Neuro-dynamic programming as a new framework for decision support for deficit irrigation sytems

Niels Schütze[1] **and Gerd H. Schmitz**[1]

[1]Institute of Hydrology and Meteorology, Dresden University of Technology
D-01187 Dresden, Würzburger Str.46, Germany. E-Mail: niels.schuetze@tu-dresden.de

Keywords: *Deficit irrigation, dynamic optimization, neuro-dynamic programming*

ABSTRACT

The great challenge of the agricultural sector is to produce more food and/or more revenue from less water, which can be achieved by optimal irrigation management. A task of primary importance is the problem of intraseasonal irrigation scheduling under limited seasonal water supply. On the intraseasonal level, a limited amount of water is to be distributed over a number of irrigations, taking into account the crops' response to water stress at different stages during the growing season.

Dynamic programming (DP) has been extensively used for optimization of irrigation scheduling problems (Bras and Cordova, 1981; Rao et al., 1988; Sunantara and Ramirez, 1997; Prasad et al., 2006). An alternative approach to calculate optimal irrigation schedules is provided by static optimization techniques such as linear and nonlinear programming (Shang and Mao, 2006; Gorantiwar et al., 2006). Dynamic optimization is a closed-loop optimization strategy designed for obtaining an optimal look up table for selecting – at each stage of the atmosphere-plant-soil system during a growing season – the optimal irrigation decision for each possible state of the system. The popularity and success of this technique can be attributed to the fact that nonlinear and stochastic features of scheduling problems can be handled by DP (Bertsekas, 2000). However, it is well known that computational requirements of DP become overwhelming when the number of state and control variables is too large (Bellman and Dreyfus, 1962). For this reason all the studies applying DP for optimal irrigation scheduling have their limitations (Bras and Cordova, 1981; Rao et al., 1988; Sunantara and Ramirez, 1997). A second disadvantage of the classical DP optimization strategies lies in the necessary discretization of the state variables of the water balance models. This limits the predictive reliability of the models significantly which, in turn, affects the computation of the optimal schedules.

A neuro-dynamic programming technique (NDP), which overcomes numerous limitations of dynamic programming (DP), is used for determining the optimal irrigation policy in deficit irrigation. This new simulation-based approach combines a broader range of simulation models with optimization algorithm for solving deterministic and stochastic optimization problems. In the context of simulation-based optimization, a simulation model can be thought of as a function (whose explicit form is a black box for the optimizer) that turns input parameters into output performance measures (Gosavi, 2003). The developed neuro-dynamic programming algorithm for single crop intraseasonal scheduling operates together with general water flow and crop growth simulation models. In the contribution, different management schemes are considered and crop-yield functions generated with the NDP optimization algorithm are compared.

The paper is organized as follows. In section 2, we review the new Least-Squares Temporal Difference (LSTD) algorithm for calculating the approximate cost-to-go function for the dynamic programming approach. In section 3, a case study involving deficit irrigation of 4 crops is presented to illustrate the new method and we discuss the results, especially crop-yield functions generated with the dynamic simulation-based scheduling algorithm under both, flexible and very restrictive irrigation constraints (see Fig.1 as a first example). In section 4, we offer some conclusions and suggestions for potential stochastic applications for the NDP approach.

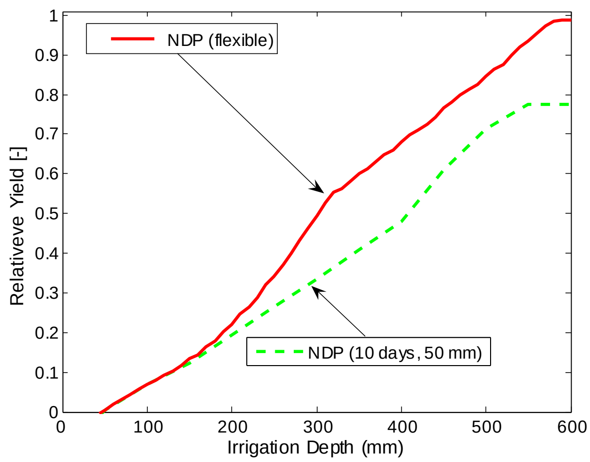

Figure 1. Normalized seasonal crop production function for maize.

