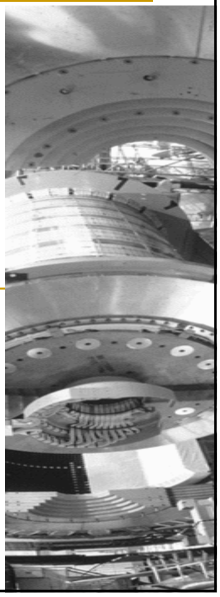
# Parallel Genetic Programming

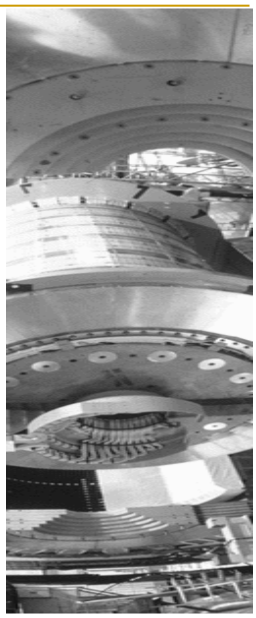
***Francisco Fernández de Vega***
***Grupo de Evolución Artificial***
***University of Extremadura (Spain)***



grupo de evolución artificial
gea
artificial evolution group

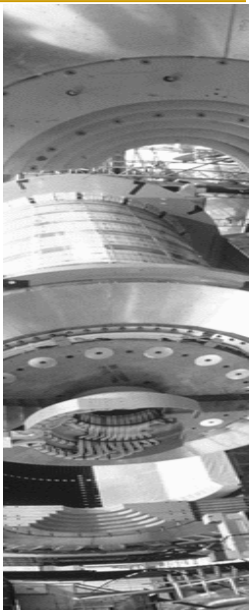
http://gea.unex.es

# Summary:



- Introduction.
- History.
- Parallel GP.
- The Island Model.
- Successful Applications.
- Future.

## Summary:



- Introduction.
- History.
- Parallel GP.
- The Island Model.
- Successful Applications.
- Future.

## Some considerations

- Genetic Programming can be considered a Machine Learning system:
  - “[machine learning] is the study of computer algorithms that improve automatically through experience [Mitchell, 1996].
- The emphasis is on *learning* (instead of on *knowledge).*
- The dream of computers that program themselves [Samuel, 1963] could be reached soon.

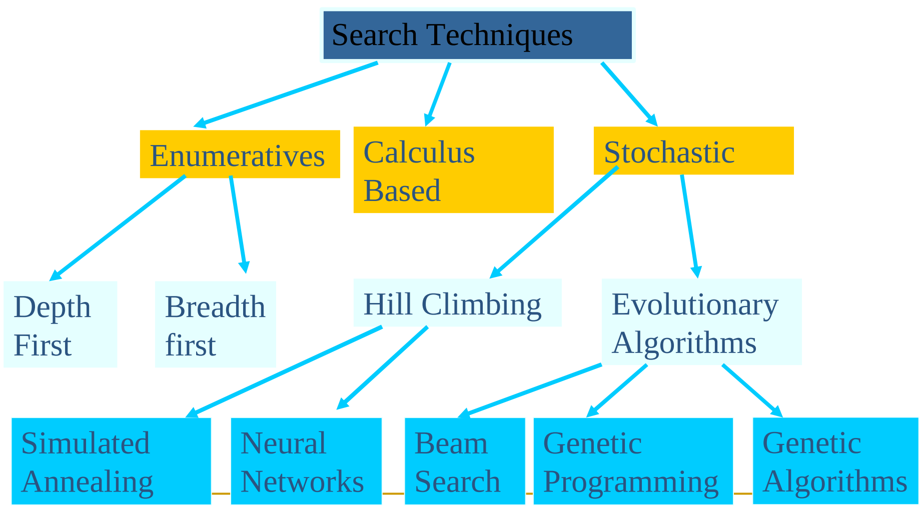
## What's Genetic Programming?

- According to Banzhaf et al*, GP is a system that induce computer programs by evolutionary means.
- GP (Koza, 1992) is a kind of Evolutionary Algorithm (EA). (Genetic Alorithms, Genetic Programming, Evolutinary Strategies, Evolutionary Programming).
- EAs can be seen as search techniques (stochastic search technique).

*Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D. Genetic Programming, an Introduction. Morgan Kaufmann 1998.



## Different Search Techniques



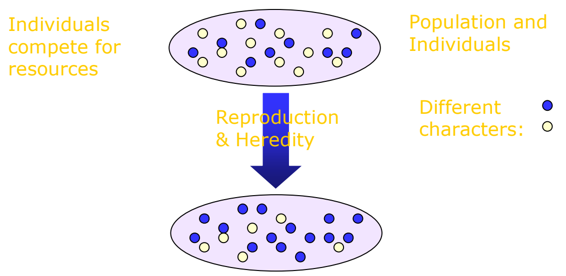

As classified by Banzhaf et al

## How does an EA work

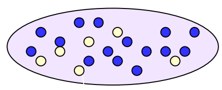
- A summary:
  - T=0;
  - Initialize and evaluate [P(t)]
  - While not stop_condition do
    - P´(t)=variation [P(t)]
    - Evaluate P´(t)
    - P(t+1)=select [P´(t),P(t)]
    - T=t+1
  - end while

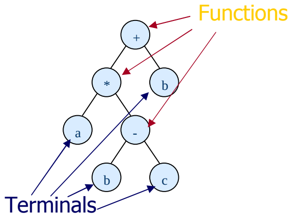


## How does GP work?



# How does GP work?






GP Individual

Genetic Operators:

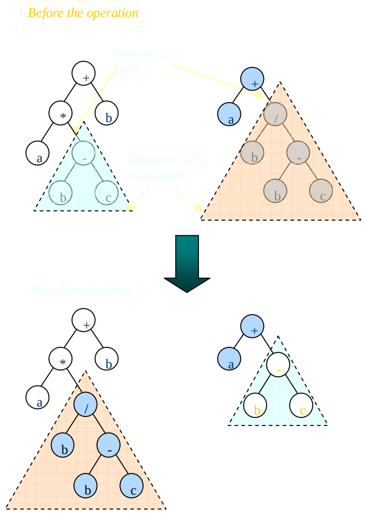
- Crossover.
- Mutation.
- Selection.
- Reproduction.

