Comparison between Dynamic Programming and Reinforcement Learning
A case study on maize irrigation management

Jacques-Eric BERGEZ

INRA Agronomy, Toulouse

Mark EIGENRAAM

Dpt. of Natural Resources and Environment, Melbourne, Australia

Frédérick GARCIA

INRA Biometry and Artificial Intelligence, Toulouse

Context and motivations

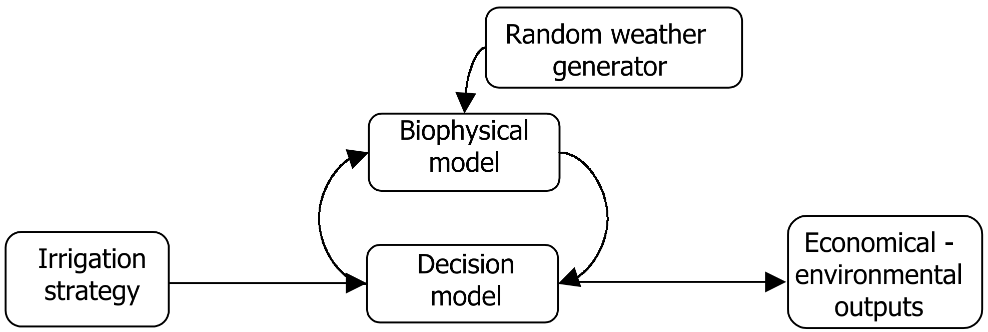
Irrigation scheduling is an important decision problem in agriculture that has to be managed carefully

- Irrigation has a major effect on yield and gross margin
- Applying too much water can create environmental or political problems
- A good timing of application can increase the efficiency of irrigation

Optimizing irrigation strategies

We develop simulation optimization methods for designing new irrigation scheduling approaches

These methods use MODERATO, a growth simulator and an irrigation strategy simulator for maize crops, coupled with a stochastic weather generator



Objective of the present study

To compare 2 different methods

- Stochastic Dynamic Programming
- Reinforcement Learning

for solving the subproblem of

deciding when to start the irrigation campaign

Modeling the problem as a MDP

A strategy = what to do in any state at any time

- State-space $S = \Delta \times \Sigma \cup \{s_{end}\}$
  $\delta \in \Delta$ is the soil water deficit,
  $\sigma \in \Sigma$ is the accumulated thermal units above 6°C since sowing.

- In each state (δ, σ) two possible actions :
  - wait until the next day (W),
  - start irrigation today (I).
  I leads to s_{end} , then a specific strategy is followed.

- Decision times: each day

Optimality criterion

Expected value of the gross margin obtained in s_{end}

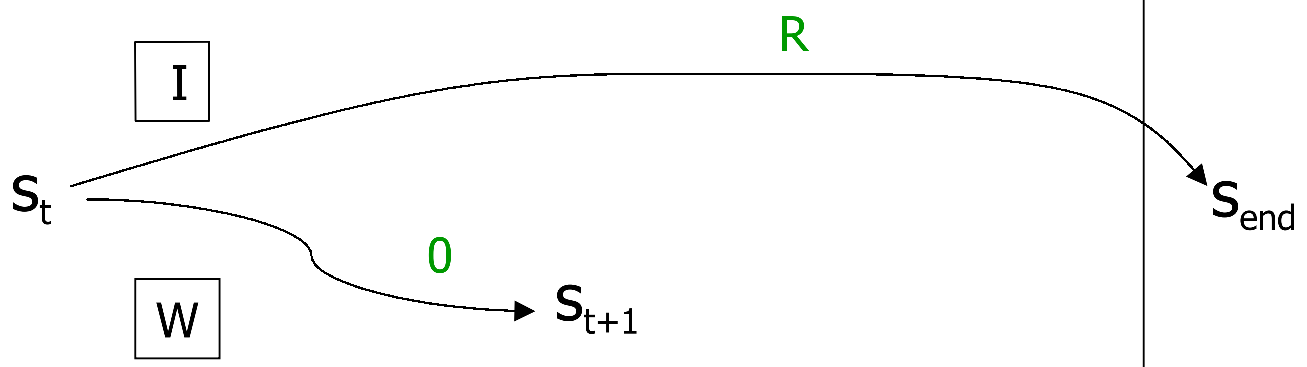
$$R = p \, . \, Y - l \, . \, N - q \, . \, C - X$$

