UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING, SCIENCES & MATHEMATICS
School Of Civil Engineering & the Environment

# Application of Genetic Algorithms for Irrigation Water Scheduling

by

**Zia Ul Haq**

Thesis for the degree of Doctor of Philosophy

April 2009

UNIVERSITY OF SOUTHAMPTON


ABSTRACT

FACULTY OF ENGINEERING, SCIENCES & MATHEMATICS
SCHOOL OF CIVIL ENGINEERING & THE ENVIRONMENT
Doctor of Philosophy
APPLICATION OF GENETIC ALGORITHMS FOR IRRIGATION WATER SCHEDULING
by Zia Ul Haq

A typical irrigation scheduling problem is one of preparing a schedule to service a group of outlets. These outlets may either be serviced sequentially or simultaneously. This problem has an analogy with the classical earliness/tardiness machine scheduling problems in operations research (OR). In previous published work integer programme were used to solve such problems; however, such scheduling problems belong to a class of combinatorial problems known to be computationally demanding (NP-hard). This is widely reported in OR. Hence integer programme can only be used to solve relatively small problems usually in a research environment where considerable computational resources and time can be allocated to solve a single schedule. For practical applications meta-heuristics such as genetic algorithms, simulated annealing or tabu search methods need to be used. However as reported in the literature, these need to be formulated carefully and tested thoroughly.

This thesis demonstrates how arranged-demand irrigation scheduling problems can be correctly formulated and solved using genetic algorithms (GA). By interpreting arranged-demand irrigation scheduling problems as single or multi-machine scheduling problems, the wealth of information accumulated over decades in OR is capitalized on. The objective is to schedule irrigation supplies as close as possible to the requested supply time of the farmers to provide a better level of service. This is in line with the concept of Service Oriented Management (SOM), described as the central goal of irrigation modernization in recent literature. This thesis also emphasizes the importance of rigorous evaluation of heuristics such as GA.

First, a series of single machine models is presented that models the *warabandi* (rotation) type of irrigation distribution systems, where farmers are supplied water sequentially. Next, the multimachine models are presented which model the irrigation water distribution systems where several farmers may be supplied water simultaneously. Two types of multimachine models are defined. The simple multimachine models where all the farmers are supplied with identical discharges and the complex multimachine models where the farmers are allowed to demand different discharges. Two different approaches i.e. the stream tube approach and the time block approach are used to develop the multimachine models. These approaches are evaluated and compared to determine the suitability of either for the irrigation scheduling problems, which is one of the significant contributions of this thesis. The multimachine models are further enhanced by incorporating travel times which is an important part of the surface irrigation canal system and need to be taken into account when determining irrigation schedules. The models presented in this thesis are unique in many aspects. The potential of GA for a wide range of irrigation scheduling problems under arranged demand irrigation system is fully explored through a series of computational experiments.

# Table of contents

ABSTRACT ..... i
Table of contents ..... ii
DECLARATION OF AUTHORSHIP ..... iv
Acknowledgement ..... v
Notations ..... vi
1 Irrigation management and Research aim ..... 1
1.1 Research aim ..... 5
2 Irrigation scheduling ..... 5
3 Irrigation water delivery schedules ..... 7
3.1 Demand schedules ..... 11
3.2 Arranged schedules ..... 12
3.3 Rotation Schedules ..... 13
3.4 Comparison of irrigation schedules ..... 14
4 Tertiary unit water management ..... 15
5 Irrigation modelling ..... 16
6 Optimization and Operational Research ..... 18
6.1 Static and dynamic models ..... 19
6.2 Linear and nonlinear models ..... 19
6.3 Integer and noninteger models ..... 19
6.4 Deterministic and stochastic models ..... 20
7 Irrigation scheduling and Operations Research ..... 21
7.1 Irrigation scheduling as a single machine problem ..... 23
7.2 Idle time ..... 27
7.3 Sequence-dependant setup times (travel time) ..... 29
7.4 Irrigation scheduling as a Multimachine problem ..... 31
7.5 Solution methods ..... 36
7.6 GA vs. SA ..... 37
7.7 Genetic algorithms (GA) ..... 38
8 Aim and objectives ..... 43
8.1 Problem description ..... 43
8.2 Aim of research ..... 44
8.3 Hypothesis ..... 45
8.4 Objectives ..... 45
9 Methodology ..... 47
9.1 Genetic representation ..... 49
9.2 Infeasibility ..... 51
9.3 Computational testing ..... 56
10 Single machine models ..... 59
10.1 Mathematical formulation ..... 59
10.1.1 Model 1 ..... 59
10.1.2 Model 2 ..... 62
10.2 GA implementation ..... 64
10.3 Results and discussions ..... 66
10.3.1 Model 2a ..... 68
10.3.2 Model 2b ..... 71

10.3.3 Model 2c....74
10.3.4 Conclusions....76
11 Multimachine models....78
11.1 Simple multimachine model....78
11.1.1 Stream tube approach (STA)....79
11.1.2 Time Block Approach (TBA)....85
11.2 GA implementation for stream tube and time block approaches....88
11.3 Comparison of stream tube GA and time block GA....91
11.3.1 Experiment 1....92
11.3.2 Experiment 2....98
11.3.3 Experiment 3....102
11.3.4 Experiment 4....109
11.4 Simple multimachine with setup times....113
11.4.1 Mathematical formulation....116
11.4.2 Experiment 5....119
11.5 Complex multimachine model....124
11.5.1 Experiment 6....125
11.5.1 Practical application....127
11.6 Complex multimachine with setup times....129
12 Summary of results....134
12.1 Sequential irrigation models....134
12.2 Simultaneous irrigation models....134
12.3 Travel time models....136
13 Conclusion and recommendations....137
13.1 Sequential irrigation models....138
13.2 Simultaneous irrigation models....139
13.3 Travel time models....140
13.4 Recommendations for future research....141
Appendix A: Application of Genetic Algorithms for Irrigation Water Scheduling....143
References....152

# DECLARATION OF AUTHORSHIP

I, Zia Ul Haq, declare that the thesis entitled:

"Application of Genetic Algorithms for Irrigation Water Scheduling"

and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as: Haq et al. (2008)

Signed: …………………………………………………………………………..

Date:……………………………………………………………………………….

# Acknowledgement

I would like to thank my supervisor Dr. Arif Anwar for his close supervision, support and encouragement during this work. Without his encouragement and help I would not have been able to complete this work. The support and cooperation of my sponsor (N.W.F.P. University of Engineering and technology, Peshawar, Pakistan) is also gratefully acknowledged.

Last but not least I would like to express my gratitude to my family for their patience, support and encouragement. I dedicate this thesis to my mother who has always been the source of inspiration and motivation for me. Without her endless prayers I may not have been able to achieve success.

## Notations

$\check{D}_{i-1}$ = duration of the job preceding the job at the $i^{th}$ position in the jobs sequence;

$D_j$ = duration of outlet $j$;

$D_k$ = duration of job $k$;

$E_j$ = earliness of outlet $j$;

$F$ = fitness function;

$G$ = irrigation interval;

$j$ = represents the outlet index;

$J$ = total number of outlets;

$Ķ_m$ = index of the earliest job on machine $m$;

$m$ = machine index = 1, 2… $M$;

$M$ = total number of machines available;

$M_{\boldsymbol{j}}$ = machine used by job $j$;

$M_{\boldsymbol{k}}$ = machine used by job $k \neq j$;

$P_C$ = penalty for capacity constraint violation;

$P_I$ = penalty for irrigation interval violation;

$P_O$ = penalty for overlap of jobs;

$q_j$ = required discharge of outlet $j$;

$Q$ = total discharge available or channel capacity;

$Q_{max}$ = count of distinct stream tubes used (total supply);

$r_{DS}$ = demand-supply ratio;

$R_C$ = penalty weight for $P_C$;

$R_I$ = penalty weight for $P_I$;

$R_O$ = penalty weight for $P_O$;

$\check{S}_1$ = the scheduled start time of the first job in sequence;

$\check{S}_i$ = scheduled start time of the job at the $i^{th}$ position in the jobs sequence (chromosome);

$\check{S}_{i-1}$ = scheduled start time of the job preceding the job at the $i^{th}$ position in the jobs sequence;

$S_{int}$ = start time of the irrigation interval;

$S_j$ = an element of the scheduled start time row vector;

$S_k$ = start time of any other job $k \neq j$ on the same machine as $j$;

$\hat{S}_{jk}$ = start time of the job proceeding job $j$;

$S_{Ķ_m}$ = scheduled start time of the earliest job on machine $m$;

$t$ = time block index = 1, 2…$T$;

$T$ = total number of time blocks;

$T_j$ = tardiness of outlet $j$;

$T_{jk}$ = travel time from $j$ to $k$;

$T_{0Ķ_m}$ = travel time from the head of the channel to the earliest job on machine $m$;

$Ŧ_j$ = target start time of outlet $j$;

$\alpha_j$ = cost of earliness per unit of time for job $j$;

$\beta_j$ = cost of tardiness per unit of time for job $j$;

$\gamma_{jk}$ = binary variable;

$\delta_j$ = binary variable;

$\boldsymbol{\eta}_{JJ}$ = a matrix that stores the information of jobs assignment to machines;

$\eta_{mk}$ = an element of matrix $\boldsymbol{\eta}_{JJ}$;

$\lambda_j$ = binary variable;

$\theta_{\boldsymbol{m}}$ = binary variable;

$\mu_{jk}$ = binary variable;

$\rho_{jk}$ = binary variable;

$\sigma_{jk}$ = binary variable;

$\chi_m$ = binary variable;

$\psi_{tj}$ = binary variable;

$\tilde{\psi}_{tj}$ = a variable that assumes a value $q_j$ if job $j$ is active in time block $t$, otherwise 0;

$\boldsymbol{\Omega}_{JJ}$ = a matrix that stores the information of the index and the start time of the earliest job on each machine used;

$\Omega_{mk}$ = an element of matrix $\boldsymbol{\Omega}_{JJ}$;

$\boldsymbol{\vartheta}_M$ = a vector that represents the maximum among the discharges of outlets serviced by the same machine;

$\vartheta_m$ = an element of vector $\boldsymbol{\vartheta}_M$;

$\tilde{\boldsymbol{q}}_{JJ}$ = a matrix that stores the information of jobs assignment to machines and their discharge requirements;

$\tilde{q}_{mj}$ = an element of matrix $\tilde{\boldsymbol{q}}_{JJ}$;

# 1 Irrigation management and Research aim

Water is one of the most precious and important of all the natural resources as the existence of life on Earth is dependent on it. The deep involvement of water in life processes makes living matter vulnerable to changes in the quantity and quality of water. It is reported by Clarke and King (2004) and several others that there is approximately 1,386 million km $^3$ of water in the whole world that covers seventy percent of the earth's surface. Nearly all (97.5 percent) of this water is contained in oceans, seas, salt water lakes and salty aquifers, and hence is salty and unsuitable for drinking or irrigation. Of the remaining 2.5 percent of freshwater, only 0.3 percent is found in rivers and lakes, the rest being frozen. This is depicted in Figure 1.1.

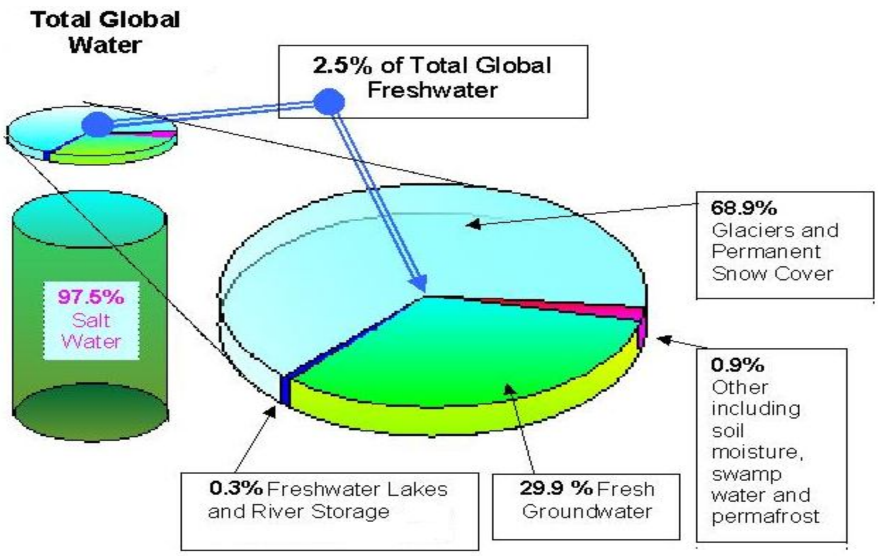



**Figure 1.1** Earth's hydrosphere (adapted from: Shiklomanov, 1999)

Water use can be broadly classified into three categories: agriculture, industry and domestic. Agriculture is the largest consumer of water worldwide for the production of food and fibre. The estimated total consumptive use of water worldwide for irrigated agriculture is nearly 85% of total human consumptive use, and is vital for food production (Falkenmark and Rockstorm, 2006; Gleick, 2003). Figure 1.2 presents the distribution of water used by different sectors mentioned above.

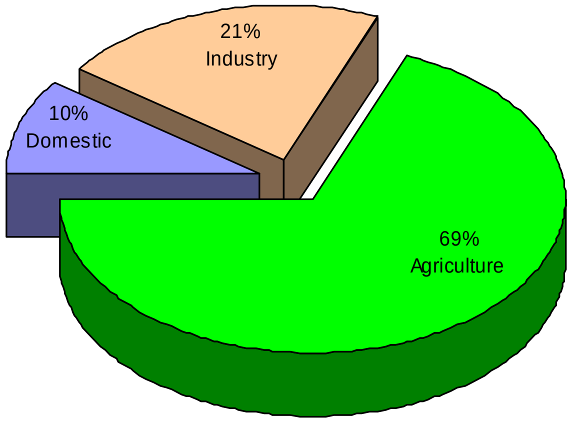



**Figure 1.2** Global water uses by sector (Data source: Clarke and King, 2004)

Nearly 4,000 cubic kilometres of fresh water is withdrawn every year and most of it is withdrawn for use in agriculture (Figure 1.2). Savenije (2000) citing Gardener-Outlaw and Engelmann (1997) reported that of the 1700 cubic metre per capita per year of renewable fresh water that is considered an individual's requirement, approximately 90 percent is needed for food production. In 2000, around 270 million hectares of land were irrigated worldwide, which is 18% of total cropland. Around 40% of all agricultural produce comes from these irrigated areas (Gleick, 2003). It has been also estimated that over 80 percent of the total increase in cereal production in Asia since the 1960s has been from irrigated land (Seckler et al., 1999).

However, with the increase in population, estimated to reach 8.9 billion by 2050 as compared to 6 billion in 2000 (Clarke and King, 2004), urbanization and industrial development competition for water is anticipated to be increased considerably. De Sherbinin and Dompka (1996) described population growth as the most important demographic trend affecting water resources. It is estimated that by 2025 the total amount of water withdrawn per year on average will be 5,235 cubic kilometres compared to 3,973 cubic kilometres in 2000 and 1,382 cubic kilometres in 1950 (Clarke and King, 2004), although the total volume of water around the globe, remains nearly fixed. FAO described the situation in the following terms:

*"Agriculture is the largest water user globally and faces increasing difficulty in securing a share of water resources that is sufficient to meet the needs of a growing world population and in managing the impacts of its activities on the resource base"* (FAO, 2006).

Similarly by projecting the current trends, a disturbing picture of the future is also drawn by IWMI (2003). According to the IWMI (2003) report it is estimated that by 2025 because of increased competition for water among different sectors, less water will be available for irrigation and as such will cause an annual global loss of 350 million metric ton of food production which is slightly more than the entire U.S. grain crop in 2003. The report further cautioned that if investment in sustainable water policy and management decreased over the next 20 years, the result will be major declines in food production and skyrocketing food prices. Figure 1.3 shows the increasing competition for water by industrial and municipal uses, including for energy generation relative to demand for agriculture. As competition for water from these other sectors intensifies, agriculture can expect to receive a decreasing share of developed freshwater resources (Comprehensive Assessment of Water Management in Agriculture, 2007).

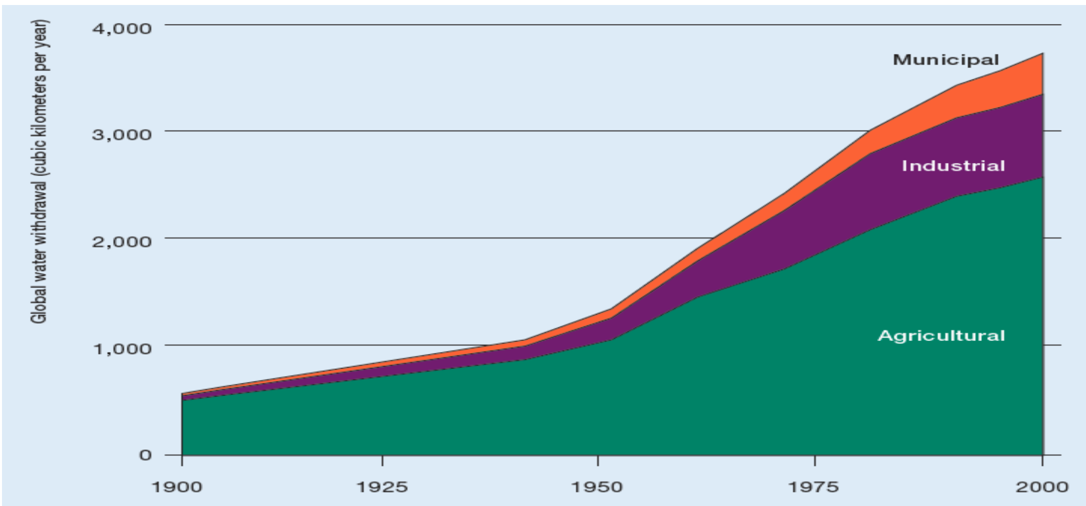



**Figure 1. 3** Sectoral competition for blue water (rivers, lakes, aquifer etc.) withdrawals for human uses (Comprehensive Assessment of Water Management in Agriculture, 2007)

Seckler et al. (1999) also reported that an estimated quarter of the world's population or a third of the population in developing countries live in regions that will experience severe water scarcity within the first quarter of the next century. Although Savenije (2000) indicated some serious flaws in water scarcity indicators that were used so far to indicate the level of water shortage in the different parts of the world. It does not however change the situation since the problem of water scarcity is still looming at large and the unprecedented growth rate of the world population does require an unprecedented increase in food production. Rijsberman (2006) discussed thoroughly, whether water is truly scarce in the physical sense (a supply problem) at a global scale or it is available but should be used better (a demand problem). Rijsberman (2006) suggested a “soft path for water” approach as described by Gleick (2003) to be the appropriate response to water scarcity i.e. a shift from supply management to demand management or in other words improving the overall productivity of water rather than endlessly seeking new supplies.

Smith (2004) discussed in detail how irrigation can contribute to poverty reduction and sustainable livelihoods. However, he also cautioned that badly designed and managed irrigation can negatively impact on poverty. Highlighting the benefits of effective irrigation management, Jahangir et al. (2003) described avoidance of risks in farming and sustainability in productivity as the two main benefits. Other inputs like fertilizer and improved technologies etc. become meaningless if a reliable source/supply of water is not available.

The above arguments provide justification for investing time and money in the proper management of water, and in particular in irrigation water, which is, as mentioned, the major share of water consumption globally. It also suggests that the management of water resources requires a priority in consideration. Irrigation scheduling is one of the managerial activities that aim at effective and efficient utilization of water for agriculture. Chambers (1983) put it in the following terms:

> “*Irrigation scheduling is a means of conserving water which helps in making decisions on allocation of quantity and timing of water supply commensurate with crop needs. It is one of the key activities that have the potential to improve performance of the system, especially its productivity, equity and stability*”.

## 1.1 Research aim

The aim of the research is to develop a decision support tool for arranged demand irrigation systems that schedules irrigation water deliveries to farmers as close as possible to their desired irrigation start times. The concept of classical machine scheduling in operational research (OR) is utilized to develop a series of irrigation scheduling models. A range of single machine (where farmers irrigate sequentially) and multimachine (where farmers irrigate simultaneously) scheduling problems will be studied.

# 2 Irrigation scheduling

Hill and Allen (1996) defined irrigation scheduling as the process of determining when to irrigate and how much water to apply. This is the most widely accepted definition of irrigation scheduling e.g. Heermann (1980), Singh et al. (1992). However sometimes irrigation scheduling is used synonymously with water delivery scheduling. For example, Clemens (1987a) used delivery schedule while Merriam (1987a) used the term irrigation schedule for the same purpose of determination of when a farmer should receive water from a distribution system. Both described delivery rate of flow, irrigation frequency and duration as the three inherent features of a schedule. However in contrast, Buchleiter and Heermann (1987) drew a clear distinction between different scheduling definitions, based on an individual's perspective of an irrigation system:

> *"A water supply district or the operator of an irrigation water delivery system may define an irrigation schedule as the starting time for a rate and volume of water to be delivered at each delivery point in the distribution network. ……For an individual irrigator who maintains a soil water budget and calculates crop water use for each field, irrigation scheduling is forecasting the time and amount of water to apply for the next irrigation."* (Buchleiter and Heermann, 1987).

This distinction in definitions is further highlighted by Goussard (1995). He stated that an irrigation schedule at the farm level would result in a delivery schedule at the distribution system level or scheme level. For planning and designing purposes and because of their close inter relationship, Goussard (1995) emphasized considering farm irrigation scheduling, delivery scheduling and delivery system as a whole.

The determination of irrigation schedules or when to irrigate and how much to apply is mainly based on either actual soil water measurements or soil water balance calculations using water balance approach. In water balance approach the change in soil moisture over a period is given by the difference between the inputs (irrigation plus precipitation) and the losses (seepage plus runoff plus drainage plus evapotranspiration). Another approach is, the plant stress sensing where irrigation scheduling decisions are based on plant responses rather than direct or indirect measurement of soil water status. The advantage of this approach is that many features of plant physiology respond directly to changes in water status in the plant tissues, rather than to changes in the bulk soil moisture. The disadvantage is the practical difficulties of implementation thus limiting the development of a commercially successful system (Jones, 2004). Similar views were expressed by Goldhamer and Fereres (2004).They termed plant indicators as ideal for irrigation scheduling; however they also described the dynamic nature of plant water status and the lack of suitable indicators, relative to established methods based on atmospheric and soil observations as the main shortcomings of the method. Similarly Singh et al. (1992) also described plant indicators, soil indicators and water balance techniques as the three principal methods for determining when to irrigate. These methods, together with additions, were also described by Hill and Allen (1996). The additional methods described by Hill and Allen (1996) are:

- Irrigation on fixed intervals following a simple calendar or a predetermined schedule.
- Irrigation when one's neighbour irrigates.
- Any combination of the different methods.

From the above discussion it is evident that the irrigation scheduling problem is basically a two step process. The first step is obviously the determination of when to irrigate and how much water to apply during irrigation for satisfying all crop water requirements. The second step is then the determination of a water delivery schedule to supply the requirements/demands as determined in the first step. In contrast to the first step, the second step is more of an art than an exact science. It involves a great deal of decision making based on judgment and experience in addition to the prerequisite knowledge. Some of the decisions that are required to be made include, who is to be receiving water i.e. the **sequence** and how much each user is to be receiving and for how long i.e.

**rate of flow** and **duration**. Other decisions include whether full demand (full irrigation) or partial demand (deficit irrigation) is to be supplied and whether to supply water to increase production on per unit area basis or per unit water consumed. Similarly decision regarding the level of service or flexibility is also required. Thus the second step in the irrigation scheduling process is indeed a challenging task in totality for the irrigation managers and is the focus of this study. For the purpose of this study, irrigation scheduling is the scheduling of irrigation water i.e. determining the timing of irrigation water delivery to an individual farmer unless otherwise specified.

# 3 Irrigation water delivery schedules

Several delivery methods are in practice in irrigated agriculture throughout the world and a variety of classifications have been suggested by different researchers. The different classifications of irrigation schedules found in literature are:

1. Classification by Replogle and Merriam (1980)-Table 3.1
2. Classification by Clemens (1987a)- Table 3.2
3. Classification by Sagardoy et al. (1982)- Table 3.3
4. Classification by FAO (1982)- Table 3.4
5. Classification by Horst (in Eggink and Ubels 1984)- Table 3.5

Replogle and Merriam (1980) based their classification on the three main variables in an irrigation schedules i.e. frequency, rate, and duration of irrigation water supply. Replogle and Merriam (1980) defined nine different types of irrigation water delivery schedules by using a combination of different control criteria over the frequency, rate, and duration of the irrigation water supply, as given in Table 3.1 in detail. The control criteria were whether the frequency, rate, and duration of the irrigation water supply are unlimited or arranged or fixed. Replogle and Merriam (1980) also indicated that the schedules may be either supplier controlled or user controlled and rigid or flexible. Clemens (1987a) classification as given in Table 3.2 is not very different than the Replogle and Merriam (1980), however, Clemens (1987a) clearly differentiated between three control regimes i.e. local control, intermediate control, and central control. Under each control regime subclasses were defined based on different combinations of control over the frequency, rate and duration of the irrigation water supply. Similarly the classification given in Table3.3 by Sagardoy et al. (1982) as quoted by Manz

(1988) is a combination of the classifications given by Replogle and Merriam (1980) and Clemens (1987a). The classification by FAO (1982) as quoted by Jurriens et al. (1989), given in Table 3.4 is simpler than other classifications discussed, and is easily understandable. The five different types of schedules by FAO (1982) are adequately defined in Table 3.4.The classification by Horst (in Eggink and Ubels 1984) is almost similar to that by FAO (1982), however, is not as generic and comprehensive as the latter, and is presented in Table 3.5.

**Table 3.1** Classification according to Replogle and Merriam (1980)

| Schedule category | Frequency | Rate | Duration |
|---|---|---|---|
| Demand | Unlimited | Unlimited | Unlimited |
| Limited-rate, demand | Unlimited | Limited | Unlimited |
| Arranged | Arranged | Unlimited | Unlimited |
| Limited-rate, arranged | Arranged | Limited | Unlimited |
| Restricted-arranged | Arranged | Constant | Constant |
| Fixed-duration, restricted-arranged | Arranged | Constant | Fixed by policy |
| Varied-amount, constant-frequency (modified-amount rotation) | Fixed | Varied as fixed | Fixed |
| Constant amount, varied frequency (modified-frequency rotation) | Varied as fixed | Fixed | Fixed |
| Constant-amount, constant-frequency (rotation) | Fixed | Fixed | Fixed |

| Terminology used by Replogle and Merriam (1980): | |
|---|---|
| Unlimited | Unlimited and controlled by the user. |
| Limited | Maximum flow rate limited by physical size of system or turn out capacity but causing only moderate to negligible problems in farm operations. The applied rate is controlled by the user and may be varied as desired. |
| Arranged | Day or days of water availability are arranged between the water agency and the user. |
| Constant | The condition of rate or duration remains constant as arranged during the specific irrigation run. |
| Fixed | The condition is predetermined by the water agency. |

**Table 3.2** Classification according to Clemens (1987a)

| **Local control** | **Intermediate control** | **Central control** | |
|---|---|---|---|
| (Demand schedules) | (Arranged schedules) | Central system schedules | Rotation schedules |
| Demand | Arranged | Central system | Rotation |
| Limited rate demand<br>Arranged frequency demand | Limited rate arranged<br>Restricted arranged<br>Fixed duration arranged<br>Fixed rate/rest. arranged | Fixed amount | Varied amount rotation<br>Varied frequency rotation<br>Continuous flow |

**Table 3.3** Classification according to Sagardoy et al. (1982), quoted by Manz (1988)

| **Schedule category** | **Frequency** | **Rate** | **Duration** |
|---|---|---|---|
| On demand | Farmer controlled, Unlimited flexibility | Farmer controlled, Unlimited flexibility | Farmer controlled, Unlimited flexibility |
| Semi demand | Collaborative control, Unlimited flexibility | Collaborative control, Limited varied as fixed flexibility[a] | Collaborative control, Limited varied as fixed[a] |
| Canal rotation and free demand | Agency controlled, inflexible | Farmer controlled, Unlimited flexibility | Farmer controlled, Unlimited flexibility |
| Rotational system | Agency controlled, Inflexible[b] | Agency controlled, Inflexible[b] | Agency controlled, Inflexible[b] |
| Restricted arranged | Arranged | Constant as arranged | Constant as arranged |
| Continuous flow | Agency controlled, inflexible | Agency controlled, inflexible | Agency controlled, inflexible |

[a] Volume limited

[b] Collaborative control and more flexibility may be possible within the general definition provided for rotational method of water distribution

**Table 3.4** Classification according to FAO (1982) as quoted by Jurriens et al. (1989)

| Schedule category | Explanations |
|---|---|
| On-demand | Water is available to the farmer any time that the intake of hydrant is opened. Therefore the amounts to be used are not limited but water consumption is usually metered and paid for per cubic metre. |
| Semi-demand | Water is made available to the farmer within a few days (generally 2—7 days) of his request. The amount is often limited to a certain volume per hectare. |
| Canal rotation and free demand | Secondary canals receive water by turns, for example every 7 days, and once the canal has water farmers can take the amount they need at the time they wish. |
| Rotational system | Secondary canals receive water by turns and the individual farmers within a given canal area receive the water at a pre-set time and generally in a limited quantity. |
| Continuous flow | Throughout the irrigation season, the farmer receives a small but continuous flow that compensates the daily crop evapotranspiration |

**Table 3.5** Classification according to Horst (in Eggink and Ubels 1984), quoted by Jurriens et al. (1989)

| Schedule category | Explanations |
|---|---|
| Continuous delivery | Continuous flow to each farm, adjusted during the growing season. Not suitable for small holdings |
| Free delivery/ continuous full supply | Continuous flow through all canals from which farmers can take water as needed. Unused water flows back to the river. Only feasible if water is not scare, e.g. if diverted by gravity from a river or lake. |
| Free delivery/ various flows | System delivery automatically adjusts itself to farmer's demand. Only possible by means of sophisticated and vulnerable automatic downstream control structures. |
| On demand delivery | Due to strong variations in demand, this requires frequent readjustment structures. |
| Rotational delivery/ varying intervals | Constant flow to terminal unit is rotated among farmers within that unit. Interval is adjusted to water requirements over growing season. |
| Rotational delivery/ varying flows | Interval remains constant, flow is adjusted. Complicated structures required. Danger for inefficient use of water and canal capacities |

Similarly Merriam (1987a) described demand, arranged, and rotation as the three types of irrigation schedules. Manz (1988) reviewed the classification given in Table 3.1 and 3.3, criticized them as confusing and presented his own simple version as follows:

1) Unlimited flexibility: frequency, rate and duration are all unconstrained.
2) Limited flexibility: frequency, rate and duration unconstrained within a limited range.
3) Inflexible: frequency, rate and duration constant throughout irrigation season.
4) Unlimited varied as fixed flexibility: frequency, rate and duration are held constant for each run though unconstrained but may be different in different runs.
5) Limited varied as fixed flexibility: the same as in 4) above but frequency, rate, and duration are unconstrained within a limited range

The literature suggests that there is no single standard, and comprehensive classification system, a view also expressed by Jurriens et al. (1989) and Manz (1988). Jurriens et al. pointed out that often it is difficult to find the precise meaning of the different terms used in irrigation schedule classifications and it is also not often clear whether one is dealing with the main system level or with the tertiary unit level. However the majority schedules do embody the three main characteristics related to any irrigation schedules i.e. frequency, rate, and duration. Demand, arranged, and rotation that turn out to be the three main types of irrigation schedules/delivery methods are hereby discussed briefly.

## 3.1 Demand schedules

Demand, on-demand, and free delivery are the terms normally used in literature for such schedules (Table 3.4 and 3.5). They are termed as the most flexible schedule by Clemens (1987a) and the most sophisticated by Bishop and Long (1983). Merriam (1987a) described demand schedules as the schedules that are completely controlled by the farmer with some practical compromise between the farmer and water supplier as to the maximum flow rate. It consists of making water delivery to the farmer based on his requested time and amount. The flow rate, though usually limited to a certain maximum, is much higher than other methods.

When there is no automatic supply or abundant water, a pure on-demand method is not possible because the canal flows cannot immediately be adapted to the changing demands downstream. This needs, some time and requires arrangements between users and suppliers. Hence comes the term arranged schedules. Clemens (1987a) described limited rate demand and arranged frequency demands as the two variations of a demand schedule. Flow rate, frequency of irrigation, and duration are determined by the farmers in a limited rate demand however, the flow rate is limited to a certain maximum amount. In arranged frequency demand schedules, the irrigation start time is also arranged.

## 3.2 Arranged schedules

In this kind of schedule, the rate, frequency, and duration are arranged between the farmer and the water supply agency. According to Clemens (1987a), these arrangements are often on a more local level than on the project level and thus allow for last minute changes in arrangements. Merriam (1987a) described it useful only when the restriction on making arrangements is minimal. He described it a compromise between the positive values of increasing flexibility with fewer constraints and a lesser capital cost resulting in less automation and a more rigid system.

There are many variations of arranged schedules as described earlier under the different classification systems. Clemens (1987a) noted that the limited rate arranged schedule is very flexible, where the restriction only applies to the flow rate and the frequency and duration are arranged according to farmer needs. Even changes in duration and frequency are allowed during irrigation, but through arrangements. However Merriam (1992a) described limited rate arranged schedule as the one where only the frequency is to be arranged, the flow rate is either unlimited or limited to a certain degree, and duration is also unlimited. Clemens (1987a) described restricted arranged schedules to be another variation of arranged schedules, where the rate and duration once set, are unchangeable during irrigation. These restrictions make it less flexible. Similarly, different variations are possible with a range of combinations of arrangements.

## 3.3 Rotation Schedules

The rate, frequency, and duration are all fixed and remain fixed for the entire irrigation season in rotation schedules and each farmer is supplied water sequentially for a specified period of time. They are also termed rigid and central controlled or agency controlled schedules, as can be seen in Tables 3.1 to 3.5. Rotation schedules are described by Clemens (1987a) as the most restrictive of all the irrigation schedules. However, Clemens (1987a) described it as fairer and equitable than a more flexible delivery schedule in situations where proper administrative controls are lacking. They also require the least capital cost. Bishop and Long (1983) described rotation schedules as the most widely used of the modern irrigation delivery methods. Rotation schedule is the predominant irrigation delivery method in Pakistan (Latif and Sarwar, 1994) which has the largest integrated irrigation system in the world (Khan et al., 2006). Locally in Pakistan, rotation schedule is known as *warabandi* ("*wara*" means turn, "*bandi*" means fixed). However different variations and local names for *warabandi* can be found in different parts of the subcontinent.

As described earlier, and noted by Jurriens et al. (1989) a certain degree of confusion prevails in irrigation literature regarding the way rotation schedules are described. Any definition of rotation schedules should specifically mention the level at which the rotation is taking place. For example it is possible that rotation of supply takes place between canals and then the tertiary unit serviced by each canal is operated with any other method as the "Canal rotation and free demand" schedule given in Table 3.4. The selection of a manageable supply stream of sufficient size is essential in rotation schedules. Bishop and Long (1983) suggested a stream size of 30-50 litre/second for small sized farms generally found in developing countries as an easily manageable size. He described 1.5 litre/second/hectare as the conventional rule of thumb with the area in rotation ranging from 20-40 hectares to a maximum of 60-70 hectares.

Clemens (1987a) suggested continuous flow, varied amount rotation, and varied frequency rotation as some of the possible variations of rotation schedules. In continuous flow schedules the duration is the entire season and the frequency is once per year. Flow rates can be varied over the season to better match crop water requirements. Generally the frequency remains fixed, while duration and/or rate is varied to apply more or less water to a particular area in a varied amount rotation schedule. In varied frequency rotation schedules the frequency of water delivery is varied to make adjustments for crop water requirements. Similarly other variations are also possible as in Table 3.1 and 3.5.

## 3.4 Comparison of irrigation schedules

The use of flexible delivery schedules (demand and arranged) as against rigid rotation schedules has been largely advocated by researchers. The reasons for the popularity of flexible schedules are: increased yields, conservation of water and energy due to the application of the right amount of water at the right time thus satisfying all crop water requirements. It also provides opportunity to farmers for optimum utilization of their resources (Clemens, 1987a, 1987b; Merriam, 1987a, 1987b, 2007; Replogle, 1987 etc.). Clemens (1987a) commented that the added costs in case of flexible schedules for infrastructure and management can be offset by improvement in operational efficiency. However Jurriens and Wester (1995) while citing some theoretical research by other researchers, observed that the yields, labour requirements, and infrastructure costs of rotation or *warabandi* schedules are as good as those of more flexible schedules. But they believed warabandi still suffers from a substantial performance problem which ultimately can be traced back to inequitable distribution. Existing rotational practices allocate equal time per unit land area and ignore water losses along the supply watercourse resulting in inequitable distribution downstream (Latif and Sarwar, 1994). It also does not take into consideration soil characteristics and shape or layout of the fields. Carrying forward this discussion of for or against *warabandi*, Jurriens and Wester (1995) quoted Perry (1993) as:

> "*The increased costs (in terms of infrastructure, maintenance and management requirements) of sophisticated, flexible water delivery schedules are unlikely to be offset by significant yield increase*."

Jurriens and Wester (1995) believed that *warabandi* should be improved by including water losses in the rosters, making main system management more reliable, and accepting more flexible implementation, rather than going for a complete new system that is entirely foreign to the local conditions. However the vast majority of irrigation experts still believe that it is necessary to establish a high degree of flexibility in water delivery to adapt irrigation applications to crop water requirements and farming needs. For example, Merriam et al. (2007) considered flexibility essential, to optimizing farming operations and maintaining sustainable irrigated agriculture.

Clemens (1987a) suggested, each situation must be examined individually and it is important to select the proper degree of delivery flexibility that provides reasonable control to the farmer while still maintaining economical and efficient distribution system operations. Clemens (1987a) described the following reasons as a criteria for selecting one schedule over another: "*type of irrigation system and crops, size and complexity of farming operations and level of farmers' knowledge about irrigation, type of physical controls, manpower requirements and availability, communication requirements, level of technology required to operate and maintain project, system capacities required and the effect of delivery schedule on overall project efficiency*".

# 4 Tertiary unit water management

The objective of this topic is to highlight the importance and place of the tertiary unit in the whole irrigation scheme and to clarify that the focus of the irrigation scheduling models to be researched in this study is related to the distribution of irrigation water in a tertiary unit. Irrigation water management is defined by Jurriens and Wester (1995) as

> "*the organized use of resources (human, physical, financial) for the planning, operation and monitoring of tasks and activities related to the water distribution and use for irrigated agriculture, including maintenance, drainage, conflict control and cost recovery, including also organizational structures and communications, all for the realization of goals and objectives of organizations and individuals involved*".

An irrigation scheme may consist of different levels: main system (main canals, secondary canals), tertiary unit and farm level. As described earlier in the preceding paragraphs, the distinction between these different levels is not usually specifically mentioned in literature while describing irrigation scheduling methods. Also the distinction between management at these levels and the scheme management is not always clear. By and large, irrigation water management is about the main system, the tertiary unit and the interaction between them. In literature mostly the term farm management or on-farm management is used, referring largely to tertiary unit level (Jurriens and Wester, 1995). A tertiary unit is basically the terminal unit in an irrigation scheme. However it does not apply to very large farms where there is no tertiary system in between the farm and the secondary canal or main system.

Over the last several years there have been many shifts in focus from one aspect of irrigation management to another. As described by Jurriens and Wester (1995), "*besides the shift in focus from technical aspects to organizational and socio-economic aspects, there was also another shift of focus from an individual farmer to tertiary unit as whole*". Improvement of the water distribution is one of the measures that are generally considered necessary for effective water management at the tertiary level. In fact, all other measures like infrastructure, farmers training and institutions etc., are largely meant to finally improve the water distribution. The proper use of water on the farm can only be achieved after having established an adequate and reliable distribution within the tertiary unit (Jurriens and Wester, 1995). Makin and Cornish (1995) observed that the increased competition for water resources among different sectors stimulated the efforts to improve water use efficiency. As a result improved water management is being included in the objectives of many rehabilitation projects, with computer-based irrigation scheduling viewed as a promising tool.

# 5 Irrigation modelling

Determining water supply schedule for an individual field at farm level with a single crop assuming uniform soil and climatic conditions is not too difficult a task. However when it comes to large irrigation schemes/districts where crops, soils and climatic conditions are different in different areas/sectors/units of the scheme, then the situation becomes more complex. This is increasingly the case when the water supply is limited and irrigation managers are unable to supply the full demands of all individual users. However with the help of simulation and optimization techniques it has been made possible to deal with any such situation. Models are available that are capable of scheduling optimal canal release for an on-demand or rotation system (Mishra et al. 2005), allocate water and land optimally to different crops at different level i.e. interseasonal, seasonal and intraseasonal under water scarcity (Prasad et al. 2006) and allocate resources, i.e. water and land in a heterogeneous irrigation schemes under rotational water supply (Gorantiwar and Smout 2003; Gorantiwar and Smout 2005; Smout and Gorantiwar 2005; Smout and Gorantiwar 2006a; Smout and Gorantiwar 2006b). Chen (1997) presented a genetic algorithm solution to the problem of allocating scarce water resources to several irrigation districts for maximization of economic benefits.

Specific issues related to equity like losses due to seepage and other operational losses have also been addressed in some delivery scheduling models. For example, Latif and Sarwar (1994) presented a variable time model which allocates a constant volume of water per unit area to all the farmers in the command area for achieving equity, rather than equal time per unit of area thus accounting for the transmission losses along the canal in the *warabandi* system. The difference between this and earlier variable time models is that in this model the total losses are deducted from the total available volume of water at the watercourse head before fixing the warabandi schedule. The same problem of inequitable distribution was tackled by Khepar et al. (2000) by taking into account the seepage losses along the watercourse. Their model ensured equitable distribution of water according to the land holding of a farmer irrespective of his location on the watercourse by the introduction of a seepage factor (i.e. the ratio of discharge released at the water course inlet to the actual discharge being received by the farmer and calculated by the model). Hamilton and De Vries (1986) presented SCHEDULE, a series of three microcomputer programme for scheduling irrigation canal water deliveries. The objective was to enable the evaluation of alternative distribution schemes in a rapid and efficient manner and match demands with supply in a group rotation distribution. Based on farmers flow requests, SCHEDULE computes the desired flow in each canal and supplies farmers' demands in a unique way by dividing the area to be irrigated by the canal into three rotation groups and then establishing priority for each group. Although the programme addresses the demands of farmers in terms of flow rate, it does not take into consideration the time at which water is to be supplied. Similarly, Zimbelman and Bedworth (1983) developed an algorithm for the automated control of an open channel/canal water distribution system. The algorithm requires water surface elevation in the channel as an input and computes the necessary gate adjustment required to supply the exact demand in an on-demand system.

Another category of models deals with the basic question of when to irrigate and how much to apply to an agricultural field. The objective is to satisfy all crop water requirements at different growth stages throughout the season. This has been an extensively researched area and computer models are available to make water supply schedules for any crop on any soil under any climate. For example Singh et al. (1992) developed AISSUM, a fully automated computer assisted system for monitoring and analyzing the requisite weather, crop, and soil data for determining the timing and amounts of irrigation applications. Similarly OSIRI by Chopart et al. (2007), INCA by Makin and Cornish (1995), and WISE by Leib et al. (2001) etc. are the other computer models in this category.

It may be concluded that irrigation models developed so far are either for allocation of water and/or land optimally to crops to maximise crop yields and overall benefits etc. or for equitable distribution of water or for determination of time and amount of crop water requirements. The majority of these models use some kind of optimisation and simulation techniques which has its roots in operational research (OR). However there are also some irrigation scheduling models which directly capitalize on the wealth of information in OR and use established OR models and terminology. Such models and the connection between optimization and OR, and between irrigation scheduling and OR will be elaborated in the subsequent sections.

# 6 Optimization and Operational Research

The official definition of operational research adopted by the U.K. Operational Research Society as quoted by Spedding (1980) is:

> "*the application of the methods of science to complex problems arising in the direction and management of large systems of men, machines, materials and money in industry, business, government and defence. The distinctive approach is to develop a scientific model of the system, incorporating measurements of factors such as chance and risks, with which to predict and compare the outcomes of alternative decisions, strategies and controls. The purpose is to help management determine its policy and actions scientifically*".

It was further mentioned by Spedding (1980) that at least 38 other definitions were known at the time when this definition was adopted by the U.K. Operational Research Society. Winston (2004) defined OR (often referred to as management science) as "*a scientific approach to decision making that seeks to best design and operate a system, usually under conditions requiring the allocation of scarce resources*." Mathematical models are the usual way of developing a scientific approach to decision making. A mathematical model is a mathematical representation of some phenomenon in order to gain a better understanding of that phenomenon which in turn may be used to make better decisions.

Most of the mathematical models used in OR are optimization models. An optimization model seeks to find values of the decision variables that optimize (maximize or minimize) an objective function among the set of all feasible values for the decision variables. Thus objective function(s), decision variables, and constraints are the three components of any optimization model. The function whose value is to be maximized or minimized is called the model's objective function. Decision variables are the variables whose values are in the control of the modeler and influence the performance of the system. Restrictions on the values of decision variables are called constraints because only certain values of the decision variables are possible or feasible. Of the many types of optimization models, only some are discussed here briefly.

## 6.1 Static and dynamic models

When the decision variables do not involve sequences of decisions over multiple periods i.e. the values of problem parameters do not change over multiple periods, the model is then called a static model. In a dynamic model the decision variables do involve sequences of decision over multiple periods i.e. the values of problem parameters do change over multiple periods (Winston, 2004).

