5-2012

On-Farm Water Management Game With Heuristic Capabilities

Mohammed Z. Shaban
Utah State University

Recommended Citation

Shaban, Mohammed Z., "On-Farm Water Management Game With Heuristic Capabilities" (2012). *All Graduate Theses and Dissertations*. 1255.
https://digitalcommons.usu.edu/etd/1255

ON-FARM WATER MANAGEMENT GAME WITH HEURISTIC CAPABILITIES

by

Mohammed Z. Shaban

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Irrigation Engineering

Approved:

Dr. Gary P. Merkley Committee Chair	Dr. Christopher M. U. Neale Committee Member
Dr. Andrew Keller Committee Member	Dr. Mac McKee Committee Member
Dr. Gilberto Urroz Committee Member	Dr. Mark R. McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2012

ABSTRACT

On-farm Water Management Game with Heuristic Capabilities

by

Mohammed Z. Shaban, Doctor of Philosophy

Utah State University, 2012

Major Professor: Dr. Gary P. Merkley
Department: Civil and Environmental Engineering

A modern computer-based simulation tool (WaterMan) in the form of a game for on-farm water management was developed for application in training events for farmers, students, and irrigators. The WaterMan game utilizes an interactive framework, thereby allowing the user to develop scenarios and test alternatives in a convenient, risk-free environment. It includes a comprehensive soil water and salt balance calculation algorithm. It also employs heuristic capabilities for modeling all of the important aspects of on-farm water management, and to provide reasonable scores and advice to the trainees.

Random events (both favorable and unfavorable) and different strategic decisions are included in the game for more realism and to provide an appropriate level of challenge according to player performance. Thus, the ability to anticipate the player skill level, and to reply with random events appropriate to the anticipated level, is provided by the heuristic capabilities used in the software. These heuristic features were developed based on a combination of two artificial intelligence approaches: (1) a pattern recognition approach; and (2) reinforcement learning based on a Markov Decision Processes approach, specifically, the Q-learning method. These two approaches were combined in a new way to account for the difference in the effect of actions taken by the player and action taken by the system on the game world. The reward function for the Q-learning method was modified to reflect the anticipated type of the WaterMan game as what is referred to as a partially competitive and partially cooperative game.

Twenty-two different persons classified under three major categories (1) practicing farmers; (2) persons without an irrigation background; and (3) persons with an irrigation background, were observed while playing the game, and each of them filled out a questionnaire about the game. The technical module of the game was validated in two ways: through conducting mass balance calculations for soil water content and salt content over a period of simulation time, and through comparing the WaterMan technical module output data in calculating the irrigation requirements and the use of irrigation scheduling recommendations with those obtained from the same set of input data to the FAO CropWat 8 software. The testing results and the technical validation outcomes demonstrate the high performance of the WaterMan game as a heuristic training tool for on-farm water management.

(165 pages)

PUBLIC ABSTRACT

On-farm Water Management Game with Heuristic Capabilities

by

Mohammed Z. Shaban, Doctor of Philosophy

Utah State University, 2012

Major Professor: Dr. Gary P. Merkley
Department: Civil and Environmental Engineering

Improved on-farm irrigation practices can result in more economical farming, and better productivity. Very little has been done with regard to improved training tools that can be used to promote better and more effective on-farm irrigation practices. Games considered as an effective decision support tools in which players are able to test alternatives, and demonstrate the effects of their decisions, in a short time, and without being afraid of making mistakes. Training tools in the form of games promotes what is called “learning based on experience” through a schematic version of reality, and observing the effects.

The WaterMan game was developed on the sense of being a training tool for on-farm water management, and to offers an interactive framework with different technical and operational options that allow the user to develop scenarios and test alternatives in a convenient environment. A very detailed, consistence, and robust technical model that reflect and respond to various alternatives was developed within the software. Heuristic capabilities were employed in the software, to provide more realistic modeling for the important aspects of on-farm water management, and to automatically analyze the performance of the player, based on optimal scenarios, to provide feedback and recommendations at the end of the play. Artificial intelligence capabilities were also included in the software to anticipate player level of skills in irrigation and to reply back with different random events based on the anticipated level to provide a potentially more challenging game play. These capabilities provide the unique characteristics of WaterMan as a game with unpredictable scenarios, thereby making it more challenging and more engaging, and an enhanced tool for learning.

Two options of game play were developed to accommodate different trainee requirements and interests: “quick play” and “play.” If the player chooses the “quick play” option, he or she will move directly to play with a predetermined set of input data. If the player chooses the “play” option, a new window for data input appears, and the player is asked to select from various options. The model has many options of crops, climatic zones, water delivery methods, irrigation methods, and soil texture, in addition to a flexible planting dates.

Twenty-two persons were asked to play the WaterMan game and to give their feedback. The majority of the players classify the game as an excellent training tool for on-farm water management with some very challenging random events. The technical module of the game was validated in two ways: conducting mass-balance calculations for the daily soil water and salt content, and comparing the game-generated results of irrigation water requirements and irrigation scheduling calculations with those generated by the FAO CropWat 8 software.

DEDICATION

To My Brother Abu Zuhdi
The Person Who Surrounded Me With His Kindliness

ACKNOWLEDGMENTS

I would like to express my appreciation and great gratitude to my major advisor, Dr. Gary P. Merkley, for his support, guidance, and encouragement throughout each step of the preparation of this dissertation. My thanks and appreciations are also extended to the committee members, Dr. Mac McKee, Dr. Christopher Neale, Dr. Andrew Keller, and Dr. Gilberto Urroz, for their advice and suggestions in developing the game, and for their discussions, patiently reading, and contributions in the editing of this dissertation.

Deep gratitude goes to all my family members. To my wife and my best friend, Nawal, for her encouragement, constant support, and patience throughout our life together and during this critical period of my life as I worked my way through the completion of this dissertation. To my children, Ammar, Loui, and Laith, who give my life a meaning and a reason for continuation and achievements. For my brothers and sisters who surrounded me with their love, encouragement and support all the time, and never lost their faith in me. Making all of you proud of me is my ultimate goal.

I would like to thank the twenty-two persons who tested the game and provided their valuable comments to me, as well as all people with whom I collaborated during the completion of this work and made this project possible.

I would also like to extend my thanks to all my friends and colleagues who made my life easier through their support and presence whenever needed. I could not have done it without all of you -- Thanks.

Financial support for this project came from the Utah Water Research Laboratory. I am grateful for this support that helped me accomplish this work.

Above all, I thank God for supporting me with power, patience, and determination throughout each step of the preparation of this dissertation.

Mohammed Z. Shaban

CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The global demand for fresh water is progressively increasing as the demand for industrial and domestic water supplies increase due to population growth, economic development and climatic changes (Feitelson et al. 2007; Ritchie and Basso 2008). Sustaining an adequate amount of high-quality water has become an important and pressing issue; thus, it is no wonder that many countries have been spending millions of dollars over the past several decades for the development of new water resources. But the effectiveness of this approach has greatly diminished. Instead, improved management of the existing resources can be more feasible to secure needed water.

An understanding of agricultural water requirements is a critical input in resolving water resources issues. Worldwide, agriculture consumes approximately 70 percent of available water resources, with estimated overall efficiency of only 30-40 percent (Molle and Berkoff 2006). The growing demands on the existing water resources necessitates that the agricultural sector improve its water management. Even moderate improvements in agricultural management could free huge quantities of good quality water which can be used directly by other water user sectors (Ritchie and Basso 2008).

Much of the emphasis and resources toward dealing with the water scarcity problems in recent years have been dedicated to infrastructure and technological improvements, as well as organizational and institutional changes. These measures alone are not enough to significantly improve water management, unless they are accompanied by better and more effective on-farm irrigation practices. Improved on-farm irrigation practices can result in more economical farming, and better productivity. However, the effective implementation of water resource management options is dependent on broad acceptance by all relevant stakeholders, especially those considered the main actors in the water resources management sector. The main actors are generally farmers, irrigators, and irrigation and drainage agencies (Schultz et al. 2005).

An extensive educational program may be required to correct the occasionally excessive use of irrigation water by farmers (Johnston et al. 1991). Despite the information availability, experience shows that an educational program is necessary to teach the actors in the field of agricultural water how to manage their water resources in a better way. Very little has been done with regard to improved training tools that can be used to promote more complete understanding of the problems faced by farmers and irrigators, and the difficulty of the operational decisions they face with respect to water management. Successful performance requires effective communication between all individuals related to the operation and management of an irrigation system (Skogerboe and Merkley 1996). Simply providing handouts and other written materials to them is insufficient. It may be more efficient to teach them in what is called “learning based on experience” through a schematic version of reality, and observing the effects.

Games considered as an effective decision support tools in which players become mutually dependent decision makers (Ubbels and Verhallen 2000, cited in Lankford et al. 2004). Gaming simulation provides the mean to test the consequences of a decision without the need to use or jeopardize the system that it is testing (Burton 1989). Simulations and role-playing games allows the participants to demonstrate the effect of their decisions on the system in a short time, without being afraid of making mistakes (Clarke 2004).

Through intelligent and heuristic simulation tools in the form of a game in which the effect of decisions can be seen, a great deal of understanding of the parameter and variable interrelationships for a variety of situations can be attained in a much shorter time that it would take (many years) by field experience alone. This understanding can lead directly to improvements in the on-farm water management.

The proposed game will be a training tool to teach the actors in the field of agricultural water (farmers, irrigators, canal operators, and students) how to better manage their water resources in what is called “learning based on experience” through a schematic version of reality, and visualizing the effects. Therefore, this research project developed a software application in the form of a game for simulating different technical and operational aspects of on-farm water management, and automatic analysis of the results to provide feedback to the trainees. This training game can be useful throughout the world regardless of economic and cultural differences.

The principal objective of the research is to develop a modern software-based simulation tool for on-farm irrigation water management, which can be used in training events for farmers and irrigators. The specific research objectives are enumerated below:

1. To design an educational and training tool for personnel involved in the management and operation of on-farm irrigation water to help them better understand and manage their water resources and to actively react to realistic scenarios. The target audience includes farmers, irrigators, and students;

2. To develop and test the software application in the form of a game for simulating different technical and institutional aspects of agricultural water resources management, with automatic analysis of the results to provide feedback to the trainee(s);
3. To produce the software application using a modern graphical interface and a modern computer programming environment; and,
4. To produce a users' manual and technical reference for the software application to facilitate application by individual trainees and groups of trainees.

CHAPTER 2

LITERATURE REVIEW

2.1 Water Resources Management

Improvements in on-farm water management may require substantial changes in the operation and physical control of water within a district (Johnston et al., 1991). García-Vila et al. (2008) conducted a study aimed at characterizing the behavior of an irrigated district over an area of 700 ha in southern Spain. They concluded that the lack of improvement in water productivity, the low irrigation water usage, and the changes in cropping patterns indicated that performance trends in irrigated agriculture are determined by a complex mix of technical, economic, and socio-cultural factors. Popova and Pereira (2008) performed simulations for the present and scenario-built weather conditions that include a pessimistic scenario of precipitation decrease over the next 25 years. In their study, the irrigation scheduling simulation model ISAREG was calibrated for two maize varieties**.** They concluded that the results of simulations do not allow selecting one among the other alternatives as the best irrigation scheduling strategy, but are useful for creating of an information system for farmers using actual weather data.

With the rapid development of microcomputer technology, many irrigation models and software are now available to improve the coordination process between the scheme manager and the farmers, and help scheme managers' in their water allocating decisions to meeting farmers' water demand. However, these techniques show various limitations when addressing strategic issues such as the interaction between farmers' water demand and the scheme manager's supply. Accordingly, optimal solutions are difficult to find due to: (a) the large number of variables to be considered; and, (b) the different interests, objectives and strategies of the different stakeholders, especially farmers. Therefore, the decision-making processes should be based on negotiation leading to trade-off solutions. To support this negotiation process, simulation tools provide a suitable answer as they allow: (a) producing a clear representation of the current management context that can be shared by all stakeholders; and, (b) producing expected information about the future of the scheme, e.g. by designing and comparing various ‘‘what-if’’ scenarios (De Nys et al. 2008).

2.2 Stakeholder Involvement and Decision Making

It is important to help people understand the dynamic nature of sustainability attributes and to better address the issues, tradeoffs and conflict associated with sustainable management of natural resources (García-Barrios et al. 2008). Van Paassen et al. (2007) mentioned that the capacity to identify options for sustainable and equitable development depends on the acquisition of knowledge and skills for: (a) complete analysis of the biophysical system dynamics; (b) analysis of the multiple positions, perceptions, values, beliefs and interests of the stakeholders; and, (c) consideration of the action needed to fill the gap between the desired socio-technical system and the real-world situation.

Social learning, also sometimes called co-learning, is an approach in which the participatory use of tools plays an important role. Social learning requires the design of processes that promote shared learning from experimentation and adaptation of organizational and individual procedures, standards, and behavior. Stakeholders co-

construct the identity of the resource management problem in ways that help them move towards a shared understanding and to learn their way toward solutions. It involves changes in norms, practices, and behavior, as well as changes in perception and understanding among stakeholders (Roling et al. 2004).

2.3 Games as Training Tools

Simulation and management games can provide a means to direct thinking, illustrate complex inter-relationships and weight priorities. The use of simulation and games have made them valuable tool and teaching aid. This includes a role in research, education, and training. Simulation and games are particularly effective in describing underlying processes, identifying issues and simulating discussion. They provide a reasonable tool for understanding the behavior of human beings and for training people to adapt to extreme situation (Smith 1989; Kos and Prenosilova 1999; Clarke 2004)

The advantages of games and simulations over other learning techniques can be significant. Carter (1989) listed three particular important advantages:

1. The opportunity for the trainee to face complex situations which encourage him or her to actively seek or develop problem-solving techniques;
2. The possibility of testing decisions in an environment without fearing the results (whereas in reality some decisions may seriously affect people's life or livelihoods or be very expensive); and,
3. The ability of a game to combine techniques or rules that are usually taught in isolated packages.

In their paper, Jonoski and Harvey (2004) mentioned that computer-based simulation games of the SimCity kind make use of the increasing processing power of computing hardware to provide game play based around real time simulation of actual or hypothetical realities. The game play depends on the evolving properties of connected simple simulations. These games can provide an environment that supports learning about complex systems and evolving properties in general, and about particular classes of systems. The authors also pointed out that the use of realistic, even if hypothetical, scenarios enables such games to help the transfer of knowledge, while the removal from a real situation in which fearing the consequences is eliminated allows the unrestricted exploration on novel ideas. In fact, bringing together a variety of people with different interests brings in a variety of knowledge.

Training games includes role-playing games as well as computer-based learning simulations. Using variety of simulation and role playing exercises enable the trainees to explore the consequences of decision making based on limited information and to appreciate that similar trainees may response differently under similar conditions. Computer-based simulation games have the advantage that they require less time to run than role-playing simulations. However, computer-based games lack the interpersonal relations and interactions, which can be important in role-playing games (Clarke 2004). Although, role-playing games might practice issues of cultural concerns with social relations (Barreteau and Daré 2007) which are not considered in computer-bases games.

Clarke (2004) listed the following elements that must be contained in a game:

1. Relevance: the game must be of interest to the trainee and reflect his/her needs, so that he/she can understand it easily and benefit from it;
2. Simplicity: the game should be set in a simple, and clear form so that it does not confuse the trainee when playing it;
3. Realism: avoiding complexity, the program should produce realistic results and applied recommendations that could be easily understood and implemented by the player;
4. Interaction: rapid response, different alternatives, and good use of visual effects will attract the player's interest;
5. Flexibility: the ability of the program to modify itself in response to the user needs. To predict all possible actions and reactions within the simulation may be difficult, but allowing for as many as possible of them gives the game more flexibility in response to user needs and the game goals;
6. Excitement: to be a game, the simulation should be stimulating. Therefore, challenging alternatives, the good use of random events, and the visual effects in the game will add more enjoyment; and,
7. Discussion: in order to achieve the training goals and to explore the different experiences and the decision-making processes that individuals undertook, a group de-briefing discussion is recommended once the simulation is completed.

2.4 Irrigation Management Games

This section of the literature review describes several irrigation management games that have been developed by different individuals and groups over the past few decades. Although each game has its own unique features, there is some degree of overlap among them.

The Green Revolution Game This role-playing game simulates the life of rural farmers in South Bihar province, India. It addresses the issues of variable strategies to obtaining high yield of rice varieties (the concept of high inputs-high outputs). The game introduce players to the conflicting decision made on survival farmers and explores the different strategies that can be adopted taking into considerations the uncertainty of agricultural inputs such as labor, water and fertilizer. The game is designed to be played by a group of 12-24 participants, and takes a minimum of 6 hours to run and can run over 2 or even 3 days. The aim is to imitate the growth of rain-fed rice in an area where rainfall is variable and for the participants to develop strategies for dealing with this uncertainty in addition to other uncertainties related to other inputs such as rice varieties, fertilizers and labor availability (Chapman 1982, cited in Clarke 2004).

This game combines a set of dimensions: a dynamic physical environment, agronomy, sociology, politics, and a simple economy. Because of this general set it has proved its usefulness to a wide number of disciplines and professional groups. For example, it is used by bankers in India to train rural bank managers to understand the problems of small farmers, and by political scientists in the UK who are interested in small-group dynamics (Chapman 1989).

The Juba Sugar Estate Game This role-playing game is described by Carter (1989). The game is suitable for 3-20 participants. It is based on the Juba Sugar Estate in Somalia and it addresses the issue of management under scarce inputs (water and fertilizers) and resources (labor, capital equipment, money and fuel). Each participant takes a role within small management teams and asked to make decisions concerning fertilizers application, irrigation, maintenance, harvest, and cane haulage. The game is aimed to introduce the participants to the complex interactions among resources, inputs, activities and management decisions. The game shows evidence of team work enhancement and increased mutual understanding of privileges job functions. The Juba Sugar Estate game focused on the logistics and prioritization of resource allocation and thus has been proven to be more useful for managers than for irrigation engineers.

The River Basin Game The River Basin Game was devised by Bruce Lankford in 2000 at the University of East Anglia, United Kingdom (UK) to teach undergraduate students the principles of common property resource management as applied to surface water. The game shows students that water-claiming strategies result in certain members of the community gaining while excluding others (Lankford et al. 2004).

The River Basin Game, as described by Lankford et al. (2004), is a dialogue tool for decision-makers and water users; it has been tested in medium to small catchments in Tanzania. The game consists of a large wooden board that physically represents the catchment and reflects a central river flows between the upper catchment and a downstream wetland, and has several intakes into irrigation systems of varying sizes. The game is played by allowing glass marbles to flow down the channel as a represent of river water. Participants are asked to place small sticks acting as weirs across the river to trap the marbles and divert them into irrigation systems where they sit in small holes, in order to meet the water requirement of that particular plot of rice or irrigation activity.

The players practiced that being at the top of the river has advantages of water availability as compared to the tail-end systems that tend to experience water shortages. The game allows stakeholder groups to evaluation different management strategies.

The game shows evidences of mutual understanding among different people's who have different access levels to water and also teaches participants to understand the effect of their decisions on the other users within the same catchment. This promotes more understanding to others needs and the concept of sharing same resources. The authors' experience shows that participants often become highly animated and, by the end of the game, have a good understanding of system dynamics, common-property pitfalls, and how to effectively react to scenarios. The authors suggested that if the game-playing is part of a workshop is to be spread over two days at which the second day to be use for discussing the lessons learnt and bring together various scenarios and institutions to assist improving the equity of supply for all users.

<u>The Wye College Irrigation Game</u> This irrigation management game, called *"Stop the Breach,"* was described by Smith (1989). The game is a mixture between a role-playing game and a computer-base game. Participants in the game are requested to act as farmers or as scheme managers and asked to make decisions regarding the operation of a scheme which are then processed by a computer and shows various conclusions. The purpose of the game is to provide experience about the complexities of a real irrigation management decision-making, and the interrelationship between farmers and scheme managers. It also allows the participants to practice in the application of basic theoretical concepts and methods that are relevant to the successful operation of an

irrigation scheme. Smith (1989) described some experiences with the game concluding that the game can achieve its training objectives, stimulates discussion of the different factors that affecting the success operation of irrigation schemes, and has the potential for further development.

The Irrigation Management Game (Classroom Version) This game was introduced in 1982 by Burton and Carruthers at Wye College, University of London. There have been many changes to the game since then. This game as described by Burton (1989, 1994) is a role-playing exercise primarily developed for the training of irrigation engineers and scheme managers. This game is set on an Indonesian run-of-river irrigation scheme. The game is designed to be played by between 10-26 participants. The game managed with two trainers: the game controller and assistance as trader. The game takes 6 hours to play with a 1-hour debriefing and discussion at the end. Participants are requested to choose the role farmers or irrigation agency staff.

In the game, an irrigation system scheme is laid out on a 3 x 2 m wall poster, showing eight tertiary units supplied by a main canal on a run-of-the river system. Each tertiary unit is divided into four blocks. Each block can be planted with one crop; either rice, maize or soybeans. The participants playing as farmers are dependent on irrigation water supplies from the main canal system, and are responsible for selecting and planting crops and schedule water supplies within their tertiary units based on the supplies received. The participants playing as agency staff are responsible for allocating the water available at the main canal intake between the tertiary units.

The Irrigation Management Game (Computer Version) This is a computer version of the Irrigation Management Game which aims to fulfill the objectives of the original (classroom) game in the form of a computer simulation (Clarke 2004). This version was designed to be played by one person, who is taking the role of a farmer and asked to choose one tertiary unit to farm. To represent the irrigation scheme management, the behavior of the other farmers and the irrigation agency staff (players in the classroom version) is simulated by the program. The computer version of the game is expected to be operated in a shorter time compared to the classroom version, although it lacks the interaction and social relations with other players.

The game as described by Clarke (2004), allows the trainee to play up to five growing seasons in a game run. This normally takes between 45 minutes and two hours, depending on trainee speed and familiarity with the game. The player asked to choose the crops to be planted in each of the four blocks in his/her tertiary unit. Each crop has three growth stages during the growing season. Each growth stage has its specific data on crop water requirements, rainfall, and water availability. The computer allocates a volume of water for the growth stage and asked the player to decide on how to distribute it amongst his/her crops. Information on crop yield and crop prices is available to assist in this allocation. Numerical calculations of irrigation water requirements, crops stress, and crop yield are carried out by the computer. The game can be played as many times as the player like, and the player can choose the number of seasons to play each time.

Irrigation Management Simulation Game (Irrigame) The irrigation management simulation game as described by Parrish (1982) is a computer-based simulation game that gives the participant the chance to experience a full season of on-farm irrigation management. The game is structured to allow the trainees to schedule their irrigations using weather predictions based on historical data, and using a soil moisture budget. The trainees can use their experience in taking all decision related to the management of their irrigation water. Based on the decisions taken throughout the season, the game simulates their irrigation efficiency, their effective use of rain, and their yield. As a result of this experience, it is expected that the trainees can then decide about the decisions they made and whether or not they were able to manage their field properly, and how they were able overcome difficulties caused by weather forecasting failures, rain when it was not needed by the crop, a semi-demand system that may or may not have met their needs, and system failures such as canal breaks. As stated by the author, this game was designed to reflect the problems, responsibilities, and consequences of irrigation project water management in a convincing and realistic manner.

Aquavoice Jonoski and Harvey (2004) introduced the concept of a first prototype of the Network Distribution Decision Support System (NDDSS), which has been developed in the form of an educational role playing game named “Aquavoice”. In the game, players were asked to take the roles of stakeholders in a decision regarding water abstraction plans. Based on the simulation results from hydrological and economic models, the institutional players asked to judge the performance of each scenario against their particular interest and can evaluate scenarios by providing weights indicating their relative values with respect to the main issues, or they can raise their own new issue. The judgment engine built in the simulation provides aggregated scores for the different scenarios, which can be updated in real time as players adjust their evaluations.

2.5 Heuristic Simulation

In the Encarta Dictionary (www.bing.com/Dictionary), it is mentioned that "heuristics" in the computer field describes a computer program that modifies itself in response to the user, e.g. a spellchecker, while in the education field "heuristic" is described as using a method of teaching that encourages learners to discover solutions for themselves. In both definitions, an intelligent and self-learning meaning can be abstracted.

In their book, Russell and Norvig (2003) mentioned that artificial intelligence is one of the newest sciences and is considered a universal field due to its existence in many other fields, includes, speech recognition, data mining, pattern recognition, statistics, machine learning, probabilistic reasoning, robotic, computer vision, and knowledge representation. Also, they pointed that AI has advanced more rapidly in the past decade because of the greater use of its methods in many applications ranging from general purpose areas, such as learning and perception to a specific tasks as playing chess, or diagnosing diseases.

As part of AI methods, Kaukoranta et al. (2003) discussed pattern recognition in the context of computer games and its role in extracting relevant information from the game environment, cluster this information into patterns, and pass this information to the decision making system to choose appropriate action. They stated that, in real-time strategic games, recognizing the behavioral pattern of a human player is very important aspect in implementing a challenging computer opponent. Through controlling how much information is provided, pattern recognition system can be formed of several levels of detail, and can be employed in designing different levels of difficulty to match a human player with varying skills.

Following recent probabilistic techniques in user modeling, Hui and Boutilier (2006) used a dynamic Bayesian network approach to model and infer a human user's “type.” They stated that an encouraging results were obtained from using this approach, but concluded that they are examining the construction of a decision policies using partially observed Markov decision processes (POMDP), in an effort to utilize the sequential nature of human-computer interaction. Fern et al. (2007) presented and evaluated a theoretical decision framework that observes a goal-directed agent in order to select assistive actions that minimize the overall cost. They modeled the environment as a POMDP, where the hidden states are related to the agent’s unobserved goal. The approach was evaluated on two domains where human subject asked to perform tasks in game-like computer environment. The results show a substantial reduction in user efforts with only minimal computational efforts.

Learning from observation is a skill well mastered by human beings (Stensrud and Gonzalez 2008). The idea of using systems that can learn to solve problems became popular in the artificial intelligence field (Barrios-Aranibar and Gonçalves 2009). The theory of Markov decision processes (MDPs) underlies much of the recent work on reinforcement learning (Littman 1994). Reinforcement learning has become one of the most active research areas in machine learning, artificial intelligence, and neural network research. Reinforcement learning is learning what decision to make, how to link situation to actions with the purpose of maximizing a numerical reward signal. The learner is not told which actions to take, as in most form of machine learning, but instead must try all actions applicable in a situation to discover which actions yield the most reward (Sutton and Barto 1998). Reinforcement learning algorithms calculate a value function for action or for state-action pair, and use observed rewards to learn an optimal policy for the environment that maximize the expected total reward (Russell and Norvig 2003). Among reinforcement techniques, Q-learning is one of the most commonly used algorithms and has a strong foundation in the theory of Markov decision processes. It is also easy to use, and has been widely applied in several applications like learning in robotics, communication systems, trading, and others (Hu and Wellman 2003).

CHAPTER 3

MODEL DEVELOPMENT

The design and development of a computer-based training tool (or model) in the form of a game that can be used to analyze both strategic and operational issues related to the management of irrigation water resources is described herein. WaterMan is the name used herein for this game, meaning "Water Management." The WaterMan game utilizes an interactive framework, thereby allowing the user to develop scenarios and test alternatives in a user-friendly environment. It employs heuristic capabilities in a simulation approach for modeling all of the important aspects of on-farm water management that are essential to effective tactical and strategic planning.

3.1 Features of the WaterMan Game

The game was developed using the Visual Basic .NET programming language in Microsoft Visual Studio 2008. The game has the following target audiences: farmers, irrigators, irrigation extension specialists, and students. Two game play options were developed to match different trainee requirements and interests. The model includes the following options:

1. Distribution system delivery methods: fixed rotation, and on-demand.
2. On-farm irrigation methods: surface, sprinkler, and localized (trickle).
3. Water quality: different salinity levels.

The software consists of three modules: (1) the technical module, which is considered the “brain” of the game; (2) the scenario-based module, representing the user interface, and including the heuristic algorithms outputs; and, (3) the scoring and recommendation module, which provides an overall evaluation of the decisions taken by the player at the end of a simulated irrigation season (Fig. 3.1).

The technical module uses a database containing the input data (parameters) that are provided to the program (software) by the player in the scenario-based module. The scenario-based module mathematically analyzes the decisions and reactions made by the player, based on the different events, and automatically composes a scenario-based (heuristic) simulation. Random events are generated according to the evaluation of the player type by the artificial intelligence method encoded in the program (see Chapter 4). Based on the tactical decisions taken in response to the different random events, a sequence of results is obtained. Processing a comparison between the results obtained from the scenario-based module with that obtained from the technical module (the reference results), through an optimization function search algorithm, enables the scoring and recommendations module to evaluate the decisions made by the player. In terms of results scoring, the player will have a certain set of goals or objectives to meet: maximize crop yield, maximize production function, or maximize profit. The scoring results will be based on the achievement of these objectives. After a simulated irrigation season, the program summarizes the overall decision implications (scoring), and makes suggestions for improvement and/or other optimal scenarios.

The technical module includes a comprehensive algorithm that has been developed to calculate soil water and salt balances in a crop root zone, and it uses a daily time step. The algorithm is described in detail in the following sections.

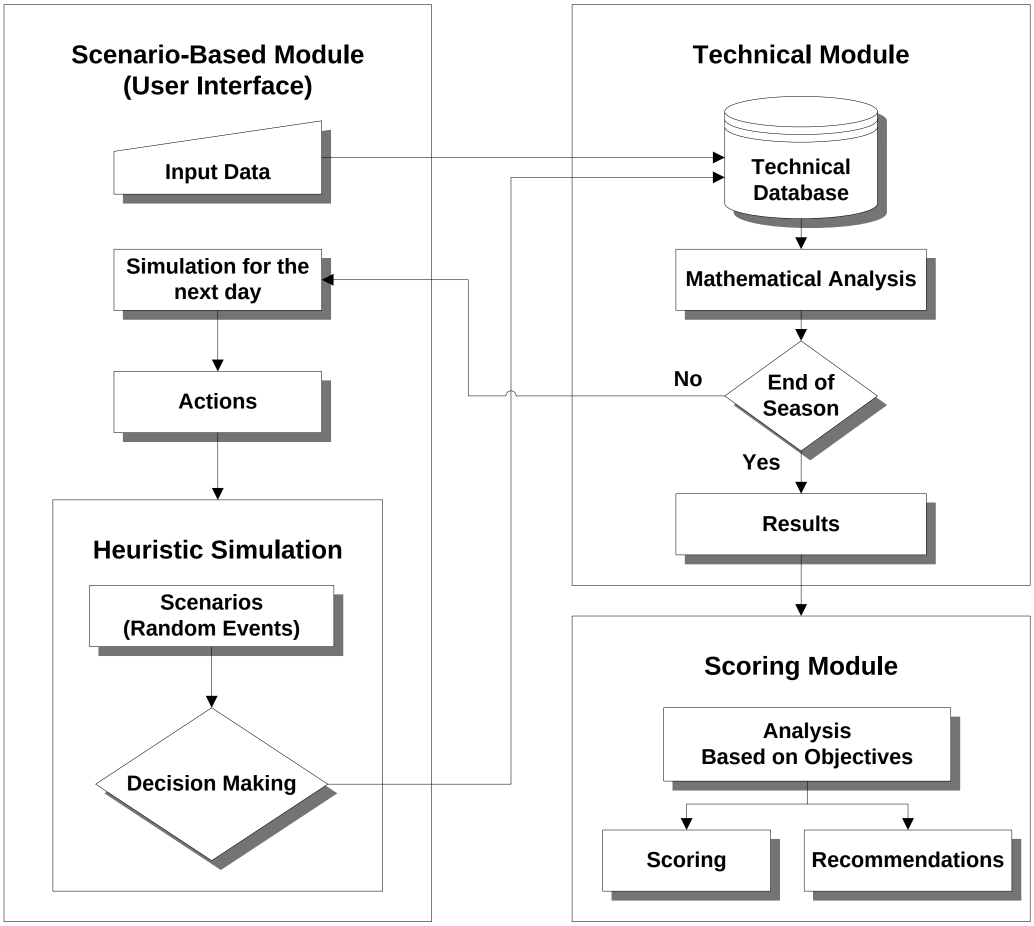


Fig. 3.1. Schematic diagram of the simulation model

Random events (both favorable and unfavorable) and their effect on crop growth, phenological stage sensitivity, best management practices, and overall agricultural productivity and profitability are also included in the software. The kinds of random events are: unexpected rain, sudden change in air temperature (weather), canal breaks and pump/motor failures (water supply interruptions), unexpected increases in the available water supply (when it was previously constrained), sudden failure of the on-farm irrigation system, temporary electrical outages, labor strikes, water theft (effect on available quantity and flow rate), unexpected decreases in the available water supply, and possible use of saline water. Each of these random events has its own impact on the irrigation decision, but they are not linked in their effect on each other.