where

- Y is the final grain yield,
- N the number of irrigation rounds,
- C the total amount of water used for irrigation,
- p, l, q the unit prices/costs,
- X a fixed production cost.

Y, N and C depend on the climate.

Optimal Stopping Problem



I or W ?

Value of acting in s_t

$Q(s_t, I) > Q(s_t, W)$?

The optimal Q-values depend on the rewards and on the transition probabilities

Optimization procedure
Stochastic Dynamic programming

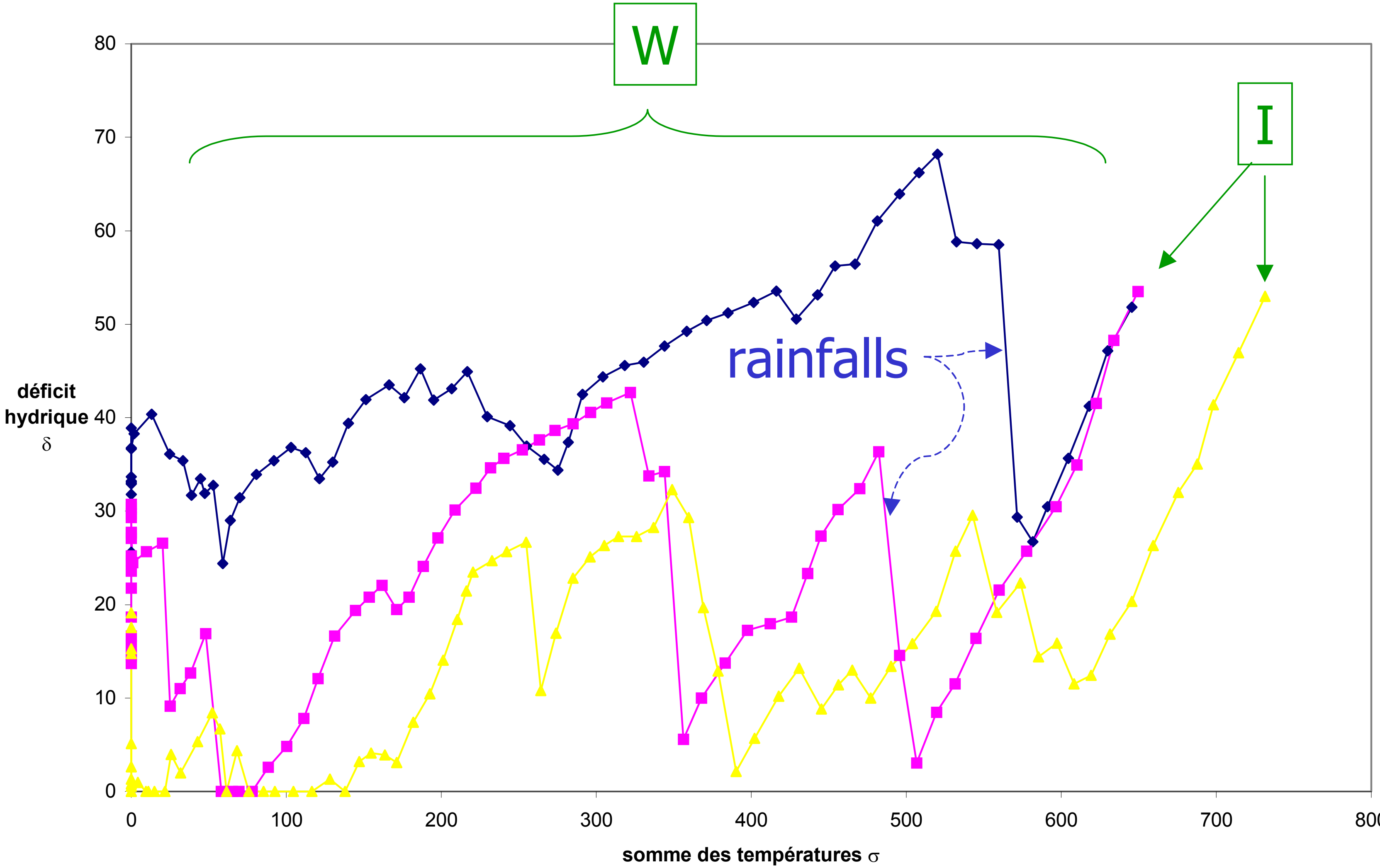
• The domain $\Delta \times \Sigma$ is represented as a set of grid-points (δ_i, σ_j) for $0 \leq i < I$, $0 \leq j < J$.

