Concurrent Methodologies for Global Optimization

Luca Bortolussi[1]

[1]Department of Mathematics and Computer Science
University of Udine, Italy.

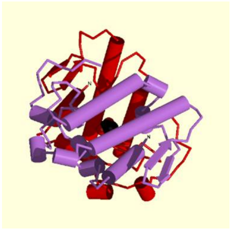
ICLP 2005 Doctoral Consortium, Sitges, 2nd October 2005

Outline

Overview

- Main subject of my research: concurrent optimization systems.
- Staring point: Protein Structure Prediction — concurrent predictor (CCP/Multi-Agent based).
- Need of a general framework. Development of a CCP-based language.
- Now working on analysis techniques.

Some biology: Proteins



- Proteins are fundamental biological molecules, made from aminoacids (primary structure).
- Their functionality derives from their peculiar 3D structure (native or tertiary structure).
- It is supposed to be the state of minimum free energy.

- The protein structure prediction problem is the problem of identifying the 3D fold of a protein, given its primary structure.
- It is an optimization problem.

Designing a Concurrent Optimization Strategy

- Idea: mimic concurrency of natural processes.
- Every aminoacid is an independent process/agent:
  - communicating its position to other agents
  - moving in the space using a Monte Carlo criterion
- The overall heuristic used is a simulated annealing.
- Communications are optimized, focusing on neighbors.

Improving the Framework

- The space exploration is slow. Improved by making big jumps (performed by a dedicated agent).
- External information (e.g. secondary structure predictors) can be exploited through cooperation: agents cooperate to reach suitable configurations (modification of the energy function)

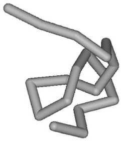
Results up to Now

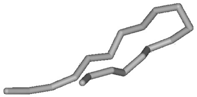
- The energy model used is too coarse, and must be improved.
- Using cooperation, decent prediction can be obtained in reasonable time.

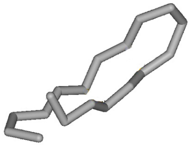
without coop. with coop. from PDB

Results up to Now

- The energy model used is too coarse, and must be improved.
- Using cooperation, decent prediction can be obtained in reasonable time.


without coop.


with coop.


from PDB

The Messy Land of Concurrent Metaheuristics

- There are several strategies (metaheuristics) in literature for parallel/concurrent optimization.
- They are hardly comparable (test depend on the low-level implementation), and they even use different linguistic metaphors and notations.
- We felt the need of a common basis where to describe and analyze such strategies.
- The first step is the identification of a suitable language in which defining them.

Features of the Language

We wanted to model distributed optimization metaheuristics.
We needed:

- constraints $\longrightarrow$ *constraint based language*;
- concurrency $\longrightarrow$ *CCP*;
- distribution $\longrightarrow$ *distributed CCP*;
- probabilities $\longrightarrow$ *probabilistic and distributed CCP*;

Distributing CCP

- CCP is difficult to distribute: communication is performed through globally shared variables.
- Idea: fragmenting the constraint store in independent sets (nodes), and exchanging information between them through *communication channels*.
- In each node, computations evolve like in CCP.
- Communication is synchronous.
- There are several message types: constraint templates, agent templates and channels (which work like in π-calculus).
- There is also the possibility of remotely linking variables belonging to different nodes, so that information flows automatically among them.

Adding Probabilities to CCP

- We add probabilities to the language, in order to reason quantitatively on computations.
- The operational semantic of the language is given by a labeled transition relation, where labels are probabilities associated with transitions.
- Non-deterministic choice, local and global parallel operators are all weighted by a (discrete) probability distribution.
- The model of time is discrete
- Every computational trace has a probability associated to it, and the output of the program is a p.d. over the constraint store.

Extending the Protein Predictor

- Using a more detailed energy model.
- Implementing a true parallel version in MPI.
- Enhancing the exploration of the state space by using more complex internal representation of aminoacids.
- Refining the metaheuristics
- Enhancing the cooperative features.
- Implement it in our language.

Extending the Language

- We have an implementation (meta-interpreter in SICStus prolog), for a subset of the language.
- We plan to extend it to the full language, and writing it in LINDA.
- We want to design a continuous time version of the language, using rates instead of probabilities.
- We want to make the network topology more dynamic.

Analyzing the Programs

- We want to reason on properties like convergence, average execution time, average quality of solutions found.
- We want to compare different methods/heuristics w.r.t. the previous properties.
- We need to explore the transition graph of the model: probabilistic model checking.
- We need to develop a version for CCP end its extensions.
- We can use the language and the analysis tools also for creating an high level framework to model and study biological processes.

The End

THANKS FOR THE ATTENTION!

QUESTIONS?