## 6.2 Linear and nonlinear models

Linear models deal with the optimization of a function, subject to a set of constraints in the form of linear equations and or inequalities. In nonlinear models the objective function and or one or more of the constraints are nonlinear (Gupta and Hira, 2007).

## 6.3 Integer and noninteger models

If one or more decision variables must be an integer, then the model is termed as an integer model. When all the variables are restricted to be integers then it is termed a pure integer model and in case only some of the variables are constrained to be integers then it becomes a mixed integer model. In some cases the variables assume only binary values (zero or one), such models are then referred to as zero-one models (Gupta and Hira, 2007). If all the variables are free to assume fractional values then it becomes a noninteger model (Winston, 2004).

## 6.4 Deterministic and stochastic models

When for any values of the decision variables the value of the objective function and constraint satisfaction or dissatisfaction is known with certainty, the model is termed deterministic model, otherwise it is a stochastic model (Winston, 2004). Gupta and Hira (2007) defined stochastic programming models, also called probabilistic models, as those that refer to linear programming and include an evaluation of relative risks and uncertainties in various alternatives of choice for management decisions.

In most of the OR techniques, solutions are obtained by using algorithms. An algorithm is a set of instructions or computational rules that are applied iteratively to the problem, with each repetition moving the solution closer to the optimum. It is necessary for these algorithms to be executed on computers because usually the associated computations are tedious and voluminous. However some mathematical models may be so complex or computationally demanding, that it may not be possible to solve them by any of the available optimization algorithms within at least, practical time limits. In such cases it may be required to use some other methods such as heuristics or rules of thumb. Heuristics may or may not find an optimum solution but may find a near optimum solution within reasonable time limits (Taha, 2007).

# 7 Irrigation scheduling and Operations Research

Since structural infrastructure referred to by Clemens (1987a) as hardware for irrigation requires a huge financial investment, an appropriate effort is also essential in developing scientific management tools. Mujumdar (2002) noted that developments in system science, operations research, and mathematical modelling for decision making have been usefully exploited for water resources management in many developed countries. Mujumdar (2002) observed that in many developing countries, irrigation water management has been very inefficient technologically, economically, and environmentally, thus proper scientific management of irrigation water in these countries may form a significant non-structural or as referred to by Clemens (1987a), a software measure.

Scheduling has been the subject of a significant amount of research in the field of operations research. Baker (1974) defined scheduling as the allocation of resources over time to perform a collection of tasks. Scheduling problems arise in many practical situations and have a wide range of forms e.g. arranging grocery deliveries or rubbish collections, creating staff work rotas for hospitals, or for bus or train services, arranging examination timetables in universities, arranging for a set of items being produced in a factory to each visit certain machines, scheduling equipment maintenance, and so on. The variety is enormous and as observed by Hart et al. (2005) there is no natural and obvious taxonomy for categorising scheduling problems in general. Many are basically optimization problems having the form: given a collection of tasks to be scheduled on a particular processing system, subject to various constraints and the goal is optimizing the value of the objective function (Garey et al., 1978). They are commonly described as machine scheduling problems. Jain and Meeran (1999) have given a comprehensive review of the classical scheduling problems, techniques used for their solution and a compact representation of the different classes of these techniques.

There has been great emphasis on investigating machine scheduling problems in literature where jobs represent activities/tasks and machines represent resources. Machine scheduling problems can be classified in a number of ways. Based on the number of machines or processors, it can be classified as either single machine or multi-machine (or parallel-machine) scheduling problems. In a single machine scheduling problem jobs are given with each job having a specified duration and a preferred starting time (or, equivalently, a preferred completion time). The jobs are scheduled on a single machine/processor nonpreemptively (i.e. jobs cannot be split). In multi-machine scheduling problems a set of machines is available for processing a set of jobs. There are a number of variations of single machine and multi-machine problems. Irrigation scheduling is also not any different than the classical scheduling problems found in OR. Considerable work has been done to develop mathematical models for irrigation water management for a variety of different objectives. Table 7.1 shows a number of different models used in water management. Details like specific issues addressed by each model, type of model, objectives and references are all described in the table. Therefore a separate description of each model is not given to avoid repetition. However some additional models, not given in Table7.1, that are worth mentioning are described.

**Table 7.1** Mathematical tools used in water management (Mujumdar, 2002)

| *Objective* | *Mathematical Tools* | *Issues Addressed* | *Recent Literature* |
|---|---|---|---|
| Long-term, steady state reservoir operation for irrigation | Stochastic Dynamic Programming | Uncertainty due to randomness of hydrologic variables; single decision-making mechanisms for reservoir operation and crop water allocations | Houghtalen and Loftis (1988); Dudley (1988); Dudley and Scott (1993); Vedula and Mujumdar (1992); Vedula and Nagesh Kumar (1996); Ravikumar and Venugopal (1998) |
| Field-level irrigation scheduling | Stochastic/Deterministic Dynamic Programming; Linear Programming | Seasonal and intraseasonal allocation of deficit water; competition among crops; crop yield optimization | Paudyal and Das Gupta (1988); Rao et al. (1990); Azar et al. (1992); Mannocchi and Marcelli (1994); Sunantara and Ramirez (1997); Paul et al. (2000); Anwar and Clake (2001) |
| Real-time operation of irrigation systems | Dynamic Programming; Linear Programming; Simulation | Adaptive operation; real-time forecasts of hydrologic variables | Dariane and Hughes (1991), Rao et al. (1992); Mujumdar and Ramesh (1997); Wardlaw and Barnes (1999) |
| Performance evaluation | Monte-Carlo Simulation | Evaluation of reliability, resiliency, vulnerability and productivity index | Hashimoto et al. (1982); Mujumdar and Vedula (1992); Srinivasan and Philipose (1988); Srinivasan et al. (1999); Sahoo et al. (2001) |
| Conflict Resolution | Fuzzy Optimization | Uncertainly due to imprecision in goals and constraints | Kindler (1992); Fontane et al. (1997); Bender and Simonovic (2000) |
| Qualification of Imprecision | Fuzzy Inference Systems; Fuzzy Relation Analysis | Formulation of fuzzy rules | Shreshta et al. (1996); Russell and Campbell (1996); Yin et al. (1999); Teegavarapu and Simonovic (2000); Despic and Simonovic (2000); Panigrahi and Mujumdar (2000) |

Suryavanshi and Reddy (1986) for the first time used mathematical modelling (0-1 linear programming) for obtaining the optimal operational schedule of canal outlets. The work by Wang et al. (1995), Reddy et al. (1999) and Anwar and Clark (2001) were an improvement over Suryavanshi and Reddy (1986). De Vries and Anwar (2004), for the first time, demonstrated irrigation scheduling under *warabandi* as a single machine problem in OR and presented a solution using integer programming. Anwar and De Vries (2004) presented a heuristic solution to the same problem. Wardlaw and Bhaktikul (2004) presented a genetic algorithm (GA) solution to the problem described by Anwar and Clark (2001) and compared their results with the latter. Anwar et al. (2006) demonstrated arranged demand irrigation to be a continuous function with on-demand on one extreme and fully arranged demand schedule as the other by introducing the index of relative timeliness. De Vries and Anwar (2006) added travel time to the models developed by De Vries and Anwar (2004) and presented a mixed integer linear programming solution. Travel time in irrigation is the equivalent of setup times in classical scheduling problems of OR.

## 7.1 Irrigation scheduling as a single machine problem

Sequential or rotation schedules are in practice in many irrigation systems through out the world. In a rotation schedule water is distributed sequentially amongst a group of users either starting from the upstream user, or starting from the most distant user, and scheduling each adjacent user in turn. This traditional approach does not allow users to specify when they wish to receive water for irrigation. De Vries and Anwar (2004) presented irrigation scheduling models that allow users to specify when they wish to use water, under an arranged demand irrigation system. De Vries and Anwar (2004) were the first to demonstrate sequential irrigation under arranged demand irrigation system analogous to single machine scheduling problem found in OR. Table 7.2 provides details of this analogy. The supply of irrigation water to farmers in an irrigation scheme under rotation (*warabandi*) can be described as a single machine scheduling problem, as there is a single resource/machine i.e. water and a number of jobs to be processed, i.e. farmers supplied with water. The duration of water required by any farmer is comparable to the processing time of any job. A single machine irrigation schedule is illustrated in Figure 7.1.

**Table 7.2** Single machine scheduling vs. Irrigation scheduling (De Vries and Anwar, 2004)

| | **Single machine scheduling ( some common characteristics)** | **Irrigation scheduling (under *warabandi* system)** |
|---|---|---|
| 1. | Jobs cannot be pre-empted. | The farmer must be allowed to irrigate without interruption. |
| 2. | Jobs cannot be serviced simultaneously. | No two farmers abstract water from the supply channel simultaneously. |
| 3. | Idle time between jobs may or may not be permitted i.e. the machine may or may not be available continuously for processing jobs. Idle time is sometimes necessary to process jobs neither early nor tardy i.e. just-in-time. | It would be desirable to provide water to farmers exactly when requested. However in a pure warabandi system this is not usually the case. |
| 4. | Jobs are processed only once. | Each farmer is supplied water only once in a given irrigation interval. |

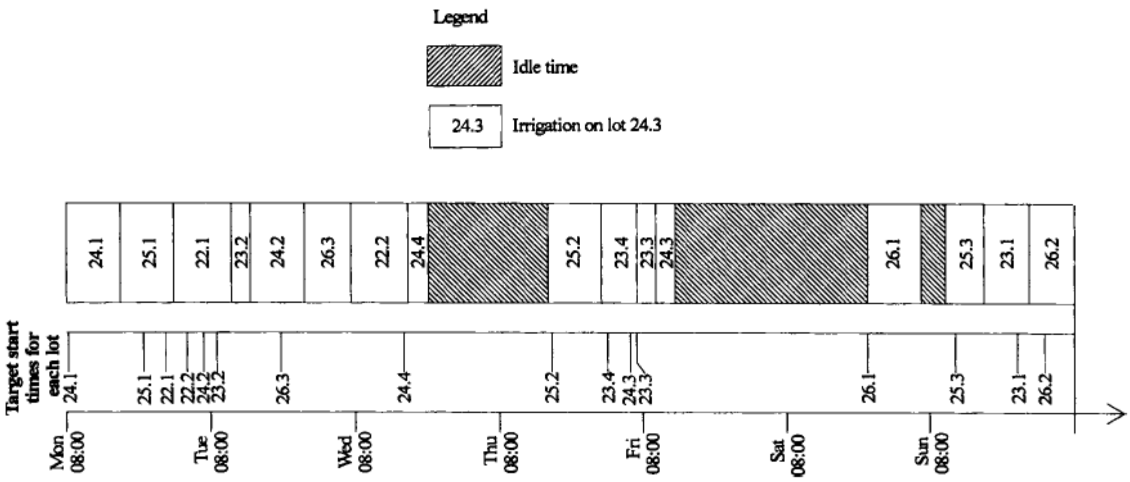



**Figure 7.1** A single machine irrigation schedule (De Vries and Anwar, 2004)

Figure 7.1 shows the sequence of fields (represented by lot number), scheduled to receive irrigation water. The corresponding target start time of irrigation for each field is also indicated. Field represented by lot 24.1 is the first in the sequence to receive irrigation and receive irrigation on its target start time. Lot 26.2 is the last field to receive irrigation and receive irrigation earlier than its target start time. Idle time represents the time when no field is receiving any irrigation. A detailed discussion on idle time is presented in Section 7.2

De Vries and Anwar (2004) applied their integer programming models to four different types of schedules under arranged demand irrigation system:

- Non-contiguous schedules where idle time is allowed between jobs.
- Contiguous schedules where all the idle time is inserted after the last job is finished.
- Contiguous schedules where all the idle time inserted before the start of the first job.
- Contiguous schedules where idle time precedes and /or proceeds all jobs

Integer programming falls under the category of exact algorithms, and as such for problems of higher complexity are not considered a practical tool. A fact widely reported in literature (e.g. Garey et al., 1988; Heady and Zhu, 1998; Sourd, 2005). For example, Anwar and De Vries (2004) reported that for the contiguous single machine schedules with 15 or more jobs, the integer programme was not able to reach the global optimum within the allocated time of three hours. To circumvent this problem, Anwar and De Vries (2004) presented a heuristic solution to the models by De Vries and Anwar (2004). Anwar and De Vries (2004) concluded that heuristics presented a computationally efficient method. However, they also indicated that the solution quality deteriorated when jobs were scheduled non-contiguously or when idle time was permitted both before and after the contiguous jobs. Therefore, they recommended developing better heuristics for the problem. Anwar and De Vries (2004) also argued that new heuristics needs to be tested against other heuristics for problems whose optimum solution are not possible to be found within reasonable computation time with exact algorithms like integer programme. The current research is to pursue these ideas identified in the literature cited above and present alternative and better solutions.

In irrigation scheduling the term irrigation interval is used to describe the time period within which all farmers have to be supplied with water. A comparable term for irrigation interval in OR is the deadline which is different than the due dates mostly used in OR for the desirable completion time of jobs. Baker and Scudder (1990) argued that deadlines must be met and cannot be violated in contrast to due dates which may be violated. This is also compatible with irrigation where the irrigation interval will never be violated but the supply of water i.e. jobs may be either early or tardy. Garey et al. (1988) used the term preferred starting time at which it would be desirable to start processing a job and preferred completion time as an equivalent term for due dates. Although there is no such term as due date used in irrigation, however it may be conveniently replaced by target start time. Target start time plus duration of a certain job becomes the due date for that job. However it is more convenient to use the target start time in irrigation, as the farmers usually place their orders in terms of the start time of their irrigation not completion time.

Heady and Zhu (1998) described the just-in-time (JIT) philosophy to be a popular management concept since its introduction in 1970s. Earliness and tardiness problem (ET) where both early and late jobs are undesirable is one of the key problem areas in JIT implementation. Lauff and Werner (2004) observed that JIT production philosophy led to a growing interest in scheduling problems considering both earliness and tardiness penalties and that a majority of which are devoted to single machine problems. Similarly Heady and Zhu (1998) also concluded that the vast majority of published ET research dealt with sequencing jobs on a single machine.

As discussed earlier there are different ways whereby the distribution of irrigation water can be managed. The better among them are the flexible distribution systems, where an effort is made by the supplier to match the scheduled irrigation start times to the target start times requested by the farmers. It will be more intuitive to judge the suitability of any such schedule by the determination of how close the scheduled start times are to the target start times. This constitutes a typical OR scheduling problem, i.e. sequencing with earliness and tardiness, distinct due dates and a common deadline as described by Baker and Scudder (1990). Baker and Scudder (1990) presented a comprehensive review of the scheduling problems with earliness and tardiness penalties. Baker and Scudder (1990) described the problem of sequencing with earliness and tardiness, distinct due dates and a common deadline, a hard problem to solve and presented a review of the techniques used by researchers for dealing with this problem. Heady and Zhu (1998) described distinct due dates as one of the classes of single machine ET problems and argued that the ET model with the distinct due date assumption intentionally minimizes the sum of job’s earliness and tardiness, and facilitates a feasible delivery schedule. Lee and Choi (1995) described the job scheduling problem with distinct due dates, single machine and general penalty weights for early and tardy jobs. They presented a two step solution to the problem, i.e. a timing algorithm first to determine the optimal start for each job and then a genetic algorithm for determining near optimal sequences with idle time inserted between blocks with contiguous jobs. However, Kanet and Sridharan

(2000) demonstrated how this two step solution i.e. sequencing and scheduling separately, could lead to deterioration in solution. Colin and Quinino (2005) described considering sequencing and scheduling simultaneously with inserted idle time as a new area of research and recommended further research on it. This wealth of information may be very effectively applied to irrigation scheduling which is as demonstrated, not any different from the classical scheduling problems found in OR.

## 7.2 Idle time

In machine scheduling problems idle time insertion can result in a better schedule. Kanet and Sridharan (2000) defined inserted idle time (IIT) schedules as a feasible schedule in which a machine is kept idle when it could begin processing an operation. Colin and Quinino (2005) described idle time insertion necessary in just-in-time (JIT) environments where costs associated with even early completion of jobs are relevant i.e. the performance measure is nonregular. Similarly Baker and Scudder (1990) considered the assumption of no inserted idle time to be inconsistent with JIT philosophy or earliness and tardiness (ET) criteria where jobs are neither allowed to be early nor tardy. Heady and Zhu (1998) also emphasized to take into consideration idle time while earliness is part of the problem objective. They cautioned that an ET solution procedure may fail to find a true solution if idle time is not treated properly.

Kanet and Sridharan (2000) while quoting Conway et al. (1967) described the following situations where it is unnecessary to consider idle time i.e. (1) for single machine problem, (2) with all jobs simultaneously available, and (3) for a regular performance measure i.e. where a later job completion time has no influence on the cost associated to the schedule. Kanet and Sridharan (2000) described seven different machine scheduling scenarios where idle time needs to be inserted i.e.

1. *Single machine, nonidentical ready times, regular performance measure*
2. *Multimachine, identical ready times, regular performance measure*
3. *Single machine, identical ready times, nonregular performance measure*
4. *Multimachine, nonidentical ready times, regular performance measure*
5. *Multimachine, identical ready times, nonregular performance measure*
6. *Single machine, nonidentical ready times, nonregular performance measure*

7. *Multimachine, nonidentical ready times, nonregular performance measure*

It may be concluded that the insertion of idle time is essential in scheduling problems with nonregular performance measure or where both earliness and tardiness costs are considered i.e. both early and tardy jobs are penalised. From irrigation scheduling perspective, idle time can only be inserted in a sequential irrigation system when the sum of all the individual farmers' irrigation (or jobs') durations called makespan in OR is less than the irrigation interval. However if simultaneous application to several farmers is allowed this may not hold true.

If idle time has to be inserted, two scenarios could be imagined. One is that the supply channel is continuously flowing; farmers abstract water as scheduled; and when water is not being used i.e. idle time inserted, it is either drained and/or if possible reused. The other scenario could be to shut the channel each time idle time is inserted or water is not being used. The former may result in wastage of water while the later would result in an excessive number of gate operations. An alternative solution would be to schedule the irrigation water supply contiguously, i.e. when one farmer finishes his turn of irrigation the supply is diverted to the next adjacent farmer and so on (De Vries and Anwar, 2004). There are some implicit assumptions here that either no time is taken by water to travel from one farmer's outlet to another or the travel time is part of each farmer's irrigation duration or is very small and hence negligible. If all jobs are scheduled contiguously the gates are only needed to be opened at the beginning of first farmer irrigation and closed when the last farmer has finished his turn. However the idle time insertion still needs to be addressed. There are three options:

- either to insert all the idle time in the beginning of the schedule,
- or to insert all the idle time at the end of the schedule
- and/or both at the beginning and end of schedule

Although corresponding parallel with OR literature could not be drawn for either of these options; however, examples of jobs scheduled contiguously in blocks or groups and then inserting idle time between different blocks could be found in OR literature e.g. Lee and Choi (1995). The three options discussed in the preceding lines for idle time insertion and also considered by De Vries and Anwar (2004) in their contiguous sequential irrigation models are illustrated in Figure 7.2. The decision to insert idle time or not, or schedule jobs contiguously is dictated by several factors. These include the type of distribution system in vogue, the level of service provided, the total amount of water available, and canal capacities and automation.

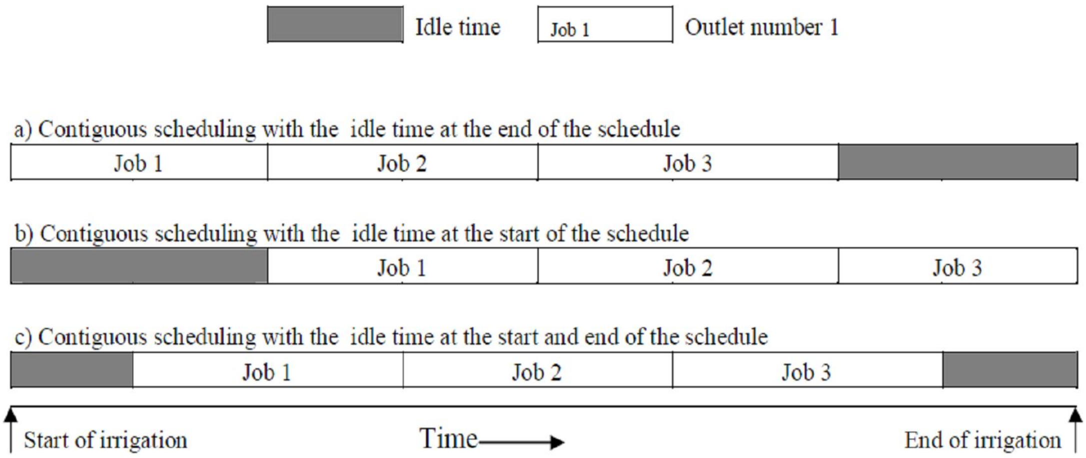



**Figure 7.2** Three types of contiguous single machine irrigation schedules.

## 7.3 Sequence-dependant setup times (travel time)

As mentioned earlier usually some time is required for water to travel from one farmer's field outlet to another farmer's outlet after the first one has completed his irrigation. This travel time obviously depends on the distance between the two outlets and could be significant. Time is also required for outlets opening and closing operations, but this may be insignificant as compared to the duration of irrigation and hence may be easily ignored. The time required for a channel to fill up to operating depth could also be important, depending on the location of the outlets and the way outlets are operated. Figure 7.3 demonstrates the dependency of travel time on the sequence of operation of outlets. Travel time is thus analogous to the sequence-dependent setup in OR.

Allahverdi et al. (1999) while citing another reference defined setup "*to include work to prepare the machine, process or bench for product parts or the cycle*". Allahverdi et al. (1999) described setup as sequence-dependent if the duration of setup depends both on the current and the immediately preceding job. In contrast setup is sequence-independent if the duration of setup depends only on the current job to be processed. Randhawa and Kuo (1997) classified setup time into processor-dependent, product-dependent, and both. They defined processor-dependent to depend only on the processor, regardless of the product type and product-dependent to depend on the production sequence.

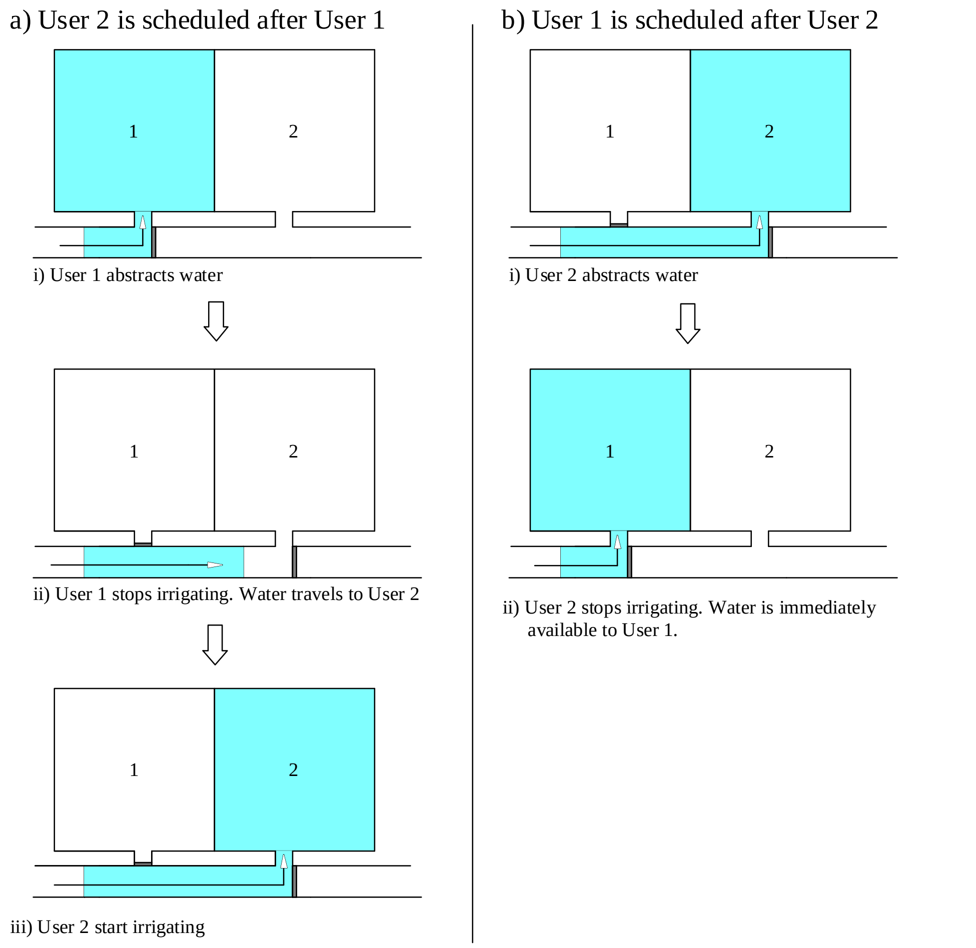



**Figure 7.3** Sequence dependant set-up times (De Vries, 2003)

Sethanan (2001) while citing other researchers stated that there is an enormous amount of research on the flowshop scheduling problem; however research where setup times are sequence-dependent is rare. Similar views have also been expressed by Zhu and Heady (2000). A comprehensive review of scheduling research involving setup considerations can be found in Allahverdi et al. (1999). Sourd (2005) compared different approaches to solve earliness/tardiness problem with setup on a single machine. De Vries and Anwar (2006) presented a mixed integer linear programming model for irrigation scheduling with travel time (setup) considerations and demonstrated the importance of considering travel time while making irrigation schedules. However to find a solution within an acceptable time period they recommended using heuristics or GA for such kind of computationally demanding problems. The current research is to fully explore this avenue.

## 7.4 Irrigation scheduling as a Multimachine problem

Depending upon the situation, irrigation water may be supplied to farmers in a tertiary unit either sequentially, turn by turn, or simultaneously to several farmers. It has already been demonstrated that supplying water sequentially to farmers constitutes a single machine problem. Similarly, simultaneous supply of irrigation water to several farmers from the same supply channel may be described as a multimachine scheduling problem found in OR. A multimachine irrigation schedule is presented in Figure 7.4. Suryavanshi and Reddy (1986) for the first introduced the concept of stream tubes. They considered the supply channel to consist of a number of imaginary and equal discharge stream tubes. A stream tube would supply outlets sequentially, one outlet at a time but not simultaneously and that the discharge of each stream tube would be equal to the discharge of the outlet to be serviced. If at a certain point in time more than one stream tube is operational i.e. servicing different outlets, the situation may be described as a multimachine problem.

Suryavanshi and Reddy (1986) formulated an integer programme for sequencing irrigation outlets with the objective of minimizing the channel capacity thereby reducing the cost of construction. However, the model by Suryavanshi and Reddy (1986) is incorrectly formulated and does not minimize the number of stream tubes operating simultaneously; rather the model minimizes the total number of stream tubes. For example, eight outlets requiring 30 L/s could be supplied by one stream tube of 30 L/s feeding each outlet sequentially. Alternatively, there could be two stream tubes each of 30 L/s capacity operating simultaneously; the first stream tube supplying four of the eight outlets, and the second stream tube supplying the remaining four, or eight stream tubes each of 30 L/s operating simultaneously (if there is no supply limitation) each stream tube supplying one outlet. The Suryavanshi and Reddy (1986) model does not distinguish between these cases since the objective function is identical in all cases and would result in a value of "8" in all cases, if all other factors are kept equal to unity. Wang et al. (1995) corrected this shortcoming in Suryavanshi and Reddy (1986) model and presented an improved formulation by introducing a tube activation function. For a given stream tube, the activation function assumes a value of one if that stream tube feeds one or several outlets. If the stream tube does not feed any outlets, the activation function takes a value of zero. The objective function effectively minimizes the number of stream tubes operating simultaneously by minimizing the sum of the activation function values.

Reddy et al. (1999) formulated the scheduling of irrigation canal outlets with different rates of discharge and durations as an integer programme problem by introducing the concept of a time window whereby an outlet can only be operated within this time window. The distinction of the model by Reddy et al. (1999) is that it introduced the concept of scheduled start time for outlets and the approach adopted for solution was based on time blocks rather than imaginary stream tubes. However, the model developed by Reddy et al. (1999) only minimizes the difference between the actual capacity of the main supply canal and the required capacity of the main canal. It does not completely prevent required capacity to exceed actual capacity, i.e. it is possible that the model finds a schedule where the required capacity of the main canal is more than its actual capacity. The time block approach is distinct from the stream tube approach, although they both aim to solve the same type of problem, a view also expressed by Anwar and Clarke (2001). In contrast to the stream tube approach, the time block approach does not consider imaginary stream tubes, rather the irrigation interval is divided into a number of time blocks and the number of outlets serviced in each time block is recorded. In time block approach the main canal capacity is expressed as the maximum number of outlets operated in any time block while in stream tube approach it is the total number of distinct stream tubes or machines utilised. However, as also mentioned by De Vries and Anwar (2006) both these earlier models do not directly produce schedules for water distribution at a tertiary level rather address capacity constraints at the main supply canal.

Anwar and Clarke (2001) further developed the stream tube model using mixed-integer linear programming. The objective was to minimize the number of stream tubes (hence channel discharge) and at the same time schedule the delivery of water to each outlet as close as possible to the time requested. The model by Anwar and Clarke (2001) give priority to the goal of discharge minimization over earliness and tardiness. Anwar and Clarke (2001) applied their model using data published by Suryavanshi and Reddy (1986) for Distributary Number 3 of the Meena Branch of Kukadi Project, Maharashtra, India. A schematic of the tertiary unit served by the Distributary Number 3 of the Meena Branch is presented in Figure 7.4. In the data by Suryavanshi and Reddy (1986), the outlets do not have any target start times, therefore Anwar and Clarke (2001) generated random target start times to complete the input data required by their model. The model by Anwar and Clarke (2001) was able to obtain an optimum schedule for this practical problem with eight users. The optimum schedule obtained by Anwar and Clarke (2001) has a total discharge requirement of 90 L/s, allowing 3 users to irrigate simultaneously each having a discharge of 30 L/s. The total earliness/tardiness of the schedule is 4.73 days over 6 days of the irrigation interval. This optimum schedule is presented in Figure 7.5, where each block represents the actual start time and duration of each outlet. The target start time for each outlet has also been represented on Figure 7.5. Figure 7.5 shows that at any point in time during the six days irrigation interval the number of outlets receiving water simultaneously is not more than three. This means that the schedule requires a total discharge of three multiplied by the individual discharge of an outlet. For the current example all the outlets have identical discharge requirement which is 30 L/s; hence, the total discharge requirement of the tertiary unit supplied by Distributary Number 3 of the Meena Branch of Kukadi Project (Figure 7.4), for the six days irrigation interval is 90 L/s.

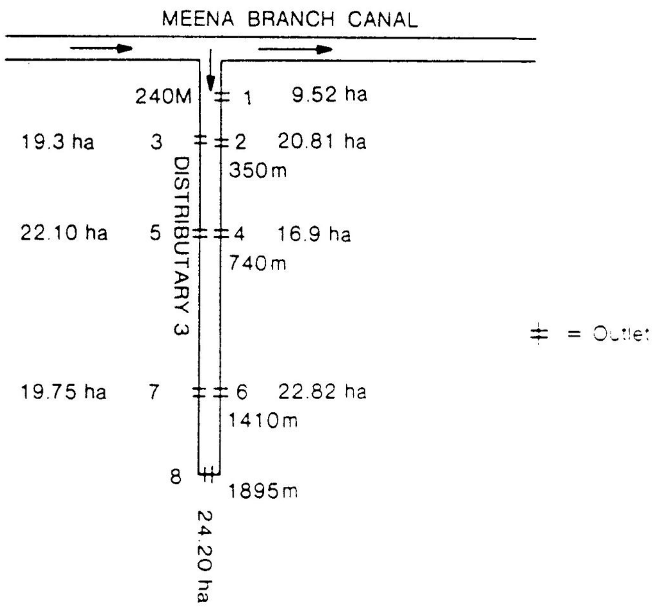



**Figure 7.4** Tertiary unit, Kukadi Project, India (Suryavanshi and Reddy, 1986)

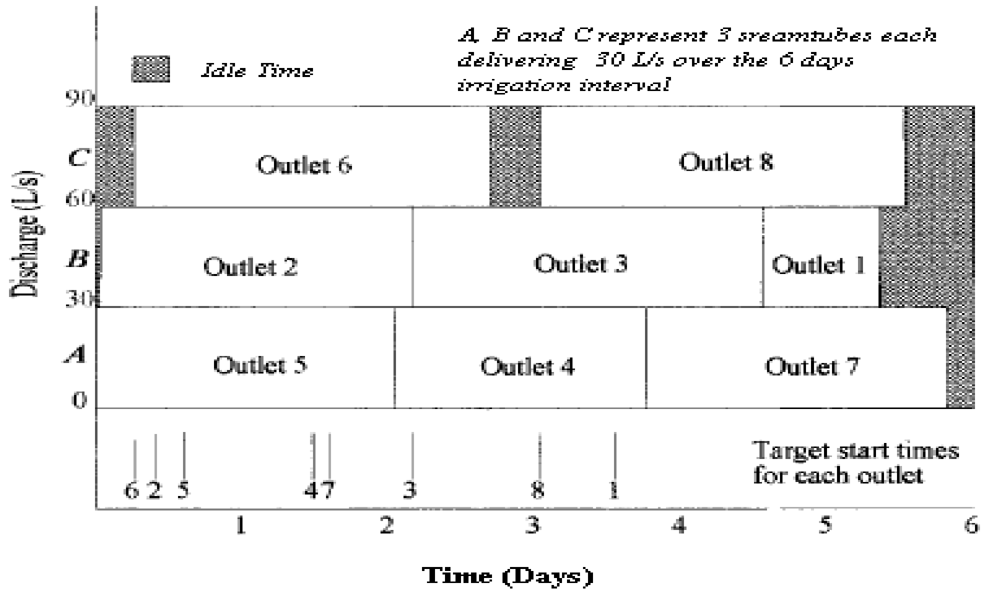



**Figure 7.5** A multimachine irrigation schedule by Anwar and Clarke (2001)

The model by Anwar and Clarke (2001) was the first OR tool, applied to irrigation scheduling which incorporated farmers' requested or preferred irrigation starting time into the schedule more explicitly. However, as described earlier and also by De Vries and Anwar (2006), for hard optimization problems like machine scheduling problems, larger problems (jobs equal to or greater than 15) may require excessive solution times using exact algorithms like the integer programme. Approximate algorithms or heuristics e.g. GA, are considered the appropriate choice for such problems. Similar views are also found in abundance in OR literature (e.g. Heady and Zhu, 1998). Heady and Zhu (1998) described multimachine scheduling a hard problem to solve, which for large scale problems requires a heuristic procedure for its solution. The importance of exact algorithms like integer programme, however, can still not be denied. Exact algorithms may serve as a benchmark to test the solution quality of these approximate algorithms, though for a small problem size. Heady and Zhu (1998) also observed that the available published multimachine ET literature is scarce compared to single machine ET problems and that the majority of multimachine ET studies are actually an extension of a single machine problem.

Wardlaw and Bhaktikul (2004) represented the lateral canal scheduling problem by Reddy et al. (1999) and by Anwar and Clarke (2001), using the time block approach and genetic algorithms. In application to Reddy et al. (1999) example, Wardlaw and Bhaktikul (2004) used two approaches:

- Reddy et al. (1999) approach, in which the range of starting time blocks for each lateral was prespecified, and
- Wardlaw and Bhaktikul (2004) own approach, in which the range of starting time blocks for each lateral was unconstrained.

Wardlaw and Bhaktikul (2004) claimed superiority of their GA formulation in both approaches over Reddy et al. (1999) integer programming formulation. Wardlaw and Bhaktikul (2004) also applied GA to the problem presented by Anwar and Clarke (2001) and again claimed better solution quality and faster execution time than the integer program by Anwar and Clarke (2001). However some shortcomings are identified in the formulation by Wardlaw and Bhaktikul (2004). They are: for feasible schedules, their formulation does not minimize stream tubes i.e. channel capacity and earliness/tardiness but rather it minimizes earliness/tardiness only; their formulation does not avoid infeasible solutions effectively and may adjudge infeasible solutions better than a feasible but costly solutions (Haq et al., 2008)[1]. Haq et al. (2008) emphasize thorough testing of heuristics such as GA, before any conclusion about their performance could be drawn.

The earlier work found in literature and discussed in the preceding paragraphs, suggests that two distinct approaches for dealing with the problem of simultaneous irrigation exist. They are the stream tube approach and the time block approach. However, these approaches have not been compared in the past to evaluate the suitability of either for irrigation scheduling problems. Exploring these approaches constitutes a major part of the current research. For the purpose of this thesis two classes of multimachine scheduling problems are differentiated. The same is also adopted by De Vries (2003). They are:

- the simple multimachine scheduling;
- the complex multimachine scheduling.

[1] Haq et al. (2008) is the first paper from this thesis, published in ASCE Journal of Irrigation and Drainage Engineering. Copy of the paper can be found in Appendix A.

In simple multimachine all the outlets have identical discharges i.e. the same number of stream tubes are servicing each outlet (e.g. Anwar and Clark, 2001). In complex multimachine the discharges of the outlets are not identical. The selection of an approximate algorithm for solutions to the scheduling problem is, however, a difficult job. There are a number of choices available. A detailed discussion is therefore presented in Section 7.6 and 7.7.

## 7.5 Solution methods

In OR literature (e.g. Garey et al., 1988; Heady and Zhu, 1998; Sourd, 2005) it is a well established fact that the single machine and multimachine ET scheduling problems, even without the addition of any further complexities, are very hard optimization problems i.e. NP-hard. Jain and Meeran (1999) defined NP-hard as problems which require computation time that increases exponentially with the problem size. The literature (e.g. Blum and Roli, 2003; Jensen, 2001; Wall, 1996 etc.) suggests that no polynomial time algorithm exists for such problems and an exact algorithm might require exponential computation time which often leads to a computation time too high for practical purposes. Approximate algorithms are resorted to for these problems. In approximate algorithms a sacrifice has to be made of a guaranteed optimal solution in favour of a near optimum solution, with reasonably less computation time (Blum and Roli, 2003). The optimization problems that are being considered in this research are one of the broad classes of optimization, known as combinatorial optimization (CO). Blum and Roli (2003) defined CO as an optimization problem where the search space is discrete. They also described metaheuristics as successful algorithmic concepts to generate approximate solutions to NP-hard combinatorial optimization problems. Higgins and Wirth (1995) estimated that about 90% of scheduling problems are NP-hard and described their exponential time behavior as the "*curse of dimensionality*".There are many examples of approximate algorithms in literature for tackling such problems. There are only few, however, which stand out among the crowd. They are the important types of metaheuristics i.e. genetic algorithm (GA) and simulated annealing (SA). De Jong and Spears (1989) described GA and SA (a class of neural networks) as powerful and general problem solving methods.  Similarly, Arostegui Jr. et al. (2006) described tabu search (TS), SA, and GA as the most well known general heuristics methods and also noted that there are only a few studies that compare these heuristics. Tsang (1995) described it extremely difficult to choose between various optimization techniques for a specific problem and believed any kind of generalization next to impossible. Zolfaghari et al. (2002) described the selection of a good search method as “*a non trivial issue*”. Kim and Kim (1996) noted that both GA and SA are significantly affected by the choice of parameters and recommended further research into the methodology of applying both GA and SA to different problems. However there are some inherent advantages and disadvantages associated with both GA and SA that may make them appropriate or inappropriate for different problems under different circumstances.

## 7.6 GA vs. SA

Some of the most commonly mentioned advantages associated with GA are: global search ability (Min, et. al., 2006; Montana, et. al., 1998; Tsang, 1995), flexibility in adaptation to a wide range of problems (Montana, et. al., 1998), versatility, power and hybridizability (Lucasius, et. al., 1994), maintaining a population of solutions rather than a unique solution and consistency in reporting better solutions (Damodaran, et. al., 2006; Lee, et. al. 1996; Lucasius, et. al., 1994). Also Kimms (1999) described GA as the most popular heuristic approach for optimization. Although it is not a panacea, it has the potential to handle complex, large-scale problems (Lucasius, et. al., 1993). However, there are some shortcomings of GA as well. Man et al. (1996) described deception, genetic drift and randomness as some of the inherent shortcomings of GA, while Lucasius et al. (1994) described poor accessibility, mechanical complexity and search imprecision as the main shortcomings. Min and Cheng (2006) described GA as relatively weaker in local search capability and hence recommended hybridization for improvement. Dealing with codes rather than directly with parameters is described as an advantage in most of the literature because it makes it more domain-independent. However, Hwang and He (2006) considered it responsible for increase in computational burden and a waste of time (in coding and decoding processes).

On the other side, simplicity, ease of implementation and high solution quality are the main advantages affixed to SA (Brown, et. al., 1992; Radhakrishnan and Ventura, 2000). Also, SA has been termed flexible in computation time and useful for both optimization and constraints satisfaction when near optimal solutions are acceptable (Tsang, 1995). Monem and Namdarian (2005) considered SA well suited for several decision variables with different natures, and termed it insensitive or completely independent of the number of decision variables, constraints and objective functions. Hwang and He (2006) considered SA to be very powerful in solving combinatorial problems and very good at hill climbing for the optimum solutions. Yagiura and Ibaraki (1996) have recommended using SA if higher solution quality is important. However, it is also evident from literature that SA needs more computational time than GA in some applications (Arostegui, et. al. 2006; Sadegheih, 2006) though, in case of Kim and Kim (1996); Suman and Kumar (2006); Brown et. al. (1992) the results were completely opposite. According to Hwang and He (2006) the slow speed of SA is due to the random processes to search the minimum energy state while Suman and Kumar (2006) experienced less CPU time than GA and described the point by point iteration rather search over the whole population as the reason for this advantage. Kuo et al. (2003) have concluded that in their application, SA performed as well as GA and that both methods could be applied to even more complicated water resource management problems.

In light of the above argument it could be stated that the selection between GA and SA depends largely on the type of problem, its representation and most importantly the ingenuity of the practitioner. It also seems like GA feels more intimidating and complex compared with SA to the novice practitioner. However, there is no evidence to date which describes either of them as a complete failure under any circumstances or an absolute success. In this thesis, genetic algorithms are applied, because they are known to be more robust in finding the global optimum (Min, et. al., 2006; Montana, et. al., 1998; Tsang, 1995),; they have the broadest field of applications (Montana, et. al., 1998; Davis, 1991; Goldberg, 1989); they have been applied to machine scheduling problems successfully(Davis, 1991; Goldberg, 1989); they support a wide range of problem representation schemes which make them easily adapted to real world problems(Cheng et al., 1996); and a huge amount of literature and software support is available(e.g. Medaglia and Gutiérrez, 2006a; Medaglia and Gutiérrez, 2006b).

## 7.7 Genetic algorithms (GA)

Coley (1999) described GA to be invented by John Holland in 1960's. Goldberg (1989) described Holland (1975) to be the primary monograph on GA. Thereafter a series of literature and reports became available. Good introductions to genetic algorithms can be found in Coley (1999), Davis (1991), and Goldberg (1989) etc. Whitley (1994) differentiated between two definitions of GA. He defined GA, in strict sense, to be the one that refers only to the models introduced by John Holland and his students. In a broader sense he defined GA as any population-based model that uses selection and recombination operators to generate new sample points in a search space. In the latter case the focus is more application oriented and GA is used mostly as an optimization tool. The present study is one such example.

GA mimics some of the processes observed in nature i.e. natural selection, which is based on the principle of the survival of the fittest. In natural selection stronger individuals are likely to be the winners in a competing environment. GA uses a direct analogy with such form of natural evolution. In GA the potential solution of a problem is an individual and can be represented by a set of parameters, regarded as genes of a chromosome. The chromosome (a candidate solution to a problem) may be represented by a number of ways. The most basic is the binary form. Jensen (2001) noted that since GA is inspired by the principles of genetics, genetic algorithms place a stronger emphasis on the distinction between the genetic representation of an individual (the genotype) and the actual expression of the individual (the phenotype) than evolution strategies. Each chromosome that represents a solution to the problem is assigned a value that determines the goodness of the chromosome for solving the problem and is closely related to its objective value. A population of chromosomes is generated, usually randomly. There are also other methods available for creating an initial population. The size of the population depends on the nature of the problem. (Man et al., 1996)

An implementation of a GA begins with a population of chromosomes. Chromosomes are then evaluated and through a reproductive mechanism, those chromosomes which represent a better solution to the target problem are given more chances to reproduce than those which are poorer solutions. The goodness of the chromosome is relative to the current population. Whitley (1994) distinguished between the two terms, objective function and fitness function, which is sometimes used interchangeably in GA literature. Objective function was defined as a measure of performance with respect to a particular set of parameters and fitness function as a transformation of that measure of performance into an allocation of reproductive opportunities. The objective function of a string representing a set of parameters is independent of the evaluation of any other string. However, the fitness of that string is always defined with respect to other members of the current population. Fitness function is a measure to differentiate between solutions.

In each cycle of the genetic operation, termed an evolving process, a new generation is created from the chromosomes in the current population by selecting a pool of fitter chromosomes called parents and then cross mating them according to a certain criteria. According to this criteria the genes of the parents are to be mixed and recombined for the production of children chromosomes in the next generation. The process is repeated in a hope that with each evolution better and better chromosomes will be accumulated until a desired termination criteria is reached. This criteria may be based on either of these conditions: there is no further improvement in the fitness values of the individuals (termed as convergence); or a certain fixed number of generations (evolution cycles) is reached; or a certain predefined percentage of the amount of variation of individuals between different generations is reached; or a predefined value of fitness is achieved. There is no standard flow chart in applications of GA (Lucasius and Kateman, 1993); however a simple and widely adopted GA methodology may be described as follows and presented diagrammatically as in Figure 7.6.

