A Case Study of Scheduling Storage Tanks Using a Hybrid Genetic Algorithm

K. P. Dahal, G. M. Burt, J. R. McDonald, and A. Moyes

***Abstract*—This paper proposes the application of a hybrid genetic algorithm (GA) for scheduling storage tanks. The proposed approach integrates GAs and heuristic rule-based techniques, decomposing the complex mixed-integer optimization problem into integer and real-number subproblems. The GA string considers the integer problem and the heuristic approach solves the real-number problems within the GA framework. The algorithm is demonstrated for three test scenarios of a water treatment facility at a port and has been found to be robust and to give a significantly better schedule than those generated using a random search and a heuristic-based approach.**

***Index Terms*—Genetic algorithms, heuristic approach, scheduling, storage tanks.**

I. Introduction

A. Problem Domain

Scheduling problems such as resource allocation, job-shop scheduling, equipment utilization, and process scheduling occur in a broad range of industries including chemical processing, batch processing, oil refinery, and water treatment facilities. Solving these scheduling problems is important for the economic operation of facilities and has been studied in the literature. Example problems include scheduling of crude oil supply to a refinery [1], inventory management of a refinery [2], scheduling of multiproduct plants consisting of a sequence of stages [3], scheduling of jobs through a chemical processing tankline [4], and scheduling of pumping pipelines [5].

A real-life problem involving the short-term scheduling of the filling and emptying of tanks in a ballast water treatment facility at a port is considered here. The layout of this facility is shown in Fig. 1. During a given scheduling horizon, ships with ballast water arrive at a berth to take on a cargo of oil. Ships berthing at a jetty must, however, discharge their contaminated ballast water before they can take on cargo. If ships cannot discharge ballast water due to some facility constraints, they must wait until discharging is possible and the ship operators have the right to charge demurrage costs for the time the vessel waits to deballast.

Ballast water can be pumped at each jetty station from the ship through the ballast pipeline to one of a number of receipt tanks. The ballast water is then left in the tanks to settle, thereby allowing the oil and water to separate, before the remaining oily-water is run down through further treatment facilities via a run-down line. Therefore, a cycle of the operation of a tank consists of empty, filling, settling, stationary, and running-down stages. There may be several fillings and several settlings in a cycle of the operation of a tank. In order to maximize the water quality, the running-down rate should be at a minimum. In addition, the subsequent treatment facility demands continuous and steady flow of the oily-water.

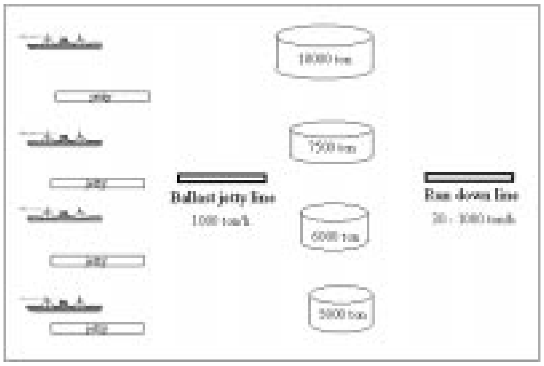

Fig. 1. Overview of a ballast water treatment facility.

The solution of the reported problem involves determining the details of an unloading schedule for the ships, as described later, allocating tanks to fill up the ballast water of the ships, determining the rates at which this should be done, and allocating tanks for running down the oily-water and determining the rates at which this should be done. This requires minimizing delays to ships, maximizing water quality by minimizing the running-down rate, and ensuring a continuous and steady supply of oily-water to the subsequent process. Furthermore, the solution must satisfy physical and operating constraints of the facility. This represents a complex constrained mixed-integer combinatorial optimization problem.

This scheduling problem is typical of those posed in chemical facilities and oil refineries [1]–[5] and is vital to solve in order to operate the facility optimally.

B. Solution Techniques

For many real-world resource management problems, the material flow process yields to the application of many traditional techniques such as mathematical programming [1], linear programming-based branch and bound [2], and Bender decomposition [3]. Other such applications have been tackled with heuristic methods in an attempt to incorporate domain-specific knowledge as in [4] and [5].

The main drawback of the mathematical programming techniques is that the number of combinations of states that must be searched increases exponentially with the size of problem and becomes computationally prohibitive (the "curse of dimensionality" [6]). Furthermore, these techniques are poor in handling the nonlinear objective and constraint functions and several assumptions are generally required to make the problem solvable using reasonable computational resources [4], [5]. Alternatively, some heuristic-based techniques use a trial-and-error method to

Manuscript received December 20, 1999; revised June 21, 1999, June 20, 2000, and December 14, 2000.

The authors are with the Centre for Electrical Power Engineering, University of Strathclyde, Glasgow G1 1XW, U.K.

Publisher Item Identifier S 1089-778X(01)03460-9.

1089–778X/01$10.00 © 2001 IEEE

evaluate the objective function in the time interval under examination. This is troublesome and time consuming as it requires significant operator input [4].

In order to overcome the above limitations, a wide range of complex scheduling problems have been attempted by using genetic algorithms (GAs) [7]–[9]. GA-based approaches have been employed to solve job shop scheduling [10], [11], production planning [12]–[15], task allocation [16], [17], and timetabling [18] problems.

GAs capable of dealing with dynamic changes in the problem environment have been developed using efficient decoding [10], combining with heuristic [11], and employing special encoding and operators [17]. Large-scale problems have been decomposed into smaller subproblems to apply the GA-based approach more effectively [12], [18]. A multiplier updating method has been incorporated with GAs to handle the problem constraints in [13], whereas a nonlinear penalty approach has been employed in [16]. Problem-specific encoding and operators have been employed in some GA solutions [14], [15]. Other applications have coupled simulation models with GA-based optimization components in an attempt to improve solution quality [19], [20].

These reported applications present GA-based approaches for scheduling problems in industries such as manufacturing [10], [13], [14], [16], agriculture [12], [19], and steelworks [20]. There is, however, little reported research work on the application of these approaches to scheduling problems in processing facilities.

Previous work [21] presented a GA approach to a single test problem relating to the water treatment facility reported here. This paper demonstrates the improved performance of a hybrid GA/heuristic approach as detailed later by considering a wider problem set.

The proposed GA-based technique decomposes the problem into integer and continuous (real-number) problem elements. The GA string characterizes the integer problem whereas the real-number problems are solved within the GA framework by using the rule-based heuristic component. The results obtained by using the technique for the test scenarios are promising and are better than those found using a heuristic approach based on common operational practice and those found using a random search technique.

The paper is organized as follows. Section II gives a general problem statement and mathematical model in the form of a mixed-integer programming problem for the above-described problem. Section III gives an overview of the proposed approach. The test problems are described in Section IV. The performance of the solution technique and the obtained results are discussed in Section V, while conclusions are noted in Section VI.

