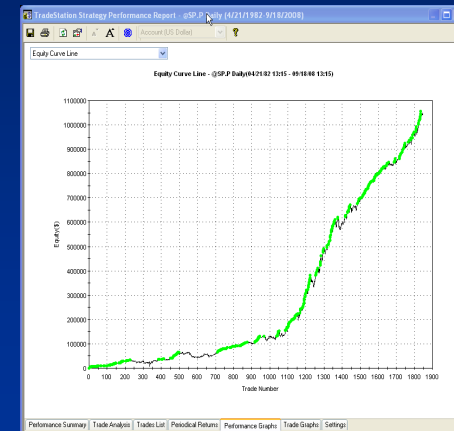
Percent Accuracy



Profit Factor



Average Trade



PRODUCES
CODE

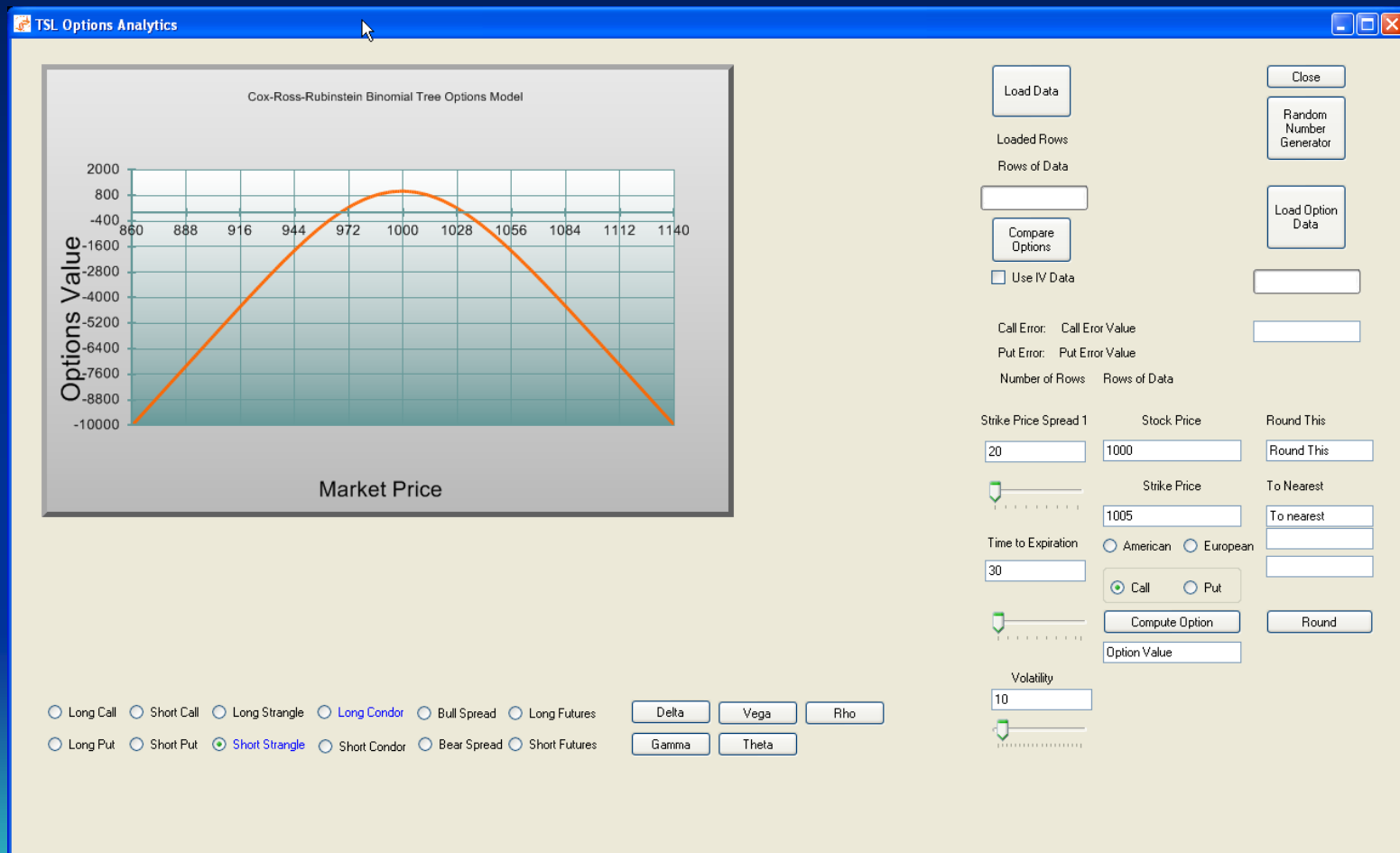
PARAMETER OPTIMIZATION

Using a GA, for example

- High risk of over fitting data
- Very, very slow
- Limits search to preprogrammed sets
- Requires an Existing System (GA)

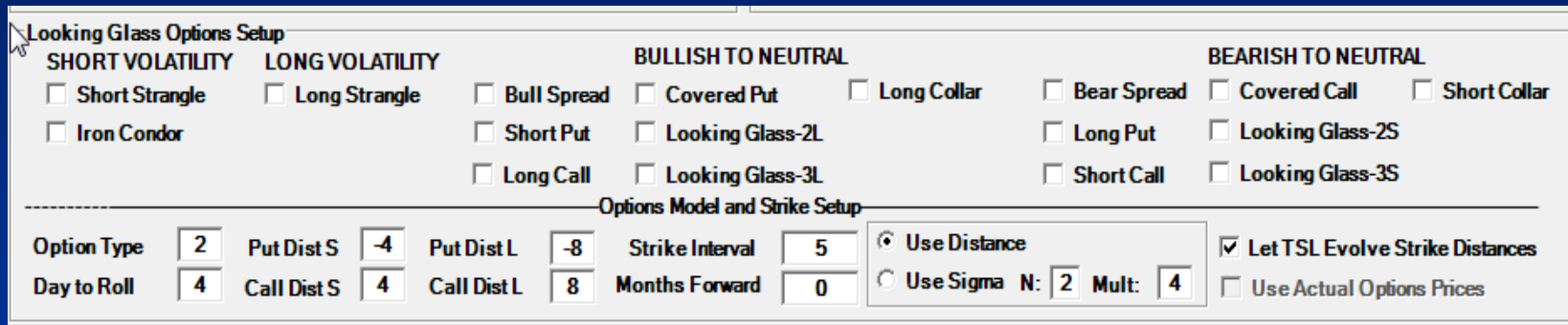


OPTIONS COMBINATIONS GP EVOLVED STRUCTURES



TSL OPTIONS

TSL will Auto-Design Options Strategies and Combinations



Looking Glass Options Setup

SHORT VOLATILITY LONG VOLATILITY BULLISH TO NEUTRAL BEARISH TO NEUTRAL

☐ Short Strangle ☐ Long Strangle ☐ Bull Spread ☐ Covered Put ☐ Long Collar ☐ Bear Spread ☐ Covered Call ☐ Short Collar

☐ Iron Condor ☐ Short Put ☐ Looking Glass-2L ☐ Long Put ☐ Looking Glass-2S

☐ Long Call ☐ Looking Glass-3L ☐ Short Call ☐ Looking Glass-3S

Options Model and Strike Setup

Option Type Put Dist S Put Dist L Strike Interval ☒ Use Distance ☐ Let TSL Evolve Strike Distances

Day to Roll Call Dist S Call Dist L Months Forward ☐ Use Sigma N: Mult: ☐ Use Actual Options Prices

- The GP will evolve strike distances, entry and exit locations against fitness choice
- All Greeks are exported in spreadsheet
- Signals produced in TradeStation Print Log
- CRR B-Tree and Bjerk-Stens using StdDev Volatility
- Enhancement: Use TSL developed real options look-up engine

TSL HFT ALGOS NEEDS

CPU or FPGA solutions

- Low latency <10 microseconds internal (FPGA)
- Simulation (to test prior to release)
- Algo Hosting (for TSL algo drops)
- Scalability (and international connectivity)
- Exchanges
- Risk
- Execution gateways (multiple order types)
- Feed handlers
- Algo design can blend Best Execution and Alpha



Limit Fill Adjust

What if the Limit fill is assumed to be on the extreme of a bar during evolution?