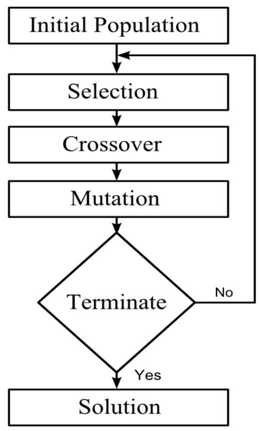



**Figure 7.6** A simple GA flow chart

1. **Initial population:** As stated earlier a GA implementation starts with a population of chromosomes representing solutions to the problem at hand. So the first question that needs to be answered is how to represent a candidate solution. A variety of representation schemes is available in literature. The one that suits a specific problem is a non-trivial issue and a separate discussion on the issue is to be presented in a separate section to follow. After a

proper representation is selected for the chromosome, the next step is to generate a population of chromosomes of some suitable size. Whitely (1994) stated that the size of the population varies from problem to problem, however he recommended a reference for some general guidelines. Lucasius and Kateman (1993) described a range of 50 to 500, as the commonly used population size in most practical applications. Davis (1991) stated that there is no simple answer to the question of population size and that it depends largely on the problem being solved, the representation used, and the operators manipulating the representation. He further elaborated that the question of, "*will the best of a number of short runs be better than the best of a longer run?*" requires experimentation to answer. However, Damodaran et al. (2006) while citing other references observed that several researchers have proposed to use an initial population of twice the number of jobs. Another important issue is the generation of population. In many GA applications, a population is generated completely randomly, however there are examples where a population is generated heuristically. In its simple forms GA always uses random generation of initial population, as is the case in the present study.

2. **Selection:** Coley (1999) described selection as a GA operator that applies pressure upon the population in a manner similar to that of natural selection found in biological systems where fitter individuals have a greater than average chance of promoting the information they contain within the next generation. Selection comes into play after an evaluation of all individuals in the population is carried out and their fitness determined. Several selection approaches are available. The most common is fitness-proportional or roulette wheel selection where the probability of selection is proportional to an individual's fitness (Coley, 1999). An even simpler approach will be to select a certain number of just the top best.

3. **Crossover:** Crossover or the recombination operator is perhaps the most important fundamental GA operator. In crossover two parents are combined to produce offspring. This combination is usually performed by taking part of the genotype of each parent, and combining the two parts to obtain a new genotype sharing characteristics of both parents. Single point crossover, two point crossover and uniform crossover are the most commonly used crossover operators found in literature. In single point crossover the exchange or recombination of genes occurs about a single point chosen randomly in both parents. In two point crossover two such points are selected and the segment of chromosome between these points is exchanged. The pair of individuals (parents) in crossover operation is selected with a certain probability typically in the range of 0.4 to 0.9 (Coley, 1999). Uniform crossover operates on individual genes with each gene being considered in turn for possible crossover.

A certain probability may also be applied to the exchange of genes to control or moderate disruption (Coley, 1999). Since there is no way of knowing which part of each genotype is good and which is bad, the combination of parents can only take place in a random fashion, thus the recombination of two good parents can also lead to the combination of two bad parts of the genotype. As a result a feasible solution may become infeasible after crossover. Infeasibility will be discussed in a separate section to follow.

4. **Mutation:** Mutation is applied to each child individually after crossover. It randomly alters each gene with a small probability typically 0.001 (Beasley et al., 1993a). Mutation operators are used in genetic algorithms to make sure, genetic material lost early in the search process can be reintroduced later. This is necessary, since usually crossover cannot introduce new genetic material it merely recombines material already present in the population. Thus, without a mutation operator genetic material not present in the population can never be introduced. Davis (1991) has pointed out that mutation becomes more productive and crossover less productive as the population progresses.

5. **Termination:** If the GA has been correctly implemented, the population will evolve over successive generations. When a satisfactory solution is reached or the population is dominated by good chromosomes with optimum or near optimum value, termination of the evolution cycle is enforced. Convergence is a term used frequently in such occasions in GA literature. Beasley et al. (1993a) described convergence as the progression towards increasing uniformity. As the population converges, the average fitness of the population will approach that of the best individual.

# 8 Aim and objectives

## 8.1 Problem description

Irrigation delivery scheduling is the procedure to establish a roster of irrigation turns or water applications for a specific period of time, for example an irrigation (or crop) season. Water delivery to farmers (rate, duration and frequency) is to a large extent dependent on the infrastructure/technologies and irrigation scheduling system. However, as described by Renault et al. (2007) the primary goal of any irrigation system is to deliver irrigation water to farmers according to an acceptable level of service that is well adapted to their requirements for water use and cropping systems. The level of service could be assessed by considering different indicators e.g. adequacy, flexibility and reliability etc. However for the purpose of this thesis the level of service is the determination of how close the supplies are matched with demands, i.e. the scheduled irrigation start times are matched with the target irrigation start times. In flexible irrigation systems like on-demand the main objective is to achieve the highest possible level of service. This is in line with the concept of Service Oriented Management (SOM), introduced by Renault et al. (2007) for irrigation management. They described SOM as the central goal of irrigation modernization. However with rigid delivery schedules like *warabandi* it is difficult to achieve any reasonable level of service. Some degree of flexibility is possible with modified frequency rotation, modified amount and continuous flow systems as described in Section 3.

Whatever the system of irrigation and level of service may be, before delivering water to the farmers, a quick way of preparing an irrigation schedule is essential. A schedule which represents the best possible sequence of irrigation turns and match farmers' requested target start time as close as possible, under the given conditions. If only a few farmers are the target users then it may be possible to do this job manually very easily. But if the number of farmers is large and there are a variety of restrictions then it may become a highly complex problem. Some pioneering work has been conducted by a handful of researchers on these lines; however, as reviewed in Section 7, there are some limitations and shortcomings of the previous work. The purpose of the current research is to improve and advance that work by overcoming those limitations and shortcomings.

## 8.2 Aim of research

The importance of effective irrigation water management and its contribution to world poverty alleviation has been discussed in the earlier sections 1 and 2. Just to reinforce that discussion a couple of citations are presented, that also highlight the importance of the current research. Kirpich et al. (1999) quoted a former president of the International Commission for Irrigation and Drainage (ICID), John Hennessy, in a keynote address in 1992 as: ‘‘*Irrigation schemes in many parts of the world are known to be performing well below their full potential ... [There is now] wide recognition that deficiencies in management and related institutional problems, rather than the technology of irrigation, were the chief constraints of poor performance of irrigation systems*.’’ Similarly Renault et al. (2007) described water resources management, service to irrigated agriculture and cost-effectiveness of infrastructure management as the key areas that critically needed improvement. Flexible irrigation systems as discussed in detail in Section 3 are largely considered efficient irrigation systems. The current research contributes towards effective irrigation water management by developing decision support tools for flexible irrigation systems.

The main focus of the research is to schedule irrigation water deliveries to farmers at a tertiary unit, as close as possible to their requested irrigation start times. To do so the concept of classical machine scheduling in OR is utilized to develop a series of irrigation scheduling models. The purpose is to equip the irrigation managers with an optimization tool to schedule irrigation deliveries optimally and efficiently. Both single machine (where farmers irrigate sequentially) and multimachine (where farmers irrigate simultaneously) scheduling problems with earliness/tardiness penalties will be studied, taking into consideration idle time and sequence-dependent setup time. Related literature (Section 7) suggests that the models presented by earlier researchers are either, computationally very demanding and hence are not practical for large size real world problems e.g. Anwar and Clarke (2001), De Vries and Anwar (2004), De Vries and Anwar (2006); or the models are not robust enough to be relied upon e.g. Anwar and De Vries (2004); or they are incorrectly formulated e.g. Wardlaw and Bhaktikul (2004). The models to be developed in the current research are intended to overcome these limitations and shortcomings and evaluate different approaches for solutions which have not been done in the past to the best knowledge of the author. Computational experiments will also be carried out to test the sensitivity of the models to different problem specific parameters.

## 8.3 Hypothesis

"A Genetic algorithm (GA) is a computationally efficient and robust optimization tool that can provide good quality solutions for an irrigation scheduling problem".

## 8.4 Objectives

To fully explore the potential of GA for the irrigation scheduling problems the following objectives are set forth:

1. To develop a series of single machine irrigation scheduling models with earliness/tardiness penalties and idle time insertion for sequential or *warabandi* irrigation systems. Four scenarios for idle time will be considered: 1) idle time inserted between jobs i.e. non-contiguous scheduling of the irrigation turns 2) all the idle time inserted after the last job is finished, allowing contiguous irrigation and 3) all the idle time inserted before the start of the first job and the irrigation turns are scheduled contiguously 4) idle time precedes and /or proceeds all jobs, and again irrigation turns are scheduled contiguously.

2. (a) To develop a simple multi-machine irrigation scheduling model allowing simultaneous irrigation, with earliness/tardiness penalties and equal discharge for all farmers.
   (b) Present and compare different approaches in developing these models.
   (c) To include sequence-dependent setup or travel times in these models which though increase the complexity of the models, however, increase the utility of the models as well. The incorporation of travel time is not a trivial job. This makes the model very difficult to formulate and implement.

3. (a) To develop a complex multi-machine irrigation scheduling model allowing the farmers to demand different discharges. By allowing the farmers to demand different discharges makes the model more flexible, however, computationally more demanding and complex.

(b) To include sequence-dependent setup or travel times in these models.

4. To determine the sensitivity of the models to some problem specific parameters.

# 9 Methodology

A series of single and multimachine scheduling models will be developed using GA for achieving the objectives set forth in Section 8. The multimachine models have more emphasis in this thesis because they have the dual goal objective of minimizing machines (discharge) as well as earliness/tardiness. Thus the dual goal objective multimachine models may be applied to the non-contiguous single machine scheduling problems where idle time is allowed to be inserted between jobs. The multimachine models handle the non-contiguous single machine problems by minimizing the number of machines to a single machine. The multimachine models to be presented in this study, however, are not applicable to the single machine scheduling problems where idle time is not allowed to be inserted between jobs or jobs are processed contiguously.

In the contiguous single machine models a farmer receives water immediately after the preceding one has finished his turn of irrigation. The main supply channel gate is opened at the beginning of the irrigation and closed when the irrigation is complete. Such irrigation systems, as discussed in Section 3 and 7, are known as *warabandi*. *Warabandi* is widely practiced in the subcontinent, particularly in Pakistan, India and a number of other countries. Therefore, keeping in view this wide acceptance of the *warabandi* system, a series of contiguous single machine models will be developed that are only applicable to *warabandi* systems. In this regard three different contiguous single machine models will be developed. The model developed will be considering three different management options regarding idle time insertion i.e. all the idle time inserted at end of irrigation, all the idle time inserted at the beginning of irrigation, and idle time inserted both at the beginning and/or end of irrigation. These three models are illustrated in Figure 7.2. Formulations and implementation details of the models are presented in Section 10.

Two types of the multimachine models will be developed, i.e. the simple multimachine models and the complex multimachine models. The simple multimachine models apply to situations where all the farmers are restricted to receive the same discharge. Two different approaches in the development of these models i.e. the time block approach and the stream tube approach are fully explored. These simple multimachine models are further augmented by incorporating travel time. Another enhancement over the simple multimachine model is the complex multimachine model. The complex multimachine model allows the farmers to have different discharges. The complex multimachine model is further augmented by incorporating travel times. Formulations and implementation details are presented in Section 11. Table 9.1 describes all the models to be developed in this study.

**Table 9.1** Different types of models to be developed

| Model type | | | Description |
|---|---|---|---|
| **Sequential irrigation scheduling models** | 1. | Model 1 | Non-contiguous single machine model which allows idle time to be inserted between jobs. |
| | 2. | Model 2a | Contiguous single machine model with all idle time at the end of irrigation interval |
| | 3. | Model 2b | Contiguous single machine model with all idle time at the start of irrigation interval |
| | 4. | Model 2c | Contiguous single machine model with all idle time at the start and/ or the end of irrigation interval |
| **Simultaneous irrigation scheduling models** | 5. | Simple multimachine model | Simultaneous irrigation scheduling model with identical discharges for all users, using both Steam tube, and Time block approaches. |
| | 6. | Simple multimachine model with setup | Simultaneous irrigation scheduling model with identical discharges and travel consideration using stream tube approach only. |
| | 7. | Complex multimachine model | Simultaneous irrigation scheduling model with non-identical discharges for all users using time block approach only. |
| | 8. | Complex multimachine model with setup | Simultaneous irrigation scheduling model with non-identical discharges and travel consideration using stream tube approach only. |

The generic features of GA have been discussed at length in Section 7.7. GA however, requires appropriate selection and formulation of its components from a wide range of available choices for each individual application. The first and the most important among them is the selection of appropriate representation scheme (chromosome). A detailed discussion of the representation schemes is therefore presented separately in Section 9.1. Another important decision that is required to be made is the selection of appropriate, problem specific criteria for dealing with infeasible solutions and then the formulation of the fitness function accordingly. A wide range of techniques for controlling infeasibility is available; therefore a detailed discussion is presented in Section 9.2. The selection of the remaining parameters and operators will be discussed for each model when its formulation and implementation will be presented. After the formulation and implementation details of a model is decided, an appropriate experimentation strategy is required to evaluate the performance of the model. Such strategies are thus discussed in detail in Section 9.3.

## 9.1 Genetic representation

Before a GA can be run, a suitable encoding/representation for the problem must be devised. Which representation is best, greatly depends on the requirements and constraints of the problem to be solved. Among other factors, Ruiz et al. (2006) described encoding as an important factor greatly affecting the effectiveness of GA. Versatility and configurational flexibility of GA allows any selection of representation considered useful or convenient for the problem concerned. However as stated by Lucasius and Kateman (1994), there is no unique answer to the question which problem representation is best. The issue of selecting a suitable representation is not trivial, a view strongly supported by Whitely et al. (1997):

*"Choosing a good representation is a vital component of solving any search problem. However, choosing a good representation for a problem is as difficult as choosing a good search algorithm for a problem."* (Whitely et al., 1997)

Man et al. (1996) described bit string representation (i.e. binary representation) as the most classical approach used by GA researchers because of its simplicity and traceability. This simple representation has an appeal, and also the theoretical grounding of GA (i.e. schema theorem) is based on binary representation (Li et al. 1998). However in some situations the binary representation may result in a chromosome too large to effectively handle and/or may not adequately represent the problem. But GA is not limited to binary representation only. There are numerous examples of other encoding techniques used in a wide range of GA applications e.g. Li et al. (1998); Davis (1991) and Goldberg (1989) etc. have used real number representation for some applications where it deemed appropriate to replace the traditional binary representation.

A direct conclusion from literature review is that any representation that best describes the problem at hand and produce the best possible solution may be used. Though, the pure GA researchers still believe in sticking to the fundamentals i.e. binary representation, random initialization of the population and the normal operators as stated by Davis (1991):

*"One's feeling for and against binary encoding can be very strong. Some researchers refer to binary genetic algorithms as "real" genetic algorithms and leave unspoken their characterization of the rest"* (Davis, 1991).

A comprehensive review of a number of representation techniques for job-shop scheduling problem is given by Cheng et al., (1996). They described the following list of different representation schemes:

- *Operation -based representation*
- *Job-based representation*
- *Preference list-based representation*
- *Job pair relationship-based representation*
- *Priority rule-based representation*
- *Disjunctive graph-based representation*
- *Completion time-based representation*
- *Machine-based representation*
- *Random keys representation*

Since the literature is inconclusive about representation and initialization issue, an integer representation scheme is proposed for the GA models to be developed in this study based on the recommendation in the relevant literature e.g. Cheng et al., (1996); Li et al. (1998). For the single machine irrigation non-contiguous scheduling models, a chromosome contains random positive integer values representing the scheduled start times of jobs. For contiguous models it is a permutation of integers representing jobs. A chromosome inspired by the representation scheme given in Min and Cheng (2006), Montana et al. (1998) and Tamaki et al. (1999), is used for the multimachine models. The chromosome for the multimachine models is a concatenation of two row vectors of randomly generated integers. The first row vector (machine vector) represents machines utilized. The other row vector (scheduled start time vector) represents the scheduled start times of jobs. The corresponding positions of genes in both sections provide information about the assignment of jobs to machines.

For example, in a four machines-four jobs scenario, a schedule may be represented by the chromosome [2 1 4 2] [300 300 200 100]. The machine vector [2 1 4 2] represents the three machines used in the schedule i.e. machine 1, 2 and 4. The first and the last elements in the machine vector (i.e. 2) are identical indicating that the first and last jobs both have been assigned to machine 2. The second and third jobs have been assigned to machine 1 and 4 respectively. The second row vector (scheduled start time vector) also contains four numbers i.e. 300, 300, 200, 100 representing scheduled start times of the four jobs respectively. The first job starts at 300, the second also starts at 300, the third job starts at 200 and the fourth job starts at 100. Thus the chromosome adequately describes a multi-machine scheduling problem with all the required parameters, to be optimized, fully incorporated into it. This chromosomal representation is, however, used only in the stream tube approach. For the time block approach the chromosomal representation is similar to that of the non-contiguous single machine model and the second vector that represents the scheduled start time of jobs of the multimachine model. The chromosomal representation for each model will be further elaborated when each model is individually presented.

## 9.2 Infeasibility

The chromosomal representation of a solution is an important design feature of a GA. However, while running GA quite often infeasible solutions are generated either in the randomly generated initial population or in the subsequent generations by manipulation via genetic operators e.g. crossover and or mutation. Cheng et al. (1996) defined feasibility of a chromosome as the phenomenon of whether or not a solution decoded from a chromosome lied in the feasible region of a given problem. They also differentiated between infeasibility and illegality of a chromosome and described a chromosome illegal if it did not represent the solution space at all. The problem of maintaining feasibility has been addressed by different researchers in a number of ways. Gen and Cheng (1996) have classified the techniques for handling infeasibility into the following four categories:

- *Rejecting strategy*
- *Repairing strategy*
- *Modifying genetic operators strategy*
- *Penalizing strategy*

Similarly Dadios and Ashraf (2006) also abstracted these four approaches from literature. However Richardson et al. (1989) described only two approaches: 1) modification of the genetic operators, and 2) penalizing chromosomes which violate constraints. Coello (2002) presented a complete state of the art survey of constraint handling techniques. Coello (2002) discussed in detail the different constraint handling approaches, i.e. penalty functions, special representations and operators, repair algorithms, separation of objectives and constraints and hybrid methods. Some of the conclusions of the Coello (2002) study are:

- *For beginners penalty based techniques are recommended as they are simple and quite efficient.*
- *For combinatorial optimization problems repair algorithms may be the best choice.*
- *For linear constraint the use of special representation and operators may become necessary.*
- *For highly constrained search spaces the use of techniques that separate constraints and objective may be useful.*
- *Most of the comparative studies of constraint handling techniques reported in literature are inconclusive. Hence the choice of a certain technique in the absence of knowledge about the domain remains as an open research problem.*

Further literature search also reveals some other approaches for handling infeasibility. One interesting approach is the random keys GA (RKGA) first introduced by Bean (1994) for sequencing and scheduling problems. The random keys representation encodes a solution with random numbers (drawn from [0, 1]). These values are used as sort keys to decode the solution. For n-jobs m-machines scheduling problem, each gene (a random key) consists of an integer in set {1, 2… m} and a fraction generated randomly from (0, 1).The integer part of any random key is interpreted as the machine assignment for the job represented by the fractional part. Sorting the fractional parts provides the job sequence on each machine.

The random keys approach has been well received by the research community and has been adapted by several researchers for other variants of the problem. For example Goncalves et al. (2005) used random keys representation combined with schedule generation procedure and a local search procedure for job shop scheduling with the objective of makespan minimization. Haral et al. (2007) used random keys approach for multiobjective single machine scheduling with flow time and maximum tardiness minimization and also some nontraditional objective. Norman and Bean (1997a) combined random keys with delay factor encoding and move search procedure to enhance performance in a job shop scheduling problem. Norman and Bean (1997b) and (2000) used it for scheduling operations on parallel machine tools. Norman and Bean (1999) used it for complex scheduling problems with certain complexities like multiple, nonidentical machines, nonzero ready times, sequence dependent setups, tool constraints, and precedence. Valente et al. (2006) used random keys in combination with several local search and initialization procedures for a version of the general early/tardy scheduling problem with no idle time. One interesting conclusion they have made is that the different initialization heuristics had very little effect on the solution quality, though greatly accelerated convergence. Wang and Uzsoy (2002) adapted dynamic programming algorithm combined with random keys encoding for minimization of maximum lateness on batch processing machine. However they have expressed their concerns about the possible loss of information with random keys encoding. In light of the literature presented it is evident that the random keys approach is no doubt an interesting approach and as described by Snyder and Daskin (2006) useful for permutations kind of problems where the traditional, one or two-point crossover presents feasibility problems. However its usefulness for the problem at hand, without the need for some kind of repair mechanism/heuristics, could not be established. For the majority of the problems described above the random keys approach may be considered as a sub-class of the repairing strategy, of the four classes of infeasibility control methods mentioned earlier.

Another approach presented by Deb (2000) is a simple penalty function approach which does not require any penalty parameter (or factor), thereby making the approach applicable to a wide range of constrained optimization problems. This approach belongs to the penalty strategy and modifying genetic operators strategy classes of the constraint handling techniques classes described above. This technique which uses a tournament selection operator may be described as follows:

1. If both chromosomes are in the feasible region, the one with a better objective function value is selected.
2. If one chromosome is in the feasible region and the other out of the feasible region, the one in the feasible region is preferred.

3. If both chromosomes are infeasible, the one with smaller constraint violation i.e. closer to the feasible region is selected.

In the above three cases the objective function is calculated only in situation 1, while in 3 only constraint violations are determined. In 2 neither objective function values nor constraint violations are calculated. Since solutions are never compared in terms of both objective function value and constraint violation, thus the problem of attaching appropriate weights/penalty factors to different objectives and constraint violations is completely eliminated, as is the case in penalty strategy. The approach has been tested on nine commonly used test problems in literature including an engineering design problem of a welded beam. An exactly identical approach is also given by Andrzej and Stanislaw (2000) for multicriteria optimization. They have found it very efficient in the optimum design of a beam and robot gripper design. However the requirement of a niching method to maintain diversity and the use of other special operators make it less attractive for the problem at hand, especially in the presence of a simpler and most commonly used constraint handling technique, i.e. penalty strategy (or penalty function).

Penalty technique in essence transforms the constrained problem into an unconstrained problem by penalizing infeasible solution. Gen and Cheng (1996); Michalewicz et al. (1996) and others also, described it as the most common technique used in the genetic algorithms community for handling infeasibility. It allows movement through infeasible regions of the search space for better exploration, as opposed to rejection strategy which excludes infeasible solution altogether. However a unanimous view held by the above authors and Fonseca (1998) is that the penalty function is quite problem-dependent. They all have quoted the recommendations given by Richardson et al. (1989) for designing an efficient penalty function. These are:

- *Penalties which are functions of the distance from feasibility are better performers than those which are merely functions of the number of violated constraints.*
- *For a problem having few constraints, and few full solutions, penalties which are solely functions of the number of violated constraints are not likely to find solutions.*
- *Good penalty functions can be constructed from two quantities, the maximum completion cost and the expected completion cost.* The completion cost refers to the distance to feasibility.

- *Penalties should be close to the expected completion cost, but should not frequently fall below it. The more accurate the penalty, the better will be the solutions found. When penalty often underestimates the completion cost, then the search may not find a solution.*

Based on these guidelines, several researchers have proposed good techniques to build penalty functions. However Coello (2002) expressed concerns about the implementation of these guidelines in some cases. Further Coello (2002) described an ideal penalty to be as low as possible just above the limit, below which the best infeasible solutions exist i.e. the fitness of the worst feasible solution is better than the best infeasible solution. A large penalty discourages the exploration of the infeasible region while on the other hand too low a penalty may increase the search time in the infeasible region and fail to converge successfully.

Similarly Michalewicz et al. (1996) described the use of penalty functions to be non-trivial and that only some partial analysis of their properties was available. They argued that an individual solution might be penalized just for being infeasible (regardless of the amount constraint violation), or the “amount” of its infeasibility measured to determine the penalty value, or the effort of repairing i.e. the cost of making it feasible might be taken into account. They further suggested that the penalty may depend on:

- *The ratio between sizes of the feasible and the whole search space.*
- *The topological properties of the feasible search space.*
- *The type of the objective function.*
- *The number of variables.*
- *Number of constraints.*
- *Types of constraints.*
- *Number of active constraints at the optimum.*

Gen and Cheng (1996) have broadly classified penalty functions into two classes: i.e. constant penalty and variable penalty. They described constant penalty as less effective for complex problems. Variable penalty was further classified into static and dynamic penalties. In static penalty the penalty depends only on the amount of constraint violation and is not affected by the number of generation in the evolutionary process, while in dynamic penalty the penalty pressure increases as the evolution progresses. They also suggested that an ideal penalty should consider both the distance from feasible region as well as optima. However they described it difficult to embed the information of how close a solution is to the optimum and that all existing methods only considered the distance from the feasible region.

Coello (2002) presented a different classification: exterior and interior penalty functions. Interior penalty requires a feasible solution to start with and hence is the main drawback of this approach. However, the exterior penalty function starts with an infeasible solution and from there move towards the feasible region. This is the main reason of its popularity in GA because finding a feasible solution is itself NP-hard. (Coello, 2002; Yeniay, 2005).

In addition to the above methods Yeniay (2005) presented some further types e.g. annealing penalties, adaptive penalties, segregated GA, and co-evolutionary penalties, developed by different researchers. Annealing penalties, as the name implies, are based on annealing algorithm. In adaptive penalties, penalty parameters are updated for every generation according to information gathered from the population. Segregated GA uses two penalty parameters in two different populations to overcome the problem of too high and too low penalties. In co-evolutionary penalties the penalty is split into two values i.e. the number of constraint violations and the amount of constraint violation. However, Yeniay (2005) concluded that it was not possible to say which one of the methods was the best for every problem. The main problem of most of the methods is to set appropriate values of the penalty parameters which have to be set by experimentation.

Based on the literature review presented, penalty strategy that turns out to be the most simple and widely practiced technique for controlling infeasibility, is adopted for the GA models under study. For contiguous scheduling models, which in essence are permutations of jobs, the simple penalty technique is insufficient to control infeasibility hence a modified genetic operator strategy is adapted for these models.

## 9.3 Computational testing

Hall and Posner (2001) described that the purpose of computational experiments is to know, whether the model has the potential to work in specific situations, whether it is practical, to know its strength, weaknesses and place in comparison to other models. They have also mentioned purpose, comparability, unbiasedness and reproducibility as the four general principles of test data generations. Test instances generation is necessary as: 1) real data sets are rarely available; 2) few instances of real data sets may not evaluate the effects of various characteristics. Hooker (1995) distinguished between competitive testing (where benchmark solutions are used) and scientific testing (where control experiments are used). Hooker (1995) further highlighted the difference and argued that benchmarks are appropriate for development, while controlled experiments are needed for research. For computational experiments

Rardin and Uzsoy (2001) suggested that although the best test instances are those taken from real applications, it is rare to find more than a few data sets. This would be insufficient to test a heuristic comprehensively. Alternative sources are; random variation of real data sets; published on-line libraries; and/or randomly generated instances. Hall and Posner (2001) have pointed out the disadvantages of using library problems and hence why most research studies use random generated problem instances.

As noted by Anwar and De Vries (2004) and several others that it is not always possible to get exact solutions for problems, particularly for large problem sizes. Therefore approximate algorithms need to be tested against other approximate algorithms. To validate the single machine models developed in this study, the data used by Anwar and De Vries (2004) for their experiments (Experiment 3 and 4) is used here. The test instances generated by Anwar and De Vries (2004) have been chosen for two reasons: 1) no such comprehensive data set exists for irrigation scheduling problems; 2) it provides a fair ground for validation by comparing GA results to other techniques applied to the same test data. Anwar and De Vries (2004) have followed the common norms in OR for test instances generation and modified them as required for the irrigation scheduling problems. Details of the test instances used in this study are given in Anwar and De Vries (2004) with all terminology explained. The experiment designed for the contiguous single machine models examines the effect of the number of jobs on the quality of solution obtained from GA. Number jobs or the problem size is the most important problem parameter as the justification for the use of GA lies in the fact that for large problem sizes exact algorithms are not able to find solutions within reasonable time. Therefore through this experiment it is explored whether GA can perform more efficiently than integer programme and other heuristics as the problem size is increased.

For multimachine models a series of experiments are also designed. IP solutions are used as benchmarks for all multimachine models[2]. Experiment 1 is designed to test the quality of the solution of the simple multimachine GA models against demand for the single goal objective of minimizing earliness and tardiness under channel capacity restrictions. A range of demand-supply ratio is used for Experiment 1 with demand-supply ratio as low as 10% to as high as 90%. Experiment 2 is designed to test the quality of the solution of the simple multimachine GA models as the problem size (number of outlets/jobs) increases, for the single goal objective of minimizing earliness/tardiness under channel capacity restrictions. Experiment 3 is designed to test the quality of the solution of the simple multimachine GA models as the problem size (number of outlets) increases, for the dual goal objective of minimizing earliness/tardiness and discharge. Experiment 4 tests whether the simple multimachine GA models have the potential to solve a non-contiguous single machine problem with the same degree of performance as could be achieved through a single machine model developed specifically for the non-contiguous single machine problems. In other words it could be stated as: whether the multimachine models are able to help an irrigation manager to decide whether to supply irrigation water sequentially or simultaneously. This objective of Experiment 4 is achieved by comparing the performance of the simple multimachine GA models with a non-contiguous single machine GA model applied to the same data. Experiment 5 is designed to examine the effect of travel time on the performance of the simple multimachine GA models with travel time. Experiment 6 is designed to examine the effect of non identical discharges on the performance of the complex multimachine GA model. Details of the data set generation are given for each experiment when they are presented individually in Section 11. Complex multimachine GA model with travel time can not be tested rigorously as no benchmark solutions are available. However its use for an irrigation scheduling problem is demonstrated through its application to a single instance.

[2] The work done by Dr. Arif Anwar for obtaining IP solutions is acknowledged.

# 10 Single machine models

The analogy between sequential irrigation and single machine ET problems in OR has already been established in Section 7. Literature suggests that for solving such problems approximate algorithms are the alternative option, as exact solutions for larger problem sizes (problem size or number of jobs greater than 12) are not possible within reasonable time. In this study, therefore, single machine irrigation scheduling models are developed using GA. These GA models are applied to the same data set as used by Anwar and De Vries (2004), for validation and evaluation purposes. To undertake an objective comparison no change is made to the scenarios modeled by Anwar and De Vries (2004).

## 10.1 Mathematical formulation

### 10.1.1 Model 1

Model 1 refers to the non-contiguous single machine ET model that allows idle time to be inserted between jobs i.e. there will be times within the irrigation interval where water will not be used by any farmer. This arrangement may require an excessive number of gate opening and closing operations, depending on the number of times idle time is inserted between jobs. Alternatively, a continuous flow system may be adopted and water allowed draining when not in use. A detailed description of the decision variables, objective function and the constraint is given below.

**Decision variables**

There are two decisions to be made: which outlet to receive water and at what time, i.e. the sequencing and scheduling. Thus the genes of a chromosome representing solution to this problem must have answers to these questions. The answers to these questions are incorporated into a single decision variable. This decision variable is represented by a scheduled start time row vector. Each element in the vector is a positive integer representing the point in time at which an outlet is scheduled to start receiving water and is expressed as follows.

$S_j$ = an element of the scheduled start time row vector (schedule start time of outlet *j*) (10.1)

where subscript *j* represents the outlet index i.e. the position of job in the chromosome and hence the sequence.

**Objective function**

The objective of the model is to find a sequence of jobs and the scheduled start times for all jobs with a minimum difference between the scheduled start time and the target start time. This is achieved by penalizing both early and tardy jobs. Some farmers may have higher priority for getting water supply earlier than others for a variety of reasons. For example, his/her crops have more value than others or more sensitive to water stress or perhaps for social/political reasons, etc. By using different unit costs for either earliness or tardiness, jobs may be prioritized. The objective function can be expressed as

$$\text{Minimize } [\sum_{j=1}^{J}(\alpha_j E_j + \beta_j T_j)] \qquad \forall\ j = 1, 2 \ldots J \tag{10.2}$$

where $E_j$ = earliness of job *j* (the difference of the target start time and the scheduled start time of outlet *j*); $T_j$ = tardiness of job *j* (the difference of the scheduled start time and the target start time of outlet *j*); $\alpha_j$ = cost of earliness per unit of time for job *j*; $\beta_j$ = cost of tardiness per unit of time for job *j*; *j* = job/outlet index = 1, 2… *J*; and *J*= total number of jobs/outlets.

**Constraints**

Any constraint violation causes a schedule to become infeasible. There are different techniques available to control infeasibility in genetic algorithm (as discussed in Section 9.2). Based on the literature review presented in Section 9.2, penalty strategy that turns out to be the most simple and widely practiced technique for controlling infeasibility, is adopted for the present model. In the penalty function technique each instance of infeasibility is appropriately penalized and (constraint violations expressed as) penalties are then added to the objective function. The resulting objective function may then be termed as fitness function.

There are two constraints in the current model. The first constraint is the irrigation interval constraint and the second is the overlap constraint. The penalties for constraint violations in the present formulation are as follows.

*i) Irrigation interval constraint*

Each outlet is to be scheduled with in the specified irrigation period. Any outlet scheduled outside this period will result in infeasible schedule. The penalty for this constraint violation may be mathematically expressed as:

$$P_I = \sum_{j=1}^{J} [(S_j + D_j - G)\delta_j + (\mathrm{S}_{\mathrm{int}} - S_j)\lambda_j] \qquad \forall\ j = 1, 2 \ldots J \tag{10.3}$$

where $P_I$ = penalty for irrigation interval violation; $G$ = total irrigation time available; $S_{int}$ = start time of the irrigation interval; $S_j$ = scheduled start time of outlet $j$; and $D_j$ = duration of outlet $j$.

$$\delta_j = 1 \quad \text{if} \quad S_j + D_j > G \quad \forall\ j \tag{10.4}$$
$$= 0 \quad \text{otherwise}$$

$$\lambda_j = 1 \quad \text{if} \quad S_j < S_{int} \quad \forall\ j \tag{10.5}$$
$$= 0 \quad \text{otherwise.}$$

*(ii) Overlap constraint*

Only one outlet is to be served at a time. The penalty for violation of this constraint is determined by summation of the number of times overlap occurs in all time blocks and is expressed mathematically as follows.

$$P_O = \sum_{t=1}^{T} \left(\sum_{j=1}^{J} \psi_{tj}\right) \tag{10.6}$$

$$\text{if } \left(\sum_{j=1}^{J} \psi_{tj}\right) \leq 1 \text{ then } \left(\sum_{j=1}^{J} \psi_{tj}\right) = 0 \qquad \forall \ t = 1, 2\ldots T \tag{10.7}$$

where $P_O$ = penalty for overlap of jobs; $t$ = time block index = 1, 2…$T$; and, $T$ = total number of time blocks.

$$\psi_{tj} = 1 \quad \text{if } S_j \leq t < S_j + D_j; \qquad \forall\, t, \ \forall\, j \tag{10.8}$$

$$= 0 \quad \text{otherwise}$$

By adding these penalties for constraint violations to the objective the resultant fitness function may then be mathematically expressed as follows.

$$\text{Minimize } F = \left[\sum_{j=1}^{J} (\alpha_j E_j + \beta_j T_j) + R_I P_I + R_O P_O\right] \tag{10.9}$$

where $F$ = fitness function; $R_I$ = penalty weight for $P_I$ ; and $R_O$ = penalty weight for $P_O$ .

### 10.1.2 Model 2

Model 2 refers to a series of single machine ET contiguous irrigation scheduling models. There are three variations of Model 2 i.e. 2a, 2b, and 2c. In Model 2a jobs are scheduled contiguously and idle time inserted at the end of the last job. In Model 2b all the jobs are scheduled contiguously and idle time inserted before the start of the first job. In Model 2c the jobs are scheduled contiguously and idle is inserted preceding the start of first job and /or proceeding the end of the last job.

#### Decision variables

The only decision to be made in these permutation models is the sequence of jobs. The chromosome is a permutation of jobs where each gene represents job *j*. Once the sequence of jobs is decided, the scheduled start time of each job can then be calculated. For Model 2a the scheduled start time of the first job in the sequence is “0” or the time the irrigation interval starts. The scheduled start time for the rest of the jobs can be calculated as follows.

$$\check{S}_i = \check{S}_{i-1} + \check{D}_{i-1} \qquad \forall\ i = 2, 3 \ldots J \qquad (10.10)$$

where $\check{S}_i$ = scheduled start time of the job at the $i^{th}$ position in the jobs sequence (chromosome); $\check{S}_{i-1}$ = scheduled start time of the job preceding the job at the $i^{th}$ position in the jobs sequence; $\check{D}_{i-1}$ = duration of the job preceding the job at the $i^{th}$ position in the jobs sequence; and $i$ = position of the job in the jobs sequence, so that $i$ = 2 represents the second job in the sequence whereas $i = J$ the last job in the sequence. For Model 2b the scheduled start time of the first job is the end of the idle time and can be expressed as:

$$\check{S}_1 = G - \sum_{j=1}^{J} Dj \qquad (10.11)$$

where $\check{S}_1$ = the scheduled start time of the first job in the sequence. For the remaining jobs the scheduled start time can be calculated as in (10.10) after $\check{S}_1$ has been calculated. For Model 2c the scheduled start time of the first job is the end of the idle time inserted before the start of first job. Idle time in this case has a value equal to a random integer number in the range of irrigation interval minus the makespan. For the remaining jobs the scheduled start time can be calculated as in (10.10) after $\check{S}_1$ has been calculated as follows.

$\check{S}_1$ = an integer randomly selected in the range between 0 and $(G - \sum_{j=1}^{J} D_j)$ (10.12)

## Objective function

There is no change in the objective function and is similar to that of Model 1 (10.2) for all contiguous models i.e. 2a, 2b, and 2c. The objective is to find a sequence or permutation of jobs that best matches the scheduled start times with the target start times.

**Constraints**

Since in single machine contiguous models the population consists of a permutation of jobs, no irrigation interval constraint violation occurs. Other infeasibility problems are controlled via modified genetic operators. The objective function for all models in this category is the fitness function for all the individuals of the population.

## 10.2 GA implementation

The GA, for all the models described in the preceding sections, was implemented using JGA, a java genetic algorithms library (Medaglia and Gutiérrez, 2006a). Some of the built-in classes were modified and some additional new classes were added to develop a complete GA implementation. The logic for this implementation of the genetic algorithm is presented in Figure 10.1; where, *t* is the generation counter; *T* is the maximum number of generations; *P(t)* is the population at generation *t*; *Cm(t)* and *Cc(t)* are the children populations obtained by the mutation and crossover operators, respectively; *C(t)* is the children population; and *E(t)* is the expanded population formed by the current population and their children.

$t \leftarrow 1$<br>
initialize $\mathcal{P}(t)$<br>
evaluate $\mathcal{P}(t)$<br>
**while** $t \leq T$ **do**<br>
    mutate $\mathcal{P}(t)$ and generate $\mathcal{C}_m(t)$<br>
    cross $\mathcal{P}(t)$ and generate $\mathcal{C}_c(t)$<br>
    $\mathcal{C}(t) \leftarrow \mathcal{C}_m(t) \cup \mathcal{C}_c(t)$<br>
    evaluate $\mathcal{C}(t)$<br>
    $\mathcal{E}(t) \leftarrow \mathcal{P}(t) \cup \mathcal{C}(t)$<br>
    select $\mathcal{P}(t+1)$ from $\mathcal{E}(t)$<br>
    $t \leftarrow t + 1$<br>
**end while**

**Figure 10.1** The logic behind GA implementation (Medaglia and Gutiérrez, 2006a).

Since the main objective of the development of the non-contiguous model is to evaluate the performance of the multimachine model, its implementation detail will be presented later when the multimachine models will be evaluated. The GA implementation for the contiguous single machine models is, however, presented as follows.

## Initial Population

For all versions of Model 2 the chromosome consists of non repeated integer valued genes, of length equal to the number of jobs. Each gene is an integer in the range between 1 and the total number of jobs. The population consists of randomly generated permutations of job sequences because permutations are considered natural representation for sequences.

## Selection

The best individual selection is used for the present models. Best individual selection is described by Coley (1999) as "*elitism*", where the elite member is not only selected but a copy of it is also preserved and becomes a part of the next generation without any perturbation by crossover or mutation operators. In the best individual selection used here, best individuals are selected from an enlarged population. The enlarged population is formed by offspring produced from crossover and mutation of parents as well as by the individuals from the current population (Medaglia and Gutiérrez, 2006b).

## Crossover

For all versions of Model 2, order-based crossover (OX) is used (Davis, 1991). The OX operator selects at random two cut points along the strings. The substrings between the two cut points of both parents are exchanged. Starting from the second (right) cut point of both parents the remaining positions for each chromosome are completed by omitting the duplicated genes. When the end position of the string is reached, it continues from the first position till the chromosome is completed. In this way the OX operator avoid any infeasibility due to repeated genes in a sequence. A crossover probability of 0.8 was used for all models after satisfactory initial experimentation, and which is also used by Medaglia and Gutiérrez (2006b) in some of their JGA application.

## Mutation

For all contiguous models, inversion mutation is used as a mutation operator. In inversion mutation the order of a randomly picked permutation section is inverted. For instance, if 1-2-

3-4-5-6 is a sequence, and 3-4-5 is the randomly picked section; the mutated permutation is 1-2-5-4-3-6. The purpose is to maintain diversity as well as feasibility. A mutation rate of 0.2, which was found satisfactory during preliminary experimentation, is used in the current models. The mutation rate is interpreted as the chance of mutation of a given genotype. The same has been used in some of the application of JGA by Medaglia and Gutiérrez (2006b) and several other applications in literature.

**Termination**

The number of generation has been used as the termination criteria for the present models. Although it is simple and easily implemented, however the drawback is that if the best solution is found in the early generations the evolution cycles still run unnecessarily till the end. This may be circumvented if the maximum number of generations is carefully chosen.

## 10.3 Results and discussions

The solution quality of the three contiguous GA models i.e. Model 2a, 2b, and 2c were tested against the integer programme (IP) and the heuristics (H) by Anwar and De Vries (2004) for the same data set. Anwar and De Vries (2004) used the parameters given in Table 10.1 for the experimental data generation. Anwar and De Vries (2004) defined two terms, the tardiness factor (*F*) and the range factor (*R*) for determining the upper and the lower limits for randomly generating the target start times. A high tardiness factor value means that all jobs have target start times nearer the beginning of an interval, therefore a high number of jobs will be scheduled tardy. A tardiness factor of 0.5 means, that there are likely to be as many tardy jobs as early. Similarly, range factor determines the range within which target start times will lie within the given irrigation interval. A range factor of 1 means that, the farmers are allowed to request any time for water supply within the irrigation interval.

Since the IP takes very long to solve scheduling problems particularly with large problem sizes (number of jobs greater than 12), its execution time was limited to $10^4$ seconds. If the IP did not solve a problem within this time limit, it was terminated. It is worth mentioning that the IP would have obtained a solution equal to or better than that from the GA if it were allowed to continue running beyond the allocated time. The solution quality of all the models was tested against problem size. The problem size or outlet numbers of 8, 10, 12, 15, 20, and 25 were used. The irrigation interval was arbitrarily set equal to the number of outlets served, multiplied by 100. For example, for an 8 job problem the irrigation interval is equal to 800 time units. To put it into context, 800 may represent an irrigation interval of eight days i.e. all times and durations are rounded off to $1/100^{th}$ of a day (approximately 15 minutes), thus irrigation duration for an outlet, of 40 time units is approximately 10 hours. For each problem size 25 different instances were tested. GA models were run for a number of combinations of maximum generation number and population size. Generation numbers of 500, 1000, 2000, and 2500 were used with an initial population size of 100. Generation numbers of 2500, 3000, and 3500 were used with an initial population size of 200.

**Table 10.1 Parameters for the computational experiments of Model 2a, 2b, and 2c.**

| Parameter | Values selected |
|---|---|
| Number of jobs/outlets | 8, 10, 12, 15, 20, 25 |
| Irrigation duration of individual farmers | Uniformly distributed random integer from the range [1, 100] |
| Tardiness factor | 0.5 |
| Range factor | 1.0 |
| Target start time | Uniformly distributed random integer $[(1\text{-}F\text{-}R/2)G, (1\text{-}F\text{+}R/2)G]$ |
| Cost of earliness/tardiness per unit of time | Uniformly distributed random integer from the range [0, 5] |
| Irrigation interval | Number of jobs x 100 |
| Number of instances | 25 for each number of jobs—total instances 150 |

The relative error of GA with IP and heuristics (H) were calculated as in equation 10.13 and 10.14 respectively. The average relative error values of all the 25 instances for each problem size were used as criteria for comparison.

$$\text{Relative error IP} = \left(\frac{\text{GA} - \text{IP}}{\text{IP}}\right) * 100 \qquad (10.13)$$

$$\text{Relative error H} = \left(\frac{\text{GA} - \text{H}}{\text{H}}\right) * 100 \tag{10.14}$$

### 10.3.1 Model 2a

Model 2a using GA is referred to in this thesis by Model 2a, for brevity. Figure 10.2 shows the objective function values of Model 2a relative to that of integer programming (IP) and heuristics (H). Figures 10.2(a-b), shows that for problem sizes 8 and 10 the objective values of Model 2a fall on the line of perfect fit, indicating that Model 2a and IP have identical results. As the problem size is increased, the difference between Model 2a and IP becomes more visible as is indicated by Figures 10.2 (c-f). For problem size 15 and beyond, the objective function values of Model 2a fall below the line of perfect as indicated by Figures 10.2 (d-f). These are the instances where the IP was unable to solve optimally with in the allocated time of 3 hours and Model 2a performed better than IP for all those instances within the allocated time. Figure 10.2 also shows that the objective function values of H are in close proximity of Model 2a. A more comprehensive analysis of results for Model 2a is presented in Table 10.2a and 10.2b which shows the relative error values for all the 25 instances for each problem size, the minimum and maximum values of errors and their standard deviations. The relative errors are calculated using equations 10.13 and 10.14. The minimum, maximum, and the standard deviation of the relative errors for each problem size have been given to show consistency in performance within the same problem size for the 25 different instances. Table 10.2 b shows the comparison of the number of optimum and feasible schedules obtained by the three different models.