Fitness Function



## Genetic Programming

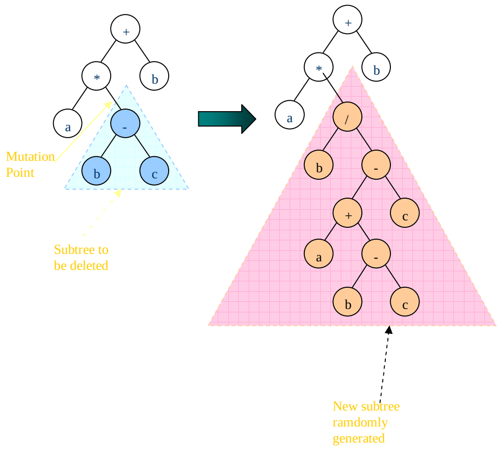
C
R
O
S
S
O
V
E




Relies on the Building block hyp

## Genetic Programming

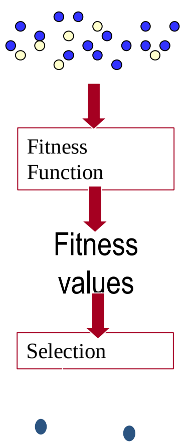
M U T A T I O N





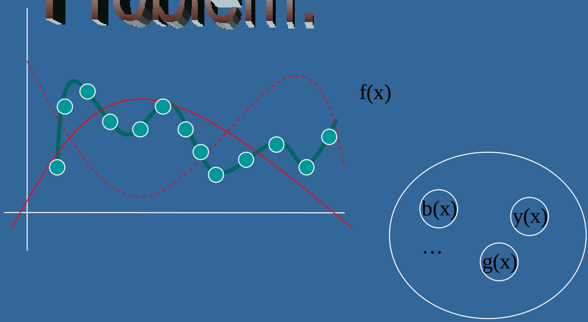

## Genetic Programming

S E L E C T I O



Genetic Programming

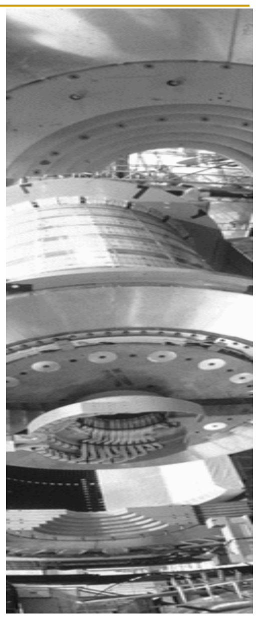
# Problem:






## Summary:

- Introduction.
- History.
- Parallel GP.
- The Island Model.
- Successful Applications.
- Future.



## History

- There are two main ideas behind Parallel Eas:
  - Increase performances:
    - in principle, by adding processors, memory and interconnection networks and putting them to work together on a given problem.
  - Modifying the underlying algorithm can also help in the finding of solutions.



## History

- Ideas involving both EAs and Parallel Computing can be traced back to Holland, 1976.
- But the field had to wait until early 1980s when parallel implementations appear.
- Grefenstete, 1981, was one of the first in examaining some issues concerning parallel implementations of Gas.

## History

- Other researchers began more systematic studies: Gross, Cohoon, Tanese, Pettey, Georges-Schleter, Mühlenbein, and Manderick*.
- They studied Hypercubes parallel architecture, distributed models, theoretic models, island models, and cellular model.

*See "Parallelism and Evolutionary Algorithms", E. Alba & M. Tomassini, IEEE Transactions on Evolutionary Computation, Vol. 6, NO. 5, Oct 2002, 443-462 for a review summary.



## Parallelism & EAs

- Flynn model is still widely accepted for classifying computer architectures.
- The taxonmy is based on the notion of instruction and data stream:
  - SISD: Single instruction, Single Data stream.
  - SIMD: Single Instruction, Multiple Data stream (the preferred model).
  - MISD: Multiple instruction, single data stream.
  - MIMD: Multiple instruction, multiple data stream.
- Shared or Distributed Memory.

## Taxonomy

- Parallel EAs can be classyfied attending to different features.
- M Nowostawski and R. Poli 1999:
  - Master/Worker: A single population and the fitness evaluation of multiple individuals in parallel.
  - Static subpopulations with migration.
  - Static overlapping subpopulations without migration.
  - Massively Parallel genetic algorithms.
  - Dynamic demes.
  - Parallel Steady-state Ga
  - Parallel Messy Ga
  - Hybrid methods.



## Taxonomy

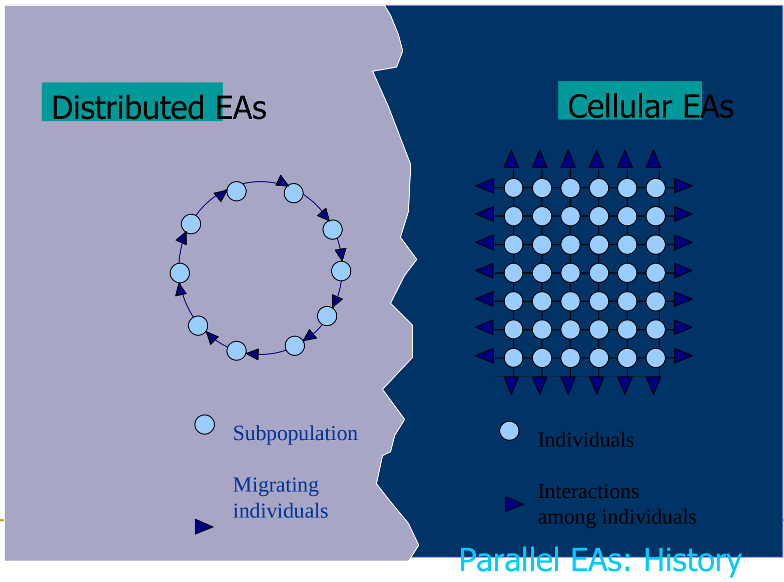
- M. Tomassini, 1999:
  - Global parallel evolutionary algorithm (parallelization at the fitness level) also called by the author master/worker model (coarse-grained model).
  - Island distributed evolutionary algorithms (population based approach).
  - Cellular evolutionary algorithm (fine-grained model).

## Structured EAs

- Structured populations has been used for improving EAs.
- Two main types of algorithms:
  - Distributed EAs.
  - Cellular EAs.
- On the oposite side, panmictic EA is the classic model.
- None of the models require a parallel implementation.



