Bayesian Optimization Approach of General Bi-level Problems

SAEOPT workshop

Emmanuel Kieffer

Contents:

Context

- Problems with two decision makers playing iteratively
- Each decision maker controls a part of the decision
- Closely related to Game Theory
- First modelled as Games, i.e. Stackelberg Games (1952)
- Non-cooperative and hierarchical games
- First player is called the « Leader »
- Second player is the « Follower »

From Stack. to Bi-level

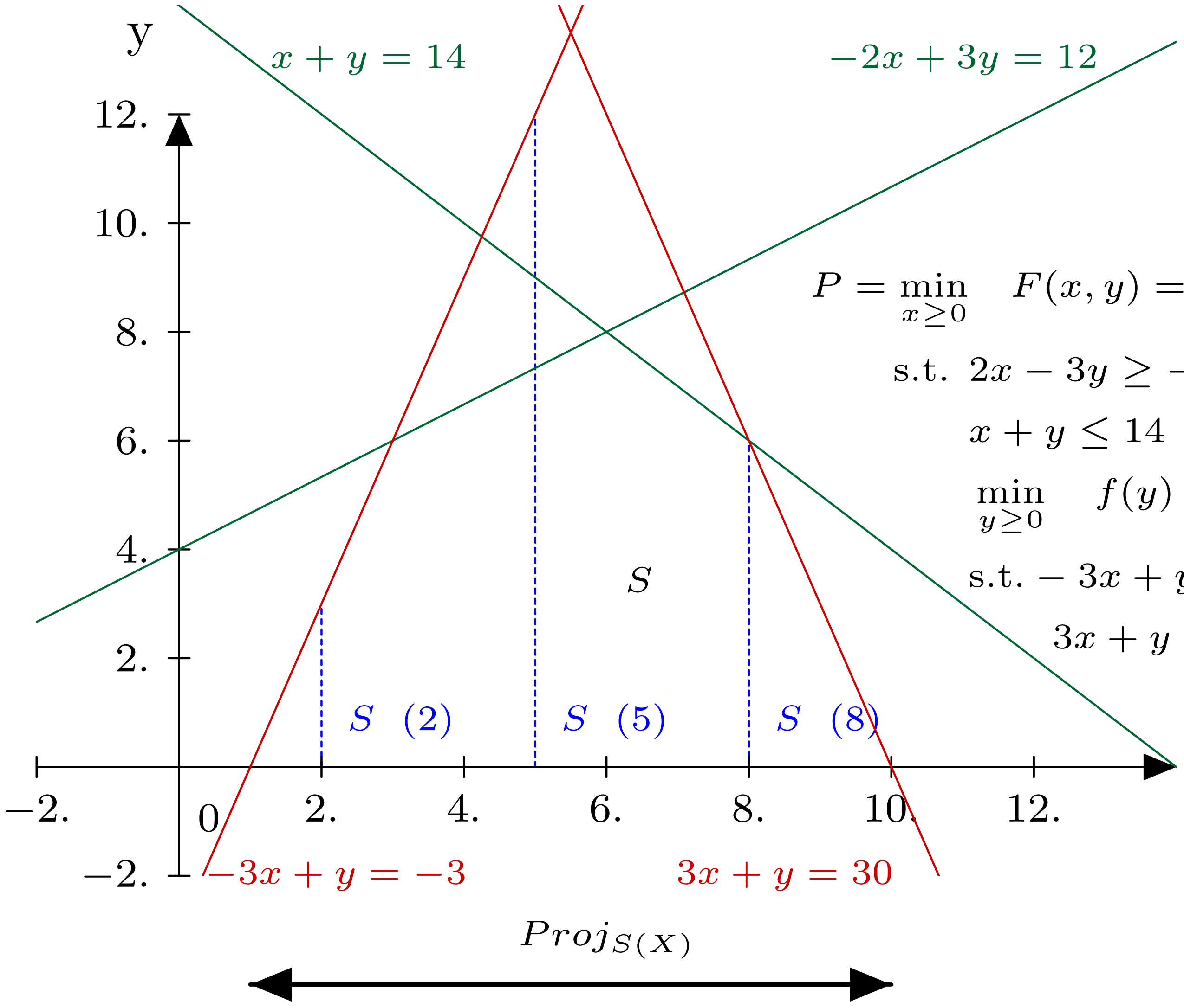
- Bi-level problems generalize Stackelberg Games