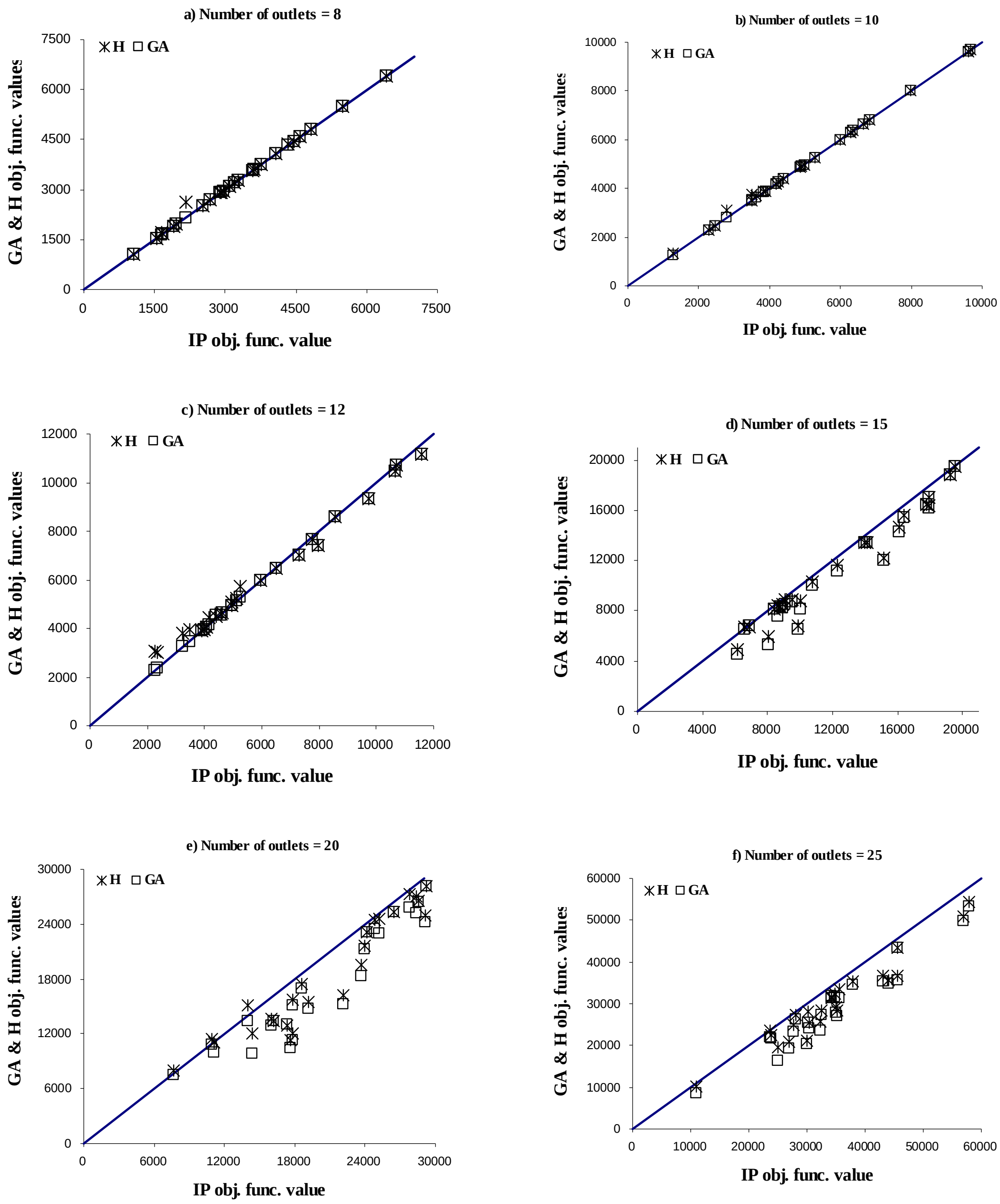



**Figure 10.2** Model 2a -Objective function values (GA vs. IP & GA vs. H)

**Table 10.2a Analysis of results for Model 2a**

| Statistics | Model 2a vs. IP | | | | | | Model 2a vs. H | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Number of jobs | | | | | | Number of jobs | | | | | |
| | **8** | **10** | **12** | **15** | **20** | **25** | **8** | **10** | **12** | **15** | **20** | **25** |
| Mean error (%) | 0.0 | 0.0 | -0.9 | -10.0 | -14.8 | -16.4 | -0.9 | -1.0 | -3.8 | -2.6 | -4.8 | -5.0 |
| Maximum error (%) | 0.0 | 0.2 | 2.5 | 0.0 | -1.2 | -5.1 | 0.0 | 0.2 | 0.5 | 1.0 | 0.8 | 0.5 |
| Minimum error (%) | 0.0 | 0.0 | -7.5 | -34.0 | -40.4 | -34.2 | -17.3 | -9.5 | -25.5 | -11.1 | -18.4 | -15.6 |
| Standard deviation | 0.0 | 0.0 | 2.0 | 9.5 | 11.3 | 8.5 | 3.5 | 2.4 | 7.2 | 3.3 | 4.2 | 4.0 |

**Table 10.2b Analysis of results for Model 2a**

| Model | Number of solutions | Number of jobs | | | | | |
|---|---|---|---|---|---|---|---|
| | | **8** | **10** | **12** | **15** | **20** | **25** |
| **IP** | Optimum | 25 | 25 | 17 | 1 | 0* | 0* |
| | Feasible | 25 | 25 | 25 | 25 | 25 | 25 |
| **Model 2a** | Optimum | 25 | 23 | 14 | 1 | n/a* | n/a* |
| | Feasible | 25 | 25 | 25 | 25 | 25 | 25 |
| **H** | Optimum | 19 | 18 | 8 | 0 | n/a* | n/a* |
| | Feasible | 25 | 25 | 25 | 25 | 25 | 25 |

* n/a: no optimum solution was found by IP within the allocated time

It can be seen from the Table 10.2a that generally the differences between the minimum and maximum values of the relative errors increase as the problem size is increased and so are the standard deviations. For 8 jobs problem Model 2a was able to find the global optimum for all the 25 instances, as against heuristics which obtained 19 out of 25 optimum solutions. For 10 jobs the number of optimum solution by Model 2a was 23 out of 25, as against heuristics by Anwar and De Vries (2004) which obtained 18 out of 25 optimum solutions. For the12 jobs problem, the IP was unable to solve within the time limit of $10^4$ seconds for 8 of the 25 instances. For every instance where the IP failed to solve within the allocated time, the GA was able to find better solution than the IP and therefore mean error of GA relative to IP is negative for the 12 jobs problem. The number of optimum solution obtained by GA was 14 out of 17, as against 8 by heuristics (Table 10.2b).

For the 15 jobs problem all but one instance failed to solve within the allocated time using the IP. GA was even able to find optimum solution to that single instance whereas the heuristics did not. For 24 of the 25 instances, where the IP failed to solve within the allocated time the GA was able to obtain a feasible solution. For the 20 and the 25 jobs problem, all instances failed, within the time allocated, to reach a global optimum using the IP. In such situations the count of exact solutions is no longer applicable as a measure of solution quality. For both 20 and 25 jobs problems the GA was able to obtain feasible solutions for all the 25 instances in considerably less time. For example for instance 1 in the 25 jobs problem the GA was able to find better solution than IP in just 31 seconds as compared to the 10000 seconds by IP. Overall the GA performed significantly better particularly at large problem sizes (equal to or greater 15) and completely outperformed the heuristics by Anwar and De Vries (2004) and was able obtain feasible schedules to all instances, even for those where the IP failed to obtain optimum solution with in the allocated time of 3 hours.

### 10.3.2 Model 2b

Model 2b using GA is referred to in this thesis by Model 2b, for brevity. Figure 10.3 shows the objective function values of Model 2b relative to that of integer programming (IP) and heuristics (H). A detailed analysis of results for Model 2b is presented in Table 10.3a and 10.3b. The results for Model 2b are not significantly different than Model 2a and almost follow the same trends. For the 8 jobs problem the GA was able to obtain the exact solution for 24 of the 25 instances tested, as against 17 of the 25 instances by the heuristics reported in Anwar and De Vries (2004). For the 10 jobs problem again in 24 of the 25 instances, the GA was able to obtain the exact solution, as against 14 of the 25 instances by the heuristics reported in Anwar and De Vries (2004). For the 12 jobs problem, the IP was not able to obtain optimum solutions for all the 25 instances in the allocated time and obtained only 12 optimum solutions. The GA was able to obtain optimum solutions for the 11 of the 12 instances, as against 6 of the 12 instances by the heuristics reported in Anwar and De Vries (2004). The

GA was able to obtain feasible solutions for all the instances, even for those instances where the IP did not obtain optimum solutions within the allocated time.

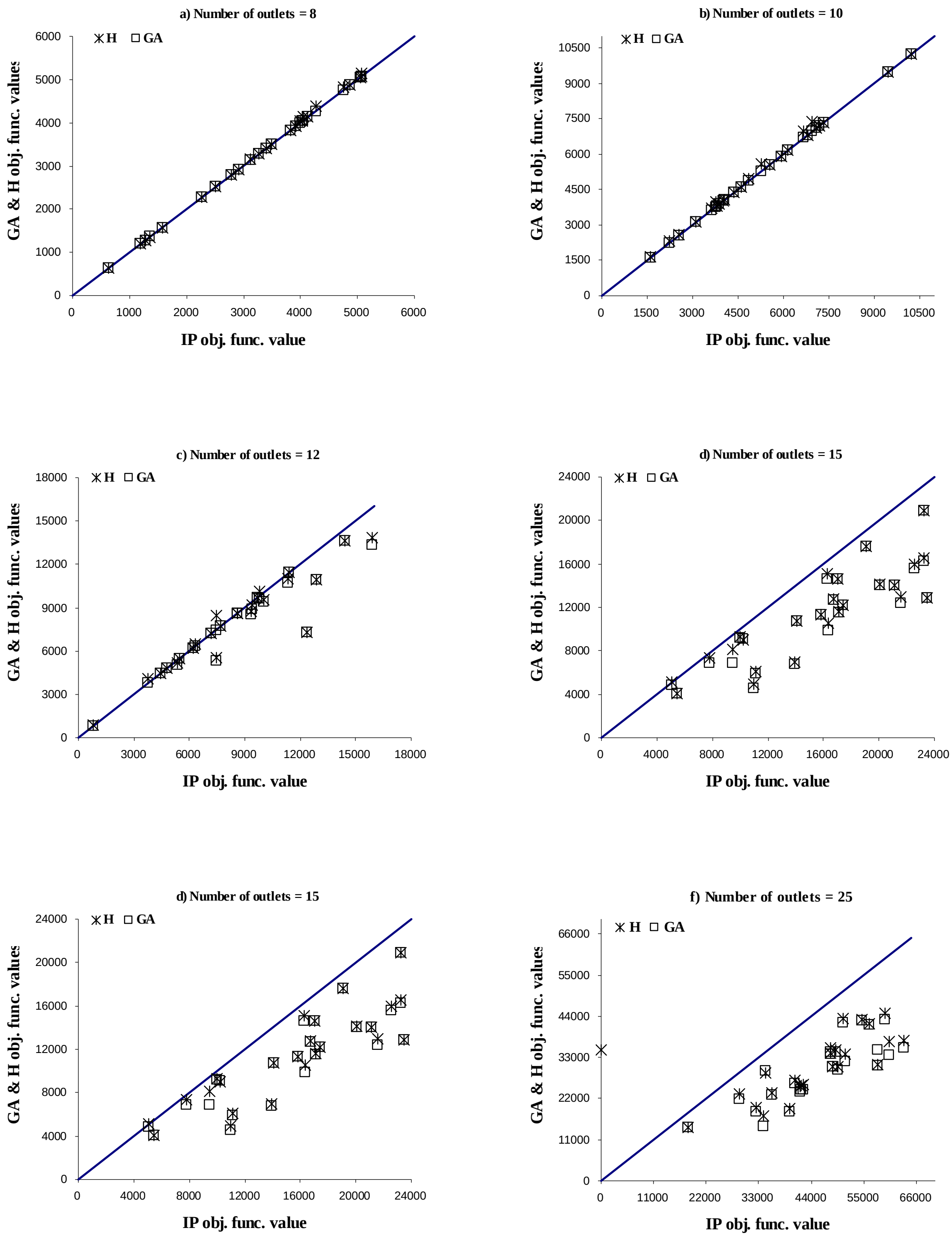



**Figure 10.3** Model 2b -Objective function values (GA vs. IP & GA vs. H)

**Table 10.3a Analysis of results for Model 2b**

| Statistics | Model 2b vs. IP | | | | | | Model 2b vs. H | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Number of jobs | | | | | | Number of jobs | | | | | |
| | **8** | **10** | **12** | **15** | **20** | **25** | **8** | **10** | **12** | **15** | **20** | **25** |
| Mean error (%) | 0.0 | 0.0 | -5.6 | -27.1 | -32.0 | -35.3 | -0.3 | -1.3 | -2.4 | -2.6 | -3.0 | -3.6 |
| Maximum error (%) | 0.7 | 0.5 | 0.2 | -3.4 | -1.4 | -14.7 | 0.7 | 0.0 | 0.0 | 0.8 | 2.8 | 1.7 |
| Minimum error (%) | 0.0 | 0.0 | -40.9 | -59.0 | -58.3 | -57.4 | -2.5 | -5.8 | -12.2 | -15.9 | -18.9 | -16.9 |
| Standard deviation | 0.1 | 0.1 | 10.1 | 14.9 | 12.3 | 10.9 | 0.8 | 2.1 | 3.5 | 3.7 | 4.4 | 3.7 |

**Table 10.3b Analysis of results for Model 2b**

| **Model** | **Number of solutions** | **Number of jobs** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **8** | **10** | **12** | **15** | **20** | **25** |
| **IP** | Optimum | 25 | 25 | 12 | 0* | 0* | 0* |
| | Feasible | 25 | 25 | 25 | 25 | 25 | 25 |
| **Model 2b** | Optimum | 24 | 24 | 11 | n/a* | n/a* | n/a* |
| | Feasible | 25 | 25 | 25 | 25 | 25 | 25 |
| **H** | Optimum | 17 | 14 | 6 | n/a* | n/a* | n/a* |
| | Feasible | 25 | 25 | 25 | 25 | 25 | 25 |

* n/a: no optimum solution was found by IP within the allocated time

For the 15 jobs and the larger problems, the IP was not able to obtain any optimum solution in the allocated time and in each problem the GA was able to obtain a better solution than IP in considerably less time. For example for instance 1 in the 25 jobs problem the GA was able to find a better solution in just 65 seconds as compared to the solution obtained by the IP after $10^4$ seconds of execution time. The IP did not obtain the optimum solution in the time limit of $10^4$ seconds. It is worth noting that Model 2a found its best solution for the same instance in 31 seconds as compared to 65 seconds by Model 2b. This escalation in solution time and also the increased values of the standard deviations for Model 2b as compared to Model 2a indicates that Model 2b is computationally more demanding than Model 2a. The solution quality of Model 2b is, however, slightly better than Model 2a, which is indicated by the increased values of the mean negative relative errors with IP as compared to the mean negative relative errors with IP for Model 2a.

### 10.3.3 Model 2c

Model 2c using GA is referred to in this thesis by Model 2c, for brevity. Figure 10.4 shows the objective function values of Model 2c relative to that of integer programming (IP) and heuristics (H). Figure 10.4 shows that the results for Model 2c either plot on, or below the line of perfect fit, in contrast to H which plot mostly above the line. This indicates better solution quality of Model2c than H across all problem sizes. A detailed analysis of results for the Model 2c is presented in Table 10.4a and 10.4b. For the 8 jobs problem the GA was able to obtain optimum solutions for all the 25 instances, whereas the heuristics did not obtain a single optimum solution. For the 10 jobs problem the GA obtained 20 optimum solutions out of 25 as against no optimum solution by the heuristics. Model 2c proved less complex for the IP as the IP was able to obtain optimum solutions for all the 25 instances for even the 12 jobs problem. The IP in the previous two models failed to obtain optimum solutions for all the 25 instances within the allocated time for the 12 jobs problem. For the 12 jobs problem the GA obtained 15 optimum solutions out of 25 as against no optimum solution by the heuristics. Similarly the IP was also able to obtain 13 optimum solutions for the 15 jobs problem. Of the 13 optimum solutions by IP the GA was able to obtain 7 optimum solutions whereas again the heuristics was unable to produce any optimum solution. For the 20 and 25 jobs problems again the IP, as in the previous two models, was unable to solve to optimality, whereas the GA was able to obtain feasible schedules to all instances.

The GA once again was able to obtain better solutions in considerably less time than IP for instances where the IP was unable obtain optimum solution within the allocated time of 3 hours. For example for instance 1 in the 25 jobs problem the GA was able to find a feasible schedule solution in just 30 seconds, where the IP did not reach to the optimum solution within the allocated time. This solution time for instance 1 of the 25 jobs problem is less than half of that by Model 2b and 1 second less than that by Model 2a for the same instance.

Model 2c performed consistently better against the heuristics over the whole range of the problem size. The improvement over heuristics was very significant, in fact much better than Model 2a and 2b and did not vary with increasing problem size. The improved performance of Model 2c may be attributed to the increased flexibility when idle is allowed on both side of the irrigation interval.

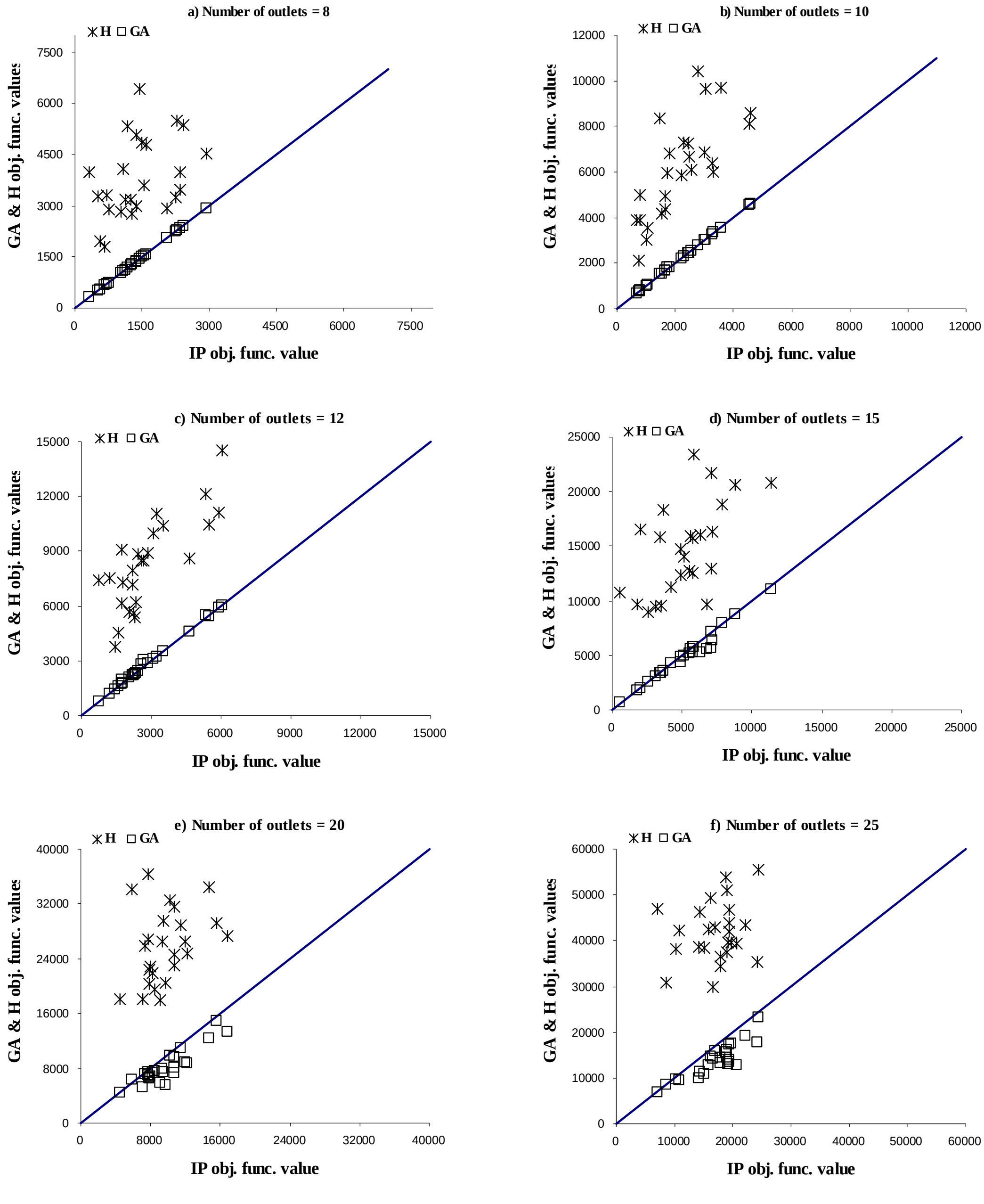



**Figure10.4** Model 2c -Objective function values (GA vs. IP & GA vs. H)

**Table 10.4a Results for Model 2c**

| Statistics | Model 2c vs. IP | | | | | | Model 2c vs. H | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Number of jobs | | | | | | Number of jobs | | | | | |
| | **8** | **10** | **12** | **15** | **20** | **25** | **8** | **10** | **12** | **15** | **20** | **25** |
| Mean error (%) | 0.0 | 0.4 | 1.8 | -3.0 | -15.5 | -17.8 | -61.5 | -65.0 | -65.5 | -66.3 | -68.2 | -66.4 |
| Maximum error (%) | 0.0 | 4.5 | 13.7 | 17.1 | 7.9 | -2.1 | -29.6 | -43.9 | -46.1 | -42.6 | -48.9 | -49.9 |
| Minimum error (%) | 0.0 | 0.0 | 0.0 | -19.0 | -42.8 | -38.0 | -91.8 | -84.0 | -89.5 | -94.0 | -81.5 | -85.5 |
| Standard deviation | 0.0 | 1.0 | 3.8 | 7.5 | 12.0 | 10.1 | 17.2 | 11.2 | 10.7 | 11.9 | 7.3 | 8.6 |

**Table 10.4b Analysis of results for Model 2c**

| Model | Number of solutions | Number of jobs | | | | | |
|---|---|---|---|---|---|---|---|
| | | **8** | **10** | **12** | **15** | **20** | **25** |
| **IP** | Optimum | 25 | 25 | 25 | 13 | 0* | 0* |
| | Feasible | 25 | 25 | 25 | 25 | 25 | 25 |
| **Model 2c** | Optimum | 25 | 20 | 15 | 7 | n/a* | n/a* |
| | Feasible | 25 | 25 | 25 | 25 | 25 | 25 |
| **H** | Optimum | 0 | 0 | 0 | 0 | n/a* | n/a* |
| | Feasible | 25 | 25 | 25 | 25 | 25 | 25 |

* n/a: no optimum solution was found by IP within the allocated time

### 10.3.4 Conclusions

Overall the GA models have performed very well and completely outperformed the heuristics by Anwar and De Vries (2004). The GA was able to obtain feasible solution solutions for larger problems (i.e. problems with number of jobs equal to or greater than15) where the IP was unable to obtain optimum solutions within the allocated time of 3 hours. The difference in performance of the three models shows the sensitivity of the models to the insertion of the idle time. Inserting idle time on both sides of a schedule (before the start of irrigation and after the irrigation is complete), has been found useful. This is indicated by the fact that Model 2c performed better than Model 2a and 2b in terms of solution quality and computational efficiency. Having some spare time before the start of irrigation and after the irrigation is complete, may provide some flexibility on the operational level as well.

It was also observed with all the single machine models that increasing the generation numbers for the same population size does not result in any improvement in solution quality beyond certain number of generations. The GA runs unnecessarily without any improvement till the maximum number of generations is reached. Therefore it is recommended for a GA to have a certain stopping criteria based on improvement in solution quality. For example if there is no improvement in solution quality over a certain number of generations then the GA should be terminated to avoid unnecessary use of computing resources and increase in computational time. A similar criteria is therefore planned for all the remaining models to be developed in the current study.

# 11 Multimachine models

A typical irrigation scheduling problem is one of preparing a schedule to service a group of outlets which may be serviced simultaneously. This problem has an analogy with the classical multimachine earliness/tardiness scheduling problem in operations research. As discussed in detail in Section 7.4, in previously published work integer programme were used to solve simultaneous irrigation scheduling problems, however such scheduling problems belong to a class of combinatorial optimization problems known to be computationally demanding (NP-hard). This is widely reported in operations research. Hence integer programme can only be used to solve relatively small problems usually in a research environment where considerable computational resources and time can be allocated to solve a single schedule. For practical applications metaheuristics such as genetic algorithms, simulated annealing, or tabu search methods need to be used. However as reported in the literature, these need to be formulated carefully and tested thoroughly. The current research is to explore the potential of genetic algorithm to solve the simultaneous irrigation scheduling problem. Figure 7.5 illustrate a simultaneous irrigation scheduling problem. Related literature that provides justification and importance of the current research is already presented in detail in Section 7. In the following sections the development of a series of multimachine models (simple multimachine and complex multimachine problems), using GA, is presented to achieve the objectives of the research given in Section 8.

## 11.1 Simple multimachine model

The analogy between simultaneous irrigation (where more than one farmer is allowed to receive water) and the classical multimachine scheduling problem has already been established. The case where all farmers receive the same discharge is referred to as simple multimachine scheduling. GA Models based on two different approaches (i.e. the stream tube approach and the time block approach) for solving the simple multimachine scheduling problems are developed and evaluated in the following sections.

### 11.1.1 Stream tube approach (STA)

This model uses the concept of stream tubes as originally put forward by Suryavanshi and Reddy (1986) and further developed by Anwar and Clarke (2001). The stream tubes approach considers the supply channel to consist of a number of identical stream tubes. By allowing different stream tubes to supply different outlets at any given time, simultaneous supply of water to several users is possible. The model is formulated so that the total number of stream tubes (or machines) is minimized or is not allowed to exceed a certain limit and at the same time outlets (or jobs) are scheduled as close as possible to their target start times. The given data includes the number of outlets, the duration of flow at each outlet, and the target start time for each outlet. The total irrigation period or interval is also given. It is assumed for the present model that the rate of supply (discharge) of water to each outlet is the same and hence the name “simple”. This is implemented by allowing only one stream tube to supply water to an outlet at a given time and thus limiting the maximum number of stream tubes equal to the number of outlets. Since the integer programme as used by Anwar and Clarke (2001) is not a practical tool to solve simultaneous irrigation scheduling problem within reasonable time, a GA solution to the problem is presented in this section. A detailed description of the model, using GA, is given below.

**Decision-variables**

The chromosome for stream tube approach is a concatenation of two row vectors containing integers only. Figure 11.1 shows a chromosome for the stream tube approach as an example. The first vector on the left hand side in the Figure 11.1 contains machines used by corresponding jobs and is termed the machine vector. The second vector contains scheduled start times of jobs and is termed the scheduled start time vector. The number of columns in each vector is equal to the number of jobs. The machine row vector provides the information of which job is assigned to which machine; alternatively which stream tube supplies which outlet. The indices in the machine vector describe jobs while the elements of the machine vector describe machines. For example the 2$^{nd}$ and 4$^{th}$ element (i.e.1) of the machine vector indicate that job 2 and 4 have been assigned to machine 1 (Figure 11.1). Similarly the index of the scheduled start time vector represents a job and, its corresponding element the scheduled start time of that job. For example the 4$^{th}$ element in the scheduled start time vector

(Figure 11.1) is 400. This means that job 4 has a scheduled start time of 400. The elements of the machine vector and the scheduled start time vector constitute the two decision variables for the STA. The first decision variable for the stream tube approach is represented as follows.

$M_j$ = an element of the machine row vector (machine to which job $j$ is assigned). (11.1)

The second decision variable, which is the scheduled start time of each job, is expressed by a positive integer representing the point in time at which an outlet is scheduled to start receiving water.

$S_j$ = an element of the scheduled start time row vector (schedule start time of outlet $j$). (11.2)

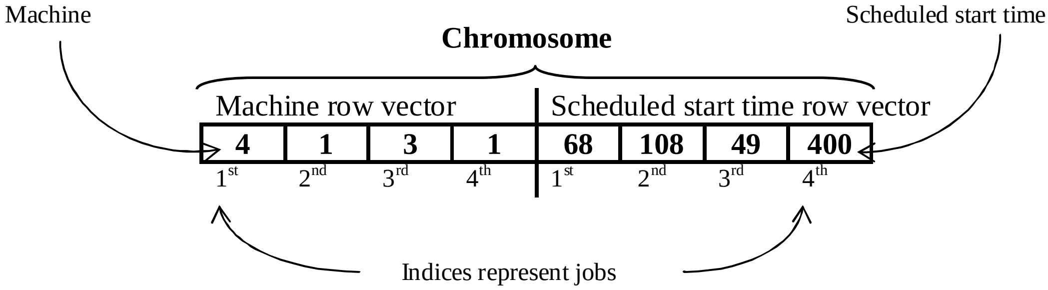



**Figure 11.1** Chromosomal representation

**Objective function**

The model has a dual-goal objective function. The first objective is to minimize the number of stream tubes that can be used to provide water to every outlet for the duration specified for each outlet within a given irrigation interval. In other words, the objective is to find the minimum capacity for a supply channel that would satisfy the users' requirements within the irrigation interval. The second objective is to minimize the sum of earliness and tardiness over all outlets, i.e. minimize the sum of the differences between target start time and scheduled start time of all outlets, within the number of stream tubes minimized by the first objective. A job is early when an outlet is scheduled earlier than its target start time. Similarly, a job is tardy if an outlet is scheduled later than its target start time. The model gives the same weight to earliness and tardiness so that a job early by 100 minutes will incur the same penalty as the job tardy by 100 minutes. Mathematically the objective function is expressed as follows.

$$\text{Minimize } [\sum_{j=1}^{J}(E_j + T_j) + Q_{max}] \tag{11.3}$$

where $E_j$ = earliness of job $j$ (the difference of the target start time and the scheduled start time of outlet $j$); $T_j$ = tardiness of job $j$ (the difference of the scheduled start time and the target start time of outlet $j$). The unit costs of earliness and tardiness ($\alpha_j$, $\beta_j$) as used in the single machine models have been ignored in all the multimachine models for simplicity as the multimachine problems are complex even without the consideration of the unit costs of earliness and tardiness. It is assumed that no farmer has any priority over the use of water on any other farmer. $Q_{max}$ = count of distinct stream tubes (supply discharge) and is calculated as

$$Q_{max} = \sum_{m=1}^{M} \theta_m \tag{11.4}$$

where $m$ = machine index = 1, 2… $M$; and, $M$ = total number of machines available; and

$$\theta_m \begin{cases} = 1 & \text{if machine } m \text{ is used} \\ = 0 & \text{otherwise.} \end{cases} \tag{11.5}$$

**Constraints**

Different techniques are available to control infeasibility in genetic algorithm as described in detail in Section 9.4. The penalty function technique is one of the techniques and is adopted partially for the present model. In the penalty function technique each instance of infeasibility is appropriately penalized and (constraint violations expressed as) penalties are then added to the objective function. The resulting objective function is termed the fitness function. However, a repair technique is also adopted occasionally. In a repair technique an infeasible schedule is not penalized rather repaired to make it feasible. For example, the mutation operator for the current model can be designed so that it never results in infeasibility caused by interval constraint violation.

There are three constraints in the present model, i.e. the capacity constraint, the irrigation interval constraint and the overlap constraint. These three constraints are discussed in detail as follows.

(i) *Capacity constraint*

This constraint ensures that at any point in time the supply should never exceed the capacity of the channel. However the constraint is used only if the model is used for minimizing earliness/tardiness under a fixed capacity. Any schedule that violates this constraint is penalized by an amount equal to the difference between supply and capacity and multiplied by a large positive integer. The penalty may be expressed mathematically as:

$$P_C = Q_{max} - Q \tag{11.6}$$

where $P_C$ = penalty for capacity constraint violation; and $Q$ = channel capacity.

(ii) *Irrigation interval constraint*

Each outlet is to be scheduled with in the specified irrigation interval. Any outlet scheduled outside this interval will result in an infeasible schedule. The penalty for this constraint violation may be mathematically expressed in the following way.

$$P_I = \sum_{j=1}^{J} [(S_j + D_j - G)\delta_j + (S_{int} - S_j)\lambda_j] \tag{11.7}$$

where $P_I$ = penalty for irrigation interval violation; $G$ = irrigation interval; $S_{int}$ = start time of the irrigation interval; $S_j$ = scheduled start time of outlet $j$; and $Dj$ = duration of outlet $j$.

$$\delta_j \quad = 1 \quad \text{if} \quad S_j + Dj > G \quad \forall\ j \tag{11.8}$$
$$\quad = 0 \quad \text{otherwise}$$

$$\lambda_j \quad = 1 \quad \text{if} \quad S_j < S_{int} \quad \forall\ j \tag{11.9}$$
$$\quad = 0 \quad \text{otherwise.}$$

(iii) *Overlap constraint*

Only one outlet is to be serviced by a single stream tube at a time. The penalty for violation of this constraint is determined by summation of the number of times overlap occurs on all machines. Information required for determining whether overlap exist between jobs on the same machine, is obtained from the machine vector and the scheduled start time vector. Each element of the machine vector is compared with subsequent elements of the vector. If an element is equal to any subsequent element, a value of 1 is stored in a matrix $\rho$ other wise a 0 is stored. A value of 1 would indicate more than one job on the same machine. For example, the chromosome presented in Figure 11.1 will result in a $\rho$ matrix as in Figure 11.2. An element at each index, in the machine row vector, is only compared with subsequent elements. For example, the element at index 1 with elements at index 2, 3, and 4; and element at index 2 with elements at index 3, and 4; and element at index 3 with the element at index 4. As it would be wasteful to compare for example an element at index 1 with the element at index 3 and then again compare the element at index 3 with the element at index 1. Thus values of 0's and 1's are stored in the $\rho$ matrix only above the diagonal. The values at the diagonal and below the diagonal are undefined and are not taken into account. This arrangement results in a more efficient algorithm.

| $M_j$ | **4** | **1** | **3** | **1** |
|---|---|---|---|---|

| $\rho_{jk}$ | | *k* | | | |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 |
| *j* | 1 | x | 0 | 0 | 0 |
| | 2 | x | x | 0 | 1 |
| | 3 | x | x | x | 0 |
| | 4 | x | x | x | x |

**Figure 11.2** $\rho$ matrix ( x = undefined )

From the machine row vector it is clear that only two of its elements are equal, i.e. the elements at index 2 and index 4 are equal. The element at index 2 and 4 is 1, which indicates that job 2 and job 4 are running on machine 1. This information from the machine row vector is stored in the $\rho$ matrix in a less compressed form by storing $\rho_{2,4} = 1$ and 0's at all other indices above diagonal (Figure 11.2).

Similarly the chromosome presented in Figure 11.1 also results in the matrix $\sigma$ as in Figure 11.3. $\sigma$ contains information of overlaps between jobs on the same machines. Jobs on the same machine as identified by the $\rho$ matrix are checked for overlap. For example, it is now known from $\rho$ matrix that only job 2 and job 4 are on the same machine. The duration of job 2, as given in Figure 11.3, is 300. So the completion time of job 2 is 408. The start time of job 4 is 400, which means job 4 starts running on machine 1 before job 2 is completed. Hence an overlap exists between job 2 and job 4. This information is stored in the $\sigma$ matrix in a less compressed form by storing $\sigma_{2,4} = 1$ and 0's at all other indices above diagonal as in Figure 11.3.

| $S_j$ | **68** | **108** | **49** | **400** |
|---|---|---|---|---|
| $D_j$ | 151 | 300 | 172 | 57 |

| $\sigma_{jk}$ | | $k$ | | | |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 |
| $j$ | 1 | x | 0 | 0 | 0 |
| | 2 | x | x | 0 | 1 |
| | 3 | x | x | x | 0 |
| | 4 | x | x | x | x |

**Figure 11.3** $\sigma$ matrix (x = undefined )

The elements of the matrices $\rho$ and $\sigma$ are then multiplied and the result is summed up as shown in Figure 11.4. The product of elements of two matrices of the same dimensions is known as the entry-wise product. The sum of this product across the rows and columns represents the penalty for overlap, which is equal to 1 for the example under consideration. A penalty equal to 1 indicates an occurrence of only one overlap. Mathematically this penalty is expressed as follows.

$$P_O = \sum_{j=1}^{J-1} \sum_{k=j+1}^{J} (\rho \bullet \sigma .............\forall j,k) \tag{11.10}$$

$$\rho_{jk} = 1 \quad \text{if } M_j = M_k \quad \forall\ j,k \quad \text{where } k > j \tag{11.11}$$

$$= 0 \quad \text{otherwise.} \quad \forall\ j,k \quad \text{where } k > j$$

$$\sigma_{jk} = 1 \quad \text{if } S_j < S_k + D_k \text{ AND } S_k < S_j + D_j \quad \forall\ j,k \quad \text{where } k>j \tag{11.12}$$

$$= 0 \quad \text{otherwise.} \quad \forall\ j,k \quad \text{where } k>j$$

where $P_O$ = penalty for overlap of jobs on the same machine; $M_j$ = machine used by job $j$; $M_k$ = machine used by job $k \neq j$; $S_k$ = start time of any other job $k \neq j$ on the same machine as $j$; $D_k$= duration of job $k$; $D_j$= duration of job $j$.

| $\rho_{jk}.\sigma_{jk}$ | | k | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | Sum |
| j | 1 | x | 0 | 0 | 0 | 0 |
| | 2 | x | x | 0 | 1 | 1 |
| | 3 | x | x | x | 0 | 0 |
| | 4 | x | x | x | x | $\sum(\rho \bullet \sigma) = 1$ |

**Figure 11.4** Entry wise product of $\rho$ and $\sigma$ (x = undefined)

The violation of any of the constraints discussed will cause a schedule to become infeasible. However, the mutation operator used in the current model is designed in such a manner that it never results in infeasibility caused by interval constraint violation. Similarly, the initial population is also randomly generated in such a manner that it never results in interval infeasibility. Also in the dual goal objective, capacity is not a constraint rather a part of the objective function. The overlap constraint then remains the only constraint and the penalty associated with its violation is thus only included in the fitness function. However, mathematical formulations for all the three penalties, resulting from violations of the three constraints, have been presented for the sake of completeness and future reference.

The fitness function for the dual goal objective is given by

$$\text{Minimize } F = [\{\sum_{j=1}^{J}(E_j + T_j) + Q_{\max}\} + R_O P_O] \tag{11.13}$$

where $F$ = fitness function; $R_O$ = penalty weight for $P_O$.

### 11.1.2 Time Block Approach (TBA)

Reddy et al. (1999) introduced the concept of scheduled start time for outlets while modelling the lateral canal scheduling problem in an irrigation system using integer programming. The approach adopted for solution was based on time blocks in contrast to the imaginary stream tube concept presented by Suryavanshi and Reddy (1986). In a time block approach the main canal capacity is expressed as the maximum number of outlets operated in any time block. Wardlaw and Bhaktikul (2004) also used time block approach in their GA model to solve the lateral canal scheduling problem addressed by Anwar and Clarke (2001) with stream tube approach. However limitations and shortcomings in these previous models, as discussed earlier, provide motivation for further research into the approach. The time block approach presented here is addressing the same problem as the stream tube approach. Hence the mathematical formulation for stream tube approach can be used for time block approach with the following modifications.

**Decision-variables**

The only decision variable is the scheduled start time of each job and is expressed by a positive integer representing the point in time at which an outlet is scheduled to start receiving water as given by equation (11.2). The chromosome for time block approach is represented by the scheduled start time row vector. This scheduled start time vector is the same as discussed earlier for stream tube approach.

**Objective function**

The objective function for the dual goal of minimizing earliness/tardiness and channel capacity is the same as given by equation (11.3). However, in the time block approach the capacity term used in equation (11.3) is calculated by a different approach. In time block approach the channel capacity is determined by the maximum number of jobs active in any time block and thus equation (11.4) is replaced by the following equation.

$$Q_{max} = \max[\sum_{j=1}^{J} \psi_{tj}] \qquad \forall \quad t = 1, 2\ldots T. \tag{11.14}$$

where $t$ = time block index = 1, 2…$T$; and, $T$ = total number of time blocks.

$$\psi_{tj} \quad = 1 \quad \text{if } S_j \leq t < S_j + Dj; \tag{11.15}$$
$$\quad = 0 \quad \text{otherwise.}$$

**Constraints**

The capacity constraint and the irrigation interval constraint presented for stream tube approach are valid for time block approach. The penalties as a result of violation of these constraints, expressed through equation (11.6) and equation (11.7) respectively are equally valid for time block approach. The overlap constraint as described for the stream tube approach is irrelevant for the time block approach. In the time block approach it is not necessary to explicitly assign a job to a particular machine as is the case in the stream tube approach. In time block approach each time block is checked for any job active in that time block and the number of jobs active in any time block is recorded. The maximum number of jobs active in any time block determines the capacity of the supply channel. In contrast in the stream tube approach more than one job on the same machine in any time block is an overlap. Thus more than one job active in any time block is regarded as an overlap in the stream tube approach if the jobs are on the same machine and in the time block approach it is interpreted as additional machines. Hence overlap of jobs is not used as a constraint in the time block approach. This is further illustrated through Figure 11.5, which depicts two jobs (Job1 and Job2) processed by a single machine "A". In the time block 2 between time 1 and 2 on the time line, Job1 and Job2 overlaps. The stream tube approach would regard this schedule as an infeasible schedule and hence penalize the overlap of Job 1 and Job 2 through penalty for overlap of jobs on the same machine as given by equation (11.10). In the time block approach this is interpreted as a violation of the capacity constraint as in the time block 2 it would record 2 jobs active. Thus in the time block 2 the system would require 2 machines to process the two jobs simultaneously. This is not possible under the given circumstances as only one machine "A" is available. Hence, even the time block approach would regard the schedule infeasible but through a mechanism different than the stream tube approach.

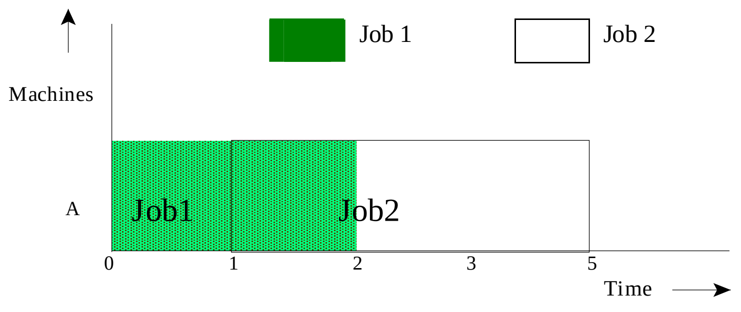



**Figure 11.5** Illustration of jobs overlap

The mutation operator applied to the stream tube approach model, never results in infeasibility caused by interval constraint violation. Since the same mutation operator is applied to the time block approach model, a penalty term for interval constraint violation is not required in the fitness function. Similarly, the initial population is also randomly generated in such a manner that it never results in interval infeasibility. Also in the dual goal objective, capacity is not a constraint rather a part of the objective function. Hence the fitness function for a dual goal objective in time block approach does not contain any penalty term. The fitness function for time block approach is thus the same as the objective function, as given by equation 11.3.

## 11.2 GA implementation for stream tube and time block approaches

In this thesis the GA model based on the stream tube approach will be termed as the stream tube GA and that based on the time block approach as the time block GA for the sake of brevity. The stream tube GA and the time block GA are both implemented using JGA, a java genetic algorithms library (Medaglia and Gutiérrez, 2006a). Some of the built-in classes are modified and some additional new classes are added to develop a complete GA implementation. The logic for this implementation of the genetic algorithm is the same as presented in Figure 10.1.

**Initial Population**

The initial population for stream tube GA is randomly generated such that each individual of the population is within the given irrigation interval. The range for the values of the machine vector is between “1” and a value equal the number of jobs. The second row vector contains scheduled start time of jobs. The values range between “0” and a value equal to the irrigation interval minus duration of the job for which scheduled start is being randomly generated. This ensures that the completion time of the job never exceeds the irrigation interval. A complete description of the chromosome for the stream tube GA has been presented earlier in Section 11.1.1.

In the time block GA the chromosome is a single row vector with randomly generated positive integers in contrast to the concatenation of two vectors in the stream tube GA. The values range between 0 and the irrigation interval minus duration of the job for which scheduled start is randomly generated. In effect the time block GA chromosome is identical to the second vector i.e. the scheduled start time vector in the stream tube GA.

**Selection**

The best individual selection is used for both models. Best individual selection is described by Coley (1999) as "*elitism*", where the elite member is not only selected but a copy of it is also preserved and becomes a part of the next generation without any perturbation by crossover or mutation operators. In the best individual selection used here best individuals are selected from an enlarged population. The enlarged population is formed by offspring produced from crossover and mutation of parents as well as by the individuals from the current population (Medaglia and Gutiérrez, 2006b). During initial experimentation for selecting algorithm parameters for the present models, best individual selection produced better results than the roulette wheel selection.