1 INTRODUCTION

Agriculture is still the greatest water user of all while having the lowest water use efficiency. Especially, irrigated agriculture is particularly guilty of inefficient water use, the pollution of ground and surface water and land degradation. Thus, good water management practices in irrigation aim to improve water use efficiency, along with preserving the soil and water resources, without sacrificing crop productivity. When irrigation is constrained by limited water availability, a maximum crop yield is not achievable. With deficit irrigation, the plants are consciously under-supplied with water and a reduced crop yield is accepted as the penalty. However, each plant's level of water stress sensitivity fluctuates with respect to its different growth phases. For this reason, when laying down the irrigation schedules for an entire growth period, it is important to decide beforehand when the growth phases requiring generous irrigation water volumes will occur and, on the other hand, when smaller volumes will suffice.

Dynamic programming (DP) has been extensively used for optimization of deficit irrigation scheduling problems (Bras and Cordova, 1981; Rao et al., 1988; Sunantara and Ramirez, 1997; Prasad et al., 2006). An alternative approach to calculate optimal irrigation schedules is provided by static optimization techniques such as linear and nonlinear programming (Shang and Mao, 2006; Gorantiwar et al., 2006).

Dynamic optimization is a closed-loop optimization strategy designed for obtaining an optimal look up table for selecting – at each stage of the atmosphere-plant-soil system during a growing season – the optimal irrigation decision for each possible state of the system. The popularity and success of this technique can be attributed to the fact that nonlinear and stochastic features of scheduling problems can be handled by DP (Bertsekas, 2000). However, it is well known that computational requirements of DP become overwhelming when the number of state and control variables is too large (Bellman and Dreyfus, 1962). For this reason all the studies applying DP for optimal irrigation scheduling have their limitations (Bras and Cordova, 1981; Rao et al., 1988; Sunantara and Ramirez, 1997). A second disadvantage of the classical DP optimization strategies lies in the necessary discretization of the state variables of the water balance models. This limits the predictive reliability of the models significantly which, in turn, affects the computation of the optimal schedules.

The objective of this study is to demonstrate how to overcome the current restrictions in irrigation scheduling by a new method for dynamic optimization using a simulation-based strategy. The simulation-based approach combines a broader range of simulation models with optimization algorithm for solving deterministic and stochastic optimization problems. In the context of simulation-based optimization, a simulation model can be thought of as a function (whose explicit form is a black box for the optimizer) that turns input parameters into output performance measures (Gosavi, 2003). Evolutionary or genetic algorithms (EA) are popular heuristic methods which are capable of achieving global or near global optimal solutions to static simulation-based optimization problems.

For solving dynamic simulation-based optimization problems neuro-dynamic programming (NDP) or reinforcement learning (RL) is employed. NDP avoids the exponential increase of computations through the use of parametric approximate representations of the cost-to-go function (Bertsekas, 2000). Compared to the classical numerical solution approach for DP, which performs exhaustive sampling of the entire state space in solving the stage-wise optimization, these approaches sample only a small, crucial fraction of the state space and thus require dramatically less computation.

2 METHODOLOGY

2.1 Formulation of the dynamic optimization problem

The dynamic problem for a limited water supply in deficit irrigation can be formulated as summarized in Box 1. In equation 1 is **S** an irrigation schedule with the daily irrigation decision v_i. The recursive Bellman equation of the DP (shown in Eq.2) defines the calculation of the state-dependend value function Y where θ_i and V_i are the state variables of the irrigation system, i.e. the mean moisture content in the soil and the water volume available for the remaining decision stages, respectively.

