A functional programming approach for an energy planning problem

Julien Darlay Louis Esperet Yann Kieffer Guyslain Naves
Valentin Weber

Laboratoire G-SCOP
Grenoble, France

July 14^{th} 2010

"A functional programming approach to the challenge??"

Some Q & As

- What does it mean?
  Tackle the challenge using a different kind of programming language
- What is functional programming?
  A programming paradigm with equal status for functions and data
- Where does it come from?
  Artificial Intelligence: LISP, Scheme, ...

So what's the deal?

- Try it (chosen language: Ocaml)
- Examine the pros/cons for this experiment

G-SCOP
SCIENCES POUR LA CONCEPTION, L'OPTIMISATION ET LA PRODUCTION

What are the benefits of using functional languages?

Benefits

- High flexibility together with strong typing
- Code is more conceptual
- Code is more compact
- Code can better reflect mathematical thinking
  $\longrightarrow$ thus programming language should be less of an obstacle (as compared to C++ and other imperative languages)

G-SCOP
SCIENCES POUR LA CONCEPTION, L'OPTIMISATION ET LA PRODUCTION

More details about the benefits

Easy-to-use and easy-to-create types

- Types do not have to be declared to be used - ex.: (2, "abc")
- Genericity (type variables)

Higher-level abstraction

- Functions are first-class types
- Modularity through functional interfaces
- Code reuse not limited to low-level stuff

G-SCOP
SCIENCES POUR LA CONCEPTION, L'OPTIMISATION ET LA PRODUCTION

Examples

Functional interface for data

- All arrays presented as functions indexed by integer arguments
- *outages*: integer function of 2 parameters: int -> int -> int
- *pp1_production*: real fct of 3 param.: int -> int -> int -> float

Example of a reusable routine: sum_float_function_range

- Usage: computes the sum of the elements of the array; call: `sum_float_function_range array low_index high_index`
- Type: (int -> float) -> int -> int -> float

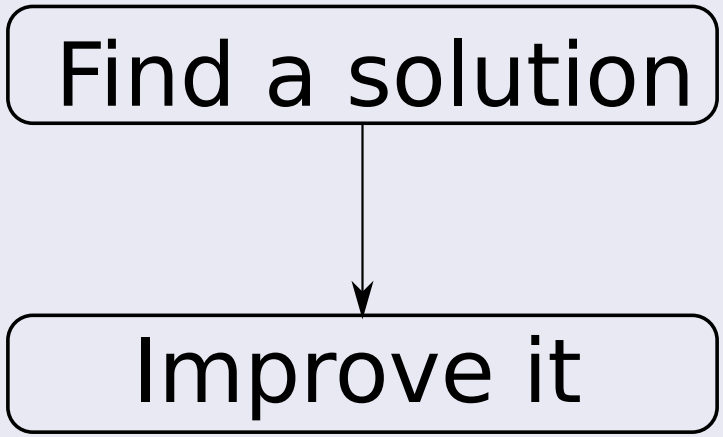
Example of a generic utility routine: memo

- Usage: add a cache system to a function; call: `memo function`
- Type: ('a -> 'b) -> ('a -> 'b)

SCIENCES POUR LA CONCEPTION, L'OPTIMISATION ET LA PRODUCTION

So what did you end up doing?

A two-step approach

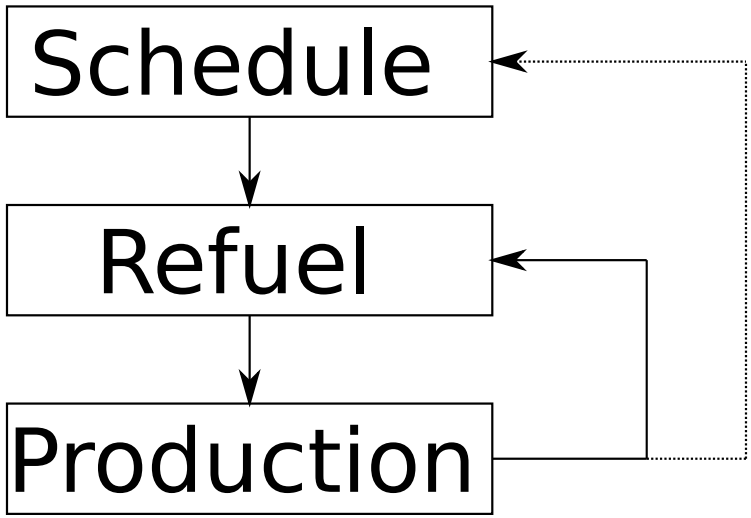


A constraint checker

- Spacing / Overlapping (constraints 14 to 18)
- Simultaneous outages (constraints 19 to 21)

SCIENCES POUR LA CONCEPTION, L'OPTIMISATION ET LA PRODUCTION

Finding a first solution



Finding a schedule

- Divide the horizon in intervals
- For each outage in each interval
  - Find a starting date
  - Check the partial solution
  - Backtrack if no solution
- Fix the optional outages with a greedy procedure