The game has various options for making strategic management decisions. These options are presented to the player in the form of random events. For example, the player can choose to invest in buy or rent system parts, use other water source with lower qualities. The software also gives the player the option to purchase additional water shares (quantity) from other water users, or to sell the unexpected available water to other users.

At the beginning of the game, the player is asked to choose one of the two play mode options offered by the game: “quick play” or “play.” If the player chooses the “quick play” option, he/she will move directly to play with a predetermined set of input data. If the player chooses the “play” option, a new window for data input appears. In the data input window under the “play” option, the player is asked to select the desired climatic zone from seven options. The player has the following climatic-zone options based on Keoppen’s climate classification (www.blueplanetbiomes.org/climate.htm):

1. tropical moist;
2. temperate;
3. mountains;
4. continental;
5. Mediterranean;
6. semi-arid; and,
7. arid.

The planted crop(s) can be chosen from a list of 25 different crop types as found in Doorenbos and Kassam (1979). The player can choose from five different on-farm irrigation methods: furrow, border, basin, solid-set sprinkler, and drip irrigation. The player can also choose from one of two water delivery options: on-demand, and fixed rotation. Crop phenology, such as initial and maximum root depths, crop spacing, crop harvesting date, potential productivity (crop yield), market price of the product, and threshold soil water salinity (EC_e) are fixed by the program. The cost of the irrigation method, the cost of agronomic inputs and labor, the delivery system flow rate, water table depth, and irrigation supply and groundwater salinities are also set automatically by the program.

In the play option, the game allows the player to manage a 10-ha farm consisting of four different irrigated sections, a single crop type and a single soil texture class. The player is allowed to choose one on-farm irrigation method for all four sections. If the player decides to irrigate part of the farm due to water shortage or for any other reason, he/she has the option of choosing which section is to be irrigated at each irrigation event. The player has the option of specifying the planting dates, but the harvest date is set by the game, as mentioned above.

In the quick play option, the player is asked to manage a 10-ha farm, with four different irrigated section, same as under the play option except that all the input options are preset in the game and the player has no other choices. The climatic zone is set to "Mediterranean", the total the delivery method is set to fixed rotation with a 15-day delivery interval, the border irrigation method is used, the soil texture is “medium,” the planted crop is spring wheat with April 15 as the planting date, and all the other phenological characteristics of this crop. This option was added to the game in order to offer an introductory way to explore the game for players without a strong background in farming (beginners).

After completing the data input tasks, the simulation window is displayed in which the computer-player interaction starts and the artificial intelligence coding method is activated to detect the player type, based on the player decisions. In this window, the total available water for the entire season is specified by the program, with the option of increasing/decreasing this quantity by certain random events which may or may not occur. The daily available water quantity for the remainder of the season is schematically made available to the player.

A five-day weather forecast, options for irrigation duration, and irrigation water quantity, are made available for the player to make management decisions. Evaluation of the player's performance by the game's artificial intelligence system occur based on the player's reactions to the random events and the decisions options he/she has made. The generation of random events is adjusted by the program to meet the evaluated player type.

A dynamic sketch showing the daily cropping conditions, based on the decisions taken by the player, is continuously presented in the simulation window. The sketch includes information about daily soil water balance, soil water excess or shortage, daily root growth, daily expected relative yield, plant growth conditions; animation of crop performance, whether it is performing well or shows symptoms of stress, whether the crop is still alive or has died, and so on.

After the end of the growing season, the game will display the final window. In this window, an economic analysis of the cropping season is presented based on the crop yield, production cost, and on-farm water use indicators. A final score based on the overall consequences of the decisions which were made, in addition to recommendations for management improvements, are presented to the player.

The following sections describe the details of the three different modules included in the program.

3.2 The Technical Module

3.2.1 Software Database

Binary files containing plant types, soils, and climate zone parameters data were embedded in the program (software). The parameters contained in each binary-formatted file are mentioned below. The technical model runs scenario-based simulation uses a database containing the crops, soils, irrigation method, and climate zones parameters data. When the game initializes, the software reads the data from the database, and keeps all information available for further uses by the technical model. The input data by the player are also saved in arrays to be available for the technical model when running a simulation.

The software database contains crop phenological data for 25 different crop types as found in Doorenbos and Kassam (1979), and Allen et al. (1998), such as crop growing season, length of each of the four crop growth stages, three single crop coefficient values

(Kc initial, Kc mid, and Kc end), initial and maximum root depths, depletion fraction (p), yield response factor (Ky) for each growth stage, potential productivity (crop yield), market price of the product, and threshold salinity of the soil water extract (ECe). In addition, data for three types of soil texture classes (light, medium, and heavy), such as the soil water content at field capacity (FC), the soil water content at the permanent welting point, the soil water content at saturation, the residual soil water content at the beginning of the growing season, the capillary rise height, and the saturated hydraulic conductivity (K_{sat}) are also included in the program. Weather data for the seven climatic zones (as described above) were also made available in the program. Five on-farm irrigation method characteristics, such as cost of the irrigation method, application efficiency, and minimum water delivery interval are fixed by the program. The cost of agronomic inputs and labor, the delivery system flow rate, the water table depth, and the irrigation supply and groundwater salinities are also coded into the program or read from binary data files included with the program.

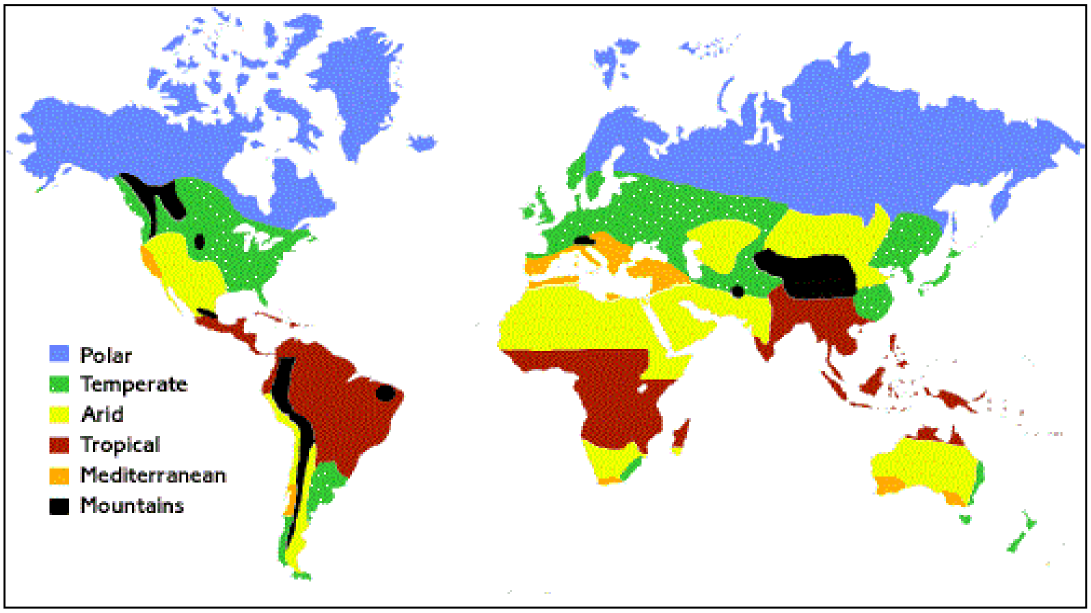
As mentioned above, seven climatic zones are available to the player to choose from. This requires seven different sets of weather data to be available in the program database. To make the game more realistic, long-term real weather data for all climatic zones are required. In addition, and in order to make the game more interesting, new sets of weather data are made available each time the game played, so the player cannot simply memorize the weather patterns and gain an “unfair” advantage. The following section describe the procedures followed to obtain the different weather data sets, and the next chapter shows the procedure of generating a new weather data each time the game played.

3.2.2 Weather Data

Seven major climatic zones as found in the world, are, namely: Tropical, Temperate, Mountains, Continental, Mediterranean, Semi-Arid, and Arid. These zones were identified based on the map shown in Fig. 3.2 (http://www.geography.learnontheinternet.co.uk/topics/climatezones.html#zones). The Polar climatic zones were neglected because they present extreme conditions for agricultural production. Instead, a continental and a semiarid climatic zone were added based on the existence of these zones in Keoppen's classification (www.blueplanetbiomes.org/climate.htm).

Global daily climate data records for precipitation, minimum air temperature and maximum air temperature for 1950 to 1999 time period discussed in Adam and Lettenmaier (2003), and Maurer et al. (2009), were used in the model. The climatic data used has 0.5-degree spatial resolution. The spatial extent of data ranges from 179.75W to 179.75E in longitude, and from 55.25S to 55.25N in latitude (www.engr.scu.edu/~emaurer/data.shtml).

After identifying the climatic zones spatial extend, a set of points within each climate zone was identified (Fig. 3.3). An average daily value for each climatic variable was calculated over the climatic zone gridded points for the whole period of record (1950-1999). The output is a one-year daily record of the specified parameter (e.g. minimum and maximum air temperatures) over the selected climatic zone. The daily



(Image courtesy of the UK Meteorological Office)

Fig. 3.2. World classification of climatic zones

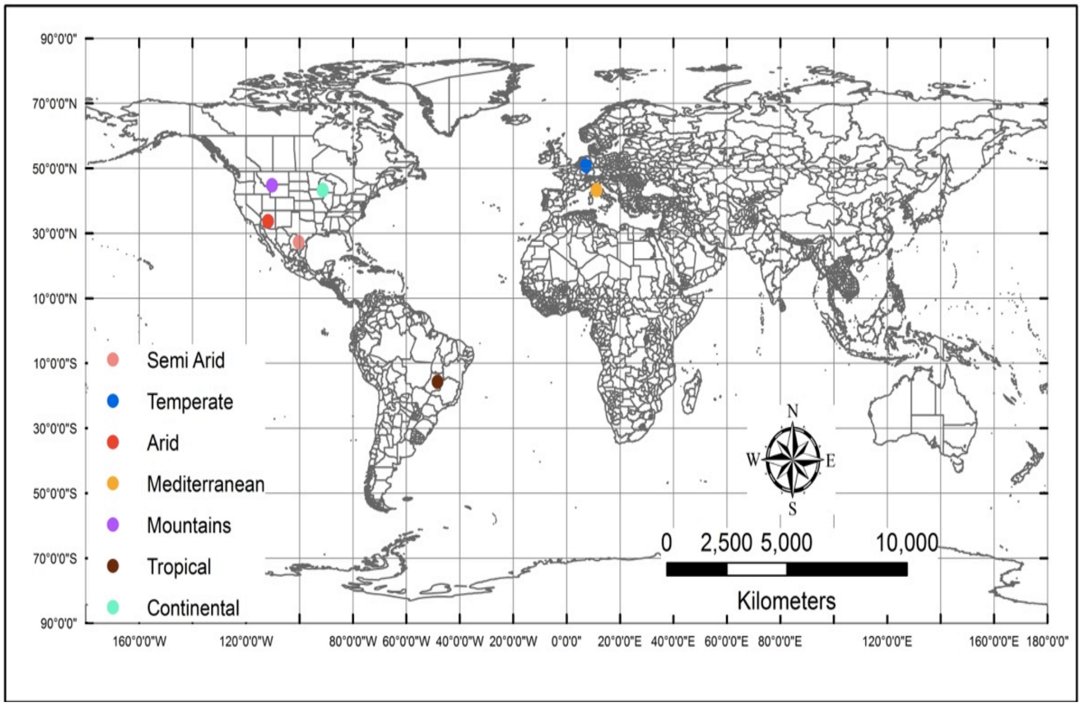


Fig. 3.3. Seven climatic zones used to process the daily records during the period from 1950 to 1999 for precipitation, minimum temperature, and maximum temperature

precipitation averaged over the 50-year period shows unreasonable output, where all days have at least some precipitation data in all climatic zones. Therefore, the precipitation data were taken from the selected points over one year only, and that year was 1990. After retrieving the one-year daily data averaged over 50 years, the mean monthly average and the monthly standard deviation were obtained for the three parameters (precipitation, minimum air temperature, and maximum air temperature).

3.2.3 Water Balance Model

This model is considered the main part of the technical module of the game. It simulates the field soil water and salinity balances on a daily basis and calculates crop growth, consumptive use, soil water content, soil salinity, and relative yield response to irrigation events. Thus, the model monitors the daily irrigation scheduling program and its effect on crop conditions and productivity (Fig. 3.4a, b).

Required Input Data

The data required for the model to calculate the daily soil water balance are divided into two categories:

1. Farm data: data that are applicable for the whole farm, such as latitude, climate zone, depth to the groundwater table, salinity of the irrigation water, and farm irrigation water delivery method; and,
2. Field data: data that are specific for each field within the farm, such as the planted crop, planting date, irrigation method, and soil texture.

Various parameters that affect the daily soil and salt water balance are considered, such as: depth of applied irrigation water, depth of precipitation, groundwater contribution, evapotranspiration, deep percolation, and surface runoff. Calculations of water balance are based on the following equation (Allen et al. 1998):

$$Dr_{EndofDay}(J) = Dr_{BeginningofDay}(J) - P_{net}(J) - I_{net}(J) - GW_{net}(J) + ET_a(J) + DP_a(J) \tag{3.1}$$

where *J* is the day of the year; $Dr_{EndofDay}(J)$ is the depth of water depletion in the root zone at the end of day *J*; $Dr_{BeginningofDay}(J)$ is the depth of water depletion in the root zone at the beginning of day *J*; $P_{net}(J)$ is the actual amount of precipitation that enters the root zone during day *J*; $I_{net}(J)$ is the amount of irrigation water that infiltrates into the soil during day *J*; $GW_{net}(J)$ is the amount of groundwater contribution in the root zone area during day *J*; $ET_a(J)$ is the actual depth of crop evapotranspiration during day *J*; and, $DP_a(J)$ is the actual depth of water deep-percolated below the root zone during day *J*. All terms in Eq. (3.1) have units of millimeters.

For the first day of simulation, the following are assumed: $Dr_{BeginningofDay}(J)$ = 0; $\theta_{BeginningofDay}(J)$ = soil water content at field capacity (θ_{FC}); ponded water at soil surface from irrigation or precipitation, *PW (J)* = 0; runoff water, *RO (J)* = 0; *relative yield (J)* = 100%. And, for the subsequent days until the end of the crop growing season the following are assumed: $Dr_{BeginningofDay}(J) = Dr_{EndofDay}(J-1)$; $\theta_{BeginningofDay}(J)$ = $\theta_{EndofDay}(J-1)$; *PW(J) = PW (J-1)*; and, *RO (J) = RO (J-1).*

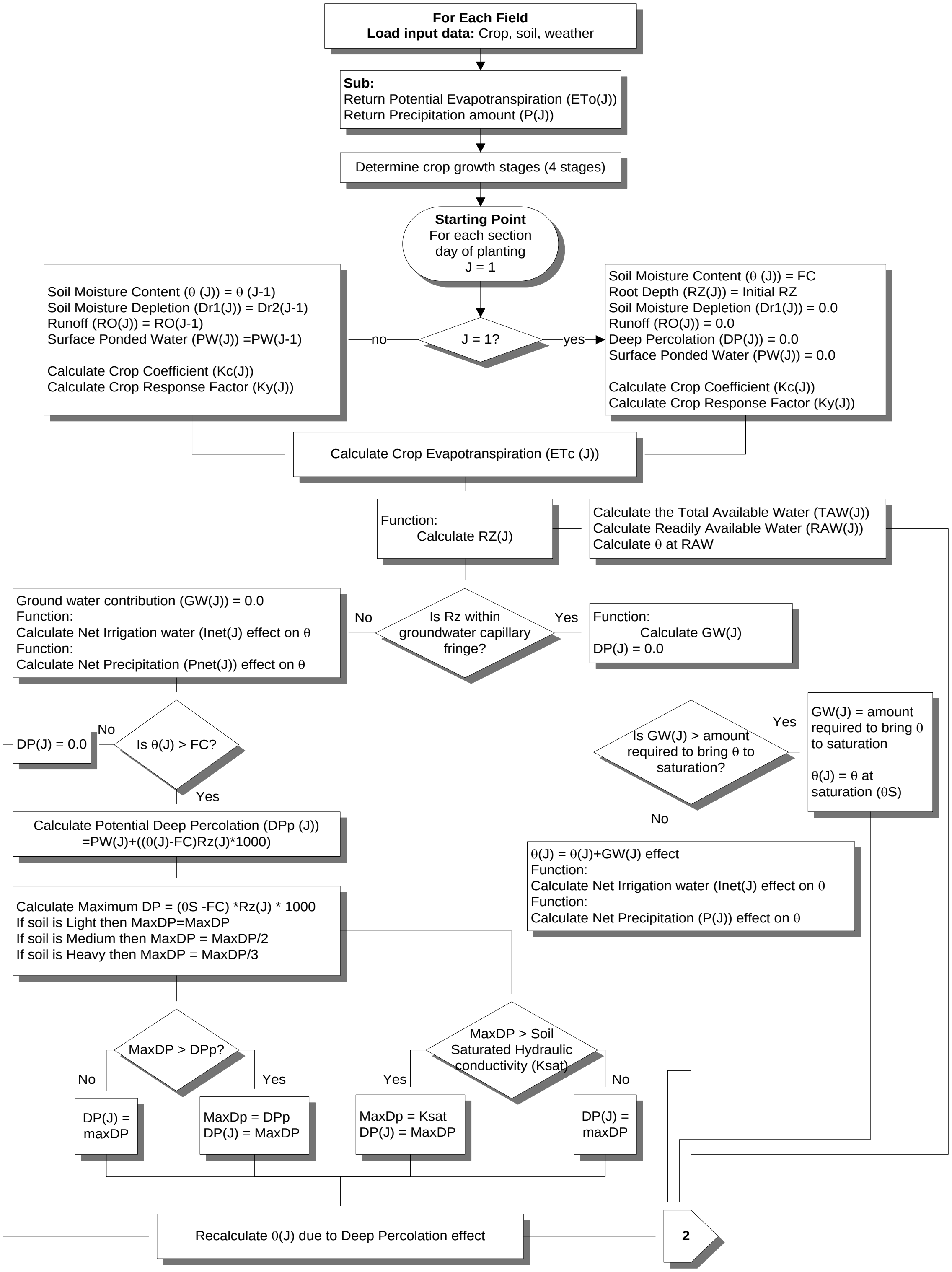


Fig. 3.4a. Flowchart of the soil water balance calculations in the model

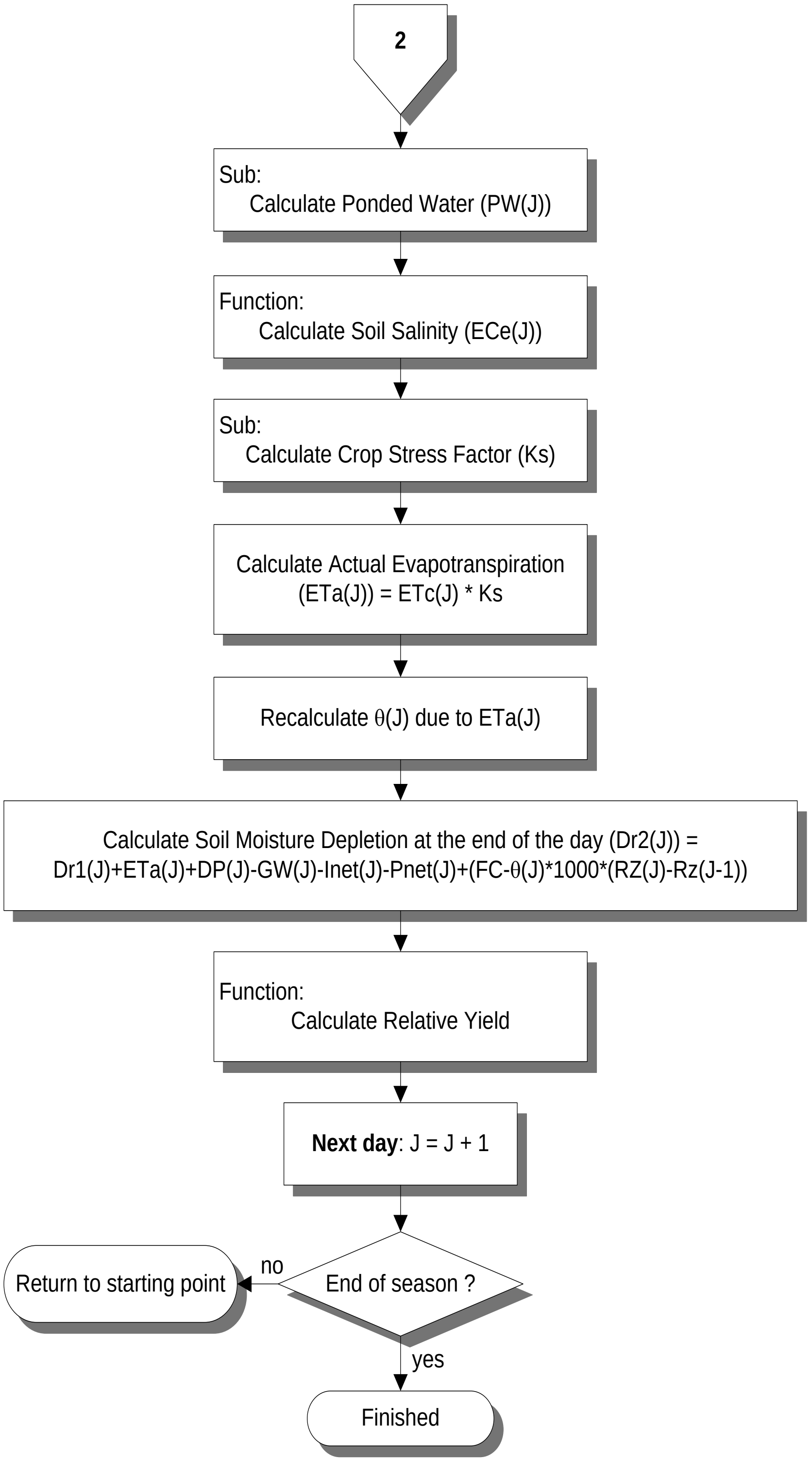


Fig. 3.4b. Flowchart of the soil water balance calculations in the model

Simplified assumptions were made to estimate all parameters on the right side of Eq. (3.1). These assumptions are as follows:

- Homogeneous soil profile (in both texture and structure) throughout the root zone and has only one soil layer. Therefore, soil water content and salt concentration is uniform throughout the depth of the root zone for each 24-h simulation interval.
- Soil water depletion at the beginning of the planting day is assumed to be zero, and the soil water content at this time is at field capacity.
- The depth to the water table is taken to be independent of internal variables such as deep percolation or capillary rise.
- Lateral flow of soil water between adjacent fields is considered to be negligible.
- If irrigation, precipitation, and groundwater contributions all enter the crop root zone in any given day of a simulation, it is assumed that the groundwater contribution occurs first, followed by irrigation, and finally by precipitation.
- At each day, if the net groundwater contribution to the root zone is more than zero, then net deep percolation from the root zone is equal zero.

The calculations of all parameters within the water balance model were as follows:

Root depth (R_z): If there is no vertical barrier within the root zone (e.g. water table), the daily root depth is calculated based on the assumption that the daily root growth rate is constant and increases linearly from the date of planting (Fig. 3.5). The following equation was used to calculate the daily root depth (Prajamwong 1994):

$$R_z(J) = R_z(J-1) + \frac{(R_z)_{\max} - R_z(J-1)}{J_{full\,cover} - J_{planting}} \tag{3.2}$$

where $R_z(J-1)$ is the root depth at the previous day; $(R_z)_{\max}$ is the maximum root depth of the specific crop, usually reached at the end of the development growth stage; and, $J_{planting}$ is the planting day.

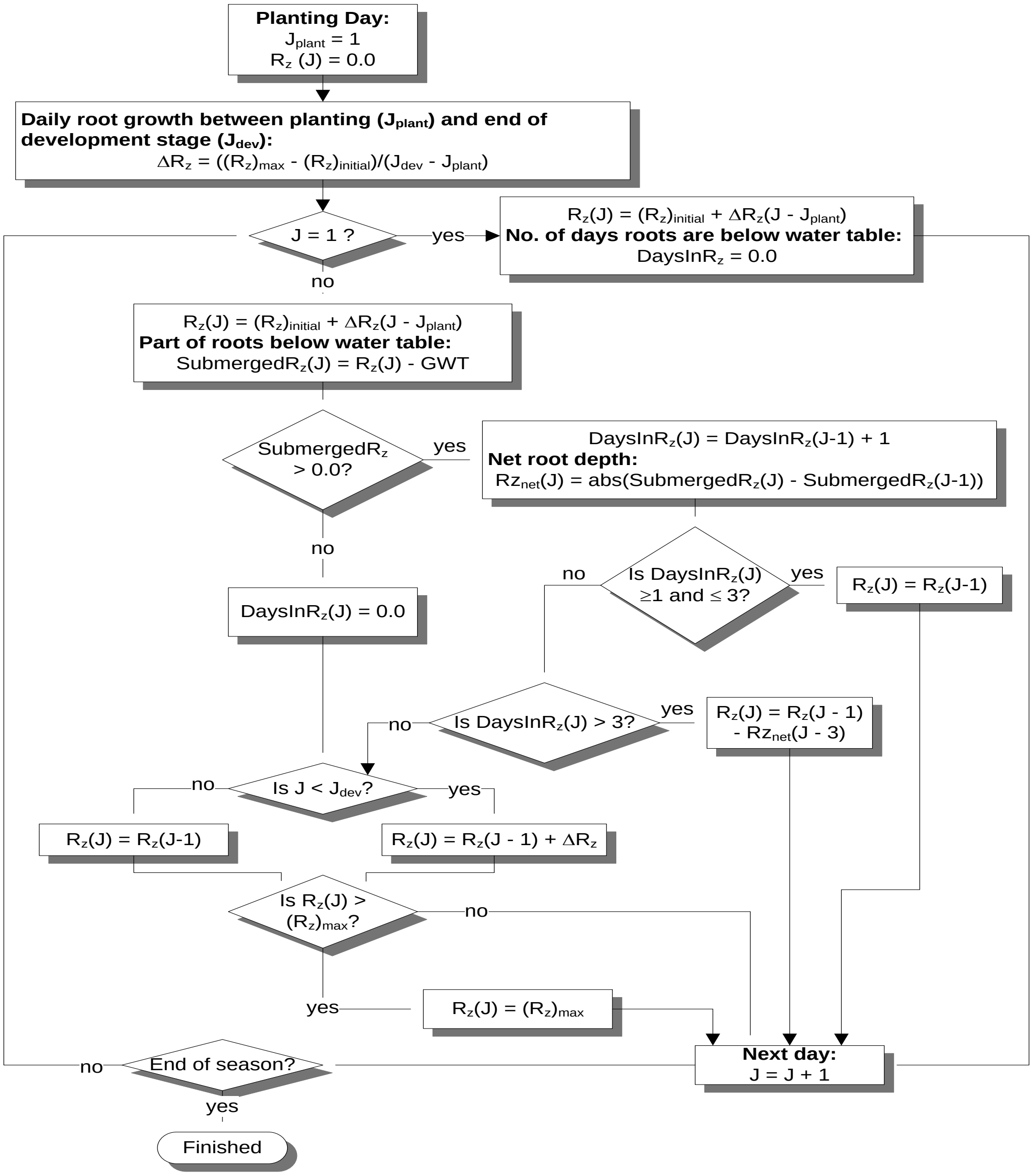


Fig. 3.5. Daily root depth calculation procedure

The model will not allow the root depth to exceed the maximum reported root depth for the specific crop (Allen et al. 1998). Also, in calculating the root depth, the sub-model considers the depth of the groundwater table. If the bottom of the root zone is at the water table, there will be no root growth during that day. Likewise, there will not be any root growth if the water table is inside the root zone. If any portion of the root zone stays within groundwater table for more than three days, that portion will die.

The model also considers whether the part of the root that atrophied due to saturated soil water conditions will grow back or not, based on the crop growth stage. If the crop has passed the development stage, there will be no additional root growth. Also, if groundwater table coincides with the ground surface for more than three days, the crop will die. The one exception considered herein is that of rice, which can survive fully saturated root-zone conditions.

Actual crop consumptive use (ET_a): The daily actual consumptive use is calculated based on the following equation:

$$ET_a = K_s K_c ET_o \tag{3.3}$$

where K_s is used to account for the effect of soil water stress due to water shortage and salinity level in the root zone K_e; ET_o is the grass reference evapotranspiration (mm/day), calculated using the Hargreaves equation; and, K_c is the crop coefficient, a function of growth stage (Allen et al. 1998).

The climatic data required to calculate reference evapotranspiration (ET_o) are included in the software. The climatic data parameters are: maximum mean daily temperature, T_{max} (^oC); and, minimum mean daily temperature, T_{min} (^oC). The player must choose from one of seven climate zones, as described previously in this chapter. Each time the game is played, a new set of climatic data will be used for this calculation.

The daily reference evapotranspiration, ET_o, (Fig. 3.6) is calculated according to the Hargreaves equation (Merkley 2007).

$$ET_o = 0.0023 R_a (T + 17.8)\sqrt{TR} \quad (3.\ 4)$$

where R_a is extraterrestrial radiation (MJ/m^2/day); T is the mean air temperature in ^oC; and, TR is the average daily temperature range in ^oC (mean daily maximum minus mean daily minimum air temperature).

The calculation of $ET_o(J)$ for J=1 is based on converting the planting month and planting day entered by the player into day of the year, as a number between 1 and 365. This is done using the following equation (Allen et al. 1998):

$$\begin{aligned} &Planting\ date = Truncate\,(275(M/9) - 30 + D),\ \ if\ M < 3,\ or \\ &Planting\ date = Truncate\,((275(M/9) - 30 + D) - 2),\ \ if\ M \geq 3 \end{aligned} \quad (3.\ 5)$$

where M is the planting month (1-12); and, D is the planting day (1-31) in the month. These parameters are entered by the player.