The reward y_i is the daily contribution to the crop yield response, which is determined by an additive formulation derived from the FAO-33 crop yield response model (Doorenbos and Kassam, 1979). In equation 3 is Y_k the cumulative yield according to the crop sensitivity factor $K_{y,k}$ for the jth growing period which starts at decision stage $n_k + 1$ and ends at stage n_{k+1} . To solve the optimality equation (2) by DP, a sequential calculation of Y_i^* is performed for all stages and at each stage for all states, usually in a backward direction starting from the terminal stage N.

$$Y^* = \max_{\text{all } \mathbf{S}} \left(\sum_{i=1}^{N} y_i^{\mathbf{S}} \right) \quad \text{with} \quad \mathbf{S} = \{v_1, \ldots, v_i, \ldots, v_N\} \quad \text{subject to:} \quad \sum_{i=1}^{N} v_i \leq V_0 \tag{1}$$

$$\begin{aligned} Y_i^*(V_i, \theta_i) = y_i(V_i, \theta_i) + \max_{v_i \in [0, \min(V_i, V_{max})]} [Y_{i+1}(V_i - v_i, \theta_{i+1})] \\ \text{for } i = 1 \ldots N-1 \\ \text{and} \\ Y_N^*(V_N, \theta_N) = y_N(V_N, \theta_N,) \text{ for } i = N \end{aligned} \tag{2}$$

$$\begin{aligned} y &= \frac{Y_a}{Y_{max}} = \prod_{j=1}^{M} \left(1 - K_{Y,j} \left(1 - \frac{\sum_{i=n_{j-1}+1}^{n_J} AET_i}{\sum_{i=n_{j-1}+1}^{n_J} {}^{n_j} PET_i} \right) \right) \\ &= 1 + \sum_{j=1}^{M} \left(\sum_{i=n_{j-1}+1}^{n_j} \left(\prod_{k=0}^{j-1} Y_k \right) K_{Y,j} \left(\frac{AET_i}{\sum_{i=n_{j-1}+1}^{n_j} PET} - \frac{1}{n_j - n_{j-1}} \right) \right) \end{aligned} \tag{3}$$

Box 1: Dynamic optimization framework with the additive crop response model.

2.2 Solving the dynamic optimization problem with neuro-dynamic programming

Classical dynamic programming is based on the premise that the number of states $\vec{x}$ of a system is finite. This is not the case if we apply irrigation simulation models which use continuous variables $(x_1, \ldots, x_n)$. The simulation-based approach of DP approximates the cost-to-go Function $Y^*(\vec{x})$ by a good approximation function $\tilde{Y}(\vec{x}, \mathbf{W})$ in an iterative loop. The parameters $\mathbf{W}$ are determined by some form of optimization, e.g. by using a least squares framework, minimizing the error of the Temporal Differences (TD)

$$e = \tilde{Y}_i(\vec{x}_i, \mathbf{W}) - (y_i + \tilde{Y}_{i+1}(\vec{x}_{i+1}, \mathbf{W}))$$

which would be equal to zero in the ideal case for all simulated states of the irrigation system.

Neuro-dynamic programming (NDP) uses linear basis function approximators (Taylor series, Tile coding or radial basis function) or nonlinear universal approximators like Multilayer Perceptron (MLP) for learning the cost-to-go function $\tilde{Y}$(Sutton and Barto, 1998). In this study we employed a linear approximation approach where the cost-to-go function is given by a linear combination of l basis functions ϕ_k

$$\tilde{Y}(\vec{x}, \mathbf{W}) = \sum_{k=1}^{l} w_k \phi_k(\vec{x}) \tag{4}$$

with the parameter vector w_k and a radial basis function (RBF) as the choice of ϕ_k:

$$\phi_k(\vec{x}) = \exp\left(-\frac{\|\vec{x} - \vec{c_k}\|^2}{2\sigma^2} \right). \tag{5}$$

where σ is a suitable chosen radius and $\vec{c}_k$ are the centers of the l basis functions. For obtaining the approximation function $\tilde{Y}(\vec{x}, \mathbf{W})$ we use the *policy iteration algorithm* that alternates between approximate *policy evaluation* steps and *policy improvement* steps (see Fig.2). Before we describe the algorithm in brief we also need to introduce the time t as a state variable. Thus, we define $v_i(\vec{x}) \equiv v_i(V_i, \theta_i, t_i)$ and now work with a stationary policy. This is a precondition for the application of the policy iteration algorithm.