Limit Fill Adjust forces evolution to assume Limit based system fills are one tick inside of extremes

Limit or Close

What if the fill is assumed to be on the extreme of a bar?



Limit or Close forces evolution to assume entry on close if the limit price is at a Bar extreme.

Note: TSL includes a “Randomize Fill” Parameter.

FUTURE PLANS

- HFT Colocation (Execution, Feed, Risk and Algo Host)
- Ticker Plant evaluations
- Market depth strategy evolution
- Options data engine integration
- Version 2.x development
- Asynchronous: “evolution on the fly”

WHAT IS EVORUN™

- Industry first “Feature” AutoDesigner
- Reduces design time
- Auto-designs over features in one pass
- Determines optimum Bar Granularity
- Determines optimum TradeType
- Determines optimum Fitness Function
- Determines optimum PreProcessor

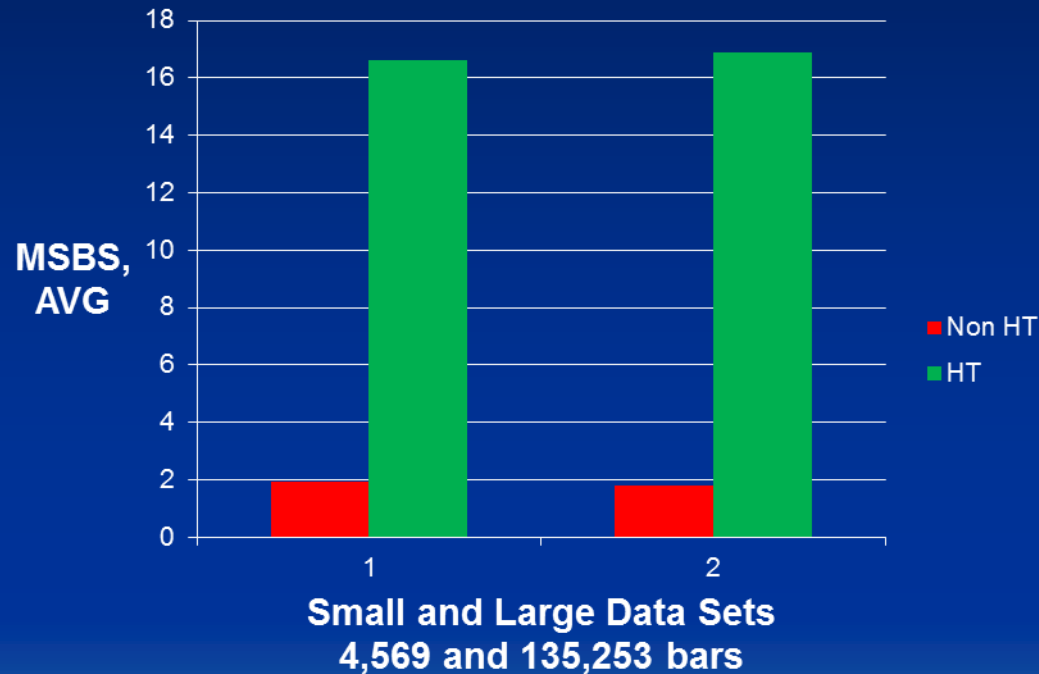


HYPERTHREADED TSL

Million System-Bars per Second (**900 MISBS**)