• The transition probabilities and the expected local rewards are estimated by simulation with MODERATO.

Two types of trajectory are simulated:

- W until s_{end}
- I from a random threshold σ_I

Simulated process trajectories



Optimization codes

- Policy iteration algorithm, infinite horizon, no discount factor
- Use of General Purpose Dynamic Programming (GPDP, J. Kennedy) software
- A more efficient C code has been developed

Optimization procedure
Reinforcement Learning

- No estimation of the model
- simulated trajectories

  I from a random threshold σ_I

  are directly used for estimating the optimal Q-values
- Q-learning + TD(λ) algorithms
- C codes

Reinforcement Learning
Q-learning

After each trajectory

$$\{(s_0,W), (s_1,W), \ldots, (s_{t-1},W), (s_t,I), R_t\}$$

Q-values are updated

$$Q(s_k, W) \leftarrow (1-\varepsilon)\, Q(s_k, W) + \varepsilon \max\{Q(s_{k+1}, W), Q(s_{k+1}, I)\},\ k<t$$

$$Q(s_t, d_t) \leftarrow (1-\varepsilon)\, Q(s_t, d_t) + \varepsilon\ R_t$$

Converges to optimal Q-values

CMAC approximation of the Q-values

- Standard approximation scheme in RL
- The domain $\Delta\times\Sigma$ is represented as p shifted grids (δ^k_i, σ^k_j) for $0\leq i<I$, $0\leq j<J$, and $1\leq k\leq p$.
- Q-values are approximated by

$$Q(\delta, \sigma, d) = \Sigma_k Q(\delta^k_i, \sigma^k_j, d) \text{ for } d\in\{W,I\},$$

where (δ^k_i, σ^k_j) represents (δ, σ) on grid k.

- The update rule turns now on the $Q(\delta^k_i, \sigma^k_j, d)$ values

Numerical application

Specific case based on data from SW France

- δ ranges from 0 to 150 mm, σ from 0 to 1800°C day.