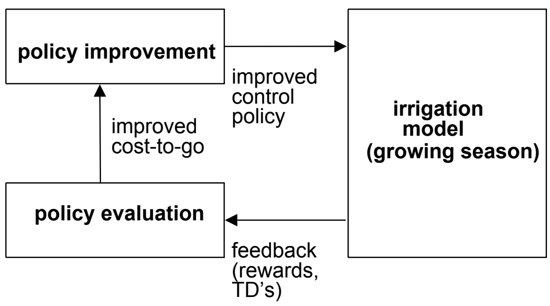

Figure 2. Basic structure of the policy iteration algorithm.

Algorithm 1 Approximate policy iteration using LSTD(λ)-policy evaluation.

$\triangleright$ assign parameters n_{max}, eps, λ, σ, $\{\vec{c}_k\}$,d_{min}, V_0, V_{min}, V_{max}, $n = 0$
$\triangleright$ initialize a random policy $\mathbf{S}_1 = \{\text{rand}(\tilde{v}_i)\}$
while $(n < n_{max})$ or $(\mathbf{S}_{n-1} \equiv \mathbf{S}_n)$
  $\triangleright$ $\mathbf{A} = 0$; $\mathbf{b} = 0$; $n = n + 1$;
  for $(i = 1; i < N; i++)$
    $\triangleright$ calculate $y_i(\vec{x}_i, \tilde{v}_i)$ and $\vec{x}_{i+1}$ using simulation
    $\triangleright$ accumulate the temporal differences
      $\mathbf{A} = \mathbf{A} + \mathbf{z}_i(\phi(\vec{x}_i) - \phi(\vec{x}_{i+1}))^T$
      $\mathbf{b} = \mathbf{b} + \mathbf{z}_i y_i$
      $\mathbf{z}_{i+1} = \lambda \mathbf{z}_i + \phi(\vec{x}_{i+1})$
  endfor
  $\triangleright$ evaluate the approximate cost-to-go $\tilde{Y}(\vec{x}, \mathbf{W})$ using pseudoinverse of $\mathbf{A}$
    $\mathbf{W} = \mathbf{A}^{-1}\mathbf{b}$
  $\triangleright$ improve policy

$$\mathbf{S}_{n+1} = \{\tilde{v}_i\} \text{ with } \begin{cases} p(1-\epsilon) \rightarrow \tilde{v}_i = \arg\left(\max\limits_{v_i \in [0, \min(V_i, V_{max})]} \left[y_i(V_i, \theta_i) + \tilde{Y}(V_i - v_i, \theta_{i+1}, t_i, \mathbf{W}) \right] \right) \\ else \quad\quad \rightarrow \tilde{v}_i = \text{rand}([0, V_i]) \end{cases}$$

  subject to $\text{abs}(d_i - d_j) \geq d_{min}$ and $V_{max} \geq \tilde{v}_i \geq V_{min}$ for all $i = 1 \ldots N$-1
endwhile

The cost-to-go approximation function is constructed by a Least-Squares Temporal Differences policy evaluation LSTD(λ) (Boyan, 2002) and a ϵ-greedy policy improvement, which finds a new policy by maximizing the actual cost-to-go function in the space of feasible policies (Sutton and Barto, 1998). The policy iteration algorithm (see Alg.1) contains the following procedures:

Simulation: The irrigation model simulates a scenario (trajectory) with the actual policy $\mathbf{S}_n$ and calculates the rewards $y_1(\vec{x}_i, \tilde{v}_i)$ for all the states that are on the trajectory.