When choosing an outage date

- Avoid the weeks of high demand
- Allow enough time to consume the fuel
- Reduce the search space of remaining outages

Finding a first solution
A production plan

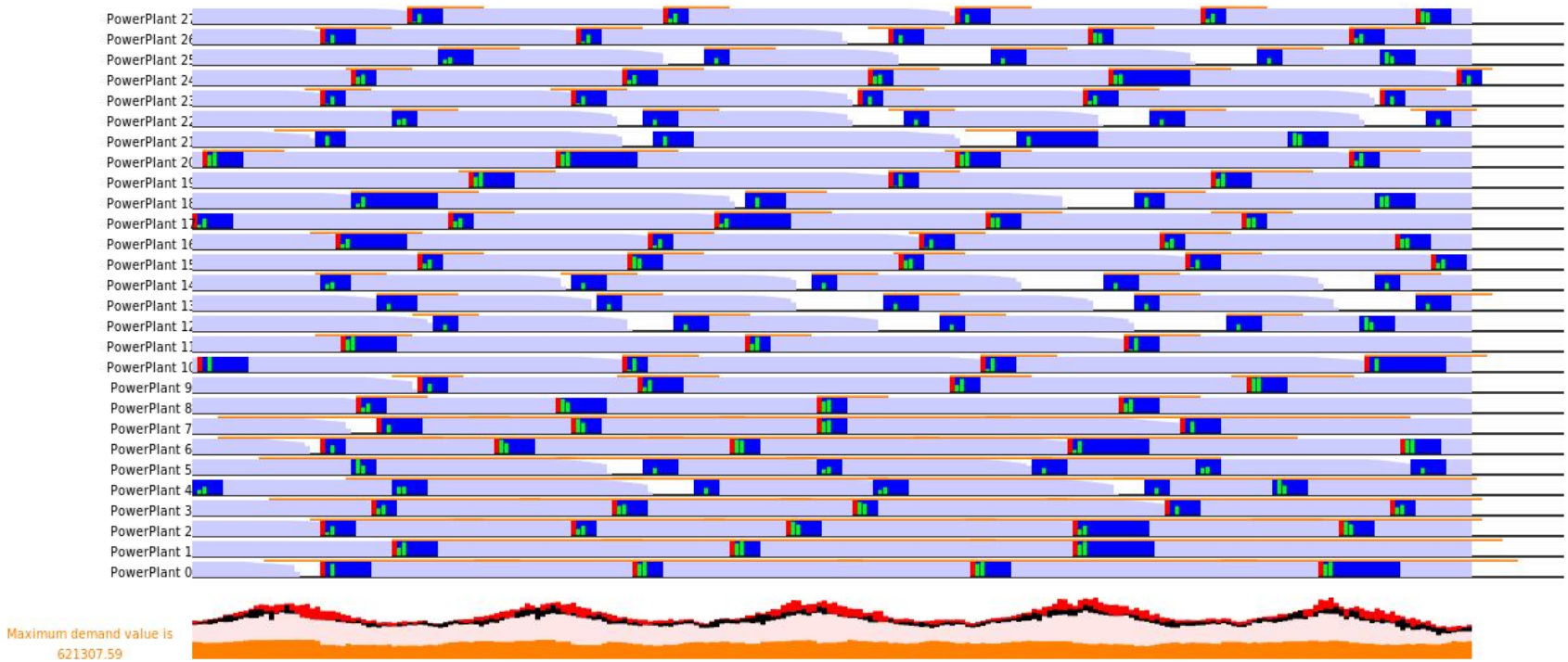
Finding a production plan

- Refuel with the minimum amount
- Greedy production plan for type 2
- Modulation to avoid overproduction
- Complete the remaining demand with type 1

No production plan $\Rightarrow$ Add more space between outages during the previous step

