Optimality for Tough Combinatorial Hydro Valley Problems

Claudia D'Ambrosio

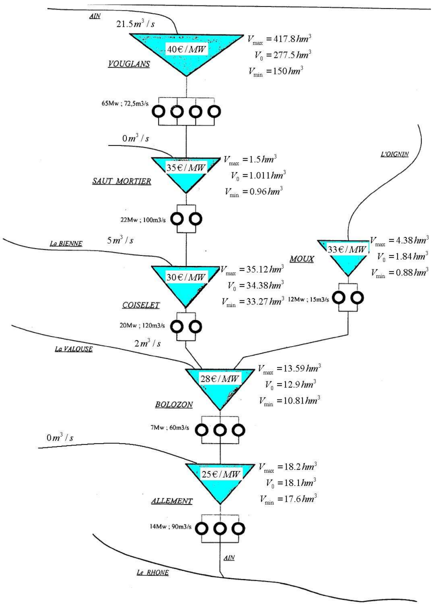
CNRS LIX, Ecole Polytechnique, France

PGMO seminars 2012, November 20th, 2012

PGMO project and PhD thesis

- Crucial problem in energy management: **hydro valley management**.
- Combinatorial elements leads to far tougher hydro valley problems.
- French Hydro valleys (see last hour).

Example



AIN
21.5 m³ / s
40€ / MW
VOUGLANS
Vmax = 417.8 hm³
V0 = 277.5 hm³
Vmin = 150 hm³
65Mw ; 72,5m3/s
0 m³ / s
35€ / MW
SAUT MORTIER
Vmax = 1.5 hm³
V0 = 1.011 hm³
Vmin = 0.96 hm³
22Mw ; 100m3/s
L'OIGNIN
La BIENNE
5 m³ / s
33€ / MW
MOUX
Vmax = 4.38 hm³
V0 = 1.84 hm³
Vmin = 0.88 hm³
30€ / MW
COISELET
Vmax = 35.12 hm³
V0 = 34.38 hm³
Vmin = 33.27 hm³
12Mw ; 15m3/s
20Mw ; 120m3/s
2 m³ / s
La VALOUSE
28€ / MW
BOLOZON
Vmax = 13.59 hm³
V0 = 12.9 hm³
Vmin = 10.81 hm³
7Mw ; 60m3/s
0 m³ / s
25€ / MW
ALLEMENT
Vmax = 18.2 hm³
V0 = 18.1 hm³
Vmin = 17.6 hm³
14Mw ; 90m3/s
AIN
Le RHONE

Notation

Sets:

$(\mathcal{N}, \mathcal{V})$ = directed graph representing the hydro valley.

$\mathcal{T}$ = set of time steps.

$\mathcal{U}$ = set of turbines.

$\mathcal{P}$ = set of pumping stations.

$\mathcal{S}$ = set of active power and spinning reserves.

$\mathcal{F}_n = \{m \in \mathcal{N} | a_{mn} = 1\}$ $(n \in \mathcal{N})$.

$\mathcal{D}_n = \{m \in \mathcal{N} | a_{nm} = 1\}$ $(n \in \mathcal{N})$.

$\mathcal{X}_{ut}$ = set of operational levels for turbine u at time period t $(u \in \mathcal{U}, t \in \mathcal{T})$

Notation

Parameters:

A_{nm} = connection matrix element (it is 1 when the water released from reservoir n flows into reservoir m, 0 otherwise).

T = time step size [hours].

D_v = the number of time steps the water takes to flow (get pumped) through arc v ($v \in \mathcal{V}$).

I_{nt} = inflows at period t to reservoir n ($t \in \mathcal{T}, n \in \mathcal{N}$) [m^3/h]

G = gradient slopes [m^3/h^2]

X_{uti} = i-th operational level for turbine u at time period t ($u \in \mathcal{U}, t \in \mathcal{T}, i \in \mathcal{X}_{ut}$) [m^3/h]

$\overline{X}_u$ = upper bound on the water flow passing through turbine u ($u \in \mathcal{U}$) [m^3/h]

$\overline{Y}_p$ = upper bound on the water pumped by pump p ($p \in \mathcal{P}$) [m^3/h]

$\mu_u = \{v \in \mathcal{V} |$ turbine u represents arc $v\}$

$\mu'_p = \{v \in \mathcal{V} |$ pump p represents arc $v\}$

$\rho_u(x, v)$ = efficiency function of turbine u [MW]

$\theta_p(y)$ = efficiency function of pump p [MW]