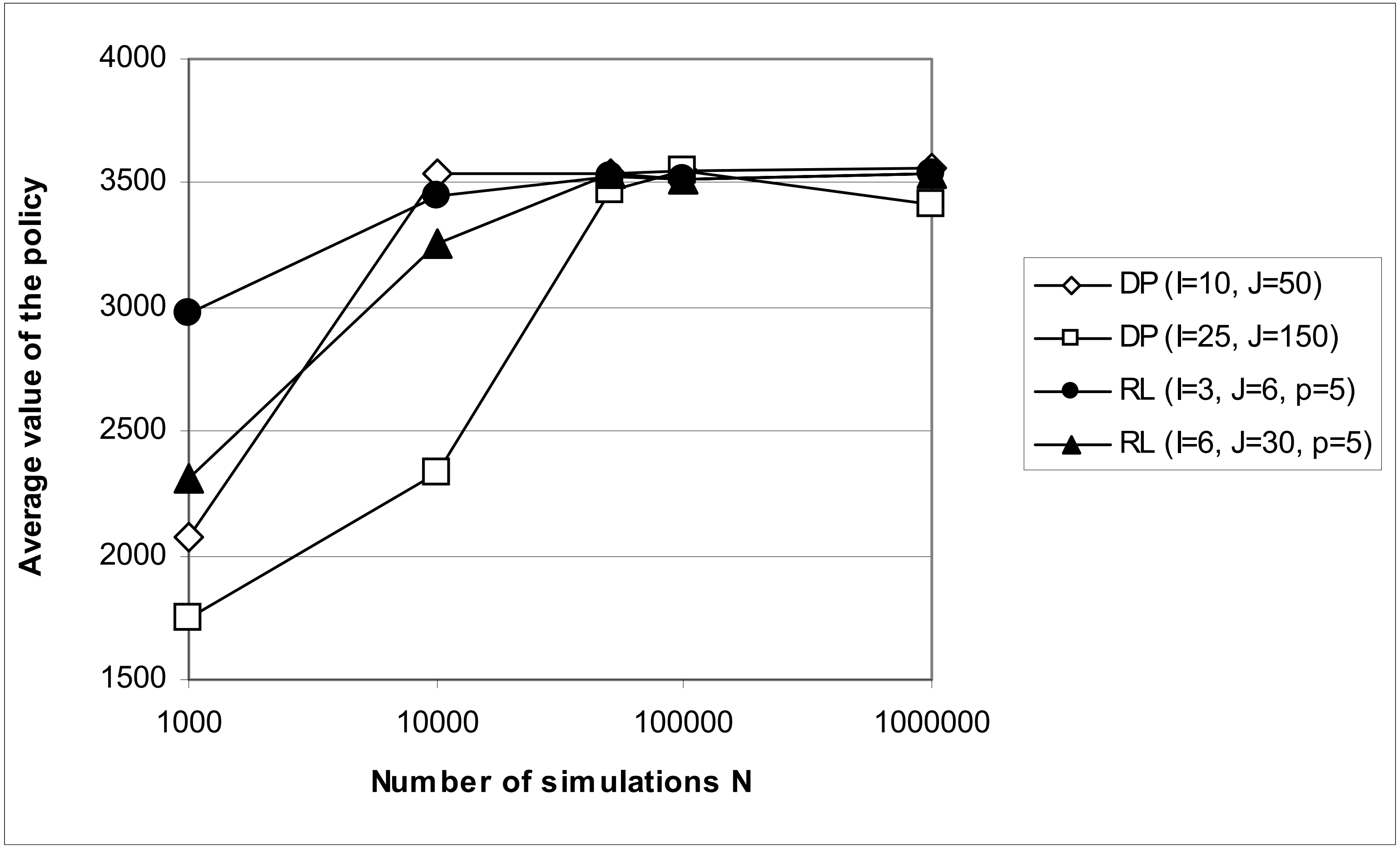
- DP : I=10, J=50 (500 states),
  I=25, J=150 (3750 states)

- RL : I=3, J=6, p=5 (90 cells)
  I=6, J=30, p=5 (900 cells)

About the same spatial discretisation

- Nmax=10^6 simulations (~ 30 hours)