## Structured EAs

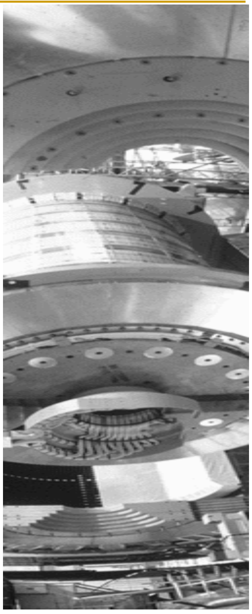


## Nonstandard Structured EAs

- We could use different parameters/representations in different subpopulations (Tanese, Lin & Punch & Goodman, Herrera & Lozano & Moraga).
- These algorithms are sometimes called heterogenous.



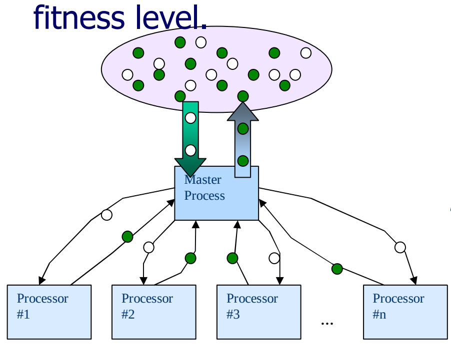
## Summary:



- Introduction.
- History.
- Parallel GP.
- The Island Model.
- Successful Applications.
- Future.

## Parallel Models

Parallelising at the fitness level.




This model is also called "global model".

Why should we use a parallel model?

- We want to increase performances.

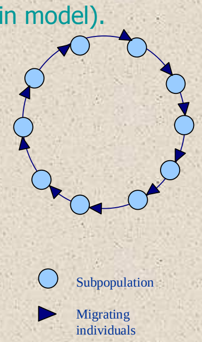
How could we parallelise?

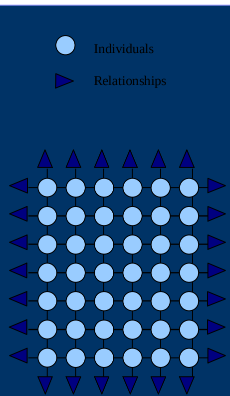
- At the individual level.
- At the population level.
- At the fitness evaluation level



## Parallel Models

Parallelising at the population level (also called island model or coarse-grain model).




Fine-grained model (also called Grid or Cellular model)

## Parallel GP: A review

- Juillé and Pollack 1995 presented one of the first attempts to parallelize GP.
- They implemented a global model, although also presented some result using sub-populations.
- The proposal was somehow specific for the SIMD model they were using.
- One of their aims was to reduce interprocessor communication.



## Parallel GP : A review

- P. Tuffs, 1995, presented the same year another master/worker parallel version of GP.
- He approached a classification problem by means of GP. -the development of a system to do data mining on a fairly large (multi-gigabyte) database of credit-card transactions. The task was to classify customers and predict their future behavior-
- Probably it was the first attempt to parallelize GP (the book correspond to year 1993, although was published on 1995).

## Parallel GP : A review

- Andre and Koza 1996 presented another Parallel GP Implementation using a network of 66 transputers (VLSI device containing 32 bit on-chip processor, memory and links)
- They employed the island-model.
- An appropriate migration rate showed improvement in the computational effort required for the Boolean 5-parity function.(64 demes, SubPop_size=500, Migration_rate=5%)
- The main conclusion was that Parallel GP could achieve super-linear speedups.



## Parallel GP : A review

- K. Stoffel and L. Spector 1996, implemented a parallel version of GP using linear programs evaluated by means of a stack.
- Several processors generate independently its own segment of the next generation.
- They implemented crossover in parallel: Individuals from different processors may undergo crossover.

## Parallel GP : A review

- Oussaidène et al 1997, presented a parallel implementation of GP for trading model induction.
- They employed the global model architecture (parallelization at the fitness level), employing a master/worker model, where each node from the network is in charge of evaluating individuals comming from a master node.
- The master node is in charge of the main GP algorithm.
- The model may undergo a load imbalance problem.



## Parallel GP: A review

- Results offered by Koza were questioned later by W. F. Punch 1998. His experiments on the Royal Tree problem were not so optimistics. His main conclusions were that multiple-solution problems would be more amenable to multiple populations than single-solution problems.
- On the other hand, non-deceptive problems would be more amenable to multiple populations than deceptive problems.
- He only tried a set of parameters for the parallel model.

## Parallel GP : A review

- Fernandez et al 1999, presented experimental results on an island-model implementation of GP.
- Although results were preliminar, this is the first time some important parameters of the island-model are tested (communication topology).



## Parallel GP : A review

- Folino et al, 2000, presented a new implementation of GP using the cellular model.
- Fernández et al, 2000, studied more deeply the relationship among several important parameters for the island-model (subpop size, number of subpop, communication topology).

## Parallel GP : A review

- The latest results on both the island and cellular GP models have been presented very recently:
  - F. Fernández et al., 2003, described latest results with the island-model, while G. Folino et al, 2003, presented a Cellular Scalable implementation for GP. Their results are compared with previously described results using parallel GP.



## Parallel GP tools

- Many GP tools allow the use of “demes” but simmulated in a sequential fashion.
- There have been several parallel implementations during the last few years.
- Several languages (C, java, C++) and communication frameworks (sockets, java rpc, pvm, mpi …) have been employed.

## Communication Tools

- PVM: Parallel Virtual Machine (V.S. Sunderam, "PVM: A framework for parallel distributed computing, " J. Concurr. Practice and Experience, vol.2, no.4, pp. 315-339, 1990")
- MPI: Message Passing Interface.(Message Passing Interface Forum, "MPI: A message-passing interface standard," Int. J. Supercomput, Applic., vol.8, no.3-4, pp.165-414, 1994)
- GLOBUS.
- Others (Sockets, Java-RMI…)



## Parallel GP tools

- Chong, 1998, presented DGP, a java based distributed approach to genetic programming on the Internet.
- F. Fernández et al, 1999, developed a parallel GP tool implementing the island-model, and communicating subpopulations by means of PVM. This tool was later improved by means of MPI (Fernández et al, 2000).
- Spezzano et al, 2001, presented CAGE: A tool for parallel genetic programming applications. They implemented the cellular model.
- Classic LilGP software (Punch) also has a couple of parallel version implemented using PVM and MPI (see Fernández parallelilgp)

## Parallel GP tools

- DREAM project: It is aimed at providing a framework for evolutionary computation.
- It allows distributed computing.
- Any Evolutionary Algorithm could be used, by adjusting some parameters, within DREAM.
- Founded by European Union.
- See: http://www.dcs.napier.ac.uk/~benp/dream/dream.htm