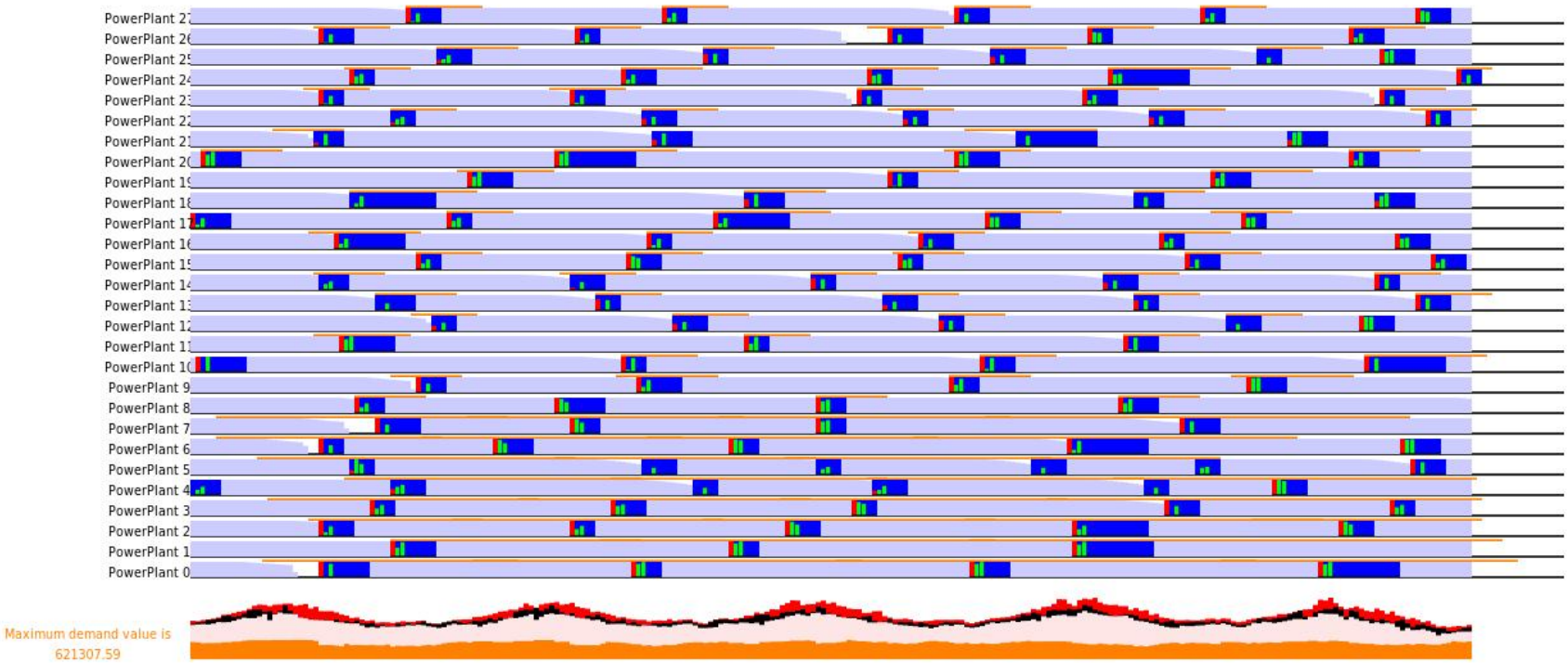
G-SCOP
SCIENCES POUR LA CONCEPTION, L'OPTIMISATION ET LA PRODUCTION

A first solution



G-SCOP
SCIENCES POUR LA CONCEPTION, L'OPTIMISATION ET LA PRODUCTION

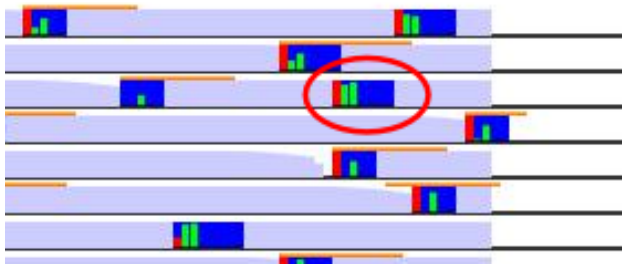
Lets add more fuel



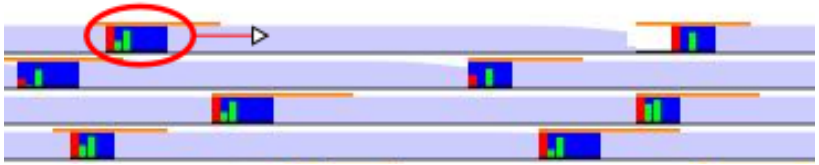
G-SCOP
SCIENCES POUR LA CONCEPTION, L'OPTIMISATION ET LA PRODUCTION

How to improve the solution ?

- Remove the unnecessary outages



- Delay the outages



$\Rightarrow$ Simple local modifications

G-SCOP
SCIENCES POUR LA CONCEPTION, L'OPTIMISATION ET LA PRODUCTION

Pros:

- Great expressivity
- Fast prototyping
- Fast compiled code
- Interactive mode

Cons:

- Lack of bindings with classical OR libraries
- Purely functional data structures can be memory consuming

G-SCOP
SCIENCES POUR LA CONCEPTION, L'OPTIMISATION ET LA PRODUCTION

Conclusion
Optimization

- Two-step approach
- CSP-like algorithm for outage planification
- Greedy procedure for production
- Fast algorithm ($<$ 20 min)
- Run with less than 3 GB of RAM

Perpectives:

- More complex improvements
- Remove inactivity weeks
- Linear programming approach to the production plan

G-SCOP
SCIENCES POUR LA CONCEPTION, L'OPTIMISATION ET LA PRODUCTION