Parallel Models for Evolutionary Algorithms

Adrian Grajdeanu
agrajdea@gmu.edu

ECLab
George Mason University
2003 Summer Lecture Series

Presentation Plan

- →History and origins of parallel models for Eas
- →Design Issues of a Parallel Model
- →Canonical parallel models
  - → Master/Slave
  - → Finely grained
  - → Coarsely grained
- →Hierarchical parallel models
- →Final considerations
- →Questions

History and Origins of Parallel EAs

Bethke (1976) in his study he describes a (global) parallel implementation of a conventional GA and of a GA with a generation gap (i.e., it only replaces a portion of its population in every generation)

Grosso (1985) proposed an implementation of a serial simulation for a concurrent formulation

Tanese (1987), Pettey *et al* (1987) are two of the earliest parallel implementations. The population of a GA was broken into a relatively small number of sub-populations and each processing element in the architecture was assigned an entire sub-population and executed a rather standard GA.

Then Cohoon *et al* (1987) shows that the *punctuated equilibria* theory of natural systems transfers to parallel implementations of EAs and leads to bursts of evolutionary progress.

Gordon *et al* (1992) and Adamidis (1994) consecrated the term of *island model* parallel GA.

Presentation Plan

- →History and origins of parallel models for Eas
- →Design Issues of a Parallel Model
- →Canonical parallel models
  - → Master/Slave
  - → Finely grained
  - → Coarsely grained
- →Hierarchical parallel models
- →Final considerations
- →Questions

Design Issues of a Parallel Model

→Decide first what steps of the EA want to parallelize?
- →Look at problem specifics
- →Consider available hardware

→Choose a parallel model
→Decide on the individual EAs.

→Choose a topology
→Synchronous or Asynchronous model?
→Consider communication overhead: is it worth it?
→Envision your migration amount, frequency and policy.

→Use a toolkit or not?
→Tune your parameters

→Duplication is next to impossible, so collect extensive results and keep them!
→Decide clearly how to compare this parallel algorithm with other (non)parallel algorithms.

Presentation Plan

→History and origins of parallel models for EAs
→Design Issues of a Parallel Model
→Canonical parallel models
- → Master/Slave
- → Finely grained
- → Coarsely grained

→Hierarchical parallel models
→Final considerations
→Questions

Master Slave model for parallel EA

a.k.a. **Global Parallel EA**

Idea

→Maintain a single population
→Have one process perform management of population
→Dispatch to slave processes the evaluation of new individuals
→May perform genetic operators application in parallel too

Advantages

→Works well for small genotype footprint and expensive evaluations
→Scales very well up to a certain number of small processes
→Suited for star topology and heterogeneous systems

Disadvantages

→Scalability beyond a certain level - master process becomes bottleneck
→Keeping the workload balanced is tricky in heterogeneous systems

Master Slave model for parallel EA

Synchronous or Asynchronous

If the main process stops and waits for all evaluations to complete before moving to next generation, the algorithm is said to be synchronous. Otherwise asynchronous.

A synchronous Global Parallel EA carries the same search as a serial EA, with speed being the only difference.

An asynchronous Global Parallel EA has different population dynamics. Its results are also more difficult to duplicate.

Master Slave model for parallel EA

Shared or Distributed Memory

Can be efficiently implemented in both models. In shared memory model each slave process writes back its results without conflicts. On distributed memory model, inter-process communication comes into play.

Balancing strategy

The number of individuals dispatched to each slave process can be fixed or dynamically determined. A fixed strategy is less efficient because it may lead to unbalance. A dynamic strategy where each slave process takes on more load as it becomes available inherits communication or shared resource synchronization overhead.

Master Slave model for parallel EA

Topology and Speedups

Reasonable speedups were obtained with up to 16 processors, but further attempts proved futile due to:

- →The overhead of communication between the master process and the slaves becoming significant
- →Operating systems inadequacies when scheduling computationally intensive processes to available processors

With such a small number of processes different topologies proved to lead to similar results, leaving researchers without a clear winner.