II. Optimization Problem Formulation

A. Problem Statement

Given the configuration of the facility as well as the arrival times and contents of ships and the equipment capacity limitations (refer to Fig. 1), the problem is one of determining a ship-unloading, tank-filling, and tank-emptying schedule. The ship-unloading schedule defines the start time, rates, and completion time of the unloading of ships. The tank-filling schedule defines the allocation and rates for the filling of tanks, whereas the tank-emptying schedule defines the allocation and rates for the running down of tanks. This is subject to the given configuration of the facility and the arrival times and contents of ships. This problem is representative of a tightly constrained scheduling problem characterized by daily arrival of material, processing of the material, and feeding of the material to meet a continuous demand, for example, oil refinery [1], [2], [5], chemical facilities [4], manufacturing industries [14], and steelworks [20]. The following objective and constraints apply.

Objective: Minimize the actual operating cost of the facility and maximize the product (ballast water) quality. The objective function, therefore, includes actual costs associated with waiting time for ships and penalty costs associated with the quality of running down water, specifically for excessive and nonuniform of the run-down rates.

Constraints:

1) Physical limits and operational rules of the facility.
2) Ship unloading rules.
3) Tank-filling rules.
4) Tank running-down rules.

B. Mathematical Model

This section presents the optimization problem as a mixed-integer programming problem. The proposed scheduling model is based on a uniform discretization of time in the given scheduling horizon. A complete list of the notation used is given and the objective and constraints of the optimization problem are set out.

1) Notation:

a) Indices:

$i = 1, 2, \ldots, N$	Tank.
$j = 1, 2, \ldots, J$	Ship.
$k = 0, 1, \ldots$	Filling stage within a cycle.
$s = 1, 2, \ldots$	Cycle of tank operation.
$t = 1, 2, \ldots, T$	Time.

b) Initial conditions for the facility:

qRp	Running-down rate of the facility at the end of the previous scheduling period (ton/h).
τ_i	Elapsed settling time for tank i at the start (h).
$VT_{i,0}$	Volume of ballast water in tank i at start (ton).

c) Initial conditions for ships:

TA_j	Time when ship j is ready for unloading after arrival at the jetty station.
$VS_{j,TAj}$	Volume of ballast water in ship j at arrival (ton).

d) Parameters:

c_{RD}	Cost per unit excess run-down rate (cost units·h/ton).
cw_j	Waiting cost of ship j per unit excess time (cost units/h).
c_{nu}	Cost associated with nonuniformity of run-down rate of the facility (cost units·h/ton).
ds_i	Settling time for tank i (h).
$qB\max$	Discharging capacity of ballast jetty line (ton/h).

$qR\max$	Maximum running-down rate for the facility (ton/h).
$\Delta qR\max$	Maximum rate of change of run-down rate for the facility (ton/h^2).
$qR\min$	Minimum running-down rate for the facility (ton/h).
$qs\max_j$	Maximum unloading rate for ship j (ton/h).
Δt	Discrete time interval (h).
$VT\max_i$	Maximum volume of tank i (ton).

e) Decision variables:

$FW_{i,t}$	0–1 variable indicating if tank i is in filling stage at time t.
qR_t	Run-down rate of the facility at time t (ton/h).
$qs_{j,t}$	Discharge rate of ship j at time t (ton/h).
$RW_{i,t}$	0–1 variable indicating if tank i is in running-down stage at time t.

f) General variables:

iF_t	Index of filling up tank at time t.
iR_t	Index of running-down tank at time t.
$K\max_{i,s}$	Number of filling stages within cycle s for tank i.
qB_t	Discharge rate through ballast line at time t (ton/h).
$S\max_i$	Number of cycles of operation for tank i in schedule.
$TFF_{i,k,s}$	Start time of kth filling of tank i in cycle s.
$TFL_{i,k,s}$	Completion time of kth filling of tank i in cycle s.
$TRF_{i,s}$	Running-down start time of tank i in cycle s.
$TRL_{i,s}$	Running-down completion of tank i in cycle s.
TYF_j	Unloading start time for ship j.
TYL_j	Unloading completion time for ship j.
$VS_{j,t}$	Volume of ballast water in ship j at time t (ton).
$VT_{i,t}$	Volume of ballast water in tank i at time t (ton).

2) Formulation of Objective and Constraints: The objective of this problem can be defined as minimizing the sum of costs associated with the waiting time for ships due to the filling constraints, costs involved with excessive run-down rates, and costs associated with the nonuniformity of the run-down rates. The problem, therefore, can be stated as

$$\text{minimize} \sum_j cw_j \left(TYL_j - TA_j - \left\lceil \frac{VS_{j,TA_j}}{qs\max_j} \right\rceil \right) + \sum_t (c_{RD}(qR_t - qR\min) + c_{nu}|qR_t - qR_{t-1}|) \tag{1}$$

where $\lceil\ \rceil$ corresponds to round-up to an integer value and $qR_0 = qRp$ is the run-down rate of the facility at the start of the scheduling period. The first summation in (1) relates to the operating costs of the facility, while the second summation relates to the quality of the oily-water taken from the facility.

The following group of constraints relates to the physical limits and operational rules of the facility as a whole.

1) Only one tank can be connected to the ballast jetty line at a time

$$\sum_i FW_{i,t} \le 1, \qquad \forall t. \tag{2}$$

2) Only one tank can be in the running-down stage at a time

$$\sum_i RW_{i,t} = 1, \qquad \forall t. \tag{3}$$

3) The tank being filled must not be in the running-down stage

$$FW_{i,t} + RW_{i,t} \le 1, \qquad \forall i, \forall t. \tag{4}$$

4) A tank cannot be filled to more than its capacity

$$VT_{i,t} \le VT\max_\mathrm{i}, \qquad \forall \mathrm{i}, \forall \mathrm{t}. \tag{5}$$

5) A tank must stand stationary for at least the given settling time after filling up and before running-down

$$\begin{aligned} TRF_{i,s} &\ge TFL_{i,K\max_s,s} + ds_i, \qquad \forall i,\ s = 2, \ldots, S\max_\mathrm{i}, \\ TRF_{i,1} &\ge ds_i - \tau_i, \qquad \forall i. \end{aligned} \tag{6}$$

6) The running-down rate of the facility must be greater than the specified minimum value

$$qR_t \ge qR\min, \qquad \forall t. \tag{7}$$

7) A tank must remain in the running-down stage for a minimum length of time to empty its contents

$$TRL_{i,s} \ge TRF_{i,s} + \left\lceil \frac{VT_{i,TRF_{i,s}}}{qR\max} \right\rceil, \qquad \forall i,\ s = 1, 2, \ldots, S\max_\mathrm{i}. \tag{8}$$

8) The instantaneous change in the running-down rate must be less than the specified limit

$$\begin{aligned} |qR_t - qRp| &\le \Delta qR\max, \qquad t = 1, \\ |qR_t - qR_{t-1}| &\le \Delta qR\max, \qquad t = 2, \ldots, SCH. \end{aligned} \tag{9}$$

9) A tank continues to run down until it is emptied

$$\sum_{t=TRF_{i,s}}^{TRL_{i,s}} qR_t.\Delta t = VT_{i,TRF_{i,s}}, \qquad \forall i,\ s = 1, 2, \ldots, S\max_\mathrm{i}. \tag{10}$$

The following group of constraints relates to the ship unloading rules.