Comparison between DP et RL



Conclusions

- RL performs better than DP for small N

- Small-sized grids are preferable

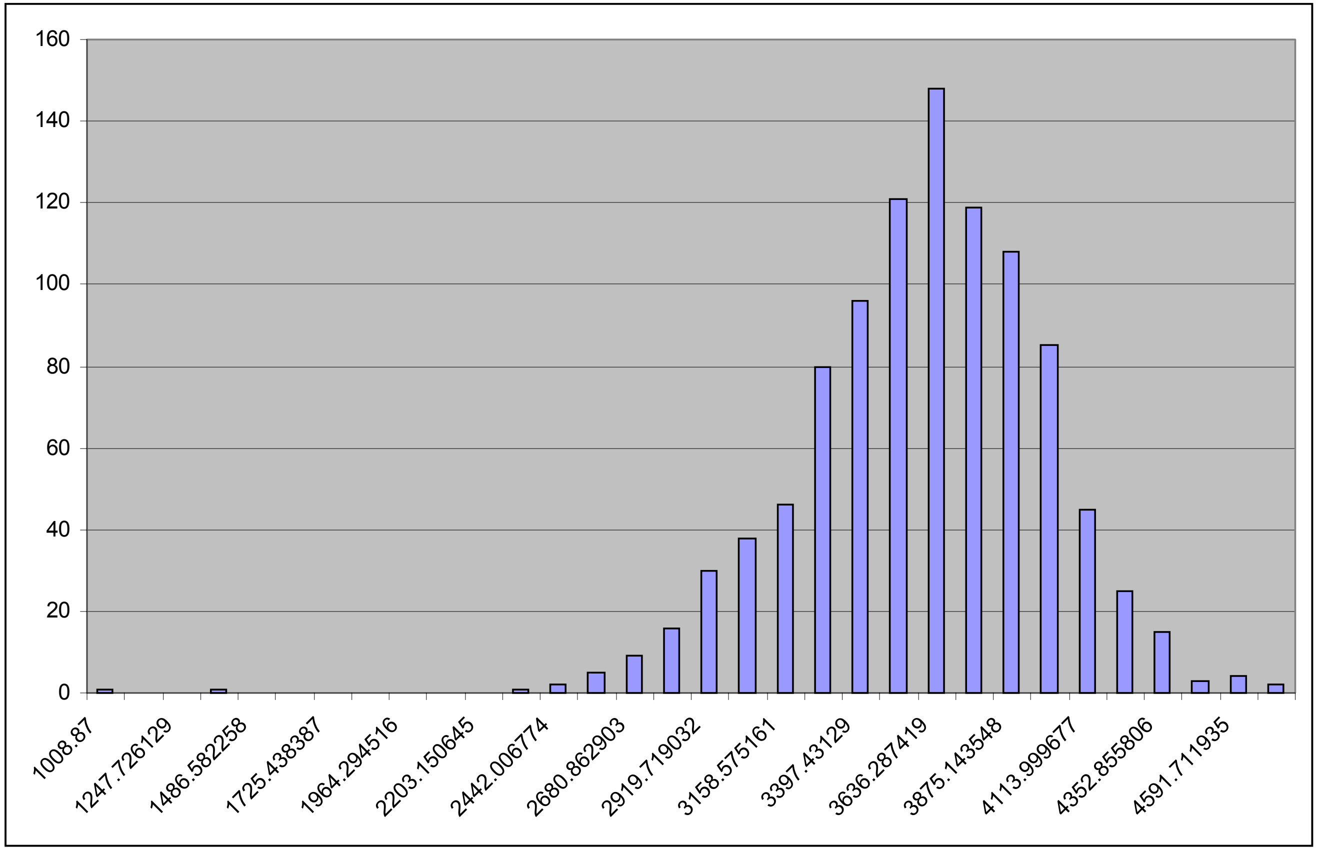
- Near-optimal control limit policies