**Crossover**

The uniform crossover is used as a crossover operator for both models. In uniform crossover two parents are selected to produce two children. For each position on the two children it is randomly decided which parent contributes its value to which child (Davis, 1991). Uniform crossover may bring more diversity into population as compared to one point and two point crossover. As a result, completely different and inferior children may be produced. However as noted by Davis (1991), for some problems the ability of uniform crossover to combine good features regardless of its location on the chromosome outweighs the destruction it could possibly bring when using it on two radically dissimilar chromosomes. Some initial experiments were conducted with the crossover operator the only variable. Uniform crossover was found superior to one point and two point crossover. For the logic of the GA in this implementation, the probability of crossover or crossover rate is a real value between 0 and 1. It is the probability of choosing an individual of the population as a parent for the formation of the crossover pool.

**Mutation**

In a simple GA, mutation is the probability of randomly altering the value of a string on a chromosome (Goldberg, 1989). For a binary chromosome which is the traditional mode of representation in GA, it would simply mean changing a 1 to 0 and vice versa. However in the models under consideration the representation is not binary, rather it is based on integer values. The operation of mutation on binary representation is better explained through an example. Let a gene be represented by the binary number 1111, the equivalent of which in decimal system is 15. A mutation operator will visit each bit of the gene and replace a 1 by 0 and vice versa if a probability test is passed. Let the mutation results in the child gene 1110 the equivalent of which in decimal is 14. If the result of mutation is 1100 then the decimal equivalent is 12 and if 1000 then it is 8. This means that each change in the value of bits at positions, starting from the right hand side make a difference equal to the multiples of 2, except the right most bit. The same idea of changing digits at different level within individual genes, rather than randomly changing the whole gene, is adopted for the integer chromosomal representation used here. However, this requires a different mutation operator.

The mutation operator designed for the present models is close to the non-uniform mutation operator presented by Michalewicz (1992) for real valued representation and the creep-mutation by Beasley et al. (1993b). In the non-uniform mutation a certain amount is either added to or subtracted from genes when a probability test is passed. This amount is equal to the difference of the gene value and either upper or lower domain bounds set for the genes. This amount is generation dependant, and is calculated in a manner that the probability of it being close to 0 increases as generation increases. In contrast, the creep-mutation adds or subtracts a small, randomly generated amount. Influenced by these ideas, the mutation operator adopted for the present models mutates the third digit from right (hundreds) i.e. add or subtract 100 from the gene if a probability test is passed, for the first 500 generations. It mutates the second digit (tens) by adding or subtracting 10 until 1250 generations. Similarly, the operator mutates the first digit (units) by adding or subtracting 1 if a probability test is passed for all generations, from the first generation till the end of generations. The mutation operator visits every single chromosome in the population and then mutates its genes by a given mutation rate. The mutation rate is interpreted as the chance of mutation of a given gene. However, this modified mutation operator is only applied to the second part of the chromosome representing the scheduled start time (i.e. scheduled start time vector) in the stream tube GA. The first part representing machines (i.e. machine vector) is randomly mutated within the range of the number of machines available. Each gene in the machine vector is visited and is replaced with a random number within the range of the number of machines available if the probability test is passed. This strategy of using the simple random mutation for the machine vector and the modified mutation operator for the scheduled start time vector of the chromosome showed superiority over other mutation operators used during initial experiments conducted for the purpose. The modified mutation operator works well with time block GA as well.

**Termination**

The number of generations has been used as the termination criteria for the single machine models presented earlier in Section 10. Although it is simple and easily implemented, however the drawback is that if the best solution is found in the early generations, the iterations still continue unnecessarily till the end. For this reason an early stopping criteria is used for the current models. The improvement in the fitness function is monitored over 1000 generations. If the improvement is less than or equal to 0.001%, the programme is terminated otherwise it continues until the given maximum number of generation is reached.

## 11.3 Comparison of stream tube GA and time block GA

The performance of the stream tube GA and the time block GA is compared to establish which approach can better handle the problem of irrigation scheduling under an arranged demand system. For this purpose both approaches were tested for the single goal objective of minimizing earliness/tardiness under certain capacity restriction, and the dual goal objective of minimizing earliness/tardiness and channel capacity. However, for the single goal objective the fitness function for both the stream tube GA and the time block GA as used for the dual goal objective needs to be modified. For the single goal objective channel capacity minimization is not a part of the objective and fitness function. The fitness function in the stream tube GA for a single goal objective is represented by the following equation.

$$\text{Minimize } F = [\{\sum_{\boldsymbol{j}=1}^{\boldsymbol{J}} (\boldsymbol{E}_{\boldsymbol{j}} + \boldsymbol{T}_{\boldsymbol{j}})\} + R_C P_C + R_O P_O] \tag{11.16}$$

where $R_C$ = penalty weight for $P_C$; and $R_O$ = penalty weight for $P_O$. Equation (11.16) shows the fitness function for stream tube approach without any term for channel capacity. Any violation of the capacity is controlled by the penalty ($P_C$) in the fitness function. Similarly, in the time block GA fitness function includes only the penalty for capacity constraint violation. Other constraints for time block approach are controlled by a repair strategy as discussed earlier. Mathematically the fitness function for the time block GA for a single goal objective is given by

$$\text{Minimize } F = [\{\sum_{\boldsymbol{j}=1}^{\boldsymbol{J}} (\boldsymbol{E}_{\boldsymbol{j}} + \boldsymbol{T}_{\boldsymbol{j}})\} + R_C P_C] \tag{11.17}$$

Three experiments were designed to evaluate the performance of the models across two different problem specific parameters i.e. demand-supply ratio and problem size. The objective was to establish whether the GA performance is consistent or not with increasing problem complexity. It was also aimed to have a detailed insight into the performance of the two approaches. A fourth experiment is designed to explore whether the GA multimachine scheduling model is able to help an irrigation manager in selection between sequential and simultaneous irrigation. The solutions generated by the integer programme formulation by Anwar and Clarke (2001) are used as benchmarks.

### 11.3.1 Experiment 1

This experiment is designed to test the quality of the solution of the simple multimachine GA models against demand for the single goal objective of minimizing earliness and tardiness under channel capacity restrictions. At higher levels of demand, the scheduling problem is assumed to become computationally more complex as there is less idle time available within the scheduling interval. To test this hypothesis, the parameter demand-supply ratio is introduced which is expressed as:

$$r_{DS} = \frac{Q_{max}}{Q} \tag{11.18}$$

where $r_{DS}$ = demand-supply ratio. The demand-supply ratio is a measure of the surplus capacity available in the irrigation schedule. It is similar to the interval-makespan ratio used by Anwar and De Vries (2004). The latter is only applicable when outlets operate sequentially, whereas the demand-supply ratio is applicable when outlets operate simultaneously as is the present case. In contrast to the interval-makespan ratio, the demand-supply ratio ranges from 0 to1.

Table11.1 summarizes the parameters used to generate data for the experiment. For this experiment, there are 8 outlets to be serviced, each with 1 unit of discharge, and the channel capacity is 4 units of discharge i.e. four outlets can be operated simultaneously. The irrigation interval within which all outlets must be serviced is 800 units of time. The duration each outlet is operated is a uniformly distributed random number over the range 0-400. The target start time of each outlet is also a uniformly distributed random number over the range 0-800. For each demand-supply ratio in Table 11.1, a test instance was generated. A test instance consists of 8 uniformly distributed random numbers representing the duration of each outlet and 8 uniformly distributed random numbers representing the target start time. The ranges for these numbers are shown in Table 11.1. If the sum of target start time and duration for any outlet exceeded the irrigation interval the test instance is rejected and a new test instance is generated. For a given instance, from the generated durations of each outlet the demand-supply ratio is calculated. If this lies within a tolerance of +0.1% of the demand-supply ratio in Table11.1, the test instance is retained, otherwise it is rejected and a new test instance is generated. This process is repeated until 140 instances are produced for each demand-supply ratio.

**Table 11.1. Problem parameters for Experiment 1:**

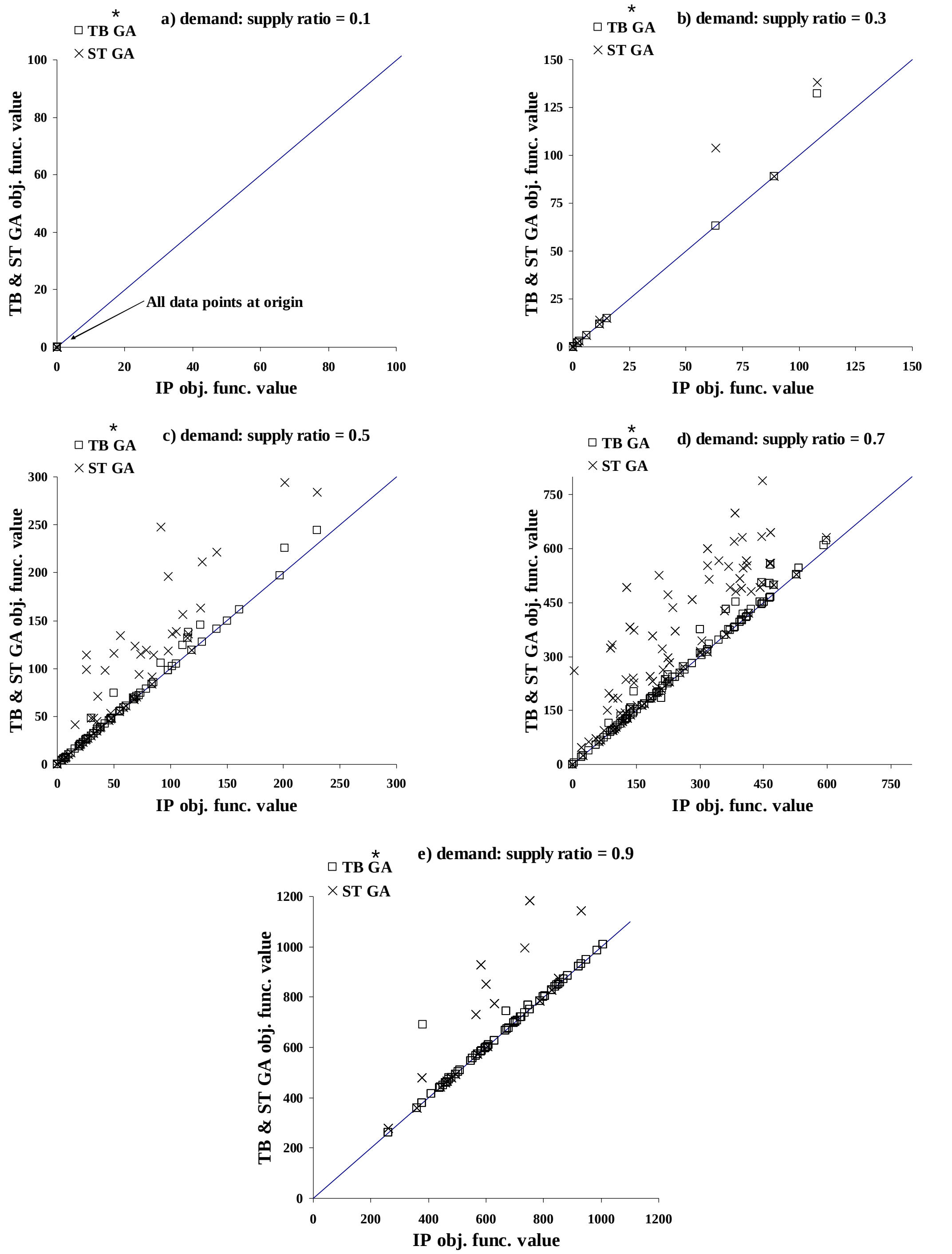
| | |
|---|---|
| Number of outlets (jobs) | 8 |
| Channel capacity | 4 |
| Irrigation interval | 800 |
| Duration of each outlet | Uniformly distributed random integer from the range (0 ,400) |
| Target start time | Uniformly distributed random integer from the range (0 ,800) |
| Demand-supply ratio | 0.1, 0.3, 0.5, 0.7, 0.9 |
| Number of instances | 140 for each demand-supply ratio |

Both the IP and the GA models are run using this test data. The IP is terminated when it obtains a global optimum, whereas the GA has different stopping criteria. Both the GA models are terminated if the improvement in the objective function value is less than or equal to 0.001 % over 1000 generations otherwise they are allowed to run till the maximum number of generation is reached. The generation number at which the best solution is found is recorded. Initial tests were conducted to get a better combination of algorithm parameters for GA. The algorithm parameters used for the GA models are as given in Table 11.2. It was found that the time block GA performance was not very sensitive to mutation and crossover rates at different demand-supply ratios and hence the same mutation rate of 0.2 and crossover rate of 0.75 were used through out the experiments for the time block GA. However the stream tube GA was found to be more sensitive to these parameters and hence repeated tests were conducted at each demand-supply ratio for a range of mutation and crossover rates. Table 11.2 presents only the algorithm parameters that generated the best results. The criteria for best is based on the number of optimum solutions (IP solutions). The more the optimum solutions and the fewer the number of infeasible solutions the better the performance is.

**Table 11.2. Algorithm parameters for Experiment 1:**

| Parameters | Time Block GA | Stream tube GA (parameters at different demand-supply ratio) | | | | |
|---|---|---|---|---|---|---|
| | | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 |
| Population size | 100 | 100 | | | | |
| Probability of mutation | 0.2 | 0.3 | | | | |
| Probability of crossover | 0.75 | 0.75 | 0.75 | 0.75 | 0.8 | 0.85 |
| Max. number of generations | 2500 | 5000 | | | | |

Figure 11.6 (a-e) shows the objective function values of the GA models relative to that of the IP. For very low demand: supply ratios (0.10), it is relatively easy to find a solution. At this low level of demand, each of the 140 schedules can be prepared to deliver water to each outlet at the target start time and therefore the earliness/tardiness in every schedule is zero, and all schedules plot at the origin in Figure 11.6a. A detailed analysis of results for Experiment 1 is presented in Table 11.3. The relative error as used for the single machine models can not be used for this experiment as the IP obtains schedules with 0 earliness/tardiness which results in the division by 0 errors while calculating the relative error for the GA models. Therefore, the average absolute differences between the total earliness/tardiness values (i.e. the objective function values) of the schedules developed by the GA models and that of the IP (or the deviation from the optimum schedule) are divided by the number of outlets, as given in Table 11.3. In order to put the absolute difference into context it is necessary to consider an individual instance rather than averages. For one particular instance the objective function value from the IP is 228 against that obtained by the GA of 242. The difference is therefore 14 units of time. If the interval of 800 days is taken to represent 8 days i.e. all times, durations are rounded off to 1/100th of a day (approximately 15 minutes), then a total earliness/tardiness of 14 time units is approximately 210 minutes, or given there are 8 outlets, 26 minutes per outlet.




*TB GA = Time block GA; ST GA = Stream tube GA; IP = Integer programming

**Figure11.6** Simple multimachine GA models vs. IP, at various demand: supply ratios

**Table 11.3. Analysis of results for Experiment 1**

| Models | Statistics (GA vs. IP) | Demand/Supply ratio 0.10 | 0.30 | 0.50 | 0.70 | 0.90 | *Average |
|---|---|---|---|---|---|---|---|
| **Stream Tube GA** | Avg. diff./No. of outlets | 0 | 0 | 2 | 14 | 13 | 6 |
| | St. Deviation | 0 | 4 | 54 | 139 | 137 | 67 |
| | Optimum solutions/140 | 140 | 137 | 106 | 37 | 21 | 88 |
| | Feasible solutions/140 | 140 | 140 | 140 | 140 | 12 | 18 |
| | Average generations | 560 | 690 | 774 | 1126 | 675 | 765 |
| | Computation time/instance (s) | 7 | 16 | 7 | 9 | 7 | 9 |
| **Time Block GA** | Avg. diff./No. of outlets | 0 | 0 | 0 | 1 | 1 | 0 |
| | St. Deviation | 0 | 2 | 5 | 19 | 28 | 11 |
| | Optimum solutions/140 | 140 | 139 | 129 | 96 | 124 | 126 |
| | Feasible solutions/140 | 140 | 140 | 140 | 140 | 140 | 0 |
| | Average generations | 92 | 94 | 138 | 176 | 76 | 115 |
| | Computation time/instance (s) | 287 | 160 | 260 | 291 | 306 | 261 |
| **IP** | Optimum solutions | 140 | 140 | 140 | 140 | 140 | 140 |
| | Feasible solutions | 140 | 140 | 140 | 140 | 140 | 140 |
| | Computation time/instance (s) | 2 | 6 | 30 | 247 | 205 | 98 |

*Average values across all ratios rounded to the nearest integer.

The average number of optimum solutions obtained by the stream tube GA is 88 out of 140 while for the time block GA it is 126 out of 140. In addition to the lower number of optimum solutions obtained, the stream tube GA also produced 88 schedules that were infeasible at 90% demand: supply ratio. The performance of the stream tube GA deteriorated with increasing demand: supply ratio which is evident from the decreasing number of optimum solutions as the demand: supply ratio is increased from 10% to 90%. The differences in standard deviation of the objective function values between the stream tube GA and the IP are also higher than that of the time block GA and the IP. However the stream tube GA proved to be computationally more efficient than the time block GA. For example at the 90% demand: supply ratio, for the stream tube GA, the average solution time per instance is about 0.12 minutes while for the time block GA it is about 5 minutes per instance, even though the maximum generations set for the stream tube GA was 5000 as compared to 2500 for the time block GA. Stopping criterion for both approaches was kept the same as discussed earlier. However the solution quality of the time block GA was much better and comparatively more consistent with the same set of GA parameters across all tests.

The time block GA performed better at the 90% demand: supply ratio, obtaining 124 optimum schedules out of 140 as compared to 96 at the 70% demand: supply ratio. One possible explanation could be that the number of feasible schedules in the 90% problem is very limited because of the fewer free space available (i.e. only 10%) for the GA. The opportunity to find an optimum schedule is comparatively higher among a few feasible schedules as compared to the large number of feasible schedules at low demand: supply ratios. The rapid convergence of the time block GA at a 90% demand: supply ratio as compared to its slower convergence at low demand: supply ratios indicates that the time block GA has less difficulty in finding optimum schedules at 90% demand: supply ratio. It may be inferred that the ratio of idle time inserted and its distribution in a schedule is a significant factor in GA performance.

Table 11.3 provides evidence that the time block GA performed better than the stream tube GA across a range of demand: supply ratios and that the performance of the stream tube GA was more sensitive to the demand: supply ratio than the time block GA. It is concluded that the time block GA is better than the stream tube GA for dealing with the simultaneous irrigation scheduling problem, although the stream tube GA is computationally more efficient than the time block GA.

### 11.3.2 Experiment 2

This experiment is designed to test the quality of the solution of the time block GA and the stream tube GA as the problem size (number of outlets/jobs) increases, for the single goal objective of minimizing earliness/tardiness under channel capacity restrictions. Table 11.4 summarizes the parameters used to generate test data for this experiment. The duration each outlet is operated is a uniformly distributed random number over the range 0-400 (half of the irrigation interval). The target start time of each outlet is also a uniformly distributed random number over a range equal to the irrigation interval. If the sum of target start time and duration for any outlet exceeded the irrigation interval the test instance is rejected and a new test instance is generated. For each problem size, a data set was generated with the parameters shown in Table 11.4 and the demand-supply ratio calculated using equation (11.8). If the calculated ratio is within a tolerance of +0.1% the test instance is retained, otherwise it is rejected and another test instance generated. This process is repeated to obtain 140 test instances for a problem with 8 jobs (outlets), 10 jobs and 12 jobs. For both the GA models the initial population and the maximum generations are increased pro-rata with the increase in problem size. For example in the time block GA a population size of 100 for 8 jobs is increased to a population size of 125 and 150 for 10 and 12 jobs respectively (Table 11.5). Similarly the maximum number of generations is increased from 2000 for problem size 8 to 2500 for problem size 10, and 3750 for problem size 12, though an early-stopping criteria as used in Experiment 1 is again used to prevent unnecessary iterations.

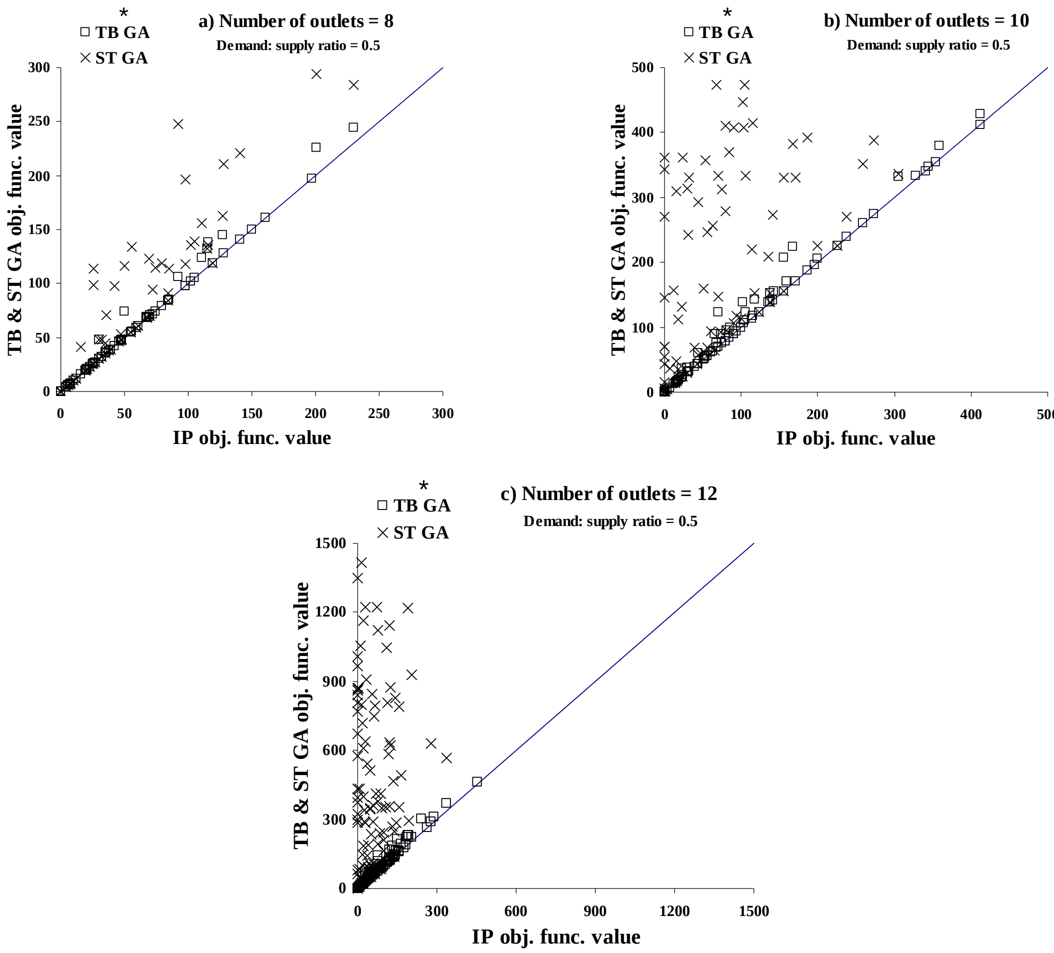
**Table 11.4. Problem parameters for Experiment 2:**

| **Parameters** | **Stream tube GA and Time Block GA** | | |
|---|---|---|---|
| Number of outlets (jobs) | 8 | 10 | 12 |
| Channel capacity | 4 | 4 | 4 |
| Irrigation interval | 800 | 1000 | 1200 |
| Duration of each outlet (Uniformly distributed random integer from the range): | (0,400) | (0,400) | (0,400) |
| Target start time (Uniformly distributed random integer from the range): | (0,800) | (0,1000) | (0,1200) |
| Demand-supply ratio | 0.5 | 0.5 | 0.5 |
| Number of instances | 140 | 140 | 140 |

**Table 11.5. Algorithm parameters for Experiment 2:**

| **Parameters** | Time Block GA | | | Stream tube GA | | |
|---|---|---|---|---|---|---|
| Number of outlets (jobs) | 8 | 10 | 12 | 8 | 10 | 12 |
| Population size | 100 | 125 | 150 | 100 | 125 | 150 |
| Probability of mutation | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.09 |
| Probability of crossover | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 |
| Max. number of generations | 2000 | 2500 | 3750 | 5000 | 6250 | 7500 |

A summary of the best algorithm parameters selected during initial experiments is given in Table 11.5. Using the IP and the GA models, schedules were obtained for each of the data sets and the objective function values compared. Due to the excessive time the IP takes to solve larger problem sizes the range of problem sizes was limited to 8, 10 and 12 jobs only. Figure 11.7 (a-c) shows the objective function values of the GA models relative to that of the IP at various problem sizes. As the problem size increases the objective function values of the stream tube GA plot farther away from the line of perfect, indicating deviation from optimum solutions, however the time block GA plots close to the optimum solutions, across the problem size. Table 11.7 presents a detailed analysis of the results from Experiment 2.




*TB GA = Time block GA; ST GA = Stream tube GA; IP = Integer programming

**Figure11.7** Simple multimachine GA models vs. IP, at various problem sizes

**Table 11.6. Analysis of results for Experiment 2**

| Models | Statistics (GA vs. IP) | Number of jobs 8 | 10 | 12 | *Average |
|---|---|---|---|---|---|
| **Stream Tube GA** | Avg. diff./No. of outlets | 2 | 11 | 37 | 17 |
| | St. Deviation | 50 | 166 | 517 | 244 |
| | Optimum solutions/140 | 106 | 62 | 31 | 66 |
| | Feasible solutions/140 | 140 | 140 | 134 | 2 |
| | Average generations | 790 | 1755 | 1183 | 1243 |
| | Computation time/instance (s) | 11 | 25 | 31 | 22 |
| **Time Block GA** | Avg. diff./No. of outlets | 0.13 | 0.25 | 0.42 | 0.26 |
| | St. Deviation | 5 | 8 | 14 | 9 |
| | Optimum solutions/140 | 129 | 116 | 104 | 116 |
| | Feasible solutions/140 | 140 | 140 | 140 | 140 |
| | Average generations | 138 | 298 | 503 | 313 |
| | Computation time/instance (s) | 298 | 509 | 1186 | 664 |
| **IP** | Optimum solutions | 140 | 140 | 140 | 140 |
| | Feasible solutions | 140 | 140 | 140 | 140 |
| | Computation time/instance (s) | 30 | 1003 | 2030 | 1021 |

*Average values across all problem sizes rounded to the nearest integer.

The average number of optimum solutions across all problem sizes found by the stream tube GA is 66 out of 140. The stream tube GA also resulted in 6 infeasible schedules for the problem size 12. The average number of optimum solutions across all problem sizes found by the time block GA is 116 out of 140. The time block GA did not result in an infeasible schedule in any problem size. The average difference of the earliness/tardiness from the optimum solutions is divided by the number outlets in the schedules to make it a uniform measure of performance across the range of outlet numbers. The average difference of the earliness/tardiness from the optimum solutions per outlet for the stream tube GA is 17 units of time while that for the time block GA is 0.26 units of time. With an increase in problem size, the performance of the stream tube GA deteriorated more than the time block GA. The stream tube GA in contrast to the time block GA is more sensitive to the magnitude of the penalty weights in the fitness function. Both the GA models showed deterioration in solution quality with increasing problem size which is evident from the decreasing number of optimum solution and increasing standard deviation with increasing problem size. However the deterioration in solution quality in the time block GA is considerably less than that in the stream tube GA.

The low performance of the stream tube GA compared to the time block GA could be the result of the additional complexity in the stream tube GA of assigning a job to a specific machine. This assignment of jobs to specific machines i.e. determination of which job is operated on which machine makes it difficult for the stream tube GA to find optimum solutions. Also the stream tube GA chromosome in contrast to the time block GA is a concatenation of two different vectors which do influence each other when it comes to making a schedule and determining its fitness. And that also seems to be the reason of its sensitivity to GA parameters. Based on this argument it may be inferred that this artificial arrangement of machines (stream tubes) and jobs (outlets) in the stream tube GA, and as pursued by several other researchers for simultaneous irrigation scheduling problem, is counterproductive for GA. What we really need to know in an irrigation environment is the number of outlets operated simultaneously at any time instance, and that is what the time block GA is based on. However, the ability of the stream tube GA to assign jobs to specific machines makes it more flexible and convenient to incorporate additional problem parameters like travel time. Based on the ability of finding optimum solutions calculated from the numbers of optimum solutions found by the time block GA (as reported in Table 11.3 and Table 11.6), the time block GA proved to be more reliable with 87% (90% in Experiment 1 and 83% in Experiment 2) in contrast to 55% (63% in Experiment 1 and 47% in Experiment 2) by the stream tube GA.

### 11.3.3 Experiment 3

This experiment is designed to test the quality of the solution of the GA models (i.e. the time block GA and the stream tube GA) as the problem size (number of outlets) increases, for the dual goal objective of minimizing earliness/tardiness and discharge. In this experiment with the dual goal objective, the demand-supply ratio is kept at 90%. The range of different ratios as used in Experiment 1 is not considered here because for the dual goal objective, the demand-supply ratio is not considered a relevant problem parameter. For example, a tertiary unit in an irrigation system having the sum of durations of all (outlets) jobs equal to 1600 time units and irrigation interval equal to 800 time units has a 50% demand-supply ratio if the channel is operated with 4 units of discharge as illustrated in Figure 11.8a.

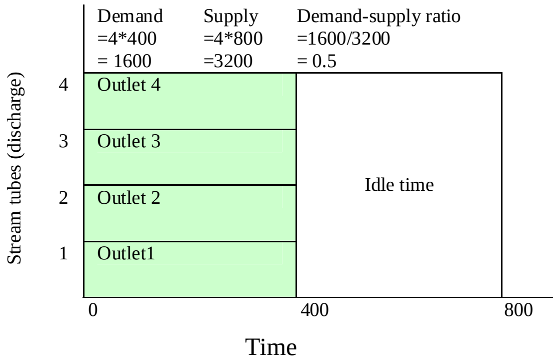



**Figure 11.8a** Channel discharge 4 units and demand-supply ratio 0.5.

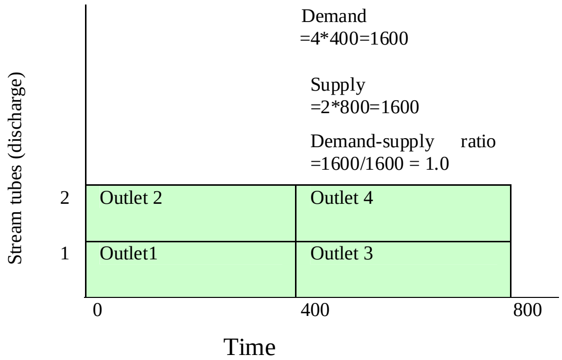



**Figure 11.8b** Channel discharge 2 units and demand-supply ratio 1.0

In Figure 11.8a outlets 1 to 4 all have the same durations and are all serviced simultaneously. If each outlet requires 1 unit of discharge then the supply channel should be operated with 4 units of discharge. It is clear from figure 11.6a that half of the irrigation interval with 4 units of discharge is not utilized. However, if the dual goal objective model minimizes the capacity to 2 units of discharge which is also a feasible solution then the schedule becomes a 100% demand-supply ratio problem as illustrated in Figure 11.8b. To circumvent this problem to a certain extent, the GA models with the dual goal objective of minimizing earliness/tardiness and discharge is tested only against increasing problem size at the fixed demand-supply ratio of 90%. With 90% demand-supply ratio the channel capacity can only be minimized at most to the units of discharge for which the problem is defined to be a 90% demand-supply ratio problem. For example, consider a tertiary unit with the sum of durations equal to 2900 units of time and 800 units of time irrigation interval as illustrated in Figure 11.9. The supply channel under this scenario can only be run at 4 units of discharge at minimum to be a 90% demand-supply ratio problem. Operating the channel at any discharge, less than 4 units of discharge say 3 makes the supply less than the demand. It is still possible that the optimum solution found is with a discharge requirement more than 4 units, thus lowering the demand-supply ratio. However, the optimum solutions still lie close to 90% demand-supply ratio. For example, operating the channel with 5 units of discharge makes the supply 4000 units of volume against a demand of 2900 units of volume, which results in a lower demand-supply ratio of 73%. The 90% demand-supply ratio is also found in some real world examples e.g. the one presented by Suryavanshi and Reddy (1986) which further justifies the use of 90% demand-supply ratio.

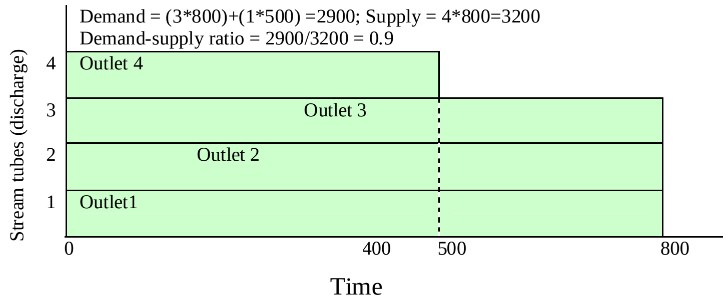



**Figure 11.9** Channel discharge 4 units and demand-supply ratio 0.9.

The data set for this experiment was generated in a similar manner as Experiment 2 but with some modification. For Experiment 3 all the instances are at 90% demand-supply ratio. The durations are uniformly distributed numbers over a range between 0 and half of the irrigation interval for the respective problem. Problem parameters for Experiment 3 are given in Table 11.7. The comparison of the dual goal objective GA models with IP dual goal objective is not as straight forward as the single goal objective. In the dual goal objective two parameters needs to be compared i.e. the discharge as well as earliness/tardiness. The models presented here give priority to discharge minimization over earliness/tardiness minimization. Therefore both the stream tube GA and the time block GA are first compared with IP for discharge. If the discharge is minimized to the same value by these models as IP, only then earliness/tardiness values can be compared.

Algorithm parameters for Experiment 3 are given in Table 11.8. To test the stream tube GA against IP several combination of mutation and crossover rates were evaluated. The two mutation rates of 0.2 and 0.3 which had proved useful in Experiment 1 and 2 were also used for Experiment 3. A range of crossover rates (0.75, 0.8, 0.85, and 0.9) were tested, however, only results for the best are presented. The initial population size was increased as the problem size increased. The population size was kept 100 for problem with 8 jobs, 150 for problem with 10 jobs, and 200 for 12 jobs. The maximum generations were kept at 20,000 which proved sufficient for all problem sizes. However the same stopping criteria as used in

Experiment 1 and 2 was also adopted for Experiment 3, which allowed the programme to terminate if no improvement in solution quality was recorded.

**Table 11.7. Problem parameters for Experiment 3:**

| Parameters | Stream tube GA and Time Block GA | | |
|---|---|---|---|
| Number of outlets (jobs) | 8 | 10 | 12 |
| Channel capacity | 4 | 4 | 4 |
| Irrigation interval | 800 | 1000 | 1200 |
| Duration of each outlet (Uniformly distributed random integer from the range): | (0,400) | (0,500) | (0,600) |
| Target start time (Uniformly distributed random integer from the range): | (0,800) | (0,1000) | (0,1200) |
| Demand-supply ratio | 0.9 | 0.9 | 0.9 |
| Number of instances | 100 | 100 | 100 |

**Table 11.8. Algorithm parameters for Experiment 3:**

| Parameters | Time Block GA | | | Stream tube GA | | |
|---|---|---|---|---|---|---|
| Number of outlets | 8 | 10 | 12 | 8 | 10 | 12 |
| Population size | 100 | 125 | 150 | 100 | 150 | 200 |
| Probability of mutation | 0.3 | | | 0.2, 0.3 | | |
| Probability of crossover | 0.75 | | | 0.75, 0.8, 0.85, 0.9. | | |
| Max. number of generations | 1500 | 2000 | 2500 | 20,000 | | |

For the time block GA a limited number of algorithm parameters were tested for two reasons

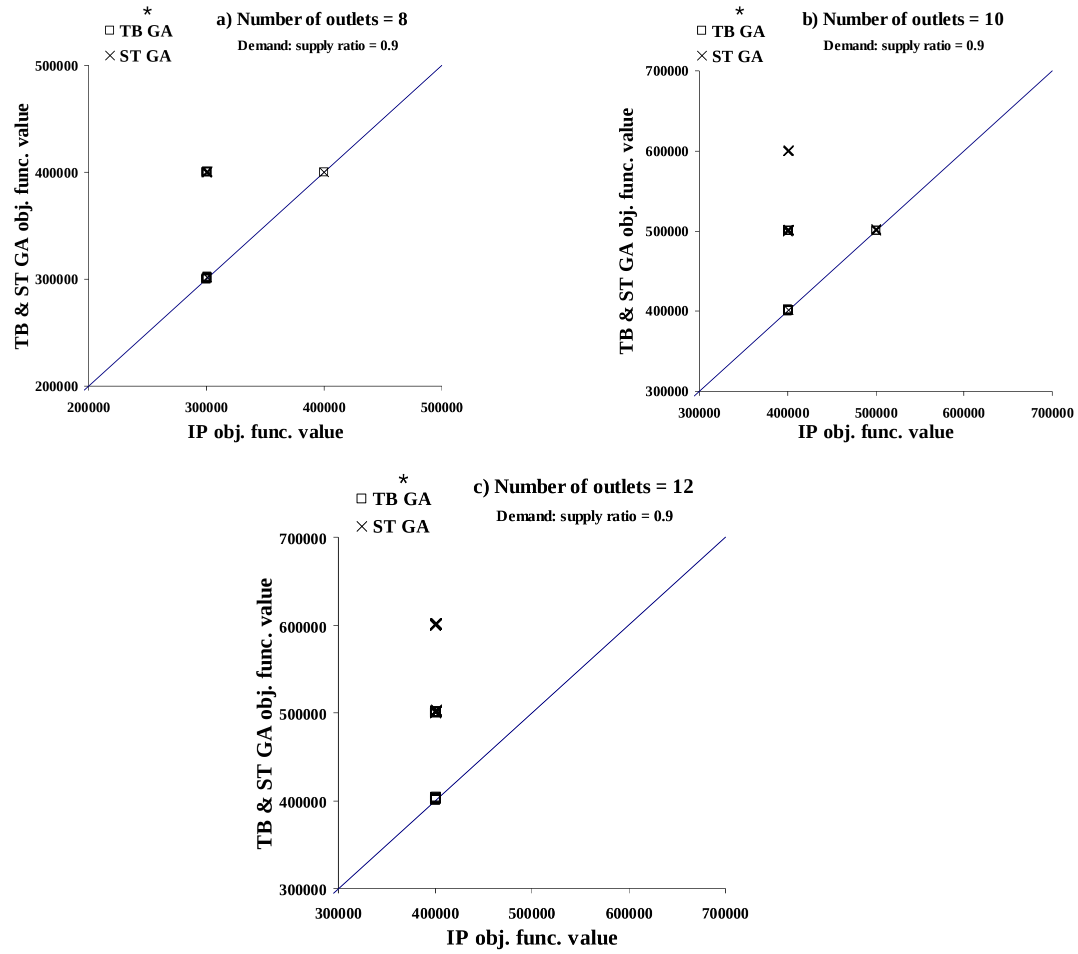
- it takes longer to execute as compared to the stream tube GA,
- it proved consistent over a range of parameters in all the previous experiments.

The crossover rate used was 0.75 and mutation rate as 0.3. The population size was kept 100 for problem with 8 jobs, 125 for problem with 10 jobs, and 150 for 12 jobs. The maximum generations set for 8, 10, and 12 problem sizes were 1500, 2000, and 2500 respectively. The maximum limit of generations for the time block GA is less than the stream tube GA because initial experiments proved these limits sufficient and in most of the cases the time block GA finds the best solutions in very early generations. The same early stopping criteria as used in the stream tube GA was adopted for the time block GA as well.

Figure 11.10 (a-c) shows the objective function values of the GA models relative to that of the IP at various problem sizes. The models tested in this experiment have the dual goal objective of minimizing discharge and earliness/tardiness. The objective function is set so that it gives priority to the goal of discharge minimization. So the objective function values are first compared for discharge and then with in the same discharge for earliness/tardiness. However, the variation in earliness/tardiness for all those instances having the same discharge can not be shown in the graph as the objective function values for these instances overlap. The graphs in Figure 11.10 shows that as the problem size is increased the performance of the stream tube GA deteriorates more than the time block GA. For example, it can be seen in Figure 11.10c that majority of the stream tube GA solutions are at the 600000 mark (on y-coordinate), indicating schedules with 6 units of discharge, in contrast to the time block GA which correspond to 500000 and 400000 marks, indicating 5 and 4 units of discharge. A detailed analysis of the results for Experiment 3 is given in Table 11.9a and 11.9b.

For a problem with 8 jobs to schedule, the stream tube GA was able to find 1 optimum solution out of the 100 test instances (with mutation rate of 0.3 and a crossover rate of 0.9 from the range of algorithm parameters tested). In contrast, the time block GA was able to find 13 optimum solutions out of 100 and found a total of 41 solutions with 3 units of discharge against 99 by the IP and 1 by the stream tube GA. The total number of schedules found by the time block GA with the same discharge requirements as IP was 42 out of 100. The average earliness/tardiness (mean error) per outlet for the 42 schedules was 77 time units as compared to 71 time units from IP schedules an error of 8.6%.

For schedules with 10 jobs, the stream tube GA could only minimize 1 instance to 4 units of discharge compared to 96 instances by the IP (at 0.2 mutation and 0.85 crossover rates). All other combinations of mutation and crossover rates did not produce any schedule with 4 units of discharge. The stream tube GA was unable to find any optimum solutions. In contrast the time block GA was able to find 10 optimum solutions out of 100 instances. The time block GA also found 41 schedules with 4 units of discharge against 96 by IP and 1 by the stream tube GA. The total number of schedules found by the time block GA with the same discharge as IP was 45 out of 100. The average earliness/tardiness per outlet for these 45 schedules was 94 time units as compared 83 time units for schedules for these instances solved by the IP – error of 13.4%.




***TB GA = Time block GA; ST GA = Stream tube GA; IP = Integer programming**

**Figure11.10** Simple multimachine dual goal GA models vs. IP, at various problem sizes

**Table 11.9a.** Analysis of results for Experiment 3

| Problem size | Algorithm | Instances (Total =100) | | | No. of optimum | No. of feasible | Avg. diff./No. of outlets |
|---|---|---|---|---|---|---|---|
| | | 3 units of discharge | 4 units of discharge | 5 units of discharge | | | |
| 8 | GA(stream tube) | 1 | 99 | 0 | 1 | 100 | - |
| | GA(time block) | 41 | 59 | 0 | 13 | 100 | 77 |
| | IP | 99 | 1 | 0 | 100 | 100 | 71 |
| 10 | GA(stream tube) | 0 | 1 | 84 | 0 | 100 | - |
| | GA(time block) | 0 | 41 | 59 | 10 | 100 | 94 |
| | IP | 0 | 96 | 4 | 100 | 100 | 83 |
| 12 | GA(stream tube) | 0 | 0 | 0 | 0 | 100 | - |
| | GA(time block) | 0 | 25 | 75 | 2 | 100 | 107 |
| | IP | 0 | 43 | 0 | 43 | 89 | 65 |

**Table 11.9b** Computation time/instance (s)

| **Models** | Number of jobs | | |
|---|---|---|---|
| | 8 | 10 | 12 |
| **Stream Tube GA** | 8 | 18 | 28 |
| **Time Block GA** | 197 | 577 | 995 |
| **IP** | 14 | 2888 | 10000 |

For schedules with 12 jobs the IP only solved 43 of the 100 instances to a global minimum within the 3 hour computation time limit. The stream tube GA was unable to minimize the discharge to that obtained by the IP for any instance. No optimum solution was found with any of the 8 different combinations of GA algorithm parameters. The time block GA was able to find 2 optimum solutions out of 43 instances. The total number of schedules found by the time block GA with the same discharge as IP was 14 out of 43. The average earliness/tardiness per outlet for these 14 schedules was 107 time units as compared to 65 time units from IP schedules, - error of 64.8%.

The above results for the dual goal objective of minimizing both earliness/tardiness and discharge indicates the deterioration in solution quality of GA models as the problem size increases. The performance of the stream tube GA is poorer than with a single goal objective of minimizing earliness/tardiness under a restricted capacity. The time block GA shows better performance than the stream tube GA. However, again with the dual goal objective the performance deteriorates as compared to its performance with the single goal objective.

Based on the results and above arguments, the time block GA is a preferred choice over the stream tube GA for simultaneous irrigation scheduling problems with identical discharge for all outlets. However the use of the stream tube GA can still not be excluded altogether and it may occur to be a better tool provided some special problem specific mutation, crossover operator, and problem representation scheme are developed. A general conclusion drawn from the above discussion is that the time block GA can be a useful decision support tool in managing an arranged demand irrigation system.