1) A ship can unload its ballast water only after a certain time has elapsed from its arrival at the jetty station

$$TYF_j \ge TA_j, \qquad \forall j. \tag{11}$$

2) The duration of unloading for a ship is bounded by the initial volume of ballast water in the ship divided by the maximum unloading rate

$$TYL_j \ge TYF_j + \left\lceil \frac{VS_{j,TAj}}{\min(qs\max_j,\ qB\max)} \right\rceil, \qquad \forall j. \tag{12}$$

3) The "first come, first serve" principle applies for the ship unloading. The first ship to arrive discharges at the highest possible discharge rate

$$qs_{j,t} = \min\left(qs\max_j,\ qB\max - \sum_{m=1}^{j-1} qs_{m,t},\ \frac{VS_{j,t}}{\Delta t} \right), \qquad \forall j,\ t = TYF_j, \ldots, TYL_j. \tag{13}$$

4) The ballast line discharge rate is the sum of the discharge rates of the ships unloading at a given time

$$qB_t = \sum_j qs_{j,\,t}, \qquad \forall t. \tag{14}$$

5) A ship must unload all of its ballast before leaving the jetty station

$$\sum_{t=TYF_j}^{TYL_j} qs_{j,\,t}\Delta t = VS_{j,\,TA_j}, \qquad \forall j. \tag{15}$$

6) The volume of ballast water in a ship at each time is the volume at the previous time less the volume of the ballast water transferred from the ship in that time interval

$$\begin{aligned} VS_{j,\,t} &= VS_{j,\,t-1} - qs_{j,\,t}\Delta t, \\ &\forall j,\, t = TYF_j, \ldots, TYL_j. \end{aligned} \tag{16}$$

The following group of constraints relates to the tank-filling rules.

1) The start time for each filling of a tank in a particular cycle of operation is less than its completion time

$$\begin{aligned} TFF_{i,\,k,\,s} &< TFL_{i,\,k,\,s}, \\ &\forall i,\ s = 1, 2, \ldots, S\mathrm{max_i}, \\ &k = 1, 2, \ldots, K\mathrm{max}_{i,\,s}. \end{aligned} \tag{17}$$

2) The filling start time of a tank in a particular cycle of operation is greater than the completion time of the previous filling in that cycle

$$\begin{aligned} TFL_{i,\,k,\,s} &< TFF_{i,\,k+1,\,s}, \\ &\forall i,\ s = 1, 2, \ldots, S\mathrm{max_i}, \\ &k = 1, 2, \ldots, K\mathrm{max}_{i,\,s} - 1. \end{aligned} \tag{18}$$

3) The start time of the first filling of a tank in a cycle is greater than the running-down completion time of the tank in the previous cycle of operation

$$TFF_{i,\,1,\,s} > TRL_{i,\,s-1}, \qquad \forall i,\ s = 2, \ldots, S\mathrm{max_i}. \tag{19}$$

The following group of constraints relates to the tank running-down rules.

1) Material balance must be satisfied at all times

$$\begin{aligned} VT_{i,\,t} &= VT_{i,\,t-1} + \sum_j qs_{j,\,t} FW_{i,\,t}\Delta t - qR_t RW_{i,\,t}\Delta t, \\ &\forall i,\, \forall t. \end{aligned} \tag{20}$$

2) The running-down rate of the facility must be less than the specified upper limit

$$qR_t \le qR\,\mathrm{max}, \qquad \forall t. \tag{21}$$

3) The volume of the contents of a storage tank cannot be less than zero

$$VT_{i,\,t} \ge 0, \qquad \forall i,\, \forall t. \tag{22}$$

It is clear from the above that this formulation represents a nonlinear mixed-integer programming problem, which cannot be solved using conventional mathematical programming methods within a reasonable computational time.

III. Proposed Solution Approach

A. Decomposition of Problem

The proposed GA-based solution technique decomposes the scheduling problem into three subproblems as shown in Fig. 2. Subproblem 1 is modeled as an integer-programming problem, the solution to which consists of an allocation of tanks for filling up and for running down at each time step. Subproblems 2 and 3 are continuous (real-number) programming problems. Subproblem 2 uniquely determines the ship unloading schedule using constraints represented by constraints (5), (11)–(16), and (20), the ship arrival details, and the allocation of tanks for filling up given by the solution to subproblem 1. In addition to this, subproblem 2 also uniquely determines the filling rates for the allocated tanks using the calculated ship unloading schedule and constraints (5), (14), and (17)–(20). Similarly, subproblem 3 calculates the running-down rates using constraints (6)–(10), (19)–(22), and the allocation of tanks for running down given by the solution to subproblem 1. This does not define a unique solution to subproblem 3; the strategy adopted is to implement the maximum possible change in run-down rates as early as possible.

The decomposition of this problem allows the use of an integrated approach as a solution technique. Subproblem 1 is a combinatorial problem and as such represents the most difficult among the three subproblems. Subproblem 1, therefore, forms a natural target for a GA approach. Consequently, the GA solution string adopted here consists of only the decision (integer) variables for subproblem 1, while subproblems 2 and 3 are solved during the evaluation value determination using heuristic approaches.

The algorithms for the heuristic solution of subproblems 2 and 3 have been successfully developed from well-established operational procedures discussed below. In general, however, such knowledge may be difficult to obtain. Specialist domain knowledge associated with operational practices can be captured by adopting a structured approach [22] prior to implementation within the heuristic-based component. Indeed, through the conduct of a series of knowledge elicitation meetings with experienced operators and the preparation of structured transcripts, the operational rules, constraints, requirements, and performance of the water treatment facility were identified.

B. GA Implementation

As mentioned above, the GA string represents the solution of subproblem 1, that is, an allocation of tanks for filling up and running down for all time steps. Two generic types of representation can be used to encode a candidate solution for subproblem 1: binary or integer.

With a binary representation, the binary variables $FW_{i,\,t}$ and $RW_{i,\,t}$ can be encoded explicitly using a one-dimensional binary array. The size of the GA search space with this type of representation is $2^{2\times N\times T}$. The size of the feasible solution space, however, is comparatively small.

The number of variables and the size of the GA search space can be reduced if the indices of the tanks in the filling up stage (iF_t) and in the running-down stage (iR_t) are encoded in the

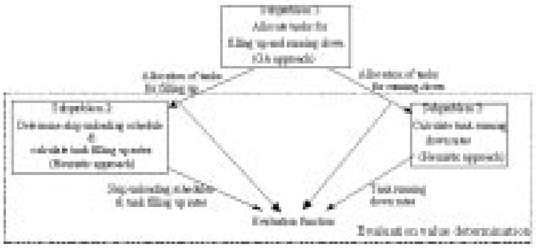

Fig. 2. Decomposition of the tank scheduling problem and structure of GA evaluation function.

solution string. This approach allows the use of an integer representation to explicitly encode integer variables (iF_t and iR_t) in the GA solution string. The integer representation significantly reduces the length of the GA solution string from the $2 \times N \times T$ bits required for the binary representation to $2 \times T$ integers. This integer encoding has been implemented in this work. Each scheduling interval is represented by two integers, one identifying the tank being filled and the other identifying the tank being run down as given by

$$iF_1, iR_1, iF_2, iR_2, \ldots, iF_{T-1}, iR_{T-1}, iF_T, iR_T.$$

This type of representation automatically satisfies constraints (2) and (3) and reduces the GA search space to $N^T(N+1)^T$, though this still includes many infeasible solutions.