$$\text{I when } \sigma > \sigma^{*}, \delta > \delta^{*}$$

consistent with expert-knowledge

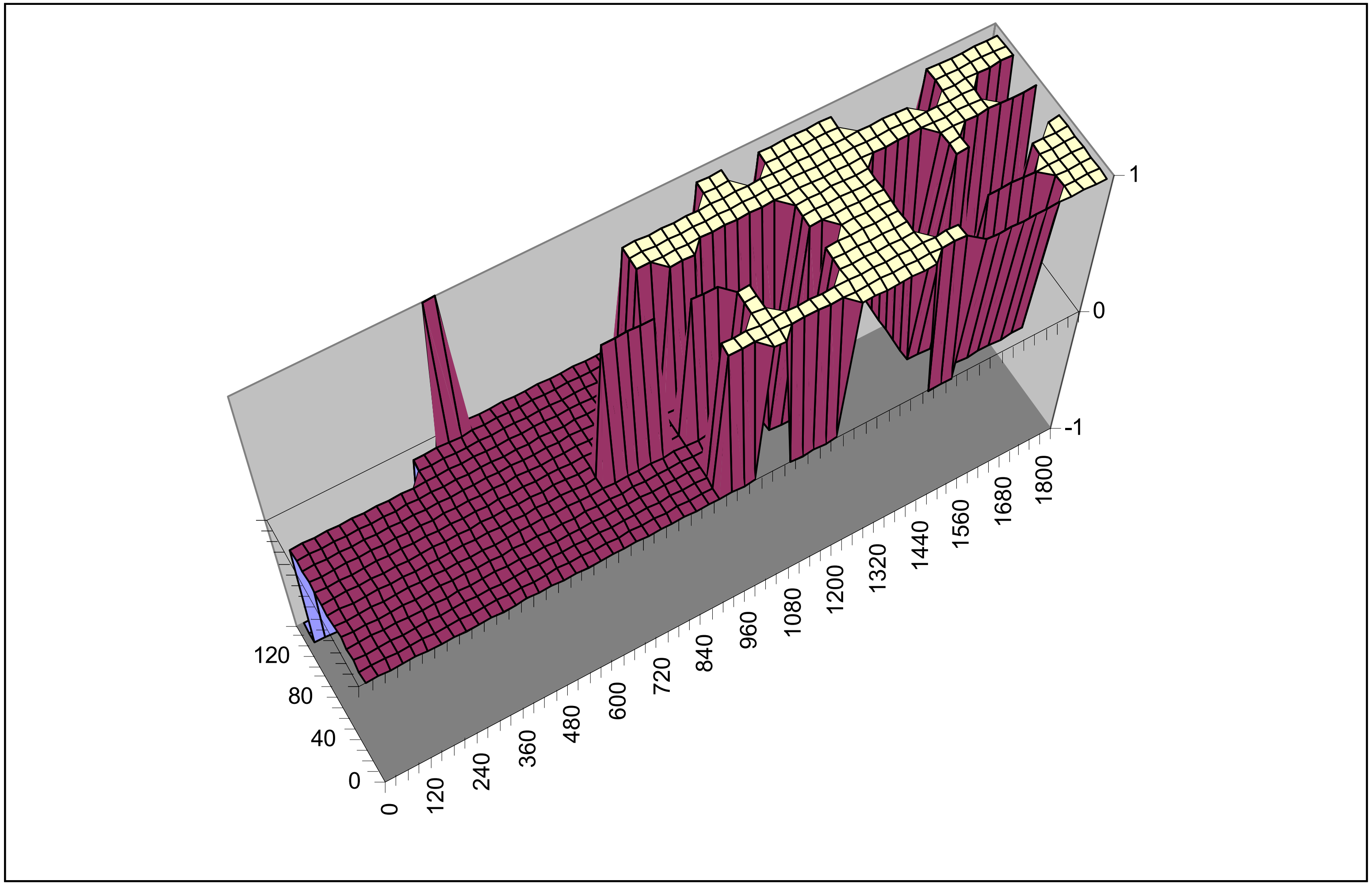
- To compare with other optimization approaches