core i7 990x, 3.47ghz

HT verses Non HT Speed



Even without HT TSL is 60-200 times faster than other algorithms

HIGH FREQUENCY EXAMPLE

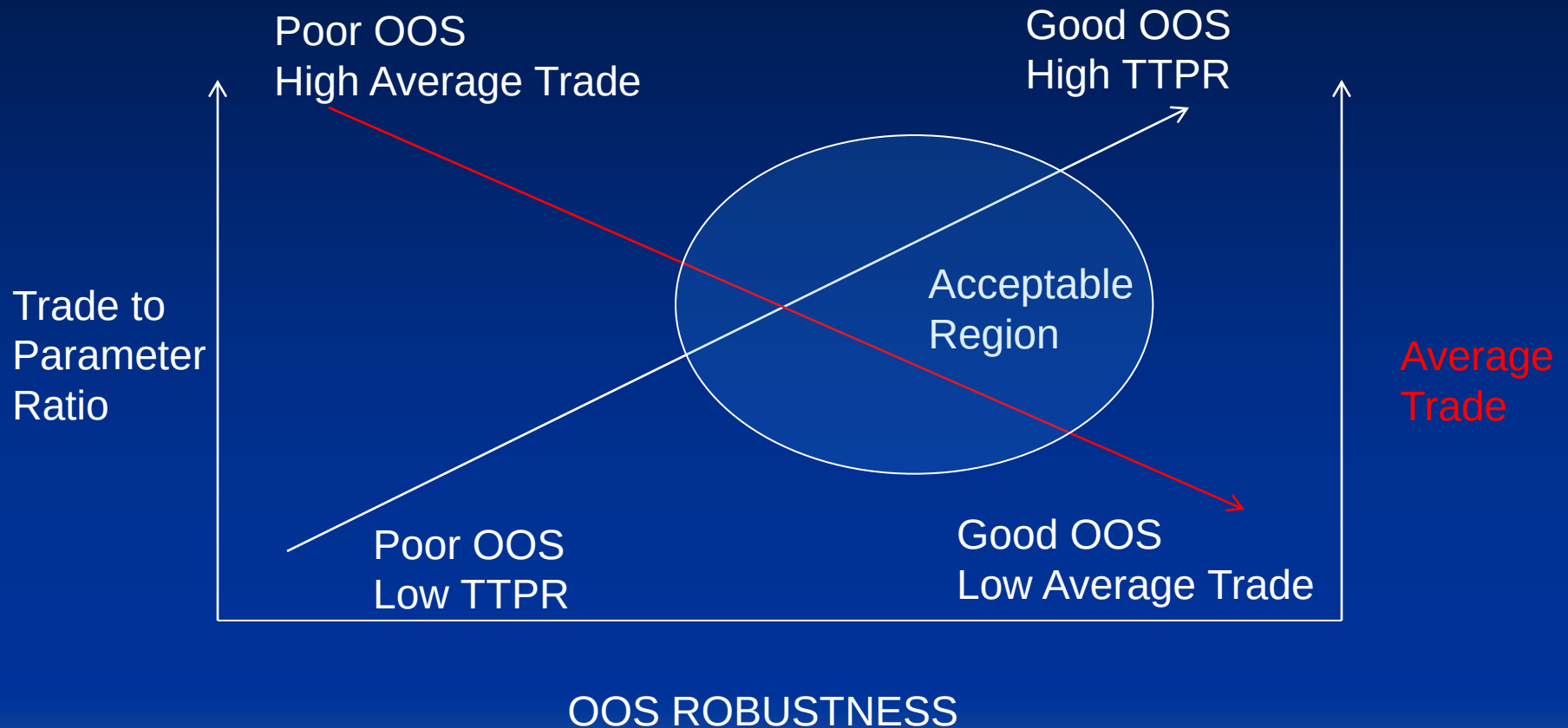
- One Market Data/OneTick OB Aggregator
- 250 millisecond granularity
- Bid/Ask limit directional
- TSL Preprocess
- Evolve Strategy in TSL GP learning engine
- Fill/Market Impact scenarios
- Implement
- See Demo #50