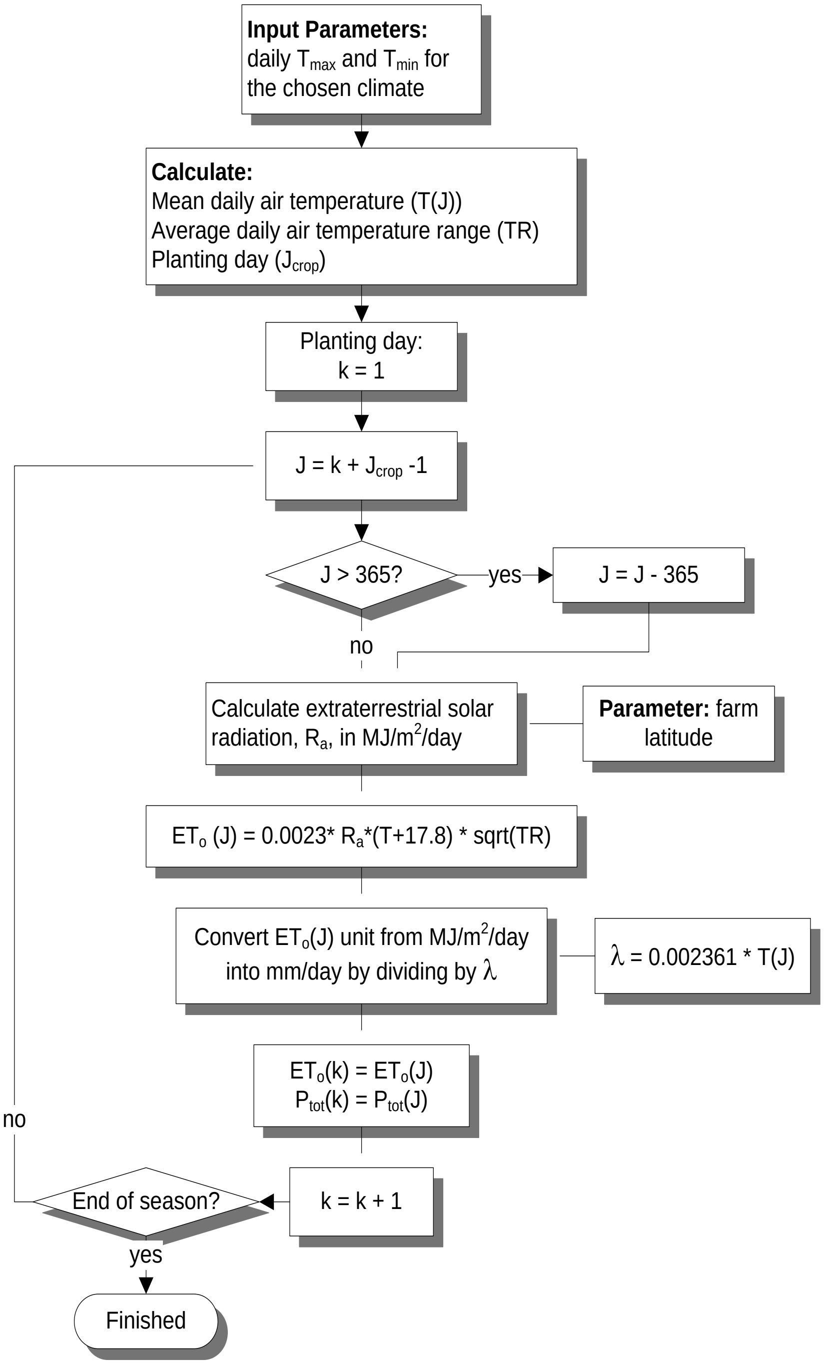


Fig. 3.6. Daily grass reference evapotranspiration calculation procedure

To estimate K_c on a daily basis, the following equations were used (Allen et al. 1998):

$$K_c(J) = K_{c_{prev}} + \left(K_{c_{next}} - K_{c_{prerv}}\right)\left(\frac{J_c - \sum L_{prev}}{L_{stage}}\right) \tag{3.6}$$

where J$_c$ is day number within the growing season; $K_{c_{prev}}$ is crop coefficient for the previous growth stage; $K_{c_{next}}$ is crop coefficient for the next growth stage; $\sum L_{prev}$ is sum of the length of all previous stages (days); and, L_{stage} is length of the stage under consideration (days).

The soil water and salinity stress factor, K_s (Fig. 3.7), is calculated using the following equation (Allen et al., 1998):

$$K_s(J) = \left[1 - \frac{b}{100K_y}(EC_e(J) - EC_{threshold})\right]\left[\left(\frac{TAW(J) - D_r(J)}{TAW(J) - RAW(J)}\right)\right] \tag{3.7}$$

where *TAW* is total available water in root zone (mm); *RAW* is readily-available water (mm); *b* is the reduction in crop yield per increase in EC_e *(%/dSm^{-1})*; $EC_{threshold}$ is the electrical conductivity of the saturation extract at the threshold when crop yield first reduces below the potential crop yield (*dS/m*); and, K_y is a yield response factor (Doorenbos and Kassam 1979).

The first part of the equation represents the stress due to soil water salinity, while the second part represents the stress due to water deficit.

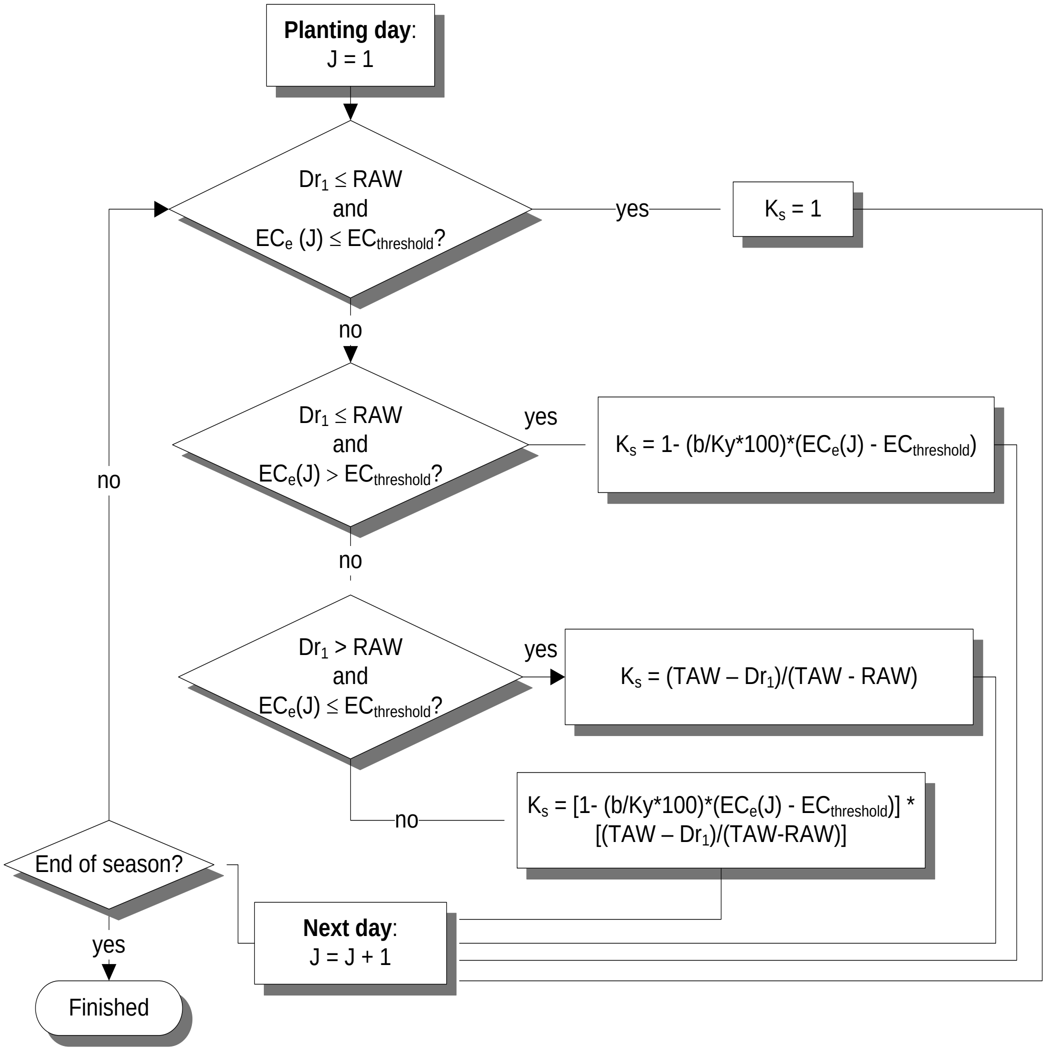


Fig. 3.7. Daily stress factor calculation procedure

Groundwater contribution (*GW*): Based on the depth of the groundwater table (*GWT*), if the water table is not inside the root zone, the groundwater contribution can affect the plant only if capillary rise (capillary fringe) from the groundwater table reaches the bottom of the root zone (Table 3.1). An average of the values is considered in the model for each textural classification.

Table 3.1. Capillary rise values for various soil types (Tanji and Kielen 2002)

Soil Texture	**Capillary Rise (cm)**
Coarse	20 to 50 cm
Medium	50 to 80 cm
Fine	90 cm

The groundwater contribution is the up-flux due to capillarity from the water table (*m/day*) and can be calculated based on Darcy's Law (Eching et al. 1994):

$$GW = -K(\theta)\frac{\partial h(\theta)}{\partial Z} = \frac{h(\theta)}{GWT} \tag{3.8}$$

where $K(\theta)$ is the unsaturated hydraulic conductivity (cm/day); GWT is the depth to the water table from the ground surface (cm); and, h is the soil water head (cm).

Unsaturated hydraulic conductivity is calculated from the following equation (Eching et al. 1994):

$$K(\theta) = K_{sat}\left[\frac{\theta(J)-\theta_r}{\theta_s-\theta r}\right]^{0.5}\left[1-(1-\left[\frac{\theta(J)-\theta_r}{\theta_s-\theta_r}\right]^{1/m})^m\right]^2 \tag{3.9}$$

where θ_r is residual soil water content (cm^3/cm^3); θ_s is saturated soil water content (cm^3/cm^3); K_{sat} is the saturated hydraulic conductivity (cm/day); and, m is an empirical parameter, defined as follows:

$$m = 1 - \frac{1}{n} \tag{3.10}$$

where n is also an empirical parameter (defined in Table 3.2;) and h is soil water head (cm), and is calculated as follows (Raes 2009):

$$h(\theta) = \left(\frac{1}{\alpha} \left[\frac{\theta_s - \theta_r}{\theta(J) - \theta_r} - 1 \right]^{1/m} \right)^{1/n} \tag{3. 11}$$

As a logical assumption, if there is any amount of groundwater contribution to the soil root zone, no deep percolation occurs on that day. The sub-model will also calculate the effect of this quantity on the soil water content of that day. If the contributed quantity is enough to saturate the soil root zone, then no irrigation or rainfall water can enter the soil profile on that day; if any of these happen, the model will add these quantities to the runoff, or to the ponded water if the basin irrigation method is used in the simulation. If the quantity is not enough to saturate the root zone, then if any amount of irrigation or precipitation occurred at that day, the sub-model will allow for an amount of irrigation and/or precipitation to enter the soil profile such that the maximum of this amount is enough to raise the root zone soil water content to saturation. Any amount above that level will be added on the runoff quantity, unless the irrigation method is basin irrigation method, in which case this amount will be added to the ponded water. Figure 3.8 describes the calculation procedure for groundwater contributions to the root zone.

Table 3.2. Class average values of Van Genuchten water retention parameters (Schaap et al. 1999 cited in Raes 2009).

Soil Type	***n***	***α*** **(*cm*$^{-1}$)**	**θ$_s$(*cm*^{3}/*cm*^{3})**	**θ$_r$(*cm*^{3}/*cm*^{3})**
Sand	3.18	0.035	0.375	0.053
Loam	1.48	0.0098	0.4	0.062
Clay	1.27	0.011	0.457	0.1

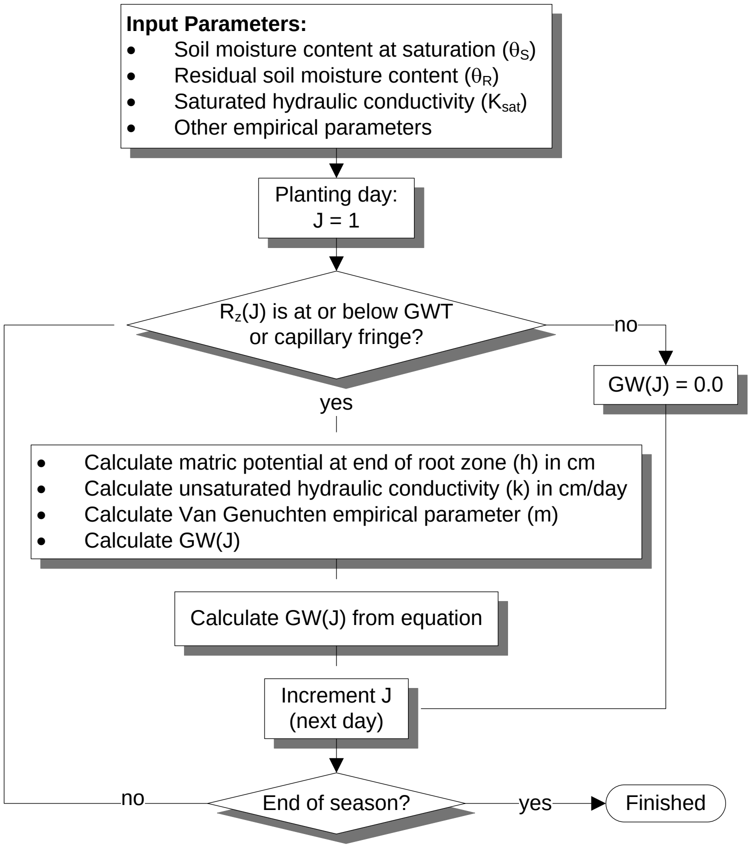


Fig. 3.8. Calculation procedure for groundwater contributions to the root zone

Amount of irrigation water (*Inet*): Based on the chosen on-farm irrigation method, the model calculates the net amount of irrigation water that enters the soil profile as a result of each irrigation event. For basin irrigation, the total amount of irrigation water has the potential to enter the soil profile, with no surface runoff losses. The sub-model checks if the amount of total irrigation water is enough to saturate the soil root zone. If it does, it means there will be some extra water, which will be stored on the soil surface as ponded water. By assuming that the ponded water will substitute for the water consumptive use of that day, the ponded water might take more than one day to infiltrate in the soil. The sub-model accounts for this and calculates the depth (which may be zero) of ponded water on a daily basis.

With furrow, border, sprinkler, and drip irrigation methods, no ponded water is allowed to remain on the soil surface. Also, not all of the irrigation water will infiltrate the soil even if the amount of water is less than the amount required to bring the water content to saturation. Some of the irrigation water will be lost from the field due to runoff. The amount of runoff is estimated as a fraction of the total irrigation water (p). The fraction was based on information from Walker (2010, personal communications) and is presented in Table (3.3).

After calculating the net amount of irrigation water that enters the soil profile, the model calculates the effect of that irrigation amount on soil water content (Fig. 3.9).

Table 3.3. Assumed fraction (p) of total irrigation water lost as runoff (Walker 2010, personal communications)

Soil Texture	**Irrigation Method**	**p**
Coarse	Furrow	0.1
Coarse	Border	0.1
Coarse	Drip	0.0
Coarse	Sprinkler	0.01
Medium	Furrow	0.2
Medium	Border	0.15
Medium	Drip	0.0
Medium	Sprinkler	0.02
Fine	Furrow	0.3
Fine	Border	0.2
Fine	Drip	0.0
Fine	Sprinkler	0.05

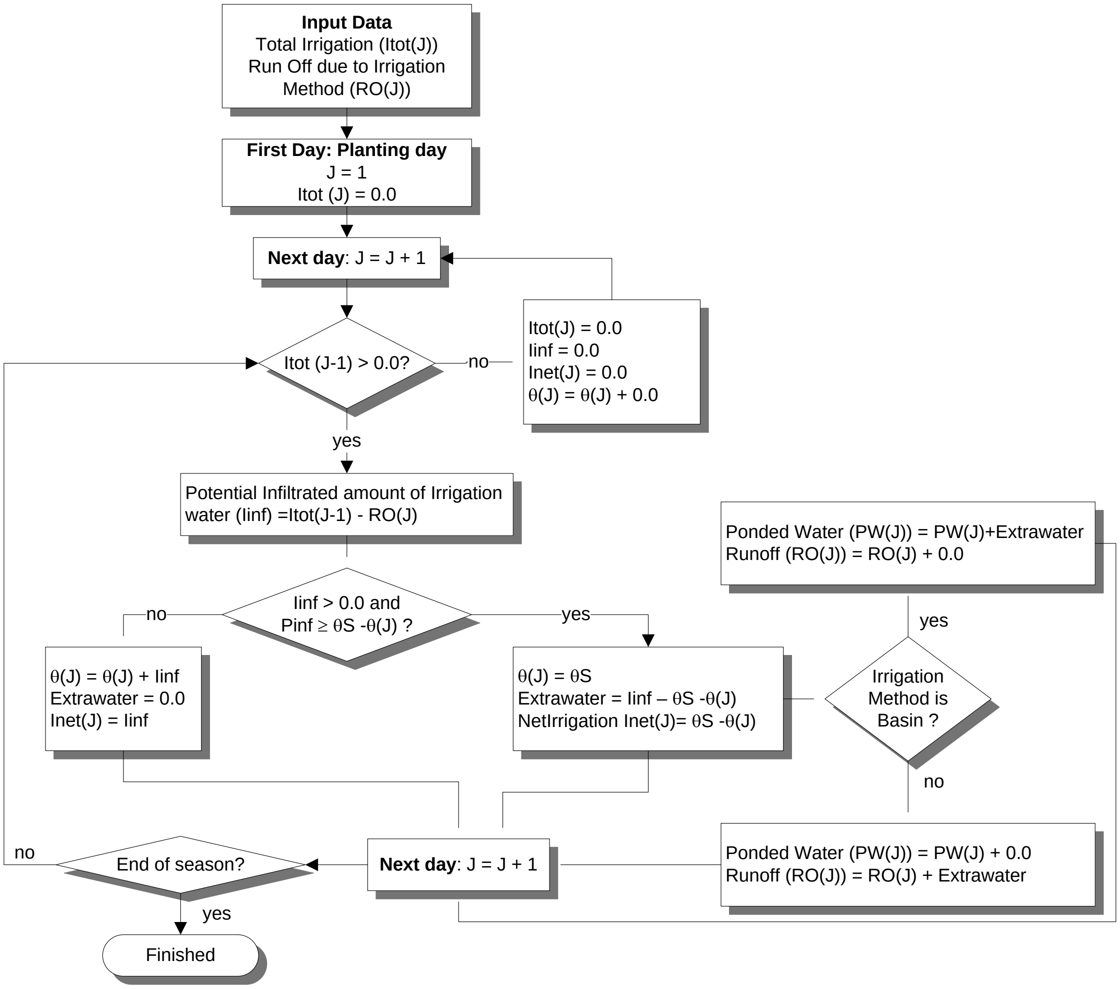


Fig. 3.9. Calculation procedure for the effect of net irrigation on soil water content

<u>Amount of precipitation water (*Pnet*):</u> The calculation of the amount of precipitation water follows the same reasoning as the calculation of the net irrigation that enters the soil profile, taking into consideration the irrigation method used. But, instead of taking the runoff quantity as a fraction (percentage) from the total precipitation, the sub-model calculates the effective precipitation by following the FAO-AGLW approach, after adapting it for daily calculations using the following equations (Smith 1988):

$$P_{inf} = 0.6P_{total} - \frac{10}{30}; P_{total} \le \frac{70}{30} mm \tag{3.12}$$

$$P_{inf} = 0.8P_{total} - \frac{25}{30}; P_{total} > \frac{70}{30} mm \tag{3.13}$$

where P_{inf} is the amount of precipitation that infiltrates the soil at the surface.

If there is any precipitation that goes to runoff or ponded water, these amounts will be added to the previously calculated runoff and ponded water quantities due to irrigation. The effect of the net amount of precipitation that enters the soil surface, on the soil moisture content is also calculated by the model (Fig. 3.10).

Deep Percolation (*DP*): If the soil water content in the root zone is more than the field capacity, and there is no groundwater contribution to the soil root zone, there will be some amount of water deep percolated at the bottom of the root zone, to be considered in the sub-model. The sub-model calculates the potential deep percolation (*DPp*), which is the amount of water that could potentially percolate below the root zone (and which includes the soil water content above field capacity and any ponded water on the soil surface).

Since only a specific amount of water can percolate below the root zone, according to the soil texture, not all the potential deep percolation amount can leave the root zone in one day. The sub-model defined the maximum amount of water that can be deep percolated in one day based on the saturated hydraulic conductivity of that soil texture class. For the normal range of agricultural soil textures, it will take one to four days for the extra water (above field capacity) to drain from the root zone due to gravity

(Hargreaves and Merkley 1998). The model considers three days for heavy soils (clays), two days for medium soils, and one day for light soil textures, such as sands.

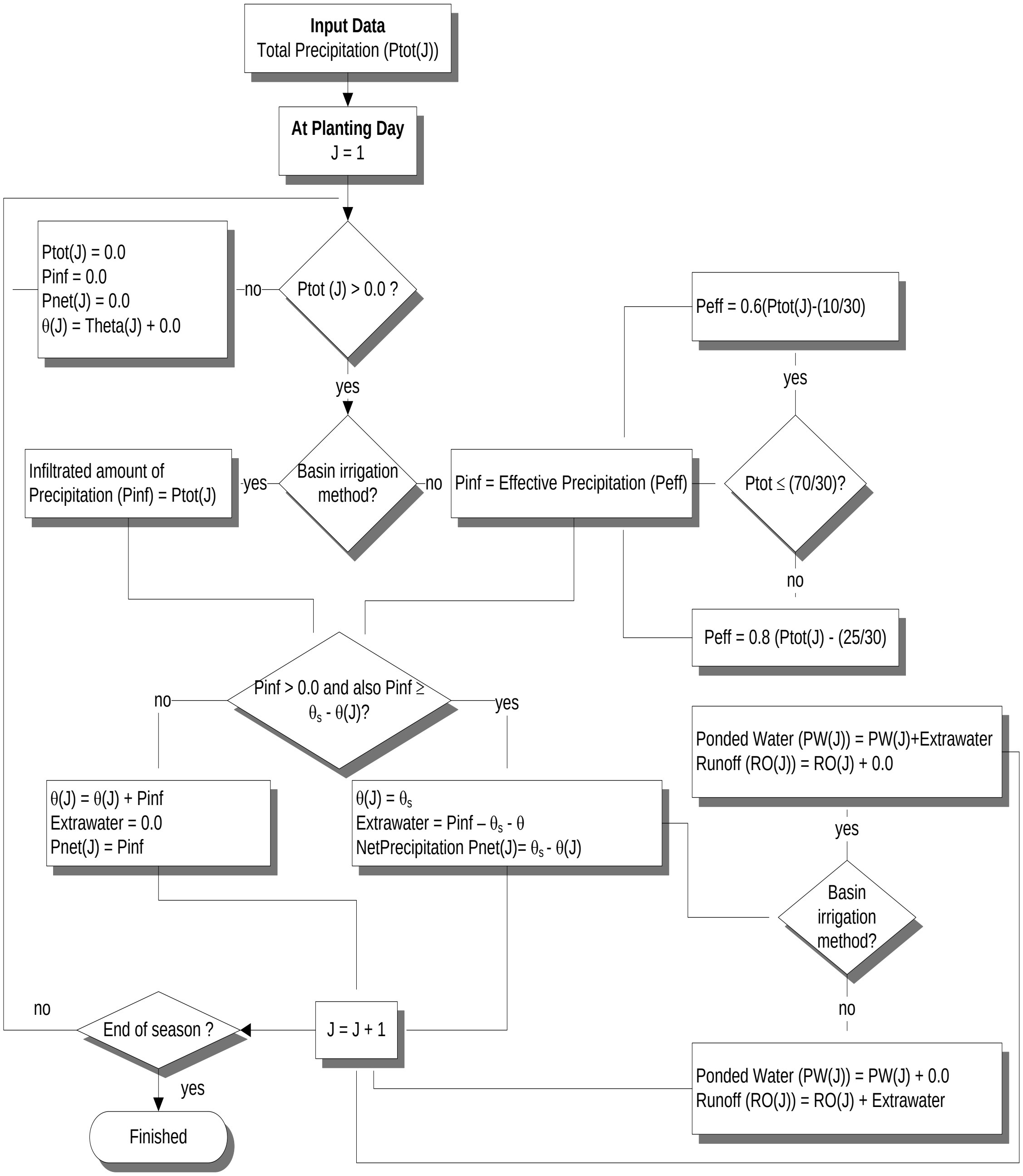


Fig. 3.10. Calculation procedure for the effect of net precipitation on soil water content

Due to actual deep percolation of soil water below the root zone, the soil moisture content will change and must be recalculated as follows:

$$\theta(J) = \theta(J) - \frac{DP_a(J)}{1000R_z(J)} \tag{3.14}$$

where R_z is in m.

<u>Soil Moisture Depletion at the end of the day</u> ($Dr_{EndofDay}(J)$): In the calculating of the water stress due to water shortage, the calculation of soil moisture depletion at each day is required. The calculations in the soil moisture depletion procedure are based on Eq. (3.1), which includes all the parameters affecting soil moisture content during the day. Using that equation without considering the growth of the root zone at the end of the day, and its related change in soil water content within the increased portion of the soil root zone, will result in erroneous calculations of the stress factor. Therefore, for the calculation of the soil moisture depletion at the end of the day, Eq. (3.1) should be amended to read as follows:

$$\begin{aligned} Dr_{EndofDay}(J) = Dr_{BeginningofDay}(J) - P_{net}(J) - I_{net}(J) - GW_{net}(J) + ET_a(J) + DP_a(J) \\ + (\theta_{fc} - \theta(J-1) * Rz(J) - Rz(J-1) * 1000) \end{aligned} \tag{3.15}$$

or the following simple equation can also be used in the determination of the depletion at end of day:

$$Dr_{EndofDay}(J) = (\theta_{fc} - \theta(J)_{\,End\,of\,Day}) * Rz(J) * 1000 \tag{3.16}$$

The use of this equation is subject to the way the soil water content $\theta(J)_{End\ of\ Day}$ is calculated, and whether θ is recalculated after the calculation of each parameter affecting soil moisture content or not.

3.2.4 Salt Balance Calculations

The root-zone salt balance is calculated on a daily basis in order to determine the daily *ECe* in the root-zone (Fig. 3.11). The sub-model calculating root-zone salt balance is based on the following concept:

$$S_{today} = S_{yesterday} + \Delta S \tag{3.17}$$

where S is mass of salt per unit area (mg/m^2); and, ΔS is the change in salt mass in the root zone.

The sub-model will start calculating the mass of salt per unit area in the day of planting with an initial value of EC_e and an initial value of root depth Rz_{initial}. Then, the salt mass in the root zone for the following days in the season is calculated based on the following equation:

$$S_j = EC_{e_{j-1}} * 6.4(10)^5 * R_{z_j} * \theta_s + \Delta S \tag{3.18}$$

where j is the day of the year; j-1 is the previous day; EC_e is the soil water extract salinity in dS/m; R_z is the root depth in m; θ_s is the volumetric water content at saturation (fraction); and ΔS is the change in mass of salt in the root zone per unit area (mg/m^2) over the given time interval.

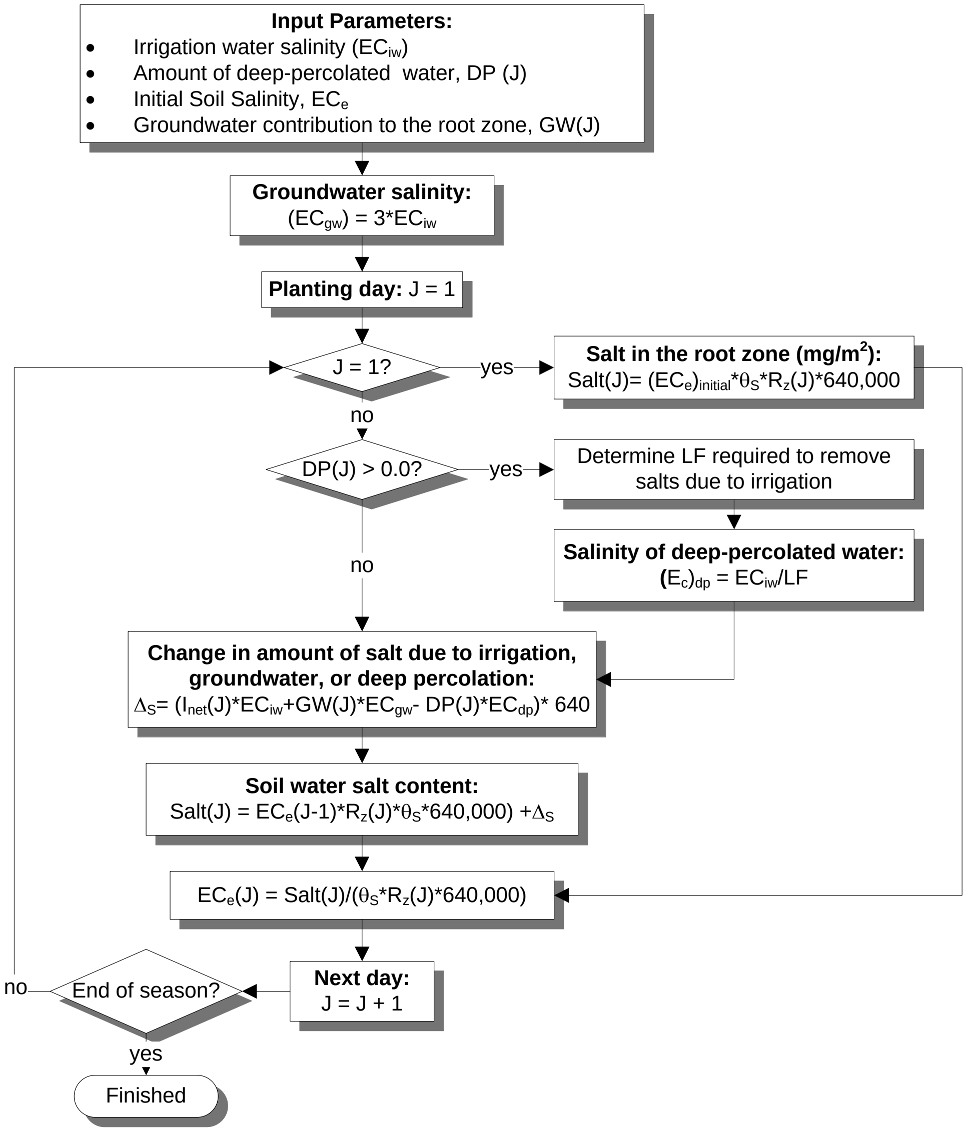


Fig. 3.11. Daily salt balance calculation procedure

Salt can enter the root zone in irrigation water and in groundwater, and it can leave only as deep percolation. Therefore, ΔS can be calculated as follows:

$$\Delta S = S_{iw} + S_{gw} - S_{dp} \tag{3.19}$$

where S_{iw} is the mass of salt entering the root zone from irrigation water (mg/m^2); S_{gw} is the mass of salt entering the root zone from ground water (mg/m^2); and S_{dp} is the mass of salt leaving the root zone by deep percolation (mg/m^2). Note that $S_{iw} \geq 0$, $S_{gw} \geq 0$, $S_{dp} \geq 0$.

The algorithm never allows both net groundwater contribution and net deep percolation water on the same day.

The mass of salt from irrigation water is calculated as:

$$S_{iw} = EC_{iw} I_i \left(640 \frac{mg/l}{dS/m} \right) \tag{3.20}$$

where EC_{iw} is irrigation water salinity (dS/m); and I_i is the depth of irrigation water that entered the soil root zone (mm). Similarly,

$$S_{gw} = EC_{gw} I_g \left(640 \frac{mg/l}{dS/m} \right) \tag{3.21}$$

and

$$S_{dp} = EC_{dp} I_{dp} \left(640 \frac{mg/l}{dS/m} \right) \tag{3.22}$$

where I_g is the contribution from groundwater (mm); I_{dp} is the amount of deep percolation (mm); EC_{gw} is the salinity of groundwater (ds/m); and EC_{dp} is the salinity of deep-percolated water (dS/m).