### 11.3.4 Experiment 4

The purpose of this experiment is to compare the performance of the time block GA against the non-contiguous single machine GA model referred to as Model 1 (presented in Section 10.1.1). The objective is to establish whether the time block GA with the dual goal objective is able to solve a single machine problem with the same level of performance as a dedicated single machine model would do. In other words, it is intended to investigate whether the time block GA with the dual goal objective is able to help an irrigation manager make an optimum or near optimum schedule for sequential as well as simultaneous irrigations, excluding the need for having separate models for the two different schedules. For this purpose the time block GA with the dual goal objective is applied to the irrigation scheduling problems for which sequential or single machine optimum solutions exist. Both the time block GA and the Model 1 are applied to the same data for which single machine optimum (IP) solutions are known. To make an objective comparison the crossover, mutation, and other operators and all algorithm parameters as used for the time block GA are also used for the Model 1; therefore a separate GA implementation for Model 1 is not presented. Algorithm parameters that produced the best results in preliminary experiments and also in all the previous experiments are presented in Table 11.10. An early stopping criteria as used in the previous experiments, was used. According to the criteria the GA terminates if there is no improvement in solution quality over 1000 generations. The mathematical formulation for Model 1 has already been presented in Section 10.1.1 under the single machine models. The data for Experiment 4 is taken from Anwar and De Vries (2004). Anwar and De Vries (2004) used the parameters given in Table 10.10 for the experimental data generation. For each problem size 25 different instances were tested.

**Table 11.10 Algorithm parameters for Experiment 4**

| **Parameters** | **Model 1** and **Time Block GA** |
|---|---|
| Number of outlets/jobs | 8,10,12,15,20,25 |
| Population size | 100 |
| Probability of mutation | 0.3 |
| Probability of crossover | 0.75 |
| Max. number of generations | 10000, 15000 |

Figure 11.11 (a-f) shows the objective function values of the GA models relative to that of the IP at various problem sizes of only feasible schedules. Figure 11.11 shows that as the number of outlets is increased the values of the objective function of both the time block GA and Model 1 fall away from the line of perfect fit (optimum solutions), indicating a deterioration in solution quality. A detailed analysis of results for Experiment 4 is presented in Table 11.11.

For the 8 jobs problem, Model 1 obtained 10 optimum solutions while the time block GA obtained 11 optimum solutions out of 25. Both the models did not produce any infeasible schedule for the 8 jobs problem. For the 10 jobs problem, Model 1 obtained 2 optimum solutions while the time block GA obtained 6 optimum solutions. Model 1 did not produce any infeasible schedule while the time block GA obtained 2 infeasible schedules (2 machine solutions). For the 12 jobs problem, Model 1 again obtained 2 optimum solutions while the time block GA also obtained 2 optimum solutions. None of the solutions obtained by Model 1 was infeasible. In contrast, the time block GA obtained 3 infeasible schedules for the 12 jobs problem.

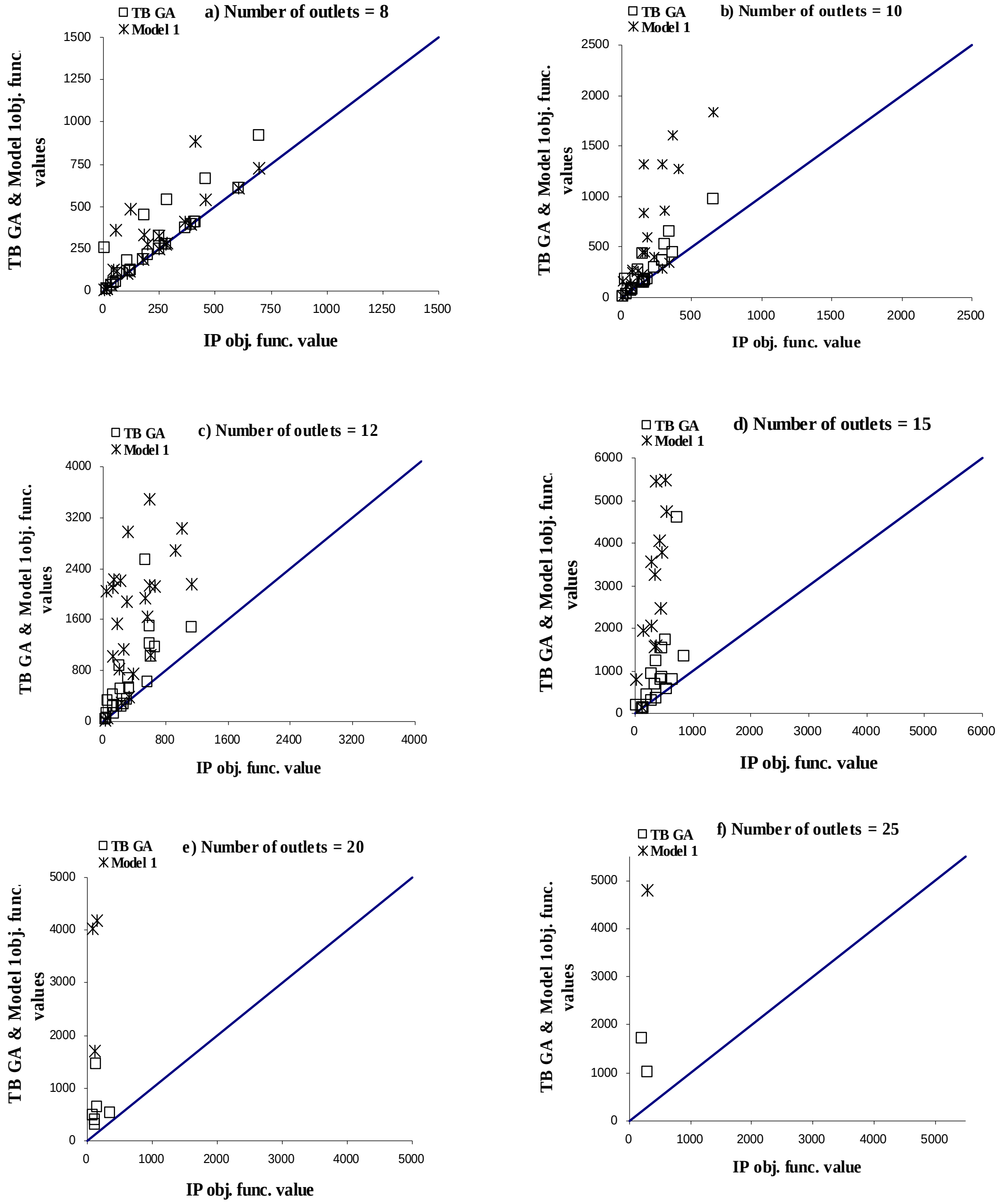



**Figure 11.11** Time block GA vs. Model 1

For the 15 jobs and larger problems, Model 1 did not obtain any optimum solution; however, the time block GA obtained 3 optimum solutions for the 15 jobs problem and no optimum solutions for problems with jobs more than 15. Model 1 obtained only 1 infeasible schedule while 8 of the solutions were infeasible by the time block GA for the15 jobs problem. For the 20 jobs problem 19 of the solutions obtained by Model 1 and the time block GA were infeasible. It is also worth mentioning that the IP did not solve within the allocated time of three hours for 2 of the 25 instances with 20 jobs.

Similarly, 9 of the 25 instances with 25 jobs did not reach a global optimum within the allocated time limit. For the 25 jobs problem, Model 1 did not obtain any feasible schedule while the time block GA obtained at least 2 feasible schedules. It may be concluded that the time block GA slightly performed better than Model 1 in finding optimum solutions which is also indicated by the graph in Figure 11.12.

**Table 11.11 Analysis of results for Experiment 4**

| Model | Statistics | Number of jobs | | | | | |
|---|---|---|---|---|---|---|---|
| | | 8 | 10 | 12 | 15 | 20 | 25 |
| **Model 1** | Optimum | 10 | 2 | 2 | 0 | 0 | 0 |
| | Mean error* | 89.6 | 287.7 | 1000.3 | 1311 | 3381.7 | - |
| | Infeasible/25 | 0 | 0 | 0 | 1 | 19 | 24 |
| **Time Block GA** | Optimum | 11 | 6 | 2 | 3 | 0 | 0 |
| | Mean error* | 21.5 | 63.2 | 541.2 | 185.7 | 362.1 | - |
| | Infeasible/25 | 0 | 2** | 3** | 8** | 19** | 23** |
| **IP** | Optimum | 25 | 25 | 25 | 25 | 23 | 18 |
| | Feasible | 25 | 25 | 25 | 25 | 25 | 25 |

* Mean error, relative to IP, of feasible schedules only, as calculated by (10.13).
** Two machines solutions (i.e. schedules with two simultaneous users)


**Figure 11.12** Time Block GA Vs. Model 1

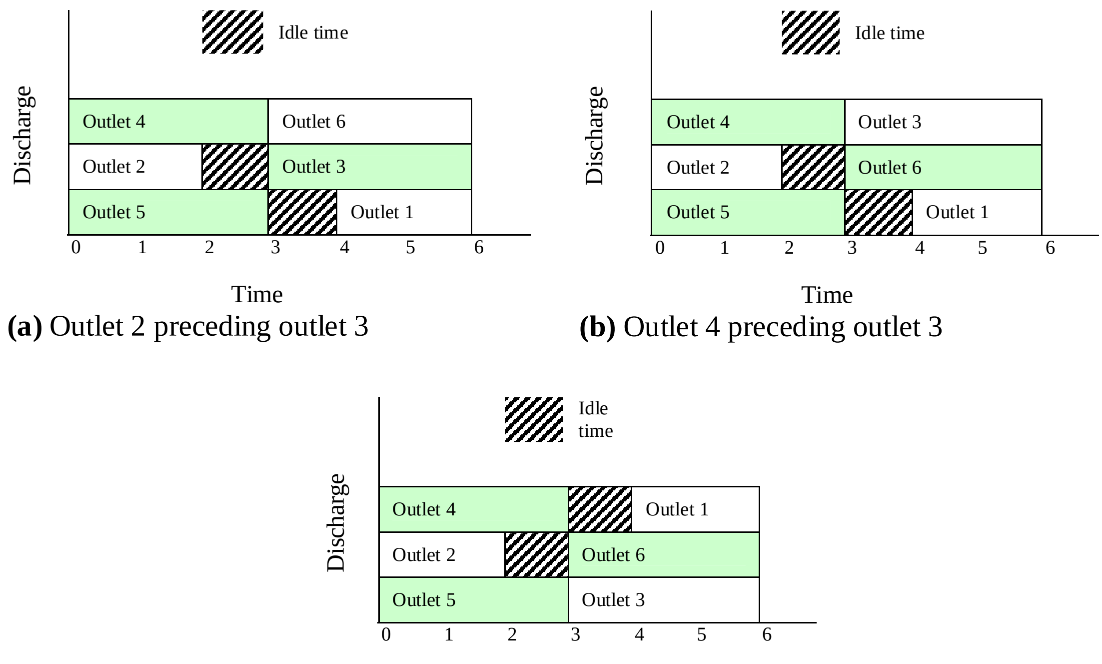
Over all, Model 1 and the time block GA for the non-contiguous scheduling problems did not perform as well as Model 2a, 2b, and 2c for the contiguous scheduling problems. In the non-contiguous scheduling problems idle time is allowed to be inserted between jobs which increase the complexity of the problem by increasing the search space. This is also evident from the increased solution time and the increased number of generations. The low performance of Model 1 and the time block GA for the non-contiguous problems as compared to Model 2a, 2b, and 2c, may be attributed to this increased complexity.

The objective of Experiment 4 was to determine whether the time block GA is able to minimize a schedule to a single machine, if it exists, thus excluding the need for using a stand alone single machine model for sequential irrigation scheduling. The time block GA was proved better than the dedicated single machine model (Model 1) in finding optimum solutions. Though the time block GA with the dual goal objective has proved to be applicable to both sequential as well as simultaneous irrigation problems, however, it needs further tuning of the algorithm parameters and the operators to enhance its performance for the sequential irrigation problems (single machine problems).

## 11.4 Simple multimachine with setup times

It has been shown that the multimachine scheduling problem with earliness/tardiness costs even without setup consideration is computationally very demanding and optimum solutions are not possible in practical time limits. The addition of sequence-dependent setup time and the dual goal of minimizing earliness/tardiness and the number of machines makes it further difficult, complicated, and novel.

As discussed in Section 7.3 sequence-dependent setup times are analogous to the irrigation water travel times between outlets in a canal irrigation system. Like in an industrial scheduling problem, the importance of travel time in an irrigation scheduling problem also can not be denied. Any feasible schedule without travel time may become an infeasible schedule when travel times are considered. However, the travel time in a simultaneous irrigation scheduling problem is even more complicated than the sequence-dependent setup times in an industrial scheduling problem. Several outlets are operated simultaneously and water is diverted not just from one outlet to another outlet, rather it is diverted from several outlets to several other outlets. In this situation it is extremely difficult to specify which outlet is followed by which other outlet and hence to determine the travel time. For travel time determination two absolute points are essential, whether these are any two outlets, or the head of the supply channel and any specific outlet. There is another issue with travel time also, and that is the travel time in a dry channel is different than in a partially filled channel. In simultaneous or multimachine irrigation scheduling problem, most of the time during the irrigation interval the channel is partially filled. In Figure 11.13 three different simultaneous irrigation schedules for 6 outlets are given.




**(a)** Outlet 2 preceding outlet 3

**(b)** Outlet 4 preceding outlet 3

**(c)** Outlet 5 preceding outlet 3

**Figure 11.13** Simultaneous irrigation schedules

Outlets are numbered so that outlet 1 represent the outlet near the head of the supply channel at the upstream and outlet 6 the last outlet downstream. In Figure 11.13a Outlet 3 is scheduled to receive water after outlet 2 has finished receiving water and an idle time of 1 time unit (i.e. at time 3). At the same time outlet 4 and 5 also stop to receive water. Now if travel time is to be considered for outlet 3, then it is a complex problem for any model, whether the stream tube GA or the time block GA. The problem for the time block GA would be the identification of the immediately preceding outlet because the time block GA does not assign outlets to specific stream tubes. As such the time block GA does not recognize the difference between the different positions of outlet 3 in Figure 11.13a, 11.13b, and 11.13c. It considers all the three schedules identical.

The stream tube GA would consider the travel time to outlet 3 only from the immediately preceding outlet on the same stream tube, which is outlet 2 in Figure 11.13a, outlet 4 in Figure 11.13b and outlet 5 in Figure 11.13c. In either case it is difficult to model the true picture, because in reality the travel time to any outlet not only depends on the position of the immediately preceding outlet on the same stream tube but also on the positions of the preceding outlets receiving water from other stream tubes. For example in Figure 11.13a the stream tube GA would consider travel time from outlet 2, although water is available at outlet 3 at time 3 and as such no travel time is required because both outlet 4 and 5 stop receiving water at time 3 and both are downstream of outlet 3. It is also worth mentioning that the supply channel is never completely dry which can be seen from the number of outlets serviced at any time in Figure 11.13 (a), (b), and (c). This explains the complex nature of the travel time in irrigation scheduling as compared to the setup time in machine scheduling.

Though, examples of multimachine scheduling with earliness/tardiness and setup times in an industrial environment can be found in literature. However, no work regarding simultaneous irrigation scheduling problem with the dual goal objective of minimizing earliness/tardiness and discharge with the additional complexity of travel time could be found. Although De Vries (2003) presented formulations for simple and complex multimachine problems with setup consideration using IP, however no results were reported because of the increased solution time. De Vries (2003) reported that including sequence dependent setup times into a model makes the model more complex and it was not possible to obtain solutions for multimachine models with setup times within a reasonable time (<24 hours). Similarly De Vries and Anwar (2006) described in detail the issues related to travel time and cited the relevant literature for travel time determination in a canal irrigation system. The models they presented considered travel time, however for single machine or sequential irrigation systems only. The model presented here in the current study is an improvement over De Vries and Anwar (2006), as it considers travel time in a multimachine or simultaneous irrigation system and is an improvement over De Vries (2003) as it resolves the issue of computational time by using approximate algorithm (GA) instead of IP.

### 11.4.1 Mathematical formulation

The time block GA and the stream tube GA discussed in the preceding sections, for solving the irrigation scheduling problem with the dual goal objective of minimizing earliness/ tardiness and discharge, were without travel time consideration. Although the time block GA was proved far better than the stream tube GA, however the time block GA in its present form is unable to handle issues related to travel time as discussed in the preceding paragraphs. The stream tube GA affords some flexibility in this regard and can be used to model the multimachine irrigation scheduling with travel time. The stream tube GA, however, may not be able to model the true scenario in its strictest sense and some assumptions have to be made to the problem in order to make the stream tube GA applicable.

These assumptions are:

- It has to be assumed that an outlet (job) or a group of outlets (jobs) are to be served by a specific stream tube/ machine.
- The channel is assumed dry when water travels from the head of the channel to any outlet or from an upstream outlet to a downstream outlet.
- Water is allowed to drain or is re-used when there is no supply to any outlet i.e. idle time is inserted. This assumption is very crucial as it has implication on the whole schedule. Ideally gates should be closed to cut off supply when water is not required to save unnecessary wastage of water. However, if gate is closed after irrigation supply to a particular outlet is completed then the travel time required for water to reach the next outlet scheduled to receive water on the same machine has to be calculated from the head of the supply channel, not the preceding outlet. This has also been implicitly assumed in the models presented by De Vries and Anwar (2006) that gates are open even if water is not required. Thus they considered travel time from the outlet directly preceding the current outlet even if idle time is inserted between the two. This makes incorporation of travel time in schedules easier and simpler.
- It is also assumed that the sequence of outlets on one machine does not influence the outlets on other machines.

Based on the above assumptions the stream tube GA presented in Section 11.1.1 can be fully adopted by modifying only the overlap constraint i.e. equation 11.10 to 11.12.

The modified overlap constraint for the current model is presented as follows.

$$P_O = \left(\sum_{j=1}^{J} \sum_{k=1}^{J} \mu_{jk} \cdot \gamma_{jk}\right) + \sum_{m=1}^{J} \chi_m \tag{11.19}$$

$$\mu_{jk} \quad = 1 \quad \text{if } M_j = M_k \quad \forall\, j, k \quad \text{where } j \neq k \tag{11.20}$$
$$= 0 \quad \text{otherwise.}$$

Equation (11.20) determines if more than one job are assigned to a machine. $\mu_{jk} = 1$ indicates that both jobs $j$ and $k$ are assigned to the same machine. The following equation determines whether any overlap exist between $j$ and $k$.

$$\gamma_{jk} \quad = 1 \quad \text{if } [\{ (S_j + D_j + T_{jk}) > \hat{S}_{jk} \} \;\forall\, k\} \;\forall\, j\,] \tag{11.21}$$
$$= 0 \quad \text{otherwise.}$$

where $j \neq k$ and $\boldsymbol{T}_{jk}$ = travel time from $j$ to $k$, and $\hat{S}_{jk}$ = start time of the job proceeding job $j$ and is defined by equation (11.22). $\gamma_{jk} = 1$ would mean that jobs $j$ and $k$ overlap each other. In other words the start time of the job proceeding job $j$ (i.e. job $k$) is less than the completion time of job $j$ plus the travel time from job $j$ to job $k$. Therefore the determination of jobs proceeding job $j$ is essential if travel time is to be considered and is done by the following equation.

$$\hat{S}_{jk} \quad = S_k \quad (\text{if } S_k > S_j \;\forall\, k) \;\forall\, j \quad (j \neq k) \tag{11.22}$$
$$= G^2 \quad \text{otherwise.}$$

The following equation (11.23) determines the infeasibility caused by the travel time from the head of the supply channel to the earliest job on a machine being greater than the start time of that job.

$$\chi_m \quad = 1 \quad \text{if } T_{0\underset{\sim}{K}_m} > S_{\underset{\sim}{K}_m} \text{ AND } S_{\underset{\sim}{K}_m} < G \quad \forall\, m \tag{11.23}$$
$$= 0 \quad \text{otherwise.}$$

where $\boldsymbol{T}_{0\,K_m}$ = travel time from the head of the channel to the earliest job on machine $m$, and $S_{K_m}$ = scheduled start time of the earliest job on machine $m$. The information of the index and the start time of the earliest job on machine $m$ are obtained from the matrix $\boldsymbol{\Omega}$ (dimension $J$x$J$) which is the result of the entry wise product of the scheduled start time row vector ($\boldsymbol{S}_J$) and every row of the matrix $\boldsymbol{\eta}$ (dimension $J$x$J$). Both the matrices $\boldsymbol{\eta}_{JJ}$ and $\boldsymbol{\Omega}_{JJ}$ are further illustrated through an example in Figure 11.14 and 11.15.

| $M_k$ | **4** | **1** | **3** | **1** |
|---|---|---|---|---|

| $\eta_{mk}$ | | *k* | | | |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 |
| *m* | 1 | *G* | 1 | *G* | 1 |
| | 2 | *G* | *G* | *G* | *G* |
| | 3 | *G* | *G* | 1 | *G* |
| | 4 | 1 | *G* | *G* | *G* |

**Figure 11.14** Matrix $\boldsymbol{\eta}_{JJ}$

| $S_k$ | **68** | **108** | **49** | **400** |
|---|---|---|---|---|

| $\Omega_{mk}$ | | *k* | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | $S_{K_m}$ | $K_m$ |
| *m* | 1 | *68*G* | 108 | *G*49* | 400 | 108 | 2 |
| | 2 | *68*G* | *108*G* | *49*G* | *400*G* | 49*G | 3 |
| | 3 | *68*G* | *108*G* | 49 | *400*G* | 49 | 3 |
| | 4 | 68 | *108*G* | *49*G* | *400*G* | 68 | 1 |

**Figure 11.15** Matrix $\boldsymbol{\Omega}_{JJ}$

Each element within the matrix $\boldsymbol{\eta}_{JJ}$ assumes a value of 1 if the $m^{th}$ machine services the $k^{th}$ job (alternatively, if machine $m$ is equal to an element $M_k$ of the machine vector) otherwise the element assumes a value equal to a large positive integer e.g. the irrigation interval ($G$) of the current model will suffice. As shown in Figure 11.14, $m$= 1 (i.e. machine 1) is equal to $M_2$ and $M_4$ of the machine vector presented in the figure. Thus $\eta_{1,2}$ and $\eta_{1,4}$ assume a value of 1. Similarly $m$= 2 (i.e. machine 2) is not equal to any element of the machine vector indicating that machine 2 is not used. Thus $\eta_{2,1}$, $\eta_{2,2}$, $\eta_{2,3}$ and $\eta_{2,4}$ all assume a value of $G$. If a machine does not service any job then the second inequality condition (i.e. $S_{K_m} < G$) in equation 11.23 does not hold true and that machine is indirectly ignored. For example job 2 and job 4 are both assigned to machine 1 as is known form Figure 11.14. The entry wise product of $M_k$ and $\eta_{1,k}$ results in $\Omega_{1,k}$ as shown in

Figure 11.15, indicating that only $\Omega_{1,2}$ and $\Omega_{1,4}$ have values less than $G$. The minimum of these two is $\Omega_{1,2}$ indicating job 2 as the earliest job on machine 1 with a start time of 108. An element of the matrix $\boldsymbol{\Omega_{JJ}}$ is given by

$$\Omega_{mk} = [S_k.\eta_{mk} \ldots\ldots \ \forall\ k] \ \ \forall\ m \tag{11.24}$$

where $\eta_{mk}$ is an element of matrix $\boldsymbol{\eta_{JJ}}$. An element of the matrix $\boldsymbol{\eta_{JJ}}$ can be defined mathematically as

$$\eta_{mk} \quad \begin{aligned} &= 1 \quad \text{if } m = M_k \quad \forall\ m = 1\ldots J,\ \forall\ k = 1\ldots J. \\ &= G \quad \text{otherwise} \end{aligned} \tag{11.25}$$

where $G$ is a large positive integer. The index of the earliest job on machine $m$ is then given by

$$Ķ_m = \arg_k [\text{Min}\ [\Omega_{mk} \ldots\ldots \ \forall\ k]] \ \ \forall\ m \tag{11.26}$$

where $Ķ_m$ = index of the earliest job on machine $m$, and the start time of the earliest job on machine $m$ is given by

$$S_{Ķ_m} = [\text{Min}\ [\Omega_{mk} \ldots\ldots \ \forall\ k]] \ \forall\ m \tag{11.27}$$

The way the values $S_{Ķ_m}$ and $Ķ_m$ are determined, is demonstrated in Figure 11.15 for the schedule in the given example.

### 11.4.2 Experiment 5

No comprehensive data set is available that completes the requirements of rigorous testing of the stream tube GA model. Therefore, to evaluate the performance of the stream tube GA model with travel time, instances were randomly generated from a uniform distribution, for three different values of travel times. The three values of travel times are 100, 43, and 21.5 minutes. Although these three values are arbitrary, however the average and maximum travel time in the practical example given by De Vries (2003) for the tertiary unit described by Bishop and Long (1983) are approximately 21.5 and 100 minutes.

Bishop and Long (1983) presented a detailed procedure for developing delivery schedules for a sequential irrigation (rotation) system taking into account canal filling time, seepage losses, and management time. Bishop and Long (1983) suggested that travel time should be calculated by determining the lengths of irrigation channels and multiplying them by a velocity that is approximately 0.7 times the average discharge velocity of the channel involved, to compensate for the lower advance velocity in dry channel as compared to when it is flowing full. De Vries (2003) used the velocities given by Bishop and Long (1983) for the tertiary unit and calculated travel time between outlets based on these velocities and the procedure described by Bishop and Long (1983). Detailed information required for travel time calculation was not provided by Bishop and Long (1983), other than the map of the tertiary unit. De Vries (2003) calculated the fields, and channels measurement from the map (Figure 11.16), and assumed fields’ outlets to be in upstream corner of each field.

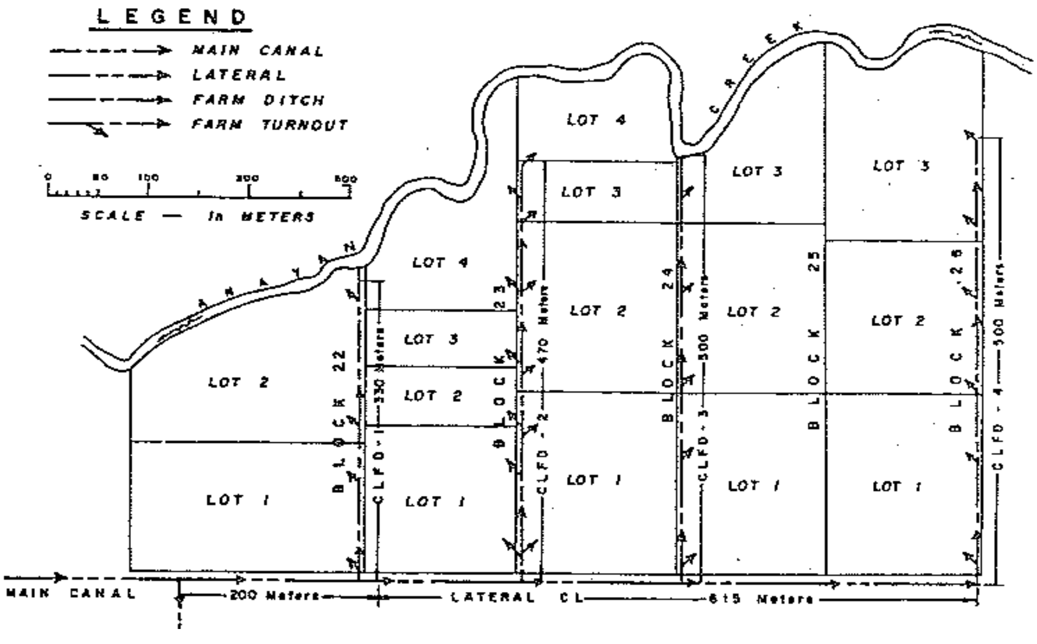
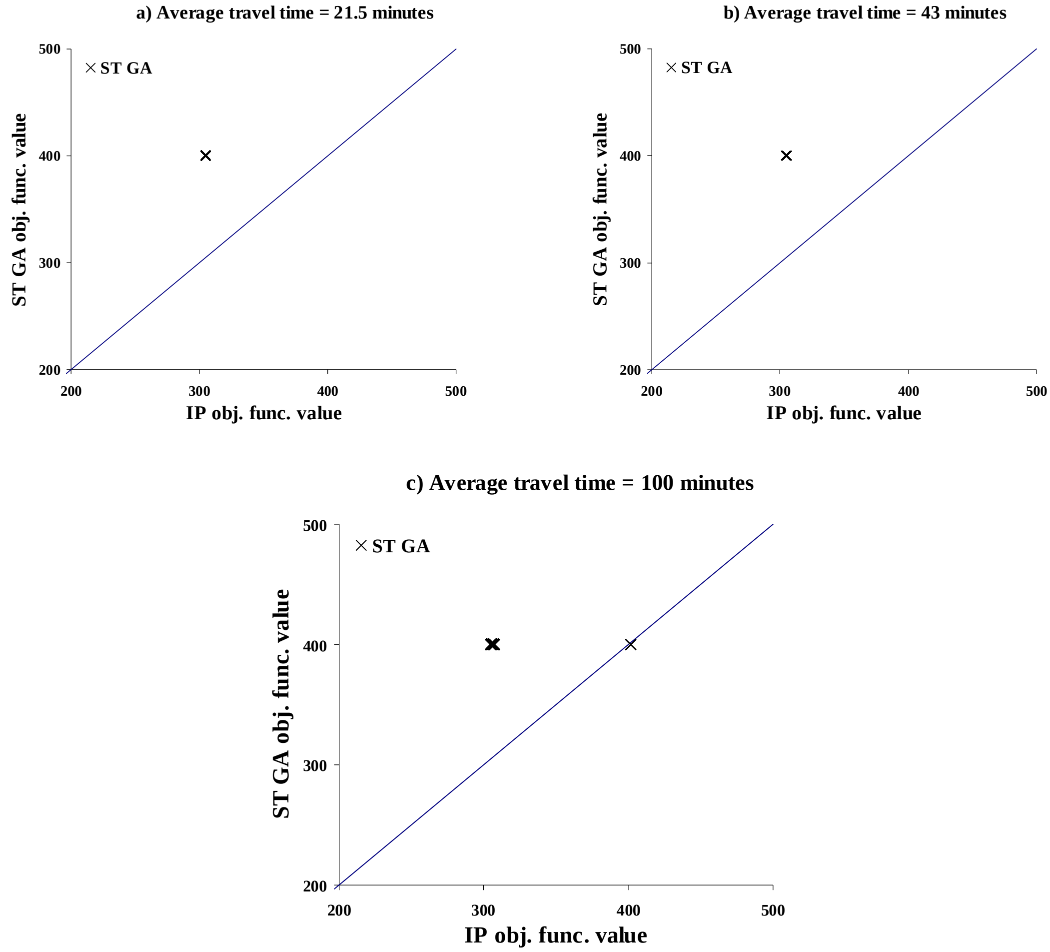



**Figure 11.6** Map a tertiary unit, Bula Project, Philippines (Bishop and Long (1983)

For this experiment, three set of travel times were randomly generated from a uniform distribution, based on 100, 43, and 21.5 minutes average travel time. For each value, 100 different instances were generated such that average travel time for each instance is approximately equal to the value it is representing. For example, the average travel for any instance in the 43 category should be approximately equal to 43 minutes. In order to evaluate the performance of the stream tube GA for travel time only, the target start time and duration were kept the same for all instances. The target start and duration are taken from a practical example presented by Anwar and Clark (2001). IP solutions for all instances were obtained and are compared with the solutions generated by the stream tube GA with travel time. The best algorithm parameters found from preliminary experimentation are used for the current experiment. They are; population size 100, mutation rate 0.3, and crossover rate 0.7. The maximum number of generations was fixed at 10,000, which proved sufficient for all instances. The early stopping criteria as used in all the previous experiment is also used in the current experiment.

Figure11.17 (a-c) presents the results graphically. As with other dual goal objective models, the discharge minimization has a priority over earliness/tardiness of a schedule. Therefore, in the objective function discharge has more weight than earliness/tardiness. Since both the information of discharge and earliness/tardiness are contained in a single value of the objective function, it is difficult to see the variation in earliness/tardiness for schedules having the same discharge. For this reason in the graphs shown in Figure 11.17, all the schedules having the same discharge are visible as a single point. A detailed analysis of the results for Experiment 5 is presented in Table11.12. For the 21.5 minutes average travel time, the stream tube GA with travel time was neither able to find an optimum solution nor any solution with the same discharge as IP. All the 100 instances were solved by the GA model with 4 units of discharge as against 3 units of discharge by IP. The average earliness/tardiness from the stream tube GA was 2.54 days. This average earliness/tardiness however, can not be compared with IP, as all the instances were solved by the IP with 3 units discharge in contrast to the 4 units of discharge by the stream tube GA with travel time. More supply would definitely result in better earliness/tardiness.




ST GA = Stream tube GA; IP = Integer programming

**Figure11.17** Simple multimachine dual goal and travel time GA vs. IP

**Table 11.12.** Results of the stream tube GA model with travel time

| Model | Setup | No. of instances with 3 units of discharge | No. of instances with 4 units of discharge | Average earliness/tardiness (days) | No. of instances with the same discharge as IP |
|---|---|---|---|---|---|
| **IP** | 21.5 | 100 | 0 | 4.8 | |
| | 43 | 100 | 0 | 4.9 | |
| | 100 | 90 | 10 | 5.4 | |
| **GA** | 21.5 | 0 | 100 | 2.4 | 0 |
| | 43 | 0 | 100 | 2.3 | 0 |
| | 100 | 0 | 100 | 2.7 | 10 |

For 43 minutes average travel time, the stream tube GA with travel time was neither able to find an optimum solution nor any solution with the same discharge as IP. All the 100 instances were solved by the GA model with 4 units of discharge as against 3 units of discharge by IP. The average earliness/tardiness from the GA model was 2.3 days. With 100 minutes average travel time the stream tube GA with travel time was able to find optimum solutions to 1 instance out of 100 and 10 instances with the same units of discharge as IP. The average earliness/tardiness of the 10 instances was 2.05 days as against 1.56 days from IP, however the average earliness/tardiness for all the 100 instances from the stream tube GA model was 2.7 days.

Table 11.13 presents the results when the stream tube GA and the IP are applied to the same data (i.e. practical example given by Anwar and Clarke (2001)) without travel time consideration. Table 11.12 and Table 11.13 provide an objective comparison of a single instance of the performance of the model with and without travel time.

**Table 11.13.** Results of the stream tube GA model without travel time

| Model | Average earliness/tardiness (days) | Units of discharge |
|---|---|---|
| **IP** | 4.73 | 3 |
| **Stream tube GA** | 1.44 | 4 |

As shown in Table 11.12 at lower values of travel time the stream tube GA neither found any optimum solution nor any solution with the same discharge as IP. In contrast, at the highest value of travel time among the three different values of travel time considered in the experiment, the stream tube GA performed better and was able to find optimum solution to 1 instance out of 100 and 10 instances with the same units of discharge as IP. As shown in Table 11.13, with no travel time the stream tube GA was also neither able to find the optimum solution nor a solution with the same units of discharge as IP, however, the earliness/tardiness is less than any of the average values of the earliness/tardiness for all instances with travel time given in Table 11.12. It may be concluded that the performance of the stream tube GA deteriorates with the addition of travel time, however it does not deteriorate with increasing travel time.

## 11.5 Complex multimachine model

All the models discussed in Section 11.1 to 11.4 assumed identical discharges for all outlets/users. However it is possible that the discharge requirements of different users are different from each other. The model discussed in this section is an improvement over the models presented in Section 11.1, by allowing different users to demand irrigation water at different discharges. This makes the model more flexible and practical to accommodate any variation in discharges. The word “complex” in the multimachine model refers to this additional complexity of non-identical discharges as described earlier in Section 7.4. Multimachine scheduling problems even with identical discharges are hard optimization problems. The incorporation of non identical discharges makes it more complex and computationally more demanding. Examples of models that solve problems with the dual goal objective of minimizing both machine and earliness/tardiness and with the additional complexity of non identical discharges can hardly be found in literature. Although De Vries (2003) presented formulations for a series of complex multimachine problems using IP, however no solution was obtained by any of the models because of the increased solution time. De Vries (2003) was only able to obtain solution to an 8 jobs problem, with only two outlets with different discharges, by using an alternative formulation. This alternative formulation consists of two models, the first minimises the discharge in the channel and the second minimises the earliness/tardiness.

Both the stream tube approach and the time block approach are flexible enough to handle this additional complexity of non identical discharges. However, since the time block approach proved better than the stream tube in the simple multimachine problem, only the time block approach is used to develop the complex multimachine model. Travel time is assumed very small and hence ignored. The time block GA presented in Section 11.1.2 can be adopted for complex multimachine model with some modification. Equations 11.14 and 11.15 of the simple multimachine model in Section 11.1.2 are replaced by the following equations.

$$Q_{\max} = \max[\sum_{j=1}^{J} \psi_{tj}^{\sim}] \qquad \forall \;\; t = 1, 2\ldots T. \qquad (11.28)$$

where $t$ = time block index = 1, 2…$T$; and, $T$ = total number of time blocks, and $\psi_{tj}^{\sim}$ is defined as:

$$\psi_{tj}^{\sim} \;\; = q_j \quad \text{if } S_j \leq t < S_j + Dj; \qquad \forall \;\; t \qquad (11.29)$$
$$= 0 \quad \text{otherwise.}$$

where $q_j$ = required discharge of outlet $j$.

### 11.5.1 Experiment 6

This experiment is designed to examine the effect of the non identical discharges on the performance of the time block GA. The data used in Experiment 3 for the 8 jobs problem, is modified by including different discharges for individual users. In Experiment 3 the discharge requirements of all individual users was assumed identical. In the current experiment all individual users are allowed to demand any discharge from a range of 1 to 4 units of discharge. Since extensive, real data of an arranged demand irrigation system with non identical discharges that completes the requirement of GA testing, is not available; each individual user is assigned a discharge randomly generated from a uniform distribution from a range of 1 to 4 units of discharge. The rest of the data is the same as that of the 8 jobs problem in Experiment 3. The purpose of the current experiment is to evaluate the performance of the time block GA for non identical discharges; therefore 100 different instances with 100 different variations of the non identical discharges, for only one problem size (i.e. 8 jobs) were tested. The IP and the time block GA was rerun for the modified data (of Experiment 3, 8 jobs problem) with non identical discharges and solutions were obtained. Since the IP takes very long to execute, its execution time per instance was limited to 3 hours. The GA was allowed to run for a maximum of 1500 generations that proved enough in Experiment 3 for the 8 jobs problem, however, an early stopping criteria was also used that terminates the GA if there is no improvement in solution quality. A population size of 100, mutation rate 0.3, and crossover rate 0.75, as used for the simple multimachine model, Experiment 3, are also used for the current model.

The complexity of the irrigation scheduling problem with non identical discharges (or complex multimachine problem) can be judged from the fact that the IP was unable to solve any of the 100 instances to optimality within the allocated time of 3 hours. It is worth mentioning that in Experiment 3, for the 8 jobs problem with identical discharges, the IP was able to obtain optimum solutions to all the 100 instances with an average time of 14 seconds per instance. Since the IP was unable to obtain any optimum solution within the allocated time, it can not be established whether the time block GA obtained any optimum solution. Results from Experiment 6 are presented in Table 11.14. A comparison of the objective function values of the time block GA and IP is also presented in Figure 11.18.

**Table 11.14** Results from Experiment 6

| **Parameters** | **Instances/100** |
|---|---|
| GA solutions with the same channel discharge as IP | 37 |
| GA solutions with less channel discharge than IP | 13 |
| GA solutions with more channel discharge than IP | 21 |
| GA solutions with the same ET as IP | 7 |
| Total number of solutions by GA where IP did not solve optimally within the allocated time | 55 |

The time block GA was able to find better solutions than IP for 55 of the 100 instances, where the IP was unable to solve optimally within the allocated time (Table11.14). The solution time of GA was also much less than that of IP. The average solution time per instance by the GA was 3.2 minutes as compared to the 3 hours by IP. It is however worth noting that the IP could have obtained better solutions, had it been allowed to execute beyond the allocated time of 3 hours per instance. The time block GA obtained 37 schedules with the same discharge as that of the schedules by IP and 13 schedules with discharge less than that of IP. The IP was unable to obtain feasible schedule to 29 instances while the GA was able to obtain feasible solutions for all the 100 instances. Based on the results presented above, it may be concluded that the time block GA has performed consistently more efficiently for the simultaneous irrigation scheduling problem with non identical discharges.

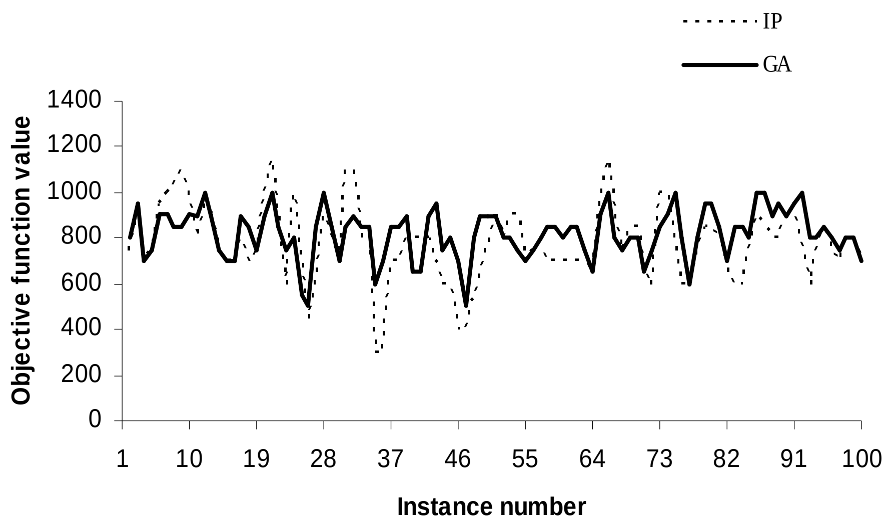



**Figure 11.18** Comparison of IP and GA for complex multimachine problem

### 11.5.1 Practical application

Suryavanshi and Reddy (1986) described a real tertiary unit with 8 users who are allowed to irrigate simultaneously. Suryavanshi and Reddy (1986) presented a schematic representation and irrigation durations for this tertiary. Figure 7.4 shows this schematic and Table 11.15 shows the irrigation durations. Anwar and Clarke (2001) generated random target start times for the tertiary unit described by Suryavanshi and Reddy (1986). De Vries (2003) doubled and tripled the discharges of two randomly chosen outlets (outlet 2 and 5) to transform the simple multimachine problem addressed by Anwar and Clarke (2001) into a complex multimachine problem. Table 11.15 shows the target start times generated by Anwar and Clarke (2001) and the discharges generated by De Vries (2003). For the sake of simplicity in calculations a unit of discharge is represented by 1. For example outlet 2 has a discharge requirement of 2 units, if a unit of discharge is 30L/s then 2 units of discharge will be equal to 60 L/s. The time block GA is applied to the problem. A population size of 100, mutation rate 0.3, and crossover rate 0.75, as used in Experiment 6 are also used for the current application.

**Table 11.15 Input data for practical application**

| Outlet number | Irrigation duration (days) | Target start time (days) | Discharge |
|---|---|---|---|
| 1 | 0.80 | 3.55 | 1* |
| 2 | 2.13 | 0.41 | 2 |
| 3 | 2.40 | 2.16 | 1 |
| 4 | 1.72 | 1.49 | 1 |
| 5 | 2.05 | 0.61 | 3 |
| 6 | 2.43 | 0.26 | 1 |
| 7 | 2.05 | 1.60 | 1 |
| 8 | 2.50 | 3.03 | 1 |

* 1 means one unit. One unit is equal to 30 L/s, so 2 units equal to 60 L/s

The time block GA obtains the best solution in 2.4 minutes at the 459$^{th}$ generation. The total supply for the schedule is 5 units (150 L/s) and the earliness/tardiness 4.12 days as against the IP solution (De Vries (2003) two stage formulation) of 5 units discharge and an earliness/tardiness of 3.96 days. The GA has done reasonably well in obtaining a schedule with the same units of discharge as IP and the earliness/tardiness with a relative error of 4 % more efficiently.

It is also worth mentioning that the time block GA was able to obtain the optimum solution to the practical problem reported by Anwar and Clarke (2001). It is the same problem as described above but with identical discharges. Anwar and Clarke (2001) reported 3 units of discharge and an earliness/tardiness of 4.73 days for this multimachine problem with identical discharges (simple multimachine problem). The time block GA obtained the same solution by using a mutation rate of 0.3, crossover rate of 0.85, and a population of 400 at the 719$^{th}$ generation.

## 11.6 Complex multimachine with setup times

The complex multimachine problems are complex and computationally very demanding as compared to the simple multimachine problem. This complexity is also indicated by the results of Experiment 6. The addition of sequence-dependent setup times makes the complex multimachine problem even more complex. Examples, where simultaneous sequencing and scheduling, minimization of machines, minimization of earliness/tardiness, consideration of non identical machines, and setup times are all addressed by a single model, could not be found in literature sighted during the course of the current study. The GA model to be presented in this section addresses all these issues concurrently and thus is a significant step forward not only in the field of irrigation scheduling but OR as well. In irrigation terminology the model to be presented in this section will prepare a water delivery schedule, considering not only the desires of the farmers as regards to their requested irrigation time and discharge but also the travel time the water takes from one farmer's outlet to another. Although De Vries (2003) presented formulations for a series of complex multimachine with setup times using IP, however, no results were reported. De Vries (2003) reported that solution times increase with the number of jobs to be scheduled and as a result only smaller problems can be solved. De Vries (2003) recommended heuristics or genetic algorithms as appropriate solution techniques to solve larger problems.