HIGH FREQUENCY EXAMPLE

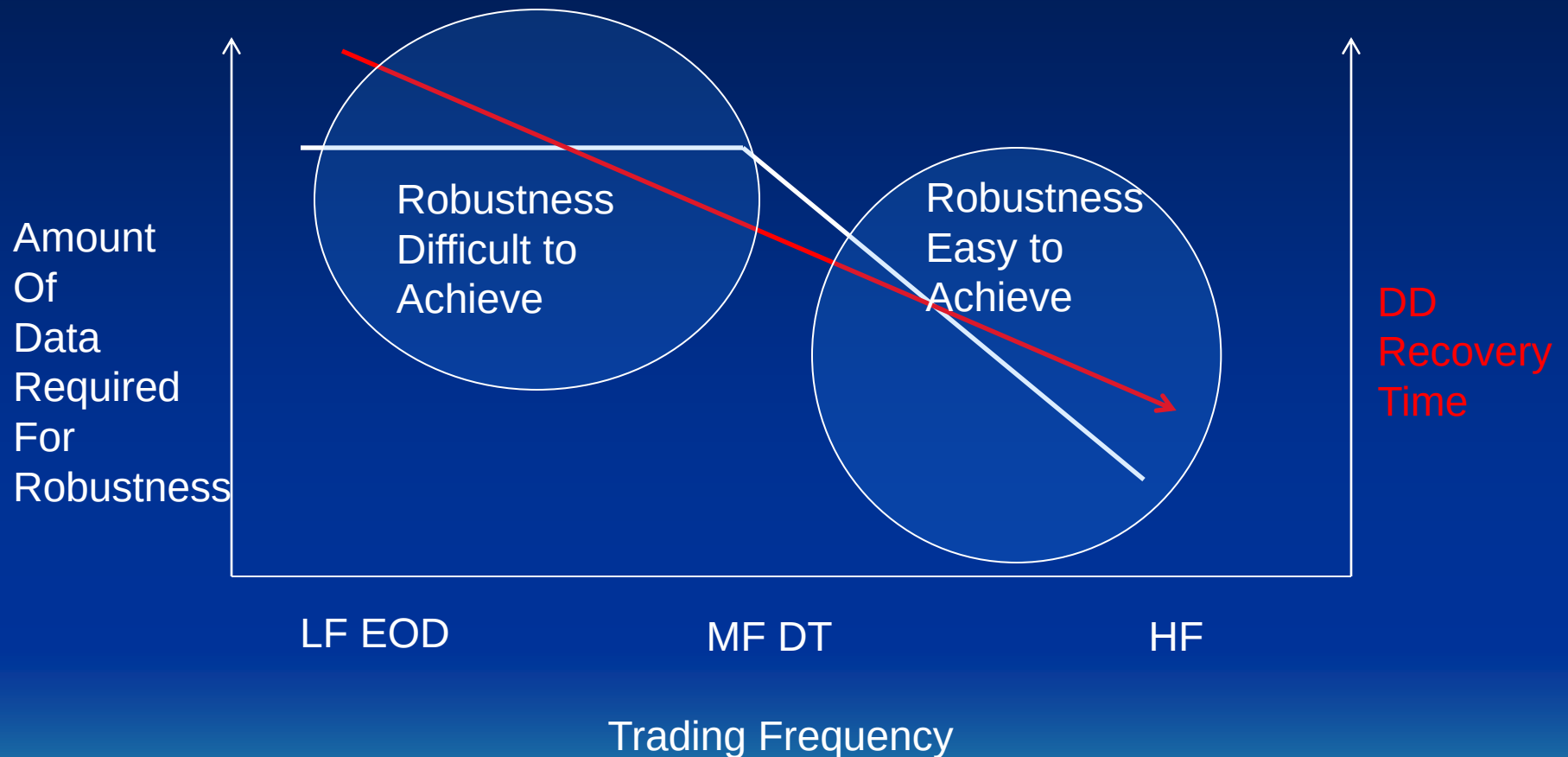
Side Notes

- HFT models need fewer “days” of data compared to LFT
- Market impact difficult to model
- Many granularities from microsecond to second
- Adding Liquidity Rebates can be $>$ total commissions

TRADING SYSTEM DESIGN TRADEOFF



TRADING SYSTEM DESIGN TRADEOFF



WHAT WOULD YOU DO IF THE MARKET IS NOW TRADING LIKE IT DID IN 1993?