The quality (fitness) of the schedule represented by the GA string is calculated by an evaluation function, which is formulated using the results of all three subproblems as shown in Fig. 2. Here, the GA string and, hence, the solution of subproblem 1, satisfies constraints (2) and (3). Solutions to subproblems 2 and 3 satisfy constraints (11)–(19) and (20)–(22), respectively.

The remaining constraints of the problem are considered by introducing penalty functions in the evaluation function. These penalty functions take into account not only the fact that constraints are violated, but also the degree of those violations by using linear functions. The constraints are divided into hard and soft constraints. The hard constraints described by (4)–(7) represent physical limits and strict operational rules of the scheduling problem. The constraints (8)–(10) are taken as the soft constraints. In addition, the evaluation function includes penalty functions for nonpreferred operation of the tanks. For example, it is preferred to have a tank remain in the filling-up stage until it is full and to avoid topping up a settled tank.

The evaluation value for a GA string is the weighted sum of the objective value, the penalty value for the violation of the soft and hard constraints and the penalty value for the nonpreferred operation of tanks as given by

$$\begin{aligned} evaluation = {} & OBJ + \sum_x \omega_{hx} \times HCV_x + \sum_y \omega_{sy} \\ & \times SCH_y + PNO \end{aligned} \tag{23}$$

where

OBJ objective value in (1);

HCV_x measure of xth hard constraint violation;

SCV_y measure of yth soft constraint violation;

PNO summation of values which penalize the nonpreferred operation of tanks;

ω_{hx} and ω_{sy} weighting coefficients for xth hard constraint and yth soft constraint, respectively.

These weighting coefficients are chosen so that the violation of the hard constraints gives greater penalty values than the soft constraints and that both are generally greater than the objective values and penalty values for the nonpreferred operation of tanks. The resulting evaluation value of an infeasible solution is therefore higher than those of comparable feasible solutions. The evaluation value of a genetic string gives an inverse indication of the overall quality of a solution represented by the genes: the lower the evaluation value of a string, the better its quality. For feasible solutions, the evaluation value is the sum of the objective value and the penalty values for nonpreferred operation (if any).

C. GA Architecture

The proposed GA-based technique has been developed in a staged way, starting from a basic GA framework. This approach allows features of other solution techniques, problem domain knowledge and suitable GA parameters to be identified and used within the GA framework.

Two alternative GA models have been designed in this work. The first model, GA-1, is a simple GA [7], [8] whereas the second model, GA-2, involves restricted operators. GA-1 generates the initial population pool of candidate solutions by sampling the search space at random. During each iteration, a tournament selection operator is used to choose a parent solution from the current population. The tournament selection method picks a subset of solutions at random from the population to form a tournament selection pool, from which one solution is selected with a probability based upon the measured quality of the solutions. The selected solutions are then subjected to crossover with a defined crossover probability (CP). Standard single and multiple point crossover operators have been employed that exchange sections between these two selected solutions with probability CP. One of the resulting solutions is then chosen to undergo mutation. A standard random mutation operator has been used to change the integer at each position in the solution within the allowed range with a defined mutation probability (MP). The algorithm is terminated when a defined stopping criterion is reached.

It is possible to start the GA from an initial population that is generated by considering some domain knowledge. If a stochastic or heuristic approach for generating reasonable solutions is known, then it may be used to generate the initial population before the GA is invoked. This can be an effective way of accelerating, and sometimes improving, the performance of the GA [9].

For this problem, constraint (4), which describes that a tank filling up cannot be in the running-down stage $(iF_t \neq iR_t)$, could be respected during the generation of initial solutions. Indeed, the GA-2 model uses this knowledge in generating the initial population. However, if the standard crossover and mutation operators described above were then employed, this constraint might be violated subsequently during the GA reproduction process. Therefore, restricted crossover and mutation operators have been designed that produce new solutions which do not violate constraint (4).

The GA-2 model fills the initial population pool of candidate solutions at random (as in the case of GA-1), but then applies a repair operator to each of the solutions to make sure that it does not violate constraint (4). This is achieved by generating a new integer value of variable iR_t for constraint violation time t. This allows the GA to focus in the area of the solution space, where constraint (4) is not violated; thus, only infeasible solutions are ignored. The restricted crossover operator exchanges sections between solutions and then checks for violation of constraint (4). If this constraint is violated at time t, then the value of iF_t is taken from the parent from whom it was not originally inherited. The restricted mutation operator changes an integer at each time t with a new random integer that is within the allowed range and ensures that $iF_t \neq iR_t$. The constraints included within the restricted operators were selected in view of the ease with which they can be implemented and their effectiveness in reducing the GA search space. In general, GA methods using these restricted operators have increased computational time per trial (i.e., fitness evaluation), but are more robust than those using standard operators (robust in terms of consistently finding feasible and improved solutions) [9]. The reported work has demonstrated this by exhibiting slower computational time per trial and more focused searching.

Both the generational (GN) and steady-state (SS) population updating approaches have been tested with GA-1 and GA-2. The GN approach replaces the population of the previous generation by a new population of solutions. The SS GA inserts a new solution directly into the population pool replacing a less-fit individual, which is adopted here using a tournament replacement operator similar to the tournament selection operator. The elitist approach, which ensures that the best individual in the population pool is always retained, has been applied in all cases.

The particular GA parameters that give the best performance for the approach are typically identified after a process of experimentation. An adaptive crossover and mutation operator can be employed to reduce the amount of experimentation required to find the favorable GA parameters [7]. However, this is not implemented in this case in order to more transparently assess the sensitivity of the proposed approach to different GA parameters. The GAs have been implemented on a Sun Sparcstation 1000 using the Reproductive Plan Language, RPL2 [23].

IV. Case Study Description

The proposed solution technique has been applied to three test scenarios of a facility involving four tanks and representing most of the features of a genuine problem (refer to Fig. 1). Four ships are to be unloaded during the scheduling period of 24 hours. The ship, tank, and ballast line and running-down line details are given in Tables I–III, respectively. These realistic problem scenarios were provided by a facility specialist for formulating test cases.

In order to test realistic situations that may occur in the facility, three different initial conditions of the tanks are considered. The paper will refer to the test problem with each of these three situations as test problems A, B, and C. Two ship arrival patterns are considered: test problems A and B share the same ship arrival pattern. The ship details and the initial conditions of the facility for these test problems are presented in Tables I and III, respectively.

Test problem A represents a tight scheduling situation for the facility, when the available free space in the tanks is smaller than the ballast volume receipts and there is not much choice of tank selection for filling up and running down at the start of the scheduling period. In test problem B, there is a greater choice of tank selection for filling up as two tanks are empty at the start of the scheduling period. Test problem C presents the greatest choice of the three problems for running down, since the contents of three tanks of the facility have fully settled at the start of the scheduling period $(\tau_i = ds_i)$.