Accumulation of the temporal differences: During the simulation on each state transition the temporal differences among all RBF ϕ are updated in $\mathbf{A}$ and $\mathbf{b}$ according to their respective eligibilities z_i. The eligibility vector may be seen as an algebraic trick by which TD propagates rewards backward over the current trajectory without having to remember the trajectory explicitly. Thus, each RBF's eligibility at time i depends on the trajectory's history. λ controls how the TD errors between successive predictions are passed back in time. If λ is set to 0, the error signal only propagates to the previous state. If it is set to 1, all previous states are affected by an exponentially decaying amount.

Policy evaluation: Updates of the approximation function $\tilde{Y}$ are carried out offline, i.e. the coefficients $\mathbf{W}$ of $\tilde{Y}$ are modified only at the end of each trajectory by solving the linear least-squares problem $\mathbf{W} = \arg\min \|\mathbf{AW} - \mathbf{b}\|^2$ using the pseudoinverse of $\mathbf{A}$.

Policy improvement: The exploration policy uses a ϵ-greedy policy: The greedy action $\tilde{v}_i$ (i.e. the one for which the sum of the reward y_i and the successor states estimated cost-to-go $\tilde{Y}$ is the maximum) is chosen with probability $1 - \epsilon$, and with probability ϵ a random action is drawn from a uniform distribution over the range zero to the remaining water volume V_i. The value of ϵ is reduced during learning, until the policy improvement step converges to the greedy one. For obtaining the greedy action $\tilde{v}_i$ a Line search method is employed.

The policy iteration algorithm continues until sub-optimal stable policies are achieved which will also be reflected by good returns from the approximate cost-to-go function. Good estimates of the initial policy can be used to accelerate the convergence of $\tilde{Y}$, and speed up the convergence of the entire algorithm. However, because function approximations are based on limited simulations during the iterations, approximation errors can be significant and convergence cannot be guaranteed (Bertsekas and Tsitsiklis, 1996).

3 APPLICATION TO INTRASEASONAL SCHEDULING IN DEFICIT IRRIGATION

For analyzing the performance of the new scheduling algorithm, we compared two management schemes. First, a full flexible scheme where no dates and no volumes were fixed (referred to as "flexible"). In the second, a simplified scheduling problem is solved, where the possible dates of the irrigation events were fixed at multiples of 10 days. In addition, only fixed irrigation volumes ($v_i = 50\,mm$) were allowed (referred to as "fixedDV").

3.1 The irrigation scenario

In a real case application a limited amount of 1 up to $72\,m^3$ water had to be distributed with optimal irrigation schedules to gain maximum crop yield. Detailed and mostly unpublished data of field experiments in Lavalette (France) regarding volumetric soil moisture content, ETP and other aspects of the experiments were kindly provided by Mailhol (2005) from *CEMAGREF* (France). In our study simulations by a water balance model (Rao et al., 1988) based on these experiments were carried out. In the irrigation scenario 4 different crops (maize, wheat, sunflower and tomato) are grown over a growing period of 132 days starting from May 26, 1999. The irrigated field is a plot of silty loam, characterized by a saturated soil moisture $\theta_s = 0.41$, a residual soil moisture $\theta_r = 0.05$, and field capacity at $\theta_{FC} = 0.4$. Values for the four crops for the development of the root zone, potential evapotranspiration (K_c-factor), crop sensitivity factors K_y and soil water depletion factor p were taken from Doorenbos and Kassam (1979).