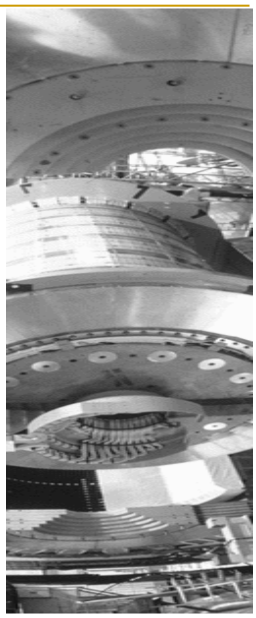
## Parallel GP tools

- Paradiseo: Parallel and Distributed Evolving Objects.
- It is based on EO (Evolutionary Computation Framework).
- Includes tools for:
  - Population Based Metaheuristics.
  - Single Solution Based Metaheuristics.
  - Multi-Objective Metaheuristics.
  - Parallel and Distributed…

## Parallel GP tools

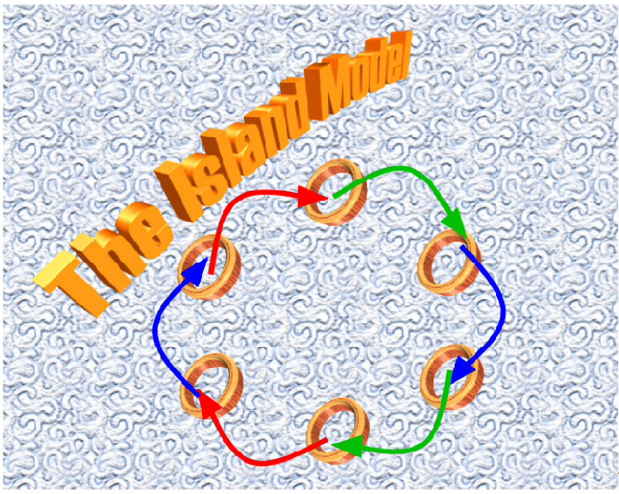
- ECJ: A Java-based Evolutionary Computation Research System.
- Includes asynchronous Island Model over TCP/IP.
- Multiobjective Optimization.,

## Summary:



- Introduction.
- History.
- Parallel GP.
- The Island Model.
- Successful Applications.
- Future.

# Island Model




Important Parameters:

Size of Subpop.

Topology.

Communication rate.

Granularity.

Synchronization.

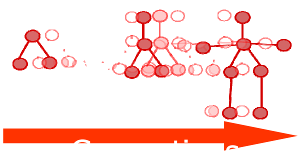
Important concern:

Comparing results.



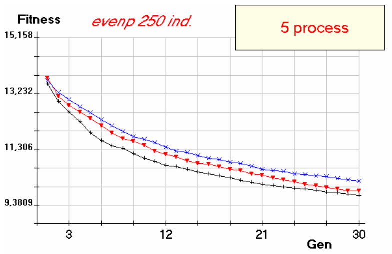
# Island Model – Comparing results

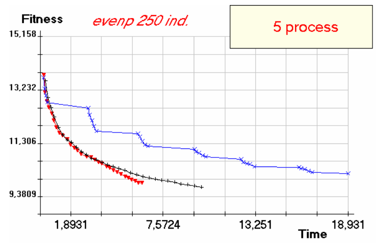
- Traditionally experimental results are shown comparing Fitness/Generation.
- Two reasons for avoiding this kind of comparisons:
  - The Bloat phenomenon in GP.

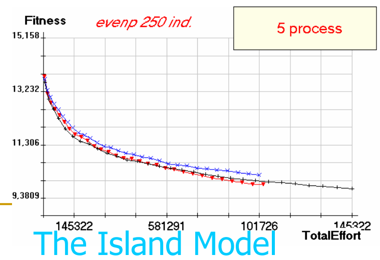


  - Populations with different size require different time to evaluate a generation.

## Island Model – Comparing Results






## Island Model – Measuring Results

- Proposal: Evaluate Fitness/Effort (for convergence) or Fitness/Time (for speedup) (Fernández, Galeano, Gómez).

**Computational effort:**

**The total number of nodes GP has evaluated for a given number of generations.**

$PE_g$=i*p*avg_length$_g$

p:the number of populations

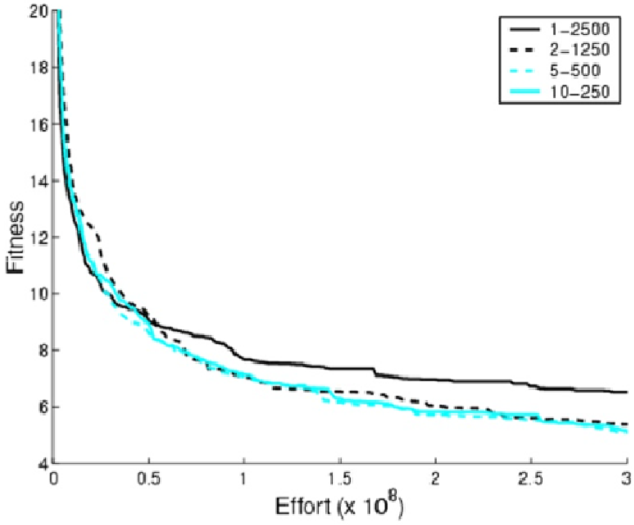
i:the number of individuals per population

avg_length$_g$: the average length of individuals in all the populations in generation g.

The computational effort $E_g$ at a generation g is:

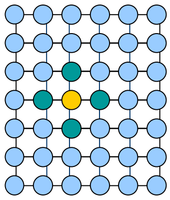
$$E_g = PE_g + PE_{g-1} + \dots + PE_1 + PE_0$$

## Island Model – Comparing Results



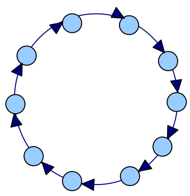

## Island Model - Topology

- Andre and Koza 1996 connected each subpopulations with 4 neighbors in the N, E, W, S directions.



# Island Model - Topology

- Punch, 1998, used a typical Island model with ring topology.



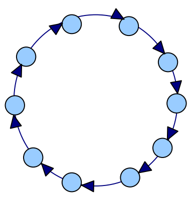
Ring Topology

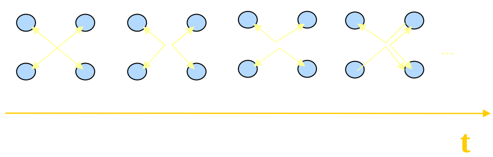
- He employed that topology for his experiments but no comparisons with different topologies were provided in the paper.



