Multiplicity Computing and SBSE

University of Málaga, SPAIN

Enrique Alba
# Introduction: SEP’s Modeled as Search Problems

## Index

<table>
<thead>
<tr>
<th>The Problem</th>
<th>Base Algorithms</th>
<th>The Platform</th>
<th>Parallel Solvers</th>
<th>Facts in PS Research</th>
<th>Software and PO</th>
<th>Summary</th>
</tr>
</thead>
</table>

## The Problem Model Phase D/C Search

### The Problem

### Modeling Phase

### D/C Search

### Knowledge

### Quantification

### Solver

---

The Problem: $f(x) = a_0 + \sum_{n=1}^{\infty} \left( a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right)$

$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

\[ \int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi} \]
Objective of a global search problem:

\[ f(\vec{x}) \rightarrow \text{max}: \quad \text{find a vector } \vec{x}^* \]
\[ \quad \text{such that } \forall \vec{x} \in M : f(\vec{x}) \leq f(\vec{x}^*) := f^* \]

- Minimizing is also possible
- Vectors can map to other data structures
Introduction: Taxonomy of Search Algorithms

Optimization Algorithms

- Exact
- Ad-hoc Heuristic
- Metaheuristic

- Calculus
- Enumerative
- Trajectory
- Population

Direct
- Newton
- Greedy

Indirect
- DP
- B&B

SA
VNS
TS
EA
ACO
PSO

*nature inspired in red*
Introduction: Multiplicity Computing

The idea: multiple levels of technology
The resources: computation, communication, storage
The approach: take all of them into account in research
The requirements: be measurable and predictable
The goals: improved scalability, quality, dependability, security

How does this relate to parallel solvers?
Introduction: Parallel Solvers for Complex Problems

- Basic components of parallel Solvers are:
  - **Nodes** performing separate search
  - **Communication pattern** among the nodes
  - **Policy** of the search (start, end, solution…)

**Ex1:** Parallel ACO
- ACO nodes
- All to all
- Exchange pheromone matrix

**Ex2:** Parallel SA
- SA nodes
- Random target
- Exchange actual best solution

**Ex3:** Parallel EA
- Subpops. of partial solutions
- Static ring
- Exchange random selected sols.
Introduction: Parallel Solvers for Search

Centralized PSO in sequential

Centralized PSO in parallel M/S

Centralized PSO in parallel distribution

Decentralized PSO in parallel
Introduction: Taxonomy

• Three features:
  1. Convergence
  2. Synchronization
  3. Homogeneity

• Other important issues are:
  1. Node granularity
  2. Central memory
  3. Static/Dynamic features
  4. ...
Working with parallel solvers is far from trivial:

1. Must know on concurrency and parallel theory
2. Must know on parallel software languages and tools
3. Must know on hardware realizations for communication
4. Must know on communication protocols and networks
5. Programming is more error prone than in sequential
6. A parallel solver is hard to analyze
7. After all the work, maybe you are not gaining in time
8. Many people just don’t want to know on parallel issues
9. ...
...but it is worthwhile!

Problems not solved before become now solvable by using parallel solvers
Facts in parallel search:

1. Model & implementation are different
2. Metrics need a revision
3. Superlinear speedup is a fact
4. Heterogeneity is a must nowadays
5. The experimental setup is important
6. Algorithms are Software
7. Other facts

E. Alba
Parallel Metaheuristics: A New Class of Algorithms

G. Luque, E. Alba
Parallel Genetic Algorithms: Theory and RW Applications
Springer-Verlag, ISBN 978-3-642-22083-8, July 2011
Fact 1: Model and Implementation are Different

Node in a decentralized EA

1. Generate initial population
2. Evaluate present evaluation
3. While not stop criterion do:
   3.1 Select partners
   3.2 Apply variation operators
   3.3 Communication with neighbors
   3.4 Replace old solutions by the new ones
   3.5 Compute statistics and performances
Fact 1: Model and Implementation are Different

Decentralized Model

Centralized Model
Fact 1: Model and Implementation are Different

Decentralized Model

Centralized Model

IMPLEMENTATION-IMPLEMENTATION-IMPLEMENTATION

1 CPU

Manycore Cluster
Fact 2: Metrics Need a Revision

- Exact and approximate algorithms become different algorithms when run in parallel
- Specially, parallel nondeterministic algorithms can stop at solutions of very different quality
- Comparing times against the sequential version could be meaningless if the two algorithms are not exactly the same or the final solution quality is different
- A clear example is the speedup (efficiency)

**A taxonomy is needed**

I. **Strong Speedup**

II. **Weak Speedup**

A. Speedup with solution-stop

1. Versus Panmixia
2. Orthodox

B. Speedup with predefined effort

\[
S_m = \frac{T_1}{T_m}
\]
Fact 2: Advanced Metrics (Entropy)

### Parallel Experiences

- **SPH16-32**
  - 8x64 individuals
  - DPX1($p_c=1.0$)
  - MUT($p_m=1/l$)

### The Problem Base Algorithms

- **Entropy**
  - $H_0,1,2,3,4,5,6,7,8,9,1$
  - mean population entropy (bits)

#### Non Distributed

- ssGA, genGA, cGA(22x23), cGA(4x128)

#### Distributed Sync ($ζ=1$)

- dcGA(4x16), dcGA(8x8), dssGA, dgenGA

### The Platform Parallel Solvers Facts in PS Research Software and PO Summary

- 21/03/2012
Fact 3: Superlinear Speedup is a Fact

\[ S(n_{\text{proc}}) = \frac{T_1}{T^{n_{\text{proc}}}} \]