TABLE I
Arrival Patterns, Contents, Maximum Discharging Rates, and Demurrage Costs of Ships

j	TA_j		$VS_{j,TAj}$	$qsmax_j$	cw_j
	Test problems A and B	Test problem C	(ton)	(ton/h)	(units/h)
1	1	19	3000	325	9.00
2	7	1	8200	650	24.60
3	13	9	2890	650	8.67
4	23	17	1400	325	4.20

TABLE II
Ballast Line Capacity and Limits and Rate of Change of Running Down Rates

$qBmax$	$qRmin$	$qRmax$	qRp	$\Delta qRmax$
ton/h	ton/h	ton/h	ton/h	ton/h^2
1000	30	1000	300	100

TABLE III
Capacities and Settling Times, Initial Contents, and Elapsed Settling Times of Tanks

Tank details			Test problem A		Test problem B		Test problem C	
i	$VTmax_i$	Ds_i	$VT_{i,0}$	τ_i	$VT_{i,0}$	τ_i	$VT_{i,0}$	τ_i
	(ton)	(h)	(ton)	(h)	(ton)	(h)	(ton)	(h)
1	10000	6	4650	0	0	0	4650	6
2	7500	6	3600	6	3600	6	3600	6
3	6000	6	6000	4	0	0	0	0
4	5000	6	4800	2	4800	2	4800	6

The following factors were used in the formulation of the objective function of the problem for all three test problems: cost per unit excess run-down rate $c_{RD} = 0.0035$ units·h/ton and cost associated with nonuniformity of run-down rate $c_{nu} = 0.015$ units·h/ton.

V. Test Results and Discussion

A. Scope

The size of the GA search space for these test problems is the same and large. Test problem A has the smallest feasible solution space of the three problems, as it is more tightly constrained. In this sense, test problem A is the hardest among the three problems and a detailed analysis of the performance of the algorithm with the different approaches, operators, and parameters for this test problem is presented here. Comparisons were made of the performances of GA-1 and GA-2 models using GN and SS population updating approaches and varying key GA parameters. The GA parameters that were varied include the CP, MP, the number of crossover points, and population size (PS). For test problems B and C, the variation of results for crossover and mutation probabilities are described. The remaining design parameters are taken to be those giving the best performance for test problem A—the experimentation for test problems B and C with these design parameters have been shown to give reasonable results.

The general approach adopted during experimentation was to conduct 30 GA runs with the selected parameters and identify the number of runs in which a feasible solution was found. A percentage success rate was thereby determined. The amount of infeasibility over 30 GA runs can also be used as a performance measurement of the GA approach (in selected test cases this gave similar results). The "percentage success rate" together with the best evaluation value in the final population is used to assess the relative merit of alternative GAs.

The total number of trials (i.e., fitness evaluations) for each run was fixed to 112 500, which was determined by analysis of the convergence of the GA technique after a number of experiments. The computational time taken for one GA run was typically 250 s. The size of the tournament selection/replacement pools was fixed to ten.

B. GA Performance Analysis

1) Test Problem A: The first set of experiments compared the ability of the GN and SS GAs to find feasible solutions to test problem A. These experiments were conducted with various GA operators and parameters. The success rates obtained using GN and SS GAs for the GA-1 model with standard two-point crossover and mutation operators are summarized in the first two blocks of Table IV and labeled GN GA-1 and SS GA-1, respectively. The CP and MP were varied in the range of 0.2–1.0 and 0.005–0.1, respectively, while the PS was fixed at 150. Table IV indicates that both GN GA-1 and SS GA-1 are sensitive to variation of the operator probabilities. In most of the cases, the performance of SS GA-1 is also shown to be better than that of GN GA-1 in terms of the success of finding feasible solutions to the problem. As a result, only the SS GA was considered for further experiments.

TABLE IV
PERCENTAGE SUCCESS RATES FOR TEST PROBLEM A WITH THE TWO-POINT CROSSOVER OPERATOR WITH DIFFERENT GA MODELS

MP	.005	.01	.05	.1	.005	.01	.05	.1	.005	.01	.05	.1
CP	GN GA-1				SS GA-1				SS GA-2			
0.2	0	0	6.7	3.3	6.7	6.7	20	6.7	10	23.3	36.7	16.7
0.4	0	3.3	6.7	10	10	13.3	20	13.3	13.3	13.3	26.7	20
0.6	0	3.3	20	3.3	10	23.3	20	23.3	10	20	23.3	20
0.8	0	3.3	3.3	3.3	10	10	6.7	10	3.3	23.3	16.7	33.3
1.0	3.3	6.7	13.3	3.3	6.7	13.3	16.7	13.3	10	20	20	26.7

The next set of experiments was intended to analyze the performance of the SS GA for the GA-2 model with the restricted operators for population initialization, two-point crossover, and mutation operators. The experiments were done for the same variation in CP and MP as in the previous cases. The obtained results are summarized in the third block of Table IV and labeled SS GA-2. Comparing the two SS GA results shown in Table IV, SS GA-2 model generally gives a better performance.

Experiments were carried out to observe the effect of varying the number of crossover points on the performance of the GA. Tests were done using SS GA-2 model with the restricted one-, three-, and four-point crossover operators with CP and MP in the range of 0.2–1.0 and 0.01–0.1, respectively. Crossover points introduce disruptions to solutions since the location of genes ("locus") defines the sequence of time periods in the representation adopted here. It has been observed that with a smaller number of crossover points a higher MP is required to provide a similar level of disruption. On the basis of the observed success rates, the GA with the one-point crossover operator was found to give a better performance. The best performance was obtained with this operator and CP = 0.2 and MP = 0.1, when 12 out of 30 GA runs found feasible solutions.

Finally, the SS GA-2 model was applied to the test problem with varied PSs in the range of 50 to 200 using the one-point crossover with CP = 0.2 and MP = 0.1. With a lower PS, the GA was found to converge prematurely at a local optimum due to the lack of diversity in the population, whereas with the higher PS, the convergence was slow. The results show that the previously used PS, PS = 150, gives the best success rate for the given number of trials and size of tournament selection/replacement pools.

The architecture of the GA that gave the best success rate is identified in the first portion of Table V. With this GA architecture, a success rate of 40% was obtained, which is better than those shown in Table IV. The lower half of Table V contains performance measures relating to the experiment that identified the lowest evaluation value. The data listed shows the minimum evaluation value (relating to a feasible solution) to be 69.43. For comparison, the average evaluation value in the final population is also listed, as are the average and minimum evaluation values in the initial population.

TABLE V
ARCHITECTURE AND PERFORMANCE MEASURES OF THE BEST PERFORMING GA WITH TEST PROBLEM A

GA model	SS GA-2
Parent selection method	Tournament
Crossover Operator	One-point
Crossover probability	0.2
Mutation probability	0.1
Population size	150
Evaluation value of the best solution in final population	69.43
Objective value of the best solution in final population	67.68
Average evaluation value in final population (over 30 exp.)	4629.31
Average evaluation value in initial population (over 30 exp.)	64247.57
Average evaluation value of the best solutions in initial population (over 30 exp.)	32500

TABLE VI
PERCENTAGE SUCCESS RATE FOR TEST PROBLEMS B AND C

MP	.01	.05	.1	.01	.05	.1
CP	Test problem B			Test problem C		
0.2	20	26.7	30	53.3	63.3	83.3
0.4	26.7	33.3	33.3	46.7	76.7	80.0
0.6	23.3	33.3	43.3	50.0	73.3	80.0
0.8	20	36.7	33.3	63.3	86.7	86.7
1.0	26.7	36.7	33.3	43.3	80.0	80.0

2) Test Problems B and C: For test problems B and C, the sensitivity of SS GA-2 model to the variation of crossover and mutation probabilities in the range of 0.2–1.0 and 0.01–0.1 respectively was established. Other GA features were as given in Table V. The success rates over 30 experiments are given in Table VI. The GA with CP = 0.6 and MP = 0.1 gives the best performance in terms of success rate for test problem B but a relatively poor best solution (evaluation value = 43.74). However, over the 30 runs, the best solution (evaluation value = 33.69) was found with CP = 0.6, MP = 0.05. For test problem C, the best solution (evaluation value = 52.56) and the best success rate were found for higher values of CP and MP as indicated in Table VI.