Master Slave model for parallel EA

Other sources of parallel gain

In some implementations the genetic operators application was orchestrated in parallel. However these operations are so simple that often times the parallel gains are offset by the communication and synchronization overhead

Selection may be also performed in parallel. Some forms of selection require information about whole population and thus are not very appropriate. However a tournament type selection would be prime candidate.
In addition, Branke *et al* (1997) parallelized different types of global selection on a 2D grid of processors and showed that their algorithms are optimal for the topology used. (O(N) on a N x N grid)

Presentation Plan

- →History and origins of parallel models for EAs
- →Design Issues of a Parallel Model
- →Canonical parallel models
  - → Master/Slave
  - → Finely grained
  - → Coarsely grained
- →Hierarchical parallel models
- →Final considerations
- →Questions

Finely Grained model for parallel EA

Idea

→Maintain a single population spatially-structured into sub-populations called demes.
→Selection and mating are restricted to a small neighborhood
→Neighborhoods overlap to allow some interaction among all individuals

Advantages

→Suited for massively parallel computers
→Can easily parallelize many features of the EA: selection, survival, mating

Disadvantages

→The gain in efficiency is worth only on parallel computers, otherwise too much communication overhead
→Decisions of the topology, neighborhood size and shape complicate the design and understanding of the dynamics.

Finely Grained model for parallel EA

Topology

Cohoon (1987): the topology doesn't matter much as long as it is densely connected and has a small diameter to insure adequate mixing as time progresses

However:
Schwehm (1992) compares the following topologies: ring, torus, 16x8x8 cube, 4x4x4x4x4 hypercube and a 10-D binary hypercube, He found that the torus topology led to fastest convergence. (no mention of the resulting quality though!!!)
Anderson and Ferris (1990) experimented with two rings, a hypercube, two meshes and an 'island' (only one individual in each deme overlapped with other demes). Conclusion was that for their problem (assembly line balancing) the ring and island structures were the best.
Baluja (1992, 1993) compared two variations on a linear structure and a 2-D mesh concluding that the mesh gave best results on almost all problems tested.

Finely Grained model for parallel EA

Neighborhood

Manderick and Spiessens (1998) observed that the performance of the algorithm degraded as the size of the neighborhood increases. At the extreme, if the size of the neighborhood was big enough, this parallel GA was equivalent to a single panmitic population.

Sarma and De Jong (1996) found that the ratio between the size of the neighborhood and the size of the whole grid is critical parameter that determines the selection pressure.

Presentation Plan

→History and origins of parallel models for EAs
→Design Issues of a Parallel Model
→Canonical parallel models
- → Master/Slave
- → Finely grained
- → Coarsely grained

→Hierarchical parallel models
→Final considerations
→Questions

Coarsely Grained model for parallel EA

a.k.a. **Island Model**

Idea

→Split the population in many sub-populations - islands
→Alternate periods of extensive isolated evolution with migration
→During isolated evolution each process runs on its island a full blown EA
→At certain times, few individuals migrate between islands

Advantages

→Uniquely suited for message passing parallel systems
→Map well on many of topologies (mesh, ring, hyper-cube, etc.)
→More then hardware accelerators for EAs

Disadvantages

→More complex design decisions due to increased number of parameters as well as dynamics of multiple EAs running in parallel

Coarsely Grained model for parallel EA

'Punctuated Equilibria' theory

→It is the biological theory that justifies the Island Model
→Introduced by Elredge and Gould (1972) to explain the missing links in fossil records.
→It states that most of the time, populations are in equilibrium, but changes in the environment can trigger rapid evolutionary changes.
→Cohoon (1987) shows that this property of bursts of rapid evolutionary progress does show up in several applications. He observed little change between migrations, but new solutions were found shortly after individuals were exchanged.

In Holland's (1975) terms

→Exploitation arises from isolated evolution
→Exploration comes from infusion of migrants
→Alteration between the two above hold the promise that island models would be more then just hardware accelerators for EAs

Coarsely Grained model for parallel EA

Definition

P is the whole population P_i are individual sub-populations.
$M = N \times \mu$.
M-overall size of population, N - number of islands, μ - size of sub-population.

Epoch is the period of isolated evolution
G_i is a number of generations each island evolves in isolation

S a N x N matrix, usually but not necessarily symmetrical and 0 diagonal. $S_{i,k}$ are the number of individuals from P_i to migrate into P_k at the end of each epoch.