The EC_{gw} is calculated from the salinity of the irrigation water as follows (Ayers and Westcot 1985):

$$EC_{gw} = 3 * EC_{iw} \tag{3.23}$$

The deep percolation water salinity (EC_{dp}) can be calculated based on the fraction of irrigation water that is required to prevent excessive accumulation of salts due to irrigation (*LF*), as follows:

$$LR = \frac{EC_{iw}}{5\left(EC_e\right) - EC_{iw}} \tag{3.24}$$

$$EC_{dp} = \frac{EC_{iw}}{LR} \tag{3.25}$$

Then, the EC_e of the soil water extract can be calculated based on the following equation:

$$EC_{e(j)} = \frac{S_j}{6.4(10)^5\,(R_z)_j\,\theta_s} \tag{3.26}$$

3.2.5 Relative Yield Calculations

Crop yield is calculated in terms of the relative value with respect to potential crop yield. The relative crop yield is estimated by considering possible yield reduction due to the relative evapotranspiration deficit, which governed by root-zone water deficit and salinity stress, and the over-irrigation events during the growing season. Relative yield calculations are based on crop growth stage. At the end of each growth stage, the effect of soil water deficit and salinity stress on the relative yield is calculated using the following equation (Stewart et al. 1977, cited in Merkley 2007):

$$Y_{r,S} = \frac{Y_{a,S}}{Y_{m,S}} = 1 - K_{y,S}\left(1 - \frac{\sum_{Sj=1}^{Sj=n} ET_a}{\sum_{Sj=1}^{Sj=n} ET_c}\right) \tag{3.27}$$

where $Y_{r,S}$ is the relative yield at the end of the growth stage; $Y_{a,S}$ is the actual expected yield at the end of the growth stage; $Y_{m,S}$ is the maximum potential expected yield at the end of the growth stage; $K_{y,S}$ is a yield response factor for the growth stage; and $\sum_{Sj=1}^{Sj=n} ET_c$ is the cumulative summation of daily maximum evapotranspiration under ideal growing conditions (mm) from Sj=1, which is day1 in the growth stage, to Sj= n, which is the last day of the growth stage. ET_c is equal to $K_c * ET_o$; and $\sum_{Sj=1}^{Sj=n} ET_a$ is the cumulative summation of daily actual evapotranspiration (mm) throughout the growth stage.

The yield response factor (K_y) relates relative yield reduction to relative evapotranspiration deficit. The K_y values for each growth stage, for the different crop types in the model, were obtained from FAO Irrigation and Drainage Paper 33 (Doorenbos and Kassam 1979). On day one of the soil water balance simulation, the relative yield starts at 100%. This percentage remains until the end of the first growth stage, when the relative yield is recalculated based on the effect of root-zone water deficit and salinity stress, if this occurs during that stage, using Eq. (3.27). In order not to carry the effect of moisture and salinity stress to the following stages, the same equation is used for the calculation of the relative yield for the subsequent growth stages until the end of the growing season.

The end of season relative yield is calculated by taking the minimum value obtained from all growth stages as follows (Prajamwong 1994):

$$Y_{r,Season} = \min(Y_{r,S1}, Y_{r,S2}, ..., Y_{r,Sk}) \tag{3.28}$$

where $Y_{r,Season}$ is the end of season relative yield (%); and *k* is number of growth stages.

Over-irrigation penalty: Over-irrigation causes aeration problems, loss of water and soil nutrients below the root zone, and generally creates unfavorable conditions for plant growth, which leads to a reduction from the potential expected yield. To measure the impact of over-irrigation on crop yield, and use it as a penalty for any player who cares about maximizing yield without considering the amount of irrigation water used, especially when water is available and abundant. The following equation is used to calculate the impact of over-irrigation and the subsequent deep percolation on seasonal relative yield reduction (Prajamwong 1994):

$$Y_{r,Season} = 1 - \alpha(\frac{dp_{season}}{d_n}) \tag{3.29}$$

where dp_{season} is the total seasonal deep percolation (mm); α is an empirical coefficient (a = 0.05, as assumed by Prajamwong 1994); and *dn* is the maximum readily available depth of water in the root zone (mm), calculated as follows:

$$d_n = p * AM * (R_z)_{\max} \tag{3.30}$$

where p is average fraction of total available soil water that can be depleted from the root zone before moisture stress (fraction); *AM* is water at maximum root depth (mm) equal to *FC* -*WP*; and $(R_z)_{\max}$ is the maximum root depth of the specific crop, usually reached at the end of a crop growth stage.

The following flowchart shows the calculation procedure of the overall relative yield and each growth stage relative yield obtained (Fig. 3.12).

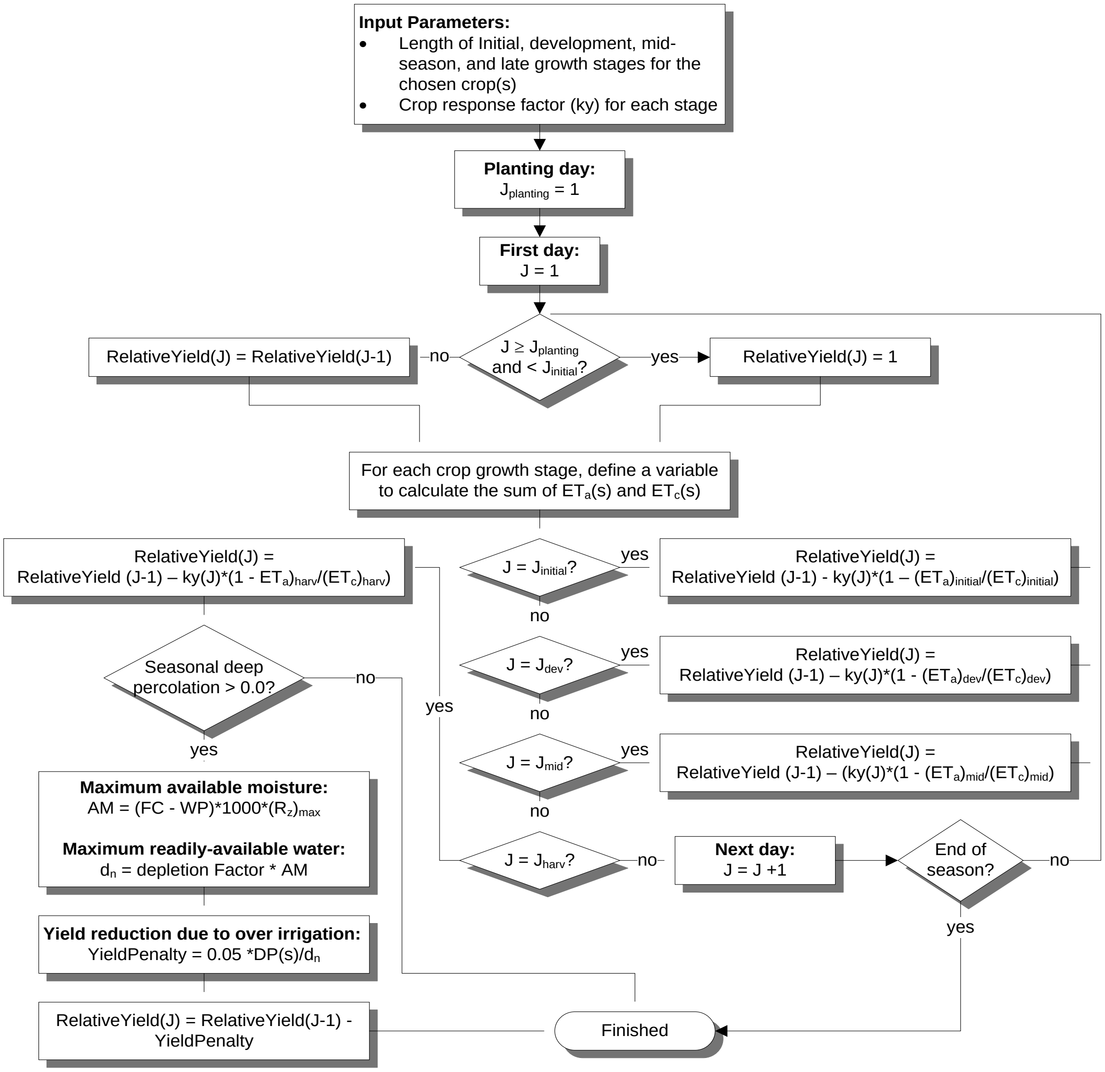


Fig. 3.12. Calculation procedure for relative yield at the end of the simulation period

3.3 Scoring Module

Processing a comparison between the results obtained from the player behavior module (management decisions taken by the player) with that considered to be the optimal management practice (the reference results) enables the scoring and recommendations module to evaluate the decisions made by the player and gives him or her a fair and reasonable score.

In terms of results scoring, the player will have a certain set of goals or objectives to meet: maximize crop yield, maximize production function, or maximize profit. The player must choose one goal before beginning a simulation. If the player does not choose a goal, the program will automatically choose to maximize profit. The scoring results will be based on the achievement of these objectives. A final score based on the overall consequences of the decisions which were made, in addition to recommendations for irrigation management improvements, is presented to the player at the end of the simulated growing season.

3.3.1 Scoring Procedure

This part of the code describes the procedure for developing the module responsible for making scoring results for the player based on his achievements throughout the simulation season. The calculation of the player performance results (score) should be referenced to the best results that can be obtained through running the simulation under the same circumstances (crop phenological properties, soil characteristics, climatic conditions, irrigation method characteristics, and water delivery method). To do so, and in order to give the player a fair and reasonable score, the program was designed to search for the optimum result irrigation management scenarios at the beginning of the growing season, and try to obtain the best results. After that the program runs a comparison between the results obtained by the player and the results obtain by its search algorithm.

Choosing the best search algorithm that gives reliable results depends on the objective function to achieve and how easy this search algorithm is to use, and how much time it takes in searching for the optimum solution. As they appear from the set of objectives, they are all optimization objective functions. A one-dimensional (one-objective) search algorithm, namely Golden Section, was used. This search algorithm was found to be appropriate as it has a low computational cost (low time consumed in performing the search). This is described in additional detail in Chapter 4.

Finding a general relation that relates the crop yield to the seasonal amount and duration of irrigation water used, for all the crops under the different climatological and environmental conditions was not found possible after reviewing many related technical articles. Therefore, the same daily soil water and salinity balance, which was described in the technical module part of this project, was used in this part of the code for the calculation of the crop production function with little modifications to consider the same management for all the sections within the field. The search algorithm goes through the calculation of all the parameters that affect the soil water balance and consequently affect the crop yield, each time searching for the only decision variable it has, which is the net irrigation amount.

The objective function was set to minimize the yield loss due to using certain percentages of net irrigation (Eq. 3.31). That will in turn calculate maximum yield obtain due to certain management practices. Because the quantity of applied water each irrigation varies due to soil water content at the time of irrigation, and due to weather and crop characteristics changes between each two irrigations, different decision variables could be considered. But, in order to make it easy to search for the optimal solution, it is found more feasible to limit the decision variable to one variable, which has a strong effect on the computational cost of the search process. The decision variable chosen to be the percentage of net irrigation water to be apply each irrigation (same percent each time) in order to bring the soil water content in the root zone to field capacity Eq. (3.32)

The general objective function used is:

$$\textbf{minimize: YieldReduction} = \textbf{1 - RelativeYieldObtained} \tag{3.31}$$

where *RelativeYieldObtained* is the relative yield (0 - 1) obtained at the end of the season due to the different irrigation scheduling scenarios (net irrigation percentage). The decision variable that used is as follows:

$$Inet(J) = 0.01 * Ipercentage * (FC - \theta(J-1) * R_z(J-1) * 1000 \tag{3.32}$$

where *Inet(J)* is net amount of water at the scheduled irrigation, which will enter the calculations for the soil water balance and crop relative yield; *Ipercent* is the percentage of total *Inet* required to bring the soil water content in the root zone to field capacity. *FC* is the soil moisture content at field capacity; $\theta(J-1)$ is the soil water content at the day of irrigation; and *Rz (J-1)* is the root depth on the day of irrigation.

As a dynamic programming objective function there is one constraint related to the amount of irrigation water that is available for the whole season. The total amount of water used for all irrigations must not exceed the total available water. Although the decision variable was the same for all the objective function, the limitations for calculating each goal are different (these minor constraints are to make the obtained results more accurately reflect the reality of the goal). The following sections describe the decision variable and constraints for each goal under different management scenarios of the water delivery methods included in the model.

3.3.2 Maximizing Crop Yield

If maximizing the crop yield is the objective function of the player, he or she will typically try to use all the available water without considering the optimum production function. In this case, the player usually opts for this objective when irrigation water is plentiful and inexpensive. But in conditions where water is limited during the whole season, different management decisions should be taken with regard to irrigation scheduling.

Under this objective function (maximizing crop yield), the search algorithm will consider the different delivery methods available in the game options; fixed rotation, or on-demand. The differences among these delivery methods is the delivery interval, where under fixed rotation the delivery interval is pre-determined, and knowing the seasonal amount of available water, the program can determine the maximum amount of water available for each irrigation. This maximum amount should not be exceeded. The difference under on-demand is that the player has no restrictions in terms of when to order water and how much water to order for each irrigation, although the total amount of available water for the whole season is fixed and can't be exceeded. Therefore, when to irrigate and how much water is available for each irrigation are not fixed for the program to search through.

In this part of the model, there are few limitations on irrigation scheduling decisions under the maximization of yield objective function; minimum irrigation interval based on the irrigation method chosen, and minimum irrigation duration . One day minimum interval set for drip irrigation, while seven days minimum interval set for border and basin, two days for sprinklers, and three days for furrow irrigation methods. So, the program considers the extreme possibility that the famer might choose to irrigate more frequently with the least amount of water at each irrigation, as this could be the optimal solution to obtain the highest crop yield. But, this minimum amount is subject to the fixed limitation (in the program) that the minimum irrigation duration is one hour. Irrigating more frequently with the least amount of water at each irrigation might not be feasible from a management point of view, but as a scoring procedure it might lead to the achievement of 100% yield, so the player might use this strategy.

3.3.3 Maximizing the Production Function

In this part, the goal in playing the game is to achieve the highest possible yield with lowest amount of irrigation water. That is, maximization of production per unit volume of water. The search algorithm in the program will study all possible percentages of net irrigation water that can give the highest production function using the same minimization function, to minimize the yield loss, and the same decision variable, same percent of net irrigation water to be apply each irrigation, as described above. Although the same decision variable is used for this goal, a set of limitations and constrains were included in the program in order to minimize the possibility of irrigating daily with a very low amount of water. Therefore, the limitation on when to irrigate is linked with the amount of soil moisture deficit. Under the fixed rotation delivery method, the limit of soil water deficit was set to the consumption of half of the readily available water (RAW) before allowing an irrigation event to take place. In contrast, under the on-demand delivery method the soil water content limitation is set to the consumption of all readily available water before allowing an irrigation event to occur, as this will lead to the optimal production function by minimizing the total irrigation water used without reducing the potential yield.

Another constraint exists for the last irrigation for all delivery methods, in which the last irrigation is limited so as to not exceed the amount of water required to raise soil moisture content to the mid-point between soil moisture content at FC and soil moisture content at RAW, because that will reduce the waste of additional water at the end of the season. The reason for this is that it is not desirable to leave the soil with high water content during harvest; this is will minimize the water that can be consumed, while having the same objective function of maximizing crop yield. Accordingly, the best production function can be achieved. The equation is:

$$\mathrm{BestProductionFunction} = \frac{0.01 * \mathrm{RelativeYield} * \mathrm{MaximumYield} * 1000 * 10}{\text{Total Water Used}} \qquad (3.33)$$

where *RelativeYield* is the best yield that was obtained by the automatic search for maximum yield (1-100); *MaximumYield* is the predefined maximum yield that could be obtained for the specified crop under the optimal management conditions (kg); and *Total Water Used* is the total amount of irrigation water used throughout the growing season (m^3).

3.3.4 Maximize Profit

The same procedure used in maximizing the production function is used for maximizing the profit. Taking into consideration that the profit is equal to revenue minus cost, revenue is related to the crop yield production, so increasing the yield will increase the revenue. Cost is related to the amount of water used throughout the growing season and does not include other possible production costs, such as agricultural chemicals, labor, cultivation practices, and others. This is because the focus of this game is irrigation water management, not overall farm management. The profit equation is:

$$\text{Profit} = \text{Revenue} - \text{Cost} \tag{3.34}$$

3.4 The Scenario-Based Module

In order to simulate the impact of both agents (system and player) on the game environment (the farm, including crops, soil, water, and weather conditions), the heuristic features of the game were developed based on a combination of two artificial intelligence technologies: (1) a pattern recognition approach; and, (2) reinforcement learning based on Markov Decision Processes (MDP). The pattern recognition approach enable the system to model the player type based on previously defined types, and the reinforcement approach teaches the system to determine the optimal policy (actions) to be taken at the specified states of the environment taking into consideration the player type (Fig. 3.13).

The task of the pattern recognition system is to develop a player model by extracting player information from the game world (player management decisions and actions), group the information into classes of similar patterns, and forward this information to the decision-making system (reinforcement learning). Based on the forwarded information, the decision-making system has the responsibility to decide the appropriate action by the system (agent) from the set of possible actions allowed by the game environment.

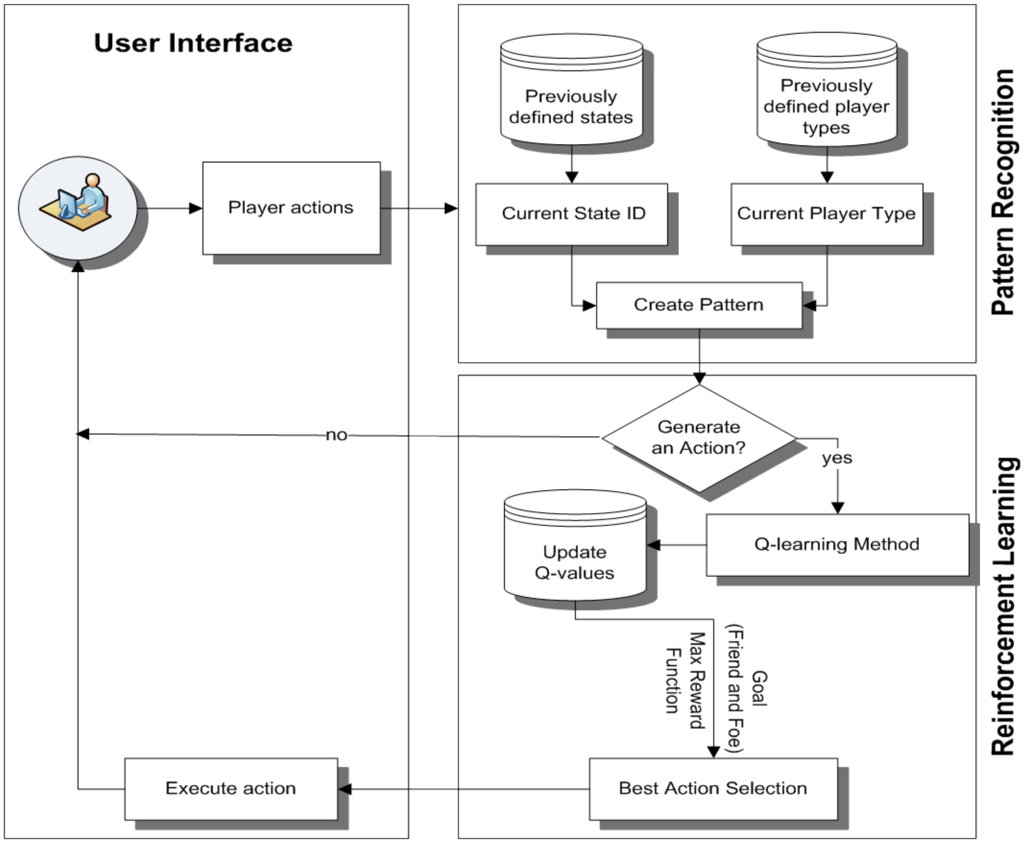


Fig. 3.13. Schematic of the heuristic model

The intention from any actions taken by the player is to successfully achieve the main goal of the game, which is generally, to maximize profit. After taking an action, the closer the player is to the goal describes how rational the player is. The player model defines the player into one of four different categories base on his or her decisions (actions) in the game environment (see Chapter 4).

In the pattern recognition system, the action taken by the player will be observed, traced with the sequence of the previous actions that were taken, and classified and learned as one case in the pattern generation process. Based on the obtained cases, the player modeling process can match human behavior and generate a player model by matching the learned case with a previously defined player models. This information will be forwarded to the decision-making system (reinforcement learning system), which has the responsibility to choose an appropriate action based on the given information and the state of the environment.

The reinforcement approach used in the decision-making system is based on the idea of learning through interaction with the environment. Learning what to do is accomplished through learning the best reward obtained from taking a particular action in a particular game-world state.

The Q-Learning system learns an action-value function, or Q-function, giving the expected utility of taking a given action in a given state. It deals with the theory of expected utility maximization. The learning system divides the environment into states based on certain criteria given by the designer, and it examines the effect of all available action on each state. Based on the reward criteria set in the game, the system returns the reward of taking each action at each state. The reward is calculated based on how close the action taken by the system (agent) toward achieving the previously defined goal. The returned reward is used to calculate the Q-value function of a state-action pair Q (a, s). Choosing the maximum Q-value state-action pair is analogous to deciding the best action that should be taken in the current world state returns the expected best action to be taken in this state. The system observes the environment after taking that action and evaluates its decision based on the new reward obtained, and learns from its decision. The algorithm used to determine the optimal policy is called the Q-learning algorithm and is based on the following equation:

$$Q_{new}(a,s) = Q_{current}(a,s) + \left(\alpha\ (r(s) + \gamma \max_{a} Q(a',s') - Q_{current}(a,s)\right) \qquad (3.35)$$

where, Q(a, s) is the value function of state action pair; α is the learning rate and reflect the difference in utility between two successive states; r is the expected reward value, reflecting the future effect of taking action a in world state s; γ is a discount factor describes the performance of an agent for current reward over future reward; (s) is the current state; and (s') is the new state the game world expected to move to after taking action “a” in world state (s).

CHAPTER 4

HEURISTIC SIMULATION

In efforts to achieve higher levels of enjoyment and realism in playing the game, different heuristic approaches were employed. The following describes implementation of the heuristic capabilities within the game for more realistic simulation of all the important aspects of on-farm water management and irrigation scheduling.

4.1 Weather Data Generation

4.1.1 Maximum and Minimum Temperatures

A model was developed to generate new sets of daily maximum and minimum temperature each time the game instantiated (Fig. 4.1). The model generates a random number using the monthly standard deviation for both maximum and minimum temperature obtained as shown previously for each climate zone. The procedure to generate new set of daily data is as follows: the year is divided into a total of 13 peaks, where the 12 in-between spaces represent the 12 months of the year. The first peak is on day 1 of the year, and the last peak is on day 365. The number of days for each peak is the same as the number of days in each month. To determine the minimum and maximum range for randomization of each month, the monthly standard deviation of the particular parameter is used. The range falls between plus and minus one standard deviation from the daily value. Accordingly, a random number is generated within that monthly numbers space will be taken as a coefficient to be added to (or subtracted from) the actual daily data.

To generate that coefficient for all the days in the month, and in order to obtain a smooth randomization of the coefficients, a 3rd-degree polynomial curve is defined between each two adjacent peaks (months). The model generates a random number coefficient) at each peak. The randomly-generated coefficient for each of the two adjacent peaks, considered the start and the end y-coordinate of the polynomial curve, while the starting day at each peak considerd the x-coordinate of that curve. To generate all the points on the curve, the Gauss-Jordan elimination method is used to solve a linear system of four equations based on the two known points and their specified end slopes to create a 4 x 4 matrix. Accordingly, 365 randomized coefficients are generated each time the module is invoked.

By adding the randomly-generated coefficient to the actual daily data, new set of daily maximum and minimum temperatures are generated each time the player starts a new game. The program makes sure that maximum temperature is always larger than or equal to the minimum temperature on each day. This daily data set is the one that is used in the technical model to calculate the potential ET and the daily soil water balance. Figure 4.1 shows the coding process for this module.

Figure 4.2 shows sample results of generated daily maximum and minimum temperature as compared to the real (measured) data that were obtained for each parameter as discussed in the previous chapter. Figure 4.3 shows a comparison between the average daily data generated and real data to show the effectiveness of the method in generating acceptable random air temperature data based on a fixed set of daily measured values.

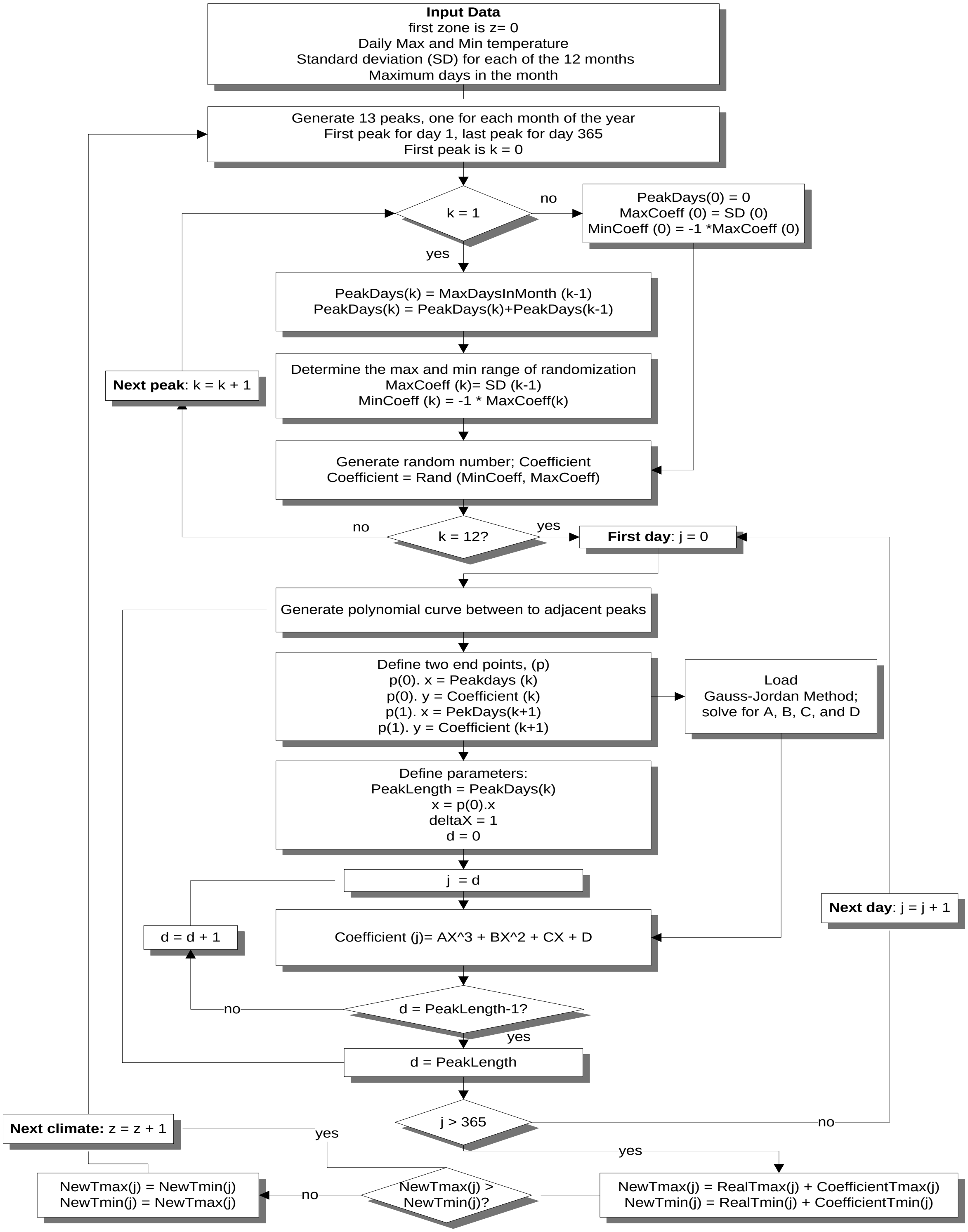


Fig. 4.1. Flow diagram of the process to randomly generate modified sets of maximum and minimum temperature

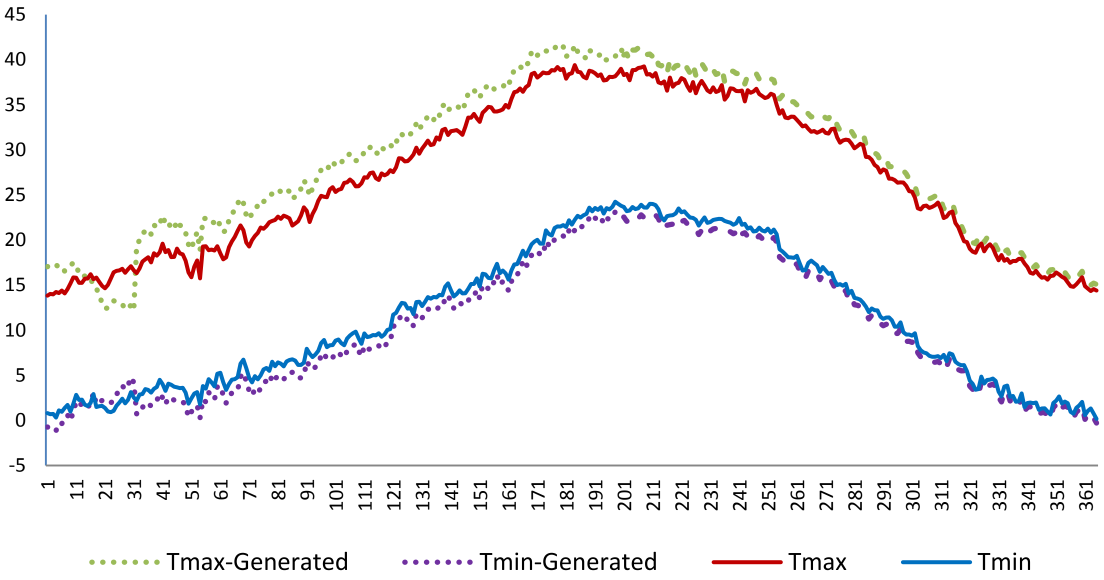


Fig. 4.2. Comparison of daily minimum and maximum temperature between actual and generated data

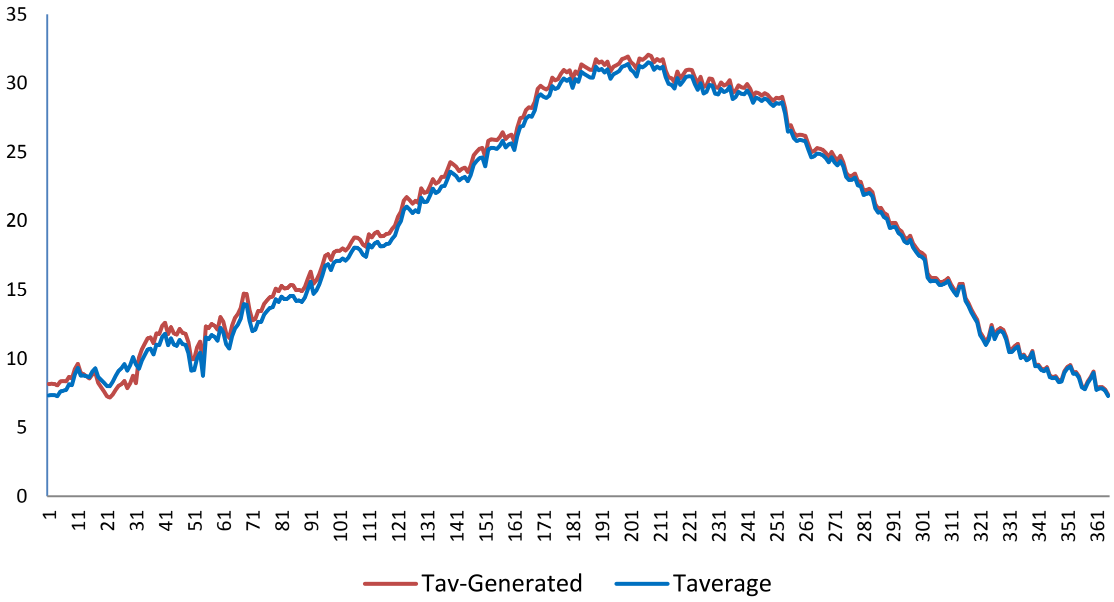


Fig. 4.3. Comparison of average daily temperature between actual and generated data

4.1.2 Rain and Number of Rainy Days

A module within the program developed to generate rainy days and rain quantity based on the real weather data extracted for each of the climate zones mentioned above (Fig. 4.4). The code start at day 1 of the year and examines if it is a rainy day or not, and keeps moving day-by-day until finding the first day with some rain (a precipitation event). At this point the program starts a "rainy days" range (J_1), and moves to find the following consecutive days with rain until finding the next day without rain. At this point the program determines the end of the range (J_2). The program then randomly determines an adjusted range up to ±3 days to be added to the beginning and the end of the range. The program makes sure that the range does not become less than zero due to the randomization process. If that happens, the program assigns a zero quantity to the days within the range.