Estimation of the policy value



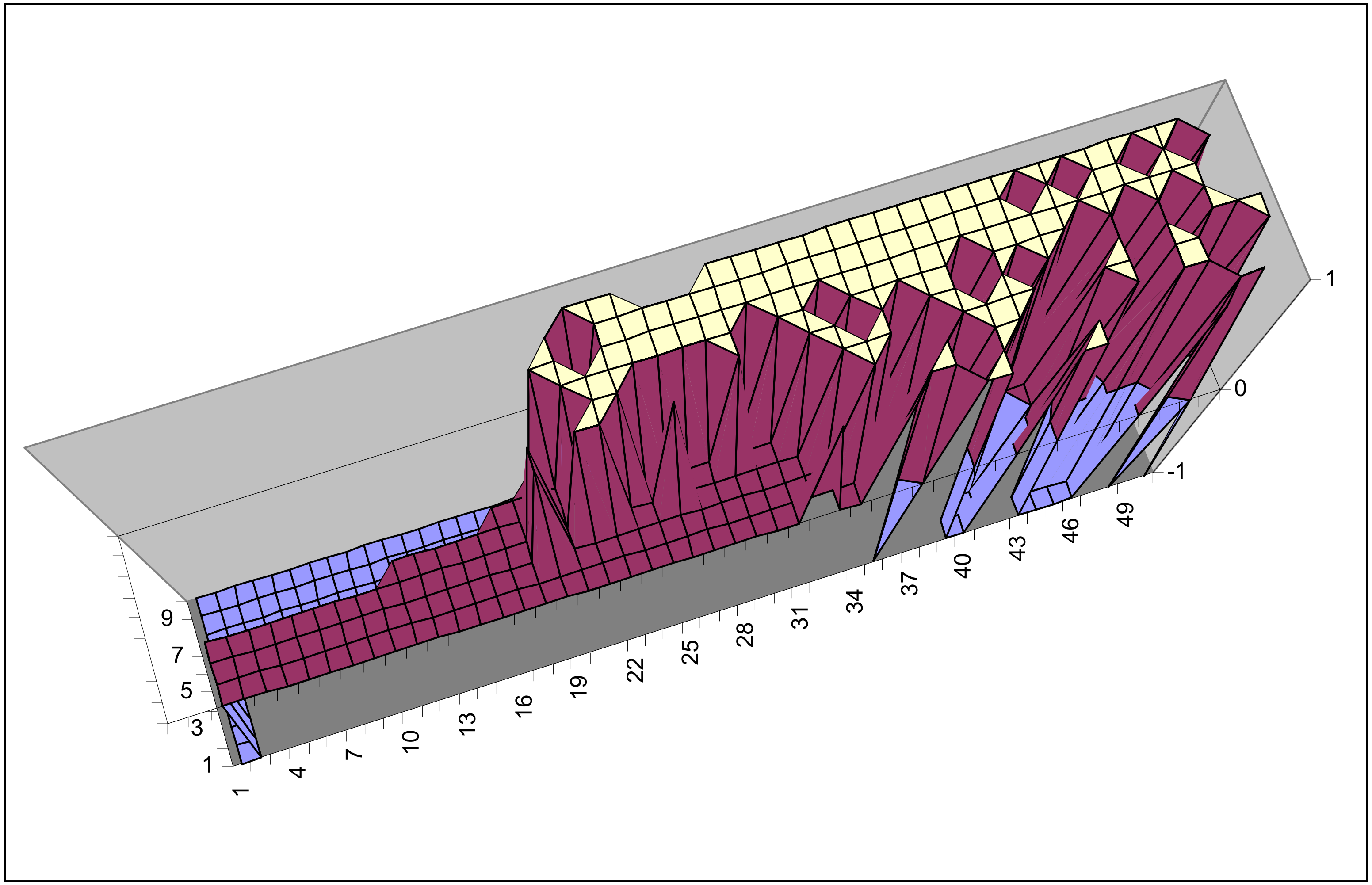
3×6×5 RL, $N=10^6$, 1000 simulations.