- Possible constraints at both levels

$$
\begin{aligned}
\min \quad & F(x,y) \\
\text{s.t.} \quad & G(x,y) \leq 0 \\
& \min \quad f(x,y) \\
& \text{s.t.} \quad g(x,y) \leq 0 \\
& x, y \geq 0
\end{aligned}
$$

where F, f : $\mathbf{R}^n \times \mathbf{R}^m \rightarrow \mathbf{R}, G : \mathbf{R}^n \times \mathbf{R}^m \rightarrow \mathbf{R}^p$ and $\mathbf{R}^n \times \mathbf{R}^m \rightarrow \mathbf{R}^q$.

Example of Bi-level program



$$
\begin{aligned}
P = \min_{x \geq 0} \quad & F(x,y) = -x - 2y \\
\text{s.t.}\ \ & 2x - 3y \geq -12 \\
& x + y \leq 14 \\
& \min_{y \geq 0} \quad f(y) = -y \\
& \text{s.t.} -3x + y \leq -3 \\
& 3x + y \leq 30
\end{aligned}
$$

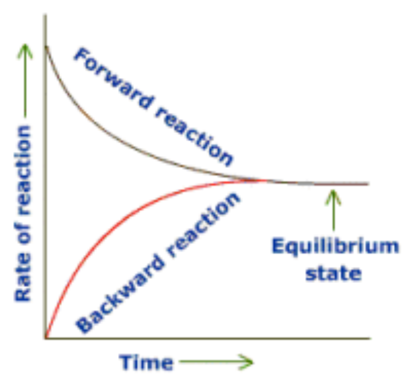
Definitions

- The constraint region: $S = \{(x, y) : x \in X, y \in Y, G(x, y) \leq 0, g(x, y) \leq 0\}$
- The feasible set for the follower parametrized by $x \in X$: $S(x) = \{y \in Y : g(x, y) \leq 0\}$
- The projection of S onto the leader's decision space: $S(X) = \{x \in X : \exists y \in Y, G(x, y) \leq 0, g(x, y) \leq 0\}$
- The *Follower's* rational decision set for $x \in S(X)$:$P(x) = \{y \in Y : y \in \arg\min[f(x, \hat{y}) : \hat{y} \in S(x)]\}$
- The Inducible Region: $IR = \{(x, y) \in S, y \in P(x)\}$

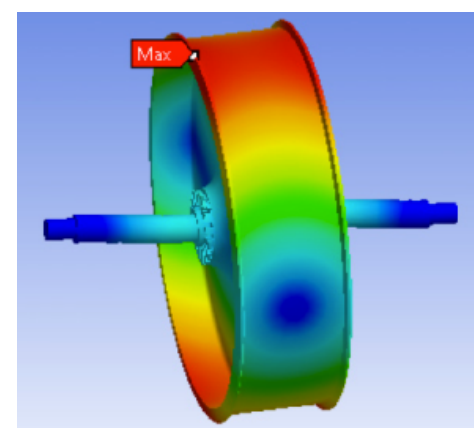
Practical Bi-level Problems

(source: http://www.bilevel.org)

- Applications in :
  - Transportation: Toll Setting Problem (Brotcorne et al. 2001)
  - Chemistry: Optima Chemical Equilibria Problem (Dempe 2002)
  - Physics of materials: Structural optimization (Christiansen et al. 2001)
- The last two examples are not related to Game Theory
- Optimization problems with equilibrium constraints (MPEC)