Coarsely Grained model for parallel EA

```
Islands Model(E, N, μ)
{
  Concurrently for i ← 1 to N                          Initialization
     Initialize(P_i, μ);

  for epoch ← 1 to E
  {
     Concurrently for i ← 1 to N                       Isolated Evolution for an epoch
        Sequential_EA(P_i, G_i);

     Concurrently for i ← 1 to N
        P_i' = P_i;
     Concurrently for i, j ← 1 to N
        P_i' ∪= Migration(P_i, S_i,j);                  Migration
     Concurrently for i ← 1 to N
        P_i = Assimilate(P_i');
  }

  ExtractProblemSolution;                              Solution Extraction
}
```

Coarsely Grained model for parallel EA

Parameters of an Island Model implementation

→Magnitude of migration
→Frequency of migration
→Epoch termination criterion
→Running regime: synchronous v. asynchronous
→Migrant selection strategy
→Number of sub-populations and their sizes
→Individual EAs: operators, selection, survival, fitness

Coarsely Grained model for parallel EA

Magnitude and Frequency of migration

→Limited migration between populations capitalizes on 'punctuated equilibria' effect. For example 2 migrants to each neighboring island every 50 generations.
→More migrants or shorter epochs have the effect of precluding isolated evolution on separate islands. Genetic diversity vanishes quickly and the behavior approximates that of a classic EA running on the whole population.
→Insufficient migration keeps the islands too far apart. The genetic richness of the neighboring populations doesn't have enough chance to spread out. In this regime the parallel run simulates N independent runs with population size N times smaller.

Coarsely Grained model for parallel EA

Epoch Determination

→In most studies the epoch duration was predetermined and kept static throughout the run. However, if the set duration is not long enough to allow the individual sub-populations to achieve equilibrium, the underlying theory is violated. On other hand, if the set duration is too long, computational cycles are wasted crunching a population beyond usefulness.
→A better way is to dynamically determine the end of the epoch at the point when stasis sets in. This could be when the BSF individual in the island doesn't improve in X many generations, or when the whole sub-population degenerates to just one distinct genotype. (This last variant is used in some theoretical works.)
→Epoch length must be long enough to allow exploitation. Given that, variable length determined via equilibrium measures within sub-populations achieve overall results slightly better (though unbalances the parallel implementation and thus may lead to longer running times)

Coarsely Grained model for parallel EA

Synchronous v. asynchronous

→Just like a GA may be generation based or steady-state, the Island Model parallel EA may have migration performed in synch or not.
→For example, Marin *et al* (1994) proposed a centralized asynchronous migration method in which all slave processes would post their best partial results to a master process. Then the master process chooses a BSF overall and broadcasts it to all other processes. Experiments showed near-linear speedup on a network of workstations with few nodes (6). Authors claim that the method scales well because communication is infrequent.
→If the migration is asynchronous that is another hurdle in making duplication difficult.

Coarsely Grained model for parallel EA

Migration Strategy

→Amount of migration must be non-disruptive. Experiments show that about 25% or more is disruptive.
→Elitist or random: experiments show that selecting better (or best) individuals to migrate leads to premature overall stagnation.
→Copy or move: once an individual is selected for migration it may get moved to the target island or may generate a clone to be exported.
→Static or dynamic: the amount of migration can be determined statically by a migration matrix, or dynamic at runtime. An example of dynamic migration would be to send a selected migrant to one of the neighbors chosen randomly. Yet another examples are to send a migrant to that one island whose genetic material is most different (Lin *et al* (1994)) or to that island that has maintained the most diversity (Munetomo *et al* (1993)).
→In the process of migration the local sub-populations may get an increase in size. In order to reduce the size back to normal a process of assimilation takes place. This assimilation bears all the properties of survival selection and could be made part of it.

Coarsely Grained model for parallel EA

Number of sub-populations and sizes

→The size of sub-populations must be above a minimum (critical mass) in order to lead to a viable evolutionary trajectory.
→Cantu-Paz (1999) derives theoretical relations between the number of sub-population and their size to the underlying topology degree of connectivity and migration rate, aiming minimization in execution time.
→In general the number of sub-populations is given by the available hardware. Experiments show however that up to 6 islands good speedup is achieved. Yet, everything else being equal, the more islands there are, the better quality of final solution one should expect.
→The islands may not have to be same sizes. In the canonical implementation M=N x μ, but may very well be that M= $\mu_1 + \mu_2 + \ldots + \mu_N$.