# Island Model - Topology

- Fernandez et al 2000 introduces a random topology and compare it with grid and ring topology.

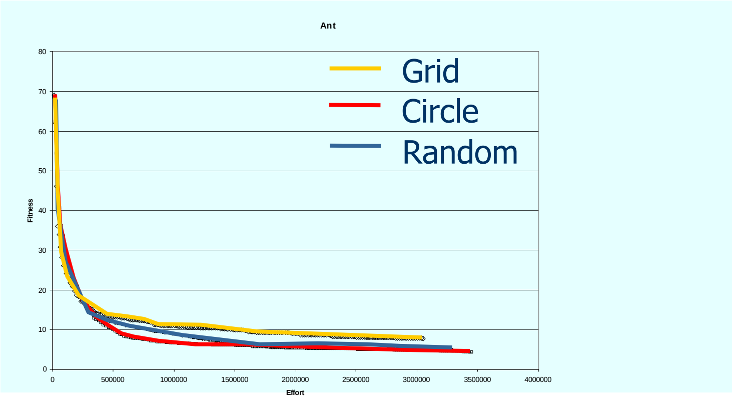





Ring Topology Random Topology: It changes dynamica

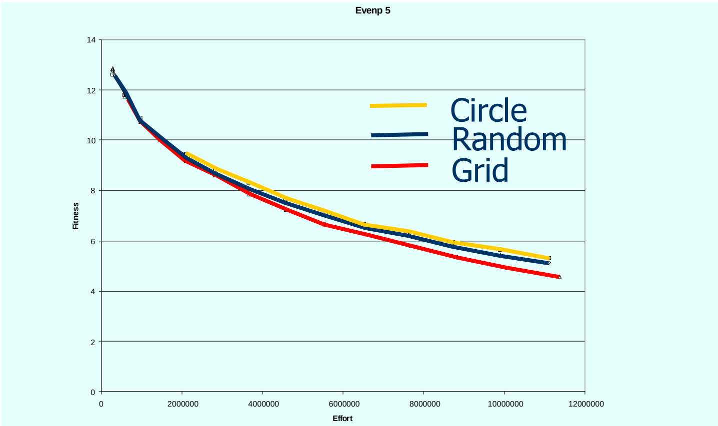
- The main conclusion is that if the remaining parameters stay fix, there are no significant differences when changing topology.

## Island Model - Topology






## Island Model - Topology



## Island Model – Migration Rate

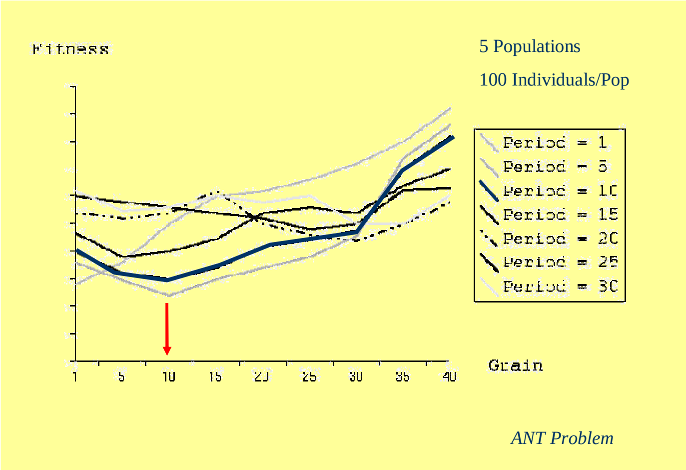
- How many individuals should migrate each migration step?
- Deppending on the number of individuals, results are different.  The limits are:
  - 0 individuals migrating (isolated populations)
  - All the individuals migrating.
- Different migration rates applied in literature:
  - Juillé & Pollack: 1 migrating individual per subpopulation.
  - Andre and Koza: 0%-8% migrating individuals per subpop.
  - Punch 1998: 2 individuals.



## Island Model – Migration Rate

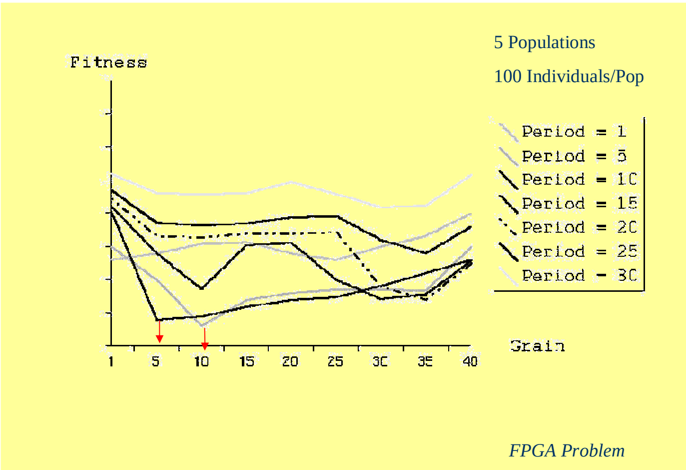
- Fernandez et al 2003: best migration rate is between 5% and 10% in 4 test problems (2 classic  and 2 real-life problems):
  - Even Parity 5.
  - Ant Problem.
  - Routing and Placing circuits on FPGAs.
  - Medical Diagnosing.