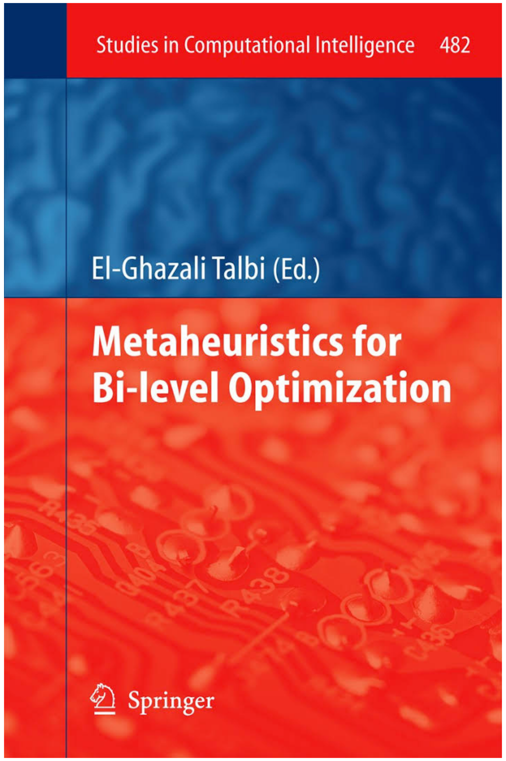
Attainment of equilibrium in a reversible reaction



Resolution approaches

UNIVERSITÉ DU LUXEMBOURG

- Bi-level problems are NP-hard problems even for two convex levels
- Tacked using Bi-level Metaheuristics
  - (see Metaheuristics for Bi-level Optimization)
- 4 categories of metaheuristics
  1. Nested sequential. (NSQ)
     - Repairing approach. (REP)
     - Constructive approach. (CST)
  2. Single-level transformation. (STA)
  3. Co-evolutionary. (COE)
  4. Multi-objective (MOA)
- Promising metaheuristics are now based on Surrogate-based optimization



Surrogate-based approach

- Bi-level problems are nested optimization problems
- A problem is constrained by another one
- Feasibility is achieved only if the inner-problem is solved to optimality
- (A) Hard to compare two resulting solutions which are not Bi-level feasible
- (B) It is also time consuming (not suitable) to solve every time to optimality the inner problem
  - Ex: Genetic algorithm – evaluation a population of solutions
- One way is to predict the value of the inner problem using prior knowledge --> solve at least (B)

Proposed resolution

- Assumptions:
  - We consider the optimistic approach if P(x) is not a singleton
  - To avoid issue (A), we select official benchmarks
- Advantages:
  - By considering the optimistic case, we can guarantee that the problem has an optimal solution
  - With benchmarks, we can compare solution of different algorithms
- Drawbacks:
  - Benchmarks are far from practical problems
  - Pessimistic case is generally more realistic
- Surrogate-based optimization exist for Bi-level Optimization
- However Bayesian Optimization (BO) seems to be new for Bi-level optimization
- <u>Question</u>: Is BO a relevant and efficient approach according to assumptions?