If the program generates a range of rainy days, with at least one day with rain within the new range, then it generates a magnitude for the amount of precipitation for each day. The program calculates the average precipitation from the actual data for the new range, and multiplies the average with a generated random number from -0.5 to 0.5. This adjusted average is then added to the actual amount of precipitation for that day within the range. If the day is a newly generated rainy day within the range (i.e. there was no measured precipitation on this day), then the amount of precipitation assigned to that day will be equal to the average rainfall depth in that range, plus the adjusted average. The program makes sure that the precipitation quantity never goes below zero. After finishing the first range, the program follows the same procedure for finding and modifying another range, until it reaches the end of the year. This procedure also repeated for all climate zones and the results are recorded for later use.

Accordingly, a new set of rainy days and precipitation quantity data will be generated each time the game is played, and used in the technical model for the calculation of daily soil water balance and other uses in the program, like weather forecasting. Other options could be investigated for the generation of rainy days, but this is the method developed for use in WaterMan. Figure 4.4 summarizes the coding procedure for rainfall modifications.

Figure 4.5 shows sample result of generated daily rainfall (days and quantity), as compared to real (measured) data.

4.1.3 Five-day Weather Forecast

To simulate the reality of the fact that the farmers usually could have access to a weather forecast that they choose to view, the model gives the player access to a five-day weather forecast. The player can see the expected maximum and minimum air temperatures, in addition to the possibility of having rainy days during the coming five days, with the respective probabilities of occurrence.

To code this part, the same procedure used to generate new sets of weather data (maximum temperature, minimum temperature, and precipitation days and amount), each time the game is played, was used to generate the weather forecast data. The only exception is that the weather data source used to generate the forecast data is the new sets of that were randomly generated, and not the actual data. This is to increase the uncertainty of the forecasted data, but within a controlled limit. After all, it is only a forecast, and it is recognized that real weather data forecasts are often incorrect, even in the short term. After preparing the database values for the parameters, the daily soil water balance is performed, as described below.

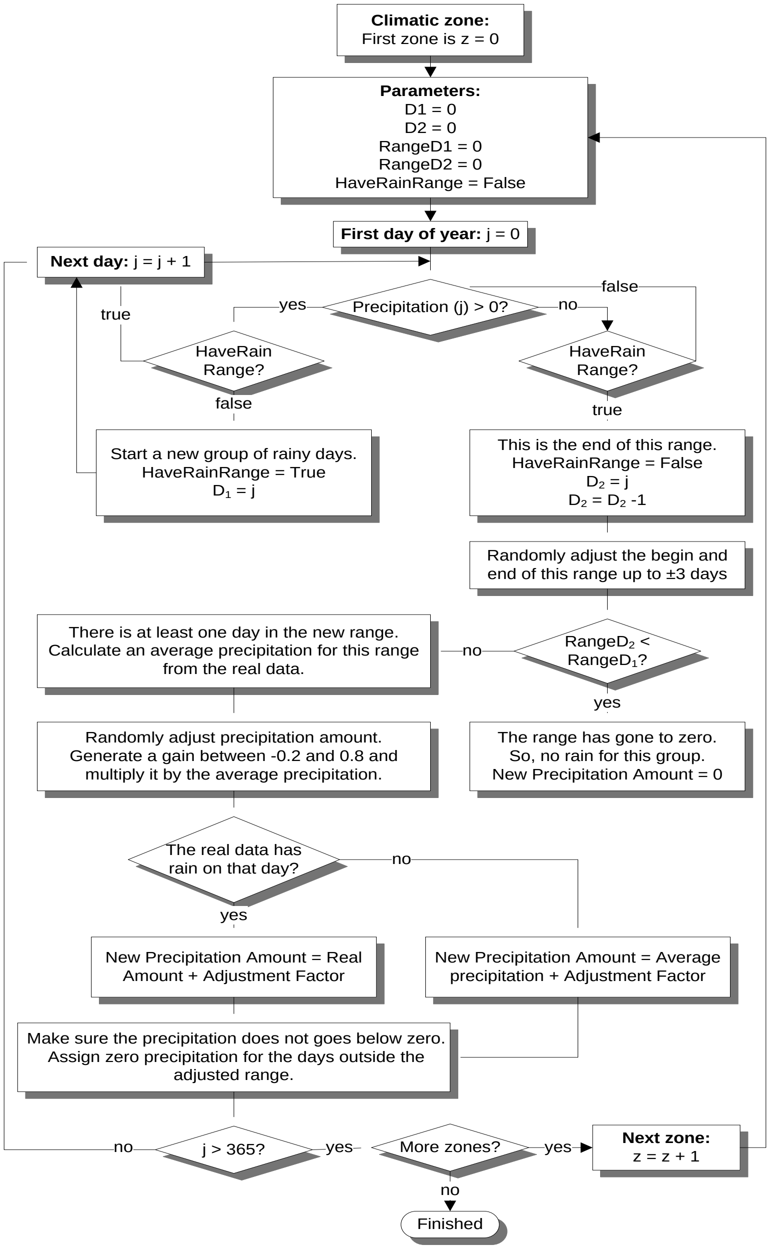


Fig. 4.4. Flow diagram of the process to randomly generate modified sets of rain and number of rainy days

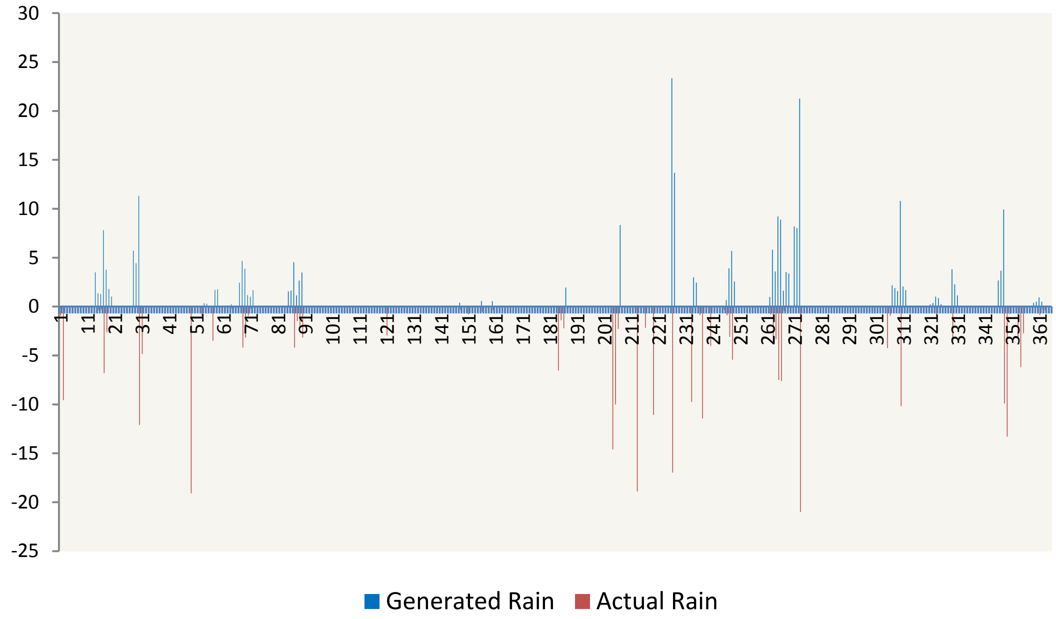


Fig. 4.5. Comparison of rain quantity and number of rainy days between actual rain data and generated rain data

4.2 Optimization of Irrigation Scheduling

Building a crop production function that describes the relationship between irrigation water and crop yield under the assumption of optimal irrigation scheduling (timing and quantity) is the basic objective in the set of objective functions that are given to the player to choose from, and is considered to be the main target for improved water management. Crop production function is defined as the amount of crop produced per unit volume of water. As such, maximizing the crop production function reflected in the optimal relation between maximizing crop yield and minimizing the amount of irrigation water used. The golden section search algorithm was applied at the beginning of the growing season to search for the optimal crop production function.

4.2.1 Golden Section Search Algorithm

In applying the golden section search algorithm to solve an optimization function (function minimization) the decision variable must be bracketed by minimum and maximum values. The minimum value is set to 10% and the maximum is 100%, as a percentage of the net amount of irrigation water to restore the soil water content to field capacity. This percentage is subject to the constraint of total water availability throughout the irrigation season, and for each individual irrigation. The field capacity limit was used because any additional water above this limit is considered to be wasted, as it will go to deep percolation, and that will affect the crop yield inversely so it will not be part of the optimal solution to exceed that limit when searching for the best solution. The minimum percentage could be less than 10%, but as long as there will be irrigations, this minimum is reasonable (why would anyone irrigate a crop to achieve less than 10% of the potential crop yield?). Figure 4.6 shows the golden section optimization coding procedure.

4.2.2 Objective Function Coding

Description of the coding method of the objective function with using one decision variable and different constraints is as given in the following. For maximizing the crop yield function (Fig. 4.7):

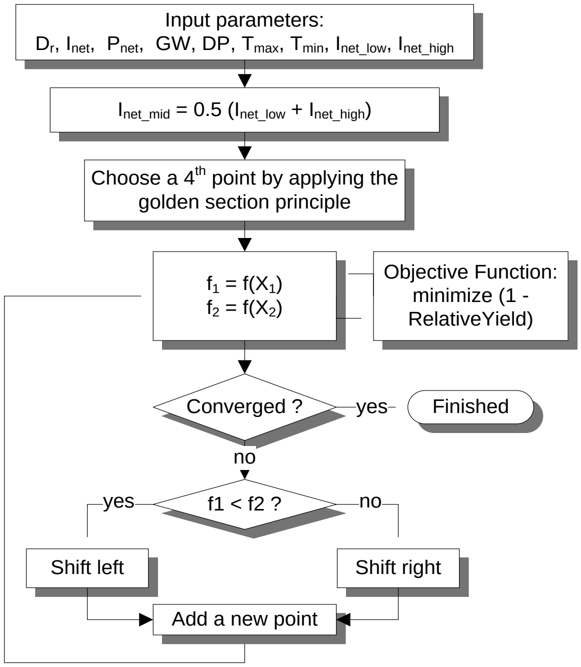


Fig. 4.6. Golden section optimization coding procedure

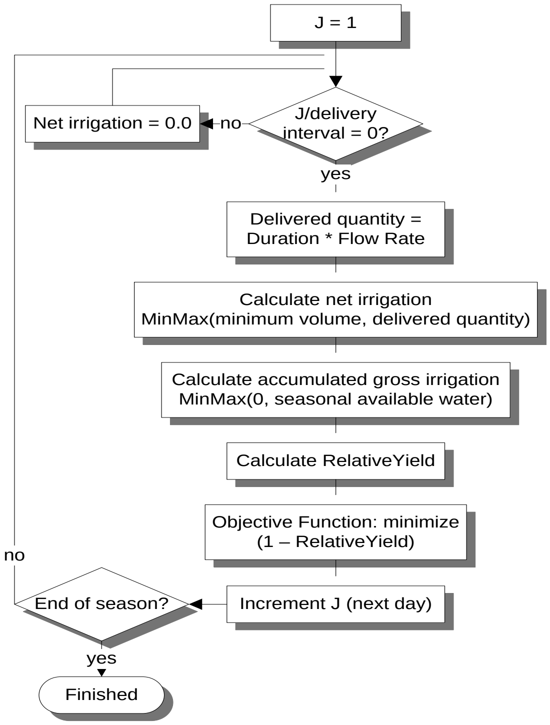


Fig. 4.7. Flowchart describing the logic used to determine maximum crop yield

At each day when there is possibility for irrigation water to become available (depending on the delivery interval, as set by the player or by the program for fixed rotation delivery method, or according to certain constraints based on the irrigation method for the on-demand delivery method), do the following:

1. Determine the amount of water delivered: quantity = duration * flow rate;
2. Search for the best value of the decision variable; percentage of irrigation water to raise the water content to FC:

$$Inet(J) = 0.01 * Ipercentage * (FC - Theta(J-1) * R_z(J-1) * 1000 \tag{4.1}$$

3. Set limits for that quantity (minimum and maximum), where minimum is the net flow rate multiply by the minimum irrigation duration, and maximum is the net delivered quantity;
4. Calculate the accumulative gross amount (volume) of irrigation water used;
5. Assign a variable to calculate the accumulated total amount of water used at each irrigation during the season in order to limit (constraints) it to the total available water for the whole season;
6. Apply the daily soil water balance technical module to calculate the relative yield objective function:

$$\text{minimize: YieldReduction} = 1 - \text{RelativeYieldObtained} \tag{4.2}$$

7. Minimize the crop yield reduction, which is equivalent to maximizing the relative crop yield, “Relative Yield = 1 - Yield reduction”;
8. Return the numerical value of the yield reduction.

For maximizing the crop production function (Fig. 4.8):

At the day when water could be available (depending on the delivery interval, as set by the player or by the program for a fixed rotation delivery method, or according to certain constraints based on the irrigation method for the on-demand delivery method), do the following:

1. Determine the mid-point between field capacity and readily available water; that is, $\frac{1}{2}(\theta_{FC} + \theta_{RAW})$;
2. Do not irrigate if the soil water content is above the mid-point, if water delivery method is fixed rotation, or do not irrigate if the soil water content is above θ_{RAW}, if water delivery method is on-demand;
3. Determine the delivered irrigation water quantity: duration * flow rate;
4. Search for the best decision variable. This is the best percentage of net irrigation water that is required to raise the soil water content, from its current amount, to field capacity. It will be the same percentage for all irrigations throughout the growing season.

$$Inet(J) = 0.01 * Ipercentage * (FC - \theta(J-1) * R_z(J-1) * 1000 \qquad (4.\ 3)$$

5. Specify a range for that quantity whereby “min” is the minimum irrigation duration multiplied by the net flow rate, and “max” is the net delivered quantity;
6. Calculate the accumulative gross amount of irrigation water used;

7. Assign a variable to calculate the accumulated total amount of water used in each irrigation during the season so that it does not exceed the total available water for the whole season;
8. If the delivery interval less than 15 days, set a limit on the amount of the last irrigation, so as to not exceed the refilling of the soil water content to the mid-point between θ_{FC} and θ_{RAW};
9. Apply the daily soil water balance technical module and calculate the relative crop yield objective function:

$$\textbf{minimize: YieldReduction} = \textbf{1 - RelativeYieldObtained} \tag{4. 4}$$

10. Minimize the crop yield reduction, which is equivalent to maximizing the relative yield (1 - yield reduction);
11. Calculate the best production function by dividing the crop yield by the total amount of irrigation water used; and,
12. Return the calculated yield reduction.

Figure 4.9 shows sample results obtained from playing the game when the objective function is to maximize profit. The score is indicated as “89” because the player obtained a net profit of \$3,093, while the program obtained \$3,484 (the ratio is equal to 0.89). The graph also shows the player’s results in terms of relative crop yield and production function, compared to the “best” results from the model, using the same parameters with the search algorithm.

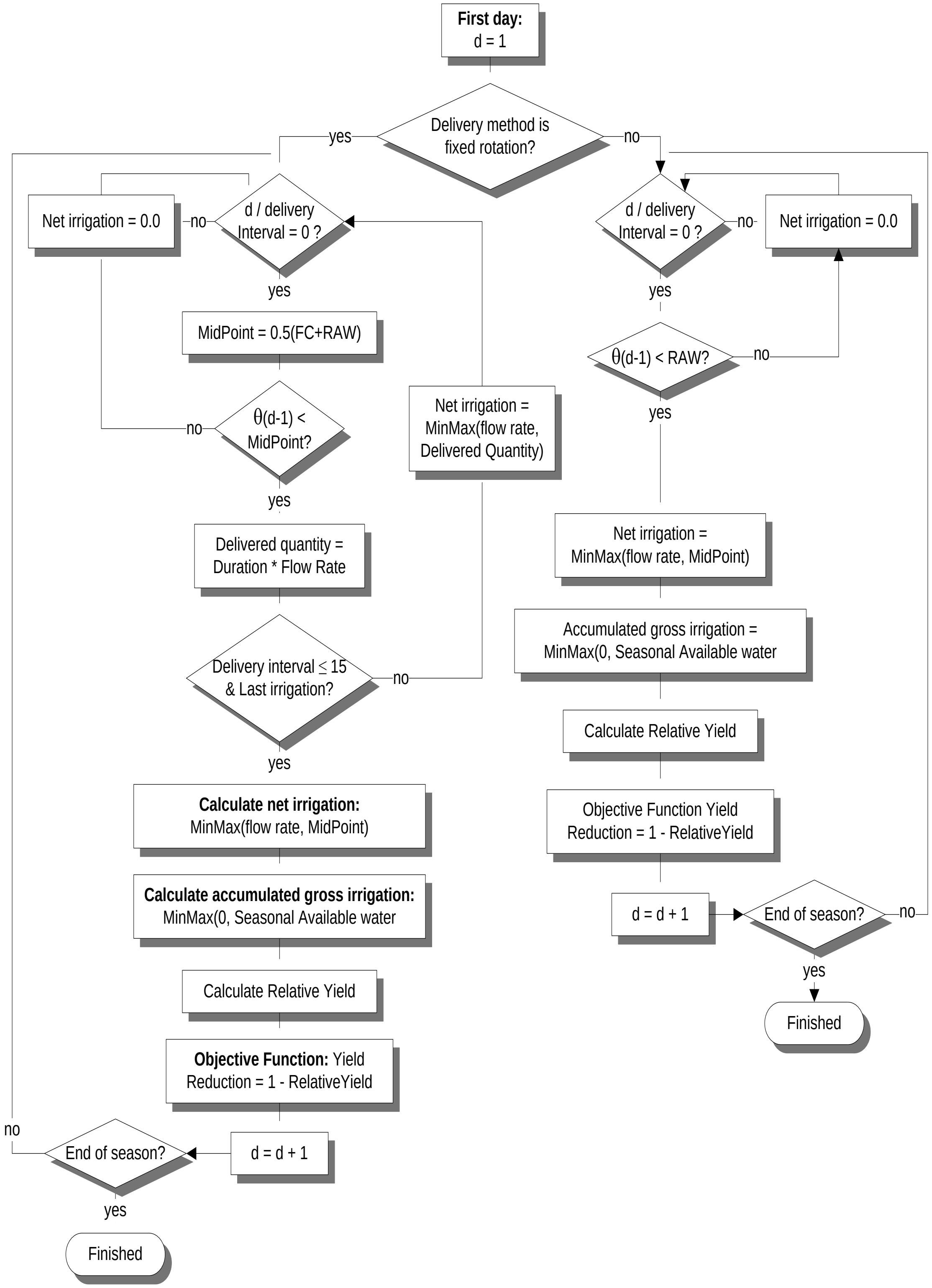


Fig. 4.8. Flowchart describing the logic used to maximize the crop production function

Recommended Results

Irrig Percentage 20.763 %
Total Water Applied 38,743 m3
Relative Yield 100.00 %
Production Function 1.29 kg/m3
Net Profit $ 3484.00

Score 89.0

Player Results

Section Index 1
Sec. Relative Yield 97.7 %
Sect. Water Applied 15,400 m3
Total Water Applied 61,600 m3
Total Relative Yield 97.7 %
Production Function 0.8 kg/m3
Net Profit $ 3093.08

Fig. 4.9. Sample results using the profit maximization objective function

4.3 Player Type and Random Event Generation

The main objective from generating random events within the game play is to reflect some of the real conditions that might occur during a cropping season, and to show the difficulties that a decision maker might face about irrigation scheduling. As a training tool, this game will be played by people with different irrigation management skills and experience. In order to make the game more interesting, the program is expected to anticipate the player skill level, and to reply with events appropriate to the anticipated level. The following shows the approaches followed in the game program to achieve this goal.

4.3.1 Artificial Intelligence Simulation

As briefly described in the previous chapter, the heuristic features of the program were developed based on a combination of two artificial intelligence approaches: (1) a pattern recognition approach; and, (2) reinforcement learning based on Markov Decision Processes, specifically, the Q-learning method. These two approaches were taken to account for the difference in the effect of actions taken by the player and action taken by the system on the game world. The pattern recognition part of the program is described as being responsible for developing a player model and the determination of the player type, then passes this information to the decision-making system. The decision-making system is responsible for learning an action Q-function, and then choosing and executing an action at the particular state based on a maximization of the utility function (maximum Q-Value). The following section describes the problem setup within the game program after giving an overview about the reinforcement learning approach and some definitions for the terms that are used.

Reinforcement Learning/MDP Approach Overview

The specification of a sequential decision problem for a “fully observed” environment with a Markovian transition model and additive reward is called a Markov Decision Process, or MDP. The environment is considered fully observed if the agent can see all the parameters that form the environment state at each time interval, and knows its current state. In the artificial intelligence field of research, Markov Decision Processes have become the most popular framework for representing and solving problems of sequential decision under uncertainty (Sigaud and Buffet 2010). Markov decision processes are defined as controlled stochastic processes satisfying the Markov property in the sense that the probability of reaching the next state of the environment from the current state depends only on the current state and not on the history of earlier states, the future is independent of the past given the present (Russell and Norvig 2003). MDP is defined by the following three components; initial state, transition model, and reward model. The states of the environment (S) represent the state space where actions (A) take place. The initial state is the starting state of the environment.

The transition probability $p(\ )$ characterizes the state dynamics of the system, and indicates which states are likely to appear (new state (s')) after taking an action (a) in the current state (s) and are represented as $p(s', a, s)$. As a result of choosing action (a), in state (s), the deciding agent receives a reward (r) = $r(s , a)$. The reward function could be positive or negative depend on how the action positioned the new state of the environment closer to the goal state.

The idea that we learn from interaction with our environment is probably the first to occur to us when we think about the nature of learning. Reinforcement learning is a computational approach that involves learning while interacting with the environment through selecting the appropriate action in a particular state so as to maximize a numerical reward signal. The agent is not told which action to take, but instead must discover the action that yields the maximum reward by trying all possible actions in that state. Sutton and Barto (1998) define four elements for reinforcement learning; a policy, a reward function, a value function, and a model of the environment. The policy is identifying the best actions to be taken at a particular state. The reward function reflects how good it is to perform a given action in a given state and redefines the state-action pair into a single number. The value function reflects the accumulated reward from being in a particular state given the chosen behavior. The model of the environment reflects the ability of the model to predict the resulting next state given the current state and action. Reinforcement learning methods are said to be model-free or model-based depending on whether they build a model of the transition and reward function $p(s' , a, s)$ and $r(s ,a)$ of the underlying MDP.

Implementation of the AI approaches in WaterMan

In order to develop the artificial intelligence capability of the software, the human-computer interactions were modeled as sequential decision problems under uncertainty (Hui and Boutilier 2006). The logic behind this approach is strongly dependent on ideas presented in the book by Sigaud and Buffet (2010), Chapters 1 and 2, and on Russell and Norvig’s book (2003), Chapters 17 and 21. The problem is classified as sequential because the current decision has a future impact on the consequent decisions and the overall results, and the system continuously moves from state to state within the game world. At each step of this sequence, the agent (decision-maker) needs to decide on the current action by taking into consideration its future impact. The uncertainty comes from the lack of knowledge about action effects on the game environment (world states).

Game Environment

Based on the definition of the agent as the decision maker whose decisions directly affect the environment and move it to a new state, the computer system in our case will be considered as our agent because it will decide which random event to be generated based on the current state and the expected reward from taking this action. The human player can also be considered as an agent because of his/her decision of whether to irrigate or not at a given state, or to agree (or not) to the option given by the system, will also affect the environment and determine the next state. For the purpose of the game implementation herein, the player decisions were considered as part of the environment after categorizing them in patterns to reflect the player types.

Game States Representation

As defined previously, the state is a representation of the environment at each time step. The game states were defined based on three parameters: (1) soil water content (four different levels); (2) irrigation water availability (binary: available or not); and, (3) the daily score (above 85, equal to 85, and below 85). Accordingly, the environment in the model consists of 24 different states.

System (Agent) Actions

Accordingly, sixteen different random messages (including the "no action" message) and their options are given to the player by the system. Most of the messages have strategic options from which the player can choose. Based on the action and the selected option of the player, the game environment is expected to either move to a new state or remain in its current state.

Player Types Identification

Four criteria were used to define the player types: (1) the game state when the player decides to irrigate; (2) if all or part of available water is used per irrigation; (3) whether the player checks the weather forecast or not; and, (4) if the player seems to have a goal and changes the default values given by the program. Accordingly, four player types are defined in the model:

1. Risk averse: The player always irrigates at high soil water level, and whenever water is available, if the delivery method is fixed rotation, with complete irrigations all the time.
2. Risk neutral: The player does not irrigate when the soil water content is very high, in spite of water availability, and checks the weather forecasts from time to time.
3. Risk taker: The player delays irrigations until all the readily available water is consumed, or allows soil water content to drop below RAW for more than five consecutive days during the season, the player might skip some irrigations at critical soil water contents, does not check the weather forecast at all.
4. Strategic: The player is risk neutral and also goal-oriented through changing default values, checking weather forecasts more often, never irrigate at high soil water level, not necessarily using all available water per irrigation when

the soil water level is not critical. A risk taker or risk-averse player is never considered to be a “strategic” player by the game.

The player type updated continuously each time an irrigation event is occur throughout the simulated season. Also, the program was made able to define and return a player type, even if there was some ambiguity occur at one of the identification processes.

Defined Player Actions

Unrepeated action: change default values, which can happen only once, at the beginning of a simulation. These repetitive actions are checked daily:

- Irrigating
- Not irrigating
- Ordering irrigation water
- Checking the weather forecasts

4.3.2 Problem Setup Within WaterMan Game

After specifying the game environment, game states, and agent (system) actions, in addition to player types and actions, the setup of the artificial approaches within the WaterMan game is described in the following parts.

Markov Decision Processes are dynamic programming described by the 5-tuple (S, A, T, p, r), where S, is a finite set of world states in which the process evolution takes place; A, is a finite set of all possible actions available for the agent (random events); T, is the set of time steps where decisions need to be made (crop growing season); p is the transition probability function of the world states, *p(s'/s, a)*, representing the probability of transitioning to state *s'* given that action a is taken in state s (unknown and should be learnt from the reward function); and, *r* is the reward function that returns all possible actions over a particular state as real numbers (the daily score after taking the action).

Value Function

The value function is calculated in the MDP using the transition probability of the world states and the reward function of state action pair, based on the following equation.

$$V(s) = \left(r(a,s) + \gamma \sum_{s' \in S} p(s'/a,s)\, V(s') \right) \tag{4. 5}$$

where V(s) is the utility function of the state s; γ, is discount factor, set between 0 and 1; p(s'/s, a) is the transition probability of moving to state s' after taking action a in state s; and, V(s') is the value of function when in the new state.

As observed in the above equation, the system should have a prior knowledge about the transition function in order to calculate the expected value function. In the case of the WaterMan game, the transition function is unknown previously by the system due to unpredicted player actions. Therefore, an artificial intelligence method was found that follows the Markov decision property needed due to the stochastic nature of the WaterMan game, but does not need a prior knowledge of the transition model of the world. This method is known as the Q-learning method, which belongs to an artificial intelligent approach called Reinforcement Learning.

In the Reinforcement Learning Approach, the transition functions are not necessary to be known previously by the system and could be learnt by interacting with the environment (world states). In this method, instead of calculating the value function, a utility function is calculated based on the following equation:

$$U(s) = \alpha\ (r + \gamma U(s') - U(s)) \tag{4.6}$$

where U(s) is the utility function of the state s; α, is the learning rate set as 1, usually set between 0 and1, a high value means that learning can occur quickly; γ, is discount factor, set as 0.5. This models the fact that future rewards are worth less than immediate rewards; U(s') the utility function of being in the new state.

Q-Learning Method and Algorithm

The Q- Learning method is a method in the reinforcement learning approach that learns an action-value representation instead of learning the utility of the action. The Q-values represented as Q (a, s) to denote the values of performing an action: (a) in state, (s), and are directly related to utility values as follows:

$$U(s) = \max_a Q(a, s) \tag{4.7}$$

The Q-learning method is considered a model-free method (Russell and Norvig 2003), but due to the special characteristics of the WaterMan game, and because not all actions are possible in all states, the game environment was given a kind of a model through grouping the set of actions available for the system into three categories in order to link each state of the world with its possible action. Furthermore, this was done to minimize the searching cost for the best action, and to make the produced actions more reasonable to the state. The criteria for the three categories are: the player’s irrigation decisions; the player’s water orders with the on-demand delivery method, or the water availability under the fixed rotation delivery method; and, a general group of actions that can take place at any time during a simulation.

The sequence of the Q-learning algorithm is as follow (Fig. 4.10):

1. Initialize the Q-values table, Q (a, s), and the reward value table, R (a, s).
2. Return the current state ID (s).
3. Does the current state match the required states category?
4. Choose an action, (a), from the possible actions set, (A), for that state based on the ε-greedy policy, in which most of the time the action associated with the highest reward value is chosen.
5. Execute the selected action, and observe the reward, (r), as well as the new state, (s').
6. Update the Q-value for the state using the observed reward and the Q-value of the next state, according to the following equation:

$$Q_{new}(a,s) = Q_{current}(a,s) + \left(\alpha\ (r(s) + \gamma \max_{a} Q(a',s') - Q_{current}(a,s)\right) \qquad (4.8)$$

7. Repeat the process daily until the end of the growing season.

Reward Function

The system action is evaluated based on the reward that will be obtained from moving the game environment from the previous state to the current state. Due to the nature of the WaterMan game as a training tool, it was designed to be neither competitive game, in which the system will be rewarded if it causes the human player to lose, nor cooperative game, in which the system is rewarded if it cooperates with the human player to help him or her achieve a goal. Instead, the WaterMan game required to be competitive when the player is performing well and to be cooperative when the player is not achieving well. The player achievement is measured based on his/her daily score. To deal with this special case, the WaterMan game reward function is constructed based on the daily score of the player. The score range for the game is set numerically between 70 and 100.