Variables

x_{ut} = water flow passing through turbine u in period t ($u \in \mathcal{U}, t \in \mathcal{T}$) [m^3/h].

y_{pt} = water pumped by pump p in period t ($p \in \mathcal{P}, t \in \mathcal{T}$) [m^3/h].

v_{nt} = water volume in reservoir n in period t ($n \in \mathcal{N}, t \in \mathcal{T}$) [m^3].

z_{uti} = auxiliary binary variable (1 when turbine u has operational level $\geq i$ in period t, 0 otherwise) $u \in \mathcal{U}, t \in \mathcal{T}$.

z'_{pti} = auxiliary binary variable (1 when pump p has operational level $\geq i$ in period t, 0 otherwise) $u \in \mathcal{U}, t \in \mathcal{T}$.

Constraints

- **Equilibrium constraint** ($\forall n \in \mathcal{N}, t \in \mathcal{T}$):

$$v_{nt} = v_{n(t-1)} + \sum_{m \in \mathcal{F}_n : D_{(m,n)} \le t} \sum_{u \in \mathcal{U} : \mu_u = (m,n)} x_{u(t-D_{(m,n)})} T - \sum_{m \in \mathcal{F}_n} \sum_{u \in \mathcal{U} : \mu_u (n,m)} x_{ut} T$$
$$+ \sum_{m \in \mathcal{D}_n : D_{(m,n)} \le t} \sum_{p \in \mathcal{P} : (n,m)} y_{p(t-D_{(m,n)})} T - \sum_{m \in \mathcal{D}_n} \sum_{p \in \mathcal{P} : \mu'_p = (m,n)} y_{pt} T + I_{nt} T$$

  with v_{n0} = initial volume of reservoir n ($n \in \mathcal{N}$).

- **Gradient constraint** ($\forall u \in \mathcal{U}, t \in \mathcal{T}$):

$$-GT \le x_{ut} - x_{u(t-1)} \le GT$$

  with x_{u0} is the initial flow in turbine u ($\forall u \in \mathcal{U}$).

- **Discrete operational levels constraints** ($u \in \mathcal{U}, t \in \mathcal{T}$):

$$x_{ut} = \sum_{i = \mathcal{X}_{ut}} z_{uti}(X_{uti} - X_{ut(i-1)})$$
$$z_{ut(i+1)} \le z_{uti} \qquad (i \in \mathcal{X}_{ut})$$
$$-1 \le z_{uti} - z_{u(t-1)i} - z_{u(t+1)(i+1)} \le 0 \qquad (i \in \mathcal{X}_{ut})$$
$$0 \le z_{ut1} + z'_{pt1} \le 1 \qquad \forall u \in \mathcal{U}, p \in \mathcal{P} : \mathfrak{i}(u,p) = 1$$
$$0 \le z_{ut1} + z'_{p(t+1)1} \le 1 \qquad 1 \forall u \in \mathcal{U}, p \in \mathcal{P} : \mathfrak{i}(u,p) = 1$$
$$0 \le z_{u(t+1)1} + z'_{pt1} \le 1 \qquad \forall u \in \mathcal{U}, p \in \mathcal{P} : \mathfrak{i}(u,p) = 1$$

  where $\mathfrak{i}(u,p) = 1$ when turbine u is reversable and acts also as pump p.

Objective function

Minimizing:

- Cost incurred by pumping
  $\sum_{t \in \mathcal{T}} \lambda_t T \sum_{p \in \mathcal{P}} \theta_p(y_{pt})$
- Cost of using water expressed by the water-values
  [different modeling possibilities]

Minus

- Gain generated by turbining

$$\sum_{t \in \mathcal{T}} \lambda_t T \sum_{u \in \mathcal{U}} \rho_u(x_{ut}, v_{nt})$$
$$+ \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{T}} \lambda_t T \sum_{u \in \mathcal{U}} (\rho_u(\sum_{s' \in \mathcal{S} : s' \leq s} f_S(x_{ut}), v_{mt})$$
$$+ \rho_u(\sum_{s' \in \mathcal{S} : s' < s} f'_s(x_u t), v_{nt}))$$

where λ_t are price signals.

Solution Approaches

Practically difficult: **complicated constraints**, and **large size of real instances.**

Looking for **provable high accuracy in a limited amount of time.**

Different research lines:

1. modeling and reformulations: formulation strengthening, cuts, decomposition methods, and approximations to efficiently provide effective lower bounds on the optimal value;
2. heuristics: matheuristics, possibly exploiting the formulations/decompositions/approximations of point 1, to efficiently provide good quality feasible solutions.