## Island Model – Migration Rate






## Island Model – Migration Rate

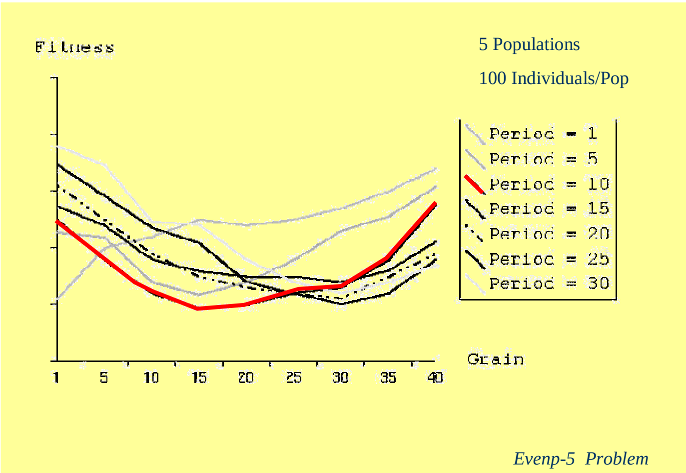


## Island Model – Migration Frequency

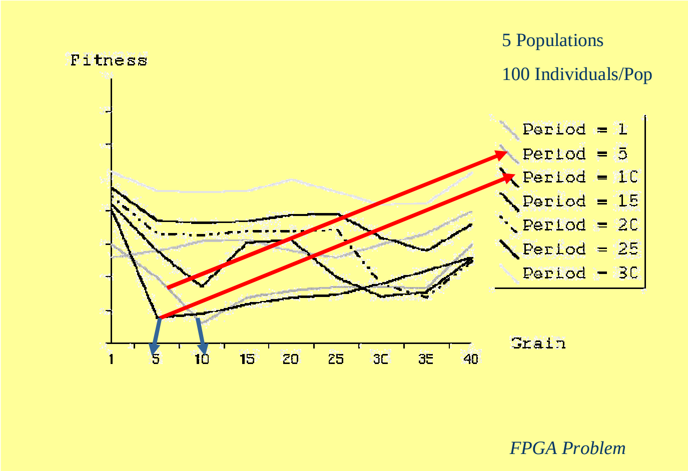
- Both Juillé & Pollack and also Andre and Koza employ migration every generation.
- In Punch 1998, subpop. wait for 10 generations before the migration step.
- In Fernandez et al 2003, a wider study have been carry on, comparing different frequencies.
- Best convergence results appear when about 10% of individuals from each subpopulation are sent every 5-10 generations.



## Island Model – Migration frecuency

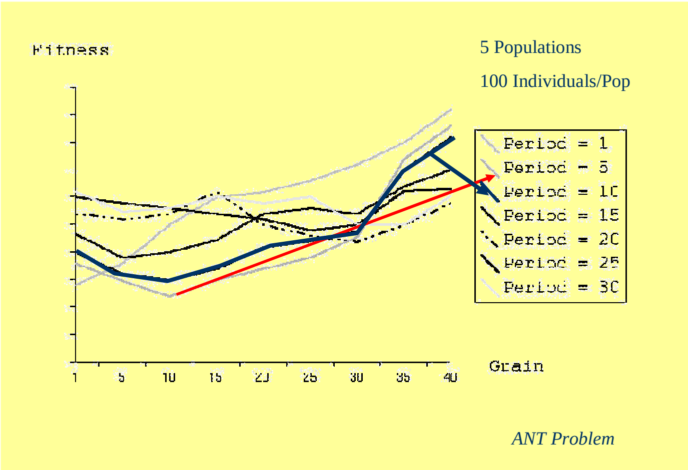


# Island Model – Migration summary






# Island Model – Migration Rate



## Island Model – Migration Summary

- For large values of the grain, exchange individuals less frequently.
- For low values of the grain, exchange more frequently.
- Recommendation: Exchange 10% of the population every 10 generations.



## Island Model – Subpop. Size

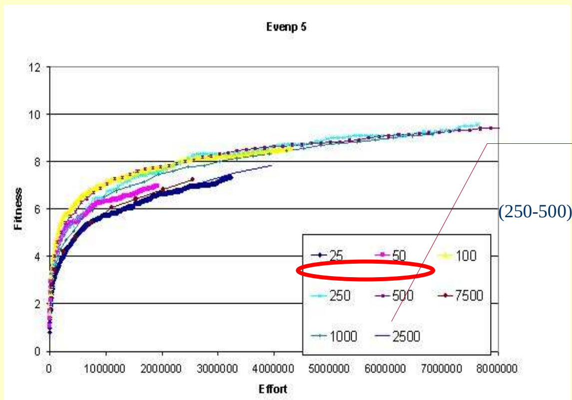
- Experiments presented by Andre & Koza, make use of the large computational capability of the network they employed.
- 32000 individuals are distributed among 64 demes, each with 500 individuals.
- Punch 1998, employed 5 populations, 200 individuals each, and also 7 populations, 700 individuals each.

## Island Model – Subpop. Size

- F. Fernández et al., 2003, presents a set of trials.
- Conclusions:
  - There is a number of individuals with which best results are obtained (regardless of the number of subpops).
  - We must carefully select the number of subpops, not any number of populations obtain the same results.



## Island Model – Subpop. Size

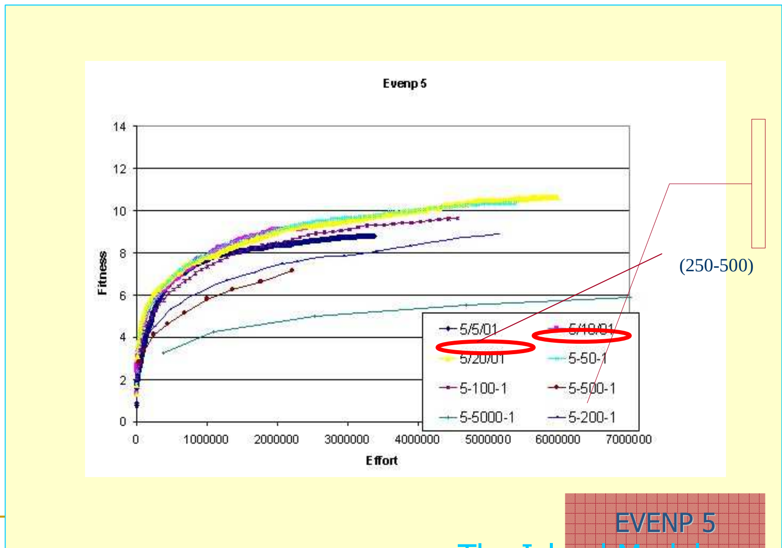



(250-500)

EVENP 5

T
64

## Island Model – Subpop. Size






## Island Model – Synchronisation

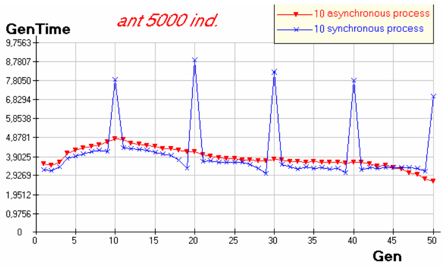
- Synchronism is an important issue when using Parallel GP: different individuals may require different processing time for their evaluation.
- Two models:
  - Synchronous: Exchange step takes place at a given generation.
  - Asynchronous: Populations send individuals when they are ready, and check every generation if new incoming individuals are awaiting.