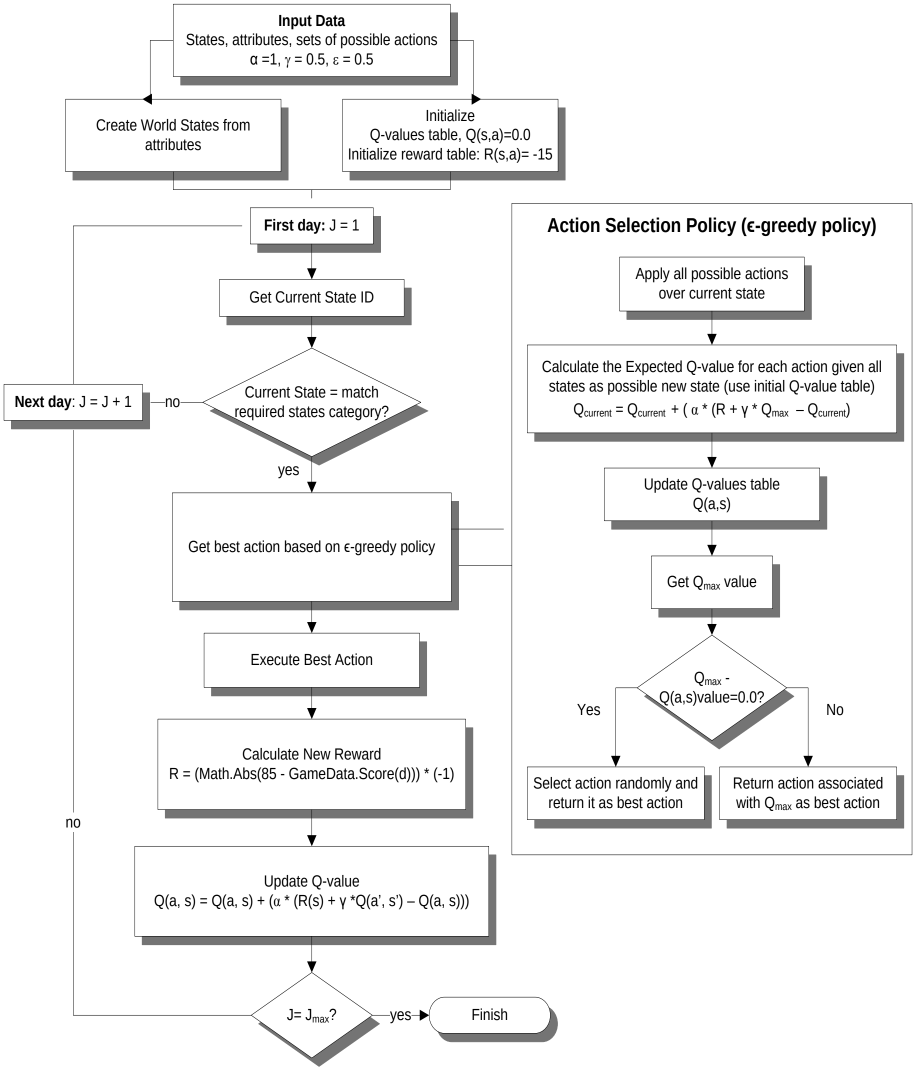


Fig. 4.10. Flowchart of the Q-learning algorithm

The goal score for the system to achieve the highest reward was set to 85. So, if the player performs in a way that maintains a high score, then the system will be rewarded if it generates actions to make the daily score drop closer toward 85. This is accomplished by generating more challenging random events. If the player performance is poor (e.g. a daily score of less than 85), then the model will generate assistive actions to help the player improve the score. The real number reflecting the achieved reward ranges between -15 and 0. It might seem that the negative reward is a penalty and not a reward, but the fact that minimizing the penalty can be considered as maximizing the reward allows the use of this concept.

To achieve this concept, the reward function was calculated based on the following maximization equation:

$$R = -\left|85 - Score(day)\right| \tag{4. 9}$$

where R is a real number representing the reward; and, Score(day) is the daily score obtained by the player.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Technical Results Validation

Throughout the development of the game’s technical module, the calculation procedures for all the input parameters were tested, both individually and in combination with other related parameters by applying different sets of data and observing the impact on the daily soil water balance calculations. To demonstrate the performance of the technical module in calculating the daily soil water balance, a validation test was carried out for the game’s technical module output results.

The validation of the output results of the technical module of the WaterMan game was carried out in two ways: (1) conducting mass-balance calculations over a period of time, during a crop growing season, for the daily soil water and salt contents; and, (2) comparing the game-generated results of irrigation water requirements and irrigation scheduling calculations with those generated by the FAO CropWat 8 software using the same set of data. The following section shows the validation steps and results.

5.1.1 Mass Balance Calculations

Mass Balance for Daily Soil Water Content

Table 5.1 shows the results obtained by estimating the mass balance of soil water content over a period of 30 days during a cropping season for spring wheat growing under Mediterranean climatic conditions.

Table 5.1. Calculations of the soil water mass balance over a period of 30 days (all values are in kg/m^2)

Inf. Rain	Total Irrig	Ground Water	ETa	Deep Perc.	Ponded Water	RunOff	Mass_In - mass_Out	Storage change	Error
0.00	0.00	0.00	4.74	0.00	0.00	0.00	-4.74	-4.74	0.00
0.00	0.00	0.00	4.47	12.94	0.00	13.20	57.38	57.38	0.00
0.00	88.00	0.00	4.60	0.00	0.00	0.00	-4.60	-4.60	0.00
0.00	0.00	0.00	4.30	0.00	0.00	0.00	-4.30	-4.30	0.00
0.00	0.00	0.00	4.14	0.00	0.00	0.00	-4.14	-4.14	0.00
0.00	0.00	0.00	4.03	0.00	0.00	0.00	-4.03	-4.03	0.00
0.00	0.00	0.00	4.22	0.00	0.00	0.00	-4.22	-4.22	0.00
0.00	0.00	0.00	3.88	0.00	0.00	0.00	-3.88	-3.88	0.00
0.00	0.00	0.00	3.70	0.00	0.00	0.00	-3.70	-3.70	0.00
0.00	0.00	0.00	3.68	0.00	0.00	0.00	-3.68	-3.68	0.00
0.00	0.00	0.00	3.34	0.00	0.00	0.00	-3.34	-3.34	0.00
0.00	0.00	0.00	3.19	0.00	0.00	0.00	-3.19	-3.19	0.00
0.00	0.00	0.00	3.17	0.00	0.00	0.00	-3.17	-3.17	0.00
0.00	0.00	0.00	3.07	0.00	0.00	0.00	-3.07	-3.07	0.00
0.00	0.00	0.00	2.83	0.00	0.00	0.00	-2.83	-2.83	0.00
0.00	0.00	0.00	2.74	0.00	0.00	0.00	-2.74	-2.74	0.00
0.00	0.00	0.00	2.68	19.45	0.00	13.20	52.68	52.68	0.00
0.00	88.00	0.00	2.48	0.00	0.00	0.00	-2.48	-2.48	0.00
0.00	0.00	0.00	2.49	0.00	0.00	0.00	-2.49	-2.49	0.00
0.00	0.00	0.00	2.12	0.00	0.00	0.00	-2.12	-2.12	0.00
0.00	0.00	0.00	2.12	0.00	0.00	0.00	-2.12	-2.12	0.00
0.00	0.00	0.00	1.98	0.00	0.00	0.00	-1.98	-1.98	0.00
0.00	0.00	0.00	1.87	0.00	0.00	0.00	-1.87	-1.87	0.00
0.00	0.00	0.00	1.70	0.00	0.00	0.00	-1.70	-1.70	0.00
0.00	0.00	0.00	1.57	0.00	0.00	0.00	-1.57	-1.57	0.00
0.00	0.00	0.00	1.44	0.00	0.00	0.00	-1.44	-1.44	0.00
0.00	0.00	0.00	1.31	0.00	0.00	0.00	-1.31	-1.31	0.00
0.00	0.00	0.00	1.20	0.00	0.00	0.00	-1.20	-1.20	0.00
0.00	0.00	0.00	1.05	0.00	0.00	0.00	-1.05	-1.05	0.00
0.00	0.00	0.00	0.94	0.00	0.00	0.00	-0.94	-0.94	0.00

The calculations are based on the following mass-balance concept:

$$mass\_in - mass\_out = change\,in\,storage \qquad (5.1)$$

where mass_in (kg) is the mass of effective rain + total irrigation + ground water contribution); mass_out (kg) is the mass of actual evapotranspiration + deep percolation + ponded water + runoff; and, change in storage (kg) is the change in the mass of soil water in the crop root zone.

It can be concluded that the technical module of the game is performing the daily soil moisture content calculations in an acceptable manner based on the calculation of the difference between the two parts of the above equation, which was equal to zero.

Mass Balance for Daily Salt Content

Table 5.2 shows results obtained by estimating the mass balance of the salt content over a period of 30 days during the cropping season of spring wheat growing under Mediterranean climatic conditions. The concept of the calculations was the same as in the calculation of daily soil water content.

It can be concluded that the technical module of the game is correctly performing The daily soil electrical conductivity (EC_e) calculations based on the calculation of daily changes in root zone salt content. The difference between the calculated soil electrical conductivity and that obtained by the model is almost zero every day of the studied period.

Table 5.2. Calculations of the mass balance of salt content over a period of 30 days (all headings and their units are explained beneath the table)

ECe_{model}	Rz	LR	EC dp	Net Irrig	Gr. Water	D. Perc.	Δ Rz	ΔS	Salt	ECe_{calc}	Error
0.88	1.25	0.29	3.42	0.00	0.00	0.00	0.00	0.00	283004.85	0.88	0.00
0.95	1.25	0.27	3.73	74.80	0.00	12.94	0.00	16997.06	300001.91	0.94	-0.01
0.95	1.25	0.27	3.73	0.00	0.00	0.00	0.00	0.00	302529.10	0.95	0.00
0.95	1.25	0.27	3.73	0.00	0.00	0.00	0.00	0.00	302529.10	0.95	0.00
0.95	1.25	0.27	3.73	0.00	0.00	0.00	0.00	0.00	302529.10	0.95	0.00
0.95	1.25	0.27	3.73	0.00	0.00	0.00	0.00	0.00	302529.10	0.95	0.00
0.95	1.25	0.27	3.73	0.00	0.00	0.00	0.00	0.00	302529.10	0.95	0.00
0.95	1.25	0.27	3.73	0.00	0.00	0.00	0.00	0.00	302529.10	0.95	0.00
0.95	1.25	0.27	3.73	0.00	0.00	0.00	0.00	0.00	302529.10	0.95	0.00
0.95	1.25	0.27	3.73	0.00	0.00	0.00	0.00	0.00	302529.10	0.95	0.00
0.95	1.25	0.27	3.73	0.00	0.00	0.00	0.00	0.00	302529.10	0.95	0.00
0.95	1.25	0.27	3.73	0.00	0.00	0.00	0.00	0.00	302529.10	0.95	0.00
0.95	1.25	0.27	3.73	0.00	0.00	0.00	0.00	0.00	302529.10	0.95	0.00
0.95	1.25	0.27	3.73	0.00	0.00	0.00	0.00	0.00	302529.10	0.95	0.00
0.95	1.25	0.27	3.73	0.00	0.00	0.00	0.00	0.00	302529.10	0.95	0.00
0.95	1.25	0.27	3.73	0.00	0.00	0.00	0.00	0.00	302529.10	0.95	0.00
0.95	1.25	0.27	3.75	74.80	0.00	19.45	0.00	1193.77	303722.87	0.95	0.00
0.95	1.25	0.27	3.75	0.00	0.00	0.00	0.00	0.00	304011.09	0.95	0.00
0.95	1.25	0.27	3.75	0.00	0.00	0.00	0.00	0.00	304011.09	0.95	0.00
0.95	1.25	0.27	3.75	0.00	0.00	0.00	0.00	0.00	304011.09	0.95	0.00
0.95	1.25	0.27	3.75	0.00	0.00	0.00	0.00	0.00	304011.09	0.95	0.00
0.95	1.25	0.27	3.75	0.00	0.00	0.00	0.00	0.00	304011.09	0.95	0.00
0.95	1.25	0.27	3.75	0.00	0.00	0.00	0.00	0.00	304011.09	0.95	0.00
0.95	1.25	0.27	3.75	0.00	0.00	0.00	0.00	0.00	304011.09	0.95	0.00
0.95	1.25	0.27	3.75	0.00	0.00	0.00	0.00	0.00	304011.09	0.95	0.00
0.95	1.25	0.27	3.75	0.00	0.00	0.00	0.00	0.00	304011.09	0.95	0.00
0.95	1.25	0.27	3.75	0.00	0.00	0.00	0.00	0.00	304011.09	0.95	0.00
0.95	1.25	0.27	3.75	0.00	0.00	0.00	0.00	0.00	304011.09	0.95	0.00
0.95	1.25	0.27	3.75	0.00	0.00	0.00	0.00	0.00	304011.09	0.95	0.00
0.95	1.25	0.27	3.75	0.00	0.00	0.00	0.00	0.00	304011.09	0.95	0.00

EC_e is soil extract salinity (dS/m); R_z is the root zone depth (m); LR is the leaching requirement, which is the portion of irrigation water that should pass through the root zone to prevent excessive accumulation of salts (%); EC_{dp} is the salinity of deep percolated water (dS/m); Net Irr. is the net irrigation (kg); Gr. Water is the ground water contribution to the root zone (kg); D.Perc. is the deep percolation water (kg); ΔR_z is the daily change in the root depth (m); ΔSR_z is the change in salt quantity due to root depth increase (mg); ΔS is the amount of salt added to the root zone (mg); Total ΔS is the total amount of salt added to the root zone including the root growth effect; Salt is the daily salt balance in the root zone (mg); Δ stored Salt is the change in salt quantity storage in the root zone (mg); and, Error is the calculated difference between the salt mass added or removed and the mass stored in the root zone (mg).

5.1.2 Comparison with CropWat 8

CropWat 8, as defined on the web page www.fao.org/nr/water/, is a decision support system developed by the Land and Water Development Division of FAO as a tool for crop water requirements and irrigation requirements. It calculates crop water requirements and irrigation requirements based on soil, climate, and crop data as entered by the user, and it uses the Penman-Monteith equation for the calculation of reference evapotranspiration (ET_o).

Because CropWat 8 and the WaterMan technical module base their calculations on the same procedures presented in FAO Irrigation and Drainage Papers 56 and 33;

therefore, it is reasonable to use results obtained from the CropWat 8 software to compare and validate the output results of the technical module of the WaterMan game.

Results of irrigation water requirements and irrigation scheduling for corn and wheat crops by using both the CropWat 8 and WaterMan technical module were compared in order to test the ability of the WaterMan technical module in performing accurate calculations for these two irrigation management parameters. All calculations on this section, were performed using weather data from Delta, UT, for the year 2010. The reference evapotranspiration calculations were based on the Penman-Monteith equation. Following are the comparison results that were obtained.

Irrigation Water Requirements

Crop water requirements are the amount of irrigation water needed for optimal plant growth. The irrigation water requirements can be calculated by subtracting any amount of water added to the soil root zone through rain or ground water from crop evapotranspiration. In the absence of any ground water contribution to the soil root zone, the irrigation water requirement is calculated by subtracting the effective rain from crop evapotranspiration.

Corn Crop

Figure 5.1 shows a comparison between calculated actual crop evapotranspiration (ET_a) for corn from CropWat 8 and the WaterMan technical module. As seen in Fig. 5.1, both applications gave almost the same trend and magnitudes for ET_a.

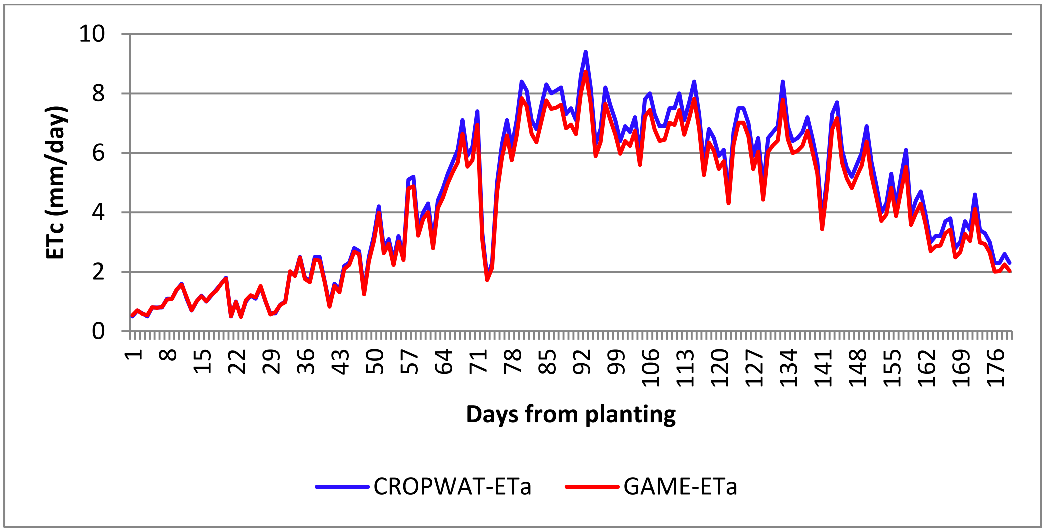


Fig. 5.1. Actual daily evapotranspiration for corn

Considering that no stress occurred during the growing season, the calculation of crop water requirements was performed by subtracting the effective rainfall quantity from the actual evapotranspiration (ET_a). The comparison between the calculated results of irrigation water requirements for corn are shown in Table 5.3.

The difference in the obtained results of the crop irrigation requirement was found to be due to the difference in the calculation of the effective rain quantity, and also the K_c values between the two applications. Although the same formula for effective rain calculation is used in the two applications (the FAO/AGLW formula), it is calculated on a decade basis in CropWat 8, while in the WaterMan technical module it is calculated on a daily basis because of the requirements of the daily nature of the game play. The K_c values given in CropWat 8 were found to be higher than that calculated in the WaterMan technical module, especially in the Mid and Late seasons.

Table 5.3. Irrigation water requirements for corn

Parameter	CropWat	WaterMan
Accumulative ET_a (mm)	808.0	756.0
Effective Rain (mm)	46.3	71.8
Irrigation Requirements (mm)	765.9	684.2
Difference in Irrig. Req		**-10.7%**

Wheat Crop

To confirm the results, another run for the same ET_o data were used to calculate the crop irrigation requirements and the actual daily evapotranspiration (ET_a) for a wheat crop with a planting date of April 15, and a 135-day growing season. Figure 5.2 shows the trend of the obtained results for ET_a.

With the same consideration that no stress occurred during the growing season, the calculation of crop irrigation requirement was performed by subtracting the effective rain quantity, received during the growing season, from the estimated ET_a data. The comparison between the calculated results of crop irrigation requirement for wheat is shown in Table 5.4.

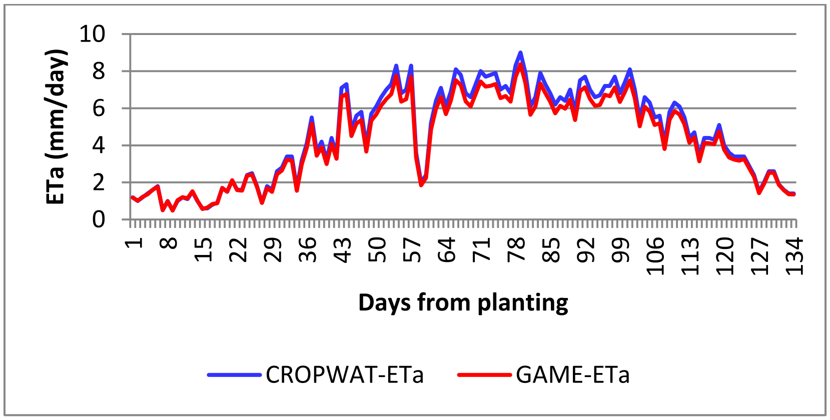


Fig. 5.2. Actual daily evapotranspiration for wheat

Table 5.4. Data of crop water requirement for wheat

Parameter	CropWat	WaterMan
Accumulative ETa (mm)	616.7	576.6
Effective Rain (mm)	43.9	63.9
Irrigation Requirements (mm)	572.8	512.7
Difference in Irrig. Req		**-10.5%**

The same argument used to explain the sources of the difference in the calculation of the corn crop irrigation requirements can be used for the wheat crop.

Irrigation Scheduling

Irrigation scheduling is the answer to two important questions: when to irrigate, and how much to irrigate at each irrigation. CropWat 8 software gives the user different alternatives to decide the calculation criteria for these two questions. According to the chosen option, the software suggests the scheduling program to the user.

For the purpose of the comparison herein, two options for when to irrigate were chosen: irrigate at critical depletion, and irrigate at an interval of 15 days. On the other hand, for the calculation of the application quantity, one option was taken: refill the soil water to field capacity. The results of the irrigation scheduling programs for corn and wheat calculated using the CropWat 8 software and the WaterMan technical module were as follows.

Corn Crop

Irrigate at Critical Depletion

Figure 5.3 and Table 5.5 show the irrigation scheduling, timing, and quantity, for corn crop, based on the critical depletion in deciding when to irrigate.

The results of corn irrigation based on the critical depletion, shows a net irrigation requirement of 650.4 mm and 638.6 mm as calculated by the CropWat 8 software and the WaterMan technical module, respectively. The absolute value of the difference in the calculated quantities was less than 2%. Also, looking to the timing of irrigation, it can be observed that both software applications give very similar recommendations for irrigation, with a maximum difference of four days between the two results.

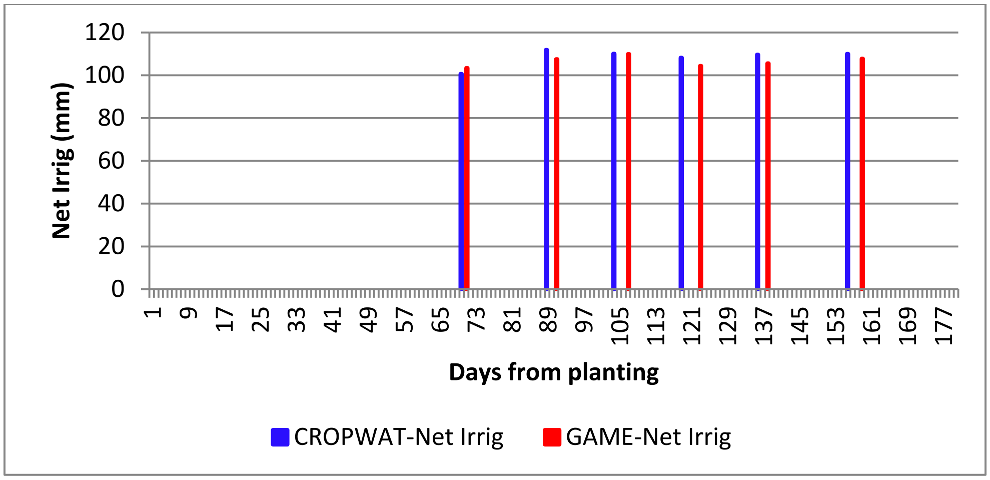


Fig. 5.3. Irrigation scheduling for corn crop based on the critical depletion

Table 5.5. Corn irrigation scheduling based on the critical depletion in deciding when to irrigate

CropWat		WaterMan	
Day after planting	Net Irrigation (mm)	Day after planting	Net Irrigation (mm)
70	100.6	71	103.4
89	111.9	91	107.5
104	110.1	107	109.9
119	108.2	123	104.4
136	109.6	138	105.6
156	110	159	107.8
Total	**650.4**		**638.6**

Irrigate at a Specified Interval

Figure 5.4 and Table 5.6 show the irrigation scheduling, timing and quantity, for corn crop, based a fixed interval of 15 days in deciding when and how much to irrigate.

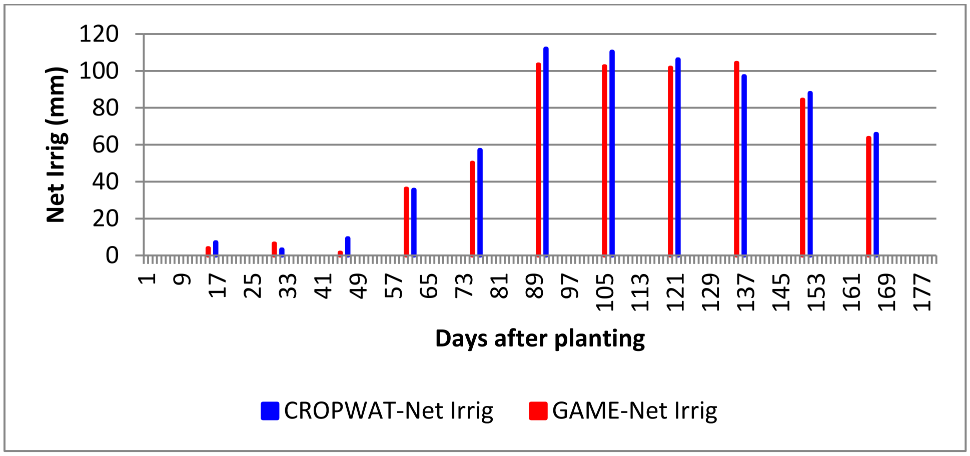


Fig. 5.4. Irrigation scheduling for corn crop, with a 15-day irrigation interval[1]

Table 5.6. Corn irrigation scheduling based on a 15-day interval in deciding when and how much to irrigate

Days after planting	CROPWAT Net Irrigation (mm)	WaterMan Net Irrigation (mm)
15	7.2	3.9
30	3.3	6.5
45	9.3	1.5
60	35.6	36.2
75	57.2	50.3
90	112.1	103.5
105	110.4	102.5
120	106.3	101.8
135	97.2	104.3
150	88.1	84.4
165	65.9	63.7
Total	**692.6**	**658.6**

---

[1] A temporal shift of 2 days was made on the CropWat data for the purpose of better visualization of the results.

The results of corn irrigation at a defined interval of 15 days, shows a net irrigation requirement of 692.6 mm and 658.6 mm, as calculated by CropWat 8 and the WaterMan software, respectively. Thus, the absolute value of the difference in the calculated quantities was less than 5%.

Wheat Crop

Irrigate at Critical Depletion

Figure 5.5 and Table 5.7 show the irrigation scheduling, timing and quantity, for wheat crop, based on the critical depletion in deciding when to irrigate.

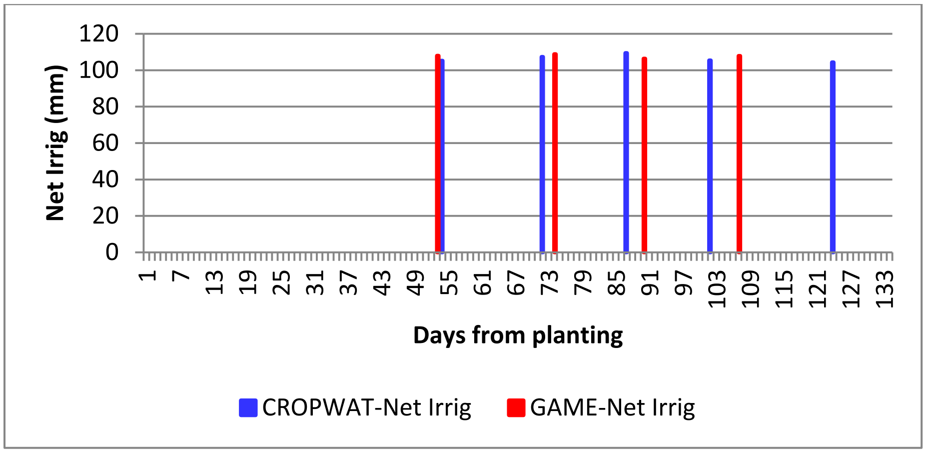


Fig. 5.5. Irrigation scheduling for wheat crop based on the critical depletion

Table 5.7. Wheat irrigation scheduling at critical depletion in deciding when to irrigate

CropWat		WaterMan	
Days after planting	Net Irrigation (mm)	Days after planting	Net Irrigation (mm)
53	105	53	108
72	107	74	109
87	109	90	106
102	105	107	108
124	104		
Total	**532**		**431**

The results of irrigation scheduling for wheat shows greater differences between the two software applications compared to the results obtained for the irrigation scheduling of a corn crop. The number of irrigation calculated by the WaterMan technical module was one irrigation less as compared to the number of irrigations calculated by the CropWat 8 software. The total number of irrigations were 4 and 5 for the WaterMan technical module and the CropWat 8 software, respectively. Comparing the days of irrigation, there is a close similarity between the two applications for the first four irrigations. The fifth irrigation, as called for by the CropWat 8 software, was recommended 10 days before the harvesting of the wheat crop, which is perhaps unreasonable and unlikely to be applied in the field (a fifth irrigation was not recommended by WaterMan).

Accordingly, it can be concluded that both applications gave similar results in terms of their irrigation scheduling calculations. But the quantity of irrigation water at each irrigation varied between the two applications. A total net irrigation requirement of 532 mm and 431 mm was calculated by the CropWat software and the WaterMan technical module, respectively, and the absolute difference in the calculated quantity was around 19%. But after excluding the last irrigation from the CropWat calculations, the results are nearly identical.

Irrigate at a Specified Interval

Figure 5.6 and Table 5.8 show the irrigation scheduling, timing and quantity, for wheat crop, based on a fixed interval of 15 days in deciding when to irrigate.

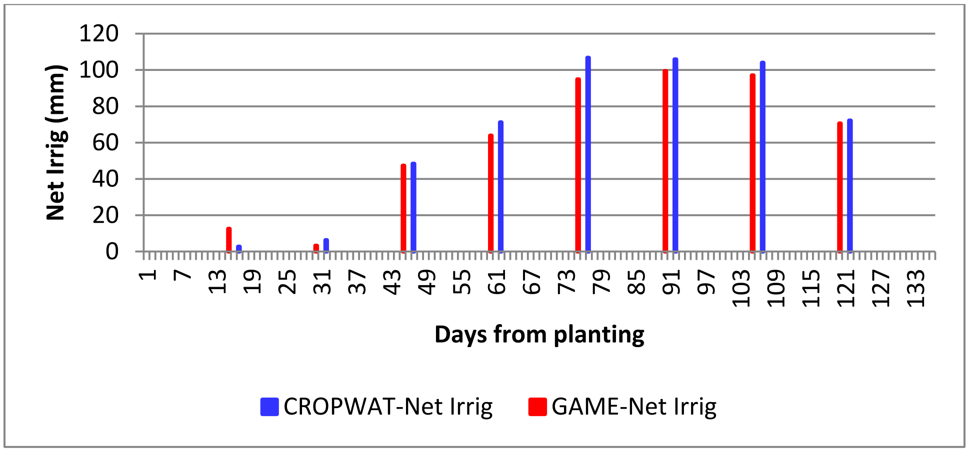


Fig. 5.6. Irrigation scheduling for wheat crop with a 15-day irrigation interval[2]

Table 5.8. Wheat irrigation scheduling based on a 15-day irrigation interval

Days after planting	CROPWAT Net Irrigation (mm)	WaterMan Net Irrigation (mm)
15	2.7	12.5
30	6.3	3.3
45	48.4	47.4
60	71.2	63.9
75	106.8	94.9
90	105.9	99.5
105	104	97.1
120	72.2	70.6
Total	**517.5**	**489.1**

The results of wheat irrigation at a defined interval of 15 days, shows a net irrigation requirement of 517.5 mm and 489.1 mm, as calculated by the CropWat 8 software and the WaterMan technical module, respectively. The absolute difference in the calculated quantity was around 5.5%.

---

[2] A temporal shift of 2 days was made on the CropWat data for the purpose of better visualization of the results.

After this comparison, it can be concluded that the WaterMan technical module shows acceptable results for calculating crop irrigation requirements and irrigation scheduling, as compared to CropWat 8. However, it is noted that in practice it is not reasonable to have irrigation amounts of only 2.7 or 3.3 mm, as shown in Table 5.8. For such small applications, most of the water would be lost as evaporation if the irrigation method is sprinkler or surface.

With the comparison results obtained, in addition to the mass balance calculation preformed previously, it can be said with confidence that the WaterMan technical module calculations for daily soil water balance are valid.

5.2 Game Testing Results

In order to evaluate WaterMan game performance as a training tool for on-farm water management and to test its robustness and its capabilities to generate reasonable and challenging random events. Twenty-two persons, with different irrigation backgrounds, were asked to play the game. The twenty-two players were chosen from three irrigation background categories: seven are practicing farmers, seven have no irrigation background, and eight have a background in irrigation studies. After playing the game, the players were asked to complete a questionnaire about their playing experience and their suggestions for improvements (Appendix).