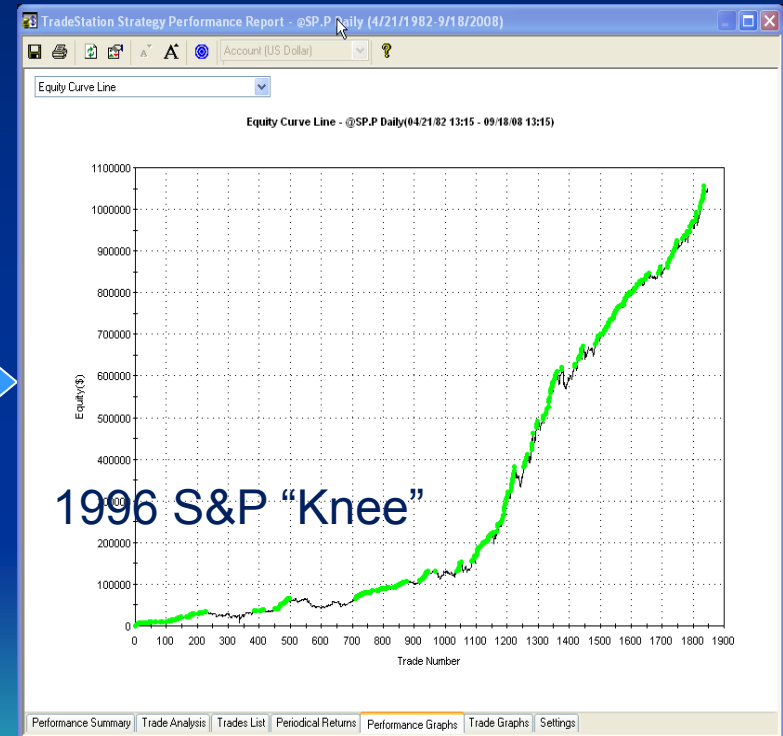
Note: eMini S&P began in 1997
What portion of the data should be OOS?

Risky, non-stationary market data



TSL- α

Controlled Risk (α , σ) Equity Stream



ES THEORETICAL TRADE PERFORMANCE

For a 1 minute bar, Theoretical EV ~ tick size
However, the period alpha is inverse to interval

Bar Size, Minutes	405	135	60	15	1
Expected Value, \$	463	297	188	47	15
Total Trades/Day	0.6	0.75	1.6	5.7	77
Drawdown	8300	5225	4837	3062	1375
Period Alpha	277.8	222.75	300.8	267.9	1155
Reward/Risk	0.03	0.04	0.06	0.09	0.84

So Why Design Symmetrical Systems?



TSL Version 1.3 Update

- EVORUN
- Custom Sessions
- Adds $\text{SQRT}(\text{TotalTrades}) * \text{AverageTrade}$ FF
- Removed TSL Preprocessor DLL. Open Source.
- Avg. Trade and Max DD Targets in FF
- Fixes Thread Leak in HT for long runs
- Minor Bug fixes

TACTICS, FITNESS AND PREPROCESSING

- (TT) Trade Types/Trading Tactics are entry techniques. Example: Enter on Limit
- (FF) Fitness Functions are “Targets” that TSL attempts to design to. Example: NetProfit/Max Drawdown
- (PP) Preprocessors are Patterns, Indicators or other facts used as DNA in TSL. Example: Close>Close[1]
- There are 25 TT's, 40 FF's and 10 PP's allowing 10,000 possible setups to be tested.
- Testing over various bar sizes requires yet more runs
- Clearly there is a need for simplicity and runs reduction

WHAT IS EVORUN™?