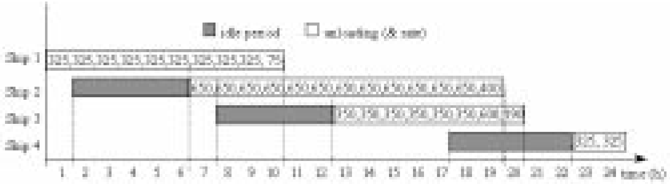

Fig. 3. Ship unloading schedule given by GA-based and heuristic approaches for test problems A and B.

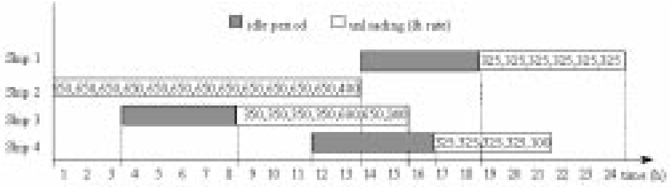

Fig. 4. Ship unloading schedule given by GA-based and heuristic approaches for test problem C.

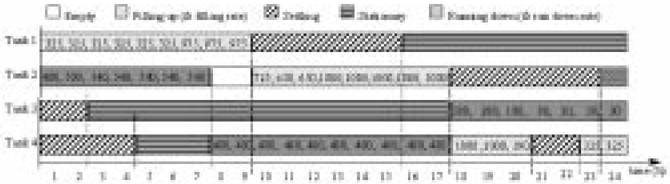

Fig. 5. Operation of tanks according to the best GA solution for test problem A.

C. Schedules from GA-Based Approach

The ship unloading schedules given by the best GA solutions for test problems A, B, and C are illustrated in Figs. 3 and 4. The ship unloading schedule (Fig. 3) is identical for test problems A and B. This is not altogether surprising, as the ship arrival pattern for these test problems is the same, and the solution gives the shortest waiting times for the ships and, hence, the lowest possible demurrage costs. Similarly, the best GA ship unloading schedule for test problem C with a different ship arrival pattern (Fig. 4) also gives the shortest waiting times for the ships. This indicates the validity of the presented approach to problems with a different ship arrival pattern.

The schedule for the operation of tanks given by the best GA-based solution (evaluation value = 69.43) for test problem A is depicted in Fig. 5. The numerical values for the tank filling-up rates and tank running-down rates are also shown in this figure. The schedule for the operation of tanks given by the best GA-based solution (evaluation value = 33.69) for test problem B is illustrated in Fig. 6 with numerical values for the tank filling-up rates and tank running-down rates. For test problem C, the schedule for the operation of tanks given by the best GA-based solution (evaluation value = 52.56) is depicted in Fig. 7.

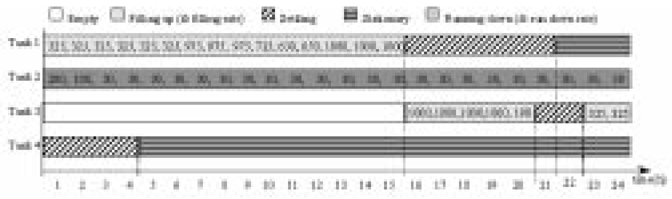

Fig. 6. Operation of tanks according to the best GA solution for test problem B.

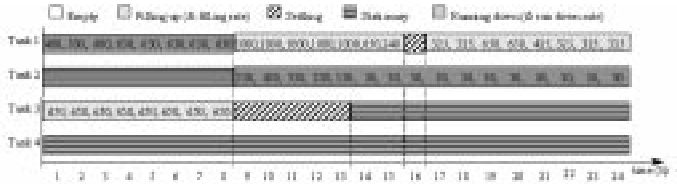

Fig. 7. Operation of tanks according to the best GA solution for test problem C.

D. Comparative Techniques

For comparison's sake, two solution techniques, namely, a random search and a heuristic method based on operational practice, have been developed.

1) Random Search Technique: The random search technique has been simulated using a GA without crossover operation and with a high MP (CP $= 0$ and MP $= 0.5$). It has been employed specifically to test problem A. After 112 500 trials, no feasible solutions had been found; the best solution found was infeasible and had an evaluation value of 649.63. This is not surprising given the large search space and small feasible solution space.

2) Schedules from the Heuristic Approach: For the three test problems, comparative heuristic schedules of filling and emptying tanks have been developed using common operational practice in order to compare with the schedules given by the GA. The operation of the facility is based on heuristic rules that focus mainly on unloading ships as early as possible and maintaining a constant run-down rate of the ballast water: if there is a choice in selecting a tank for filling up, the smallest tank that will take the complete contents of a ship (if possible) is chosen, leaving larger tanks free to receive the next ship-load of ballast; the heuristic selects the smallest settled tank for running down, in order to have an empty tank available earlier; the strategy employed for changing the run-down rates is to implement the maximum possible change as early as possible. This heuristic algorithm has been derived from the conduct of a knowledge elicitation exercise with facility operation experts and is detailed below. The approach is iterative and uses a backtracking mechanism until a feasible solution is found. The algorithm is as follows.

1) Determine the ship unloading plan considering the "first come, first served" principle, maximum unloading capacity of ships, and discharging capacity of ballast jetty line.
2) For each ship discharging time period, identify the tank currently being filled-up (if not, identify the tank with sufficient, but smallest, space) and calculate the filling-up rate for the identified tank whilst considering ship discharging rates and space available in the selected tank. If a tank is not identified for the current time period, go to step 3; otherwise, go to step 4.
3) Set the current time period as the "empty time" for the tank that is currently being run down and go to step 2.
4) For each time period, identify the tank currently being run down (if not, identify the smallest settled tank) and calculate the run-down rate for the identified tank whilst considering content of the tank, maximum change allowed, and "empty time" (if any). If a tank is not identified for the current time period, go to step 5; otherwise go to step 6.
5) Backtrack to the time period when the tank currently running down first started emptying and reduce its run-down rate then continue from step 4.
6) If a feasible solution is found or maximum possible attempts reached, stop; otherwise, change ship unloading plan by constraining ship discharging rates and go to step 2.

The schedule for the operation of the tanks given by the heuristic approach for test problem A is depicted in Fig. 8. The heuristic solution is feasible and has an evaluation value of 73.94. The heuristic solution for test problem B gives the schedule for the operation of tanks depicted in Fig. 9. The schedule is feasible and has an evaluation value of 48.28. For test problem C, the heuristic schedule is illustrated in Fig. 10, which is feasible and has an evaluation value of 55.79.