While playing, some information was collected through observing the player interaction with the game interface such as: the level of soil water at which a player decided to irrigate, and if the player checked the weather forecast or not, in addition to the generated random events throughout the growing season, and the final score he/she obtained. Additional information recorded by the game included: the selected player option ("Quick Play" or "Play"), the selected delivery method; fixed rotation or on-demand delivery, the selected goal (maximize yield, maximize production function, or maximize profit), in addition to the selected crop, climate zone, texture, and irrigation method. The following sections summarize the obtained results through testing the game.

5.2.1 WaterMan Game Robustness

Game robustness reflected in its ability to respond to the different options and actions taken by the players in a satisfactory manner. Various people with different interests and different backgrounds played the WaterMan game under the supervision of the developer. This heterogeneity of players reflected in the different choices of crops, planting dates, soil, irrigation method, delivery method, and climate. Also, the WaterMan game includes different events generated randomly to reflect real farming conditions. These random events add more challenges to the game flexibility in responding to the effect of these events on the game environment.

The WaterMan game responded to all these challenges in an acceptable manner most of the time. But, in certain cases, some problems were observed, and these were subsequently addressed in the program code. Following are some examples:

- A problem was observed when a player tried to pause irrigation of the whole field, attempting to irrigate only certain sections. The game did not respond correctly, so it was later modified to respond to this kind of request.
- One player tried to close the weather forecast window through the "X" button (not the "OK" button) and the game did not respond, so this was also fixed.

- When the irrigation duration was set to more than 24 hours, the delivered amount of water was correctly divided over a period of two consecutive days, but it was noticed that the duration timer remained running the next day for another 24 hour, and not only for the remaining hours of the said duration. This issue was also fixed in the program.
- When the player received a random event corresponding to a canal break and five days of simulation time were needed to repair it, and he/she tried to order another irrigation during the canal repair period, the game inappropriately accepted the request and supplied water. This issue was fixed by adding denial messages ("unable to deliver water") throughout the five-day repair period.
- Another important issue encountered during game testing was that the optimal profit results, which are predicted through the game search algorithm, sometimes became negative and the score was higher than it should be. This problem was fixed by not allowing the optimal profit result to drop below zero during a simulation.

5.2.2 WaterMan Game Performance

Scoring Results

WaterMan is not a competitive game to be evaluated based on how many times it is "won" or how many time the player "loses." Instead, the evaluation of the WaterMan game performance will be based on the score obtained by the players. As mentioned in Chapter 4, the game achieves good performance if it forces the player's score to be between 80 and 90, by challenging the player if his/her score is maintained above 90, and to minimize the challenges if the score is below 80. In order to test the performance of the game, data were collected at different times when the game was played.

Table 5.9 shows data collected by the WaterMan game and through observing practicing farmers playing the game. The game was played nine times by seven practicing farmers. As shown in the table, farmers varied in their choices of play level; quick play or play, the same also regarding the delivery method. But, all farmers choose to maximize the profit as their goal of playing the game, also they were noticed checking the weather forecast very extensively throughout the growing season, specifically before the irrigation decision, also it was notice that the majority of farmers who played the game were risk averse in their decisions about when to irrigate, especially under an on-demand delivery method; they ordered water and tried to irrigate so as to maintain a high soil water content.

The scoring results obtained by the farmers presented in Table 5.9 shows that six out of the nine times the game was played, the score was between 80 and 90, while two times it was above 90. Only once did it reach 70. It can be concluded that the game achieved its goal almost 67% of the time when a practicing farmer played it.

Table 5.9. Results collected by observing practicing farmers while playing WaterMan

Play Level		Goal to maximize			Delivery Method		Checked the Weather?		Score
Quick Play	Play	Yield	Prod. Function	Profit	Fixed Rotation	On-demand	Yes	No	
√				√	√		√		84
	√			√		√	√		90
	√			√	√		√		87
	√			√	√		√		98
	√			√		√	√		95
√				√	√		√		90
√				√	√		√		70
√				√	√		√		90
	√			√		√	√		81

Table 5.10 shows data collected by the WaterMan game and through observing players without an irrigation background when playing the game. The game was played thirteen times by seven players without an irrigation background. As shown in the table, the players varied in their choices for choosing the play level: quick play or play. It was expected that this category of player would chose the "Quick Play" option more often than the "Play" option, but the data shows that the players chose the "Play" option almost twice as often. Regarding the delivery method, the majority of the players under the no-irrigation background category chose fixed rotation. But, all players chose to maximize the profit as their goal for playing the game. Checking the weather forecast was noticed by the players in this category, but not extensively and not by all players.

The scoring results obtained by the players without an irrigation background, as presented in the table, shows that four times out of the thirteen, the score was between 80 and 90, and another four times it was above 90, while five times it was below 80.

Table 5.10. Results collected by observing players without an irrigation background

Play Level		Goal to maximize			Delivery Method		Checked the Weather?		Score
Quick Play	Play	Yield	Prod. Function	Profit	Fixed Rotation	On-demand	Yes	No	
√				√	√		√		89
√				√	√		√		75
√				√	√		√		84
	√			√	√		√		70
	√			√	√			√	70
√				√	√			√	70
	√			√		√	√		70
	√			√	√		√		89
	√			√	√		√		99
	√			√		√	√		92
	√			√		√	√		98
√				√	√			√	90
	√			√	√			√	98

The score of two of the players who played the game twice decreased the second time they played. Those two players mentioned that they wanted to challenge the game by playing more professionally, but the game challenged them more, with more random events and so their score was reduced because the program recognized their higher skill level.

And as for another two players, one played the game three consecutive times, and the other played the game two times, with the resulting score showing an improvement due to the increase of their knowledge about irrigation. One player was careless the first time he played the game, so his score was low. He tried to compensate the second time, but he surrendered to the challenges early in the simulated season and he again obtained a low score.

Table 5.11 shows data collected by the WaterMan game and through observation of players with a scientific (academic) irrigation background while playing the game. The game was played a total of fourteen times by eight players in this category. As shown in the table, players varied in their choices for choosing "Quick Play" or "Play", although the number of times that the "Play" option was chosen was almost double the times the "Quick Play" option was chosen, as expected. Regarding the water delivery method, these players chose almost equally between the fixed rotation and on-demand options. Again, almost all players chose to maximize profit as their goal when playing the game, except once in which the maxim yield option was selected. Checking the weather forecast was noticed by the players of this category, but not extensively and not by all players.

Table 5.11. Results collected by observing players with a scientific irrigation background

Play Level		Goal to maximize			Delivery Method		Checked the Weather?		Score
Quick Play	Play	Yield	Prod. Function	Profit	Fixed Rotation	On-demand	Yes	No	
	√			√		√			70
	√	√				√			100
√				√	√			√	77
√				√	√		√		77
	√			√		√	√		87
	√			√		√	√		70
	√			√	√		√		95
√				√	√			√	70
√				√	√			√	80
	√			√		√	√		70
	√			√	√		√		100
√				√		√		√	80
	√			√	√		√		100
	√			√	√		√		71

The scoring results presented in the table shows a wide range of scores obtained by the players with an irrigation background; three times out of the fourteen, their score was between 80 and 90, while seven times it was below 80 and four times it was above 90. A score of 100 was obtained three times by this category. It was noticed that the stronger the background of the player, the more challenges he or she faced, and the more variable their score.

Random Event Generation

The generated random events each time the game was played were recorded during the tests. Different random events were generated for the different players. Table 5.12 shows all the random events included in the WaterMan game and their descriptive abbreviations (label). Tables 5.13 through 5.15 shows the random events generated each time the game was played and for each player's irrigation background category;

practicing farmers, players without and irrigation background, and players with an irrigation background, respectively.

Table 5.12. Random events and their labels

Random Event	**Label**
You have unexpected rain today.	UXR
The main canal is broken and will be repaired in 5 days. No options available.	CBN
Due to water theft, the flow rate to your farm has been reduced to half. No options available.	WTH
Due to canal capacity limitations, you will receive your water after 3 days. Do you want to use existing well with $400 pumping cost?	CLW
The main canal is broken and will be repaired in 5 days. Do you want to use water from drainage ditches with a salinity of 4 dS/m?	CBS
Due to canal capacity limitations, you will receive half of your water quantity. Would you like to get additional water from an existing well with $200 pumping cost?	HFW
Your irrigators are on strike. Do you want to hire temporary irrigators at an additional cost of $100?	IST
Due to a sudden failure in the irrigation system, you can't irrigate one section. Do you want to rent a system for $200 to irrigate this section?	SFR
You can't irrigate today, due to an electrical outage. Do you want to rent a diesel engine? The cost is $400.	EOR
The pump on your farm is broken. Do you want to buy a new pump? The annual cost is $500.	PBB
Your crop was attacked by a disease and it requires two days to control. You can’t irrigate, you lost your water turn.	CDA
The average air temperature for the next week is 10 °C higher than previously expected, Do you want to spend $100 for an additional irrigation?	HTA
Additional water has become available, so your water share has increased. Do you want to sell the extra water for $100?	AWS
Due to water shortage, your water share has decreased. Are you interest in buying an equivalent amount from your neighbors for $100?	SWB
The weather forecast is currently unavailable.	WFN

Table 5.13. Scores and random events generated by practicing farmers

Score	Random Events														
	UXR	CBN	WTH	CLW	CBS	HFW	IST	SFR	EOR	PBB	CDA	HTA	AWS	SWB	WFN
84	√				√							√			
90					√				√		√				√
87			√												
98			√						√				√		√
95					√		√						√		√
90	√				√						√			√	
70	√										√			√	
90						√							√		
81							√					√	√		

Table 5.14. Scores and random events generated by players without an irrigation background

Score	Random Events														
	UXR	CBN	WTH	CLW	CBS	HFW	IST	SFR	EOR	PBB	CDA	HTA	AWS	SWB	WFN
89	√												√		
75										√		√			
84													√		
70								√				√			
70						√							√		
70							√				√		√	√	
70					√			√			√		√		√
89	√						√					√			
99	√							√					√		√
92		√			√		√	√			√			√	√
98				√	√				√				√		√
90			√				√	√				√			
98	√					√	√								

Table 5.15. Scores and random events generated by players with an irrigation background

Score	Random Events														
	UXR	CBN	WTH	CLW	CBS	HFW	IST	SFR	EOR	PBB	CDA	HTA	AWS	SWB	WFN
70					√								√		
100												√			
77			√		√				√			√			
77														√	
87	√				√					√			√		
70	√				√			√		√				√	
95					√			√						√	
70			√			√	√		√					√	
80	√				√			√			√	√			
70					√				√	√			√		√
100			√			√		√	√					√	
80	√														
100			√					√						√	
71					√						√			√	

As seen in the above tables, each of the possible random events occurred many times during the game testing. There was no pattern observed for the generated random events among the different categories, except the complete absence of some random events for certain categories as compared to others, and the frequent occurrence of some random events for players with an irrigation background. An average of three random events were generated by the model each time practicing farmers played the game, and also from the players without an irrigation background, while the average was four random events per simulation received by each of the players with an irrigation studies background.

5.2.3 Questionnaire Results

In the questionnaire (Appendix), players were asked to answer 11 questions; there were nine questions with multiple choices, and two questions in which they were asked to write about their opinions of the model. The questions and the labels for each question are presented in Table 5.16.

Table 5.16. Questions contained in the questionnaire

Question	**Label**
Does the game reflect real field situations?	Q 1
Does the game cover the very important aspects of on-farm irrigation managements?	Q 2
Are the random events realistic?	Q 3
Are the random events challenging?	Q 4
Are the options given realistic?	Q 5
Are the final score and recommendations reasonable?	Q 6
Do you consider the game well designed with a friendly user interface?	Q 7
Playing the game increases your knowledge about on-farm irrigation management?	Q 8
In your opinion. What are the limitations within the game?	Q 9
What are your suggestions for improvements?	Q 10
What is your overall evaluation of the game as a training tool for on-farm irrigation management?	Q 11

Because the testing sample was not large enough to be statistically significant, no statistical analysis was performed on the data; instead, results were explained through tables and through graphs as shown above.

Results for Questions from One to Eight

Table 5.17 shows the number of players who choose to select each of the questionnaire choices, and it helps explain the results presented in Fig. 5.7.

Figure 5.7 shows the number of players who choose to select each choice given by the questionnaire. As observed in the graph, 14 players out of the 22 who played the game chose “agree” on the first question, while eight chose the “totally agree” option, and none of the responses were “not sure,” “disagree,” or “totally disagree.” The answers for question 2 shows six players totally agree, fifteen were agree, and one player was not sure, while for question 3, six players were totally agree, and sixteen players were agree, and none was not sure, disagree, or totally disagree. Question 4 answers showed that seven player totally agree, while fourteen were agree, and one was “not sure.”

Table 5.17. Player answers to the questions

	Totally Agree	**Agree**	**Disagree**	**Totally Disagree**	**Not Sure**
Q 1	8	14	0	0	0
Q 2	6	15	0	0	1
Q 3	6	16	0	0	0
Q 4	7	14	0	0	1
Q 5	5	16	1	0	0
Q 6	7	15	0	0	0
Q 7	8	13	0	0	1
Q 8	10	10	0	0	2

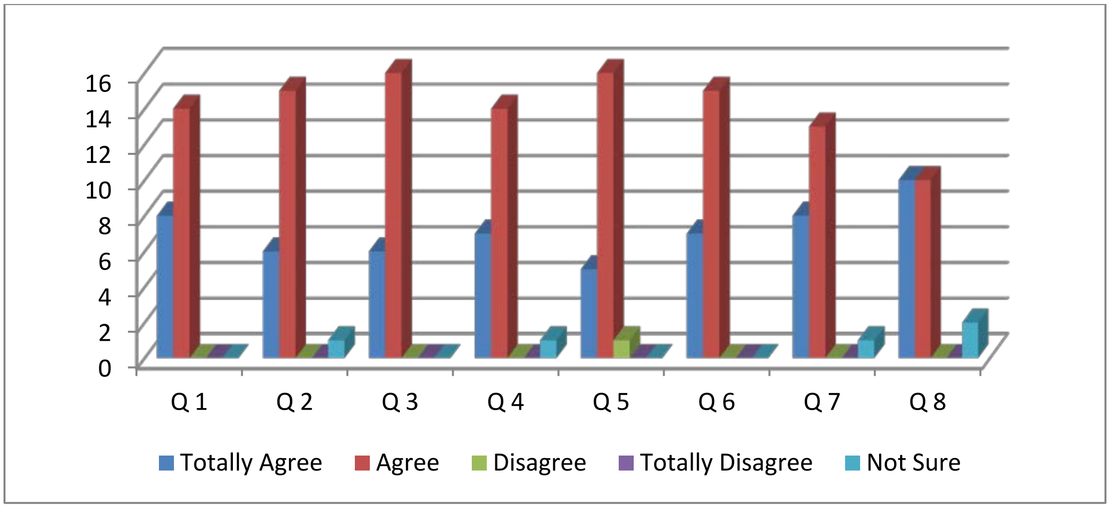


Fig. 5.7. Number of players who answered each of the questions in the questionnaire

For question 5, five players "totally agreed," sixteen players "agreed," and one player "disagreed." For question 6, seven player select the "totally agree" option, while fifteen selected the "agree" option. For question 7, the answers were; eight "totally agreed," thirteen "agreed," and one was "not sure." Ten of the players "totally agreed" with question 8, another ten "agreed," and two said they were "not sure."

Considering the results for each irrigation background category, the following tables and figures show the obtained results. Table 5.18 shows the number of farmers who chose to select each choice for the questions given, and the results are presented in Fig. 5.8.

Table 5.19 shows the number of players without an irrigation background who choose to select each choice for the questions given, and the results are presented in Fig. 5.9.

Table 5.18. Farmers' answers to the given questions

	Totally Agree	Agree	Disagree	Totally Disagree	Not Sure
Q 1	1	6	0	0	0
Q 2	0	6	0	0	1
Q 3	1	6	0	0	0
Q 4	2	4	0	0	1
Q 5	2	4	1	0	0
Q 6	1	6	0	0	0
Q 7	1	5	0	0	1
Q 8	0	5	0	0	2

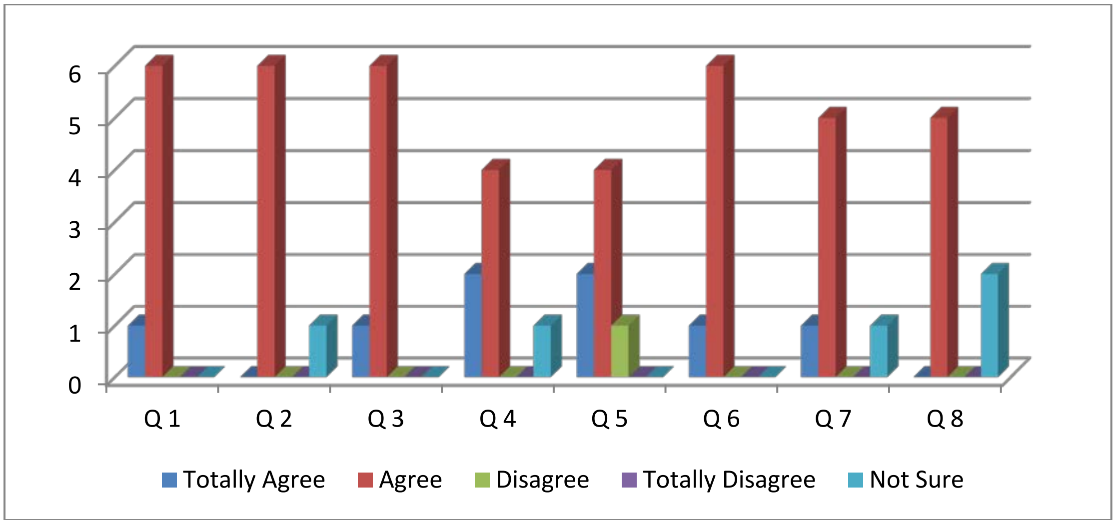


Fig. 5.8. Number of farmers who answered each question in the questionnaire

Table 5.19. Answers to the questions by players without an irrigation background

	Totally Agree	Agree	Disagree	Totally Disagree	Not Sure
Q 1	5	2	0	0	0
Q 2	4	3	0	0	0
Q 3	2	5	0	0	0
Q 4	2	5	0	0	0
Q 5	2	5	0	0	0
Q 6	3	4	0	0	0
Q 7	3	4	0	0	0
Q 8	6	1	0	0	0

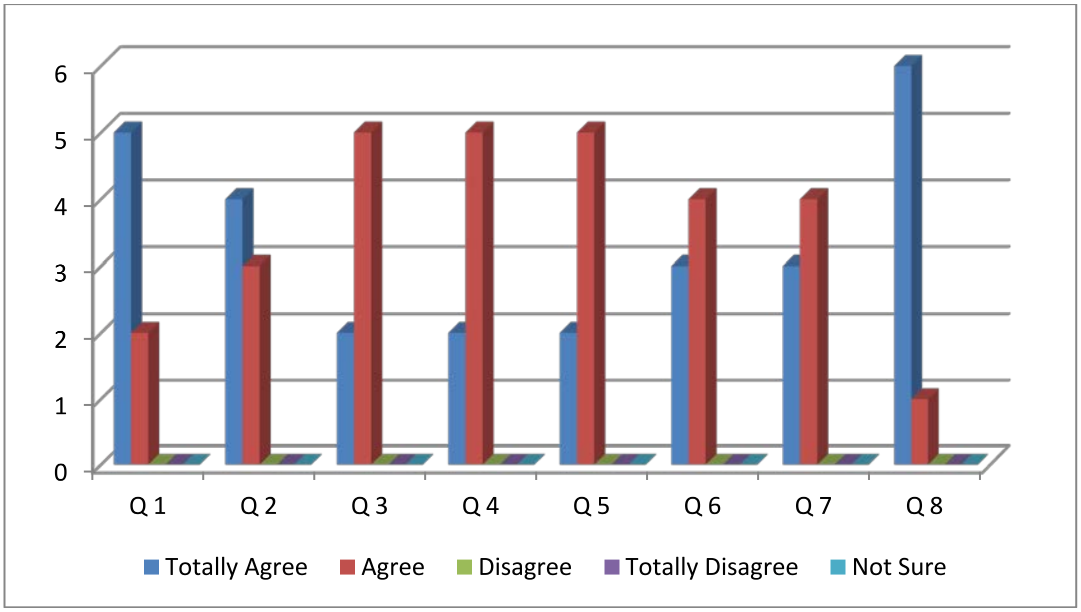


Fig. 5.9. Number of players without an irrigation background who answered each question in the questionnaire

Table 5.20 shows the number of players with an irrigation background who chose to select each choice for the questions given, and explains the results presented in Fig. 5.10.

Table 5.20. Answers to the questions by players with an irrigation background

	Totally Agree	Agree	Disagree	Totally Disagree	Not Sure
Q 1	2	6	0	0	0
Q 2	2	6	0	0	0
Q 3	3	5	0	0	0
Q 4	3	5	0	0	0
Q 5	1	7	0	0	0
Q 6	3	5	0	0	0
Q 7	4	4	0	0	0
Q 8	4	4	0	0	0

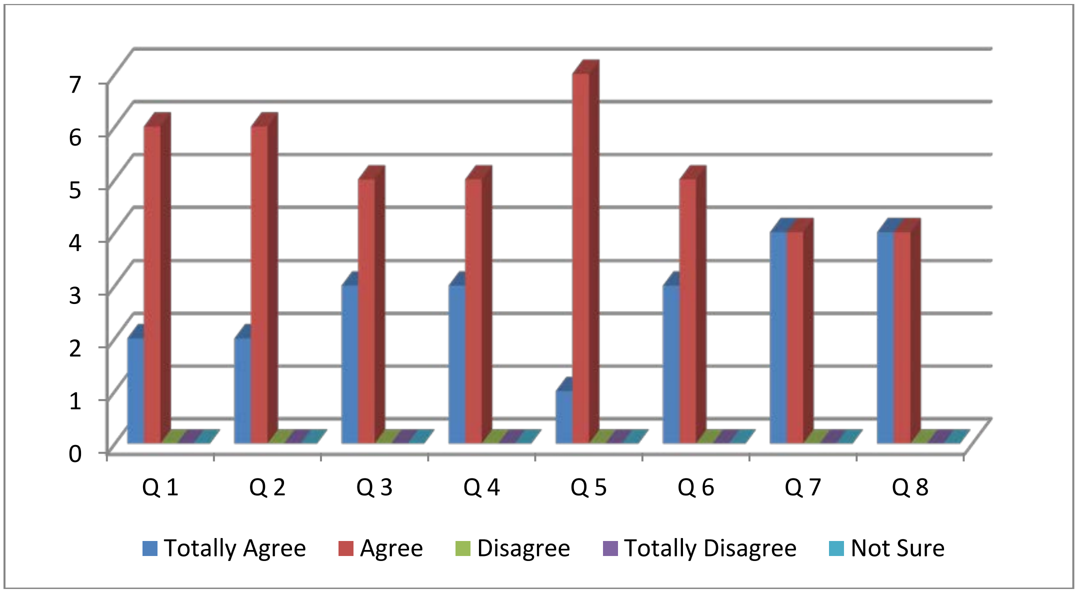


Fig. 5.10. Number of players with irrigation background who answer each question presented in the questionnaire

Results for Question Eleven

The answers for question 11, which was about the overall evaluation of the game as a training tool, are presented in Table 5.21 and Fig. 5.11. The general results show that eleven players considered the game as "excellent," nine considered it a "very good tool," while two players classified it as a "good tool." Taking each category classification shows that one farmer classified it as "excellent," five considered it "very good," and one considered it to be a "good tool." The game was considered an "excellent tool" by all seven players without an irrigation background, while three players with a scientific irrigation background considered the game as an excellent training tool, four considered to be a very good tool, and one player had the opinion that the game is a "good tool." The overall classification of the game by the majority of the players was that it is an "excellent training tool."

Table 5.21. Players' answers about their overall evaluation of the game

	All Players	**Farmers**	**No-Irrigation Background**	**With irrigation Background**
Excellent	11	1	7	3
Very good	9	5	0	4
Good	2	1	0	1
Fair	0	0	0	0
Not sure	0	0	0	0

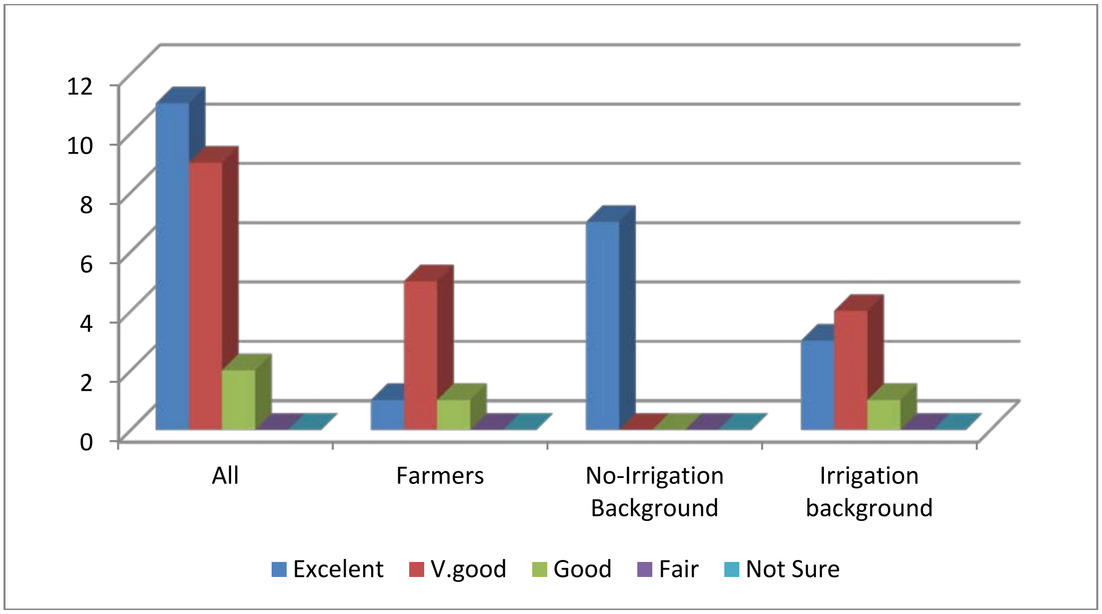


Fig. 5.11. Players' answers about their overall evaluation of the game

<u>Limitations and Suggestions For Improvement</u>

Players were asked for their opinion about the game's limitations, and for their suggestions to improve it in questions 9 and10. The players' unedited answers were presented in three different tables representing the opinion of each player category (Appendix). Following are samples of the limitations and suggestions given by the respondents:

Limitations:

1. “Decision of irrigation had to be made quickly”
2. “The screen is very busy, could have multiple screen to look at for decision”
3. “Should have an excellent strategy for training users on how to use the program, or make it more self explanatory with pop ups or windows”
4. “Needs more random events to be added”

Suggestions:

1. “Another version that simulate all on-farm practices”
2. “Allow for user input data”
3. “Allow the storage of the additional water for future use”
4. “The possibility of buying additional water when needed”
5. “Allow for the management of more than one crop at the same time”
6. “Include suggestion about how early or late crop is planted”

Suggestions already considered:

Some of the suggestions were considered immediately in the game to improve its performance, such as:

1. To show the crop name and the days of the growing season on the main form
2. To have different flow rates for the different irrigation systems
3. To pause the game when the message "No water is available for irrigation" appears

CHAPTER 6

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

6.1 Summary

This dissertation describes the development of a software application in the form of a game that simulates different technical and operational aspects of on-farm water management for use in training events for farmers, irrigators, irrigation extension specialists, and students, on how to manage on-farm water resources for more economical farming and better productivity. Irrigators face difficult operational decisions when deciding when and how much to irrigate. This game provides an interactive framework to enable trainees to develop different management scenarios, and to test alternative approaches without worrying about the consequences of their decisions in a real farming environment. “WaterMan” is the given name to this game, reflecting its nature as a simulation approach for the different aspects of "On-Farm Water Management," in which various related water resources managerial issues are integrated. This game has the potential to be a universal tool, for application in almost any irrigated agricultural area in the world, easily spanning economic and cultural differences.

The Visual Basic .NET programming language in Microsoft Visual Studio 2008 was used in for the development of the WaterMan game. This modern computer programming environment offers great capabilities in designing a user-friendly interface to visualize the important aspects of on-farm water management. The interface was designed in a way that reflects a real farming environment, in which a dynamic graphic showing the daily cropping conditions is continuously presented in the simulation window. The graphic includes information about daily soil water content, daily expected relative yield, daily root zone development, plant growth conditions; animation of crop growth status, whether it is performing well or shows symptoms of stress, whether the crop is still alive or has died, and so on, are included in the animated sketch. In addition, a five-day (probable) weather forecast, the available water for the remaining of the season, and irrigation water quantity based on the farm flow rate and the delivery interval, are made available to the player for making irrigation management decisions. The interface was designed to integrate all information that is usually available to irrigators and to be easily accessed by the player.

Timers control the simulation to reflect actual conditions and to foster the active engagement of the players with the game environment. Three seconds were set to represent a day in the growing season of the crop, with a capability to speed it up or to reduce the speed at the convenience of the player. Accordingly, the duration of the game play is governed by the player choices of the crop (different growing seasons), and the chosen game speed.

Two options of game play were developed to accommodate different trainee requirements and interests: “quick play” and “play.” The “quick play” option was added to the game in order to offer an introductory way to explore the game for players without a strong background in farming (i.e. novices). In this option the player moves directly to simulate irrigation water management with a predetermined set of input data. If the “play” option is selected, a new window for data input appears, and the player is asked to select from various options. The different options for planted crops, climatic conditions, water delivery methods, soil texture classes, and on-farm irrigation methods included in the WaterMan game are supported by a specially constructed database and heuristic capabilities in order to accommodate the different players, needs as best as possible.

The software database contains crop phenological data for 25 different crop types, three soil texture classes, five on-farm irrigation methods, and seven climate zones. The database and the heuristics capabilities were added to the program to provide the player with a realistic game environment without the need for a massive amount of "calibration data." The diversity of options also reflects the flexibility of the WaterMan game in introducing different scenarios to enable the player to test different alternatives while playing the game repeatedly. This flexibility of the WaterMan game is achieved through the development of special programming modules encoded in the software to deal with the database and to respond effectively to the player's behavior. This kind of flexibility and heuristic capability is seldom (if ever) found in other irrigation management training games.

A very detailed technical module was developed within the WaterMan game to simulate the daily soil water and salt balances. Based on procedures presented in the FAO Irrigation and Drainage papers 56 and 29, all variables affecting soil water and salt balances were considered in the calculations. An Excel spreadsheet was developed outside of the game software to track and evaluate the results obtained from the interaction between the different equations used in the calculations. Accordingly, a modification was made on Eq. (85), page 170, in FAO Irrigation and Drainage Paper 56 for the calculation of the daily soil water balance (the soil water depletion at the end of the day), to include the effect of the root growth at the end of the day.

Also, an extensive search was done to include an accurate calculation of the capillary rise from the ground water table, which is a one component of the above-mentioned equation. In the calculation of the salt balance, it was determined that the use of Eq. (2), page 25, in FAO Irrigation and Drainage paper 29 is inappropriate, as was intended for the calculation of the leaching fraction (LF). Instead, the deep-percolated quantity was considered as a percentage from the leaching requirement and its salinity calculated proportional to that percentage.