Bayesian Optimization

Algorithm 1 Bayesian Optimization

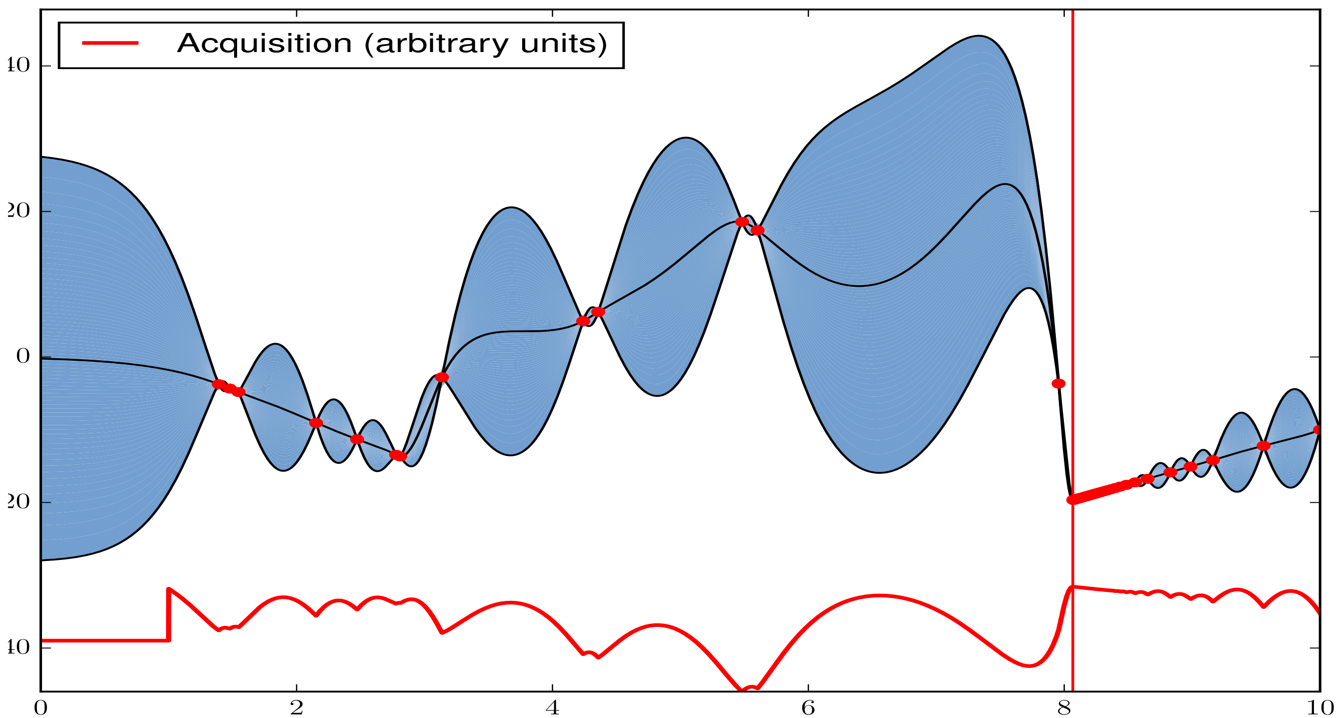
function SOLVE(problem,n,k)
    X =initRandom(n);
    Y =problem.evaluate(X)
    model=$\mathcal{GP}(X, Y)$
    model.update()
    while not has_converged() **do**
        acq = getAcquisition(k);
        x_{new} =acq.optimize();
        y_{new} =problem.evaluate(x_{new});
        model.update(x_{new},y_{new});
    end whilereturn model.best;
end function

- Based on Gaussian processes – Distribution over functions
- This distribution provides knowledge on the location of the optimal solution
- The distribution is updated according to an acquisition function

Bayesian Optimization (BO)

UNIVERSITÉ DU LUXEMBOURG

- Well-known acquisition functions:
  - Maximum Probability of Improvement (MPI) $\mathrm{acq}_{MPI}(x) = \Phi(\gamma(x))$
  - Expected Improvement (EI) $\mathrm{acq}_{EI}(x) = \sigma x(\gamma(x)\,\Phi(\gamma(x)) + \phi(\gamma(x)))$
  - Lower-Confidence Bounds (LCB) $\mathrm{acq}_{LCB}(x) = \mu(x) - k\sigma(x)$



BO on Bi-level problems

x parametrizes the inner problem

F(x,y) can be computed

$$\begin{aligned} \min \quad & F(x,y) \\ \text{s.t.} \quad & G(x,y) \leq 0 \end{aligned}$$

$$\boxed{\begin{aligned} \min \quad & f(x,y) \\ \text{s.t.} \quad & g(x,y) \leq 0 \\ & x, y \geq 0 \end{aligned}}$$

Need to determine $\hat{y}$
(opt. sol. of inner problem)
Indeed, here x is a parameter

$\hat{y}$ is the response

- Try to avoid as much as possible explicit computation of $\hat{y}$
- To be Bi-level feasible, we have $F(x, \hat{y} = \text{argmax}\{F(x,y) : y \in P(x)\}$
- One-to-one correspondence between x and $\hat{y}$
- Recall that P(x) is either a singleton or we apply optimistic case