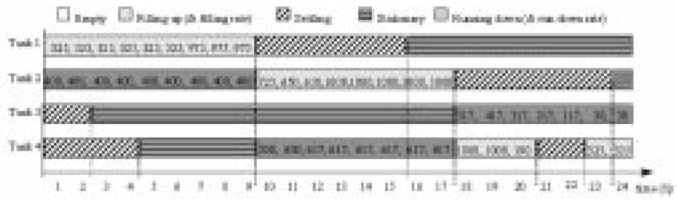

Fig. 8. Operation of tanks according to the heuristic approach for test problem A.

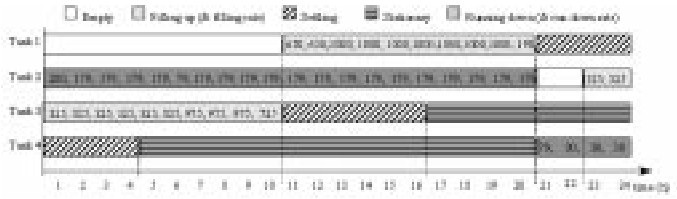

Fig. 9. Operation of tanks according to the heuristic approach for test problem B.

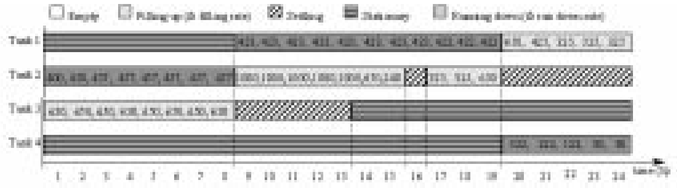

Fig. 10. Operation of tanks according to the heuristic approach for test problem C.

TABLE VII
COMPARISON OF COSTS FOR THE HEURISTIC AND THE BEST GA SCHEDULES FOR THE TEST PROBLEMS

Test Problems		A		B		C	
Schedules		Heuristic	GA-based	Heuristic	GA-based	Heuristic	GA-based
Costs for	ship waiting times	26.01	26.01	26.01	26.01	17.34	17.34
	run-down rates	32.63	30.42	10.67	0.84	28.90	20.69
	Nonuniformity of run-down rates	13.55	11.25	4.05	4.05	8.76	13.95
Objective values		72.20	67.68	48.28	30.90	55.00	51.98
Evaluation values		73.94	69.43	48.28	33.69	55.79	52.56

E. Comparison of Schedules

The heuristic solution of the test problems result in the same ship unloading schedule as the best GA-based solutions shown in Fig. 3 for test problems A and B and in Fig. 4 for test problem C. This is because the same ship unloading rules have been embedded into the GA framework and the heuristic approach to derive the ship unloading schedule for the given filling up tanks. The best GA-based and heuristic solutions both give the shortest waiting times for the ships and, hence, the lowest possible costs associated with the ship waiting time (demurrage costs).

The schedule for the operation of tanks in the GA schedule (Fig. 5) with test problem A can be seen to differ slightly from the heuristic plan (Fig. 8). The selection and time allocation for filling-up tanks in both schedules are the same. The heuristic schedule allocates the first nine time intervals for tank 2 and the next eight time intervals for running down tank 4, whereas the GA schedule allocates the first seven intervals for tank 2 and the next ten time intervals for tank 4. As the volume of contents to be run down in tank 4 (4800 ton) is greater than that in tank 2 (3600 ton), the allocation of more time for running down the larger volume makes the GA-based schedule better than the heuristic schedule in terms of costs associated with the run-down rates and nonuniformity of run-down rates. This improvement is demonstrated by the fact that the average run-down rate over the scheduling period in the heuristic schedule is 418.6 ton/h in comparison with 388.8 ton/h in the GA solution. Furthermore, the costs associated with the nonuniformity of the run-down rates are lower in the GA schedule than that in the heuristic schedule.

In the best GA-based solution for test problem B (Fig. 6), the selection and time allocation for both filling up and running down tanks are different from the heuristic schedule (Fig. 9). For filling up, the heuristic schedule selects tank 3 (the smallest empty tank), then tank 1, and finally tank 2. The GA selects tank 1 and then tank 3 for filling up, which allows tank 2 to be selected for run down in the entire scheduling period, hence reducing the run-down rates to the minimum value after the first two time intervals. The heuristic schedule empties tank 2 in the first 21 time intervals before allocating it for filling up. The GA schedule is again better than the heuristic schedule, in this case in terms of the average run-down rate over the scheduling period. As there is enough free space available in the tanks for the given scheduling period, there is no rush to run down the contents of the tanks.

For test problem C, the best GA-based solution (Fig. 7) gives a different allocation of tanks for both filling-up and running-down than that given by the heuristic schedule (Fig. 10). For filling up, both solutions initially select tank 3 (the empty tank), then after the heuristic schedule selects, then tank 2, and finally tank 1. For running down, the heuristic schedule selects tank 3 (smallest settled tank), then tank 1, and finally tank 4. The GA empties tank 1 in the first eight time periods, which allows tank 1 to be selected for filling up and tank 2 to be allocated for running down in the remaining scheduling period. In this way, the GA schedule results in less tank switching operations and reduction in the average run-down rates over the scheduling period. Although the costs associated with the nonuniformity of run-down rates for the GA schedule are higher than those of the heuristic schedule, the overall costs for the GA schedule (51.98) is better than those for the heuristic schedule (55.00).

The costs of the best GA-based solutions and the heuristic solutions for each test problem are quantified in Table VII. The numerical values in Table VII show that the schedules found by the GA-based approach are better than the heuristic schedules developed using the current operational practice for all three test problems. As described earlier, test problem A represents a tight scheduling situation for the facility in that there is not much choice of tank selection for filling up and running down at the start of the scheduling period. The search space for the test problem is very large, while the feasible solution space appears to be characterized by small disconnected regions containing a few troughs. In such a tight situation, it is a significant achievement of the GA to obtain a better schedule than the heuristic schedule.

Test problem B provides more flexibility in terms of selecting filling-up tanks making the space of feasible solutions larger than that for test problem A. The number of troughs in the feasible space for test problem B would appear to be greater than that for test problem A. The GA-based approach found a significantly better solution than the heuristic schedule for test problem B.

Test problem C has more flexibility than test problems A and B in terms of selecting running-down tanks and, hence, a larger feasible search space. The GA is unsurprisingly more consistent in finding feasible solutions for test problem C (Table VI) than for test problems A and B. The heuristic technique similarly finds a feasible solution for problem C, although the solution was poorer than that found by the GA-based approach.

Unlike other solution techniques, the GA-based technique works with a population of solutions and offers a set of solutions instead of a single final solution. This highlights an advantage in this application. If the best solution is no longer appropriate due to some reasons, for example, flexibility in operating conditions, the GA-based technique provides alternatives (such as solutions which violate soft constraints) in its final set of solutions which may be judged and selected instead. Although no operational experience has been gained with this proposed approach, its merits have been demonstrated for the selected case studies.