As explained in section 11.4 in the case of simple multimachine with setup that it is difficult to formulate a multimachine model with setup based on the time block approach. Therefore the stream tube approach is utilized to formulate the complex multimachine model with setup. The stream tube approach used in the current model is unique. The novelty of the current formulation is that it considers machines to be of variable capacity rather than with fixed, identical capacities. A job that will require more than one identical machines (with fixed capacities) using the stream tube approach adopted by the previous researchers, will require just one machine with adjustable or variable capacity using the stream tube approach adopted in the current formulation. In the context of irrigation, this concept is explained by considering the case of two farmers (A and B) as an example. Farmer A demands 1 unit of discharge and B demands 3 units of discharge. The stream tube concept used by the previous researchers will require farmer A to be served by one stream tube (machine) of one unit discharge and farmer B with three stream tubes (machines) each of one unit discharge. In contrast, the stream tube concept used in the current formulation will require only one machine for both farmer A and B. The machine will automatically adjust its capacity according to the requirements of the job being processed. This novel concept has the added advantage that the same chromosome of the simple multimachine model with setup can be used without any modification. The maximum number of machines is again equal to the number of jobs and the capacity of any machine is dependent on the requirements of the job which is assigned to that machine. The stream tube GA as used for the simple multimachine with setup is thus fully applicable to the complex multimachine with setup with the following modification. Equation 11.4 and 11.5 are replaced by equation 11.30 to 11.32.

$$Q_{max} = \sum_{m=1}^{M} \vartheta_m \qquad (11.30)$$

where $\vartheta_m$ is an element of vector $\boldsymbol{\vartheta}_M$ that represents the maximum among the discharges of outlets serviced by the stream tube (machine) $m$, which is defined as follows.

$$\vartheta_m = [\max[\tilde{q}_{mj} \;\ldots\ldots\; \forall\, j]] \qquad \forall\, m \qquad (11.31)$$

where $\tilde{q}_{mj}$ is an element of matrix $\tilde{\boldsymbol{q}}_{JJ}$ and is mathematically defined as follows.

$$\begin{aligned} \tilde{q}_{mj} &= q_j \quad \text{if } m = M_j \;\; \text{(i.e if job } j \text{ is assigned to machine } m\text{)} \\ &= 0 \quad\; \text{otherwise.} \end{aligned} \qquad (11.32)$$

where $q_j$ = discharge of outlet $j$, and $M_j$ = an element of machine vector. The example of the chromosome presented in Figure 11.1 is reused to illustrate these equations. Figure 11.19 shows the matrix $\tilde{\boldsymbol{q}}_{JJ}$ for the given example and contains all the information that is required to calculate the total supplied discharge. It includes the genes of the chromosome ($M_j$) that defines the assignment of jobs to machines (i.e. machine row vector) and the discharge requirement of the individual users ($q_j$). For example job 2 is assigned to machine 1 and has 2 units discharge requirements. Similarly job 4 is also assigned to machine 1 and has a discharge requirement of 3 units. This information is stored in the matrix $\tilde{\boldsymbol{q}}_{JJ}$. It can be seen from matrix $\tilde{\boldsymbol{q}}_{JJ}$, shown in Figure 11.19 that job 2 and 4 are assigned to machine ($m$) 1 and have 2 and 3 units discharge requirements respectively; no job is assigned to machine 2; job 3 is assigned to machine 3 and has 1 unit of discharge; job 1 is assigned to machine 4 and has 1 unit of discharge. The vector $\vartheta_M$ can be easily obtained from this matrix, which represents the maximum value in each row of matrix $\tilde{q}_{JJ}$. The vector $\vartheta_M$ is shown in Figure 11.20. Figure 11.20 also shows how the total supply ($Q_{max}$) is obtained from the vector $\vartheta_M$.

| Machine row vector and discharges of outlets | | | | |
|---|---|---|---|---|
| Job | **1** | **2** | **3** | **4** |
| $M_j$ | **4** | **1** | **3** | **1** |
| $q_j$ | **1** | **2** | **1** | **3** |

| $\tilde{q}_{mj}$ | | $j$ | | | |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 |
| $m$ | 1 | 0 | 2 | 0 | 3 |
| | 2 | 0 | 0 | 0 | 0 |
| | 3 | 0 | 0 | 1 | 0 |
| | 4 | 1 | 0 | 0 | 0 |

**Figure 11.19** Matrix $\tilde{q}_{JJ}$

| $m$ | 1 | 2 | 3 | 4 | $Q_{max}$ |
|---|---|---|---|---|---|
| $\vartheta_m$ | **3** | **0** | **1** | **1** | (3+0+1+1) = **5** |

**Figure 11.20** Vector $\vartheta_M$

The complex multimachine with setup times is implemented by using the stream tube GA as applied to the simple multimachine with the only modification required for equations 11.30 to 11.32. All the algorithm parameters and operators as used in the simple multimachine with setup times are also used in the complex multimachine with setup times. No change was deemed necessary. Since no optimum solution could be obtained for even the complex multimachine without setup time using IP, it is harder to obtain any optimum solution for the complex multimachine with setup times using IP with the available knowledge and resources. In the absence of bench mark solutions the only option left is, comparing heuristics against other heuristics; however even no heuristic solutions are known for the model presented in this section. In this situation, therefore, a rigorous evaluation of the performance of the model can not be carried out unless benchmark solutions are known. For the purpose of demonstrating the application of the model, however, the model is applied to an 8 jobs problem with non identical discharges presented in Section 11.5.1. Since no real travel time data is available for the problem, travel times were randomly generated such that the average travel time was not more than 100 minutes. Table 11.16 shows  travel time the water takes from one outlet to another and also the travel time from the head of the supply channel to each individual outlet.

**Table 11.16** Travel times for the complex multimachine problem (minutes)

| | | Outlet *k* | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $T_{jk}$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| | 0* | 158 | 115 | 202 | 43 | 72 | 173 | 101 | 72 |
| Outlet *j* | 1 | 0 | 72 | 29 | 43 | 72 | 43 | 14 | 72 |
| | 2 | 29 | 0 | 115 | 130 | 58 | 0 | 158 | 173 |
| | 3 | 130 | 130 | 0 | 72 | 29 | 101 | 86 | 58 |
| | 4 | 187 | 29 | 58 | 0 | 43 | 43 | 14 | 86 |
| | 5 | 72 | 101 | 86 | 187 | 0 | 58 | 72 | 202 |
| | 6 | 187 | 29 | 202 | 158 | 86 | 0 | 0 | 0 |
| | 7 | 14 | 29 | 144 | 144 | 187 | 86 | 0 | 86 |
| | 8 | 158 | 115 | 58 | 173 | 130 | 158 | 173 | 0 |

* Main supply gate (head of supply channel)

The stream tube GA is applied to the 8 jobs with non identical discharges and travel times as discussed in the preceding paragraph. A population size of 100, mutation rate 0.3, and crossover rate 0.7, as used for the simple multimachine with setup times are also used for the current model. The maximum number of generations was fixed at 10,000, which proved sufficient in the case of simple multimachine with setup times. The early stopping criteria as used in all the previous experiment is also used in the current experiment. The best solution was obtained by the stream tube GA in just 5.3 seconds and at the1748$^{th}$ generation. The total earliness/tardiness of the schedule is 5.07 days and the total supply is 6 units of discharge. If 1 unit of discharge is equal to 30 L/s then the total supply becomes 180 L/s. Figure11.21 shows the irrigation schedule prepared by the complex multimachine model with setup times. It can be seen form the figure that only four machines are required to process all the 8 jobs. Machine (stream tube) D adjusts its capacity according the requirements of job (outlet) 2 and 5, thus supply them with 2 and 3 units of discharge respectively. According to the schedule obtained by the model, outlet 5 is scheduled to receive water after outlet 2 has completed its irrigation turn and the travel time of 58 minutes (Table 11.6). Any such schedule can be prepared by the model presented in this section, considering not only non identical discharges but travel times as well, with the dual goal objective of minimizing discharge as well as earliness/tardiness. Obtaining feasible schedules for such a problem with known computational complexity efficiently, is a great success of the model presented. A detailed study is, however, required to address the real life issues described earlier for incorporating the travel time. The novel concept of variable machine capacity presented may also be utilized in an industrial environment and is also an interesting concept for the OR community to further investigate.

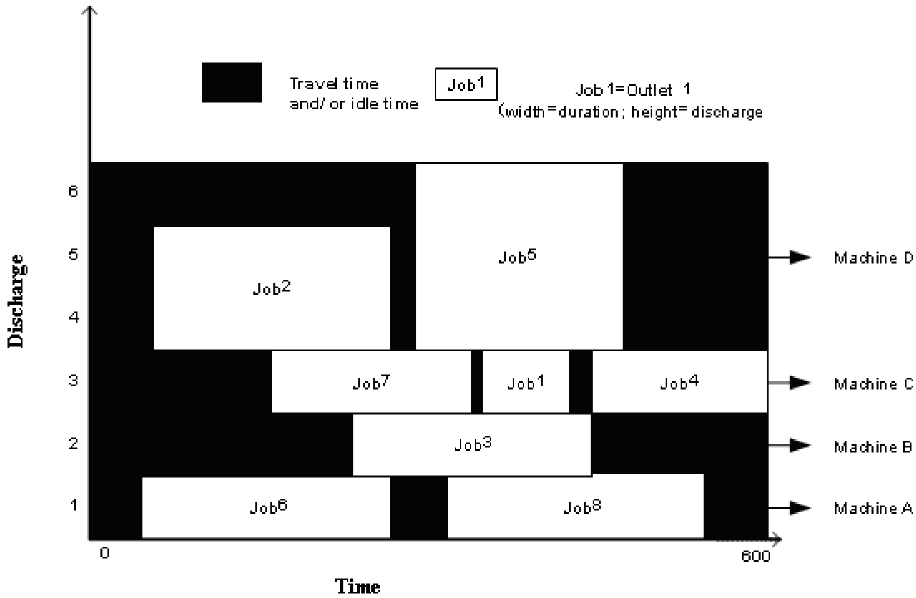



**Figure 11.21** Irrigation schedule of an 8 jobs problem with non identical discharges and travel times. (600 time units = 6 days)

# 12 Summary of results

## 12.1 Sequential irrigation models

For the sequential irrigation scheduling problems presented in Section 10, the GA models were tested for developing irrigation schedules, for a range of 8 to 25 outlets. The performance of the GA models was tested against the integer programmes and heuristic of Anwar and De Vries (2004). The GA models were able to obtain feasible schedules with better quality than the heuristic for all instances. The IP solutions were better than GA for schedules with 8 and 10 outlets. For larger schedules equal to or greater than 12 outlets, the IP was unable to obtain optimum schedules within the allocated time of 3 hours. The GA models were able to obtain feasible schedules with good quality for all those instances where the IP was unable to obtain optimum schedules within the allocated time. It is worth mentioning that the IP would have obtained optimum solutions, if it were allowed to continue running beyond the allocated time.

## 12.2 Simultaneous irrigation models

For the simultaneous irrigation scheduling problems with identical discharges, two GA models; the time block GA; and, the stream tube GA were developed and tested. IP solutions were used as benchmarks. The GA models were tested for two problem specific parameters i.e. demand: supply ratio and problem size.

The demand: supply ratios tested ranged from 0.1 to 0.9. The performance of the GA models was tested for a schedule with eight outlets only, and the single goal objective of minimizing earliness/tardiness under channel capacity restriction. The IP was able to obtain optimum schedules to all instances with 8 outlets across the range of demand: supply ratios tested. The time block GA was able to obtain optimum solutions to 90 % of the instances tested in contrast to 63% by the stream tube GA. The stream tube GA obtains schedules faster than both the IP and the time block GA. For example, at the 0.5 demand: supply ratio the average execution time is approximately 7 seconds for only one schedule using the stream tube GA, in contrast to 30 seconds by the IP and 298 seconds by the time block GA.

The simultaneous irrigation GA models were tested for a range of 8 to 12 outlets at a fixed demand: supply ratio of 0.5, with the single goal objective of minimizing earliness/tardiness under channel capacity restriction. The IP obtained optimum schedules for all instances across the range of outlet numbers. The time block GA was able to obtain optimum solutions to 83 % of the instances tested in contrast to 47% by the stream tube GA. The solution quality of the time block GA is better than the stream tube GA, however the stream tube GA obtains schedules faster than both the IP and the time block GA. For schedules with 12 outlets, the average execution time per instance by the stream tube GA was 31 seconds in contrast to 34 minutes by the IP and 20 minutes by the time block GA.

Both the stream tube GA and the time block GA for the simultaneous irrigation with identical discharges were also tested for problem sizes with 8, 10, and 12 outlets at a fixed demand: supply ratio of 0.9 and the dual goal objective of minimizing discharge and earliness/tardiness. The performances of the GA models deteriorated as the number of outlets is increased. The solution quality of the time block GA is better than the stream tube GA, however the stream GA again proved faster in obtaining schedules. For schedules with 12 outlets, the average execution time per instance by the stream tube GA was 48 seconds in contrast to 133 minutes by the IP and 17 minutes by the time block GA. The solution quality of the GA models with the dual goal objective is poorer than the GA models with the single goal objective. However, the dual goal objective of minimizing discharge and earliness/tardiness is more complex than the single goal of only minimizing earliness/tardiness. The complexity of the problem can be judged from the fact that the IP was able to solve optimally only 43 of the 100 instances for schedules with 12 outlets, within the allocated time of 3 hours. Both the GA models were able to obtain feasible schedules for all the instances much faster than the IP.

The time block GA with the dual goal objective was applied to a sequential irrigation scheduling problem. The purpose was to see if the time block GA was able to obtain a schedule to a sequential irrigation scheduling problem with the same quality as a dedicated sequential irrigation scheduling model would do. The solution quality of the time block GA was better than the non-contiguous sequential irrigation scheduling model (Model 1) when applied to the same sequential irrigation scheduling problem. The time block GA with the dual goal objective was also applied to simultaneous irrigation scheduling problems with non identical discharges. The number of outlets was limited to eight outlets only, and 100 different instances with varying discharges were randomly generated. The IP was unable to solve optimally any of the 100 instances within the allocated time of 3 hours.  The time block GA was able to find feasible schedules to all instances, with an average time of 3.2 minutes per instance.

## 12.3 Travel time models

The time block formulation concept does not allow travel time to be considered in contrast to the steam tube concept which allows travel time consideration. Therefore, only the stream tube GA was augmented by considering travel time for simultaneous irrigation scheduling problem both with identical and non identical discharges.

The performance of the stream GA with travel time and identical discharges (simple multimachine with travel time) was compared with IP for a range of travel times. With the addition of travel time the performance of the GA model deteriorated against that without travel time. The GA model was able to obtain feasible schedule to all the 300 instances, however only one of them was optimum. The GA model was much faster than IP in obtaining schedules. For example for schedules with eight outlets and 21.5 minutes average travel time between outlets, the execution time of the GA model was 5 second per schedule in contrast to 69 seconds by the IP. No benchmark solutions were available for simultaneous irrigation scheduling problems with non identical discharges and travel time (complex multimachine with travel time); hence detailed evaluation of the GA model could not be carried out. However, the use of the simultaneous irrigation scheduling problems with non identical discharges and travel time was demonstrated by applying it to a single instance with 8 outlets and an average travel time of 100 minutes between outlets. The GA model was able to obtain a feasible schedule within 5.3 seconds

# 13 Conclusion and recommendations

Several irrigation water delivery methods are in practice in irrigated agriculture throughout the world and a variety of classifications have been suggested by different researchers. Demand, arranged, and rotation are the three main types of irrigation schedules/delivery methods. Irrigation systems may also be classified as either sequential or simultaneous. Supplying water sequentially to farmers according to their requested times constitutes an irrigation scheduling problem analogous to the classical earliness/tardiness single machine scheduling problems in OR. Similarly, supplying water simultaneously to farmers according to their requested times constitutes an irrigation scheduling problem analogous to the classical earliness/tardiness multimachine scheduling problems in OR. Such scheduling problems belong to a class of combinatorial optimization problems known to be computationally demanding (NP-hard). This is widely reported in OR literature. In previous published work integer programming was used to solve the irrigation scheduling problems; however integer programming can only be used to solve relatively small problems usually in a research environment where considerable computational resources and time can be allocated to solve a single schedule. For practical applications meta-heuristics such as genetic algorithms, simulated annealing or tabu search methods need to be used. However as reported in the literature, these need to be formulated carefully and tested thoroughly. The current research applied genetic algorithms to the single and multimachine irrigation scheduling problems.

Rotation is widely practiced in the Indian subcontinent. The rate, frequency, and duration are all fixed and remain fixed for the entire irrigation season in rotation schedules. Rotation is locally known in Pakistan as *warabandi*. Therefore, keeping in view this wide acceptance of the *warabandi* system, a series of single machine models was developed that are applicable to the *warabandi* systems. Like the *warabandi* system, in the single machine models presented in this thesis, a farmer receives water after the preceding one has finished his turn of irrigation. The models presented in this study give a new dimension to the *warabandi* system by allowing farmers to request water supply at their desired times. The models also have the additional capacity of prioritizing the irrigation turns of individual users. Similarly, different models portray different management options such as contiguous and non-contiguous scheduling. These options may be chosen either to reduce the costs associated with operational spillage and gate operations, and/or to better match target start times and scheduled start times.

To explore the potential of genetic algorithm to solve the simultaneous irrigation scheduling problem more efficiently with optimum or near optimum solution, two types of multimachine models were developed i.e. the simple multimachine and the complex multimachine model. In the simple multimachine models all the farmers are supplied water with identical discharges and in the complex multimachine models farmers may be supplied with non identical discharges. The multimachine models were developed based on the stream tube approach and the time block approach. The suitability of the two approaches for the multimachine models was fully explored.

In open channels, travel times play an important role while determining irrigation turns for individual farmers. The positions of the intake outlets to different fields in a tertiary unit relative to each other and the main supply gate influence the travel time of water. Not taking account of the travel times can result in early and/or late deliveries of water that can lead to under or over irrigation. Realizing the importance of travel times, both the simple and the multimachine models were augmented by incorporating travel times. Multimachine models with the dual goal objective of machine minimization and earliness/tardiness are complex and unique. Examples of such models can not be found even in OR. The addition of travel time makes them more complex, computationally very demanding, and unique.

## 13.1 Sequential irrigation models

A series of computational experiments were carried out to evaluate the performance of the sequential irrigation (single machine) GA models across different problem sizes. The contiguous models proved better than the non contiguous models in terms of solution quality and execution time. Within the contiguous models, the model that allows idle time to be inserted before the start of irrigation and after the irrigation is complete proved computationally more efficient. Having idle time before the start of irrigation and after the irrigation is complete, may provide some flexibility on the operational level as well.
Overall the single machine GA models have performed very well and completely outperformed the heuristics by Anwar and De Vries (2004) in terms of solution quality. The GA models developed in this study were able to obtain feasible solutions efficiently for larger

problems (i.e. problems with number of jobs equal to or greater than15), for which the IP was unable to obtain optimum solutions within the allocated time of 3 hours per instance. Based on the results obtained from the experiments, it may be concluded that the GA has a considerable potential as a decision making support tool to prepare sequential irrigation schedules under arranged demand irrigation systems. The GA models developed will help the irrigation managers provide a better level of service to the farmers by supplying water as close as possible to their requested irrigation times.

## 13.2 Simultaneous irrigation models

The simultaneous irrigation scheduling models both with identical and non identical discharges (simple and the complex multimachine) developed in this study using GA are able to minimize the discharge required to satisfy farmers demand and the earliness/tardiness simultaneously. These are two conflicting objectives. Minimizing the discharge could mean increased earliness/tardiness. Similarly minimizing earliness/tardiness could mean allowing an increased discharge in the channel, to match irrigation demand and water delivery optimally. The role of an optimization tool is to satisfy these two conflicting objectives efficiently, in the best possible manner. The GA simultaneous irrigation scheduling models presented in this study achieved this objective very efficiently. The simultaneous irrigation scheduling models were also demonstrated to be equally applicable to the non contiguous sequential irrigation scheduling problems (i.e. single machine scheduling). Thus simultaneous irrigation scheduling models are able to help an irrigation manger decide whether to supply sequentially or simultaneously. Achieving the dual goal objective of discharge minimization and earliness/tardiness simultaneously and efficiently is a significant contribution not only in the field of irrigation scheduling but OR as well.

In this study, the stream tube approach and the time block approach for the development of the simultaneous irrigation scheduling models were explicitly distinguished and compared. Based on the results obtained from the computational experiments, the time block GA proved a preferred choice over the stream tube GA for the simultaneous irrigation scheduling problems without travel times. The evaluation and comparison of the stream tube approach and the time block approach has not been found in the previous published literature and hence is another significant contribution of the current research.

## 13.3 Travel time models

The time block formulation concept does not allow travel time to be considered. The steam tube concept was used to develop the simultaneous irrigation scheduling GA models with travel time consideration. The stream tube GA did not perform very well for the simple multimachine with travel time however was much faster than IP in execution. It is still believed that the GA has potential to perform better in the simple multimachine with setup and further research for its improvement is recommended. No benchmark solutions were available for the complex multimachine with travel time because the IP takes very long to solve. For example, the IP could not solve a single instance of a complex multimachine problem even without travel within the allocated time of 3 hours. Hence, the complex multimachine model with travel time could not be tested rigorously; however its use is demonstrated through its application to a practical problem. The GA model obtained a feasible schedule within only 5.3 seconds for a problem as complex as the simultaneous irrigation scheduling problem with non identical discharges and travel time.

The formulation presented for the simultaneous irrigation scheduling problem with non identical discharges and travel time (complex multimachine with setup) in this thesis, is different than the earlier notion of the stream tube approach described in literature. The novelty of the present formulation is that it considers machines to be of variable or adjustable capacity in contrast to the earlier concept of fixed identical capacities. The advantage of this new formulation is that it accommodates non identical discharges without increasing the execution time of the GA models. With this new formulation no change is required to the representation (chromosome) of the simultaneous irrigation scheduling problem for accommodating non identical discharges and travel time. The simple multimachine model with travel time can be upgraded to the complex multimachine model with travel time without any change to the chromosome. This new formulation is thus considered another significant contribution of this thesis.

The work done in this thesis has made a number of GA models available for use in irrigation scheduling under arranged demand systems. Both sequential and simultaneous irrigation are accommodated and it is possible to schedule irrigation turns for systems where different discharges to farmers are allowed. Travel time has also been considered in the simultaneous irrigation models. The overall objective of developing a computationally efficient tool that obtains solutions of good quality for the irrigation scheduling problems is achieved. The hypothesis, that genetic algorithm is a computationally efficient and robust optimization tool that can provide good quality solutions for an irrigation scheduling problem, is hence proved.

## 13.4 Recommendations for future research

In general, genetic algorithms and indeed other evolutionary heuristics do offer considerable potential to solve the scheduling problem. Numerous examples of such research and application can be found in operations research literature and these techniques are the technique of choice for large problems. It is recommended that researchers using such evolutionary heuristics to solve problems in irrigation and water management take advantage of the wealth of literature on algorithms, data sets, testing, and reporting results that can be found in operations research.

The research presented in this thesis answers some questions but at the same time generates some questions. The stream tube and the time block approach are the two approaches that could be used to develop the multimachine models presented in this study. The time block approach proved better than the stream tube approach in solution quality, however the stream tube approach was computationally more efficient. Similarly, it proved difficult to accommodate travel times using the time block approach. The stream tube approach was flexible enough to accommodate travel times, however with certain assumptions. Because of these assumptions (Section 11.4.1) the stream tube GA does not model the true scenario in its strictest sense. Further research is required to devise a better formulation that combines the better solution quality of the time block approach and the computational speed of the stream tube approach. Further work is also required on how the travel time could be accommodated in a format close to reality.

Due to the lack of actual field data, all the experimental data used in this thesis are randomly generated from a uniform distribution. To study the performance of GA at any other data sets using other distribution or actual demand patterns is another area that can be explored. Similarly, problem size and demand-supply ratio are the two problem specific parameters used to evaluate the performance of the multimachine models. For the dual goal objective of minimizing discharge and earliness/tardiness, the demand-supply ratio is not considered a relevant problem parameter for reasons explained in Section 11.3.3. Devising better parameters for evaluating the performance of the models is thus another area that may be pursued.

To derive general design equations on the capacity of channels under arranged demand systems or what capacity is needed to provide a certain level of service and vice versa, is a problem that requires further research. GA is usually considered weak in their local search capabilities and strong in their global search capabilities. Hybridization of GA with other techniques like simulated annealing, strong in the local search, is thus considered another avenue for further research.

The determination of when to irrigate and how much water to apply, is not a simple process. It requires a lot of information and then manipulation of that information. Several useful software programme are available for the purpose. Software programme are also available that optimally allocate water and area to different crops in an irrigation scheme to increase its productivity. Similarly, models are available that considers decrease in discharge along the length of a canal due to seepage, evaporation and change in canal cross section. The integration of these models with the models developed in the current study could be an interesting research project to pursue.

In this study, early and tardy jobs have been penalized equally. In an irrigation context, the tardiness of a job may be more detrimental than the earliness and vice versa. For example delaying irrigation to a field may delay the sowing job which in return has more detrimental effects on the yield of the crop. Quantifying the different costs of earliness and tardiness is thus another interesting avenue to explore.

# Appendix A: Application of Genetic Algorithms for Irrigation Water Scheduling

The following paper has been accepted for publication by the Journal of Irrigation and Drainage Engineering (American Society of Civil Engineers).

# Evaluation of a Genetic Algorithm for the Irrigation Scheduling Problem

Zia Ul Haq[1]; Arif A. Anwar[2]; and Derek Clarke[3]

**Abstract:** A typical irrigation scheduling problem is one of preparing a schedule to service a group of outlets which may be serviced simultaneously. This problem has an analogy with the classical earliness/tardiness problem in operations research. In previously published work an integer program was used to solve this problem, however such scheduling problems belong to a class of combinatorial problems known to be computationally demanding (*N-P* hard). This is widely reported in operations research. Hence integer programs can only be used to solve relatively small problems usually in a research environment where considerable computational resources and time can be allocated to solve a single schedule. For practical applications metaheuristics such as genetic algorithms, simulated annealing, or tabu search methods need to be used. However as reported in the literature, these need to be formulated carefully and tested thoroughly. This paper demonstrates the importance of robust testing of one such genetic algorithm formulated to solve the irrigation scheduling problem with simultaneous outlets serviced against an integer program formulated to solve the same problem.

**DOI:** 10.1061/(ASCE)0733-9437(2008)134:6(737)

**CE Database subject headings:** Algorithms; Irrigation scheduling; Evaluation.

## Background

Genetic algorithms (GA) are a class of adaptive heuristic search algorithms often used to solve optimization problems. The term "genetic" reflects the evolutionary nature of the algorithm and the fact that it uses mutation and crossover of "genes" and hence the obvious corollary with nature. GAs are usually attributed to John Holland (Holland, 1975) and since then have been applied to a wide range of problems. Barr et al. (1995) referred to heuristics as an approximation algorithm or inexact procedure because the solution obtained by these procedures may not be an optimal one. Hence there is the need for such an algorithm to be evaluated to demonstrate its effectiveness. Barr et al. (1995) specifically identified the following issues when evaluating heuristics:

- Solution quality—how close a solution comes to the optimum;
- Computational complexity—the time required to obtain the solution; and
- Robustness—how well the algorithm performs over a range of problems.

Barr et al. (1995) suggested that algorithms and heuristics should be tested against the best competitive solution procedures, although Hall and Posner (2001) cautioned against focusing attention on data sets where it is easiest to demonstrate improvement. Additional information on computational complexity of genetic algorithms can be found in Rylander (2001).

Rardin and Uzsoy (2001) presented an excellent tutorial on the subject of testing of heuristics. In particular Rardin and Uzsoy (2001) cautioned against testing of heuristics against integer programs if the size of the problems solved by integer programs is considerably smaller than practical problems. It may not be possible to solve the scheduling problem for larger problems (>15 jobs/outlets) within reasonable computational time using an integer program, therefore any new heuristic needs to be tested against other heuristics and that which gives a lower objective function is considered superior.

Hooker (1995) distinguished between the following two methods of evaluation:

- Analytical—relying on deductive mathematics; and
- Empirical—relying on computational experiments.

Hooker (1995) also pointed out that complex algorithms are currently beyond the reach of deductive algorithmic science ". . . the only alternative on the horizon seems to be computational testing." Deductive methods usually examine worse case and/or average case performance and not necessarily the real problem. For computational experiments Rardin and Uzsoy (2001) suggested that although the best test instances are those taken from real applications, it is rare to find more than a few data sets. This would be insufficient to test a heuristic comprehensively. Alternative sources are: random variation of real data sets, published on-line libraries, and/or randomly generated instances. Hall and Posner (2001) have pointed out the disadvantages of using library problems and hence why most research studies use random generated problem instances.

## Introduction

The lateral canal scheduling problem is the problem of preparing a schedule to service the demands from multiple users subject to

[1]Research Student, School of Civil Engineering and the Environment, Univ. of Southampton, Highfield, Southampton SO17 1BJ, U.K. E-mail: shanza@soton.ac.uk

[2]Senior Lecturer, School of Civil Engineering and the Environment, Univ. of Southampton, Highfield, Southampton SO17 1BJ, U.K. (corresponding author). E-mail: a.a.anwar@soton.ac.uk

[3]Lecturer, School of Civil Engineering and the Environment, Univ. of Southampton, Highfield, Southampton SO17 1BJ U.K. E-mail: d.clarke@soton.ac.uk

Note. Discussion open until May 1, 2009. Separate discussions must be submitted for individual papers. The manuscript for this paper was submitted for review and possible publication on May 4, 2007; approved on March 11, 2008. This paper is part of the ***Journal of Irrigation and Drainage Engineering***, Vol. 134, No. 6, December 1, 2008. ©ASCE, ISSN 0733-9437/2008/6-737–744/$25.00.

certain constraints. Reddy et al. (1999) analyzed this problem using a "time blocks" approach with each user allocated a time window within which an irrigation outlet could be scheduled within the capacity of the channel. Anwar and Clarke (2001) used the "stream tube" approach first suggested by Suryavanshi and Reddy (1986). Anwar and Clarke (2001) considered each outlet to have a target start time and a schedule was prepared where both the earliness/tardiness (difference between target and scheduled start times) and the number of stream tubes utilized, i.e., discharge at which lateral was operated—were both minimized. This was a dual goal objective function problem. These earlier works on the lateral canal scheduling problem used integer programming to implement a solution. In contrast Wardlaw and Bhaktikul (2004) used genetic algorithms to solve the lateral canal scheduling problem. In the formulation of a genetic algorithm for the lateral canal scheduling problem Wardlaw and Bhaktikul developed a model that minimized earliness/tardiness only and not both earliness/tardiness and the number of stream tubes as in the original integer program by Anwar and Clarke (2001). Therefore a simplified version of the integer program by Anwar and Clarke (2001) is presented here that has the single objective of minimizing earliness/tardiness. This integer program (IP) will be referred to as the IP and the genetic algorithm as formulated by Wardlaw and Bhaktikul (2004) will be referred to as the formulated GA.

Let $Q=[0,1,2,3\ldots N]$ be the set of irrigation outlets to be scheduled, each with a duration and target start time. A fictitious outlet 0 with duration 0 and target start time 0 is introduced to simplify writing some of the constraints. The decision variable in the lateral canal scheduling problem is the scheduled start time of each of the irrigation outlets. The objective function is to minimize the total earliness/tardiness and is given by

$$\min Z_{\mathrm{IP}} = \sum_{j=1}^{M} (\alpha_j E_j + \beta_j T_j) \tag{1}$$

where $Z_{\mathrm{IP}}$=objective function of the integer program; $j$=index $1,2\ldots M$; $M$=total number of outlets; $\alpha_j$=unit cost of earliness for outlet $j$; $E_j$=earliness of outlet $j$; $\beta_j$=unit cost of tardiness of outlet $j$; and, $T_j$=tardiness of outlet $j$. Any irrigation outlet can directly precede any other outlet on a stream tube; therefore a variable is used to define which outlet precedes which other outlet on what stream tube

$\varphi_{jkw} = 1$ if outlet $j$ *directly* precedes outlet $k$ on stream tube $w$

$= 0$ otherwise, where $j \neq k$

where $\varphi_{jkw}$=binary variable; $w$=index representing stream tubes $1,2,\ldots,W$; and $W$=number of available stream tubes. An outlet can be serviced by any stream tube, so a variable is used to define which outlet is processed by which stream tube

$\tau_{jw} = 1$ if outlet $j$ is processed by stream tube $w = 0$ otherwise

where $\tau_{jw}$=binary variable. The scheduled start time of an irrigation outlet is determined from the target start time, the earliness, and the tardiness

$$S_j = r_j + T_j - E_j \quad \forall j = 1,2\ldots\ldots M \tag{2}$$

where $S_j$=scheduled start time of outlet $j$; and $r_j$=target start time of outlet $j$. It is not possible for a schedule to incur a negative earliness or tardiness (a negative tardiness is in fact a positive earliness and vice versa), therefore the following constraints need to be satisfied:

$$E_j \geqslant 0 \quad \forall j = 1,2\ldots\ldots M \tag{3}$$

$$T_j \geqslant 0 \quad \forall j = 1,2\ldots\ldots M \tag{4}$$

Each outlet can be processed by only one streamtube

$$\sum_{w=1}^{W} \tau_{jw} = 1 \quad \forall j = 1,2\ldots\ldots M \tag{5}$$

Each outlet can at most precede one other outlet

$$\sum_{k=1}^{M} \varphi_{jkw} \leqslant \tau_{jw} \quad \forall j = 0,1,2\ldots\ldots M; \quad \forall w = 1,2\ldots\ldots W; \quad j \neq k \tag{6}$$

Each outlet (with the exception of outlet 0) must follow one other outlet

$$\sum_{j=0}^{M} \varphi_{jkw} = \tau_{kw} \quad \forall k = 1,2\ldots\ldots M; \quad \forall w = 1,2\ldots\ldots W; \quad j \neq k \tag{7}$$

No outlet is allowed to start before the previous outlet on the same stream tube has finished

$$S_j - S_k - C\varphi_{ijw} \geqslant d_j - C \quad \forall j = 0,1,2\ldots M; \quad \forall k = 1,2\ldots M; \quad \forall w = 1,2\ldots W; \quad j \neq k\ldots \tag{8}$$

where $S_k$=scheduled start time of outlet $j$; $C$=large positive constant; and $d_j$=duration of outlet $j$. The solution is not sensitive to the value of the constant $C$ although choosing a very large value may increase computation time. Each outlet should be finished within the interval

$$S_j + d_j \leqslant g \quad \forall j = 1,2\ldots M \tag{9}$$

where $g$=irrigation interval over which all jobs must be scheduled. A stream tube is activated if it services at least one outlet (supplies at least one outlet with water)

$$\sum_{j=1}^{M} \tau_{jw} \leqslant \psi_w W \quad \forall w = 1,2\ldots W \tag{10}$$

where $\psi_w$=binary variable that assumes a value 1 if the stream tube is activated and 0 otherwise. The capacity of a channel may not be exceeded

$$q\sum_{w=1}^{W} \psi_w \leqslant Q \tag{11}$$

where $q$=discharge of a stream tube; and $Q$=channel capacity. For each stream tube the sum of durations must be less than or equal to the length of the irrigation interval. This constraint is not absolutely essential to the model as it achieves the same as constraint (9), but is found to greatly reduce solution times.

$$\sum_{j=1}^{M} d_j \tau_{jw} \leqslant g \quad \forall w = 1,2\ldots\ldots W \tag{12}$$

The objective function for the formulated GA presented by Wardlaw and Bhaktikul (2004) is

$$\text{Minimize: } Z_{GA} = \sum_{i=1}^{N} \left( \left| Q - \sum_{j=1}^{M} (q_j \cdot IFLAG_{ij}) \right| + p_i \right) + \sum_{j=1}^{M} l_j \tag{13}$$

where $Z_{GA}$=objective function of the GA; $i$=index representing time block between time $i$ and time $i+1$ from $1 \ldots N$; $N$ =number of time blocks in the interval; $Q$=supply canal capacity; $j$=index representing outlet $1 \ldots M$; $M$=number of outlets; $q_j$ =capacity of outlet; $IFLAG_{ij}$=integer variable indicating whether or not outlet $j$ is operating in time block $i$; $p_i$=penalty function in time block $i$; and $l_j$=earliness/tardiness of any outlet $j$. The first part on the right-hand side of Eq. (13) is a volume term and the second part is an earliness/tardiness term.

In Eq. (13) the integer variable is defined as

$$IFLAG_{ij} = 1 \quad \text{if } i \geq S_j \text{ and } i < (S_j + d_j) \tag{14}$$

and

$$IFLAG_{ij} = 0 \text{ otherwise}$$

where $S_j$=scheduled start time of outlet $j$; and $d_j$=operating time (duration) of outlet $j$ expressed as an integer number of time blocks. The integer variable assumes a value of 1 if an outlet $j$ is operating in a time block $i$ and zero otherwise. The penalty function is defined as

$$p_i = \sum_{j=1}^{M} (q_j \cdot IFLAG_{ij}) - Q$$

$$\text{if } p_i < 0, \quad p_i = 0 \tag{15}$$

The formulated GA was developed and tested by Wardlaw and Bhahtikul (2004) against the IP for a single instance. A similar (though not identical) GA was also developed and tested by Wardlaw and Bhaktikul (2004) against a single instance of a problem described by Reddy et al. (1999). Wardlaw and Bhaktikul (2004) reported that the formulated GA obtained an objective function of 4.84 against the IP objective function of 4.73. The computation efficiency of the scheduling GA and IP cannot be compared objectively as the hardware and operating systems for the tests were dissimilar. However Wardlaw and Bhaktikul (2004) concluded that the formulated GA produced results almost identical to those of the IP and "…very efficiently." Testing an algorithm for a single instance is useful for validation and verification purposes, however a more systematic and rigorous testing of the formulated GA needs to be conducted before any general conclusions about the efficiency of an algorithm are drawn. The current work describes the results obtained from a more detailed evaluation of the formulated GA.

The comparison of multiperiod schedules as described in Anwar and Clarke (2001) or multi-interval schedules as described by De Vries and Anwar (2004), cannot be used to evaluate the formulated GA because the difference between two schedules may be due to different algorithms, or simply because the weights (unit cost of earliness/tardiness) from the preceding schedules is different. Therefore this evaluation does not consider multiperiod schedules.

This paper explores the performance of the formulated GA for

1. The solution quality as the demand: supply ratio is increased; and
2. The solution quality of the formulated GA as the size of the problem increases.

**Table 1.** Parameters for GA

| | Bhatikul (2001) | Reimplemented GA |
|---|---|---|
| Population size | 100 | 100,150 |
| Probability of mutation | 0.15 | 0.20 |
| Probability of crossover | 0.85 | 0.75 |
| Number of generations | 2,500 | 2,500,3750 |

This paper uses analytical and empirical methods of evaluation as described by Hooker (1995). For the empirical method 140 and 70 problem instances are used for eight and 12 job problems, respectively. The solutions from the formulated GA are evaluated against that of the IP. This limits the evaluation to the relatively small problems that can be solved by the IP, i.e., eight or 12 outlet scheduling problem, therefore any conclusions drawn are only applicable to the scale of problems tested in this evaluation. Although the formulated GA and IP are tested with hypothetical problem as suggested in the operations research literature, the problems have been selected keeping practical irrigation scheduling problems in mind. The eight or 12 outlet problems are similar to the smaller problems cited in the literature. Similarly an irrigation interval of 800 or 1,200 time units can refer to irrigation intervals of 8 or 12 days, respectively (each time unit is approximately 15 min). The durations for these test instances are selected from intervals of 0–400 time units, i.e., the average duration is 200 time units or 50 h. All target and scheduled start times are rounded off to the nearest time unit, e.g., nearest quarter of an hour. The channel capacity of 4,000 discharge units and each outlet of 1,000 discharge units can be interpreted as a channel capacity of 400 L/s and each outlet discharge of 100 L/s.

The source code for the formulated GA was not available to the writers therefore the formulated GA is reimplemented in Java. Key algorithm parameters for any formulated GA are: the size of the initial population, probability of mutation, probability of crossover, and number of generations. Table 1 shows the values selected for the these parameters based on validation and verification tests and compares them to the values used by Bhaktikul (2001) which are assumed to be similar to values used by Wardlaw and Bhaktikul (2004). The values for probability of mutation and probability of crossover are similar to those suggested in the literature for GAs, e.g., for example, Goldberg (1989) and De Jong (1975).

The reimplemented GA is validated against the eight-outlet problem first described by Suryavanshi and Reddy (1986) and used by Anwar and Clarke (2001) to demonstrate application of the IP. The reimplemented GA obtains an optimum solution identical to that of the IP (objective function 4.73). This is an improvement over the objective function reported for the formulated GA of 4.84. However it is incorrect to suggest that the reimplemented GA is in any way better than the formulated GA from a single test instance. Therefore, two computational experiments are designed to test the performance of the reimplemented GA.

Where errors are reported in this paper these are defined as the difference in the objective function value obtained from the reimplemented GA and the IP expressed as a percentage of the IP objective function value.

## Analysis and Discussion

1. Observation 1: The formulated GA if inappropriately scaled may obtain "fitter" genes that represent infeasible solutions.

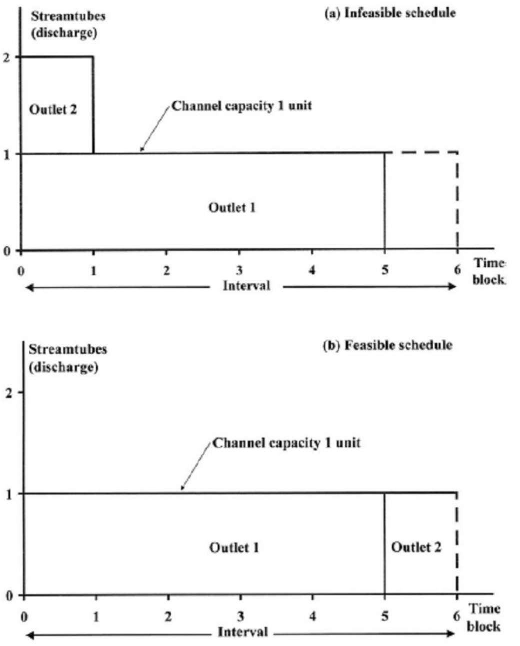



**Fig. 1.** Graphical illustration for Observation 1

It can be demonstrated (see the Appendix) that for all feasible schedules, the volume term in the formulated GA in Eq. (13) does not contribute to the objective function. If the $i$th time block in a schedule is infeasible, i.e., the demand volume exceeds the supplied volume, for this $i$th time block, from Eq. (15), the term $(|Q-\Sigma_{j=1}^{M}(q_j \cdot IFLAG_{ij})|+p_i)$ in Eq. (13) becomes $2(\Sigma_{j=1}^{M}(q_j \cdot IFLAG_{ij})-Q)$. Therefore for every time block where the demand volume of all outlets exceeds the supply volume the objective function increases by the term $2(\Sigma_{j=1}^{M}(q_j \cdot IFLAG_{ij})-Q)$ which is simply twice the excess demand. However depending on how an actual problem is scaled, i.e., whether time is represented by seconds, minutes, hours, and likewise whether discharge is represented in L/s or $m^3/s$ or some other unit, e.g., $100\times m^3/s$, the increase in the objective function from the volume term can be of the same order of magnitude as the earliness/tardiness term. Therefore if a schedule is produced such that the earliness/tardiness can be reduced by slightly more than the penalty of an infeasible schedule, the GA will evaluate the infeasible schedule as "fitter." This is illustrated with the following example.

Consider the simple system with two outlets with both outlets having a target start time of one (beginning of time block 1). However in this case assume that the capacity of the channel is one unit of discharge. Fig. 1(a) represents an infeasible schedule. Both outlets start on the target start time therefore there is no earliness/tardiness and the objective function of the formulated GA from Eq. (13) evaluates to three. In time block 1, the capacity of the channel is exceeded by one unit of discharge and in time block 6, there is one unit of discharge unused, and in all other time blocks, the capacity is equal to the sum of discharge of all outlets.

**Table 2.** Parameters for Experiment 1

| | |
|---|---|
| Number of outlets | 8 |
| Channel capacity | 4,000 |
| Irrigation interval | 800 |
| Duration of each outlet | Uniformly distributed random integer from the range (0,400) |
| Target start time | Uniformly distributed random integer from the range (0,800) |
| Demand: supply ratio | 0.1, 0.3, 0.5, 0.7, 0.9 |
| Number of instances | 140 for each demand:supply ratio |

In contrast Fig. 1(b) represents a feasible schedule as the capacity of the channel is not exceeded in any time block during the interval. Outlet 1 starts on the target start time—Time block 1, but Outlet 2 starts at Time block 6, therefore is tardy by five time blocks and the objective function of the formulated GA evaluates to five. Hence the formulated GA would evaluate the infeasible schedule of Fig. 1(a) as "fitter" than the feasible schedule of Fig. 1(b). This infeasible schedule will form part of the population for the next iteration. Such an infeasible schedule may even become the accepted solution when the formulated GA iterations are terminated.

2. Observation 2: The solution quality of the reimplemented GA is sensitive to the demand: supply ratio, and the solution quality deteriorates at high levels of this ratio.

## Experiment 1

This experiment is designed to test the quality of the solution of the reimplemented GA against demand. At higher levels of demand, the scheduling problem becomes computationally considerably more complex as there is less idle time available within the scheduling interval. To test this hypothesis, the parameter demand:supply ratio is introduced which is expressed as

$$r_{DS} = \frac{\Sigma_{j=1}^{M} q_j \cdot d_j}{NQ} \tag{16}$$

where $r_{DS}$=demand-supply ratio. The demand-supply ratio is a measure of the surplus capacity available in the irrigation schedule. It is similar to the interval:makespan ratio used by Anwar and de Vries (2004). The latter is only applicable when outlets operate sequentially, whereas the demand supply ratio is applicable when outlets operate simultaneously as is the present case. In contrast to the interval:makespan ratio, the demand supply ratio ranges from $0 < r_{DS} \leq 1$.