BO on Bi-level problems

- Therefore, we can write $F'(x) = F(x, \hat{y} = \text{argmax}\{F(x,y) : y \in P(x)\}$
- We will create a surrogate model to learn this new function
- Of course, the initialization still requires to approximate $\hat{y}$ (time-consuming)
- Initial sample generated with "latin-hypercube" approach
- Acquisition function solved by Differential Evolution Algorithm
- The evaluation algorithm of $\hat{y}$ using local search:
  - Why ? To save time
  - How ? Using Sequential Least Squares Programming algorithm (SLSQP)
  - When ? Only at initialization and after acquisition
- SLSQP requires a first guess y^0
  - Use some kind of re-optimization
  - With Lipschitz continuity assumption, if two solutions x_1 and x_2 are close to each other then $\hat{y}_1$ and $\hat{y}_2$ should be close to each other too

Experiments on Benchmarks

- Where ? A. Sinha, P. Malo, K .Deb “Test Problem Construction for Single-Objective Bilevel Optimization “
- On 10 benchmarks ------------------------------>
- Compared to BLEAQ algorithm
- Based on approximation of $\hat{y}$ (and not F(x,y))
- One of the most efficient algorithm in the literature
- Tests performed on UL HPC platform --- 30 runs/benchmark
- On single core of an Intel Xeon E3-1284L v3 @ 1,8 GHz, 32Gb of RAM server
- Code implemented in Python using GPyOpt library for Bayesian Optimization

Best known fitnesses	UL fitnesses	LL fitnesses
TP1	225.0	100.0
TP2	0.0	100.0
TP3	-18.6787	-1.0156
TP4	-29.2	3.2
TP5	-3.6	-2.0
TP6	-1.2091	7.6145
TP7	-1.96	1.96
TP8	0.0	100.0
TP9	0.0	1.0
TP10	0.0	1.0

Numerical results

- Fitnesses for both levels have been recorded
- Wilcoxon rank-sum test applied to obtain good statistical confidence

	Bayesian		BLEAQ	
Best values	ULfitness	LLfitness	ULfitness	LLfitness
TP1	225.0011	99.9984	225.0	100.0
TP2	0.0	200.0	5.4204	0.0
TP3	−18.6786	−1.0156	−18.6787	−1.0156
TP4	−29.1991	3.2001	−29.2	3.2
TP5	−3.8998	−2.0039	−2.4828	−7.705
TP6	−1.2099	7.6173	−1.2099	7.6173
TP7	−1.6833	1.6833	−1.8913	1.8913
TP8	0.0	200.0	12.2529	0.0007
TP9	0.0007	1.0	3.5373	1.0
TP10	0.0011	1.0	0.001	1.0

- Good accuracy for BO
- Ability of the BO to face multi-modal problems
- TP2 and TP8 also have (x=0 and y=200) or (x=0 and y= 100) as optimal solutions

Numerical results

- Number of function evaluation for both levels have been recorded as well

	Bayesian		BLEAQ	
Average	ULcalls	LLcalls	ULcalls	LLcalls
TP1	211.1333	1558.8667	588.6129	1543.6129
TP2	35.2581	383.0645	366.8387	1396.1935
TP3	89.6774	1128.7097	290.6452	973.0
TP4	16.9677	334.6774	560.6452	2937.3871
TP5	57.2258	319.7742	403.6452	1605.9355
TP6	12.1935	182.3871	555.3226	1689.5484
TP7	72.9615	320.2308	494.6129	26 682.4194
TP8	37.7097	413.7742	372.3226	1418.1935
TP9	16.6875	396.3125	1512.5161	141 303.7097
TP10	21.3226	974.0	1847.1	245 157.9

- BLEAQ needs more UL and LL evaluations than BO for nearly the same accuracy
- In addition BLEAQ solves the inner problem with a Genetic algorithm
- BO uses a simple local search with re-optimization

Conclusion / Future works

- Conclusion on Benchmarks:
  - We applied Bayesian Optimization to Bi-level Problems
  - Provide accurate results in term of fitness while reducing the number of evaluation
- Now, we would like to apply it on real problems
- BUT: Gaussian Processes have some limitations
- Test other models (SVM, RBF network, …)
- Use Dynamic Evolutionary Optimization instead of calling iteratively the Differential Evolutionary algorithm

Thank you

UNIVERSITÉ DU LUXEMBOURG

Questions ?