The experimental results for the test problems considered provide some evidence to the general applicability of the proposed GA approach. That is, for problems of a similar nature the GA approach adopted will be of relevance in obtaining their solution. The proposed hybrid approach can be useful to solve an optimization problem with mixed variables. This type of optimization problem can be decomposed into a combinatorial (decision) problem and a continuous (real number) problem. This allows the construction of a GA string to represent the combinatorial problem while the continuous problem is solved using known heuristic rules or a suitable technique within the GA-framework. GA-heuristic hybrids can take full advantage of both these approaches as demonstrated for the test problems considered.

VI. Conclusion

Three realistic test scenarios for scheduling ship unloading and tank filling and emptying in a ballast water treatment facility have been presented and the application of a GA-based technique to solve the problems demonstrated. The solution technique uses an "integrated" approach in which the GA string represents the allocation of tanks for filling up and for running down and a rule-based approach is used to calculate the ship unloading rates and the tank run-down rates within the evaluation function. The GA string has been encoded using integers. The evaluation function has been constructed incorporating objectives and penalties, the latter for both constraint violation and undesirable operation of the facility.

The sensitivity of the GA method to different population updating approaches, operators, and parameters has been established. Restricted GA operators have been implemented that always respect an essential problem constraint for the generation of initial solutions and during the reproduction process. The GA model with these restricted operators has been found to be more robust than the GA model with standard operators for a wide range of GA parameters. Tests have shown that high-quality solutions can be found if an appropriate architecture of the GA is selected for the problem.

Furthermore, comparisons have been made between the GA-based approach, a random search, and a heuristic method based on current operational practice for different problem scenarios with tightly constrained and flexible situations. It has been shown that the GA-based approach finds a better schedule with a lower and more uniform run-down rate. Although the GA-based approach is not guaranteed to find the global optimal solution, it is a significant achievement to obtain a good solution to a complex problem, like that discussed above, in a reasonable computational time. The results demonstrate that the GA-based approach forms the basis of an effective scheduling tool.

Acknowledgment

The authors would like to thank S. Burchell of BP for providing the realistic problem scenario to formulate the test problem. The authors also acknowledge the use of the Reproductive Plan Language, RPL2, produced by Quadstone Limited, in the conduct of this work. This work was performed in the Rolls-Royce University Technology Centre in Power Engineering at the University of Strathclyde, Glasgow, U.K.

References

[1] N. Shah, "Mathematical programming techniques for crude oil scheduling," *Comput. Chem. Eng.*, vol. 20, pp. s1227–s1232, 1996.

[2] H. Lee, J. M. Pinto, I. E. Grossmann, and S. Park, "Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management," *Ind. Eng. Chem. Res.*, vol. 35, no. 5, pp. 1630–1641, 1996.

[3] J. M. Pinto and I. E. Grossmann, "Optimal cyclic scheduling of multistage continuous multiproduct plants," *Comput. Chem. Eng.*, vol. 18, no. 9, pp. 797–816, Sept. 1994.

[4] R. L. Storch, W. Song, and Z. B. Zabinsky, "Scheduling a chemical processing tankline," in *Advances in Production Management Systems*, I. A. Pappas and I. P. Tatsiopoulos, Eds. Amsterdam, The Netherlands: Elsevier, 1993, pp. 457–468.

[5] M. Sasikumar, P. V. Prakash, S. M. Patil, and S. Ramani, "PIPES: A heuristic search model for pipeline schedule generation," *Knowl.-Based Syst.*, vol. 10, no. 3, pp. 169–175, Oct. 1997.

[6] R. G. Parker, *Deterministic Scheduling Theory*. London, U.K.: Chapman & Hall, 1995.

[7] T. Bäck, D. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*. Cambridge, U.K.: Oxford Univ. Press, 1997.

[8] T. Bäck, D. Fogel, and Z. Michalewicz, *Evolutionary Computation 1: Basic Algorithms and Operators*. London, U.K.: Inst. of Phys., 2000.

[9] ——, *Evolutionary Computation 2: Advanced Algorithms and Operators*. London, U.K.: Inst. of Phys., 2000.

[10] C. Bierwirth and D. C. Mattfeld, "Production scheduling and rescheduling with genetic algorithms," *Evol. Comput.*, vol. 7, no. 1, pp. 1–17, Spring 1999.

[11] E. Hart and P. Ross, "A heuristic combination method for solving job-shop scheduling problems," in *Parallel Problem Solving from Nature (PPSN) V*, A. E. Eiben, T. Bäck, and H.-P. Schwefel, Eds. Berlin, Germany: Springer-Verlag, 1998, pp. 845–854.

[12] E. Hart, P. Ross, and J. Nelson, "Solving a real-world problem using an evolving heuristically driven schedule builder," *Evol. Comput.*, vol. 6, no. 1, pp. 61–80, Spring 1998.

[13] Y.-C. Lin, K.-S. Hwang, and F.-S. Wang, "Plant scheduling and planning using mixed-integer hybrid differential evolution with multiplier updating," in *Proc. Congr. Evolutionary Computation*, July 2000, pp. 593–600.

[14] N. Morad and A. M. S. Zalzala, "Integrated production planning and scheduling in cellular manufacturing using genetic algorithms," in *Proc. 2nd Int. Conf. Genetic Algorithms in Engineering Systems: Innovations and Applications*, Sept. 1997, pp. 332–338.

[15] K. J. Shaw and P. J. Fleming, "Including real-life problem preferences in genetic algorithms to improve optimization of production schedules," in *Proc. 2nd Int. Conf. Genetic Algorithms in Engineering Systems: Innovations and Applications*, Sept. 1997, pp. 239–244.

[16] A. B. Hagj-Aloune, J. C. Bean, and K. G. Murty, "A hybrid genetic/optimization algorithm for a task allocation problem," *J. Schedul.*, vol. 2, no. 4, pp. 189–201, June 1999.

[17] S. Rana-Stevens, B. Lubin, and D. Montana, "The air crew scheduling system: The design of a real-world, dynamic genetic scheduler," in *Proc. Genetic and Evolutionary Computation Conf.*, July 2000.

[18] E. K. Burke and J. P. Newall, "A multistage evolutionary algorithm for timetable problem," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 63–74, Apr. 1999.

[19] R. P. S. Hart, M. T. Larcombe, R. A. Sherlock, and L. A. Smith, "Optimization techniques for a computer simulation of a pastoral dairy farm," *Comput. Electron. Agriculture*, vol. 19, no. 2, pp. 129–153, Jan. 1998.

[20] R. J. Paul and T. S. Chanev, "Optimising a complex discrete event simulation model using a genetic algorithm," *Neural Comput. App.*, vol. 6, no. 4, pp. 229–237, 1997.

[21] K. P. Dahal, C. J. Aldridge, J. R. McDonald, and G. M. Burt, "A GA-based technique for the scheduling of storage tanks," in *Proc. 1999 Congr. Evolutionary Computation*, July 1999, pp. 2199–2206.

[22] A. Moyes, G. M. Burt, J. R. McDonald, J. R. Capener, J. N. Dray, and R. Goodfellow, "Combining Design and Operational Knowledge to Enhance Generator Plant Diagnostics," *Proc. Inst. Electr. Eng.- Generation, Transmission and Distribution*, vol. 143, pp. 300–304, 1996.

[23] *Reproductive Plan Language, RPL2, User Manual*, Quadstone Ltd., Edinburgh, U.K., 1997.