Optimal RL policy



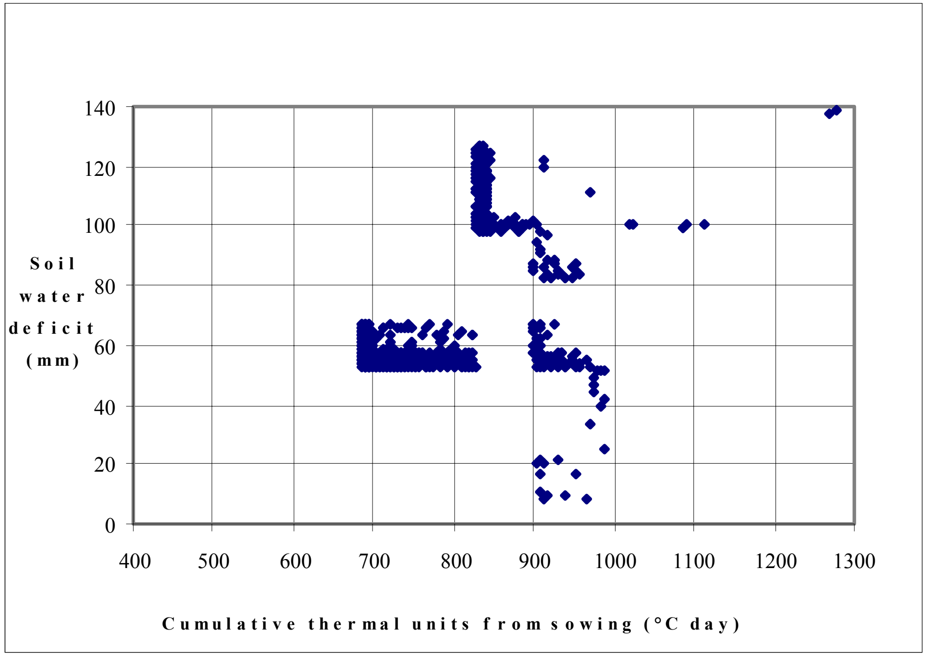
3×6×5 RL, $N=10^6$

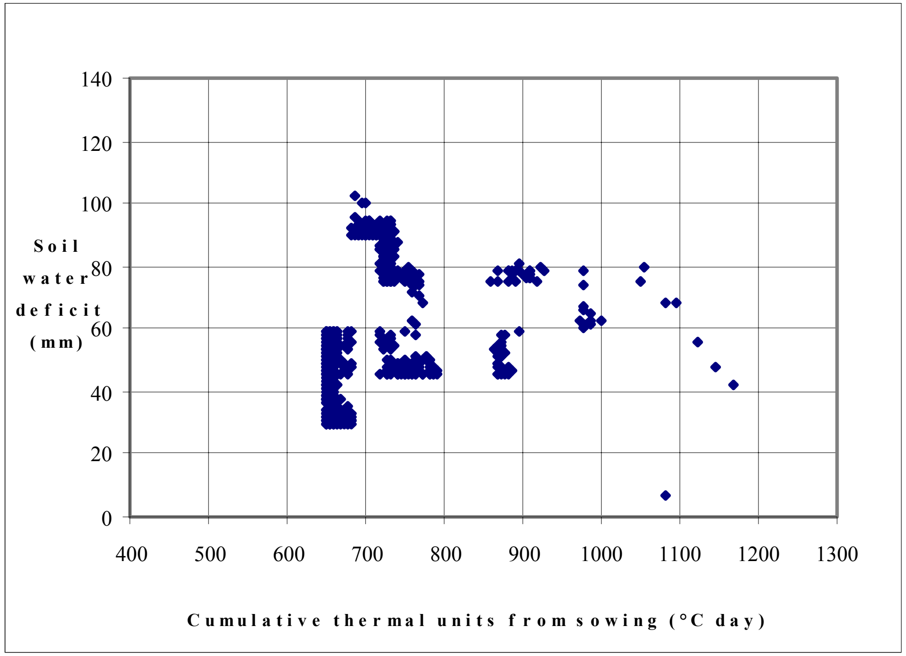
Optimal DP policy



10×50 DP, N=10^6

Simulated starts of irrigation





$3 \times 6 \times 5$ RL and 10×50 DP, $N=10^6$

Optimization of a control limit policy

Start irrigation when $\sigma > \theta$