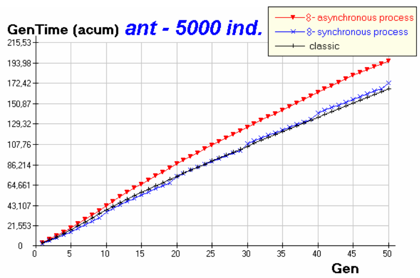
# Island Model – Synchronisation

- Andre and Koza worked with an asynchronous model:  each generation is typically working on different generations after a few ones.
- Dracopoulos and Kent, employed the synchronous model in both the global and island models.
- Fernández et al, 2002, presented a study comparing synchronous and asynchronous models in monoprocesor systems.



# Synchronous - Asynchronous




Results obtained using 1 proc

Synchronous model better for monoprocessor system.
Asynchronous model better for parallel systems.

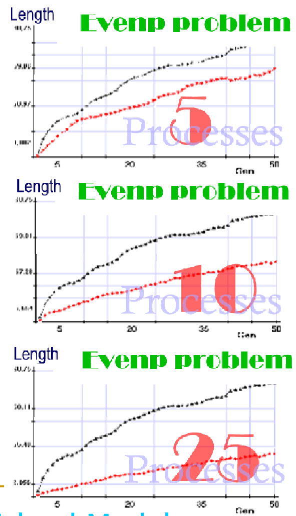
## Synchronous - Asynchronous

- Tongchim and Chongstitvatana, presented a comparison among models using a restricted migration policy.
- Their results only focus on a problem, and shows better performance with the asynchronous model.

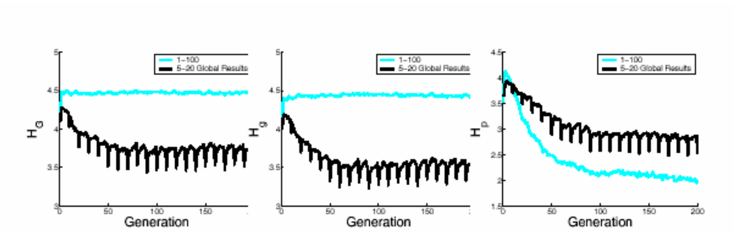


## Island Model - Bloat

- The Island Model seems to prevent the bloat phenomenon.



## Some comments on diversity




Genotypic diversity decreases.
Phenotypic diversity improves.

## What about fault tolerance?

- Fault tolerance is an important issue.
- Different techniques have been employed:
  - Check pointing.
  - Redundancy.
  - Others…
- Is GP Fault tolerant?

# What about fault tolerance?

Figure 2. Even parity 5 problem, with a population length of 2000 and 10% volatility rate

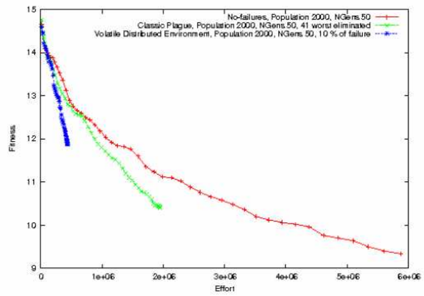
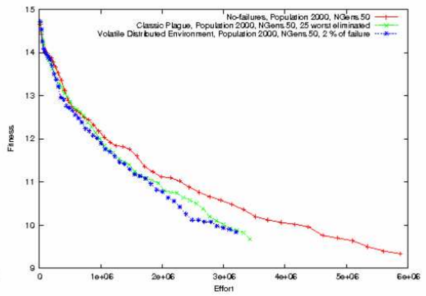



Figure 3. Even parity 5 problem, with a population length of 2000 and 2% volatility rate

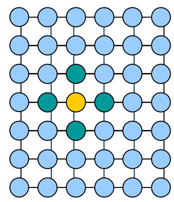



# The Cellular Model

- Folino et al, 2003, have presented a comparison with panmitic and island model approach.
- The method provides results of similar quality than the island model (an small error in the comparisons seems to favor the cellular model in the paper, but a detailed revision shows that results are similar).
- They apply the model to induce decision trees.

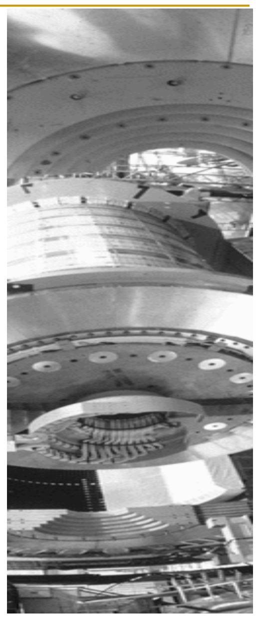
## The Cellular Model

- Each individual is located in a grid position.
- Individuals interact only with their neighboring ones.





## Summary:



- Introduction.
- History.
- Parallel GP.
- The Island Model.
- Successful Applications.
- Future.

## Some Applications

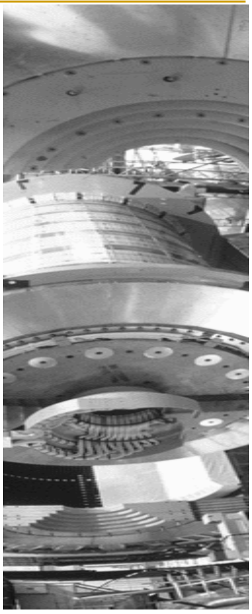
- C. Miccio et al, 1995, described an implementation on a T3D computer for inducing binary decision diagrams.
- M. Oussaidène et al, presented an application to trading model induction.
- F. Fernández, 2001, described a proposal for solving the problem of Placement and Routing of circuits on FPGAs.
- Folino apply the cellular model to generate decision trees.



## Some Applications

- Koza et al 2000, presents a list of "Human-Competitive results obtained by Means of Genetic Programming", including:
  - Synthesis of Analog Circuits.
  - Synthesis of PID controllers.
  - Applications to biomedicine (protein detection).
  - Previously patented inventions, reinvented.
  - Some patented invention.

## Summary:



- Introduction.
- History.
- Parallel GP.
- The Island Model.
- Successful Applications.
- Future.

## Future

- Some topics for future research:
  - Theoretical models.
  - Heterogeneous models.
  - Improvements by means of better scheduling policies.
  - Bloat phenomenon.

## References


- Tufts, P.  "Parallel Case Evaluation of Genetic Programming.  In 1993 Lectures in Complex Systems, Es. L. Nadel and D. Stein, SFI Studies in the Science of Complexity, Lec. Vol. VI, Addison Wesley, 1995, pp.591-596.
- H. Juillé & J. B. Pollack, Parallel Genetic Programming and fine-grained SIMD architecture" in Working Notes for the AAAI Symp. Genetic Programming, E. V. siegel and J. R. Koza, Eds. Cambridge, MA: mit, Nov. 10-12, 1995, AAAI, pp. 31-37.
- Stoffel, K., and L. Spector. 1996.  High-Performance, Parall, Stack-Based Genetic Programming.  In Koza, John R., Goldberg, David E., Fogel, David B., and riolo, Rick L. (eds) Genetic Programming 1996:  Proceedings of the First Annual Conference, 224-229.  Cambridge, MA:  The MIT Press.