Coarsely Grained model for parallel EA

Individual EAs

→On each island, the sequential EA is independent. It may be the same running on all islands or may be totally different
→If the same EA is running on all islands, it may run on each one of them with different parameters.
→On each island the fitness function is in general the same, but in some cases may not be. Of particular interest here are multi-objective optimization when on each island the fitness assigns different weight coefficients.
→Infusion systems are a particular case of island model when when one island bootstraps the next and so on until the final island actually solves the problem. The migration is unidirectional and the fitness function is increasingly complex.

Coarsely Grained model for parallel EA

Esoteric developments

In canonical Island Model only individuals may migrate. What if we were to allow migration of EA’s parameters too? This may cover co-evolution models due to modification of landscape, as well as meta-EAs.

Coarsely Grained model for parallel EA

Extract Final Solution

→In general the final solution of the problem is given by taking the BSF from all generations and all islands.
→In some special cases the final solution may be restricted to come only from a subset of islands (infusion models).
→Yet the final solution may be a combination of BSF from multiple islands when each island solves only a part of the problem.

Coarsely Grained model for parallel EA

Comparison with a single EA

→Execution time is not a good yardstick. The cost of an algorithm is comprised of fixed and variable cost. The same EA running as a single algorithm may look better or worse (time-wise) then when used in an island model of same overall population size, simply depending on the implementation details.
→In general the cost of communications is ignored. The island model algorithms are coarsely grained and therefore the computation time dominates the communication time.
→A better way to measure performance is to use for example the number of potential solutions generated (evaluations). This focuses the comparisons on the particularities of the EAs and away from low-level implementation details.
→It is also recommended to keep things about equal with regards to other EA parameters that may influence its dynamics. An appropriate single EA to compare with an island model would have a population size $P=\Sigma P_i$, would run for total generations $G= \Sigma G_i$ and would have the number of offspring per generation equal to the overall total number of offspring per generation in the island model

Coarsely Grained model for parallel EA

Presentation of results

→On the horizontal axis should be the amount of computing resources (evaluations)
→On the vertical axis may be one of the following:
- →BSF overall
- →population mean of overall BSF producing island
- →overall mean
- →highest population mean among islands
- →other?

→A good question is to figure out which island gives the overall BSF or best mean in each epoch? If it is the same island over and over again, that one island dominates the run and the other islands are useless.
→When performing statistical analysis, keep in mind that a parallel run only gives one sample point, not N (number of islands). The others are statistically dependent.

Presentation Plan

→History and origins of parallel models for EAs
→Design Issues of a Parallel Model
→Canonical parallel models
- → Master/Slave
- → Finely grained
- → Coarsely grained

→Hierarchical parallel models
→Final considerations
→Questions

Hierarchical Parallel Models

→Hybrid models that mix the canonical models. For example, an implementation may be at a higher level a coarsely grained (island model), with a single population parallel EA at low level.
→Hierarchical models combine the benefits of its components and were found to give better performance then any of them alone.

Presentation Plan

→History and origins of parallel models for EAs
→Design Issues of a Parallel Model
→Canonical parallel models
  → Master/Slave
  → Finely grained
  → Coarsely grained
→Hierarchical parallel models
→Final considerations
→Questions

Final Considerations

→Duplication is very difficult. Due to the nature of parallel implementation, the trajectory followed by each run is determined in part by the operating system scheduling. The amount of work to constrain this unknown in order to be able to duplicate results would probably void the benefits of using a parallel implementation.
→Using Toolkits. It may be a good idea to use ready available toolkits. The implementation of a parallel method is complex enough to warrant employing existing frameworks. It would save time, avoid costly mistakes and give someone else to blame when things go awry.