Heuristics

Already tested (and presented) different heuristic algorithms.

- Improve the currently available algorithms.
- Design new algorithms.
  For example, use the "local branching" constraint (Fischetti & Lodi, 2003)

$$\sum_{i:\tilde{x}_i=0} x_i + \sum_{i:\tilde{x}_i=1} (1 - x_i) \leq \Pi$$

where Π is the number of binary variables that we allow to change value wrt the rounded LP solution $\tilde{x}$.
We might re-define the objective function as

$$\min \Pi$$

(where Π becomes an integer variable ≥ 0).

Decompositions

The subproblem might be itself decomposed into smaller sub-subproblems.
For example, the only constraints that link the different hydro plants are the
Equilibrium constraint ($\forall n \in \mathcal{N}, t \in \mathcal{T}$):

$$
\begin{aligned}
v_{nt} = \; & v_{n(t-1)} \\
& + \sum_{m \in \mathcal{F}_n : D_{(m,n)} \leq t} \; \sum_{u \in \mathcal{U} : \mu_u = (m,n)} x_{u(t-D_{(m,n)})} T \\
& - \sum_{m \in \mathcal{F}_n} \; \sum_{u \in \mathcal{U} : \mu_u (n,m)} x_{ut} T \\
& + \sum_{m \in \mathcal{D}_n : D_{(m,n)} \leq t} \; \sum_{p \in \mathcal{P} : (n,m)} y_{p(t-D_{(m,n)})} T \\
& - \sum_{m \in \mathcal{D}_n} \; \sum_{p \in \mathcal{P} : \mu'_p = (m,n)} y_{pt} T + I_{nt} T
\end{aligned}
$$

Approximations

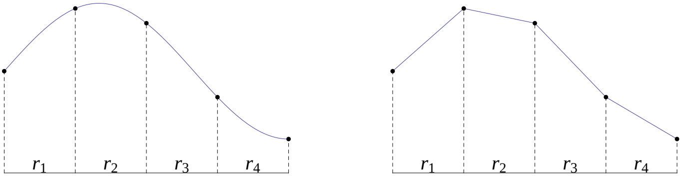
- Real-world optimization problem can be often modeled as a MINLP problem.
- What makes MINLP problem difficult?
  1. non-linear functions;
  2. integer variables.
- MILP solvers more efficient than MINLP ones and handle large-scale instances.
- Trying to get rid of the non-linear functions $\rightarrow$ "linearize" and use MILP solvers!!!!
- **Piecewise linear approximation**: Beale & Tomlin, 1970 (*Special Ordered Sets*).

For the moment, focus on MINLP with non-linear objective function and linear constraints .

Starting simple: univariate function

Consider a function $f(x)$ and construct its piecewise linear approximation.

- Divide the domain of f in $n-1$ intervals of coordinates $x_1, \ldots, x_n$.
- Sample f at each point x_i with $i = 1, \ldots, n$.
- The piecewise linear approximation of f is given by the convex combination of the samples.



(a) (b)

Function of 2 variables: Method 1

1. Simply fix the value of one of the 2 variables and obtain a univariate function: $f(x, \tilde{y})$.
2. Apply methods for approximating univariate functions (previous slide).

The quality of the approximation depends on the function at hand.

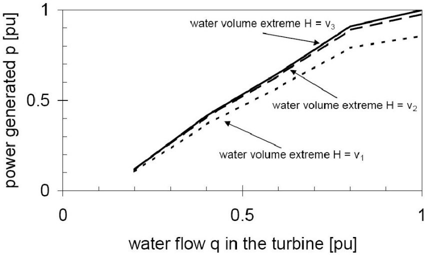
Choose to fix the "less nonlinear" variable.

Function of 2 variables: Method 2

In Conejo et al. (2002) the function $f^a = f(x, y)$ was approximated by considering three prefixed water volumes, say $\tilde{y}^1$, $\tilde{y}^2$, $\tilde{y}^3$ and interpolating, for each $\tilde{y}^r$, the resulting function

$$f^a = f(x, \tilde{y}^r)$$

by piecewise linear approximation.