Table 2 summarizes the parameters used to generate data for the experiment. For this experiment, there are eight outlets to be serviced each with 1,000 units of discharge, and the channel capacity is 4,000 units of discharge. The irrigation interval within which all outlets must be serviced is 800 units of time. The duration for which each outlet is operated is a uniformly distributed random number over the range 0–400. The target start time of each outlet is also a uniformly distributed random number over the range 0–800.

For each demand-supply ratio in Table 2 a test instance was generated. A test instance consists of eight uniformly distributed random numbers representing the duration of each outlet and eight uniformly distributed random numbers representing the target start time. The ranges for these numbers are shown in Table 2. If the sum of target start time and duration for any outlet exceeded the irrigation interval the test instance is rejected and a new test

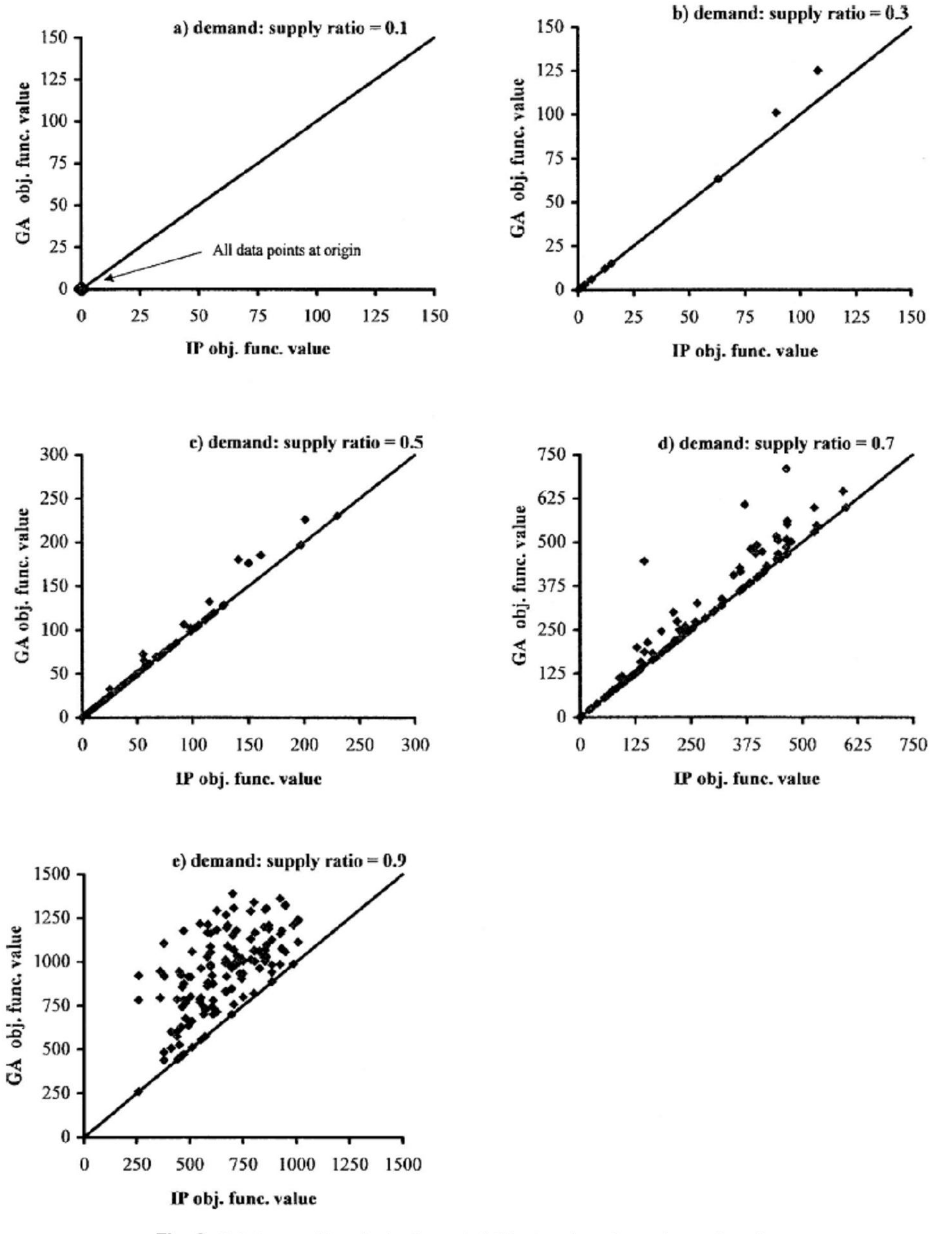



**Fig. 2.** Solution quality of reimplemented GA at various demand: supply ratios

instance is generated. For a given instance, from the generated durations of each outlet the demand-supply ratio is calculated. If this lies within a tolerance of +0.1% of the demand:supply ratio in Table 2, the test instance is retained, otherwise it is rejected and a new test instance is generated. This process is repeated until 140 instances are produced for each demand:supply ratio. Both the IP and reimplemented GA are run using this test data. The IP is terminated when it obtains a global optimum, whereas the reimplemented GA has a stopping criteria of a maximum of 2,500 generations.

Figs. 2(a–e) shows the objective function of the reimplemented GA relative to that of the IP. For very low demand:supply ratios ($\approx$0.10), it is relatively easy to find a solution. At this low level of demand, each of the 140 schedules can be prepared to

**Table 3.** Analysis of Experiment 1

| | Demand: supply ratio | | | | |
|---|---|---|---|---|---|
| | 0.10 | 0.30 | 0.50 | 0.70 | 0.90 |
| Mean error (%) | 0.00 | 0.21 | 1.32 | 7.64 | 49.67 |
| Min. error (%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Max. error (%) | 0.00 | 15.74 | 5.14 | 206.90 | 253.85 |
| Standard deviation of error | 0.00 | 0.15 | 0.05 | 0.21 | 0.42 |
| Optimum solutions | 140/140 | 138/140 | 129/140 | 87/140 | 9/140 |
| Average generation at termination | 1,274 | 1,273 | 1,280 | 1,265 | 286 |
| Maximum generation at termination | 1,314 | 1,291 | 1,316 | 1,490 | 1,331 |

deliver water to each outlet at the target start time and therefore the earliness/tardiness in every schedule is zero, and all schedules plot at the origin in Fig. 2(a). In fact at this low level it is just as easy to prepare a schedule manually. Figs. 2(b–e) show that as the demand increases, it becomes increasingly difficult to find the optimum schedule and the solution quality of the reimplemented GA deteriorates. At the highest level of demand-supply ratio tested ($\approx$0.90), of the 140 instances tested, the reimplemented GA produces 53 nonoptimum solutions.

A more comprehensive analysis of the results is presented in Table 3. When the demand-supply ratio increases from 0.70 to 0.90, the mean error increases significantly to approximately 50%. The reimplemented GA is able to find optimum solutions even at the highest demand in only nine of the 140 instances. This experiment suggests the reimplemented GA performs well up to a demand-supply ratio of 0.70, demonstrating Observation 2. In order to put the error into context it is necessary to consider an individual instance rather than averages. For one particular instance for demand-supply ratio of 0.7 the relative error is 6.3%—close to the mean error for this demand-supply ratio reported in Table 3. For this particular instance the objective function value from the IP is 228 against that obtained by the GA of 242. The difference is therefore 14 units of time. If the interval of 800 time units is taken to represent 8 days, i.e., all times and durations are rounded off to 1/100th of a day (approximately 15 min), then a total earliness/tardiness of 14 time units is approximately 210 min, or given there are eight outlets, 26 min/outlet. This implies that on average each outlet receives water either 26 min earlier or later than the target start time because the GA is unable to find a better solution which does exist as demonstrated by the IP.

Table 3 also reports the average number of generations beyond which no further improvement in the objective function was found. For example, for a demand:supply ratio of 0.10 the "best" schedule was obtained at Generation 1,274 although the algorithm was allowed to run to the limiting value of 2,500 generations. It is important to mention that this evaluation of demand-supply ratio and the performance of the reimplemented GA was only tested for a problem size of eight outlets. Experiment 2 demonstrates that the computational complexity of the scheduling problem also increases with the size of the problem (number of outlets). For larger problems, the reimplemented GA may behave differently at the demand-supply ratio tested in this experiment.

3. Observation 3: The solution quality of the reimplemented GA deteriorates as the size of the problem increases.

### Experiment 2

This experiment is designed to test the quality of the solution as the problem size (number of outlets) increases. The relative error as described in Experiment 1 is used as a measure of the solution quality of the reimplemented GA. Table 4 summarizes the parameters used to generate test data for this experiment. For each of the number of outlets, a data set was generated with the parameters shown and the demand: supply ratio calculated using Eq. (16). If the calculated ratio is within a tolerance of +0.1% the test instance is retained, otherwise it is rejected and another test instance generated. This process was repeated to obtain 140 test instances for a group of eight outlets and then for 12 outlets. For the reimplemented GA where the problem size increases by 50% from an eight outlet problem to a 12 outlet problem, the initial population is also increased by the same proportion (100–150) and the maximum generations is also increased by this proportion (2,500–3,750). A stopping criteria is also introduced to prevent unnecessary iterations. The early stopping criteria used in this experiment continuously monitors the objective function value and if no improvement is detected over 1,000 successive iterations, the algorithm is terminated. Using the IP and the reimplemented GA, schedules were obtained for each of the data sets and the objective function values compared. Due to the excessive time

**Table 4.** Parameters for Experiment 2

| | | |
|---|---|---|
| Number of outlets | 8 | 12 |
| Channel capacity | 4,000 | 4,000 |
| Irrigation interval | 800 | 1,200 |
| Duration of each outlet | Uniformly distributed random integer from the range (0,400) | Uniformly distributed random integer from the range (0,400) |
| Target start time | Uniformly distributed random integer from the range (0,800) | Uniformly distributed random integer from the range (0,1200) |
| Demand: supply ratio | 0.50 | 0.50 |
| Number of instances | 140 | 70 |
| Limiting number of generations | 2,500 | 3,750 |
| Initial population | 100 | 150 |

**Table 5.** Analysis of Experiment 2

| | Number of outlets | |
|---|---|---|
| | 8 | 12 |
| Mean error (%) | 1.32 | 6.79 |
| Min. error (%) | 0.00 | 0.00 |
| Max. error (%) | 30.91 | 133.33 |
| Standard deviation | 0.05 | 0.20 |
| Optimum solutions | 129/140 (92%) | 44/70 (63%) |
| Nonoptimal solutions | 11/140 (8%) | 26/70 (37%) |
| Average generation at termination | 1,268 | 1,546 |
| Maximum generation at termination | 1,316 | 2,363 |

the IP takes to solve (3 h on average) and produce a schedule, only 70 instances were used in the group of 12 outlets rather than the full complement of 140.

Table 5 summarizes the results from this experiment. An increase in problem size by 50% results in an increase in the mean error from 1.32 to 6.79%. The number of nonoptimum solutions also increases from 11% to 37% for this 50% increase in problem size. The standard deviation also increases with problem size from 0.05 to 0.20 (80% increase). Table 5 also shows the number of generations after which the stopping criteria was invoked. For the eight job problem the average number of generations was 1,274, i.e., although the GA was allowed to run by a further 1,000 generations to 2,274, no further improvement in the solution was obtained. Similarly for the 12 job problem the average number of generations was 1,546. Again the GA was permitted to run an additional 1,000 generations to 2,546 but no further improvement was obtained. Table 5 also reports the maximum number of generations of any one instance before the early stopping criteria halted the iterations and it can be seen that for both the eight job problem and the 12 job problem the maximum generations of all instances is far less than the 2,500 and 3,750 limits set in Table 4. For all experiments the stopping criteria halted the experiment rather than the limiting number of generations

Comparing Tables 5 and 3, the performance of the reimplemented GA with 12 outlets with a demand-supply ratio of 0.5 has deteriorated to that of eight outlets with a demand: supply ratio of 0.70. It is difficult to evaluate the reimplemented GA in this manner for problem sizes beyond 12 as the IP takes inordinately long to solve. However this experiment strongly indicates that the solution quality of the reimplemented GA is sensitive to problem size and for a problem with 12 outlets the reimplemented GA can only be used at a relatively low demand: supply ratio of 0.50.

## Conclusion

The reimplemented GA is computationally more efficient than the IP and appears to have considerable scope as a tool to solve this specific problem of preparing an irrigation schedule. For a problem with eight users, the reimplemented GA produces good schedules for minimizing earliness/tardiness for demand:supply ratio of 0.7 but the solution quality deteriorates as the demand-:supply ratio increases and as the problem size grows. The experiments conducted in this paper further suggest that the reimplemented GA may not produce reasonable quality solutions beyond a problem size of 12. This paper emphasizes the need for rigorous testing of heuristics such as genetic algorithms as is suggested in the literature cited in operations research. The reimplemented GA would need to be tested more exhaustively for the solution quality of such schedules against a suitably developed IP or other heuristic.

It is reiterated that the formulated GA refers to the formulation of the genetic algorithm by Wardlaw and Bhakitikul (2004) which was reimplemented for this research. The IP refers to a simplified version of the implementation of an integer program as described by Anwar and Clarke (2001). The conclusions drawn through this paper refer to these specific implementations and should not be construed as general comments about genetic algorithms and integer programs by the writers. In general, genetic algorithms and indeed other evolutionary heuristics do offer considerable potential to solve the scheduling problem. Numerous examples of such research and application can be found in operations research literature and these techniques are the technique of choice for large problems. It is recommended that researchers using such evolutionary heuristics to solve problems in irrigation and water management take advantage of the wealth of literature on algorithms, data sets, testing, and reporting results that can be found in operations research.


## Acknowledgments

The writers wish to acknowledge the contribution of Kristin Welter to this work who carried out some of the early coding of the reimplemented GA and some of the computational experiments.


## Appendix

The penalty function in Eq. (15) is designed to add a positive value to the objective function if in a given schedule the channel capacity is exceeded in any time block. In effect the objective function is penalized for infeasible schedules. Since the penalty factor reduces to zero for all feasible schedules where the channel capacity is not exceeded, for all feasible schedules Eq. (13) can be rewritten as

$$\text{Min.: } Z_{GA} = \left( NQ - \sum_{i=1}^{N} \sum_{j=1}^{M} (q_j \cdot IFLAG_{ij}) \right) + \sum_{j=1}^{M} l_j \qquad (17)$$

The term $\Sigma_{i=1}^{N}\Sigma_{j=1}^{M}(q_j \cdot IFLAG_{ij})$ in Eq. (17) is simply the demand volume of all outlets for all time blocks within this interval. Therefore, for feasible schedules $NQ-\Sigma_{i=1}^{N}\Sigma_{j=1}^{M}(q_j \cdot IFLAG_{ij})$ in Eq. (17) is a positive constant (or zero) which is redundant and therefore can be removed from the objective function altogether. Hence for any schedule that does not exceed the capacity of the channel Eq. (13) simply reduces to

$$\text{Min.: } Z_{GA} = \sum_{j=1}^{M} l_j \qquad (18)$$

## Notation

*The following symbols are used in this paper:*

- $C$ = large positive constant;
- $d_j$ = operating time (duration) of outlet $j$ in time block units;
- $E_j$ = earliness of outlet $i$;
- $g$ = irrigation interval;

$IFLAG_{ij}$ = integer variable;
$i$ = index representing time block $1, 2 \ldots . N$;
$j$ = index representing outlet $1, 2 \ldots . M$;
$k$ = index representing outlet $1, 2 \ldots . M$;
$l_j$ = earliness/tardiness of any outlet $j$;
$M$ = total number of irrigation outlets;
$N$ = total number of time blocks;
$p_i$ = penalty factor applied to time block $i$;
$Q$ = channel capacity;
$q$ = discharge of stream tube;
$q_j$ = discharge at outlet $j$;
$R_t$ = relative timeliness;
$r_{DS}$ = demand:supply ratio;
$r_j$ = target start time of outlet $j$;
$S_j$ = scheduled start time of outlet $j$;
$T_j$ = tardiness of outlet $j$;
$W$ = number of available stream tubes;
$w$ = index representing stream tubes $1, 2, \ldots, W$;
$Z_{GA}$ = objective function obtained from GA;
$Z_{IP}$ = objective function obtained from IP;
$\alpha_j$ = unit cost of earliness for outlet $j$;
$\beta_j$ = unit cost of tardiness of outlet $j$;
$\tau_{jw}$ = binary variable, assumes value of 1 if outlet $j$ is processed on streamtube $w$, 0 otherwise;
$\varphi_{jkw}$ = binary variable, assumes value of 1 if outlet $j$ *directly* precedes outlet $k$ on stream tube $w$, 0 otherwise; and
$\varphi_w$ = binary variable that assumes a value 1 if the stream tube $w$ is activated and 0 otherwise.

## References


Anwar, A. A., and Clarke, D. (2001). "Irrigation scheduling using mixed-integer linear programming." *J. Irrig. Drain. Eng.*, 127(2), 63–69.

Anwar, A. A., and de Vries, T. T. (2004). "Irrigation scheduling II: Heuristics approach." *J. Irrig. Drain. Eng.*, 130(1), 17–25.

Barr, R. S., Golden, B. L., Kelly, J. P., Resende, M. G. C., and Steward, W. R. (1995). "Designing and reporting on computational experiments with heuristic methods." *J. Heuristics*, 1, 9–32.

Bhaktikul, K. (2001). "The development of a genetic algorithm for real time water allocation and water scheduling in complex irrigation systems." Ph.D. thesis, Univ. of Edinburgh, Edinburgh, U.K.

De Jong, K. A. (1975). "An analysis of the behaviour of a class of genetic adaptive systems." Ph.D. dissertation, Univ. of Michigan, Ann Arbor, Mich.

De Vries, T., and Anwar, A. A. (2004). "Irrigation scheduling I: Integer programming approach." *J. Irrig. Drain. Eng.*, 130(1), 17–25.

Goldberg, D. E. (1989). "Generating experimental data for computational testing with machine scheduling applications." *Oper. Res.*, 49(7), 854–865.

Hall, N. G., and Posner, M. E. (2001). "Generating experimental data for computational testing with machine scheduling applications." *Oper. Res.*, 49(7), 854–865.

Holland, J. H. (1975). *Adaptation natural and artificial systems*, MIT Press, Cambridge, Mass.

Hooker, J. N. (1995). "Testing heuristics: We have it all wrong." *J. Heuristics*, 1, 33–42.

Rardin, R. L., and Uzsoy, R. (2001). "Experimental evaluation of heuristic optimization algorithms: A tutorial." *J. Heuristics*, 7, 261–304.

Reddy, J. M., Wilamowski, B., and Cassel-Sharmasarkar, F. (1999). "Optimal scheduling of irrigation for lateral canals." *ICID J.*, 48(3), 1–12.

Rylander, B. (2001). "Computational complexity and the genetic algorithm." Ph.D. dissertation, Univ. of Idaho, Idaho.

Suryavanshi, A. R., and Reddy, J. M. (1986). "Optimal operation schedule of irrigation distribution systems." *Agric. Water Manage.*, 11(1), 23–30.

Wardlaw, R., and Bhaktikul, K. (2004). "Comparision of genetic algorithm and linear programming approaches for lateral canal scheduling." *J. Irrig. Drain. Eng.*, 130(4), 311–317.

# References


Allahverdi, A., Gupta, J. N. D., and Aldowaisan, T. (1999). "A review of scheduling research involving setup considerations." *Omega-International Journal of Management Science*, Elsevier, 27(2), 219-239.

Andrzej, O. and Stanislaw, K. (2000). "A new constraint tournament selection method for multicriteria optimization using genetic algorithm", *IEEE Conf. Evolutionary Computation,* 501–508.

Anwar, A.A., and Clarke, D. (2001). "Irrigation scheduling using mixed integer linear programming." *J. Irrig. Drain. Eng*., 127(2), 63-69.

Anwar, A.A., and De Vries, T.T. (2004). "Irrigation scheduling II: Heuristics approach". *J. Irrig. Drain. Eng*., 130(1),17-25.

Anwar, A.A., and Clarke, D., and De Vries, T.T. (2006). "Channel capacity under arranged demand irrigation". *Agricultural Water Management,* 82, 148-160.

Arostegui, Jr., M.A., Kadipasaoglu, S.N. and Khumawala, B.M. (2006). "An empirical comparison of tabu search simulated annealing, and genetic algorithms for facilities location problems". *Int. J. Production Economics*, 103, 742–754.

Baker, K. R. (1974). *Introduction to sequencing and scheduling*, Wiley and Sons, New York, USA.

Baker K.R. and Scudder G.D. (1990)."Sequencing with earliness and tardiness penalties: A review". *Operations Research, Operations Research Society of America,* 38(1), 22-36.

Bean, J.C. (1994). "Genetics and random keys for sequencing and optimization". *ORSA J Compu., t* 6, 154–160.

Beasley, D., Bull, D.R. and Martin, R. R. (1993a). "An Overview of Genetic Algorithms: Part I, Fundamentals". *University Computing*, 15(2), 58-69.

Beasley, D., Bull, D.R. and Martin, R. R. (1993b). "An Overview of Genetic Algorithms: Part 2, Research Topics". *University Computing*, 15(4), 170-181.

Black, P.E. (2006). *NP-hard*, Dictionary of Algorithms and Data Structures [online], U.S. National Institute of Standards and Technology, Available from: http://www.nist.gov/dads/HTML/nphard.html [accessed 3 December 2007]

Bishop, A. A., and Long, A. K. (1983). ‘‘Irrigation water delivery for equity between users.’’ *J. Irrig. Drain. Eng.,* 109(4), 349–356.

Blum, C. and Roli, A. (2003). “Metaheuristics in combinatorial optimization: overview and conceptual comparison”. *ACM Computing Surveys,* 35(3):268-308.

Brown, D.E., Markowitz, B.P. and Sappington, D.E. (1992). “Rail network routing and scheduling using simulated annealing”. *IEEE International Conference on Systems, Man and Cybernetics,* 1, 589-592.

Buchleiter, G. W. and Heermann, D. F. (1987). “System scheduling for large farming operations”. *Planning, Operation, Rehabilitation and Automation of Irrigation Water Delivery Systems*: *Proceedings of a Symposium, ASCE, USA,* 45-55.

Chambers, R. (1983).”Managing Canal Irrigation.”. *Oxford Publications,* New Delhi.

Chen, Y.M. (1997). “Management of water resources using improved genetic algorithms”. *Computers and Electronics in Agriculture,* 18, 117-127.

Cheng, R., Gen, M. and Tsujimura, Y. (1996). “A tutorial survey of job-shop scheduling problems using genetic algorithms-1. Representation”. *Computer ind. Engng,* 30(4), 983-997.

Chopart, J.L., Mezino, M., Aure, F., Le Meezo, L., Mete, M. and Vauclin, M. (2007). “OSIRI: a simple decision making tool for monitoring irrigation of small farms in heterogeneous environments”. *Agricultural Water Management,* 87, 128-138.

Clark, R. and King, J. (2004). *The atlas of water,* Myriad Editions Limited, UK.

Clemens, A. J. (1987a). “Delivery system schedules and required capacities”. *Planning, Operation, Rehabilitation and Automation of Irrigation Water Delivery Systems*: *Proceedings of a Symposium, ASCE, USA,* 18-34.

Clemens, A. J. (1987b). “Arranged delivery schedules”. *Planning, Operation, Rehabilitation and Automation of Irrigation Water Delivery Systems*: *Proceedings of a Symposium, ASCE, USA,* 57-67.

Coello, C.A.C. (2002). “Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art”. *Comput. Methods Appl. Mech. Engng.,* 191, 1245-1287

Colin, E.C. and Quinino, R.C. (2005). “An algorithm for insertion of idle time in the single machine scheduling problem with convex cost functions”. *Computer & Operations Research,* 32, 2285-2296.

Coley D.A., (1999) *An Introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific Publishing Co. Pte. Ltd.

Comprehensive Assessment of Water Management in Agriculture. (2007). *Water for Food, Water for Life: A Comprehensive Assessment of Water Management in Agriculture.* Chapter 2, London: Earthscan, and Colombo: International Water Management Institute.

Dadios, E.P. and Ashraf, J. (2006). "Genetic algorithm with adaptive and dynamic penalty functions for the selection of cleaner production measures: a constrained optimization problem". *Clean Tech Environ Policy,* 8, 85-95.

Damodaran, P., Manjeshwar, P.K. and Srihari, K. (2006). "Minimizing makespan on a batch-processing machine with non-identical job sizes using genetic algorithms". *International Journal of Production Economics*, 103(2), 882-891.

Davis, L. (1991) *A handbook of genetic algorithms*, Van Nostrad Reinhold, New York.

Deb, K. (2000). "An efficient constraint handling method for genetic algorithms". *Comput. Methods Appl. Mech. Engng.*, 186, 311-338.

De Jong, K. A. and Spears, W. M. (1989). " Using genetic algorithms to solve NP-complete problems". *Proc. Of the third Int. Conf. on Genetic Algorithms*.

De Sherbinin, A. and Dompka, V. (1996). "Water and population dynamics: Case studies and policy implications." Report of a Workshop, Montreal, Canada.

De Vries, T.T. (2003). "*Irrigation scheduling with integer programming*". Ph.D. Thesis. School of Civil Engg. & the Environment, University of Southampton, U.K.

De Vries, T.T., and Anwar, A.A. (2004). "Irrigation scheduling I: Integer programming approach." *J. Irrig. Drain. Eng.,* 130(1), 9-16.

De Vries, T.T., and Anwar, A.A. (2006). "Irrigation scheduling with travel times." *J. Irrig. Drain. Eng.,* 132(3), 220-227.

Falkenmark, M. and Rockstrom, J. (2006). "The new blue and green water paradigm: Breaking new ground for water resources planning and management." Journal of Water Resources Planning and Management, Editorial, ASCE. 129-132.

F.A.O. (2006). "Stakeholder-oriented valuation to support water resources management processes". Water Report 30, www.iwmi.cgiar.org

Fonseca, C. M. and Fleming, P. J.(1998). "Multiobjective optimization and multiple constraint handling with evolutionary algorithms–Part I: A unified formulation," *IEEE Trans. Syst., Man, Cybern.*, 28(1), 26–37.

Garey, M.R., Graham, R.L., and Johnson, D.S. (1978). “Performance guarantees for scheduling algorithms”. *Operations research*, 26(1), 3-21.

Garey, M.R., Tarjan, R.E., and Wilfong, G.T. (1988). “One-processor scheduling with symmetric earliness and tardiness penalties”. *Mathematics of Operations Research*, 13(2), 330-348.

Gen, M. and Cheng, R. (1996). “A survey of penalty techniques in genetic algorithms”. *Proc. of 1996 IEEE Int. Conf. on Evolutionary Computation*, 804-809.

Gleick, P. H. (2003). “Water use” *Annual Review of Environment and Resources*, 28, 275-314.

Goldberg, D.E. (1989). “Genetic algorithms in search, optimization and machine learning”. *Addison-Wesley Publishing Co., Inc., Redwood City, CA*

Goldhamer, D. A. and Fereres, E. (2004). “Irrigation scheduling of almond trees with trunk diameter sensors”. *Irrig Sci.* 23, 11-19.

Goncalves, J.F., Mendes, J.J.M. and Resende, M.G.C. (2005). “A hybrid genetic algorithm for the job shop scheduling problem”. *European Journal of Operational Research*, 167, 77–95.

Gorantiwar, S. D., and Smout, I. K. (2003). “Allocation of scarce water resources using deficit irrigation in rotational systems.” *J. Irrig. Drain. Eng.*, 129(3), 155–163.

Gorantiwar, S. D., and Smout, I. K. (2005). “Multilevel approach for optimizing land and water resources and irrigation deliveries for tertiary units in large irrigation schemes. II: Application”. *J. Irrig. Drain. Eng.*, 131(3), 264–272.

Gupta, P.K. and Hira, D.S. (2007). *Operations research*, S. Chand & Company Ltd. ISBN 8121902819

Goussard, J. (1995). “Interaction between water delivery and irrigation scheduling”. *Irrigation Scheduling: From Theory to Practice*, Water Report 8, Proceedings of the ICID/FAO Workshop on Irrigation Scheduling, 12-13, Sep. 1995, Rome, Italy. ISBN 92-5-103968-2.

Hall, N.G. and Posner, M.E. (2001). “Generating experimental data for computational testing with machine scheduling applications”. *Operations Research*, 49 (7), 854- 865.

Haq, Z. U., Anwar, A. A., and Clarke, D. (2008). “Evaluation of a Genetic Algorithm for the Irrigation Scheduling Problem”. *J. Irrig. Drain. Eng.*, 134(6), 737-744.

Haral, U., Chen, R.W., Ferrell Jr., W.G, Kurz, M.B. (2007). "Multiobjective single machine scheduling with nontraditional requirements". *Int. J. Production Economics,* 106, 574-584.

Hart, E., Ross, P. and Corne, D. (2005). "Evolutionary scheduling: A review". *Genetic Programming and Evolvable Machines*, 6, 191–220.

Hwang, S.F. and He, R.S. (2006). "Improving real-parameter genetic algorithm with simulated annealing for engineering problems". *Advances in Engineering Software,* 37, 406-418.

Heady, R. B., Zhu, Z. (1998). "Minimizing the sum of job earliness and tardiness in a multimachine system". *Int. J. Prod. Res*. 36, 1619-1632

Heermann, D. F. (1980). "Irrigation scheduling". *Operations Research in Agriculture and Water Resources,* 501-516.

Higgins P. G. and Wirth , A. (1995). "Interactive job-shop scheduling: How to combine operations research heuristics with human abilities".*6th International Conference On Manufacturing*, Melbourne, 293-302.

Hill, R. W. and Allen, R. G. (1996). "Simple irrigation scheduling calendars". *J. Irrig. Drain. Eng.,* 122(2), 107-111.

Himilton, D.L. and DeVries, J.J. (1986). "SCHEDULE: computer programs for scheduling irrigation canal water deliveries". *Microsoftware for Engineers,* 2(2), 65-74.

Hooker (1995). "Testing heuristics: we have it all wrong".*J. Huristics,* 1, 33-42.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

IWMI, (2003). "Improving water productivity: How do we get more crops from every drop?" Water Policy Briefing, issue-8, http://www.iwmi.cgiar.org/waterpolicybriefing

Jain, A.S. and Meeran, S. (1999). "Deterministic job-shop scheduling: Past, present and future". *European Journal of Operational Research*, 113, 390-434.

Jahangir, W. A., Ashfaq, M. and Rehman, A. (2003). "Modeling for efficient use of canal water at command level". Pak. J. of Water Res. 7(1), 43-52.

Jensen, M.T. (2001). "*Robust and Flexible Scheduling with Evolutionary Computation*". Ph.D. Thesis. Faculty of Science, University of Aarhus, Denmark.

Jones, H. G. (2004). "Irrigation scheduling: advantages and pitfalls of plant-based methods". *Journal of Experimental Botany,* 55(407), 2427-2436.

Jurriens, R. and De Jong, K. (1989). *Irrigation water management - a literature survey,* Working Party "Irrigation and Development", Department of Irrigation and Soil and Water Conservation, Agricultural University Wageningen, The Netherlands, 115-133.

Jurriens, M. and Wester, P. (1995). "Warabandi revisited: to abandon or to improve a basically sound water delivery scheduling method?". *Annual Report: International Institute for land Reclamation and Improvement/ILRI*, 24-43.

Kanet, J.J and Sridharan, V. (2000). "Scheduling with inserted idle time: problem taxonomy and literature review". *Operations Research*, 48(1), 99-110.

Khan, S., Tariq, T., Yuanlai, C., and Blackwell, J. (2006). "Can irrigation be sustainable?". *Agricultural water Management*, 80, 87-99.

Khepar, S. D., Gulati, H. S. and Yadav, A. K. (2000). "A model for equitable distribution of canal water." *Irrig. Sci., Spinger-Verlag*, 19, 191-197.

Kimms, A. (1999). "A genetic algorithm for multi-level, multi-machine lot sizing and scheduling". *Computer & Operations Research*, 26, 829-848.

Kim, J.U. and Kim, Y.D. (1996). "Simulated annealing and genetic algorithms for scheduling products with multi-level product structure". *Computers Ops Res.*, 23(9), 857-868.

Kirpich, P. Z., Haman, D. Z., and Styles, S. W. (1999). "Problems of irrigation in developing countries". *Journal of Irrigation and Drainage Engineering*, 125 (1), ISSN 0733-9437/99/0001-0001–0006

Kuo, S. F., Liu, C. W. and Chen, S. K. (2003). "Comparative study of optimization techniques for irrigation project planning". *Journal of the American Water Association*, 39(1), 59-73.

Latif, M., Sarwar, S. (1994). "Proposal for equitable water allocation for rotational irrigation in Pakistan." *Irrigation and Drainage Systems*, 8 (1), 35-48.

Lauff V. and Werner F. (2004). "Scheduling with common due date, earliness and tardiness penalties for multimachine problems: A survey", *Mathematical and Computer Modeling* .40, 637-655.

Lee, C.Y. and Choi, J.Y. (1995). "A genetic algorithm for job sequencing problems with distinct due dates and general early-tardy penalty weights". *Computers Ops Res.*, 22(8), 857-869.

Lee, J. and Kim, Y. (1996). "Search heuristics for resource constrained project scheduling". *Journal of the Operational Research Society*, 47(5), May, 1996, 678-689

Leib, B.G., Elliott, T.V., and Matthews, G. (2001). “ WISE: a web linked and producer oriented program for irrigation scheduling”. *Computers and Electronics in Agriculture*, 33,1-6.

Li, Y., Ip, W.H., and Wang, D.W. (1998). “Genetic algorithm approach to earliness and tardiness production scheduling and planning problem”. *Int. J. Production Economics*. 54, 65-76.

Lucasius C.B., Kateman G. (1993). “Understanding and Using Genetic Algorithms, Part 1. Concepts, Properties and Context.” *Chemometrics and Intelligent Laboratory Systems*, 19, 1-33

Lucasius C.B., Kateman G. (1994). “Understanding and Using Genetic Algorithms, Part 2. Representation, configuration and hybridization.” *Chemometrics and Intelligent Laboratory Systems*, 25, 99-145

Makin, I.W. and Cornish, G.A. (1995). “Irrigation scheduling at system level: An analysis of practical applications of the INCA software.” *Proceedings of the ICID/FAO Workshop on Irrigation Scheduling*, Rome, Italy.)

Man, K. F., Tang, K. S. and Kwong, S. (1996). “Genetic algorithms: Concepts and applications.” *IEEE Transactions on Industrial Electronics*, 43 (5), 519 -534.

Manz, D.H. (1988). “Terminology for describing on-farm irrigation water demands”. *Journal of irrigation and drainage Engineering*, 114(2), 353-358.

Medaglia, A. L., and Gutiérrez, E. (2006a). JGA: An Object-Oriented Framework for Rapid Development of Genetic Algorithms. IN: Rennard, J.P., ed. *Handbook of Research on Nature Inspired Computing for Economics and Management*.

Medaglia, A. L., and Gutiérrez, E. (2006b). Applications of JGA to Operations Management and Vehicle Routing. IN: Rennard, J.P., ed. *Handbook of Research on Nature Inspired Computing for Economics and Management*.

Merriam, J. L. (1987a). “Introduction to the need for flexibility and automation”. *Planning, Operation, Rehabilitation and Automation of Irrigation Water Delivery Systems*, Proceedings of a Symposium, ASCE, USA, 1-17.

Merriam, J. L. (1987b). “Demand irrigation schedules”. *Planning, Operation, Rehabilitation and Automation of Irrigation Water Delivery Systems*: *Proceedings of a Symposium, ASCE, USA*, 68-71.

Merriam, J. A. (1992a). "The need on the farm for a flexible water supply schedule." *Conference Proceeding, European Committee for Water and Resource Management, the centre for Irrigation Engineering*, Belgium.147-156.

Merriam, J. L., Styles, S. W., and Freeman, B. J. (2007). "Flexible irrigation systems: Concept, design, and application." *J. Irrig. Drain. Eng.*, 133(1), 2–11.

Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*, Third, Revised and Extended Edition, Springer.

Michalewicz, Z., Dasgupta, D., Riche, R.G.L. and Schoenauer, M. (1996). " Evolutionary algorithms for constrained engineering problems". *Computers ind. Engng.*, 30(4), 851-870.

Min, L. and Cheng W. (2006). "Genetic algorithm for the optimal common due date assignment and the optimal scheduling policy in parallel machine earliness/tardiness scheduling problems". *Robotics and computer-integrated manufacturing*, 22, 279-287.

Mishra, A., Singh, R., and Raghuwanshi, N. S. (2005). "Development and application of an integrated optimization-simulation model for major irrigation projects". *J. Irrig. Drain. Eng.*, 131(6), 504-513.

Monem, M.J. and Namdarian, R. (2005). "Application of simulated annealing (SA) techniques for optimal water distribution in irrigation canals". *Irrig. and Drain.* 54: 365–373.

Montana, D., Brinn, M., Moore, S., and Bidwell, G. (1998). "Genetic algorithms for complex, real-time scheduling." *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, IEEE*, 3, 2213-2218.

Mujumdar, P.P. (2002). "Mathematical tools for irrigation water management , an overview". *Water International*, 27 (1), 47-57.

Norman, B.A. and Bean , J.C. (1997a). "Random keys genetic algorithm for job shop scheduling". *Eng. Des. Automat.* 3(2), 145–156.

Norman, B.A. and Bean , J.C. (1997b). "Operation sequencing and tool assignment for multiple spindle CNC machines". *IEEE Conf. Evolutionary Computation*, 425 - 429.

Norman, B.A. and Bean, J.C. (1999). "A genetic algorithm methodology for complex scheduling problems". *Naval Research Logistics*, 46, 199-211.

Norman, B.A. and Bean , J.C. (2000). "Scheduling operations on parallel machine tools". *IIE Transactions*, 32, 449-459.

Prasad, A. S., Umamahesh, N. V. and Viswanath, G. K. (2006). "Optimal irrigation planning under water scarcity". *J. Irrig. Drain. Eng.*, 132(3), 228-237.

Radhakrishnan, S. and Ventura, J.A. (2000). "Simulated annealing for parallel machine scheduling with earliness-tardiness penalties and sequence-dependent set-up times". *Int. J. Prod. Res.*, 38(10), 2233-2252.

Randhawa, S. U. and Kuo, C. H. (1997). "Evaluating scheduling heuristics for non-identical parallel processors". *Int. J. Prod. Res.,* 35 (4), 969-981.

Rardin R.L. and Uzsoy R. (2001). "Experimental evaluation of heuristic optimization algorithms: A tutorial". Journal of Heuristics, Kluwer Academic Publishers 7,261-304.

Reddy, J.M., Wilamowski, B., and Sharmasarkar, F. C. (1999). "Optimal scheduling of irrigation for lateral canals." *ICID J.*, 48(3), 1-12.

Renault, D., Facon, T. and Wahaj, R. (2007). "Modernizing irrigation management- the MASSCOTE approach". *FAO Irrigation and Drainage paper 63,* ISBN 978-92-5-105670-7.

Replogle, J. A. (1987). "Irrigation water management with rotation schedules". *Planning, Operation, Rehabilitation and Automation of Irrigation Water Delivery Systems*: *Proceedings of a Symposium, ASCE, USA,* 35-44.

Replogle, John A. and Merriam, John L. (1980). "Scheduling and Management of Irrigation Water Delivery Systems." *Proceeding of the ASAE 2nd National Irrigation Symposium.*112-126.

Richardson, J. T., Palmer, M. R., Liepins, G. and Hilliard, M.(1989). "Some guidelines for genetic algorithms with penalty functions". *In Proc. 3rd Int. Conf. Genetic Algorithms,* Morgan Kaufmann, 191–197.

Rijsberman, F. R. (2006). "Water scarcity: fact or fiction?". *Agricultural Water Management,* 80, 5-22.

Ruiz, R., Maroto, C. and Alcaraz, J. (2006). "Two new robust genetic algorithms for the flowshop scheduling problem". *Omega (The International Journal of Management Science),* 34, 461-476

Sadegheih, A. (2006). "Scheduling problem using genetic algorithm, simulated annealing and the effects of parameter values on GA performance." *Applied Mathematical Modeling.* 30, 147-154.

Savenije, H.H.G. (2000). "Water scarcity indicators; the deception of the numbers". *Phys. Chem. Earth (B),* 25(3), 199-204.

Seckler, D., Barker, R. and Amarassinghe, U. (1999). “Water scarcity in the twenty first century”. *International Journal of Water Resources Development*, 15(1/2), 29-42.

Sethanan, K. (2001). “*Scheduling flexible flowshops with sequence dependent setup times*”. Ph.D. Thesis. College of Engineering and Mineral Resources, West Virginia University, USA.

Shiklomanov, I.A. (1999). *World Freshwater Resources (World Water Resources and their Use)*. [online]. St. Petersburg. Available from: http://webworld.unesco.org/ihp_db/publications/GenericView.asp?KEY=449 [Accessed on 30/10/2007]

Singh, B., Boivin, J. Kirkpatrick, G. and Hum, B. (1992). “Automatic Irrigation Scheduling System (AISSUM): Principles and applications”. *J. Irrig. Drain. Eng.*, 121(1), 43-56.

Smith, L.E.D. (2004). “Assessment of the contribution of irrigation to poverty reduction and sustainable livelihoods”. Water Resources Development, 20(2), 243–257.

Smout, I. K., and Gorantiwar, S. D. (2005). “Multilevel approach for optimizing land and water resources and irrigation deliveries for tertiary units in large irrigation schemes. I: Method”. *J. Irrig. Drain. Eng.*, 131(3), 254–263.

Smout, I. K., Gorantiwar, S. D. (2006a). “Improving allocation of irrigation water in southwest India”. *Water Management*, 159(2), 95-101.

Smout, I. K., and Gorantiwar, S. D. (2006b). “Productivity and equity of different irrigation schedules under limited water supply”. *J. Irrig. Drain. Eng.*, 132(4), 349-358.

Sourd, F. (2005). “Earliness-Tardiness scheduling with setup considerations.” *Computer & operations Research*, 32, 1849-1865.

Snyder, L.V. and Daskin, M.S. (2006). “A random-key genetic algorithm for the generalized traveling salesman problem”. *European Journal of Operational Research*, 174, 38-53.

Spedding, C.R.W. (1980). “Prospects and limitations of operational research application in agriculture –agrobiological systems”. *Operations Research in Agriculture and Water Resources*, 67-77.

Suman, B., and Kumar, P. (2006). “A survey of simulated annealing as a tool for single and multiobjective optimization”. *Journal of the Operational Research Society.*, 57, 1143–1160.

Suryavanshi, A.R. and Reddy, J.M. (1986). “Optimal operation schedule of irrigation distribution systems.” *Agri. Water Manag.*, 11, 23-30

Taha, H. A. (2007). *Operations research: An introduction,* 8th ed. Pearson Education, Inc., New Jersey, USA.

Tamaki, H., Nishino, E. and Abe, S. (1999). "A genetic algorithm approach to multi-objective scheduling problems with earliness and tardiness penalties". *Proceedings of the 1999 Congress on Evolutionary Computation (CEC)*, 1, 46-52.

Tsang, E.P.K. (1995). "Scheduling techniques - a comparative study". *BT Technology Journal,* 13(1), 16-28.

Valente, J.M.S., Goncalves, J.F. and Alves, R.A.F.S. (2006). "A hybrid genetic algorithm for the early/tardy scheduling problem". *Asia-Pacific Journal of operational research,* 23(3), 393-405.

Wall, M.B. (1996). "*A Genetic Algorithm for Resource-Constrained Scheduling*". Ph.D. Thesis. Department of Mechanical Engineering, Massachusetts Institute of Technology, USA.

Wang, C.S. and Uzsoy, R. (2002). "A genetic algorithm to minimize maximum lateness on a batch processing machine". *Computers & Operations Research,* 29, 1621-1640.

Wang, Z., Reddy, J.M, and Feyen, J. (1995). "Improved 0-1 programming model for optimal flow scheduling in irrigation canals." *Irrig. And Drain. Syst.*, 9,105-116.

Wardlaw R. and Bhaktikul K., (2004). "Comparison of Genetic Algorithm and Linear Programming Approaches for Lateral Canal Scheduling". *Journal of Irrigation and Drainage Engineering*, ASCE 130(4), 311-317.

Whitley, L. D. (1994). "A genetic algorithm tutorial." *Statistic and Computing*. 4(2), 65- 85.

Whitely, D., Rana, R. and Heckendorn, R. (1997). Representation issues in neighborhood search and evolutionary algorithms. IN: Quagliarelli, D., Periaux, J., Poloni, C. and

Winter, G. *Chapter3: Genetic Algorithms in Engineering and Computer Science,* John Wiley & Sons Ltd.

Winston, W.L. (2004). *Operations research, application and algorithms.* International Student Edition, Brooks/Cole, Thomson Learning, Inc. Canada. ISBN 0-534-42362-0.

Yagiura, M. and Ibaraki, T. (1996). "Metaheuristics as robust and simple optimization tools". *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC,* 541-546.

Yeniay, O. (2005). “Penalty function methods for constrained optimization with genetic algorithms”. *Mathematical and Computational Applications,* 10(1), 45-56.

Zhu, Z. and Heady, R. B. (2000). “Minimizing the sum of earliness/tardiness in multimachine scheduling: a mixed integer programming approach”. *Computers & Industrial Engineering*, 38, 297- 305.

Zimbelman, D.D. and Bedworth, D.D. (1983). “Computer control for irrigation canal system”. *J. Irrig. Drain. Eng.,* 109(1), 43-59.

Zolfaghari, S. and Liang, M. (2002). “Comparative study of simulated annealing, genetic algorithms, and tabu search for solving binary and comprehensive machine-grouping problems”. *Int. J. Prod. Res.,* 40 (9), 2141-2158.