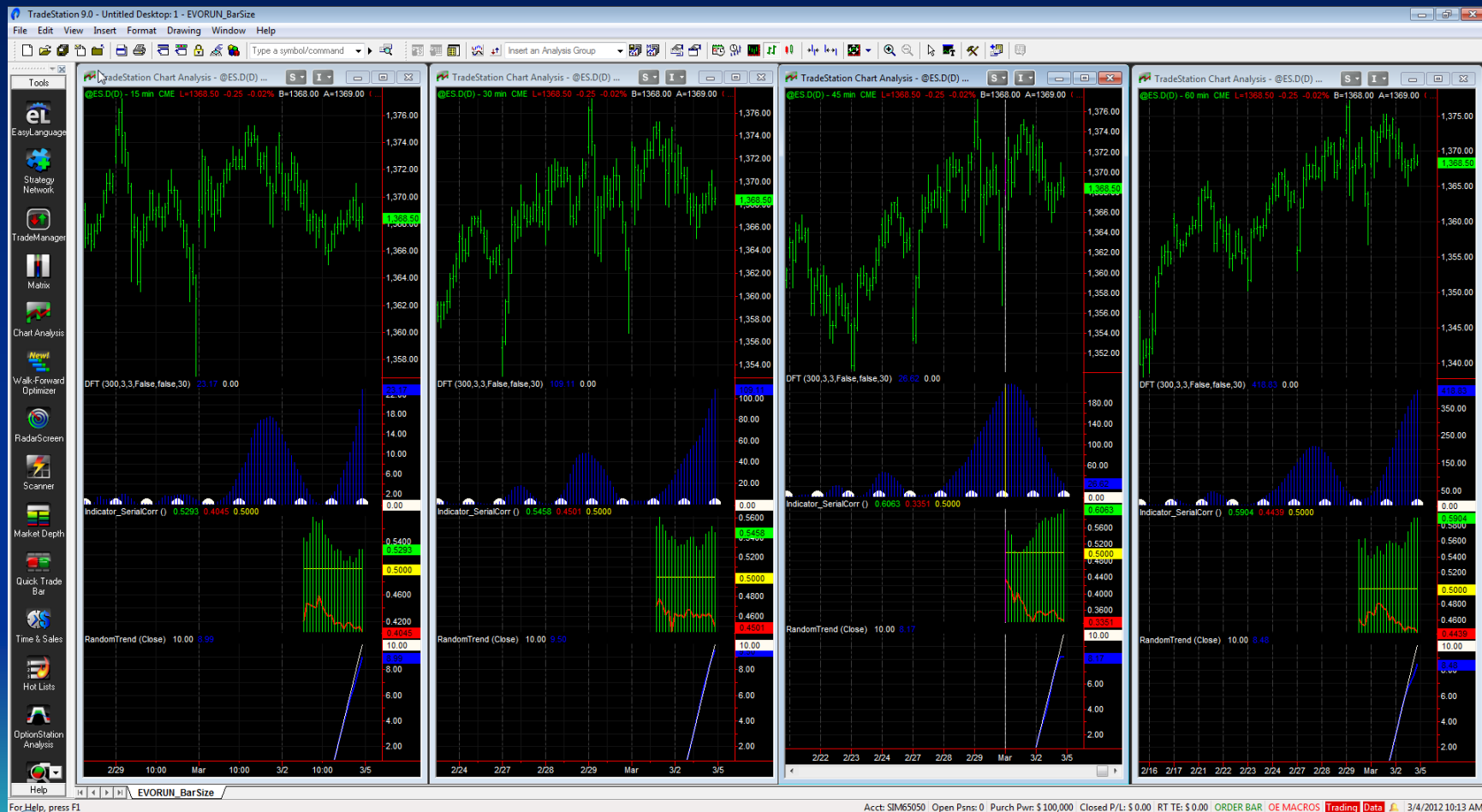
EVORUN is a multi run iterator:

1. Trade Type
2. Fitness Function
3. Preprocessor
4. Bar Size



EVORUN™

WHAT IS YOUR OPTIMUM BAR SIZE?