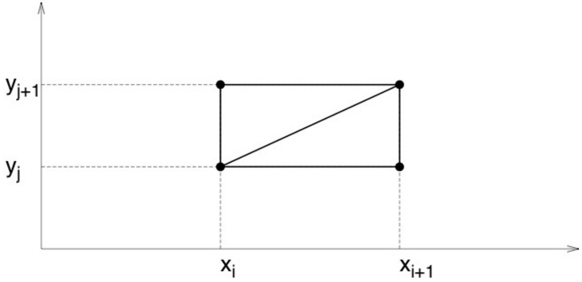


It can be **generalized** by approximating a prefixed number m of values of y.

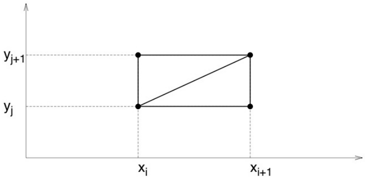
Function of 2 variables: Method 3

Consider a function $f(x, y)$ and construct its piecewise linear approximation.

- Divide the domain of f in a $(n-1) \times (m-1)$ grid of coordinates $x_1, \ldots, x_n$, $y_1, \ldots, y_m$.
- Divide the rectangles in the (x, y)-space in triangles .
- Sample f at each point (x_i, y_j) with $i = 1, \ldots, n$ and $j = 1, \ldots, m$.



Function of 2 variables: Method 3 (cont.d)



Any point $(\tilde{x}, \tilde{y})$

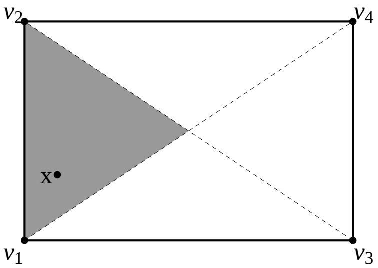
- belongs to one of the triangles;
- can be written as a convex combination of its vertices with weights α_{ij}; and
- the value of function f at $(\tilde{x}, \tilde{y})$ is approximated as

$$f^a = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_{ij} f(x_i, y_j).$$

1 triangle $\leftrightarrow$ 1 binary variable $\rightarrow$ $O(n \times m)$ binaries.

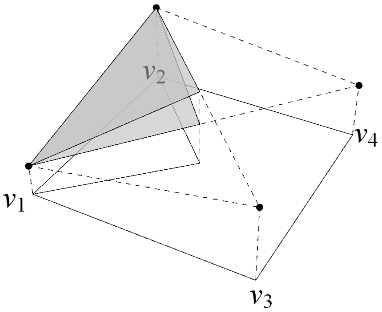
Given a rectangle identified by the four points v_1, v_2, v_3, v_4 we can divide it in 2 triangles in 2 different ways by selecting:

1. diagonal $[v_1, v_4]$; or
2. diagonal $[v_2, v_3]$.



Non-linear $f(x, y) \rightarrow$ 2 different f^a for choice 1 and 2 !

Method 3: Standard Triangulation



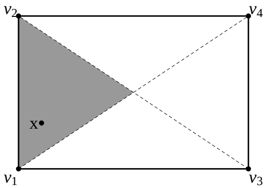
Diagonal $[v_1, v_4]$:

$$\alpha_{v_1} \le \beta_{[v_1,v_2,v_4]} + \beta_{[v_1,v_3,v_4]}$$
$$\alpha_{v_2} \le \beta_{[v_1,v_2,v_4]}$$
$$\alpha_{v_3} \le \beta_{[v_1,v_3,v_4]}$$
$$\alpha_{v_4} \le \beta_{[v_1,v_2,v_4]} + \beta_{[v_1,v_3,v_4]}$$
$$\beta_{[v_1,v_2,v_4]} + \beta_{[v_1,v_3,v_4]} = 1$$

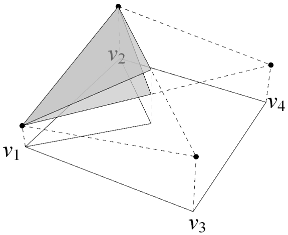
Diagonal $[v_2, v_3]$:

$$\alpha_{v_1} \le \beta_{[v_1,v_2,v_3]}$$
$$\alpha_{v_2} \le \beta_{[v_1,v_2,v_3]} + \beta_{[v_2,v_3,v_4]}$$
$$\alpha_{v_3} \le \beta_{[v_1,v_2,v_3]} + \beta_{[v_2,v_3,v_4]}$$
$$\alpha_{v_4} \le \beta_{[v_2,v_3,v_4]}$$
$$\beta_{[v_1,v_2,v_3]} + \beta_{[v_2,v_3,v_4]} = 1$$

Method 4: Optimistic Approximation



(c)



(d)

Observation is simple:

Why do we need to decide the triangle "offline"?