References

→Adamidis P 1994 *Review of Parallel Genetic Algorithms* Technical Report, Department of Electrical and Computer Engineering, Aristotle University, Thessaloniki
→Anderson E J, Ferris M C 1990 "A genetic algorithm for the assembly line balancing problem,, In Integer Programming and Combinatorial Optimization: Proceedings of a 1990 Conference Held at the University of Waterloo, p 7-18 Waterloo, ON University of Waterloo Press.
→Baluja S 1992 "A massively distributed parallel genetic algorithm (mdpGA),, Tech. Rep. No. CMU-CS-92-196R, Carnegie Mellon University, Pittsburg PA
→Baluja S 1993 "Structure and Performance of Fine-Grain Parallelism in Genetic Search,, In Forrest S, Ed. *Proceedings of the Fifth International Conference on Genetic Algorithms* p 155-162 Morgan Kaufmann (San Mateo CA)
→Bethke A D 1976 *Comparisons of Genetic Algorithms and Gradient-Based Optimizers on Parallel Processors: Efficiency of use of Processing Capacity* Tech. Rep. No. 197, University of Michigan, Logic of Computers Group, Ann Arbor MI
→Branke J, Andersen H C and Schmeck H (1997) *Parallelizing global selection in Evolutionary Algorithms* Submitted to Journal of Parallel and Distributed Computing, January 1997.
→Cantu-Paz E 1998 *A Survey of Parallel Genetic Algorithms* Department of Computer Science and Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign
→Cantu-Paz E 1999 *Topologies, Migration Rates, and Multi-Population Parallel Genetic Algorithms* Illinois Genetic Algorithms Laboratory Report No. 99007, University of Illinois at Urbana-Champaign
→Cohoon J P, Hegde S U, Martin W N and Richards D S 1987 "Punctuated equilibria: a parallel genetic algorithm,, *Proc. 2nd Int. Conf on Genetic Algorithms (Pittsburg, PA, 1987)* ed J J Grefenstette (Hillsdale, NJ: Erlbaum) pp 148-54

References (continued)

→Eldredge N and Gould S J 1972 “Punctuated Equilibria: an alternative to phyletic gradualism„ *Models of Paleobiology* ed T J M Schopf (San Francisco, CA: Freeman, Cooper) pp 82-115
→Gordon V S, Whitley D and Bohn A 1992 “Dataflow parallelism in genetic algorithms„ *Parallel Problem Solving from Nature, 2 (Brussels 1992)* ed Männer R and Manderick B (Amsterdam: Elsevier Science) pp 533-42
→Grosso P 1985 *Computer Simulations of Genetic Adaptation: Parallel Subcomponent Interaction in a Multilocus Model* PhD Thesis, Computer and Communications Sciences Department, University of Michigan
→Holland J H 1975 *Adaptation in Natural and Artificial Systems* Ann Arbor, MI: University of Michigan Press
→Manderick B, Spiessens P 1989 “Fine-grained parallel genetic algorithms„ In Schaffer J D, Ed., *Proceedings of the Third International Conference on Genetic Algorithms*, pp 428-33, Morgan Kaufmann (San Mateo, CA)
→Marin F J, Trelles-Salazar O, Sandoval F 1994 “Genetic algorithms on LAN-message passing architectures using PVM: Application to the routing problem„ In Davidor Y, Schwefel H P, Männer R, Eds. Parallel Problem Solving from Nature PPSN III, p 534-43, Springer-Verlag (Berlin)
→Martin W N, Lienig J and Cohoon J P *Island (Migration) models: evolutionary algorithms based on punctuated equilibria* in Handbook of Evolutionary Computation C6.3
→Pettey C B, Leuze M R and Grefenstette J J 1987 “A parallel genetic algorithm„ *Proc. 2nd Int. Conf on Genetic Algorithms (Pittsburg, PA, 1987)* ed J J Grefenstette (Hillsdale, NJ: Erlbaum) pp 155-61
→Sarma J, De Jong K 1996 “An analysis of the effects of neighborhood size and shape on local selection algorithms„ In *Parallel Problem Solving from Nature IV*, pp 236-44, Springer-Verlag (Berlin)
→Schwehm M 1992 “Implementation of genetic algorithms on various interconnection networks„ In Valero M, Onate E, Jane M, Larriba J L, Suarez B, Eds. Parallel Computing and Transputing Applications, pp 195-203, IOS Press Amsterdam
→Tanese R 1987 “Parallel genetic algorithms for a hypercube„ *Proc. 2nd Int. Conf on Genetic Algorithms (Pittsburg, PA, 1987)* ed J J Grefenstette (Hillsdale, NJ: Erlbaum) pp 177-83

Questions, frustrations hostilities...