3/26/2012

EVORUN™

BAR SIZE ITERATION

EVORUN™ Batch Control

☒ **Increment Bar Size each run**

Max increments

Multiplier

☐ **Increment Trade Type each run**

☐ **Increment PreProcessor each run**

☐ **Increment Fitness Function each run**

EVORUN™

GRID REPORT

Trading System Lab EvoRun Grid

	PASS	TRAINING FITNESS	TRADE TYPE	BAR SIZE	PP TYPE	FIT TYPE	TRN NetProfit	TRN DD	TRN AT	TRN TT	OOS NetProfit	OOS DD	OOS AT	OOS TT	ALL NetProfit
	977	0.004535147	21	15	1	1	11025	1237.5	69.3396	159	987.5	775	22.4432	44	11962.5
	476	0.005284016	6	75	1	1	9462.5	1312.5	197.1354	48	1687.5	150	129.8077	13	10375
	985	0.004848485	21	45	1	1	10312.5	1325	180.9211	57	-337.5	712.5	-22.5	15	10300
	285	0.005284016	4	15	1	1	9462.5	1675	12.2255	774	337.5	687.5	1.7857	189	9675
	1025	0.006289308	21	135	1	1	7950	437.5	265	29	-1250	1750	-156.25	7	9500
	589	0.005405406	13	15	1	1	9250	1037.5	91.5842	101	150	1125	4.4118	34	9350
	356	0.004819277	5	15	1	1	10375	1325	225.5435	46	-862.5	1087.5	-123.2143	7	9125
	680	0.005571031	14	30	1	1	8975	1000	115.0641	77	-1312.5	1787.5	-131.25	9	8787.5
	665	0.005044136	14	15	1	1	9912.5	1762.5	32.7145	303	-1187.5	1712.5	-18.2692	65	8725
	718	0.005025126	14	105	1	1	9950	1212.5	171.5517	58	-837.5	1312.5	-93.0556	9	8650
	906	0.005405406	19	75	1	1	9250	1000	171.2963	54	-662.5	1137.5	-50.9615	13	8525
	834	0.005405406	18	75	1	1	9250	1000	171.2963	54	-662.5	1137.5	-50.9615	13	8525
	864	0.00618238	18	165	1	1	8087.5	1050	252.7344	32	-1525	1887.5	-190.625	7	8512.5
	1082	0.004860268	24	15	1	1	10287.5	1300	205.75	50	-1575	1650	-143.1818	11	8487.5
	1032	0.006622517	21	150	1	1	7550	450	251.6667	29	-1250	1750	-156.25	7	8450
	881	0.005586592	19	15	1	1	8950	1137.5	41.8224	214	-575	1287.5	-11.9792	48	8400
	806	0.005586592	18	15	1	1	8950	1137.5	41.8224	214	-575	1287.5	-11.9792	48	8400
	691	0.006369427	14	45	1	1	7850	962.5	117.1642	67	175	612.5	5.6452	31	8375
	916	0.005449591	19	90	1	1	9175	1000	247.973	37	-1000	1887.5	-111.1111	9	8112.5
	1020	0.00536193	21	120	1	1	9325	1025	266.4286	35	-1162.5	1425	-145.3125	8	8100
	927	0.005221932	19	105	1	1	9575	1000	251.9737	38	-1887.5	1887.5	-269.6429	7	8037.5
	700	0.005555556	14	60	1	1	9000	1000	130.4348	69	-1275	1887.5	-115.9091	11	8025
	838	0.006410256	18	90	1	1	7800	1000	173.3333	45	-1750	2087.5	-145.8333	12	7937.5
	728	0.005298013	14	120	1	1	9437.5	987.5	255.0676	37	-1500	1887.5	-187.5	8	7850
	306	0.006896552	4	75	1	1	7250	2187.5	48.0132	151	1287.5	662.5	33.0128	39	7800
	709	0.005617978	14	90	1	1	8900	1000	222.5	40	-350	1150	-25	14	7587.5
	1000	0.005700000	21	90	1	1	8950	1100	247.1429	35	-1000	1887.5	-111.1111	9	7587.5

Close Save Load EvoRun Data

© Copyright 2012. TradingSystemLab

SUMMARY

- TSL is patented and trademarked
- TSL technology is unavailable anywhere else
- TSL strategies are #1, per Futures Truth
- TSL speed grows with CPU speed
- TSL produces EL code for strategies easily
- TSL is customizable
- TSL sets a paradigm shift in strategy design
- TSL has excellent future plans and enhancements

TRADING SYSTEMS DESIGNED IN 3 EASY STEPS

1. Preprocess (TradeStation EL)
2. Evolve (TSL)
3. Translate (TradeStation OMS/EMS)

Quick Notes

- The web site includes:

1. 50 Flash Demos
2. Numerous example systems
3. White Papers, Theses, literature
4. Screen Shots

TEXT BOOKS

- Francone-Genetic Programming
- Brameier/Banzhaf-Linear Genetic Programming
- Poli-A Field Guide to Genetic Programming
- Chen-Genetic Algorithms and Genetic Programming in Computational Finance
- Aldridge-High Frequency Trading



CONCLUSION

MACHINE DESIGNED TRADING
SYSTEMS ARE HERE!

www.tradingsystemlab.com

408-356-1800