Optimization in the Loop
Implementing and Testing Scheduling Algorithms with SimuLTE

Antonio Virdis
University of Pisa

• Prof. Giovanni Stea
• Giovanni Nardini
Outline

• Why Optimization
• Going into the Loop
• Methods
• Example
An everyday problem
Comparing Results

system

Algorithm 1

Algorithm 2

I’m better than you
Comparing with the best

Some Important KPI

0 20 40 60 80 100 120 140
Comparing with the best
Comparing Results

I’m better than the optimum

max \quad x_i

s.t.

x_i + p_i \leq M

...
A simple problem
Taking a Photo

Mathematical Formulation

resources
buffers

scheduler
HELP!
output speed

\[ s_1 > s_2 \]
Finite Buffer: CBR
How does the system evolve

resources

buffers

scheduler

HELP!

Decision
From outside to inside

- Resources
- Buffers
- HELP!
- Decision
- Scheduler

14
Going Into The Loop
Overview

1. Read system info
2. Build problem instance
3. Solving
4. Parse and enforce solution
2 methods

LP File

XML File

Simulator

CPLEX

CPLEX

Simulator

OMNeT++
Building A problem File

LP file

\[ \text{max } \sum_{i} x_i \]

\[ \text{s.t. } \sum_{i} x_i + p_i \leq M \]

\[ \text{for } (i = 0 ; i < N ; ++i ) \]
\[ \text{stream << "x" << i << " + ";} \]

\[ \text{for } (i = 0 ; i < N ; ++i ) \]
\[ \text{...............} \]
– XML Management
  • Built-in in OMNeT
  • Easy to implement manually

```xml
<variable name="x0" index="0" value="51"/>
<variable name="x1" index="1" value="0"/>
<variable name="x2" index="2" value="141"/>
<variable name="x3" index="3" value="0"/>
<variable name="p0" index="4" value="141"/>
<variable name="p1" index="5" value="0"/>
```
2° method: API

• Idea: can we use CPLEX as an API?

  – **Callable Library**: matrix-based C-written API

  – **Concert Tecnology**: a set of modeling objects (also) in C++
Including CPLEX

• **TELL OMNET:**
  – where the .h files are located
  – where the dynamic libraries are located
  – which dynamic library to include
  – enable the I_STD preprocessor macro

• Can be done via the *Project Properties* of OMNeT++
Matrix representation

Objective function

Constraints

ALL variables
Matrix representation
• Generally variables are in the form:

\[-X_i\]  
\[-y_{i,j}\]

One pedix

Two pedices

\[y_{i,j}\]  
\[X_i\]  
\[0 \quad i < M\]  
\[0 \quad j < K\]  
\[M \quad K\]
2° Method: variables

Name, #1st, #2nd

Access with local indexes
2° Method: constraints

• Add constraints one by one using local indexes

• Build the problem at the end one-shot
Reading The Output

XML

```
<variable name="x0" index="0" value="51"/>
<variable name="x1" index="1" value="0"/>
<variable name="x2" index="2" value="141"/>
<variable name="x3" index="3" value="0"/>
<variable name="p0" index="4" value="141"/>
<variable name="p1" index="5" value="0"/>
```

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>.......</th>
</tr>
</thead>
<tbody>
<tr>
<td>solution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Quiz 2:

$x_i \in \{0, 1\}$  

Binary values

?  

$x[i] == 1$
## Quiz 2:

<table>
<thead>
<tr>
<th>$x_0$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$x_7$</th>
<th>$x_8$</th>
<th>$x_9$</th>
<th>$x_{10}$</th>
<th>$x_{11}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

### Do not trust double values

$x_8 \rightarrow 1.0000000001$

$x_9 \rightarrow 1.0000000000$

$x_{10} \rightarrow 1.0000000000$

$x_{11} \rightarrow 0.0000000000$
Pros and Cons

• Easy to build
• Generally slower

• Generally faster
• Requires API knowledge
Optimization in SimuLTE
LTE

UE1

1ms

UE2

CQI

90 bytes

CQI

55 bytes

RBs

Buffers

UE1

UE2

eNB

90 90

55 55 55

32
Resource allocation in LTE

Allocate RBs to UEs
Multi Band Scheduling

CQI1 [0]
CQI1 [1]
CQI1 [2]
...
CQI2 [0]
CQI2 [1]
CQI2 [2]
...

RBs
SimuLTE: Scheduling structure
SimuLTE: Scheduler Hierarchy

- **Allocator**
- **eNB Scheduler**
- **Data Manager**

Scheduling Policy:
- **MAX C/I**
- **PF**

Arrows indicate the flow of information or control between the components.
Simulation Scenario

- Varying packet size
- Linear Mobility
- InLoop vs OutLoop
InLoop vs OutLoop

MAC Throughput [byte/TTI]

Algorithm | In Loop | Out Loop

Packet Size

250 | 500 | 750 | 1000
InLoop vs OutLoop
Any question while running for dinner?

Antonio Virdis
a.virdis@iet.unipi.it
simulte.com