3.2 The setup of the NDP algorithm

In neuro-dynamic programming the accuracy of the approximate cost-to-go function mainly depends on the number and the parameters of the chosen basis function, namely the radius of the Gaussian σ and their distribution in the state space. We fixed $\sigma = 0.1$ and considered only a variation of the number of RBF assuming always a uniformly spaced distribution of the RBF's centers. The amount of RBF was then fixed at 6x6x11 according to the dimensions of the state space which was an acceptable trade-off between accuracy and speed of training of the approximate $\tilde{Y}((V_i, \theta_i, t_i), \mathbf{W})$. The parameter λ in policy evaluation was set to 1 which leads to a supervised linear regression on the data of the simulated irrigation scenarios, i.e. the pairs of the simulated states and crop returns. In policy improvement we start with an initial value $\epsilon = 1$ and then decrease gradually with an increasing number of training steps n as $\epsilon = e^{-\frac{10*n}{n_{max}}}$. In the case of NDP only a single application of the policy iteration algorithm for each crop was necessary to generate an universal approximate cost-to-go function which allowed performing all the optimization runs for each of the prescribed management scheme.

3.3 Results

The crop production functions, which were generated by the NDP optimization using the approximate cost-to-go, are presented in Fig.3. For each crop the normalized seasonal production functions for the "flexible"and "fixedDV" cases are shown separate subfigures. Strong differences between the two management schemes in the achieved yield per water unit can clearly be seen in the case of maize (see Fig.3a). The production function under the "fixedDV"-scheme using the same approximate cost-to-go function shows a significant reduction in yield (up to 20 %) caused by the limitations of this management scheme. An exception can be seen in the lower part of the crop production function for water volumes below $200\,mm$. The 2nd nonlinearity-range moved from $300\,mm$ to $400\,mm$. This can be explained by the inflexibility of the management scheme, which does not always allow to irrigate in an adequate way when the stress sensitivity of corn is high. From a water volume of $550\,mm$ on, there is no more improvement in the yield if more water is applied. This implies that all the additional water is percolating because field capacity was already been achieved in all days when an irrigation is possible.

The crop production function for maize under the "flexible"-scheme is nonlinear in two ranges. The first range is in the vicinity of the point where all crop water requirements during a growing season are satisfied. The second range is between $200\,mm$ and $300\,mm$ of available water. At a water volume of $300\,mm$ the crop water requirements of the 3rd mid-seasonal period, which has the highest stress sensitivity are fully satisfied (which is $K_y = 1.3$ compared to $K_y = 0.4,\ 0.5$ in the other crop growth stages. The nonlinearity is due to a side effect caused by a more and more adequate irrigation of the 3rd period. The last growing period with a lower K_y and a higher allowable depletion benefits disproportionately from the water which is stored in the soil at the time of transition from the 3rd period to the 4th. The reduction in yield due to the management scheme of the other crops are not as significant as of maize (see Fig.3b-d). Here the different shape of the crop production function, especially for the one of tomatoes, is remarkable.

The results provided by an optimization using the approximate cost-to-go generated by the NDP algorithm are proved by a static optimization approach. For this task a high number of runs with a tailored global evolutionary algorithm (EA) were performed (Schmitz et al., 2007). Minor deviations of the NDP method were observed and are mainly due to the approximation error which could be reduced by an increased number of RBF. We also investigated the computational efficiency of the EA and the NDP algorithm on a Pentium-PC (1.3 GHz). For the EA one optimization run needed less than a $1\,min$ (convergence after maximum 1000 function evaluations). But it has to taken into account that one optimization run is necessary for each point of the crop production function. The computational effort of the NDP-Algorithm depends on various parameters. The LSTD-algorithm for the policy evaluation has a cost of O(N^2) for the accumulation of the TD's and O(N^3) for the matrix inversion where N is the