Let the point $(\tilde{x}, \tilde{y})$ be a convex combination of all the 4 vertices of the rectangle and the MILP solver (optimistically) decide based on the objective function!

$$\alpha_v \leq \beta_{[v_1, v_2, v_3, v_4]} \quad \forall v \in \{v_1, v_2, v_3, v_4\}$$

Method 4: Optimistic Approximation (cont.d)

Let the MILP (optimistically) decide based on the objective function!

In each region:

$$\check{f}(x) = \min \sum_{j=1}^{\nu} \alpha_j f(v_j) \qquad \text{or} \qquad \hat{f}(x) = \max \sum_{j=1}^{\nu} \alpha_j f(v_j)$$

subject to

$$\begin{aligned} \alpha_j &\geq 0 \\ \sum_{j=1}^{\nu} \alpha_j &= 1 \\ \sum_{j=1}^{\nu} \alpha_j x(v_j) &= x \\ \sum_{j=1}^{\nu} \alpha_j y(v_j) &= y \end{aligned}$$

where ν is the number of vertices that characterize the region.

Method 4: Optimistic Approximation Properties

Theorem

The approximations $\check{f}$ and $\hat{f}$ are such that

- $\check{f}$ *(resp. $\hat{f}$) is piecewise convex (resp. concave).*
- $\check{f}$ *and $\hat{f}$ are continuous.*
- *if f is linear then $\check{f} = \hat{f} = f$.*

Method 4: Optimistic Approximation Properties

Theorem

The approximations $\check{f}$ and $\hat{f}$ are such that

- $\Delta_r\left(f, \check{f}\right) \leq D_{\max}(r)$ *and* $\Delta_r\left(f, \hat{f}\right) \leq D_{\max}(r)$ $(\forall r \in \mathcal{R})$.
- *if f is convex (resp. concave) in any $r \in \mathcal{R}$, then $\check{f}$ (resp. $\hat{f}$) is the best possible linear interpolation of the samples $f(v_j)$ in the sense of $\Delta_r(f, \cdot)$.*

where
$\mathcal{R}$ is the collection of rectangles,
$\Delta_r(f, g) = \max_{(x,y)\in r} |f(x,y) - g(x,y)|$, and
$D_{\max}(r)$ is the maximum $\Delta_r\left(f, \tilde{f}\right)$ among all the possible linear interpolations $\tilde{f}$.

Standard vs Optimistic Approach: MILP size

Besides the nice properties, the optimistic approximation provides huge advantages when modeled with a MILP.

- Standard triangulation: 1 binary variable for each triangle $O(n \times m)$.
- Optimistic approximation: 1 binary variable for each rectangle.
- Note: Each axis treated separately, i.e.,
  n binaries for the x axis, and
  m binaries for the y axis. $\rightarrow O(n + m)$.
- For example, 3×3 grid $\rightarrow$ 6 vs 18 binaries
  10×10 grid $\rightarrow$ 20 vs 200 binaries!

$f^a = f(x, y)$: Short-Term Hydro Scheduling

Finding the optimal scheduling of a multi-unit hydro power station in a short-term time horizon.

Maximize the revenue given by power selling.

Assumptions: price-taker situation, the electricity prices and inflows forecast .

Linear constraints, while the objective function has a non-linear part.

The power production is a non-convex, non-concave function $\psi(q, v)$ of the water flow q and the water volume v in the reservoir.

We considered a specific instance of the problem with 168 time periods to be planned.

$f^a = f(x, y)$: MILP size

n	m	optimistic approximation: # var.s all	optimistic approximation: # var.s binary	optimistic approximation: # con.s	optimistic approximation: # nzs	standard approximation: # var.s all	standard approximation: # var.s binary	standard approximation: # con.s	standard approximation: # nzs
9	9	17,471	3,192	5,208	107,515	41,999	27,720	15,624	185,803
17	17	55,103	5,880	7,896	360,187	146,831	97,608	50,568	666,955
33	33	194,879	11,256	13,272	1,317,115	550,031	366,408	184,968	2,532,427
65	65	732,479	**22,008**	**24,024**	5,037,307	2,130,575	1,420,104	711,816	9,876,043

For $n = m =$65:

- Number of binary variables: 22,008 vs 1,420,104.
- Number of constraints: 24,024 vs 711,816.

$f^a = f(x, y)$: Solving the MILP

Single processor of an Intel Core2 CPU 6600, 2.40 GHz, 1.94 GB of RAM under Linux.

Cplex 10.0.1.