## References


- D. Andre & J. R. Koza, "Parallel Genetic Programming:  A Scalable Implementation Using the Transputer Network Architecture", in P. Angeline and K. Kinnear editors, Advances in Genetic Programming 2, Cambridge, MA, 1996.  The MIT Press.
- D. C. Dracopoulos and S. Kent, "Bulk Parallelisation of Genetic Programming", in Proceedings 3rd Int. Workshop Applied Parallel Computing, pp.216-226.  Springer Verlag 1996.
- M. Oussaidène, B. Chopard, O. Pictet, and M. Tomassini, "Parallel Genetic Programming and its application to trading model induction", Parallel Comput. Vol. 23, pp. 1183-1198, 1997.

# References


- P. Martin, "A Hardware Implementation of a Genetic Programming System Using FPGAs and Handle-C". In Genetic Programming and Evolvable Machines, 2, 317-343, 2001.
- M. Tomassini, "Parallel and Distributed Evolutionary Algorithms: A Review", in Engineering and Computer Science, J. Wiley and Sons, Chichster, 1999, K. Miettinen et al (eds), pp. 113-133, 1999.
- M. Tomassini & E. Alba, "Parallelism and Evolutionary Algorithms", in IEEE Transactions on Evolutionary Computation, Vol. 6, No. 5, pp. 443-462.


# References


- R. Poli, "Evolution of graph-like programs with parallel distributed genetic programming. In proceedings of the 7th International Conference on Genetic Algorithms, T. Bäck (ed.), Morgan Kaufmann, San Fco, CA, 1997, 346-353.
- M. Nowostawski and R. Poli, "Parallel genetic algorithms taxonomy", in Proc. Third Int. Conf. Knowledge-Based Intell. Inf. Engrg. Syst. KES'99, IEEE Computer Society: New York, Aug. 1999., pp.88-92.
- G. Folino, C. Pizzuti, G. Apezzano, "A Scalable Cellular Implementation of Parallel Genetic Programming", in IEEE Transactions on Evolutionary Computation, Vol. 7, No. 1, Feb 2003.

# References


- Heywood, M.I and Zincir-Heywood, A. N., "Register based genetic programming on FPGA computing platforms", in Proc EuroGP'2000, R. Poli, W. Banzhaf, W. B. Langdon, J. F. Miller, P. Nordin, and T. C. Fogarty (eds.), LNCS, vol. 1802, Edinburgh, Springer: Berlin, 15-16 Apr., 2000, pp-44-59.
- F. Fernández & M. Tomassini, "Solving the Ant and the Even Parity-5 problems by means of parallel genetic programming ", Late Breaking Papers at the 1999 Genetic and Evolutionary Computation Conference, Brave and We (eds.), pp 88-92.


# References


- Folino et al, "Genetic Programming and Simulated Annealing: a hybrid method to evolve dicision trees." in R. Poli. Et al (eds), Proceedings EuroGP 2000, LNCS 1802, pp. 294-303. Springer Verlag, Berlin, 2000.
- G. Spezzano, G. Folino, and C. Pizzuti, "CAGE: A tool for parallel genetic programming applications", in Proc. EuroGP'2001 Genetic Programming, vol. 2038, J. Miller et al eds., 2001, pp- 64-73.
- F. Fernandez et al 2000 "Experimental study of Multipopulation parallel genetic programming" in R. Poli. Et al (eds), Proceedings EuroGP 2000, LNCS 1802, pp. 283-293. Springer Verlag, Berlin, 2000.

# References


- F. Fernández et al., 2003, "An Empirical Study of Multipopulation Genetic Programming ", en Genetic Programming and Evolvable Machines, Kluwer Accademic Press, 2003, vol. 4, no. 1, pp. 21-51.
- G. Folino et al, 2003, "A Scalable Cellular Implementation of Parallel Genetic Programming" in IEEE Transactions on Evolutionary Computation, vol. 7, no. 1, 2003, pp. 37-53.
- F.S. Chong, "A Java based disributed approach to genetic programming on the Internet", Master's thesis, Univ. Birmingham, Birmingham, AL, 1998.


# References


- F. Fernández, J.M. Sánchez, M. Tomassini, and J. A. Gómez, "A parallel genetic programming tool based on pvm", in vol. 1697 LNCS pp. 241-293.
- F. Fernández, M. Tomassini, L. Vanneschi and L. Bucher, "A distributed computing environment for genetic programming using MPI", in vol 1908, LNCS, pp. 322-329.
- W. Punch, lilgp: http://garage.cse.msu.edu/software/lil-gp/lilgp-index.html
- W. B. Langdon, Distributed Genetic Programming, available at ftp://ftp.cs.bham.ac.uk/pub/authors/W.B.Langdon/gp-code/DGP/

# References


- W. F. Punch, 1998, "How Effective are Multiple Populations in Genetic Programming", Proceedings of the third Annual Genetic Programming Conference, July 22-25 1998. pp. 313-318.
- F. Fernández, G. Galeano, J. A. Gómez, "Comparing Synchronous and Asynchronous Parallel and Distributed Genetic Programming models", in LNCS 2278, Pp. 326-335. Springer Verlag 2002.
- J. R. Koza, M. A. Keane, J. Yu, F. H. Bennet, W. Mydlowec, "Automatic Creation of Human Competitive Programs and Controllers by Means of Genetic Programming". In Genetic Programming and Evolvable Machines 1, 121-164(2000). Kluwer Academic Publishers.


# References


- C. Miccio, E. Sánchez, M. Tomassini, "Parallel Genetic Programming Induction of Binary Decision Diagrams", in EPFL Supercomputing Review, N. 7, 1995. Echle Politechnique Federale de Lausanne, Switzerland.
- F. Fernández, 2001, Fernández, "Distributed Genetic Programming Models with application to Logic Synthesis on FPGAs", PhD Thesis. University of Extremadura. February 2001.
- F. Fernández, E. Cantú-Paz, Introduction Special Issue on Parallel Bioinspired Algorithms. Journal of Parallel and Distributed Computing, Elsevier Volume 66, Issue 8 , August 2006, Pages 989-990