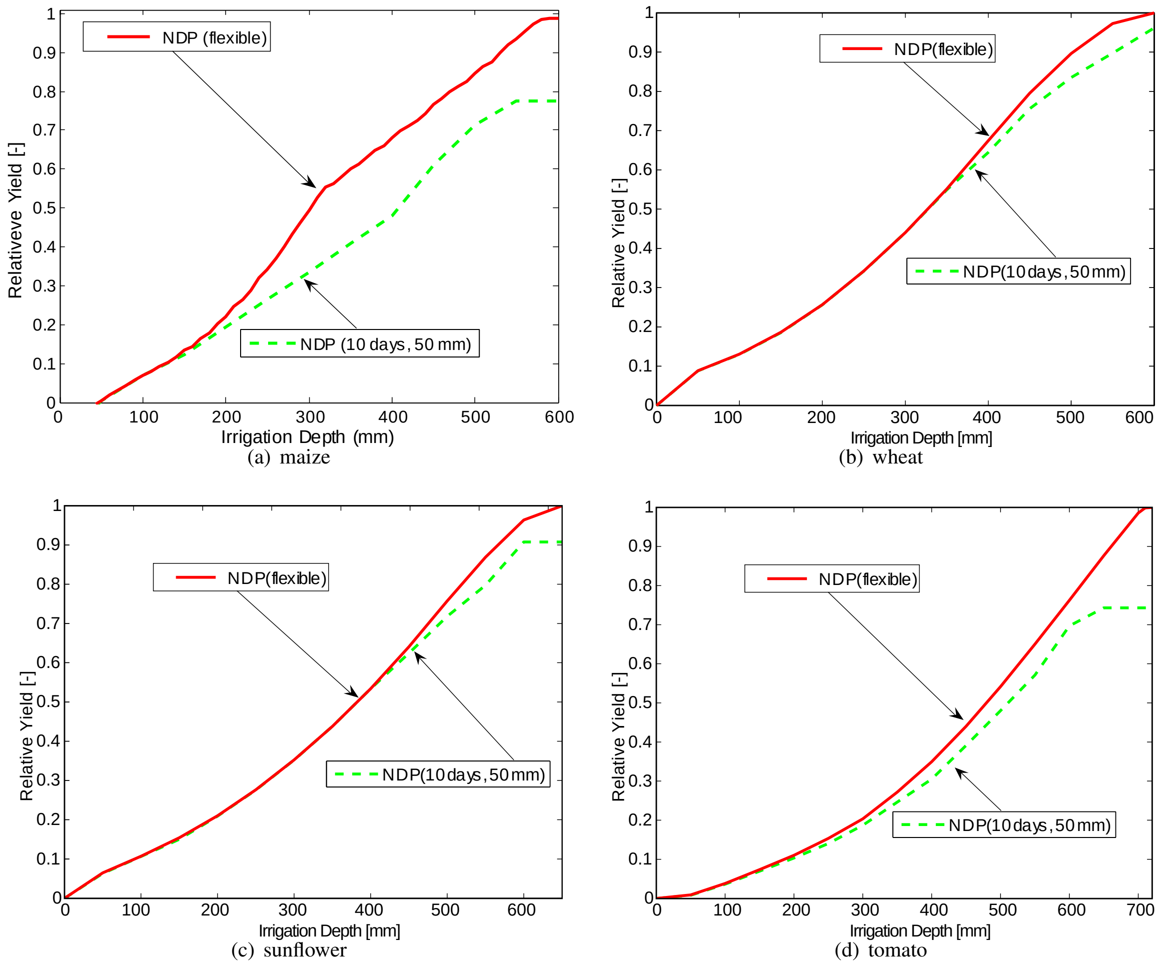

Figure 3. Crop production functions for maize, wheat, sunflower and tomato under different management schemes.

number of RBF. Overall, the time for learning or approximating the cost-to-go function was around 10 h and the application time needed less than a second. However, the NDP methodology offers improvement of the performance, taking into account that with the approximate cost-to-go function different tasks (different management schemes, different given amounts of volume etc.) can be performed with a single (expensive) approximation step.

