DistOpt: A Ptolemy-based Tool to Model and Evaluate the Solutions of Optimization Problems in Distributed Environments

Arturo Losi
Università degli Studi di Cassino, Italy
losi@unicas.it

UC Berkeley
Ptolemy Miniconference
March 22-23, 2001

DistOpt and Distributed Optimization
Cassino, Italy
DistOpt and Distributed Optimization

Contents

- Credits
- Introduction and motivation
- Optimization problem and Auxiliary Problem Principle
- DistOpt structure
- Using DistOpt

---

DistOpt and Distributed Optimization

Credits

- Publications related to DistOpt
  - A. Losi, and M. Russo, *A Note on the Application of the Auxiliary Problem Principle*, to be re-submitted (after the first review) to the Journal of Optimization Theory and Applications
DistOpt and Distributed Optimization
Introduction and motivation

- Coarse-grain algorithms, with focus on the structure of the optimization problem (not on the structure of the technique)
- Splitting of the optimization problem into subproblems, with grains corresponding to subproblems
- Decomposition/coordination approach
  - Suitable formulation of the problem
  - Decomposition of the optimization problem into smaller subproblems, iteratively solved
  - Coordination of the subproblems to drive their solutions to a solution of the original problem

DistOpt and Distributed Optimization
Introduction and motivation

- Design variables
  - decomposition/coordination method
  - splitting of the problem into subproblems
  - technique for solving the subproblems
  - synchronization
  - computing capacity of the processors
  - characteristics of the communications (other packages)
- DistOpt helps modeling and evaluating coarse-grain distributed optimization algorithms, for wide classes of optimization problems and decomposition/coordination methods
DistOpt and Distributed Optimization

Optimization problem

\[
\min_{x \in X} f(x)
\]

\[
X \text{ (feasibility set)}
\]

\[U_1 \quad U_2\]

\[
\begin{align*}
\min_{u_1 \in U_1, u_n \in U_n} f(u_1, \ldots, u_n), \\
\text{s.t. } y_{i,j} - y_{j,i} &= 0, \\
&\quad i,j = 1, \ldots, n \text{ for } j \neq i
\end{align*}
\]

with \(u_1 = \{x_{1,1,2}, \ldots, y_{1,n}\}\)

\(u_n = \{x_{n,1,2}, \ldots, y_{n,n-1}\}\)

DistOpt and Distributed Optimization

Auxiliary Problem Principle

- The Auxiliary Problem Principle (APP - by Guy Cohen, France) is a general decomposition/coordination theory that:
  - does not require separability assumptions (such as additive cost or additive constraints)
  - is a generalization of many known methods
  - encompasses both one-level and two-level methods
  - for the two-level methods, the convergence conditions are quite general

- DistOpt is based on the APP’s two-level methods
DistOpt and Distributed Optimization

**Auxiliary Problem Principle**

\[
\begin{align*}
\min_{u \in U} & \quad f(u_1, \ldots, u_n), \\
\text{subject to} & \quad y_{i,j} - y_{j,i} = 0, \quad i, j = 1, \ldots, n, j \neq i.
\end{align*}
\]

Iterative scheme (slightly modified)

\[
\begin{align*}
& \min_{u_i \in U_i} \left\{ \mathcal{K}_i(u_i) + f_i(u_{k_i}^{n_i}, \ldots, u_{k_i}^i) - \mathcal{K}'_{y_i}(u_i^k, u_i) \right\} \\
& \quad + \varepsilon \sum_{j=1}^n \left( p_{i,j}^k + c(y_{i,j}^{k} - y_{j,i}^{k}) \right), \quad i = 1, \ldots, n.
\end{align*}
\]

\[
p_{i,j}^{k+1} = p_{i,j}^k + \rho (y_{i,j}^{k+1} - y_{j,i}^{k+1}), \quad \text{for } j = 1, \ldots, n, j \neq i.
\]

**DistOpt and Distributed Optimization**

**Auxiliary Problem Principle**

- **Global convergence conditions**
  - Convex (and differentiable) objective function, \( f \)
  - Closed convex feasibility sets, \( U_i \) (\( i = 1, \ldots, n \))
  - Convex coupling constraints (affine in the equality case)
  - Strongly convex core function, \( K \)
  - Defined bounds on the parameters \( c, \rho, \varepsilon \)

- In practical non-convex cases, with APP iterative scheme optimality necessary conditions are generally met in the limit, if convergence results.
DistOpt and Distributed Optimization

DistOpt structure

- Modular, flexible, extendible and conducive to cooperation
  - OOP
- Ptolemy (classic)
  - Discrete-Event (DE) domain
- Three levels of abstraction
  - definition of the problem and its splitting into subproblems
  - formulation of the transformed problem and of the subproblems, and set-up of the two-level algorithm
  - solution of the optimization subproblems

DistOpt and Distributed Optimization

DistOpt structure

- First level: Definition of the problem and its splitting
  - description of the optimization problem (objective function and constraints), and information on the splitting into subproblems - provided by the user
  - no constraints on the coding of the problem (data structure and functions internal to the application)
  - formal definition of the methods
  - defined data structure for the data exchange - through a file
DistOpt and Distributed Optimization

**DistOpt structure**

- Second level: formulation of transformed problem, subproblems, two-level decomposition/coordination algorithm
  - problem transformation, with variable duplication
  - subproblem formulation and coordination
  - set-up of APP parameters (core function $K$, parameters $c$, $\epsilon$, $\rho$)
  - choice of synchronous/asynchronous execution of subproblems

---

**DistOpt and Distributed Optimization**

*DistOpt structure*

- Second level: algorithm building palette

```
SUB_1

SUB_2

SUB_3

the next icon goes here
```
**DistOpt and Distributed Optimization**

**DistOpt structure**

- **Second level**: block SUB_3

- **Third level**: solution of the subproblems
  - choice of the solver for each subproblem

DistOpt and Distributed Optimization
Using DistOpt

A DistOpt universe

Set-up of the computation
DistOpt and Distributed Optimization Using DistOpt

- Set-up of the computation

- Choice of the core function K
DistOpt and Distributed Optimization
Using DistOpt

Choice of convergence parameters

Some results
synchronous  asynchronous
DistOpt and Distributed Optimization

- ftp://sistelet.ing.unicas.it/DistOpt
  - readme.txt
  - userguide.pdf
  - DistOpt.tar.gz