**weak1:** against panmixia (1 proc)

**weak2:** only changing \( n_{\text{proc}} \)

SPH16-32

SSS128
Fact 4: Heterogeneity is a Must Nowadays

Algorithmic components could be heterogeneous

Parallel hardware could be heterogeneous
Fact 4: Heterogeneity is a Must Nowadays
Fact 5: The Experimental Setup is Important

Parallel heuristic/exact methods have often nondeterministic behaviors, so:

1. Multiple independent runs are necessary
2. Statistical hypothesis tests must be used: Student $t$-test, ANOVA, Wilcoxon, ...
3. Average, max and min values need to be reported, but: is that all? Run times?
4. Advises: more than 30 ind. runs, give all the parameters in one table, always report on times, give hardware and software used, ...
Fact 5: The Experimental Setup is Important: Guidelines

1. **Data sets:** $X_1, X_2, ..., X_n$

2. **Normality test** (Kolmogorov-Smirnov test)
   - **True**
     - **Normal Variables** (Mean comparison, Parametric tests)
     - Levene test and Student t-test
   - **False**
     - **Non-Normal Variables** (Median comparison, Non-Parametric tests)
     - Analysis of variance (ANOVA)
     - Post hoc mean comparison tests
     - Equality of Variance (Levene test)
     - Duncan, Student-Newman-Keuls (SNK), and/or Bonferroni tests
     - Tamhane tests
     - Mann-Witney test
     - Wilcoxon or Sign tests
     - Friedman test
     - Kruskal-Wallis test

**The Problem**

**Base Algorithms**

**The Platform**

**Parallel Solvers**

**Facts in PS Research**

**Software and PO**

**Summary**
Fact 6: Algorithms Are Software

We should worry about the design of algorithms
- At the end, they are software pieces (!)
- Take care of the design and take care of the documentation
- Efficiency is usually the more important issue

Traditional tools
- Fortran, C/C++, Java, Haskell, ...
- Generalization is in conflict with efficiency
- Apply well-accepted recomms. on GOTO, global vars, ...

Object Orientation
- Present best issue (long term development and design)
- Efficiency can be modulated
- Parallel software tools are developed nowadays
- Java versus C++

Fact 6: Algorithms Are Software

Complex Data Structures for Complex Problems

INPUT ENCODING

(a-n)...(m-n)(a-o)...(m-o)(a-p)...(m-p)(n-q)(o-q)...(n-z)(o-z)(p-z)

Input layer

Hidden layer

Output layer

String

Type I

Type II

Type III

Type IV

IF pos IS NL AND vel IS NL THEN for IS PL

Interpretation index

[10, 35, 204, 78, 27, 106]

Present State

Fires

2 active transitions

Present State

[Image of a data flow diagram with nodes and connections indicating complex data structures and algorithms.]

21/03/2012
Relevant Facts for Multiplicity Computing

- **Scalability:** Numerical ............ vs. Physical

  ![Scalability Diagram](image)

- **Asynchronism:**

  ![Asynchronism Diagram](image)

---

21/03/2012
Other Facts

- **Theory is hard but important**: convergence, time complexity, landscape theory, math oracles, …

- **Knowledge exchange between fields helps**: developing common frameworks for grid algorithms, exact plus heuristic issues, …

- **Parallel algorithms are not always better**: communication overhead, numerical search could progress slowly, …

- **Be always ready for new facts!**
There exist multiple software tools to develop parallel applications:

- **Low level tools**: Sockets, ...
- **High Level Comm Libraries**: PVM, MPI, ...
- **Language Embedded**: Java RMI, ...
- **Middleware**: CORBA, MANIFOLD, MR, ..
- **Metacomputing Systems**: Globus, Condor, BOINC, ...
- **Related to Internet**: .NET, SOAP, XML, ...
- **Plateform specific**: CUDA, OpenCL, Handle C, ...
- **Others**: OpenMP, HPF, ..
Parallel Experiences

Index

The Problem

Base Algorithms

The Platform

Parallel Solvers

Facts in PS Research

Software and PO

Summary

Our Software

http://neo.lcc.uma.es/software/index/
Library for complex optimization problems

- 3 types of techniques:
  - Exact
  - Heuristic
  - Hybrid

- 3 implementations:
  - Sequential
  - LAN
  - WAN

Goals:
- Wide genericity but low effort of instantiation
- Simplified utilization (no parallel skills needed)
- Geographically distributed computing systems

Parallel Experiences

Index

The Problem

Base Algorithms

The Platform

Parallel Solvers

Facts in PS Research

Software and PO

Summary

The Problem

Base Algorithms

The Platform

Parallel Solvers

Facts in PS Research

Software and PO

Summary

21/03/2012

R.O.S.

The Platform

Parallel Solvers

Facts in PS Research

Software and PO

Summary

21/03/2012
New Technologies Are Revitalizing this Field

Research in either separate or join levels of parallelism is welcome in companies and journals.

Grid and cloud computing

- GPU
- FPGA
- multicore
Summary of Challenges

1. **Parallel solvers are not just fast versions of sequential solvers: they are new algorithms**

2. **Create standard benchmarking for parallel search**

3. **Develop a uniform experimental setup methodology in which metrics and statistics have a niche**

4. **Connect the solver field to the application field, and to the software domain: metrics for OO, parallelization of compilers, automatic testing, optimized software protocols, scheduling…**

5. **Tackle new issues in cloud computing**

6. **New research fields: GPGPU, Multicores, programming**

7. **Create a body of knowledge in parallel algorithmics independently of their applications**
Parallel Experiences

Index

The Problem
Base Algorithms
The Platform
Parallel Solvers
Facts in PS Research
Software and PO
Summary

End of Presentation

Málaga

21/03/2012