Time limit of 1 hour.

		optimistic approximation				standard approximation				
n	*m*	solution value	% error	CPU time	# nodes	solution value	% error	final %gap	CPU time	# nodes
9	9	31,565.40	-2.34	14.71	1,507	31,565.40	-2.34	—	169.30	9,837
17	17	31,577.20	-2.31	755.96	36,507	31,577.20	-2.31	0.19	3,600.00	73,401
33	33	31,626.20	-2.35	277.13	2,567	n/a	n/a	n/a	3,600.00	5,500
65	65	31,640.30	-2.33	2,003.18	2,088	n/a	n/a	n/a	failure	failure

- Number of solved instances: 4 vs 2.

$f^a = f(x, y)$: Going Logarithmic

Vielma & Nemhauser, 2011 : MILP model for the standard triangulations with a logarithmic number of variables (binary tree structure).

Doable also for the Optimistic approximation.

		optimistic approximation				logarithmic standard approximation			
		# var.s		# con.s	# nzs	# var.s		# con.s	# nzs
n	m	all	binary			all	binary		
9	9	17,471	3,192	5,208	107,515	16,127	1,848	4,368	142,963
17	17	55,103	5,880	7,896	360,187	51,407	2,184	5,040	578,419
33	33	194,879	11,256	13,272	1,317,115	186,143	2,520	5,712	2,501,683
65	65	732,479	22,008	24,024	**5,037,307**	713,327	2,856	6,384	11,056,243

		optimistic approximation				logarithmic standard approximation			
n	m	solution value	% error	CPU time	# nodes	solution value	% error	CPU time	# nodes
9	9	31,565.40	-2.34	14.71	1,507	31,538.70	-2.26	18.69	1,723
17	17	31,577.20	-2.31	755.96	36,507	31,577.20	-2.31	20.84	369
33	33	31,626.20	-2.35	277.13	2,567	31,624.10	-2.35	231.99	1,531
65	65	31,640.30	-2.33	2,003.18	2,088	31,640.30	-2.34	530.56	435

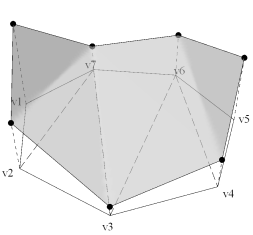
$f^a = f(x,y)$: Going Logarithmic (cont.d)

n	m	logarithmic optimistic approximation # var.s		# con.s	# nzs	logarithmic standard approximation # var.s		# con.s	# nzs
		all	binary			all	binary		
9	9	16,127	1,848	4,032	135,907	16,127	1,848	4,368	142,963
17	17	51,407	2,184	4,704	553,891	51,407	2,184	5,040	578,419
33	33	186,143	2,520	5,376	2,409,955	186,143	2,520	5,712	2,501,683
65	65	713,327	2,856	6,048	10,701,091	713,327	2,856	6,384	11,056,243

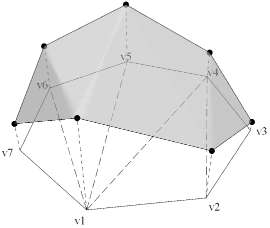
n	m	log optimistic approximation solution value	% error	initial %gap	CPU time	# nodes	log standard approximation solution value	% error	initial %gap	CPU time	# nodes
9	9	31,565.40	-2.34	1.13	17.87	1,734	31,538.70	-2.26	1.14	18.69	1,723
17	17	31,577.20	-2.31	1.35	21.08	450	31,577.20	-2.31	1.35	20.84	369
33	33	31,626.20	-2.35	1.24	263.88	2,195	31,624.10	-2.35	1.25	231.99	1,531
65	65	31,640.30	-2.33	1.20	664.15	796	31,640.30	-2.34	1.20	530.56	435

Why? $\log(nm) = \log(n) + \log(m)$
Advantages of the optimistic approximation: MILP model of limited size (tractable) and easy to implement .

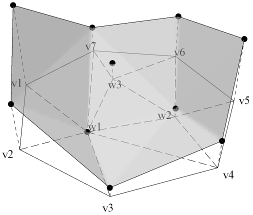
Oversampling



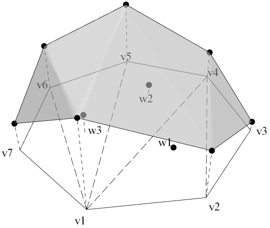
(e)



(f)



(g)



(h)

For more information...

Visit the project web site:

`http://www.lix.polytechnique.fr/~dambrosio/PGMO.php`.