Throughout the development of the technical part of the program, the calculations were evaluated step-by-step to demonstrate their effectiveness. The payoff of this continuous evaluation was shown on the near perfect results obtained when a mass balance calculation was done to validate the calculation procedure for the daily soil water and salt content over a period of time during an irrigation season, and also when making a comparison between the game-generated results of irrigation water requirements and irrigation scheduling calculations with those generated by the FAO CropWat 8 software. The comparison results between the WaterMan technical module and the CropWat applications shows that the absolute value of the difference in the calculated quantities of crop water requirements for two crops under investigation; corn and wheat, was less than 11%, for both crops. The respective recommendations for irrigation scheduling shows a very similar set of irrigations with a maximum absolute difference of four days over the duration of a growing season, and a difference of less than 2% in the recommended irrigation amount for corn and wheat crops with the critical depletion option for deciding when to irrigate. Under a specified interval option for when to irrigate, the results show an absolute difference of less than 5.5% for the two crops included in the analysis.

Different heuristic approaches were employed during the development of the WaterMan game in efforts to achieve higher levels of realism in playing the game. The areas of implementation of the heuristic capabilities within the game were: weather data generation, optimization of irrigation scheduling, and the generation of random events.

For weather data generation, seven major climatic zones are identified in the world namely: Tropical, Temperate, Mountains, Continental, Mediterranean, Semi-Arid, and Arid. A one-year daily record for precipitation, minimum and maximum air temperature obtained from a global daily climate data records for 1950 to 1999 time period over each of the specified climatic zones were included in the WaterMan software database. Two different code modules were developed within the software to generate a new set of daily maximum and minimum air temperature and a new set of rainy days and rain quantity each time a new game is instantiated. Another module was developed to generate a five-days weather forecast data including the expected maximum and minimum air temperatures, in addition to the possibility of having rainy days during the coming five days, with the respective probabilities of occurrence is also encoded in the software.

As a training tool, this game is expected to be played by people with different irrigation management skills. In order to make the game more interesting, the program is expected to anticipate the player skill level, and to reply with random events appropriate to the anticipated level. Generating random events based on the automatic evaluation of the player skill level,was the main goal of using the artificial intelligence approaches within the WaterMan game, and it is a significant new technology introduced by this game, as compared to the other games previously developed as a training tools for on-farm irrigation management.

The author studied many approaches within the artificial intelligence (AI) field of research, searching for those that assist in achieving the anticipated goal. Ultimately, two artificial intelligence approaches were found to be appropriate and were used in the program to achieve this capability: (1) a pattern recognition approach; and, (2) reinforcement learning based on Markov Decision Processes; specifically, the Q-learning method. The pattern recognition part is responsible for the determination of the player type, and passes this information to the decision-making system. The decision-making system is responsible for choosing and executing an appropriate action at the particular state based on the Q-values calculation. It is believed that this is the first documented case of combining these two artificial intelligence approaches in a game.

The pattern recognition approach classifies the player into four categories; risk-averse, risk neutral, risk-taker, and strategic. These classifications were based on player interaction with the game environment and the environment state at which the irrigation decision is taken, in addition to other action such as checking the weather forecast, and the input data options the player made. The reinforcement learning approach describes a system that learns from its interaction with the environment governed by a policy and a reward function. Q-learning is a method within the reinforcement approach that follows the selection of an action based on its calculated maximum Q-Value to be executed. After that, the system observes the effect of the executed action on the environment and evaluate its decision based on a certain criteria called the reward function. The system will be rewarded if it executed an action that moves the game environment state closer to goal set for the system to achieve, otherwise it will consider the action as inappropriate. From its interaction with the environment the system learns to execute the most appropriate action, and this was demonstrated through the game testing procedures, as described in the previous chapter.

All the identified applications that apply the Q-learning method were built with the agent (system) as a cooperative agent, in which the system will be better awarded if it helps the user to achieve his/her goal, or as a competitive agent, at which the system will be rewarded if it is able to challenge the player and attempt (to some degree) to cause him/her to have a lower score. The challenge in the WaterMan game was that the agent should be built as a competitive agent when dealing with a skilled (strategic), or a risk-averse player, and should be cooperative, or less challenging, when dealing with (neutral, or risk-taker) player types. To overcome this challenge, the player daily score (known only by the system and not by the player) was set as an indicator for the reward function. The range of player scores was set between 70% and 100%, and the goal for the Q-learning method system to achieve is to maintain a score between 80 and 90%. An equation was given to the system pointing that the highest reward achieved, that associated with the maximum Q-value is when the daily score is equal to 85%.

The artificial intelligence method searches among fifteen different events encoded in the program to select the appropriate one to be executed at the particular state of the environment. The effectiveness of the game artificial intelligence approaches was tested through playing the game by twenty-two persons with different irrigation backgrounds; seven are practicing farmers, seven have no irrigation background, and eight have a scientific irrigation background. The generated random events were found appropriate for most of the cases. This effectiveness indicated by the range of scores obtained by the different players, at which most of the scores had fallen between 80 and 90%. It was also noticed that the higher the skills level of the player, the more challenges he/she faced, and the more variable score obtained.

In order to provide feedback to the trainee(s) an economic analysis of the cropping season is presented based on the crop yield, production cost, and on-farm water use indicators. At the beginning of the simulation, the player is asked to select a goal he/she wants to achieve from playing the game. Three goals were given to the player to choose from: maximize yield, maximize production function, and maximize profit. The default goal was set to maximize profit, in-case the player didn't choose a goal. How close the player was in achieving the selected goal is the criteria on which he/she is evaluated and given the score accordingly. To give a reasonable evaluation and score, the optimal results that can be achieved under the same circumstances is estimated and used as a reference for the evaluation.

As indicated, all goals are optimization objective functions. To search for the optimal results a one-dimensional (single decision variable) search algorithm, namely the

Golden Section method, was used to search for the optimal solution for a decision variable to achieve an objective function. The decision variable that was set for the algorithm to search for is the percentage of net irrigation water to be apply each irrigation in order to bring the soil water content in the root zone to field capacity. The general objective function the algorithm is searching for is to minimize yield reduction. The "smart" selection of the decision variable and the objective function, enable the use of a one-dimensional search algorithm that minimizes the cost (time required) to search for an optimum result. The search cost by the game's algorithm was measured at approximately three seconds, indicating a quick search that is needed to avoid delays in the start of the game.

The final score given to the player is based on his/her achievement with reference to an optimal target set by the game software. A one-dimensional search algorithm based on the golden section method is used in the software to search for optimal results at the beginning of the game. A comparison is processing between the results obtained by the player based on the selected goal, with that considered as the optimal management practice (the reference results by the search algorithm) enables the scoring module to evaluate the decisions made by the player and gives him or her a fair and reasonable score.

6.2 Conclusions

The principal goal of this work was to develop a modern software-based simulation tool for on-farm irrigation water management, which can be used in training events for farmers and irrigators. This goal was accomplished by developing a model that uses a custom-designed heuristic algorithm, custom artificial intelligence coding, and various technical and interface features to make the game both realistic and interesting to play. The model randomizes various events and weather data in ways that were previously undocumented in the technical literature. Sophisticated soil and salt mass balance algorithms correctly account for all of the water entering, exiting, and stored in the crop root zone. These algorithms have significant new features not documented in other irrigation or water management software.

To avoid developing a completely predictable game, which is a characteristic of all the irrigation management games that was evaluated in the process of doing this research, the WaterMan game was developed with different heuristic approaches to produce a new game environment and some challenges each time the game is instantiated. New random events are generated each time the game is played, in terms of event type and timing, make the game more interesting and more realistic in reflecting real situations that irrigators could actually encounter in irrigated farming practice. Two artificial approaches were successfully combined to generate random events. These two approaches are known in the artificial intelligence field of study, but it is the first documented time that they have been combined in a game. The two approaches were adapted in an innovative way to reflect the nature of the interaction between the human player and the game system, and to avoid the complicity arise by considering the game as a multi-agent game. Specific reward function was developed to fulfill the requirement of the WaterMan game as an educational tool that is neither competitive nor cooperation.

After the end of the simulated season, the optimal solution obtained by the search algorithm which reflects the best yield obtained with the lowest amount of irrigation water consumed is used as a reference to generate the score obtained by the player. The ability of the search algorithm to converge on the right global minimum, gives more realistic scoring results to the player. The search algorithm was observed to converge to the right of the global minimum in 78% of the time. That proves the ability of the using of a one-dimensional search algorithm with a single objective function and a single decision variable to solve for the crop production function.

Various people with different interests and different backgrounds were asked to play the WaterMan game. Different choices of crops, planting dates, soils, irrigation methods, delivery methods, and climates were selected. In addition, the different events generated randomly by the game added more challenges to the game flexibility in responding to the effect of these events on the game environment. The WaterMan game manifests a very high robustness in responding to all these challenges.

Based on the majority of the players who tested the game, the WaterMan game was considered as an effective and innovative tool for teaching people of different irrigation background about on-farm water management, being technically correct, interesting, and challenging.

6.3 Recommendations

6.3.1 Suggestions for Improvement

Although the WaterMan game is simulating many aspects of on-farm water management through a user-friendly interface, some other refinements can be suggested for future development of the software, such as:

1. The addition of another level in the game for more professional users by allowing the management of more than one field with different cropping patterns and different irrigation methods.
2. The addition of more options for making strategic management decisions, such as:
   a) The investment in an upgrade from the current irrigation method to a more efficient one.
   b) The investment in an on-farm pond to store additional water, especially under a fixed rotation distribution water delivery option.
3. Include even more crop types when there is a potential for that in the future.
4. In the recommendation part, give suggestion regarding how appropriate the selected planting date and the irrigation method are for the chosen crop.
5. Simulate the effect of other on-farm agricultural practices on irrigation management decisions.
6. Expand the game to include another part about the water distribution system (canals) operation, with a linkage between the on-farm water management decisions and canal operating decisions.

7. The re-development of the game using other spoken languages, in order to achieve a more universal usage of the game as a training tool.

6.3.2 Recommendations for Future Research

During the development of the WaterMan game, and in looking for the development of a more realistic software application, the following areas of research were found to be important and could add more value to game if considered:

1. Other artificial intelligence approaches could be tested through considering the game as a multi-agent stochastic game.
2. Knowing the driving forces behind farmers' decisions and the prediction of farmers' behavior would enable a better classification of the player types and improve the output from the artificial intelligence approaches already used (and others that could be used in the future) in the game.
3. Porting the game to other platforms, such as iPad-type devices, with an even simpler interface and more artistic graphics would make the game more readily available and perhaps more effective.

REFERENCES

Adam, J., and Lettenmaier, D. 2003. "Adjustment of global gridded precipitation for systematic bias." *J. Geophys. Res*., 108, 1–14.

Allen, R., Pereira, L., Raes, D., and Smith, M. 1998. "Crop evapotranspiration - Guidelines for computing crop water requirements." *Irrigation and Drainage Paper 56*. Food and agriculture Organization of the United Nation, Rome, 300 pages.

Ayers, R., and Westcot, D. 1985. "Water quality for agriculture." *Irrigation and Drainage Paper 29*, Rev. 1. Food and agriculture Organization of the United Nation, Rome, 203 pages.

Barreteau, O., and Daré, W. 2007. "Role-playing games in a verity of cultures: Experiences from the ComMod group." *Simulation and Gaming,* 38 (2).

Barrios-Aranbar, D., and Gonçalves, L. 2009. "Influence value Q-Learning: A reinforcement learning algorithm for multi-agent systems." *Theory and novel applications of machine learning,* M. J. Er and Y. Zhou, eds., ISBN 978-3-902613-55-4, I-Tech, Vienna, Austria.

Burton, M. 1989. "Experiences with the Irrigation Management Game." *Irrig. and Drainage Systems*, 3, 217-228.

Burton, M. 1994. "The Irrigation Management game: A role playing exercise for training in irrigation management." *Irrig. and Drainage Systems*, 7, 305-318.

Carter. R. 1989. "The Juba Sugar Estate Game." *Irrig. and Drainage Systems*, 3, 229-240.

Chapman, G. 1989. "Developing real imaginary countries." *Irrig. and Drainage Systems,* 3, 309-313.

Clarke, D. 2004. "Simulation and role playing for irrigation water management." *Tools for Public Participation, Conflict Resolution and Decision-Making in Water Resources Management,* Seminar held on 14 October. ICID British Section, London.

De Nys, E., Le Gal, p., Raes, D., and Ana, E. 2008. "WaDI (water delivery for irrigation): A simulation tool to address strategic interaction of water demand and supply in irrigation schemes." *Agricultural Water Manage.*, 95, 224-232.

Doorenbos, J. and Kassam, A. 1979. "Yield response to water." *Irrig. and Drainage Paper 33*. Food and agriculture Organization of the United Nation, Rome, 193 pages.

Eching, S., Hopmans, J., and Wendroth, O. 1994. "Unsaturated hydraulic conductivity from transient multistep outflow and soil water pressure data." *Soil Sci. Soc. Am. J*., 58, 687-695.

Feitelson E., Fischhendler, I., and Kay, P. 2007. "Role of a central administrator in managing water resources: The case of Israeli water commissioner." *Water Resour. Res.*, 43, W11415.

Fern, A., Natarajan, S., Judah, K., and Tadepalli, P. 2007. "A decision-theoretic model of assistance." *Proceeding of 20th International Joint Conference on Artificial Intelligence*, pp. 1879-1884.

García-Barrios, L., Speelman, E., and Pimm, M. 2008. "An educational simulation tool for negotiating sustainable natural resource management strategies among stakeholders with conflict interests." *Ecolog. Modelling*, 210, 115-126.

García-Vila, M., Lorite, I., Soriano, M., and Ferares E. 2008. "Management trends and responses to water scarcity in an irrigation scheme of Southern Spain." *Agricultural Water Management,* doi:10.1016/j.agwat.2007.11.009. In Press.

Hargreaves, G. and Merkley, G. 1998. *Irrigation fundamentals*. Water Resources Publications, LLC, Highlands Ranch, Colorado, USA.

Hu, J., and Wellman, M. 2003. "Nash Q-Learning for General-Sum stochastic games." *J. Machine Learning Res.*, 4, 1039-1069.

Hui, B., and Boutilier, C. 2006. "Who's asking for help? A Bayesian approach to intelligent assistance." *Proceedings of the 11th international conference on intelligent user interfaces(IUI'06)*, 186-193.

Johnston, W., Robertson, J., and Clemmens, A. 1991. "Management, operation, and maintenance of irrigation and drainage systems." *ASCE manuals and reports in engineering practice*, No. 57. The Society, New York, N.Y.

Jonoski, A., and Harvey, H. 2004. "Aquavoice: prototyping participatory decision making." *Tools for Public Participation, Conflict Resolution and Decision-Making in Water Resources Management*, Seminar held on 14 October. ICID British Section, London.

Kaukoranta, T., Smed, J., and Hakonen, H. 2003. "Role of pattern recognition in computer games." *Proceedings of the 2nd IEEE International Conference on Cognitive Informatics (ICCI'03),* IEEE CS Press, London, UK, August.

Kos, Z., and Prenosilova, E. 1999. "Simulation and gaming in water management in Czech Republic." *Simulation & Gaming*, 30 (41), 476-481.

Lankford. B., Sokile, Ch. and Yawson, D. 2004. "The river basin game: A role-playing board game for initiating discussions on visions and strategies of water allocation." *Tools for Public Participation, Conflict Resolution and Decision-Making in Water Resources Management*, Seminar held on 14 October. ICID British Section, London.

Littman, M. 1994. "Markov games as a framework for multi-agent reinforcement learning." *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 157-163.

Maurer, E., Adam, J., and Wood, A. 2009. "Climate Model based consensus on the hydrologic impacts of climate change to the Rio Lempa basin of Central America." *Hydrology and Earth System Sciences*, 13, 183-194, (connect to online version).

Merkley, G. 2007. Irrigation engineering fundamentals, *Lecture notes*. BIE, USU.

Molle, F., and Berkoff, J. 2006. "Cities versus agriculture: Revisiting intersectoral water transfers, potential gains and conflicts." *Comprehensive Assessment Research Report 10*. Colombia, Sri Lanka: Comprehensive Assessment Secretariat.

Parrish, J. 1982. "IRRIGAME: An irrigation management simulation game." *M.Sc. Thesis*. Biological and Irrigation Department. Utah State University, Logan.

Prajamwong, S. 1994. "Command area decision support system for irrigation projects." PhD dissertation. Biological and Irrigation Department. Utah State University, Logan.

Popova, Z., and Pereira, L. 2008. "Irrigation scheduling for furrow-irrigated maize under climate uncertainties in the Thrace plain, Bulgaria." *Biosystems Engineering*, 99, 587 – 597.

Raes, D. 2009. "UPFLOW: Water movement in a soil profile from a shallow water table to the topsoil (capillary rise)." *Reference Manual*. Version 3.2.

Ritchie, J., and Basso, B. 2008. "Water use efficiency is not constant when crop water supply is adequate or fixed: The role of agronomic management." *Europ. J. Agronomy* 28, 273-281.

Roling, N., Jiggins, J., and Watson, D. 2004. "Tools and participation in water management in the context of new approaches to natural resource management including social learning." *Tools for Public Participation, Conflict Resolution and Decision-Making in Water Resources Management,* Seminar held on 14 October. ICID British Section, London.

Russell, S., and Norvig, P. 2003. *Artificial intelligence, A modern approach*. Prentice Hall, Upper Saddle River, New Jersey, second edition.

Schultz, B., Thatte, C., and Labhsetwar, V. 2005. "Irrigation and drainage. Main contributors to global food production." *Irrig. and Drainage*, 54, 263-278.

Sigaud, O., and Buffet, O. 2010. *Markov decision processes in artificial intelligence, MDPs, beyond MDPs and application*. John Wiley & Sons, Inc., Hoboken, New Jersey.

Skogerboe, G., and Merkley, G. 1996. *Irrigation maintenance and operations learning process*. Water Resources Publications, LLC. Colorado, USA.

Smith, L. 1989. "The Wye college irrigation game, 'Stop the breach'." *Irrig. and Drainage Systems*, 3, 255-246.

Smith, M. 1988. Manual for CROPWAT, version 5.2. FAO, Rome.

Stensrud, B., and Gonzalez, A. 2008. "Discovery of high-level behavior from observation of human performance in strategic game." *CYBERNETICS*, 38(3), 855-874.

Sutton, R., and Barto, A. 1998. *Reinforcement learning: An introduction*. MIT Press, Cambridge, Massachusetts.

Tanji, K., and Kielen, N. 2002. "Agricultural drainage water management in arid and semi-arid area." *Irrig. and Drainage Paper 61*. Food and agriculture Organization of the United Nation, Rome.

Van Paassen, Roetter, A, Van Keulen, H., and Hoanh, C. 2007. "Can computer models stimulate learning about sustainable land use? Experience with LUPAS in the humid (sub-) tropics of Asia." *Agric. Systems*, 94, 874-887.

APPENDIX

Institutional Review Board

USU Assurance: FWA#00003308

UtahState University
OFFICE OF RESEARCH AND GRADUATE STUDIES

Association for the Accreditation of Human Research Protection Programs, Inc.
Full Accreditation

Request for Determination of Non-human Subjects Research

Approved

FROM: Richard D. Gordin, Acting IRB Chair
True M. Rubal, IRB Administrator

Richard D. Gordin
True M. Rubal

To: Mohammed Shaban

Date: November 09, 2011

Protocol #: 4148

Title: Computer- Based Game As A Training Tool For On-Farm Water Management Using Heuristic Simulation Software

Based on the information provided to USU's IRB, it has been determined that this project does not qualify as human subject research as defined in 45 CFR 46.102(d) and (f) and is not subject to oversight by USU's IRB.

Evaluate the effectiveness of the WaterMan game as a training tool for on-farm irrigation management

Questionnaire

Age:

Occupation:

Education level:

1. Does the game reflect real field situations?

Totally Agree Agree Disagree Totally Disagree Not Sure

2. Does the game cover the very important aspects of on-farm irrigation managements?

Totally Agree Agree Disagree Totally Disagree Not Sure

3. Are the random events realistic?

Totally Agree Agree Disagree Totally Disagree Not Sure

4. Are the random events challenging?

Totally Agree Agree Disagree Totally Disagree Not Sure

5. Are the options given realistic?

Totally Agree Agree Disagree Totally Disagree Not Sure

6. Are the final score and recommendations reasonable?

Totally Agree Agree Disagree Totally Disagree Not Sure

7. Do you consider the game well designed with a friendly user interface?

Totally Agree Agree Disagree Totally Disagree Not Sure

8. Playing the game increases your knowledge about on-farm irrigation management?

Totally Agree Agree Disagree Totally Disagree Not Sure

9. In your opinion: What are the limitations within the game?

____

____

____

____

10. Do you have any suggestion/s for improvements

____

____

____

____

11. What is your overall evaluation of the game as a training tool for on-farm irrigation management?

Excellent Very Good Good Fair Not Sure

Table A.1. Farmer opinions about game limitations, and their suggestions for improvement

Limitations	Suggestions
• Decisions had to be made quickly • Not many, It is good to see the crop grow and at time suffers to make your decision • No water reservoir to store water • Uncontrollable happenings with what happens to water • The time you can water • It still a game, in the end I know it is not quite real • I believe there are things like crop varieties and management outside of water, even though water is still the most important • Need more options to manage rather than put irrigation -Speed is an option in your decision. • Would like to see the crop changing visually according to the management of the water • The screen is very busy - could have multiple screens to look at for decision • Need the ability to change parameters	• I had a lot of questions, so you should have an excellent strategy for training users on how to use the program and understand it. or make it more self explanatory with pop ups or windows • Adding more events • I would like to see a running balance of cost or expenses • No, everything is good • No, everything is very realistic, and similar to every day happenings when dealing with weather and water availability • Change from metric to standard • Maintenance of canal or ditches • How early or late crop planted • Neighbor could take water??? • Good job • A few more options and labels in both English and metric units

Table A.2. Farmer opinions about game limitations, and their suggestions for improvement

Limitations	Suggestions
• None • None • Should be provided with an option to store additional water for future use, if at present do not need to irrigate • Print results • Save results • More option messages along with the random events to give more flexibility in seeing the consequences of my decision. • Does the maintenance timing included • An option to buy water when it become deficit in the soil	• I would like to see a second version that simulates all on-farm practices • Help the player focus on the key criteria of the game (soil moisture, irrigation event) • Please provide the weather forecast all the time. A more efficient forecast would save the crop and the profit. • Allow the user to set up the game with his input data • By showing the crop, window showing its crop water demand quantity • By choosing the flow rate based on the selected irrigation process, process might not be adequate to 400 m3 • Include different cropping patterns

Table A.3. The opinion of the players with an irrigation background about game limitations, and their suggestions for improvement

Limitations	Suggestions
• Irrigation till porosity should be accounted for, with some penalty • Water logging conditions, pest infestation, w.r.t GDD's should be represented • Farmer need to have a certain level of education • Cannot control different fields/crops at the same time • None • None • Not seeing crop stress while it is happening • Random events challenge could be improve • The option of having more than one crop in the field 4 sections • Crop rotation might be interesting to include • The Pause/Stop of irrigation needs some insight, to know how much is sufficient. • Add help interface • Add units in the plots in the interface	• Should show were we went wrong, and • What could have help improve our profit • Allow more time to control during irrigation, maybe more time to control separate fields • More challenging random events • Maybe showing the calendar, so I know for example the month (to expect the weather conditions) • Pause game when we have "No water available for irrigation" message • I am interested to see its application in reality and comparison with real data • The options needs to be explained more with random events, ex: You got extra water, would you rather sell it or store it? your pump broke, would you like to fix it for $400 or skip this irrigation. • No • Add limitations and problems of system failure • Add a conservation behavior • Add draught conditions • No • A short video you-tube with indication how to use can be very useful.

CURRICULUM VITAE

Mohammed Zuhdi Shaban
Utah State University, Logan, UT 84341

Email: m.shaban@aggiemail.usu.edu

EDUCATION

Candidate for the degree of Doctor of Philosophy. Department of Civil and Environmental Engineering - USU
Dissertation: "On-farm Water Management Game with Heuristic Capabilities"

M.Sc. Soil & Irrigation Faculty of Agriculture - University of Jordan, 1994
Thesis: "Effect of Rates and Methods of Iron and Zinc Chelates Application on Production and Fruit Quality of Tow Sweet Orange Cultivars: Washington Navel and Shamouti Grown in the Jordan Valley"

B.Sc. Agricultural Engineering/ Soil & Irrigation. Faculty of Agriculture - University of Jordan, 1988

RESEARCH INTEREST

On-Farm Irrigation Scheduling
Irrigation Systems Design
Farmers Behavior
Crop Evapotranspiration and Water Requirement
Irrigation Systems Evaluation and Management

AWARDS AND HONORS

- Second place, Oral Presentations Award. USU's 2008 Graduate Student Symposium. Logan, Utah, April 2008
- First place, Poster Presentation Award. USU's Spring Runoff Conference. Logan, Utah, April 2010
- Travel Award, USCID 2010 Fort Collins Conference. Fort Collins, Colorado, Sep 2010

TEACHING AND RESEARCH EXPERIENCE

Graduate Student/Research Assistant, Aug 2007- Present
Utah Water Research Laboratory, Utah State University

- Simulating daily soil water and salt balance
- Simulating daily various plant condition and productivity
- Simulating daily weather data for different climatic zones
- Modeling heuristically capabilities in a game environment

Instructor, April, 2010.
Iraqi Agriculture Extension Revitalization Program (IAERP). Erbil/ North Iraq

- Lecturing in on-farm water management and irrigation scheduling
- Prepare tests, grading and evaluation
- Translate for Arabic speakers and led discussion in field trips in Erbil

Instructor, Utah State University, Oct 2009 & Nov 2009.
Iraqi Agriculture Extension Revitalization Program (IAERP). Logan, Utah

- Lecturing in on-farm water management and irrigation scheduling
- Prepare tests, grading and evaluation
- Translate for Arabic speakers and led discussion in field trips to southern Utah

Fertility Researcher, Nov 1996 – Dec 1998
National Center for Agricultural Research and Technology Transfer (NCARTT) /
Ministry of Agriculture, Jordan

- Conducted trials related to fertigation and plant nutrition
- Advised farmers with the appropriate agricultural practices
- Produced educational materials in the related subjects for farmers

Research Assistant, Feb 1996 – Nov 1996
University of Jordan / Water and Environment Research and Study Center

- Coordinated the work with the Agricultural Mission in the French Embassy (MREA) in their “Optimization of Irrigation Water Project” through conducting field trials of irrigation scheduling by using Water Marks for vegetables in the Jordan Valley
- Coordinated any related activities such as the preparation for international conferences in related issues

Expert Assistant, Aug 1994 – Jan 1996
UNDP / Azraq Oasis Conservation Project, Jordan

- Conducted soil survey and soil sampling
- Collected field data and reporting
- Produced soil salt maps
- Conducted water return flow characteristics study

Teaching Assistant, Sep 1991 – Jan 1993
University of Jordan / Faculty of Agriculture

- Soil Principles Lab Course
- Soil Fertility and Fertilizers Lab Course

Research Assistant, Oct 1990 – Jan 1991
Faculty of Agriculture – University of Jordan / ICARDA funded project

- Prepared a site calibration curve for neutron probe measurement
- Prepared water content curves based on neutron probe measurements

CONSULTING EXPERIENCE

Moldova. Soil Salinity Evaluation: CĂRPINENII DE SUS, CAHUL, CHIRCANI-ZÎRNEşTI, SUVOROV, and TALMAZA.–Feasibility Study, Environmental and Social Assessment. Prepared for MWH Americas, Inc. and funded by the Millennium Challenge Corporation, USA. April 2009

WORK EXPERIENCE

Senior Agricultural Engineer, Academy for Educational Development – KAFA’A Project/ Funded by USAID/Jordan, Oct 2003 – Dec 2006

- Provided technical support and leadership in development of the KAFA’A extension program. This includes assistance in all field based research activities
- Identified opportunities for the implementation of innovative extension activities
- Provided technical support and leadership in developing a knowledge sharing program with farmers and agricultural extension personnel
- Provided leadership and technical support in the establishment of programs that links Jordanian farmers with university extension programs and university research activities
- Assisted in the development of water conservation program for farmers. This included identifying and recommending changes in behavior and water use that result water conservation
- Provided a leadership for KAFA’A extension personnel

Country Manager, Riyadh Green Oasis Est./ Kingdom of Saudi Arabia, Sept 2001 – Oct 2003 and
Country Manager, Oasis Agricultural Company L.L.C / Republic of Yemen, Jul 2000 – Jul 2001

- Prepared all commercial and technical correspondences related to the import of various agricultural products such as pesticides, fertilizers, public health products, seeds, equipments, etc. from different countries and suppliers

- Prepared tender quotations for the ministries, inviting quotations from overseas follow up to delivery
- Established and maintained customer relations to promote the services and obtain contracts and work orders as well as subsequent follow up to maintain client goodwill
- Conducted market survey, introducing new products to the market
- Directed all branches, assigning duties to engineer, foremen, technicians and labor

Extension Agent, Jordan Valley Authority, Irrigation Advisory Services Project, funded by USAID – Jordan and implemented under the Ministry of Water and Irrigation. Dec 1998 – Jul 2000

- Assisted farmers to increase efficiencies in the use of irrigation water
- Assisted farmers in determine cropping pattern and irrigation water requirements
- Assisted farmers in scheduling their irrigation (when and how much to irrigate)
- Assisted farmers in setting up farm agricultural practices record system

Secretary for the Registration Committee, Environmental Department / Ministry of Environment Sept 1989 – Sept 1990

- Conducted sites inspection and reporting
- Coordinated inspection committee’s schedules and meetings

WORKSHOPS AND TRAINING COURSES

The Wet Lands Conservation, Venice- Italy	June 28 – July 2, 1996
Irrigation Management and Scheduling, France	Sept 1 – 28, 1996
The Use of Bio-Fertilizers in Agriculture, Jordan	June 16 – 21, 1998
7 habits of Highly Effective People, Jordan	March, 2001
Drought Management Tools - Jordan	June, 2004
Development of Strategic Communication Plan - Jordan	June, 2004
Customer Service - Jordan	June, 2004
Effective Team Building - Jordan	August, 2004

PUBLICATIONS

1. Marjang, N. G.P. Merkley, M. Shaban. 2011. Center-Pivot Uniformity Analysis with Variable Container Spacing. Irrigation Science. March, 2011
2. Shaban, M. and G. Merkley. 2010. Training Tool for On-Farm Water Management Using Heuristic Simulation Software. USCID Conference. Fort Collins, Colorado
3. Shaban, M. and G. Merkley. 2010. On-Farm Water Management Game Using Heuristic Simulation Software. Utah State University Spring Runoff Conference

4. Shaban, M. and G. Merkley. 2008. Agricultural Water Resource Management Training Game (WaterMan Game). Utah State University Graduate Students Symposium
5. Producing 25 Extension Information Pieces in Irrigation and related fields for Development Alternatives Incorporation and funded by USAID- Jordan, to be distributed to the farmers – 1998. Extension leaflets
6. Irrigation Consumption Determination Methods for Jordan Valley Crops – 1998. Bulletin in Arabic
7. Irrigation Scheduling by Using Water Marks under Plastic Houses for Cucumber, tomato, and strawberry in the Jordan Valley – 1996. Proceeding
8. Return Flow characteristics in Al-Azraq Depression, Jordan – 1995. UNDP-Report
9. Irrigation Systems Evaluation in Al-Azraq Depression, Jordan – 1995. UNDP-Report
10. Soil status: Pedology and Salinity in Al-Azraq Depression, Jordan – 1995. UNDP-Report
11. Effect of Rates and Methods of Iron and Zinc Chelates Application on the Production and Fruit Quality of Sweet Orange Grown in the Jordan Valley, M.Sc. Thesis – 1994

SKILLS

Languages: Arabic (fluent), English (fluent)

PROFESSIONAL AFFILIATIONS

- United State Committee on Irrigation and Drainage (USCID).
- Jordanian Agricultural Engineers Association (JAEA)
- The Jordanian Society for Desertification Control and Badia Development (JSDCBD)
- Jordan Environmental Society (JES)