4 CONCLUSIONS AND FUTURE WORK

We presented a new optimization algorithm for simulation-based optimization of scheduling under deficit irrigation throughout a whole growing season. The applied neuro-dynamic programming (NDP) algorithm for dynamic optimization has a wide range of application in irrigation operation. Once an approximate cost-to-go functions is calculated it can be used for irrigation scheduling under any arbitrary management scheme. The approximation of the cost-to-go function overcomes the “curse of dimensionality” but it still needs considerable time for the determination of the optimal weights of a linear basis function approximation of the cost-to-go by the policy iteration algorithm using LSTD. It is worthwhile to improve this method because dynamic optimization offers some advantages over static optimization including (1) feedback control which can respond immediately to external effects (e.g. rainfall), (2) stable performance even with model uncertainties or uncertainties of the initial or boundary conditions (e.g. climate conditions), and (3) reduced sensitivity to parameter variations. Thus, it is expected that new method may be generally applicable to management problems of water resources.

Future work will focus on the application of the algorithm under uncertainty of climate conditions and/or soil hydraulic parameters. In this context, NDP overcomes the "curse of modeling", which means that the transition probabilities do not have to be computed explicitly for stochastic dynamic programming. It uses the distribution of the random variables with no limitation placed on the stochastic model to simulate the system’s behavior. Further investigations are under progress which include more comprehensive irrigation models such as the furrow irrigation model (FIM) (Wöhling and Schmitz, 2006) in the optimization of deficit irrigation systems.

5 REFERENCES

Bellman, E. and Dreyfus, S. (1962), *Applied Dynamic Programming*, Princeton University Press.

Bertsekas, D. P. (2000), *Dynamic Programming and Optimal Control*, Athena Scientific.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996), *Neuro-Dynamic Programming*, Athena Scientific.

Boyan, J. A. (2002), Technical update: Least-squares temporal difference learning, *Machine Learning*, 49(2-3), 233–246.

Bras, R. L. and Cordova, J. R. (1981), Intraseasonal Water Allocation in Deficit Irrigation, *Water Resources Research*, 17(4), 866–874.

Doorenbos, J. and Kassam, A. (1979), Yield response to water, Technical report, FAO, 1979, Irrigation and Drainage paper No. 33.

Gorantiwar, S. D., Smout, K. and Vairavamoorthy, K. (2006), Performance-based optimization of land and water resources within irrigation schemes. i: Method, *Journal of Irrigation and Drainage Engineering – ASCE*, 132(4), 332–340.

Gosavi, A. (2003), *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*, Kluwer Academics Publishers, Boston, ISBN 1402074549.

Mailhol, J. (2005), Meteorological and hydraulic data from the Lavalette plot, personal communication.

Prasad, A.S., Umamahesh, N.V. and Viswanath, G.K. (2006), Optimal irrigation planning under water scarcity, *Journal of Irrigation and Drainage Engineering – ASCE*, 132(3), 228–237.

Rao, N. H., Sarma, P. B. S. and Chander, S. (1988), Irrigation Scheduling under a Limited Water Supply, *Agricultural Water Management*, 15, 165–175.

Schmitz, G., Wöhling, T., de Paly, M. and Schütze, N. (2007), Gain-p: A new strategy to increase furrow irrigation efficiency, *Arabian Journal for Science and Engineering*, 32(1(C))), 103–114.

Shang, S. and Mao, X. (2006), Application of a simulation based optimization model for winter wheat irrigation scheduling in north china, *Agricultural Water Management*, 85(3), 314–322.

Sunantara, J. D. and Ramirez, J. A. (1997), Optimal stochastic multicrop seasonal and intraseasonal irrigation control, *Journal of Irrigation and Drainage Engineering – ASCE*, 123(1), 39–48.

Sutton, R. S. and Barto, A. G. (1998), *Reinforcement Learning: An Introduction*, MIT Press.

Wöhling, T. and Schmitz, G. H. (2007), Physically based seasonal modeling of the 1d surface - 2d subsurface flow in furrow irrigation systems. I: Model development, *Journal of Irrigation and Drainage Eng.*, in print.